Владимир Дронов

Macromedia Dreamweaver MX

Наиболее полное руководство

В ПОДЛИННИКЕ®

Владимир Дронов

Macromedia Dreamweaver MX

Санкт-Петербург
«БХВ-Петербург»
2003

УДК 681.3.06
ББК 32.973.26-018.2
Д75

Дронов В. А.

Д75 Macromedia Dreamweaver MX. — СПб.: БХВ-Петербург, 2003. — 736 с.: ил.

ISBN 5-94157-190-9

Книга посвящена описанию визуального Web-редактора Macromedia Dreamweaver MX. Изложение материала ведется от простого к сложному, начиная со способов создания примитивных Web-страничек и заканчивая основами построения сложных сайтов, включая серверное программирование и работу с базами данных. Представлены приемы, позволяющие наиболее эффективно применять для этих целей Dreamweaver MX. Приводятся примеры использования мощных средств этого Web-редактора, описаны создаваемые им теги HTML и коды JavaScript, даются рекомендации по предупреждению возможных ошибок. Рассмотрены особенности Web-дизайна, включая табличный и фреймовый его виды. Книга написана простым языком в увлекательном стиле, что позволяет даже неискушенному в Web-технологиях читателю быстро освоить новейшие возможности Dreamweaver MX.

Для Web-дизайнеров и Web-программистов

УДК 681.3.06
ББК 32.973.26-018.2

Группа подготовки издания:

Главный редактор	*Екатерина Кондукова*
Зам. главного редактора	*Евгений Рыбаков*
Зав. редакцией	*Анна Кузьмина*
Редактор	*Алексей Кожедуб*
Компьютерная верстка	*Ольги Сергиенко*
Корректор	*Зинаида Дмитриева*
Дизайн обложки	*Игоря Цырульникова*
Зав. производством	*Николай Тверских*

Лицензия ИД № 02429 от 24.07.00. Подписано в печать 30.01.03.
Формат 70×100$^{1}/_{16}$. Печать офсетная. Усл. печ. л. 59,34.
Тираж 5000 экз. Заказ №
"БХВ-Петербург", 198005, Санкт-Петербург, Измайловский пр., 29.

Гигиеническое заключение на продукцию, товар № 77.99.02.953.Д.001537.03.02
от 13.03.2002 г. выдано Департаментом ГСЭН Минздрава России.

Отпечатано с готовых диапозитивов
в Академической типографии "Наука" РАН
199034, Санкт-Петербург, 9 линия, 12.

ISBN 5-94157-190-9

© Дронов В. А., 2003
© Оформление, издательство "БХВ-Петербург", 2003

Содержание

Введение **13**

ЧАСТЬ I. КАК СДЕЛАТЬ ПРОСТЕЙШИЙ WEB-САЙТ **15**

Глава 1. Как создаются Web-страницы **17**

Что такое Интернет 17
Как создаются Web-страницы 18
Гиперссылки 23
Клиенты и серверы Интернета 25
Зачем нужны HTML-редакторы 29
Что дальше? 31

Глава 2. Основные принципы работы с Dreamweaver **32**

Среда Dreamweaver MX 32
- Выбор рабочей среды 33
- Главное окно программы 34
- Управление окнами и панелями Dreamweaver 40

Работа с Web-страницами 42
- Работа в окне документа 42
- Три режима отображения Web-страницы 46
- Работа с кодом HTML 48
- Поиск и замена текста 54
- Использование регулярных выражений 58
- Просмотр Web-страницы 62
- Вызов справки 63

Настройка Dreamweaver 65
- Учим русский 66
- Настраиваем скорость интернет-соединения 70
- Добавляем программы просмотра Web-страниц 70
- Добавляем внешний HTML-редактор 73

Что дальше? 74

Глава 3. Начинаем с текста 75

Создание новой Web-страницы 75
Работа с текстом 76
Ввод текста 76
Форматирование абзацев 78
Форматирование отдельных символов 83
Вставка специальных символов 92
Использование HTML-стилей 98
Работа с гиперссылками 102
Создание обычных гиперссылок 102
Создание почтовых гиперссылок 105
Другие гиперссылки 106
Создание гиперссылок на FTP-серверы 107
Новости (USENET) 107
Использование "якорей" 108
Общие свойства Web-страницы 110
Дополнительные возможности Dreamweaver 113
Использование цветовых схем 113
Вставка и чтение комментариев 115
Вставка даты 116
"Чистка" HTML-кода 118
Проверка совместимости HTML-кода 119
Что дальше? 121

Глава 4. Рисунки, звуки, фильмы 122

Работа с графическими изображениями 123
Два вида графических изображений 124
Вставка графического изображения 126
Изображения-гиперссылки 132
Активные изображения 133
Карты-изображения 135
Графика Macromedia Flash 140
Графика Macromedia Shockwave 146
Фоновые изображения 146
Мультимедиа 148
Поддержка мультимедийных данных 149
Модули расширения 151
Элементы ActiveX 155
Что дальше? 158

Глава 5. Таблицы 159

Текст фиксированного формата 160
Простые таблицы 163
Создание таблиц 163
Работа с таблицами 165
Формирование таблиц 168

Форматирование таблиц ... 169
Выделение элементов таблиц ... 169
Параметры ячейки ... 171
Параметры строки ... 173
Параметры таблицы ... 174
Предопределенные форматы таблиц ... 177
Сортировка таблицы ... 179
Вставка табличных данных ... 182
Слияние ячеек таблиц ... 184
Использование таблиц ... 188
Текст в рамке ... 188
Текст в графической рамке ... 192
Текст с отступами ... 195
Сложные таблицы ... 198
Составные изображения ... 204
Проблемы с таблицами и их решение ... 206
Общие недостатки таблиц и их преодоление ... 206
Проблемы с таблицами в старых версиях Navigator ... 207
Что дальше? ... 210

Глава 6. Работа с Web-сайтом ... 211

Начала сайтостроения ... 212
Планирование сайта ... 212
Основные этапы планирования сайта ... 212
Логическая структура Web-сайта ... 213
Физическая структура Web-сайта ... 216
Публикация Web-сайта ... 219
Управление Web-сайтом в Dreamweaver ... 221
Регистрация сайта в Dreamweaver ... 221
Настройка прокси-сервера или брандмауэра ... 227
Панель *Site* ... 229
Работа с файлами сайта ... 233
Взаимодействие панели *Site* и окна документа ... 236
Абсолютные и относительные интернет-адреса ... 237
Проверка правильности ссылок и HTML-кода ... 240
Работа с сервером ... 244
Публикация сайта ... 244
Работа с копиями сайта ... 249
Дополнительные возможности работы с Web-страницами и Web-сайтами ... 250
Примечания ... 250
Активы ... 255
Список избранных элементов ... 257
Библиотека ... 259
Дополнительные возможности поиска и замены ... 263
Другие возможности ... 263
Что дальше? ... 264

ЧАСТЬ II. ЗАНИМАЕМСЯ ПРОФЕССИОНАЛЬНЫМ WEB-ДИЗАЙНОМ 265

Глава 7. Фреймы 267

Введение во фреймы 267
Фрейм и набор фреймов 267
Схемы наборов фреймов 269
Сложные наборы фреймов 273
Работа с фреймами 274
Создание фреймов 275
Работа с фреймами и наборами фреймов 281
Свойства наборов фреймов 281
Свойства фреймов 284
Замещение и работа с ним 286
Создание содержимого фреймов 288
Заполнение фреймов 288
Создание остальных страниц сайта 291
Цель гиперссылки 292
Полоса навигации 293
Оптимизация фреймов 299
Уменьшение объема и сложности HTML-кода 299
Ускорение обработки фреймов 301
Решение проблем с фреймами 302
Недостатки фреймов и их преодоление 302
Проблема с фреймами в старых версиях Navigator 303
Что дальше? 305

Глава 8. Табличный дизайн 306

Основы табличного дизайна 306
Зачем нужны таблицы 306
Схемы табличного дизайна 311
Построение таблиц разметки вручную 314
Построение таблиц разметки автоматически 318
Недостатки способа создания таблиц разметки вручную 318
Режим разметки страницы Dreamweaver 319
Таблицы и ячейки разметки 323
Форматирование таблиц и ячеек разметки 329
Параметры ячеек разметки 329
Задание ширины ячеек 330
Параметры таблицы разметки 333
Заполнение начальной страницы 334
Тонкая настройка и оптимизация таблицы разметки 338
Недостатки табличного дизайна 341
Недостатки таблиц 341
Недостатки табличного дизайна 345
Что дальше? 346

Глава 9. Использование шаблонов 347

Введение в шаблоны Dreamweaver 348
Работа с шаблонами 349
Создание шаблона 349
Редактирование шаблона 351
Создание изменяемых областей 353
Создание Web-страниц на основе шаблонов 356
Применение шаблонов к уже созданным Web-страницам 360
Гиперссылки в шаблонах 363
Экспорт Web-страниц, основанных на шаблонах 365
Обновление страниц, созданных на основе шаблонов 366
Управление шаблонами в окне шаблонов 366
Новые возможности шаблонов 368
Изменяемые атрибуты 368
Необязательные области 370
Повторяющиеся области 373
Необязательные изменяемые области 377
Табличная повторяющаяся область 378
Вложенные шаблоны 380
Недостатки шаблонов и их преодоление 381
Что дальше? 382

Глава 10. Каскадные таблицы стилей 383

Введение в каскадные таблицы стилей 383
Зачем они нужны 384
Три способа задания стиля 390
Почему "каскадные" 392
Псевдостили гиперссылок 394
Работа с таблицами стилей в Dreamweaver 396
Создание стилей 397
Определение стиля 399
Параметры шрифта 400
Параметры фона 403
Параметры абзаца 404
Параметры размеров и размещения 407
Параметры рамки 412
Параметры маркеров списка 415
Параметры местонахождения 417
Дополнительные параметры 417
Применение стилей 420
Управление стилями 422
Управление таблицами стилей 424
Поддержка внутренних стилей 427
Таблицы стилей и шаблоны 429
Временные таблицы стилей 430
Недостатки таблиц стилей и их преодоление 431
Что дальше? 434

ЧАСТЬ III. ИСПОЛЬЗУЕМ НОВЕЙШИЕ ТЕХНОЛОГИИ 435

Глава 11. Свободно позиционируемые элементы 437

Введение в свободно позиционируемые элементы 438
Что такое свободно позиционируемый элемент 438
Как создается свободно позиционируемый элемент 440
Зачем нужны свободно позиционируемые элементы 445
Работа со свободно позиционируемыми элементами 446
Создание свободно позиционируемых элементов 446
Параметры свободно позиционируемых элементов 450
Работа с группой свободно позиционируемых элементов 452
Работа со свободно позиционируемыми элементами 453
Пример использования свободно позиционируемых элементов 456
Недостатки свободно позиционируемых элементов и их преодоление 462
Что дальше? 466

Глава 12. Анимация элементов Web-страниц 467

Введение в Web-сценарии 468
Ограничения HTML 468
Web-сценарии 469
События 471
Основные принципы анимации 472
Простейшая анимация 472
Анимация реального времени 474
Анимация — подход Dreamweaver 477
Зачем нужна анимация 479
Создание анимации в Dreamweaver 481
Создание простейшей анимации 481
Более сложная анимация 486
Управление анимациями 491
Анимация графических изображений 494
Недостатки анимации, основанной на Web-сценариях, и их преодоление 496
Что дальше? 498

Глава 13. Использование сценариев 499

Подробнее о Web-сценариях 500
Зачем нужны Web-сценарии 500
Язык JavaScript 502
Объекты 508
Объектная модель документа (DOM) 513
Как пишутся Web-сценарии 516
Простейший Web-сценарий 519
Более сложный Web-сценарий 520
Web-сценарии — подход Dreamweaver. Поведения 525
Работа с поведениями 529
Панель *Behaviors* 529
Создание поведений 532

Вызов JavaScript-кода *(Call JavaScript)* ... 533
Изменение значения свойства *(Change Property)* ... 533
Перенаправление на другую страницу в зависимости от версии Web-обозревателя *(Check Browser)* ... 534
Проверка наличия модуля расширения *(Check Plugin)* ... 536
Управление фильмом Shockwave или Flash *(Control Shockwave or Flash)* ... 538
Перетаскивание свободно позиционируемого элемента *(Drag Layer)* ... 539
Переход на заданный кадр анимации *(Go To Timeline Frame)* ... 542
Переход на другую Web-страницу *(Go to URL)* ... 543
Скрытие меню гиперссылок *(Hide Pop-Up Menu)* ... 543
Открытие нового окна Web-обозревателя *(Open Browser Window)* ... 543
Проигрывание аудиоклипа *(Play Sound)* ... 545
Запуск проигрывания анимации *(Play Timeline)* ... 545
Вывод предупреждения *(Popup Message)* ... 545
Предварительная загрузка графических изображений *(Preload Images)* ... 546
Изменение изображения-элемента полосы навигации *(Set Nav Bar Image)* ... 548
Задание нового содержимого фрейма *(Set Text of Frame)* ... 550
Задание нового содержимого свободно позиционируемого элемента *(Set Text of Layer)* ... 551
Вывод текста в строке статуса окна Web-обозревателя *(Set Text of Status Bar)* ... 552
Вывод на экран меню гиперссылок *(Show Pop-Up Menu)* ... 553
Показ и скрытие свободно позиционируемых элементов *(Show-Hide Layers)* ... 558
Останов проигрывания анимации *(Stop Timeline)* ... 559
Изменение графического изображения *(Swap Image)* ... 560
Восстановление всех изначальных изображений *(Swap Image Restore)* ... 561
Написание своих Web-сценариев ... 561
Особые случаи создания поведений ... 562
Привязка поведений к тексту ... 562
Поведения и анимации ... 563
Отладка Web-сценариев ... 565
Какие ошибки бывают в программах ... 565
Встроенный отладчик Dreamweaver ... 567
Альтернативные технологии ... 571
Что дальше? ... 574

Глава 14. Метатеги и серверные директивы ... 575

Реклама в Интернете ... 575
Поисковые машины ... 576
Как работают поисковые агенты ... 579
Метатеги ... 580
Пассивная интернет-реклама ... 582
Работа с метатегами в среде Dreamweaver ... 584
Описание Web-страницы ... 585
Ключевые слова ... 586
Базовый интернет-адрес ... 587

Перезагрузка ... 589
Связи между Web-страницами ... 590
Специальные метатеги ... 593
Серверные директивы ... 595
Введение в серверные директивы ... 595
Стандартный набор серверных директив ... 596
include ... 597
echo ... 598
Как использовать серверные директивы ... 598
Поддержка Dreamweaver серверных директив ... 602
Что дальше? ... 604

ЧАСТЬ IV. ПИШЕМ СЕРВЕРНЫЕ ПРОГРАММЫ ... 605

Глава 15. Введение в серверное программирование ... 607

Что такое серверное программирование ... 607
Зачем нужны серверные программы ... 607
Как Web-сервер обрабатывает данные пользователя ... 608
Как Web-обозреватель отправляет введенные данные ... 612
Как данные передаются по Сети ... 615
Серверное программирование — подход Dreamweaver ... 617
Введение в базы данных ... 619
Что дальше? ... 621

Глава 16. Формы ... 622

Работа с формами в Dreamweaver ... 622
Создание формы ... 623
Элементы управления ... 624
Поле ввода ... 625
Кнопка ... 628
Флажок ... 629
Переключатели ... 630
Группа переключателей ... 631
Список ... 633
Поле ввода имени файла ... 635
Графическая кнопка ... 636
Скрытое (невидимое) поле ... 638
Дополнительные элементы управления ... 639
Метка ... 639
Группа ... 640
Раскрывающийся список гиперссылок ... 641
Поведения, предназначенные для работы с формами ... 644
Создание списка гиперссылок *(Jump Menu)* ... 644
Создание кнопки перехода для списка гиперссылок *(Jump Menu Go)* ... 644
Задание нового значения поля ввода *(Set Text of Text Field)* ... 645
Проверка данных, введенных в форму *(Validate Form)* ... 645
Простейшая Web-форма ... 647

Использование таблиц и стилей для создания форм ... 651
Основные принципы разработки форм ... 653
Что дальше? ... 655

Глава 17. Простейшие серверные приложения ... 656

Подготовка к созданию серверных приложений ... 656
Установление соединения с базой данных ... 659
Создание источника данных ODBC ... 660
Регистрация базы данных в Dreamweaver ... 663
Создание серверных страниц в Dreamweaver ... 666
Создание простейших серверных страниц ... 666
Более сложные серверные страницы ... 670
Создание набора данных ... 670
Страница, отображающая данные ... 674
Создание навигатора ... 676
Создание строки статуса набора данных ... 678
Страница для одновременного просмотра нескольких записей ... 679
Привязка элементов управления к данным ... 682
Создание сложных наборов данных ... 684
Работа с динамическими атрибутами ... 686
Получение данных от другой Web-страницы ... 688
Создание фильтров ... 689
Необязательные области серверной страницы ... 691
Быстрое создание серверных страниц ... 693
Что дальше? ... 694

Глава 18. Создание интерактивных сайтов ... 695

Принципы создания интерактивных сайтов ... 696
Административные страницы сайта ... 697
Как администрируются Web-сайты ... 697
Страница списка высказываний ... 699
Страница добавления записи ... 701
Страница изменения записи ... 702
Средства удаления записи ... 706
Страница входа на сайт ... 708
Защита страниц от несанкционированного доступа ... 711
Реализация выхода с сайта ... 713
Страницы общего доступа ... 714
Страница списка категорий ... 714
Страница списка высказываний ... 716
Страница регистрации посетителя ... 717
Реализация поиска высказываний ... 719
Что дальше? ... 720

Заключение ... 721

Предметный указатель ... 724

Введение

Всякая книга имеет введение, где излагается, зачем же она написана и о чем рассказывает. Эта книга — не исключение. Итак...

О чем эта книга

В связи с развитием Интернета во всем мире появился спрос на профессию Web-дизайнера. *Web-дизайнер* — это человек, обладающий художественным вкусом и сведущий в интернет-технологиях, который создает Web-страницы и объединяет их в Web-сайты. Художественный вкус (а желательно и соответствующее образование) — это главное, что должен иметь Web-дизайнер (и не только Web-дизайнер, но и его коллега из области полиграфии), т. к. только знание интернет-технологий еще не служит гарантией получения качественного результата. Тем более что современные программные средства позволяют ему не вникать во всякие тонкости.

Но что делать, если вы чувствуете в себе творческий порыв, желаете самореализоваться в области Web-дизайна, но с трудом представляете, что такое HTML, CSS и JavaScript? (Конечно, знание всего этого совсем не помешает, но...) Не беда! Специально для вас разработаны программы визуальных Web-редакторов, позволяющие создавать Web-страницы и целые Web-сайты, не вникая в тонкости интернет-технологий и стандартов, так, как вы создаете документы в Microsoft Word.

Эта книга рассказывает о Macromedia Dreamweaver MX — последней версии популярного Web-редактора компании Macromedia. Это исключительно мощная программа, поддерживающая все современные стандарты Интернета и невероятно облегчающая выполнение даже самых сложных задач. Кроме того, она содержит в своем составе развитую систему подсказки и интерактивных уроков, позволяющих начинающему пользователю быстро приступить к работе. По иронии судьбы, Dreamweaver "задвинут" в тень своего

более "раскрученного" собрата Microsoft FrontPage. Но — будьте уверены! — он не заслужил такой участи.

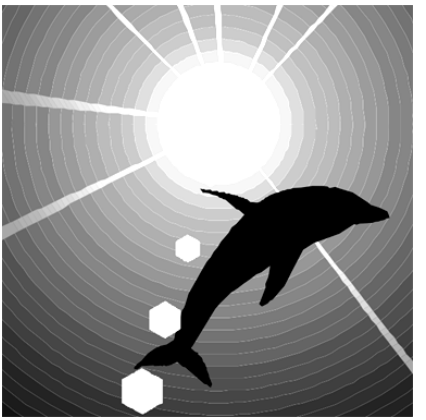
Может быть совсем и по-другому. Вы — опытный Web-дизайнер, предпочитающий все делать "врукопашную", но желающий немного автоматизировать свою работу. Dreamweaver поможет вам и в этом случае. Вы сможете быстро сделать "набросок" своей Web-страницы, а потом довести код "до ума" вручную. Поверьте, так работать гораздо удобнее — и вы быстро привыкнете.

Macromedia Dreamweaver — достаточно "старый" пакет. Первая его версия была разработана еще в 1998 году и получила популярность в России благодаря своей "благосклонности" к русскому языку и снисходительному отношению к множеству русскоязычных кодировок. Я пользуюсь Dreamweaver, начиная с версии 2.0. В прошлом году я написал книгу о предыдущей версии Dreamweaver — 4.0. Теперь же, с выходом Dreamweaver MX, я снова взялся за перо, так сказать.

Изучать Dreamweaver мы будем на конкретном примере. Мы будем делать личный Web-сайт гипотетического Web-дизайнера Ивана Ивановича Иванова. Сначала этот сайт будет совсем простеньким; на его основе мы изучим базовые инструменты программы и основные принципы Web-дизайна. Впоследствии сайт станет сложнее и красивее; мы будем использовать фреймы и таблицы, чтобы придать ему профессиональный вид. Далее мы изучим таблицы стилей, встроенные средства для анимации элементов страницы и сценарии, позволяющие "оживить" страницы нашего сайта. И на самой последней ступени мы изучим серверное программирование и сделаем для нашего сайта гостевую книгу. Ну и, конечно же, мы узнаем, как собрать разрозненные страницы в сайт и опубликовать его в Сети.

Но прежде чем начать разговор о Macromedia Dreamweaver MX, дадим несколько основных понятий и ответим на несколько вопросов, которые могут возникнуть у неподготовленного читателя. Конечно, если вы на короткой ноге с WWW, можете пропустить первую главу и начать читать сразу со второй. Но автор обязан позаботиться обо всех, кто будет читать эту книгу.

Итак, первый вопрос... Вернее, ответ. Вы найдете его в первой главе настоящей книги.



ЧАСТЬ I

Как сделать простейший Web-сайт

Глава 1. Как создаются Web-страницы

Глава 2. Основные принципы работы с Dreamweaver

Глава 3. Начинаем с текста

Глава 4. Рисунки, звуки, фильмы

Глава 5. Таблицы

Глава 6. Работа с Web-сайтом

ГЛАВА 1

Как создаются Web-страницы

Так как же делаются те красивые Web-странички, которые выводит нам Web-обозреватель? И откуда они берутся? А вот я слышал в автобусе слово "WWW" и не знаю, что это такое... И вообще, что такое Интернет и почему мы его не видим, как, например, монитор или сообщения об ошибках Windows?

Все-все! Сейчас попытаемся ответить на эти вопросы. И начнем с самого последнего.

Что такое Интернет

В самом деле, что такое Интернет? Электронный океан, таинственная стихия, заключенная в кремниевых кристаллах и медных проводах современных компьютеров. Несуществующая вселенная, иной раз кажущаяся более реальной, чем наш материальный, "настоящий" мир. То, что вторгается в каждый дом, опутывает всю планету и сознание всех людей тугой медно-кремниевой паутиной; нечто запредельное, непостижимое людскому разуму, никому не видимое, но всеми ощущаемое...

Но довольно! Вы слишком много читаете фантастики (или газет типа "Церковного вестника"). На самом деле, все намного проще. Возможно, вы даже немного разочаруетесь, узнав, что такое Интернет. Но правда должна быть сказана, чего бы это ни стоило. Ради этого и написана данная книга.

Итак, что такое Интернет и как он работает?

Вот объяснение для зануд и педантов. *Интернет* — это совокупность компьютерных сетей, связанных друг с другом и работающих по единым стандартам. Говоря простыми словами, это множество локальных сетей, плюс пользователи, подключающиеся к ним по модемам, плюс соединяющие эти сети высокоскоростные каналы связи, плюс еще пара миллионов тонн вся-

ких железяк, пара триллионов строк программного кода и пара-тройка тысяч стандартов, призванных навести в этом барахле подобие порядка.[1]

Собственно, грубое описание Интернета дано. Конечно, на самом деле все неизмеримо сложнее, но для нас сейчас достаточно этого краткого описания. Если же вы хотите знать больше, читайте специальные книги. А пока что заканчиваем этот раздел, самый краткий в настоящей книге, и переходим к следующему...

Как создаются Web-страницы

А теперь поговорим о том, как создаются Web-страницы.

Для этого используется особый язык *HTML* (HyperText Markup Language — язык гипертекстовой разметки). Этот язык определяет набор специальных команд, называемых *тегами* и используемых для задания форматирования или назначения тех или иных *элементов* Web-страницы. Особые теги используются для размещения на Web-страницах графических изображений, аудио- и видеоклипов и прочих так называемых *внедренных объектов*.

Кажется, я вас напугал. Но, несмотря на кажущуюся сложность, Web-страницы не представляют собой ничего сложного. Это обычные текстовые файлы, созданные в стандартном Блокноте или аналогичном простейшем текстовом редакторе. И содержат они текст, тот самый текст, который вы хотите поместить на страницы, только размеченный особым образом...

Да что тут говорить! Лучше все показать на примере. Взгляните на рис. 1.1. Это простейшая Web-страничка, сделанная для примера в Блокноте.

Чтобы увидеть эту страничку воочию, откройте Блокнот, наберите приведенный ниже код, проверьте его на ошибки и сохраните в файле с именем 1.1.htm. Только когда будете вводить имя файла в стандартном окне сохранения, заключите его в кавычки, иначе Блокнот добавит расширение txt, и ваш файл получит имя 1.1.htm.txt. После этого откройте полученный файл в Web-обозревателе, для чего достаточно дважды щелкнуть по нему мышью.

Теперь давайте рассмотрим сам HTML-код нашей первой странички.

```
<HTML>
  <HEAD>
    <TITLE>Web-страница</TITLE>
  </HEAD>
  <BODY>
```

[1] Говорят, в первой польской энциклопедии, изданной, кажется, в XVII столетии, термин "лошадь" описывался так: "что такое лошадь, знают все". Точно то же самое можно сейчас сказать об Интернете. (Вот только можно ли сейчас сказать то же самое о лошади?..)

```
    <H1>Пример Web-страницы</H1>
    <P>Это простейшая Web-страничка, созданная в стандартном
       <I>Блокноте</I>
    и отображенная в <I>Microsoft Internet Explorer</I>.</P>
  </BODY>
</HTML>
```

Выглядит устрашающе... Однако хорошо заметен текст, который виден в окне Web-обозревателя на рис. 1.1. Давайте рассмотрим его подробнее.

```
<H1>Пример Web-страницы</H1>
<P>Это простейшая Web-страничка, созданная в стандартном <I>Блокноте</I>
и отображенная в <I>Microsoft Internet Explorer</I>.</P>
```

Вы видите какие-то слова, заключенные в угловые скобки < и >. Это и есть теги HTML. Они задают форматирование текста. Скажем, строка `<I>Блокноте</I>` будет выведена курсивом, т. к. теги `<I>` и `</I>` задают курсивное начертание текста. Причем, тег `<I>` помечает начало курсивного фрагмента (*открывающий* тег), а тег `</I>` — конец (*закрывающий* тег). А собственно фрагмент, заключенный между открывающим и закрывающим тегами, называется *содержимым* тега.

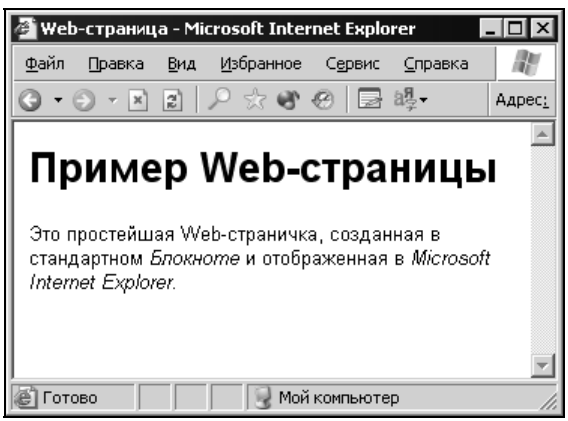


Рис. 1.1. Простейшая Web-страничка

Это были теги *физического форматирования* текста, т. е. просто задающие, как текст должен выглядеть в окне Web-обозревателя. Теперь рассмотрим теги *логического форматирования*, которые позволяют разбить текст на отдельные логические блоки. В нашем небольшом фрагменте это теги `<P>` и `<H1>` (и соответствующие им закрывающие теги `</P>` и `</H1>`). Они задают соответственно обычный текстовый параграф и заголовок первого уровня;

при этом Web-обозреватель будет знать, что <H1>Пример Web-страницы</H1> — это заголовок, и отобразит его соответствующим образом. Как и в предыдущем случае, открывающий тег помечает начало логического блока текста, а закрывающий — конец.

Здесь мы говорили о так называемых *парных* тегах, когда открывающему тегу соответствует закрывающий. HTML определяет также набор *одинарных* тегов. Один из них — это тег вставки графического изображения <IMG>.

```
<IMG SRC="picture.gif">
```

Здесь мы видим, что само графическое изображение хранится в отдельном файле picture.gif. То есть, изображение — это один из внедренных элементов Web-страницы. Тег <IMG> содержит в себе параметр SRC, задающий имя файла изображения. (Такие параметры называются *атрибутами* тега.) Web-обозреватель, встретив в HTML-коде страницы тег <IMG>, загружает файл, чье имя задано атрибутом SRC, и отображает его.

Осталось рассмотреть совсем немного тегов, используемых для служебных целей и не отображаемых Web-обозревателем. Они так и называются — *невидимые* теги.

Парный тег <BODY>...</BODY> используется для выделения *тела* Web-страницы, т. е. той ее части, которая будет отображаться в окне Web-обозревателя. Кроме тела, Web-страница также должна содержать задаваемый парным тегом <HEAD>...</HEAD> *заголовок*, где помещается служебная информация. (Этот заголовок не выводится Web-обозревателем, а используется для внутренних нужд. Не путайте заголовок Web-страницы и обычный текстовый заголовок, задаваемый тегом <H1>!) Среди этой служебной информации может быть *название* страницы, показываемое в заголовке окна Web-обозревателя; оно задается парным тегом <TITLE>...</TITLE>.

Давайте рассмотрим заголовок нашей страницы.

```
<HEAD>
  <TITLE>Web-страница</TITLE>
</HEAD>
```

Собственно, заголовок содержит только название нашей страницы, помеченное тегом <TITLE>...</TITLE>. Но в подавляющем большинстве случаев заголовок содержит множество других данных, необходимых как Web-обозревателю, так и другим программам, обеспечивающим функционирование Интернета и WWW. Мы пока не будем их рассматривать.

Взгляните еще раз на HTML-код нашей страницы. Заметьте, что одни теги вложены в другие. Так, тег <I> вложен в тег <P>, тег <P> — в тег <BODY>, а тег <BODY> — в тег <HTML>. Такая *вложенность* или *иерархия* тегов в HTML встречается на каждом шагу. Давайте представим ее в виде схемы.

```
<HTML>
  <HEAD>
    <TITLE>
  <BODY>
    <H1>
    <P>
      <I>
```

Здесь мы убрали мешающие нам закрывающие теги и их содержимое. Величина отступа показывает *уровень вложенности* того или иного тега. Так, тег <BODY> имеет первый уровень вложенности, а тег <H1> — второй. Теги предыдущих уровней вложенности называются *родительскими тегами* или *родителями*, а теги последующих уровней — *дочерними тегами* или *потомками*. Например, для тега <HEAD> родительским тегом будет <HTML>, а дочерним — <TITLE>. Для тега <BODY> родителем будет тег <HTML>, а потомками — теги <H1>, <P> и <I>.

Весь HTML-код Web-страницы вложен внутрь парного тега <HTML>...</HTML>. Этот тег находится на самом высшем, нулевом, уровне вложенности и не имеет родителя.

Я здорово перегрузил вас информацией, и вам может показаться, что HTML — это что-то запредельно сложное. На самом деле, все это очень просто, и вы можете легко создавать простейшие Web-странички вручную. И в подтверждение этого давайте немного изменим код нашей страницы. Например, выделим название фирмы Microsoft жирным шрифтом. Тег, задающий "жирность", — <B>...</B>.

Далее приведем измененный фрагмент HTML-кода.

```
<P>Это простейшая Web-страничка, созданная в стандартном <I>Блокноте</I>
и отображенная в <I><B>Microsoft</B> Internet Explorer</I>.</P>
```

Заметьте, как мы вложили один тег в другой (эти теги выделены). Теперь слово "Microsoft" будет набрано и жирным шрифтом, и курсивно. Только обратите внимание на *порядок вложенности* тегов и не нарушайте его. Если же вы случайно допустите ошибку и поместите теги так:

```
<P>Это простейшая Web-страничка, созданная в стандартном <I>Блокноте</I>
и отображенная в <B><I>Microsoft</B> Internet Explorer</I>.</P>
```

то Web-обозреватель может и не отобразить ваше творение (хотя Internet Explorer славится своим умением исправлять мелкие ошибки Web-дизайнера). Запомните простое правило: закрывающие теги должны повторяться в порядке, обратном порядку соответствующих им открывающих тегов.

Сохраните полученный файл под именем 1.2.htm и откройте его в Web-обозревателе. То, что вы увидите, показано на рис. 1.2.

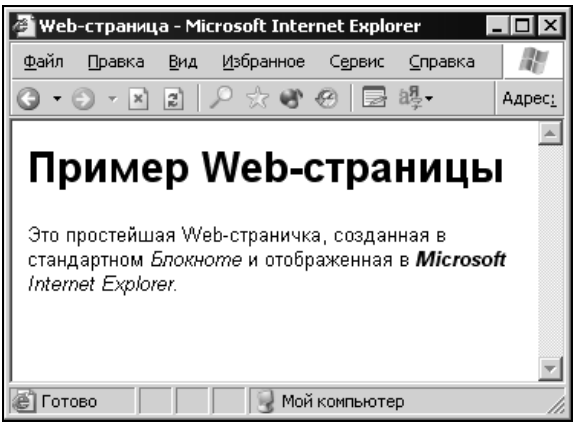


Рис. 1.2. Измененная Web-страничка

Как видите, ничего особо сложного в языке HTML нет. Единственная сложность — это запомнить все его теги, но это вопрос времени и опыта.

Для того чтобы различные программы Web-обозревателей правильно отображали одну и ту же Web-страницу, язык HTML должен быть стандартизирован. Его стандартизацией (а также множеством других стандартов Интернета) занимается особая организация, называемая *World Wide Web Consortium* или, сокращенно, WWWC. Это название можно перевести как "Комитет Всемирной паутины"... нет, лучше не переводить!... ужасное название...

WWWC издает весьма увесистые труды, описывающие различные версии стандарта HTML. Последняя версия этого языка — 4.01 — вышла в конце 90-х годов прошлого века. Все современные версии Web-обозревателей поддерживают эту версию HTML.

Наряду с тегами HTML, стандартизированными WWWC (*стандартные* теги), Web-обозреватели поддерживают множество *нестандартных* тегов. Эти теги были введены разработчиками той или иной программы Web-обозревателя, чтобы получить преимущество перед конкурентами. Они надеялись, что эти фирменные расширения впоследствии войдут в стандарт HTML, но этим надеждам так и не суждено было сбыться. Однако теги остались, и они все так же поддерживаются. Мы опишем эти теги наряду со стандартными, но предупреждаем вас, что они поддерживаются не всеми программами.

Версия 4.01 станет последней версией языка HTML. В дальнейшем он будет постепенно заменен своим потомком — языком *XHTML* (eXtensible HyperText Markup Language — расширяемый язык гипертекстовой разметки). Этот язык основан на популярном языке описания данных *XML* (eXtensible Markup Language — расширяемый язык разметки), он позволяет расширять

доступный набор тегов и считается более строгим и более подходящим для машинной обработки. К тому же, из XHTML выброшено множество устаревших тегов, все еще поддерживаемых HTML в целях совместимости.

Скоро ли наступит эра XHTML? Вряд ли. Старый добрый HTML поцарствует на земле еще не один год. Поэтому не стоит торопиться заучивать новые теги и переписывать уже написанные Web-страницы. Займитесь более полезными делами. Например, наведите порядок в ящиках своего стола или нанесите визиты своим одноклассникам.

Гиперссылки

Не только и не столько Web-страницы прославили Интернет. Свою роль внесло еще одно замечательное изобретение, буквально связавшее разрозненные документы в настоящую паутину. Она так и называется — Всемирная паутина, а по-английски — *World Wide Web* или WWW. (Вы уже встречали это название в наименовании World Wide Web Consortium — организации, ответственной за стандартизацию Интернета.) Благодаря этому изобретению вы можете с легкостью перемещаться по страницам, ведь любой уголок Всемирной сети находится от вас на расстоянии щелчка мыши.

Это *гиперссылки* — особые связи, ведущие от одной Web-страницы к другой. Именно по ним вы щелкаете мышью, если хотите перейти на другую страницу.

Гиперссылки создаются с помощью особого парного тега `<A>` и имеют следующий вид:

```
<A HREF="http://www.somesite.ru/pages/page125.html">Страница N125</A>
```

Как видите, тег `<A>` содержит атрибут `HREF`, который задает *интернет-адрес* страницы, на которую будет выполнен переход при щелчке по гиперссылке. Этот атрибут обязательно должен присутствовать в любом теге `<A>`, задающем гиперссылку — это *обязательный* атрибут. (Как выяснится далее, тег `<A>` может задавать не только гиперссылки.)

Интернет-адрес нужной Web-страницы вам нужно выяснять самим. Сделать это не так уж и сложно. Обычно он состоит из интернет-адреса Web-сайта, на котором находится нужная страница, и полного пути доступа к файлу этой страницы. В нашем случае, `http://www.somesite.com` — это адрес сайта, а `/pages/page125.htm` — путь доступа к файлу страницы.

Если нужно получить доступ к странице, находящейся на том же сайте, что и текущая, вы можете опустить адрес сайта. В этом случае код гиперссылки будет таким:

```
<A HREF="/pages/page125.html">Страница N125</A>
```

Если же нужная страница находится еще и в той же папке, что текущая, можно вообще оставить только имя файла:

```
<A HREF="page125.html">Страница N125</A>
```

А теперь давайте создадим две простейшие Web-странички и свяжем их гиперссылкой. В качестве первой страницы мы возьмем уже написанную нами ранее и сохраненную в файле 1.2.htm (см. рис. 1.2). Только добавим внизу следующий код:

```
<P><A HREF="1.4.htm">Сведения об авторе</A></P>
```

Сохраните новую страницу в файле 1.3.htm.

Код второй страницы приведен ниже. Я уверен, что вы в нем разберетесь.

```
<HTML>
  <HEAD>
    <TITLE>Сведения об авторе</TITLE>
  </HEAD>
  <BODY>
    <P>Эту страничку написал я. И я очень горд этим!</P>
  </BODY>
</HTML>
```

Сохраните ее в файле 1.4.htm.

Теперь откройте в Web-обозревателе файл 1.3.htm и щелкните по гиперссылке "Сведения об авторе". В окне Web-обозревателя появится страница со сведениями об авторе, сохраненная в файле 1.4.htm.

А теперь сделаем небольшой фокус. Измените код гиперссылки, помещенный в HTML-код первой страницы, таким образом (изменения выделены полужирным шрифтом):

```
<P><A HREF="1.4.htm" TARGET="_blank">Сведения об авторе</A></P>
```

Мы поместили в тег `<A>` атрибут `TARGET`, задающий цель гиперссылки. Цель гиперссылки задает, куда будет выведена Web-страница, на которую она указывает. Если этому атрибуту присвоено значение `_blank`, страница будет выведена в отдельное окно Web-обозревателя. Чтобы задать обычное поведение гиперссылки (новая страница выводится в то же окно), присвойте атрибуту `TARGET` значение `_self` или вообще уберите его из кода гиперссылки.

В отличие от атрибута `HREF` тега `<A>`, атрибут `TARGET` не является обязательным. Он так и называется — *необязательный* атрибут.

Сохраните измененную Web-страницу и повторно откройте ее в Web-обозревателе, после чего щелкните по гиперссылке. На экране появится но-

вое окно Web-обозревателя, в котором вы увидите страницу со сведениями об авторе. Вы можете увидеть оба этих окна на рис. 1.3.

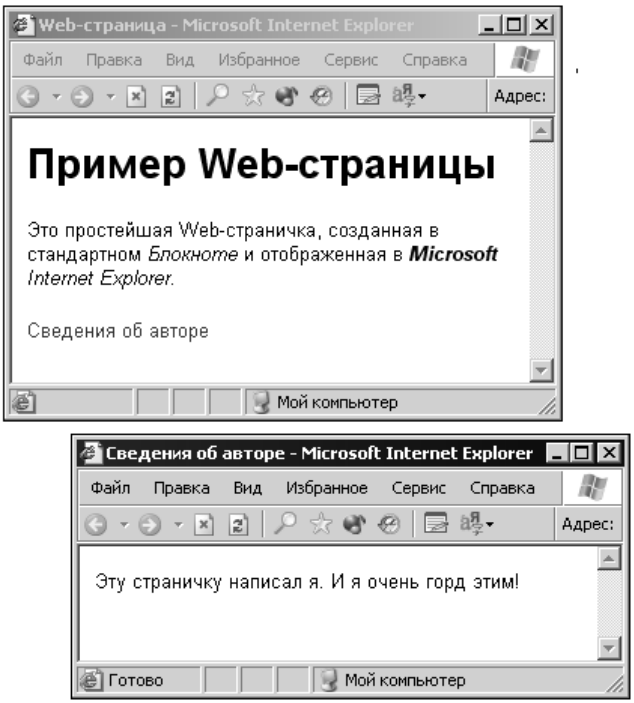


Рис. 1.3. Два окна Web-обозревателя, в которых загружены разные Web-страницы

Чтобы лучше узнать человека, нужно познакомиться с ним поближе. Так и с гиперссылками. В дальнейшем мы поговорим о них подробнее. А пока — закончим!

Клиенты и серверы Интернета

Итак, продолжим изучение основ Интернета и WWW.

Вы уже знаете, что Web-страницы сохраняются в текстовых файлах с расширением htm или html. Графические изображения, аудио- и видеоролики и прочие внедренные нетекстовые элементы сохраняются в виде отдельных файлов; их расширение зависит от формата файла. Web-обозреватель, загрузив страницу, содержащую внедренные элементы, загружает также нужные графические, звуковые и видеофайлы и формирует окончательный вид страницы.

Но как же Web-страницы, созданные кем-то, доставляются на ваш рабочий стол? По Интернету, скажете вы. И будете правы. Но как именно?

Сейчас мы это опишем.

Начнем с того, что все программы для Интернета делятся на две большие группы: клиенты и серверы. Программы-*серверы* предоставляют те или иные ресурсы для доступа программам-*клиентам*. Клиенты, если им понадобится какой-то файл или вообще какие-либо данные от сервера, формируют специальный *клиентский запрос* и посылают его на сервер. Сервер обрабатывает этот запрос и высылает *серверный ответ*, содержащий запрашиваемые данные или сообщение об ошибке, если нужные данные почему-то не доступны.

Такая компьютерная *архитектура*, иначе говоря, принцип построения вычислительной системы или сети, называется архитектурой *"клиент-сервер"* или *двухзвенной*. Этим она отличается от *однозвенной* или *одноранговой* архитектуры, когда все объединенные в сеть компьютеры равны между собой и могут как разделять, так и использовать сетевые ресурсы. Именно на основе двухзвенной архитектуры функционируют почти все сервисы Интернета. В том числе, и WWW.

В случае WWW клиентами выступают хорошо вам знакомые программы Web-обозревателей (или аналогичные программы, обрабатывающие Web-страницы, например, программы закачки Web-сайтов). Серверами же являются так называемые *Web-серверы*, обрабатывающие запросы Web-обозревателей и высылающие им нужные файлы.

С термином "Web-сервер" (как и термином "сервер") существует определенная путаница. Во-первых, так называют компьютер, подключенный к Сети по быстрому каналу и хранящий на своих жестких дисках файлы Web-страниц. Во-вторых, так называется программа, работающая на этом самом компьютере, принимающая от Web-обозревателей запросы и выдающая им соответствующие файлы. Но чаще и компьютер, и программу объединяют в единое целое и называют одним словом — Web-сервер. Так поступим и мы. (Хотя правильно называть Web-сервером именно программу.)

Когда вы набираете в поле адреса Web-обозревателя какой-либо интернет-адрес, обозреватель обращается к соответствующему Web-серверу. Сервер же извлекает со своих дисков нужные файлы (сама Web-страница, изображения, внедренные объекты, архивы, исполняемые файлы) и отправляет их Web-обозревателю. А уж он-то знает, как с ними поступить.

Люди, для того чтобы понимать друг друга, должны разговаривать на одном языке. Это же относится и к компьютерам. В их случае в качестве такого "языка общения" выступает *протокол* — набор правил обмена данными. Web-сервер и Web-обозреватель используют для обмена данными протокол *HTTP* (HyperText Transfer Protocol — протокол обмена гипертекстом). Этот *высокоуровневый* протокол работает "поверх" обычного *низкоуровневого* протокола *TCP/IP* (Transfer Control Protocol/Internet Protocol — протокол управления обменом/протокол Интернета).

Примечание

Функции Web-сервера может выполнять даже файловая система вашего компьютера. Учитывая эту возможность, мы сохранили нашу первую страничку на жестком диске, а потом щелкнули по нему мышью. Файловая система в ответ на запрос пользовательской оболочки загрузила этот файл и отобразила его в связанной с ним программе — Web-обозревателе.

Web-серверы устанавливаются на больших компьютерах, принадлежащих различным организациям. Также в Интернете доступно множество бесплатных Web-серверов, предоставляющих место для сайтов всем желающих. (Конечно, бесплатные серверы имеют много ограничений, но для большинства небольших некоммерческих сайтов это идеальный способ выйти в Сеть. Как говорится, дешево и сердито.) И наконец, существуют коммерческие организации, предоставляющие место на дисках Web-сервера всем желающим, но за деньги. Такие коммерческие серверы не имеют многих ограничений бесплатных серверов и поэтому пользуются успехом среди коммерческих организаций.

Итак, автор, чтобы сделать свои Web-страницы доступными для широкой публики, должен поместить их на жесткие диски компьютера, на котором работает программа Web-сервера. Этот процесс называется *публикацией* страниц (опытные Web-дизайнеры также часто говорят "выложить" вместо "опубликовать"). Процесс публикации страниц будет описан далее, в *главе 6*.

Установкой и настройкой Web-сервера, а также устранением проблем и наблюдением за его работой занимается особый человек — *администратор* Web-сервера. От действий (или бездействия) администратора зависит, какие возможности поддерживает Web-сервер и насколько надежно он работает. Как правило, вы не можете повлиять на действия администратора (тем более, если он администрирует бесплатный общедоступный сервер), если, конечно, сами не являетесь администратором своего собственного Web-сервера.

Одна из Web-страниц, хранящихся на диске серверного компьютера, при настройке Web-сервера задается в качестве *страницы по умолчанию* или *главной страницы*. Она будет загружена, если пользователь наберет в строке адреса Web-обозревателя только интернет-адрес сайта (например, "http://www.site.ru"), без указания имени файла страницы. Как правило, такая страница имеет имя default или index.htm и расширение htm или html, хотя, опять же, все это в руках администратора.

После получения файлов Web-страницы Web-обозреватель сохраняет их на жестком диске клиентского компьютера в особой папке, называемой *кэшем*. Зачем это нужно? Да хотя бы затем, чтобы вы смогли впоследствии просмотреть эту страницу, не подключаясь к Интернету. Все современные Web-обозреватели поддерживают так называемый *автономный режим* (по-английски — offline mode), когда они отображают только те страницы, что

находятся в кэше. Если же вы попытаетесь просмотреть страницу, которой нет в кэше, Web-обозреватель предложит вам подключиться к Интернету и загрузить ее.

Теперь познакомимся с популярнейшими в настоящее время Web-обозревателями.

Самым популярным среди них, настоящим королем виртуальных просторов, является Microsoft Internet Explorer. Он установлен на любом компьютере, работающем под управлением Windows, что, как говорят злые языки, и обусловило его популярность. Однако это очень мощная, быстрая, весьма нетребовательная к ресурсам и исключительно удобная программа, что бы там ни говорили его критики. Автор этой книги для просмотра Web-страниц пользуется именно Internet Explorer. В настоящее время доступна версия 6.0 и, по всей вероятности, разрабатывается новая версия, хотя фирма Microsoft, по своему обыкновению, хранит по этому поводу молчание.

Второе место по популярности занимает норвежская разработка Opera, выпускаемая одноименной фирмой. Эта достаточно мощная и очень быстрая программа, тем не менее, весьма охоча до системных ресурсов, особенно при отображении сложных Web-страниц. Кроме того, она является платной; в бесплатной версии она показывает рекламу. Последняя имеющаяся у автора версия носит номер 6.05 и, скорее всего, после выхода книги она устареет, т. к. новые версии Opera появляются очень часто.

Некогда властелин WWW Netscape Navigator сейчас в глубоком загоне — в настоящее время им пользуются от силы 2% интернетчиков. Хотя последняя версия Navigator — 7 — выглядит весьма неплохо, поддерживает все стандарты WWWC, корректно отображает большинство Web-страниц и не очень требовательна к системным ресурсам. Но все равно Navigator по многим параметрам проигрывает и Internet Explorer, и Opera.

Не очень давно был наконец-то выпущен в свет новый Web-обозреватель — Mozilla. Эта программа распространяется бесплатно, более того, ее исходные тексты открыты для изучения и модификации. Она построена на том же программном ядре, что и Navigator 7, точнее, наоборот, Navigator 7 построен на основе Mozilla. (Собственно, Mozilla и создавался для обкатки нового программного ядра Navigator, но в дальнейшем вырос в самостоятельный продукт.) Этот новичок весьма неплох, поддерживает все Web-стандарты, нетребователен к системным ресурсам, довольно быстр и имеет множество интересных и весьма полезных возможностей, которыми пока не может похвастаться ни один из его конкурентов. Пока что он не очень популярен, но в дальнейшем, возможно, еще себя покажет. Последняя версия, доступная в момент написания книги, — 1.1.

Существует еще некоторое количество программ Web-обозревателей, но они малопопулярны. Так что в настоящее время WWW принадлежит четверке перечисленных нами программ. Есть также некоторое количество Web-

обозревателей, построенных на основе Internet Explorer и расширяющих его возможности. Но поскольку это не самостоятельные продукты, они рассматриваться не будут.

А сколько в мире существует популярных Web-серверов? О-о-о, вероятно, не меньше, если не больше, чем популярных Web-обозревателей. Самыми популярными сейчас являются распространяемый с открытыми исходниками Apache и разработанный фирмой Microsoft Internet Information Server. Также неплох используемый автором для своего сайта Web-сервер Sambar.

Кстати, насчет Web-серверов. Если вы работаете в среде Microsoft Windows 98/ME или Windows 2000/XP, то можете установить Web-сервер, поставляемый с этой системой, и попробовать себя в качестве администратора. Это Personal Web Server (поставляется с Windows 98/ME) или Internet Information Server (Windows 2000/XP). Он нетребователен к ресурсам системы, довольно быстр в работе и удобен в настройке. Документация к нему поставляется также в составе операционной системы, к тому же, существует довольно много книг, посвященных этому Web-серверу. Советую вам поработать с ним, т. к. в дальнейшем он нам понадобится.

Вот и все о клиентах и серверах Интернета.

Зачем нужны HTML-редакторы

На этом завершим краткое введение в интернет-технологии. Вы узнали, что создавать Web-страницы очень просто, и для этого достаточно простейшего текстового редактора. Также вы познакомились с программами Web-серверов и узнали, как работает клиент-серверная архитектура. В связи с этим возникает вопрос: если язык HTML так прост, а в состав Windows входит Блокнот, прекрасно с ним справляющийся, то зачем нужны еще и Web-редакторы вроде Macromedia Dreamweaver?

А вот зачем...

"Уберите от меня подальше этот проклятый HTML!!! — кричит один читатель. — Я ничего в нем не смыслю! Он слишком сложен для меня, а у меня нет времени ему учиться. Дайте мне нормальный текстовый редактор, тот же Microsoft Word, и я буду работать в нем." И он будет прав.

HTML, конечно, прост. Но для кого-то он может оказаться невероятно сложным, ведь люди все разные. Кроме того, сложные Web-страницы писать "врукопашную" крайне неудобно. Уж поверьте! Поэтому вам понадобятся программы, автоматизирующие ваш труд — эти самые *Web-редакторы*, к славной плеяде которых относится и Macromedia Dreamweaver.

"Но мне нравится HTML! — возразит другой читатель, собаку съевший в Web-дизайне. — Мне проще вводить вручную HTML-теги, чем елозить по тексту мышкой и тыкать кнопки." И он тоже будет прав.

Ведь права старая пословица "Пуля — дура, а штык — молодец". Опытный мастер вручную или с помощью простейших инструментов может сделать такое, что не под силу механизированным и автоматизированным заводам. Не зря же все шедевры — в любой области искусства — делаются, как правило, вручную. А механизмам оставляют разве только изготовление заготовок...

Выходит, правы и первый, и второй читатели. Так как же помирить два враждующих лагеря: HTML-поклонников и HTML-ненавистников? Похоже, что никак...

Вот поэтому на свете и существуют две разновидности Web-редакторов: визуальные и невизуальные. Или, как еще говорят, WYSIWYG- и не-WYSIWYG-редакторы. (WYSIWYG или What You See Is What You Get — "что ты видишь, то ты и получишь".)

Невизуальные редакторы работают с "чистым" (или "сырым") HTML-кодом. Они предоставляют возможность быстрого ввода тегов, синтаксического подсвечивания, проверки правильности получившегося кода и ссылок. К ним относится, в частности, известная Arachnophilia.

Примечание

Невизуальные Web-редакторы часто называют HTML-редакторами.

Визуальные редакторы позволяют работать с самой Web-страницей "как она есть". Пользователь редактирует и форматирует текст, вставляет рисунки, таблицы, как в обычном текстовом редакторе, а уж сама программа формирует соответствующий HTML-код. К такого рода редакторам относится популярный Microsoft FrontPage и целый ряд программ попроще.

Как вы уже поняли, невизуальные редакторы прекрасно подходят для опытных Web-дизайнеров, съевших собаку на HTML и прекрасно представляющих, как должна выглядеть Web-страница в результате тех или иных изменений в ее коде. Визуальные редакторы — отличная стартовая площадка для начинающих Web-дизайнеров, плохо знакомых с HTML. Но и опытные дизайнеры часто пользуются визуальными редакторами, чтобы быстро "набросать" Web-страничку, а потом довести ее до ума "врукопашную".

Специально для таких пользователей, совмещающих прекрасное знание HTML с пристрастием к визуальным средствам редактирования, существуют так называемые *гибридные редакторы*. Они имеют мощные средства визуального редактирования, одновременно предоставляя доступ к получившемуся HTML-коду. Именно к таким редакторам и относится Macromedia Dreamweaver.

Примечание

Вообще-то сейчас, судя по всему, уже нет чисто визуальных редакторов — все они являются гибридными. Просто у одних доступ к HTML-коду осуществляется

проще (Dreamweaver), а у других — чуть сложнее (FrontPage). Поэтому практически всегда, когда говорят "визуальный Web-редактор", подразумевают как раз гибридные программы.

Вдобавок, как правило, все более-менее мощные редакторы Web-страниц — и визуальные, и невизуальные — имеют в своем составе развитые средства управления сайтом. Они знают, из каких страниц состоит ваш сайт, помогут опубликовать его на Web-сервере, предупредят о гиперссылках, указывающих "в никуда", и даже позволят вам составить список всех действий, которые вы хотите проделать с сайтом. Конечно же, Dreamweaver это тоже может.

Что дальше?

Вот и закончился наш краткий курс интернет-технологий. Конечно, многое здесь не описано. Все, что вам будет необходимо для создания и публикации Web-страниц, будет рассмотрено по ходу дела. А сейчас давайте, не углубляясь в дебри Интернета, перейдем к нашей главной цели — к работе с замечательным пакетом Web-редактора Macromedia Dreamweaver MX.

ГЛАВА 2

Основные принципы работы с Dreamweaver

В этой главе мы рассмотрим начальные сведения о работе с Macromedia Dreamweaver MX. Вы узнаете, зачем нужны его многочисленные окна и инструменты, какие из них действительно вам понадобятся, а какие — нет. Также мы узнаем о начальных настройках этой программы, в частности, что нужно сделать, чтобы Dreamweaver поддерживал русский язык (изначально он не знаком с русским, т. к. по происхождению — американец). Все это вам нужно знать для того, чтобы потом, во время создания нашей первой Web-странички, не задавать лишних вопросов.

Единственное пожелание: было бы очень неплохо, если бы вы знали хоть немного английский язык. В конце концов, персональный компьютер — детище американского образа жизни, и, несмотря на все "локализации" и "интернационализации", он всегда остается американцем, разговаривающим на своем языке. Хотя автор и попытается рассказать обо всех щекотливых моментах, с которыми вы можете столкнуться, и путях их решения или обхода, Dreamweaver вполне может огорошить вас каким-нибудь предупреждением, прочитать которое вы без знания английского не сможете. А раз вы не сможете его прочитать — то не сможете и отреагировать на него, что может быть весьма чревато...

Среда Dreamweaver MX

Итак, начнем рассматривать *рабочую среду* Dreamweaver MX, т. е. наборы окон и различных инструментов, предлагаемых им Web-дизайнеру. Но для начала запустим Dreamweaver.

Запустить программу в Windows проще простого! Нажмите хорошо знакомую вам кнопку **Start** (Пуск), выберите в меню пункт **Programs** (Программы), далее — пункт **Macromedia** и в появившемся подменю — пункт **Macromedia Dreamweaver MX**.

Выбор рабочей среды

Через некоторое время после запуска программы на экране появится небольшое диалоговое окно **Workspace Setup**, предлагающее вам выбрать вид рабочей среды Dreamweaver MX (рис. 2.1).

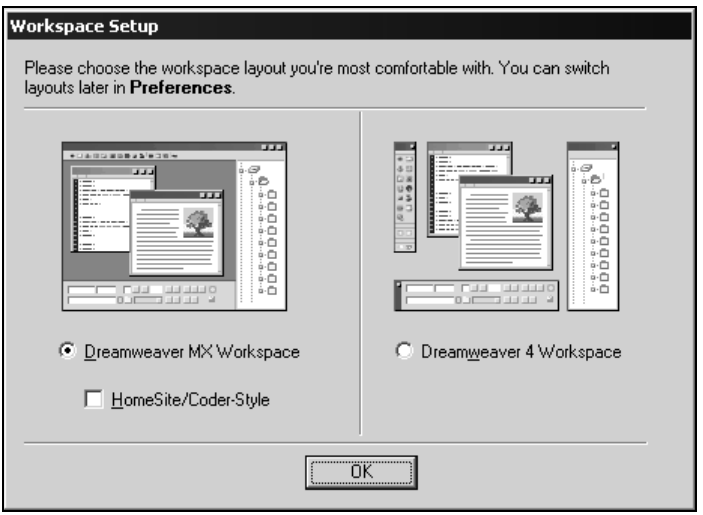


Рис. 2.1. Диалоговое окно **Workspace Setup**

Как видите, в этом окне находятся два переключателя и один флажок. Также вы можете видеть два небольших изображения, наглядно показывающие, какие виды рабочей среды вы можете выбрать. Давайте их рассмотрим.

Переключатель **Dreamweaver MX Workspace** выбирает "новый" вид рабочей среды Dreamweaver MX *("стиль MX")*. В этом случае все нужные вам инструменты будут находиться в одном большом окне, что, на взгляд автора, исключительно удобно. Рекомендуем вам включить именно этот переключатель (впрочем, он включен по умолчанию).

Переключатель **Dreamweaver 4 Workspace** выбирает "старый" вид рабочей среды, использовавшийся в старых версиях Dreamweaver — с 1-й по 4-ю *("старый стиль")*. В этом случае на экране перед вами будут находиться два окна: окно документа и окно сайта, между которыми вам придется часто переключаться. Это не очень удобно — в самом деле, когда все находится в одном окне, работать намного удобнее. Поэтому вам стоит включить этот переключатель только в том случае, если вы "пересаживаетесь" на Dreamweaver MX со старых его версий и еще не привыкли к его новой среде.

Флажок **HomeSite/Coder-Style** доступен только при выборе переключателя **Dreamweaver MX Workspace**. Он заставляет Dreamweaver сразу же после открытия очередной Web-страницы переключиться в режим редактирования

HTML-кода. Это может пригодиться только тем, кто привык набирать код HTML вручную. Если же вы не знакомы с HTML, лучше не включайте этот флажок.

Задав нужный вид рабочей среды Dreamweaver, нажмите кнопку **OK**. Через некоторое время вы увидите либо главное окно, либо окно документа, в зависимости от того, какой переключатель — **Dreamweaver MX Workspace** или **Dreamweaver 4 Workspace** — вы выбрали в окне **Workspace Setup**. Теперь можно начинать знакомство с этой замечательной программой.

Внимание!

Дальнейшее повествование будет вестись исходя из того, что вы выбрали "стиль MX" рабочей среды Dreamweaver.

Главное окно программы

Главное (или основное) окно программы Macromedia Dreamweaver MX показано на рис. 2.2. Рассмотрим его подробнее.

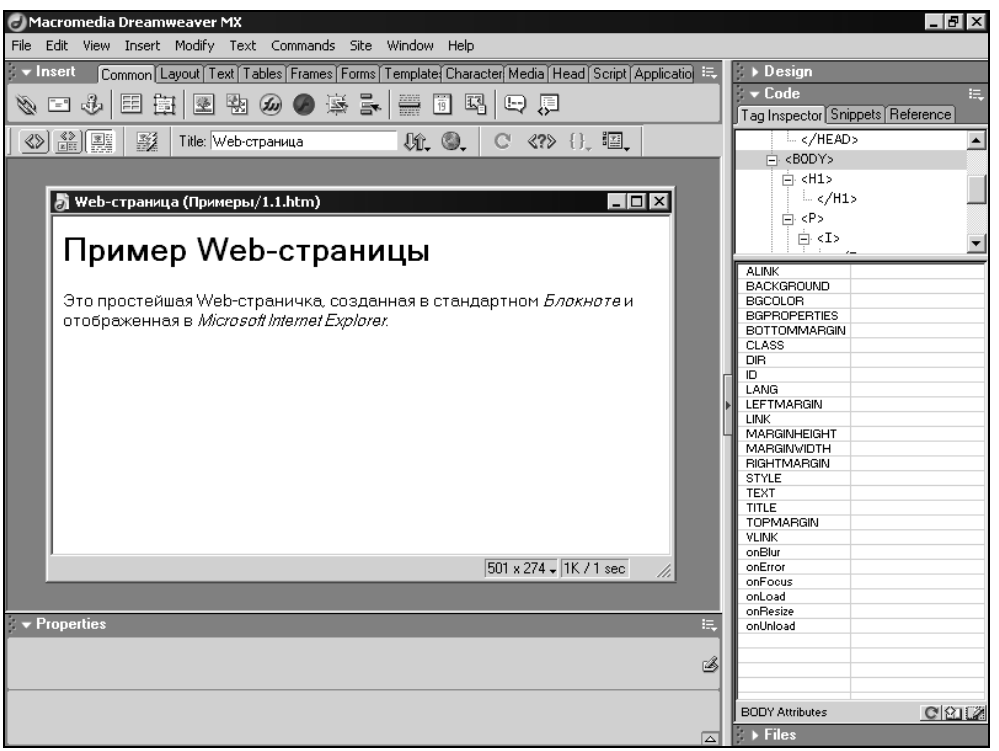


Рис. 2.2. Главное окно Dreamweaver

Скажем сразу, что Dreamweaver — программа с *многодокументным интерфейсом* или просто многодокументная программа. Это значит, что вы можете открыть в одном и том же окне программы сразу несколько документов. В этом случае окна, содержащие открытые документы, открываются внутри большого окна самой программы. К многодокументным приложениям также относятся Microsoft Word и Adobe Photoshop.

В отличие от них, программы с однодокументным интерфейсом (однодокументные программы) могут открыть только один документ; чтобы открыть второй, нужно запустить вторую копию программы. Примерами однодокументных приложений являются, в частности, текстовый редактор Microsoft WordPad и графический редактор Microsoft Paint, поставляемые в составе Windows.

Но вернемся к нашему Dreamweaver.

Главное окно служит "вместилищем" для превеликого множества других окон, содержащих как открытые Web-страницы, так и различные инструменты, предназначенные для работы с ними. Также в главном окне находится строка *главного меню*, с помощью которого вы сможете получить доступ ко всем возможностям Dreamweaver.

Окно документа Dreamweaver служит для отображения открытой Web-страницы (на рис. 2.2 вы можете видеть, что в Dreamweaver открыта одна из наших первых Web-страничек). Как и любая другая многодокументная программа, Dreamweaver может открыть в главном окне сколько угодно окон документов. Вы можете перемещать, свертывать и развертывать эти окна и изменять их размеры, в общем, проделывать с ним те же манипуляции, что и с любым другим окном Windows. Единственное исключение: вы не можете "вытащить" ни одно из этих окон за пределы главного окна программы (его еще называют *родительским окном*).

Надо сказать, что при первом запуске Dreamweaver MX выводит на экран еще одно небольшое окно — так называемое *окно-приглашение*. Это окно содержит текст рекламного характера, предлагающий пользователю прочитать некоторые справочные данные или запустить интерактивные презентации, объясняющие, как работать в Dreamweaver. Вы можете просмотреть их или сразу же закрыть это окно, щелкнув кнопку закрытия. При последующих запусках окно-приглашение появляться больше не будет.

Выше, ниже и правее окна документа находятся *панели*. Это небольшие окна, которые могут быть либо "приклеены" к одному краю главного окна, либо свободно "плавать" рядом с ним. Они предназначены для самых разных целей; подробнее мы рассмотрим их в следующих разделах книги. В верхней части каждой панели имеется ее *заголовок* — темно-серая полоса, на которой написано название панели.

Изначально все панели, имеющиеся на экране, "приклеены" к какому-либо краю главного (родительского) окна программы Разработчики из фирмы

Macromedia считают, что так будет удобно большинству пользователей Dreamweaver, и, похоже, они правы. Однако вы можете придерживаться иного мнения. Поэтому и здесь существует возможность отделить какую-либо панель от края родительского окна и превратить ее в отдельное окно (рис. 2.3) (так называемая *"плавающая"* панель). Для этого каждая панель имеет "ручку" для ее "переноски", находящуюся в левой части заголовка панели и выглядящую как пять темных точек. Ну и, конечно же, вы всегда можете присоединить панель обратно к краю главного окна, чтобы она не загораживала документ.

Рис. 2.3. Панель, отделенная от края родительского окна

"Отклеенная" от края главного окна панель может быть перемещена в любое место экрана, даже за пределы главного окна программы. Перетаскивать такую панель можно как за ее "ручку", так и за заголовок ее окна. Кроме того, вы можете изменять размеры окон панелей (многих, но не всех).

Если вы перетащите одну панель на другую, эти панели будут объединены в общую *группу* панелей, занимающую одно окно (рис. 2.4). Такие группы можно также "приклеивать" к краю главного окна программы. (Если вы посмотрите на рис. 2.2, то увидите несколько групп панелей, "приклеенных" к краю главного окна.) Рекомендуется объединять в одну группу панели, выполняющие сходные функции.

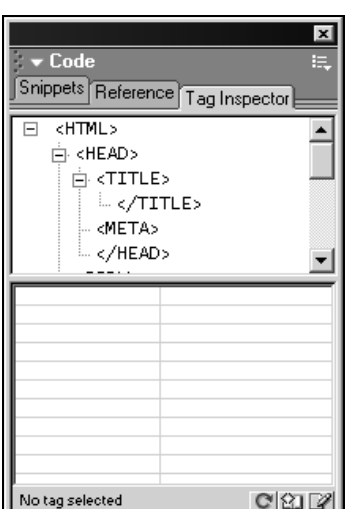


Рис. 2.4. Панели, объединенные в группу

Все панели, находящиеся в группе, представлены в виде вкладок. Чтобы переключиться на нужную панель в группе, щелкните мышью вкладку, на которой написано название этой панели. Если вы хотите вынести какую-либо панель из такой группы, щелкните правой кнопкой мыши по соответствующей вкладке, выберите пункт **Group <*название панели*> with** и в появившемся на экране подменю — пункт **New Panel Group**.

Внимание!

Когда вы выносите какую-либо панель за пределы группы, вы фактически создаете новую группу, состоящую из одной этой панели. В дальнейшем вы можете поместить в эту группу любые другие панели. Это значит, что в Dreamweaver все панели должны находиться в группах.

Чтобы временно уменьшить площадь, занимаемую "плавающей" панелью (фактически, группой панелей), вы можете ее "сжать", чтобы на экране остался только заголовок (рис. 2.5). Для этого щелкните по названию панели, находящемуся в ее заголовке. Чтобы "развернуть" панель до обычного состояния, снова щелкните по ее заголовку.

Рис. 2.5. "Сжатое" окно панели

Большинство групп панелей имеют так называемое *дополнительное меню*. Оно открывается при щелчке мышью по небольшой кнопке, расположенной в правом верхнем углу панели и имеющей изображение списка из трех позиций и небольшой стрелки, направленной вниз (рис. 2.6). В дополнительном меню находятся пункты, выполняющие редко используемые команды. В частности, вы можете найти там уже знакомое вам подменю **Group <*название панели*> with**.

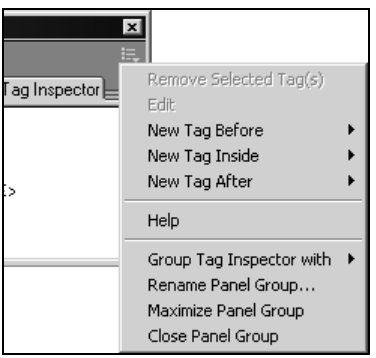


Рис. 2.6. Дополнительное меню группы панелей (открыто)

Теперь обратите внимание на правый край главного окна. Там находится настоящее сборище всяческих панелей. Это так называемый *док* — область, специально предназначенная для помещения на нее панелей. Док отделен от остального пространства главного окна толстой серой полосой, которую вы можете перетаскивать мышью, изменяя размеры дока. Вы также можете щелкать мышью довольно приметную кнопку (рис. 2.7), чтобы быстро скрыть док со всеми его панелями или открыть его снова. Поверьте — это настоящая находка программистов фирмы Macromedia; благодаря ей, пользоваться Dreamweaver стало намного удобнее.

Рис. 2.7. Кнопка скрытия-раскрытия дока

Панели всегда располагаются над окном документа, даже если в данный момент неактивны. Это сделано для того, чтобы вы могли всегда получить к ним доступ, вне зависимости от того, какое окно сейчас активно. Если же вы хотите убрать какую-либо из этих панелей, "вынесите" ее за пределы окна документа или вообще закройте, раскрыв ее дополнительное меню и выбрав пункт **Close Panel Group**. В дальнейшем вы сможете открыть нужную панель, выбрав соответствующий пункт меню **Window**.

Когда окно программы Dreamweaver перестает быть активным (например, когда пользователь переключается в другую программу), все "плавающие" панели временно скрываются. При активизации Dreamweaver они опять появляются на экране.

Как правило, все панели Dreamweaver имеют примерно одинаковые возможности (но, конечно, разное назначение). Но две из них стоят особняком. И мы сейчас их рассмотрим.

Первая из них — это *панель объектов*, показанная в "плавающем" виде на рис. 2.3 (она носит название **Insert**). Эта панель служит для быстрой вставки в документ различных элементов: рисунков, таблиц, специальных символов, расширений Web-обозревателя и пр. (Также она служит для переключения режимов работы окна документа, но об этом поговорим позже.) Панель объектов содержит множество вкладок, между которыми вы можете переключаться, — это способ сделать ее чуть более вместительной.

Если в данный момент времени панель объектов вам не нужна, можете закрыть ее, раскрыв ее дополнительное меню и выбрав пункт **Close Panel Group**. Чтобы открыть ее впоследствии, выберите в меню **Windows** пункт **Insert** или нажмите комбинацию клавиш <Ctrl>+<F2>.

Вторая панель, имеющая заголовок **Properties**, — это *редактор свойств* (рис. 2.8), один из важнейших и незаменимейших инструментов Dream-

weaver. Редактор свойств служит для задания параметров того или иного элемента Web-страницы (фактически — значений атрибутов соответствующего HTML-тега). Если панель объектов вы можете со спокойной душой закрыть, чтобы освободить место на экране, и использовать для вставки элементов страницы пункты меню главного окна, то редактор свойств вам будет нужен всегда. Поэтому держите его под рукой. Если же вы его все же случайно закрыли, выберите в меню **Windows** пункт **Properties** или нажмите комбинацию клавиш <Ctrl>+<F3>.

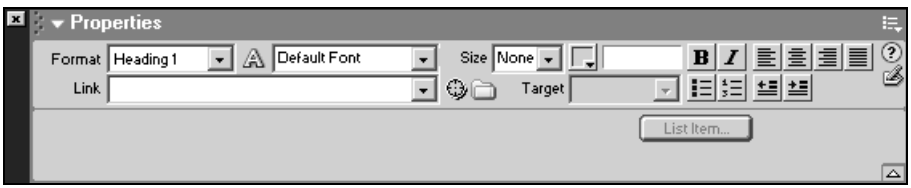


Рис. 2.8. Редактор свойств

Редактор свойств можно переключить в компактный или полный вид. Изначально он находится в компактном виде, не показывая некоторых, малоиспользуемых, с точки зрения разработчиков Dreamweaver, свойств. Автор рекомендует его сразу же развернуть, для чего щелкните мышью по кнопке, имеющей вид стрелки, направленной вниз, — она расположена в правом нижнем углу панели. После этого редактор свойств покажет вам все доступные свойства. (На рис. 2.2 редактор свойств находится как раз в развернутом виде.) Заметьте, что при этом кнопка развертывания изменит вид на стрелку вверх, и при щелчке на ней вы, наоборот, свернете редактор свойств.

Осталось рассказать совсем немного.

Взгляните на верхний край главного окна, туда, где находится панель объектов. Выше или ниже ее вы увидите узкую серую панельку, заполненную кнопками (рис. 2.9). Это один из инструментариев Dreamweaver. *Инструментарий* — особая панель, лишенная многих возможностей обычных панелей и предназначенная только для того, чтобы предоставить быстрый доступ к некоторым часто используемым операциям. Таких инструментариев в Dreamweaver два:

- *стандартный*, предоставляющий доступ к файловым операциям (создание, открытие и сохранение Web-страницы, операции с буфером обмена и откат), изначально скрыт;
- *документа*, позволяющий выполнять некоторые манипуляции с открытой Web-страницей и самой программой, виден на рис. 2.9.

Вы можете перетаскивать инструментарий, "ухватив" его за "ручку", имеющую вид расположенной у левого его конца вертикальной линии. Инстру-

ментарий, как и обычная панель, может быть "приклеен" к верхнему или нижнему краю главного окна программы или "плавать" отдельно от него.

Рис. 2.9. Инструментарий документа

Управление окнами и панелями Dreamweaver

Как видите, Dreamweaver может вывести на экран сразу множество разнообразнейших окон. Как разобраться во всем этом многообразии?

Прежде всего, нужно знать пункты меню, с помощью которых осуществляется управление этими окнами. Все эти пункты находятся в подменю **Window**. Рассмотрим их подробнее.

Если вы открыли несколько Web-страниц, разобраться в них может быть очень трудно. Окна перекрывают друг друга, и добраться до нужного окна удается далеко не сразу. Откройте подменю **Window** и посмотрите его нижнюю часть. Там будут находиться пункты, имеющие имена, схожие с именами файлов открытых страниц. Для того чтобы переключиться в окно, где открыт нужный файл, просто выберите соответствующий пункт. Dreamweaver тотчас выведет это окно на первый план, т. е. активизирует его.

Если же вы раскрыли одно из окон документов на весь экран (точнее, на все главное окно), то переключаться между окнами станет еще проще. В этом случае все открытые окна будут перечислены в нижней части окна документа в виде вкладок (рис. 2.10) — вам останется только выбрать необходимое.

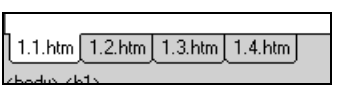


Рис. 2.10. Вкладки, обозначающие все открытые окна документа в раскрытом состоянии

Если вам нужно держать на виду сразу два или больше окон, воспользуйтесь пунктами **Cascade**, **Tile Horizontally** или **Tile Vertically** меню **Window**. Первый из них "выкладывает" все открытые окна документов в виде "стопки" в окне программы так, что вы можете видеть их заголовки и часть содержимого. Второй и третий пункты "выкладывают" в окне программы "мозаику" из окон документов так, чтобы они не перекрывались. Причем второй пункт выкладывает "мозаику" по горизонтали, а третий — по вертикали.

Пункты **Insert** и **Properties** служат для вывода на экран или скрытия соответственно панели объектов и редактора свойств. Если слева от имени одного из этих пунктов стоит галочка, это значит, что соответствующая панель выведена на экран (или, как еще говорят, что соответствующий пункт

меню "включен"). Чтобы убрать панель, снова выберите нужный пункт — и панель исчезнет вместе с галочкой. Такие пункты меню, меняющие свое состояние на противоположное при выборе, называют *выключателями*. Вместо выбора пунктов **Insert** и **Properties** вы можете нажать "горячие" комбинации клавиш <Ctrl>+<F2> и <Ctrl>+<F3> соответственно.

Бóльшую часть меню **Window** занимает набор аналогичных пунктов-выключателей, служащих для вывода на экран или скрытия различных панелей. Далее в книге будут описаны эти пункты вместе с рассмотрением той или иной панели.

Если вам нужно скрыть на время все панели, чтобы без помех просмотреть открытую Web-страницу, выберите пункт **Hide Panels** в меню **Window** или одноименный пункт в меню **View**. Этот пункт работает как выключатель, т. е. при первом выборе он скрывает все панели, а при втором — снова выводит их на экран. Вы также можете нажать клавишу <F4> — это проще и быстрее, чем лезть в меню.

Как вы уже знаете, каждая группа панелей имеет дополнительное меню. Кроме того, каждая отдельная панель имеет контекстное меню, появляющееся при щелчке правой кнопкой мыши по нужной вкладке и содержащей те же пункты. Пользуясь пунктами этого меню, вы можете выполнить над данной группой или панелью различные манипуляции:

- закрыть панель, выбрав пункт **Close Panel Group**;
- увеличить размеры панели так, чтобы она заняла весь экран компьютера по вертикали, выбрав пункт **Maximize Panel Group**;
- поместить выбранную группу или отдельную панель в другую группу, выбрав пункт **Group** ***<название группы или панели>*** **with** и далее в появившемся на экране подменю — пункт, соответствующий имени нужной группы панелей;
- переименовать группу, выбрав пункт **Rename Panel Group**. После этого вам остается ввести новое имя в поле ввода **Panel Group Name** диалогового окна **Rename Panel Group** (рис. 2.11) и нажать кнопку **OK** для его сохранения или **Cancel** — для отмены;
- поместить выбранную панель в новую группу, выбрав пункт **Group** ***<название группы или панели>*** **with** и далее в появившемся на экране подменю — пункт **New Panel Group**;
- получить справку по этой панели, выбрав пункт **Help**.

Чтобы вывести на экран нужный инструментарий, выберите соответствующий пункт-выключатель подменю **Toolbars** меню **View**. Всего таких пунктов два — по числу инструментариев:

- пункт **Standard** выводит главный инструментарий;
- пункт **Document** выводит инструментарий документа.

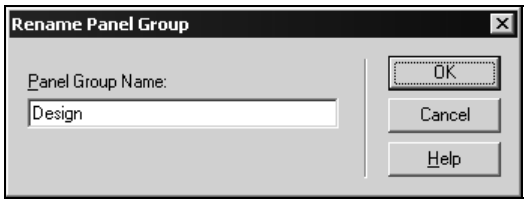


Рис. 2.11. Диалоговое окно **Rename Panel Group**

Вы также можете воспользоваться контекстным меню инструментария. Щелкните правой кнопкой мыши по любому инструментарию — и увидите небольшое меню, содержащее те же самые пункты.

Работа с Web-страницами

А теперь рассмотрим основные принципы работы с Web-страницами в Dreamweaver. И заодно поговорим об окне документа.

Работа в окне документа

Итак, как же в Dreamweaver создаются Web-страницы?

Очень просто. Точно так же, как создаются обычные текстовые документы в вашем любимом текстовом редакторе, например, Microsoft Word. Вы просто набираете нужный текст, форматируете его и, наконец, сохраняете готовый документ в виде файла. Только в случае Dreamweaver этот документ будет иметь расширение htm или html и может быть сразу же помещен на Web-сервер.

Но прежде чем писать текст, нужно создать сам документ. Для этого вам необходимо выбрать пункт **New** в меню **File** или нажать комбинацию клавиш <Ctrl>+<N>. Более детально процесс создания новой Web-страницы описан в *главе 3*.

Также вы можете открыть для правки уже существующую Web-страницу. Для этого выберите пункт **Open** в меню **File** или нажмите комбинацию клавиш <Ctrl>+<O>. После этого вам останется выбрать нужный файл в стандартном диалоговом окне открытия файла и нажать кнопку открытия.

Внимание!

Операции создания, открытия, сохранения, пересохранения, закрытия и печати документа называются *файловыми*.

Текст в Dreamweaver набирается точно так же, как и в любом текстовом редакторе (Блокнот и WordPad, поставляющиеся в составе Windows, Microsoft

Word и др.). В этом все Windows-приложения, работающие с текстом, похожи друг на друга. И это правильно — зачем обескураживать пользователя!

Мы не будем учить вас азам работы с текстом. Если вы пользовались текстовым редактором (а вы наверняка им пользовались), то без труда разберетесь с Dreamweaver. Благо разработчики сделали все, чтобы Dreamweaver ничем не отличался от других Windows-приложений. Мы просто кратко напомним вам некоторые приемы работы с текстом, которые вам скорее всего пригодятся.

Разумеется, текст набирается с помощью клавиатуры (а вы как думали?). При этом Dreamweaver самостоятельно разобьет текст на строки. Если вам нужно создать новый абзац, нажмите клавишу <Enter>. *Текстовый курсор*, т. е. мигающая вертикальная черточка, показывающая место, где будет появляться набираемый вами текст, может перемещаться во всех направлениях с помощью клавиш-стрелок. Также вы можете "листать" текст, нажимая клавиши <PgUp> и <PgUp>, мгновенно перемещаться к началу и концу строки клавишами <Home> и <End>. Чтобы быстро переместиться в начало или конец документа, нажмите, соответственно, комбинацию клавиш <Ctrl>+<Home> или <Ctrl>+<End>. Вы также можете устанавливать текстовый курсор в произвольное место, просто щелкнув там мышью.

Если вы ошиблись, то всегда можете удалить неправильно введенный текст, воспользовавшись клавишами <Backspace> и <Del>. Первая при нажатии удаляет символ, находящийся слева от текстового курсора, и идеально подходит для удаления только что введенного неправильного символа. Вторая удаляет символ, находящийся справа от текстового курсора.

Кроме того, в окне документа Dreamweaver доступны такие операции, как перемещение ("вырезание"), копирование текста в буфер обмена Windows и последующая его вставка в место, где находится текстовый курсор. Это может быть очень полезно, если вам нужно переместить фрагмент текста с места на место или поместить похожие фрагменты текста в разные места документа.

Чтобы выделить текст, просто щелкните в начале нужного фрагмента мышью и, не отпуская левой кнопки, протащите ее до конца, после чего отпустите кнопку. Вы также можете поставить текстовый курсор в начало необходимого фрагмента и воспользоваться комбинацией клавиш <Shift>+ +<Клавиша-стрелка в нужную сторону> для его выделения. Чтобы выделить остаток строки до конца, нажмите клавиши<Shift>+<End>, до начала — <Shift>+<Home>, остаток документа до конца — <Ctrl>+<Shift>+<End>, остаток документа до начала — <Ctrl>+<Shift>+<Home>. Вы также можете выделить строку, щелкнув мышью слева от нее, а если вы протащите мышь, не отпуская ее левую кнопку, то будут выделены сразу несколько строк. Чтобы выделить сразу весь текст, нажмите клавиши <Ctrl>+<A>.

Для выделения родительского тега вы можете выбрать пункт **Select Parent Tag** меню **Edit** или нажать комбинацию клавиш <Ctrl>+<Shift>+<<>. Для

выделения первого дочернего тега выберите пункт **Select Child** меню **Edit** или нажмите комбинацию клавиш <Ctrl>+<Shift>+<>>.

Чтобы вырезать выделенный фрагмент, нажмите комбинацию клавиш <Ctrl>+<X> или выберите пункт **Cut** меню **Edit**; чтобы скопировать его — комбинацию клавиш <Ctrl>+<C> или пункт **Copy** того же меню. Для вставки текста в нужное место расположите там текстовый курсор и нажмите комбинацию клавиш <Ctrl>+<V> или выберите пункт **Paste** меню **Edit**. Вы также можете стереть выделенный текст, нажав клавишу <Del> или выбрав пункт **Clear** все того же незаменимого меню **Edit**.

Если вам необходимо переместить выделенный фрагмент текста с места на место, вовсе не нужно беспокоить меню **Edit** — вы можете просто "взять" и перетащить его мышью. Чтобы скопировать этот фрагмент в другое место, перетащите его мышью при нажатой клавише <Ctrl>.

Щелкнув правой кнопкой мыши по тексту, вы получите доступ к контекстному меню, где тоже содержатся пункты **Cut**, **Copy** и **Paste**.

Если же вы сделали что-то не так, воспользуйтесь пунктом **Undo <*команда*>** меню **Edit** или нажмите комбинацию клавиш <Ctrl>+<Z>.

Форматирование текста выполняется точно так же, как и в обычных текстовых редакторах. Для этого вам достаточно выделить текст и воспользоваться редактором свойств, чтобы задать его параметры. Попробуйте, например, выделить в тексте нашей первой страницы какой-либо фрагмент и нажать кнопку **B**, которая хорошо заметна в редакторе свойств.

Однако Dreamweaver, как и любая специализированная программа, имеет свои особенности. И сейчас мы их рассмотрим.

В нижней части окна документа расположена *строка статуса*. Многие Windows-приложения используют строку статуса для отображения нужной пользователю информации. Но строка статуса окна документа Dreamweaver не только отображает информацию, но и выполняет несколько очень важных функций. Она разделена на три *секции*, каждую из которых мы рассмотрим отдельно.

Слева расположена *секция тегов*, занимающая большую часть строки статуса (рис. 2.12). С помощью этой секции можно выбрать фрагмент текста, помеченный тем или иным тегом HTML. Секция тегов имеет вид небольшой панели инструментов, чьи кнопки соответствуют тегам и расположены слева направо в порядке вложенности.

Рис. 2.12. Секция тегов

Предположим, что нам нужно выделить весь текст, помеченный тегом <I> (курсивное начертание). (В нашей первой странице — это название Web-обозревателя Microsoft Internet Explorer.) Для этого сделаем следующее. Поместим текстовый курсор куда-либо на текст "Microsoft Internet Explorer", после чего секция тегов примет вид, показанный на рис. 2.12. Далее нажмем кнопку **<i>** этой секции. Весь текст "Microsoft Internet Explorer" будет выделен, т. к. он помечен тегом <I>.

Если теперь щелкнуть кнопку **<p>**, будет выделен весь текст, помеченный тегом <P>, т. е. весь абзац. Осталось проверить, как работает кнопка **<body>**. Если ее щелкнуть, будет выделена вся страница (все содержимое тега <BODY>).

Секция тегов — мощнейший инструмент правки страницы. Благодаря ей вы можете получить доступ именно к тому тегу, который вам нужен. Попробуйте ей попользоваться — и скоро вы не сможете представить, как люди работают в других программах.

Правее находится вторая секция, отображающая размер окна документа в пикселах, — *секция размера окна*. Если вы щелкнете на ней мышью, на экране появится меню — список размеров окна (рис. 2.13). Каждый размер соответствует какому-либо разрешению экрана, которое может установить пользователь на своем компьютере. Вы можете выбрать любой такой размер и посмотреть, как будет выглядеть ваша Web-страница в окне. Это может понадобиться, если вы делаете Web-страницу под конкретное разрешение экрана — такое встречается довольно часто.

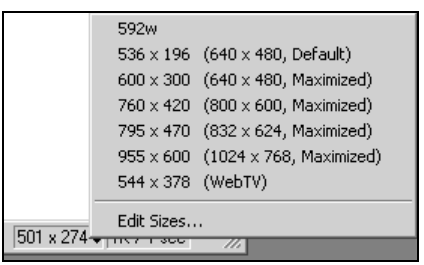


Рис. 2.13. Секция размера окна с открытым списком

Самая правая секция показывает примерный размер созданной вами Web-страницы и время, в течение которого она будет загружаться с Web-сервера. Это *секция размера страницы* (рис. 2.14). Она отображает два значения в виде дроби: в числителе ее находится размер страницы, округленный до килобайт, а в знаменателе — скорость загрузки, округленная до секунд. При этом скорость загрузки, по которой вычисляется это время, задается в настройках программы и по умолчанию равна 28,8 Кбит/с (не самый быстрый на сегодняшний день модем).

Рис. 2.14. Секция размера страницы

Закончив работу со страницей, вы должны ее сохранить. Для этого выберите пункт **Save** в меню **File** или нажмите комбинацию клавиш <Ctrl>+<S>. Если вы сохраняете страницу в первый раз, на экране появится стандартное диалоговое окно сохранения файла; введите имя файла страницы и нажмите кнопку сохранения. В противном случае Dreamweaver просто сохранит страницу.

Dreamweaver, как и большинство других программ-редакторов, позволяет вам также сохранить страницу в другом файле под другим именем. Это может быть полезно, если вы хотите создать множество примерно одинаковых страниц. Выберите пункт **Save As** в меню **File** или нажмите комбинацию клавиш <Ctrl>+<Shift>+<S>. На экране появится стандартное диалоговое окно сохранения файла; введите имя файла страницы и нажмите кнопку сохранения.

Выбрав пункт **Print Code** все того же меню **File** или нажав комбинацию клавиш <Ctrl>+<P>, вы сможете распечатать HTML-код страницы. На экране появится стандартное диалоговое окно печати; задайте необходимые параметры и нажмите кнопку запуска печати.

Все хорошее рано или поздно кончается. Готовую страницу нужно, в конце концов, закрыть, чтобы освободить системные ресурсы компьютера. Проще всего сделать это, щелкнув кнопку закрытия соответствующего окна документа (не главного окна программы!). Также можно выбрать пункт **Close** меню **File** или нажать комбинацию клавиш <Ctrl>+<W>. После этого страница будет закрыта, но сам Dreamweaver останется открытым.

Сам же Dreamweaver можно закрыть, щелкнув кнопку закрытия его главного окна. При этом он спросит вас, сохранять ли отредактированные, но еще не сохраненные страницы, которые вы к этому времени открыли. Если же вы предпочитаете пользоваться меню, то выберите пункт **Exit** меню **File** или нажмите комбинацию клавиш <Ctrl>+<Q> или <Alt>+<F4>.

Три режима отображения Web-страницы

Окно документа Dreamweaver может показывать редактируемую Web-страницу в трех *режимах отображения*. Сейчас мы их рассмотрим.

Как уже говорилось, Dreamweaver относится к гибридным Web-редакторам, позволяющим работать как с самой Web-страницей, так и непосредственно с ее HTML-кодом. Осуществляется это очень легко благодаря тому, что Dreamweaver может отображать одновременно и HTML-код, и саму страницу в одном окне.

Но как получить доступ к HTML-коду? Ведь по умолчанию Dreamweaver предлагает вам редактировать Web-страницу в режиме WYSIWYG.

Очень просто!

Взгляните на инструментарий документа (лучше всего держать его открытым). Слева видны три кнопки (рис. 2.15). Они переключают три доступных режима отображения (перечислены в порядке справа налево): страница, страница и HTML-код и только HTML-код.

Рис. 2.15. Кнопки переключения режимов отображения Web-страницы

На рис. 2.15 крайняя правая кнопка нажата. Она включает режим отображения, предлагаемый Dreamweaver по умолчанию, — *режим отображения страницы*. В этом режиме вы можете работать с Web-страницей в режиме WYSIWYG.

Теперь нажмем на среднюю кнопку, включающую *режим отображения страницы и кода*. На рис. 2.16 показано, что из этого получится.

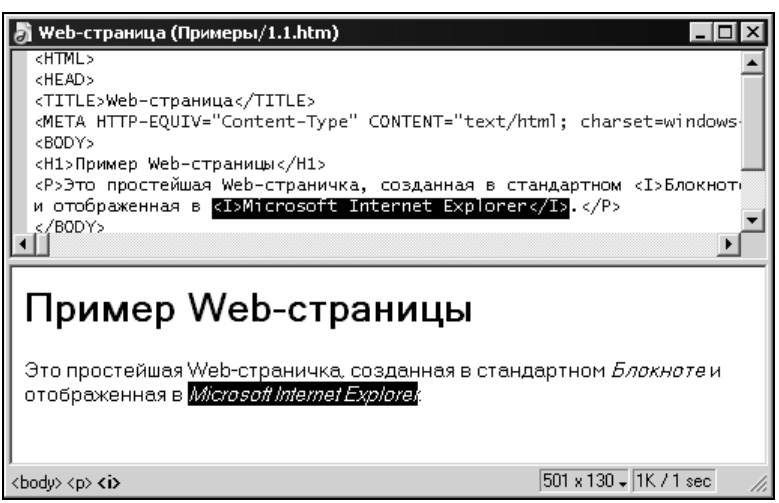


Рис. 2.16. Режим отображения страницы и кода

Как видите, окно документа разделилось на две части. В верхней части отображается HTML-код нашей страницы, а в нижней — сама страница в режиме WYSIWYG. Такой режим очень полезен, когда нужно "отшлифовать" код, при этом держа перед глазами саму Web-страницу. Вы можете перемещать мышью разделительную полосу, чтобы увеличить ту или иную часть окна.

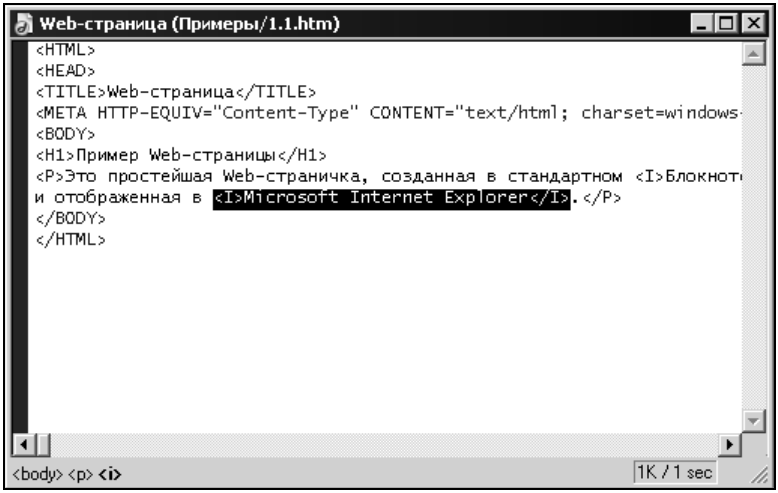


Рис. 2.17. Режим отображения кода

Ну, а левая кнопка включает *режим отображения кода* (рис. 2.17).

Для переключения режимов отображения вы также можете воспользоваться пунктами **Design**, **Code and Design** и **Code** меню **View**, соответственно. При выборе одного из этих пунктов слева от его названия появляется галочка, показывающая, что данный пункт и, следовательно, данный режим включены. Такие пункты называются *переключателями*.

Кроме того, вы можете быстро переключаться между режимом отображения страницы и режимом кода, выбрав пункт **Switch Views** меню **View** или нажав комбинацию клавиш <Ctrl>+<'>.

Работа с кодом HTML

А теперь давайте выясним, какие средства Dreamweaver предлагает тем, кто знаком с языком HTML.

Прежде всего, вы можете переключиться в режим отображения исходного кода HTML, нажав крайнюю левую кнопку из показанных на рис. 2.13. Также вы можете воспользоваться пунктом **Code** или **Switch View** меню **View**.

Кроме того, создатели Dreamweaver предусмотрели возможность просмотра исходного HTML-кода в панели, называемой **Code Inspector** (рис. 2.18). Чтобы вызвать ее на экран, выберите в меню **Window** пункт **Others** и в появившемся на экране подменю — пункт **Code Inspector**. Но проще всего нажать клавишу <F10>.

Во многих случаях этого хватает. Однако часто бывает нужно отредактировать фрагмент кода страницы, держа ее перед глазами. Для этого Dream-

weaver предоставляет так называемый *мини-редактор HTML*. Пользуясь мини-редактором, вы можете править атрибуты выбранного тега, вставлять код HTML в любое место текста — и все это без переключения в режим отображения кода.

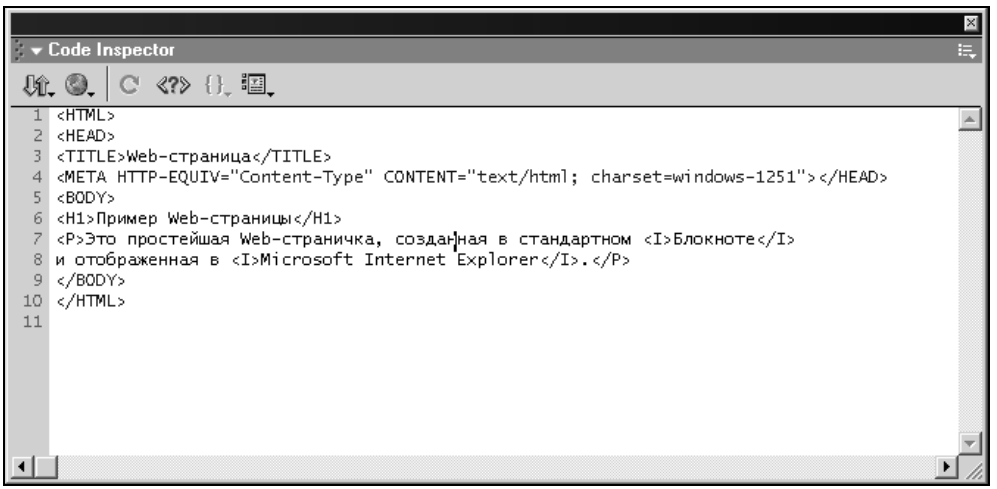


Рис. 2.18. Панель **Code Inspector**

Пусть, например, нужно немного поправить значения атрибутов какого-либо тега. Для этого поставьте текстовый курсор в его содержимое и выберите в контекстном меню пункт **Edit Tag Code <*тег*>**. На экране появится окно мини-редактора, показанное на рис. 2.19. Введите в него нужный код и нажмите клавишу <Enter>, чтобы Dreamweaver принял все изменения, или <Esc>, чтобы отказаться от них.

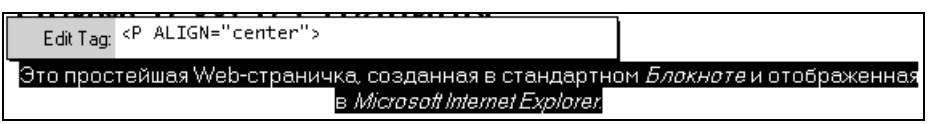


Рис. 2.19. Окно мини-редактора HTML

Как видите, окно мини-редактора позволяет вам править только теги, а не их содержимое. Исправить содержимое тегов можно и в окне документа.

Если вам нужно вставить в текст Web-страницы какой-либо тег с содержимым, поставьте в это место текстовый курсор и выберите в контекстном меню пункт **Insert Tag**. На экране также появится окно мини-редактора, на этот раз пустое (рис. 2.20).

Вы спросите: а что это за странный список, в котором перечислены непонятные слова? Это одна из замечательных возможностей, появившаяся в

Dreamweaver MX — *подсказка по коду*. Как только вы поставите где-либо в коде HTML значок <, как Dreamweaver отобразит список, в котором перечислены все теги HTML. Если вы нажмете на клавиатуре какую-либо буквенную клавишу, в списке появятся только теги, начинающиеся на эту букву. Чтобы выбрать нужный тег в списке, выделите его и нажмите клавишу <Enter> или просто щелкните по нему мышью.

Рис. 2.20. Окно мини-редактора при вставке нового тега

Введите в окно мини-редактора весь код, который вы хотите вставить (рис. 2.21), и нажмите клавишу <Enter>. Он будет помещен на то самое место, где стоит текстовый курсор.

Рис. 2.21. Окно мини-редактора с новым кодом HTML, который будет вставлен в страницу

И, наконец, вы можете "завернуть" любой выделенный фрагмент текста в тег HTML. Для этого выделите нужный текст и выберите в контекстном меню пункт **Wrap Tag**. Введите в появившемся окне нужный тег со всеми нужными атрибутами и нажмите клавишу <Enter>. Дело сделано!

Внимание!

Вообще-то существует еще пункт **Quick Tag Editor** меню **Modify** и кнопка в редакторе свойств, но лучше ими не пользоваться. По какому принципу они работают, знают, наверное, только программисты фирмы Macromedia. Поэтому пользуйтесь лучше контекстным меню окна документа — так надежнее.

Чтобы удалить тег, в который "завернут" какой-либо фрагмент текста, поставьте в него текстовый курсор и выберите в контекстном меню пункт **Remove Tag <*тег*>**. Dreamweaver удалит этот тег, но оставит его содержимое, которое "вольется" в содержимое родительского тега.

Если вы пока еще плохо знаете теги HTML, не беда. Dreamweaver MX идет вам на помощь, предоставляя несколько новых инструментов для правки

уже существующих и вставки новых тегов, которые наверняка понравятся неопытным пользователям.

Прежде всего, это *диалоговое окно правки тега*. Выделите целиком содержимое какого-либо тега, воспользовавшись секцией тегов строки статуса (рис. 2.12). После этого выберите пункт **Edit Tag <*тег*>**. На экране появится диалоговое окно, показанное на рис 2.22.

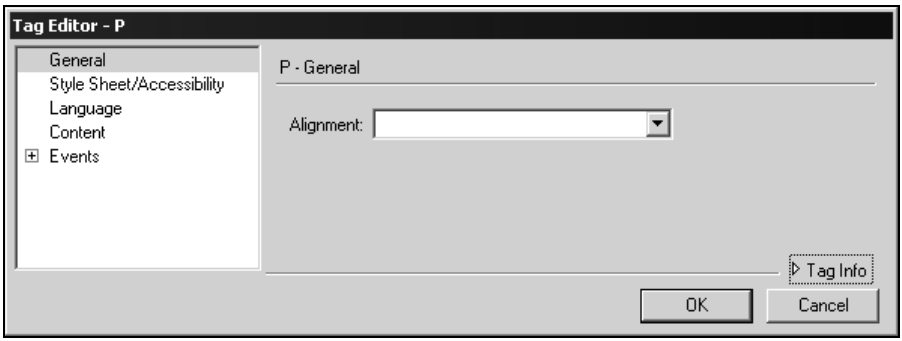


Рис. 2.22. Диалоговое окно правки тега

Пользуясь этим окном и своим знанием английского, вы можете в удобной форме задавать значения различных атрибутов выбранного тега. В левом списке выберите одну из категорий атрибутов, и в правой части окна появятся соответствующие элементы управления. Вы также можете щелкнуть *"потайную" кнопку* **Tag Info**, выглядящую как обычная надпись, после чего в окне правки тега появится краткая подсказка по выбранному тегу. Закончив правку, нажмите кнопку **OK**, чтобы сохранить все изменения, или кнопку **Cancel**, чтобы отказаться от них.

Если вы хотите исправить тег, держа страницу перед глазами, или просто не любите лишние диалоговые окна, воспользуйтесь панелью **Tag Inspector**. Чтобы вызвать ее на экран (если ее еще нет на экране), выберите пункт **Tag Inspector** в меню **Window** или просто нажмите клавишу <F9>. Сама эта панель показана на рис. 2.23.

В верхней части этой панели находится иерархический список тегов, присутствующий в открытой Web-странице. В нижней части панели располагается список атрибутов выбранного в иерархическом списке тега и их значений. Вы выбираете нужный тег и правите значения его атрибутов в нижнем списке, а в окне документов тут же отображаются все заданные вами изменения. Удобно, не правда ли?

Кстати, нажав расположенную в правом нижнем углу панели **Tag Inspector** кнопку **Edit <*тег*> Tag**, вы вызовете уже знакомое вам диалоговое окно правки тега. Эта кнопка показана на рис. 2.24.

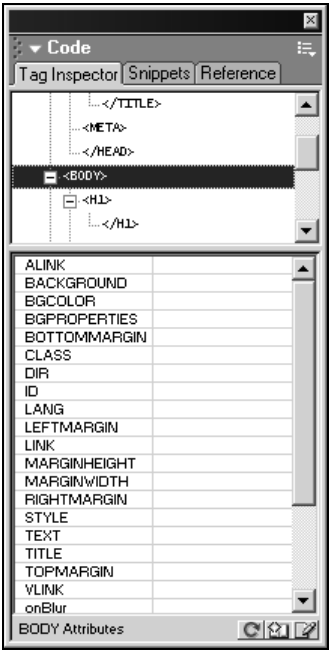


Рис. 2.23. Панель **Tag Inspector**

Рис. 2.24. Кнопка **Edit *<тег>* Tag**

Рис. 2.25. Кнопка **Tag Chooser**

С помощью другого диалогового окна вы можете поместить на страницу какой-либо тег или "завернуть" в него выделенный фрагмент текста. Для этого служит диалоговое окно **Tag Chooser**. Чтобы его вызвать, вам нужно выполнить одно из следующих действий:

- выбрать пункт **Tag** меню **Insert** или нажать комбинацию клавиш <Ctrl>+<E>;
- выбрать пункт **Insert Tag** контекстного меню окна документа (действует только тогда, когда окно документа находится в режиме отображения HTML-кода);
- нажать кнопку **Tag Chooser**, находящуюся на вкладке **Common** панели объектов (рис. 2.25).

Само окно **Tag Chooser** показано на рис. 2.26.

В иерархическом списке тегов, расположенном слева, вы можете выбрать категорию тегов. (В данный момент нам, вероятнее всего, понадобится категория, обозначенная ветвью **HTML Tags** списка.) Как только вы выберете

нужную категорию тегов, в правом списке появятся все теги, относящиеся к этой категории. Вам останется только выбрать тег и нажать кнопку **Insert**, чтобы вставить его в страницу.

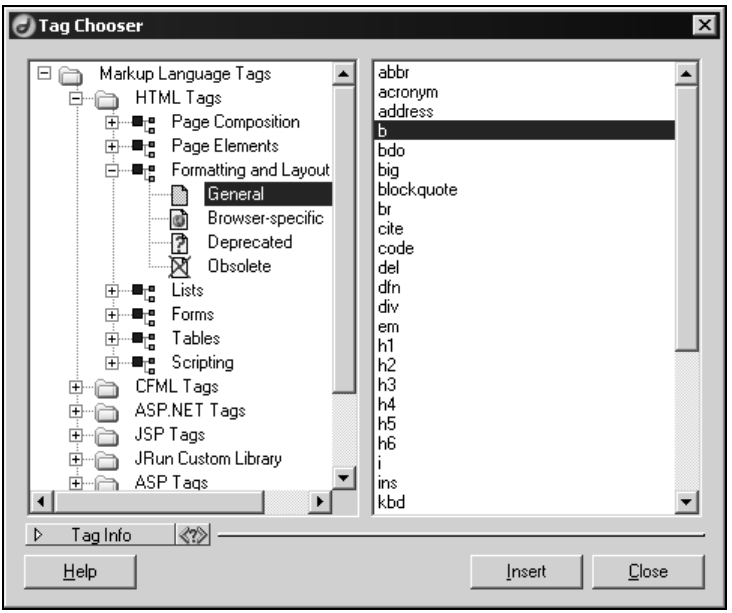


Рис. 2.26. Диалоговое окно **Tag Chooser**

Сразу после нажатия кнопки **Insert** на экране появится уже знакомое вам диалоговое окно правки тега, в котором вы сможете задать значения атрибутов выбранного тега. Сам тег будет вставлен только после нажатия кнопки **OK** окна правки тега. Заодно Dreamweaver сам переключит окно документа в режим отображения страницы и кода, чтобы вы могли видеть, где вставленный вами тег начинается и где он заканчивается.

Если вы щелкнете по кнопке **Tag Info** диалогового окна **Tag Chooser**, в этом окне появится краткое описание выбранного вами тега. Учтите, что при этом оба списка "съежатся", чтобы освободить место.

Если вы перед вызовом окна **Tag Chooser** выделите какой-либо фрагмент текста вместо того, чтобы просто поставить в какое-то его место текстовый курсор, то выделенный текст будет "завернут" во вновь вставленный тег. Как видите, Dreamweaver ведет себя достаточно "разумно", если так можно сказать об обычной программе.

Заметьте, что после вставки любого тега окно **Tag Chooser** останется на экране, так что вы сможете сразу же вставить новый тег. Еще обратите внимание, что это диалоговое окно является *немодальным*, т. е. оно не препят-

ствует доступу к любым другим окнам программы. (Абсолютное большинство других диалоговых окон Dreamweaver являются *модальными*, запрещающими пользователю доступ к остальным окнам программы, пока данное диалоговое окно не будет закрыто.) Чтобы закрыть окно **Tag Chooser**, нажмите кнопку **Close**.

Поиск и замена текста

Все текстовые редакторы, за исключением самых примитивных, предоставляют пользователю возможность найти нужный фрагмент текста и при необходимости заменить его другим. Разумеется, Dreamweaver также поддерживает эту возможность. И не просто поддерживает, а возносит ее на новую высоту.

Выберите пункт **Find and Replace** меню **Edit** или нажмите комбинацию клавиш <Ctrl>+<F>. На экране появится окно **Find and Replace**, показанное на рис. 2.27. Обратите внимание, что это также немодальное диалоговое окно, т. е. вы не потеряете доступ к окну документа.

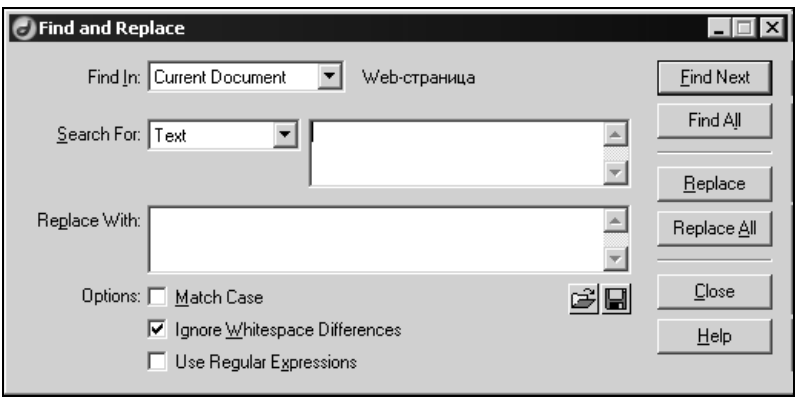


Рис. 2.27. Диалоговое окно **Find and Replace**

Всплывающее меню **Search For** позволяет задать, в каком тексте нужно произвести поиск: в тексте страницы (пункт **Text**), в "сыром" HTML-коде (**Source Code**), в тексте страницы, но более хитрым образом (**Text (Advanced)**), или в тексте заданного тега (**Specific Tag**). Если выбраны первые два пункта, то искомый текст *(подстрока)* вводится в текстовое поле **Search For**, а текст, на который нужно заменить искомый, — в текстовое поле **Replace With**.

Здесь все более-менее просто. Вы, наверное, уже пользовались функцией поиска и замены в других текстовых редакторах, так что уже знакомы со всем этим.

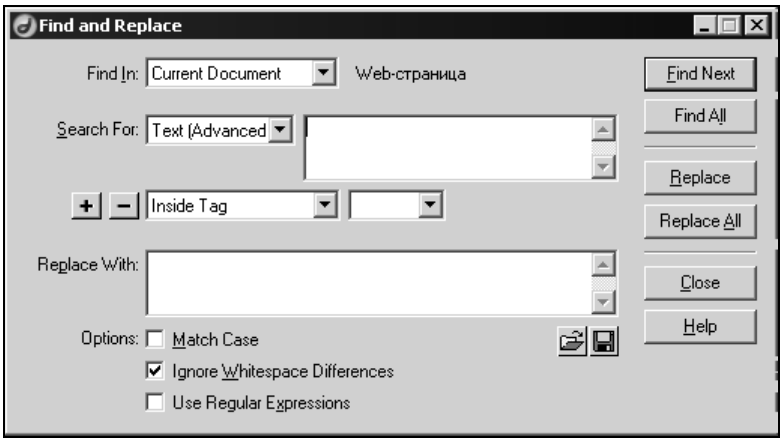


Рис. 2.28. Диалоговое окно **Find and Replace** (выбран режим "хитрого" поиска)

Если же вы выберете пункт **Text (Advanced)**, в окне поиска и замены кое-что добавится (рис. 2.28). Это будет новая группа элементов управления, состоящая из двух кнопок и двух раскрывающихся списков, позволяющих задать, внутри какого тега будет искаться тот или иной текст.

Раскрывающийся список, находящийся справа, задает тег, в котором будет искаться текст. А раскрывающийся список, находящийся слева, задает, где будет производиться поиск: внутри этого тега (пункт **Inside Tag**) или вне его (**Not Inside Tag**). Если вам нужно искать текст внутри (или вне) нескольких тегов, то, щелкая кнопку со значком плюса, вы можете добавить сколько угодно таких групп элементов управления. Соответственно, кнопка со значком минуса удаляет группу элементов, в которой она расположена.

Но самые мощные возможности Dreamweaver предоставляет для поиска и замены тегов, для чего достаточно выбрать пункт **Specific Tag** всплывающего меню **Search For**. В окне поиска и замены появится группа элементов управления для задания параметров поиска и еще одна группа — для задания параметров замены (рис. 2.29).

Искомый тег выбирается в раскрывающемся списке, расположенном правее всплывающего меню **Search For**. Если вы хотите искать любой тег, выберите пункт **[any tag]**.

В группе поиска самый левый раскрывающийся список задает, какие теги будут искаться:

- **With Attribute** — содержащие атрибут, заданный в следующем раскрывающемся списке (в порядке слева направо). В третьем раскрывающемся списке задается знак сравнения ("равно", "больше", "меньше" или "не равно"), а в четвертом — значение искомого атрибута или **[any value]**, если нужно найти атрибут, имеющий любое значение;

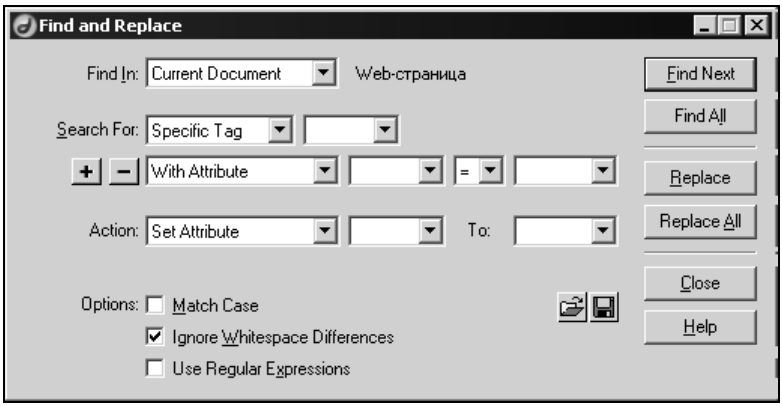


Рис. 2.29. Диалоговое окно **Find and Replace** (выбран режим поиска тега)

- **Without Attribute** — не содержащие атрибут, заданный в следующем раскрывающемся списке;
- **Containing** — содержащие внутри себя заданный текст или тег. Если необходимо найти текст, выберите пункт **Text** во втором всплывающем меню, а в поле ввода справа от него введите сам текст. Если нужно найти тег, выберите пункт **Specified Tag** во втором раскрывающемся списке, а в третьем — выберите сам тег;
- **No Containing** — не содержащие внутри себя заданный текст или тег;
- **Inside Tag** — находящиеся внутри тега, заданного во втором раскрывающемся списке;
- **No Inside Tag** — не находящиеся внутри тега, заданного во втором раскрывающемся списке.

Кнопки со знаками "плюс" и "минус" вам уже знакомы. Так что сразу перейдем к следующей группе элементов управления.

Внимание!

К сожалению, Dreamweaver не будет искать заданный вами тег, если вы не ввели никаких значений в раскрывающиеся списки группы элементов управления, задающих дополнительные условия поиска. Так что, если вы не собираетесь задавать дополнительных условий, например, атрибута искомого тега, рекомендуем вам удалить все пустые группы.

Раскрывающийся список **Action** задает действие, производимое над найденным тегом. А сделать с ними можно следующее:

- **Replace Tag & Contents** — заменить найденный тег на другой, введенный в области редактирования, расположенной правее;
- **Replace Contents Only** — заменить содержимое найденного тега на другое, введенное в области редактирования, расположенной правее;

- **Remove Tag & Contents** — удалить найденный тег вместе с содержимым;
- **Strip Tag** — удалить тег, но оставить содержимое;
- **Change Tag** — изменить тег на другой, выбранный в расположенном правее раскрывающемся списке, но оставить прежнее содержимое;
- **Set Attribute** — установить атрибут, выбранный во втором раскрывающемся списке (в порядке слева направо), со значением, заданным в комбинированном списке **To**;
- **Remove Attribute** — удалить атрибут, выбранный во втором раскрывающемся списке;
- **Add Before Start Tag** — добавить перед начальным тегом текст, введенный в текстовом поле;
- **Add After End Tag** — добавить после конечного тега текст, введенный в текстовом поле;
- **Add After Start Tag** — добавить после начального тега текст, введенный в текстовом поле;
- **Add Before End Tag** — добавить перед конечным тегом текст, введенный в текстовом поле.

Флажок **Match Case** предписывает Dreamweaver учитывать регистр символов при поиске. По умолчанию он отключен, т. е. регистр символов при поиске не учитывается.

Флажок **Ignore Whitespace Differences** предписывает Dreamweaver считать несколько расположенных подряд пробелов за один. По умолчанию он включен.

Флажок **Use Regular Expressions** включает обработку регулярных выражений. *Регулярные выражения* — это своего рода шаблоны, по которым Dreamweaver будет искать и заменять сложные последовательности символов. Мы особо опишем их использование — настолько это мощное средство.

Но как теперь запустить сам поиск (замену)?

Вы можете щелкнуть на кнопке **Find Next** — и первое вхождение найденной последовательности символов (подстроки) будет выделено в окне документа. Чтобы найти следующую подстроку, опять щелкните на кнопке **Find Next** или, если окно **Find and Replace** уже закрыто, нажмите клавишу <F3> (при активном окне документа) или выберите пункт **Find Next** в меню **Edit**. Аналогично работает кнопка **Replace**: при ее нажатии выделяется первая найденная подстрока, а при последующих нажатиях ранее выделенная подстрока заменяется заданной и выделяется следующая найденная подстрока. После того как все подстроки будут заменены, Dreamweaver выдаст окно-предупреждение с указанием, сколько замен было произведено.

Чтобы заменить сразу все подстроки, нажмите кнопку **Replace All**.

Вы также можете сразу найти все вхождения подстроки в тексте страницы, нажав кнопку **Find All**. Конечно, Dreamweaver сразу не сможет все их выделить — это не сможет сделать ни одно Windows-приложение. Поэтому он использует другой подход. На экране появляется панель **Search** (рис. 2.30), содержащая список, в котором представлены все найденные подстроки. При двойном щелчке мышью на любой его позиции в окне документа будет выделена соответствующая подстрока.

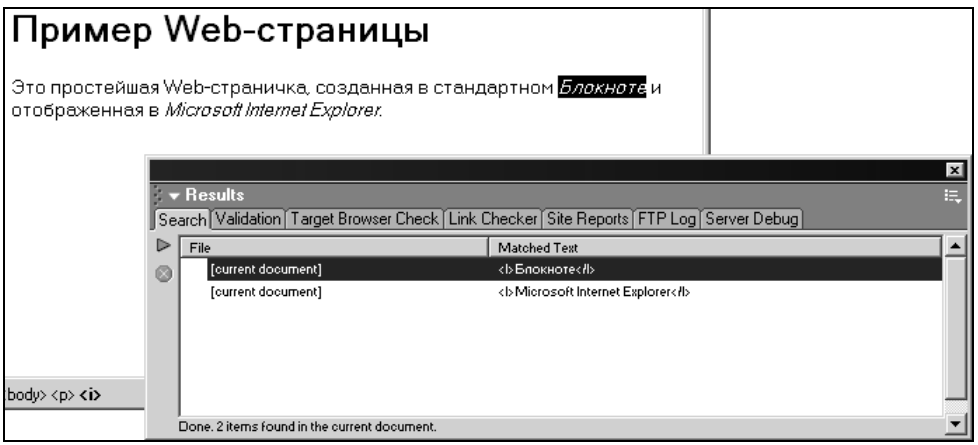


Рис. 2.30. Панель **Search**

А теперь давайте представим себе такую ситуацию. Вы задали какое-то сложное условие поиска (и, возможно, замены) и теперь хотите сохранить его для дальнейшего использования или на память потомкам. Как это сделать? Очень просто: Dreamweaver предоставляет вам и такую возможность. Просто щелкните на кнопке с изображением дискеты в окне **Find and Replace**, после чего на экране появится стандартное диалоговое окно сохранения файла Windows, где вы сможете задать имя вновь создаваемого файла или выбрать для перезаписи уже существующий. А чтобы загрузить сохраненное ранее в файле условие, щелкните на кнопке с изображением папки; на экране появится стандартное диалоговое окно открытия файла Windows, где вы сможете выбрать нужный файл условия.

Использование регулярных выражений

Как и было обещано, сейчас мы познакомимся с регулярными выражениями. Чем же они могут нам помочь?

Представим себе такую ситуацию. Вам нужно найти в исходном HTML-коде страницы интернет-адрес, причем любой. Как вы это сделаете? Правильно, введете в поле **Search For** окна поиска и замены (см. рис. 2.27) сим-

волы www. И найдете... кое-что. Но не все. Ведь интернет-адреса совсем не обязательно начинаются на "www". Что в этом случае искать? Точку? Но точек в тексте и так хватает, и далеко не всякая из них является частью интернет-адреса.

В этом случае нам на помощь придет регулярное выражение. Оно будет иметь следующий вид:

```
"http://.+\..{2,3}"
```

Ну и как оно вам? Больше похоже на бессмысленный набор символов, в котором с трудом угадывается что-то знакомое. Признаться, автор и сам сначала с трудом в них разбирался. Ему помогала вот такая табличка — см. табл. 2.1.

Таблица 2.1. *Регулярное выражение поиска интернет-адресов*

Символы	Описание
"http://	Первая кавычка и начало интернет-адреса
.	Точка обозначает любой символ
+	Плюс обозначает, что предыдущий символ должен повториться минимум один раз
\.	Обычная точка. Ее предваряет обратная косая черта, т. к. точка — служебный символ
.	Опять любой символ
{2,3}	Предыдущий символ должен повторяться от двух до трех раз
"	Закрывающая кавычка

Вы можете попробовать это регулярное выражение в действии. Только не забудьте выбрать в раскрывающемся списке **Search For** пункт **Source Code**.

Как видите, в регулярных выражениях используются специальные символы — *литералы*. С их-то помощью и задаются условия поиска тех или иных символов. Если вы хотите найти какой-либо символ, совпадающий с литералом, вы должны будете предварить его обратной косой чертой. Например, чтобы найти точку, вы должны использовать такую последовательность символов: "\."

С помощью регулярных выражений вы можете выполнять поиск самых разных слов и словосочетаний. Например, регулярное выражение совпадает со словами "multimedia" и "hypermedia", но не совпадает со словом "media". Литерал | задает поиск либо первой, либо второй подстроки (в нашем случае либо "multi", либо "hyper"), а скобки здесь использованы для того, чтобы

отделить друг от друга две части выражения. Если бы мы их не поставили, получилось бы выражение

```
"(multi|hyper)media"
"multi|hypermedia"
```

совпадающее со словами "multi" и "hypermedia".

А регулярное выражение

```
"/b.+@.+\.com"
```

ищет адреса электронной почты на серверах, чей интернет-адрес оканчивается на "com". Здесь вам все знакомо, за исключением литерала `/b`, обозначающего границу слова. То есть точка задает поиск любого символа, а `/b.` — только символа, с которого начинается слово (пробел или возврат каретки). А как насчет вот такого выражения:

```
"[\.!\?]$"
```

Оно ищет любой знак препинания, находящийся в конце текстового абзаца. В этом выражении используется много новых символов, так что мы рассмотрим его подробно. По аналогии с предыдущим регулярным выражением, расшифруем его с помощью табл. 2.2.

Таблица 2.2. *Регулярное выражение поиска последнего знака препинания абзаца*

Символы	Описание
`[\.!\?]`	Один из возможных символов: точка, восклицательный и вопросительный знаки. Заметьте, что вопросительный знак предварен обратной косой чертой, т. к. иначе Dreamweaver воспринял бы его как литерал
`$`	Этот литерал обозначает конец строки

Хорошо, при поиске подстрок регулярные выражения очень нам помогают. Но помогут ли они при замене?

Еще как!

Предположим, например, что нам необходимо найти все нужные адреса электронной почты и заменить у них окончание "com" на "ru". Для поиска в этом случае мы используем уже знакомое и работающее регулярное выражение:

```
"/b(.+@.+\.)com"
```

Постойте, но зачем мы взяли часть его в скобки? Ведь скобки используются для группировки, отделения части регулярного выражения от других частей. Здесь же ничего отделять не нужно — все и так работает.

А вот зачем.

Дело в том, что Dreamweaver, кроме всего прочего, помещает подстроку, совпадающую с заключенной в скобки частью регулярного выражения, в особую ячейку памяти, откуда ее можно потом будет извлечь. Таких ячеек памяти всего девять, и обращаться к их содержимому можно по номеру. В частности, извлечь содержимое первой ячейки в нашем случае можно так:

```
"$1"
```

Тогда заменить окончание почтового адреса можно с помощью такого регулярного выражения:

```
"$1ru"
```

Здесь все совсем просто. Мы берем первую подстроку и добавляем к ней окончание "ru". Наберите в окне документа несколько произвольных адресов электронной почты и проверьте наши выражения. Они работают.

Напоследок самое время привести полный список всех литералов регулярных выражений. Они перечислены в табл. 2.3.

Таблица 2.3. *Литералы регулярных выражений*

Литерал	Описание
`^`	Начало строки
`$`	Конец строки
`*`	Предыдущий символ должен встретиться ноль, один или больше раз
`+`	Предыдущий символ должен встретиться один или больше раз
`?`	Предыдущий символ должен встретиться ноль или один раз
`.`	Любой символ за исключением символа новой строки
`x|y`	Должен встретиться символ `x` или символ `y`
`{n}`	Предыдущий символ должен встретиться точно `n` раз
`{m,n}`	Предыдущий символ должен встретиться от `n` до `m` раз
`[abc]`	Должен встретиться один из символов, перечисленных в квадратных скобках. Можно задавать диапазоны символов, например `[a-d]` заменяет `[abcd]`
`[^abc]`	Должен встретиться любой символ, кроме перечисленных в квадратных скобках. Можно задавать диапазоны символов
`\b`	Граница слова (пробел или возврат каретки)
`\B`	Не граница слова
`\d`	Любая цифра. Эквивалентен `[0-9]`

Таблица 2.3 *(окончание)*

Литерал	Описание
`\B`	Любой символ, кроме цифры. Эквивалентен `[^0-9]`
`\f`	Прогон листа
`\n`	Перевод строки
`\r`	Возврат каретки
`\s`	Любой пробельный символ (пробел, табуляция, прогон страницы или перевод строки)
`\S`	Любой символ, кроме пробела
`\t`	Табуляция
`\w`	Любой алфавитно-цифровой символ или подчеркивание. Эквивалентен `[a-zA-Z0-9_]`
`\W`	Любой символ, кроме алфавитно-цифрового и подчеркивания. Эквивалентен `[^a-zA-Z0-9_]`

Просмотр Web-страницы

Хоть Dreamweaver в режиме просмотра страницы и представляет ее почти в таком виде, как она будет показана в Web-обозревателе, все же часто возникает необходимость увидеть ее в самом Web-обозревателе. Дело в слове "почти": все-таки Dreamweaver не может учесть многие тонкости конкретной программы просмотра. И такая возможность предусмотрена: не закрывая окна документа, вы можете вызвать любой из установленных на компьютере Web-обозревателей и оценить окончательный вид своего творения, так сказать, "в родной обстановке".

Microsoft Internet Explorer и Netscape Navigator, установленные на компьютере, Dreamweaver обнаруживает и заносит в свои настройки сам, сразу при инсталляции. Если же вы установили какую-то из этих программ уже после установки Dreamweaver или пользуетесь другой программой для просмотра Web-страниц (например, Opera или Mozilla), вам придется соответственно изменить настройки Dreamweaver, чтобы добавить ее в список установленных программ просмотра. Как это сделать, будет рассказано в конце главы.

Примечание

Профессиональные Web-дизайнеры обычно устанавливают на своих компьютерах несколько программ Web-обозревателей: Internet Explorer, Navigator, Opera и Mozilla разных версий. Каждую разрабатываемую Web-страницу они тестируют на всех этих программах на предмет совместимости друг с другом. Но, поскольку последние версии всех этих программ отображают Web-страницы бо-

лее-менее одинаково (за исключением, может быть, Opera), вы можете ограничиться Internet Explorer 6.0, который установлен по умолчанию на любой компьютер с операционной системой Windows. (Если, конечно, в ваши задачи не входит разработка страниц под экзотические программы Web-обозревателей.)

Давайте просмотрим в Internet Explorer нашу страничку, загруженную сейчас в окно документа. Для этого нажмем клавишу <F12>. На экране появится окно Web-обозревателя, знакомое нам по рис. 1.1.

Есть еще два способа вызова Web-обозревателя для просмотра редактируемой страницы. Первый из них — использование кнопки **Preview/Debug in Browser** инструментария документа. Эта кнопка показана на рис. 2.31.

Рис. 2.31. Кнопка **Preview/Debug in Browser**

Как видите, при нажатии этой кнопки появляется меню, предлагающее несколько пунктов. В данный момент нас интересует пункт **Previews in iexplore**. При выборе его, как вы уже поняли, открывается окно Internet Explorer, где будет загружена разрабатываемая вами страница.

Второй способ вызвать Web-обозреватель — это использовать подменю **Preview in Browser** меню **File**. В нем вы увидите пункт **iexplore**, аналогичный вышеописанному. Выбрав его, вы получите тот же самый результат.

Вызов справки

В процессе работы с Dreamweaver вам может понадобиться — и наверняка понадобится! — помощь. Как и все серьезные Windows-приложения, Dreamweaver снабжен мощной *справочной системой*. Для ее вызова вам нужно просто нажать клавишу <F1> или выбрать пункт **Using Dreamweaver** в меню **Help**. Окно справочной системы показано на рис. 2.32.

В левой части окна справки расположен древовидный список тем. Сами же статьи отображаются справа. Вы можете щелкнуть мышью по названию темы, чтобы вывести нужную статью, развернуть или свернуть "ветвь" "дерева" тем.

Также предусмотрен поиск статей по ключевым словам. Для этого выберите вкладку **Поиск**, введите в поле ввода нужное ключевое слово и нажмите кнопку **Разделы**. После этого в списке, занимающем нижнюю половину вкладки, появятся заголовки найденных статей. Вам остается только вы-

брать нужную и нажать кнопку **Показать**. Выбранная статья отобразится справа.

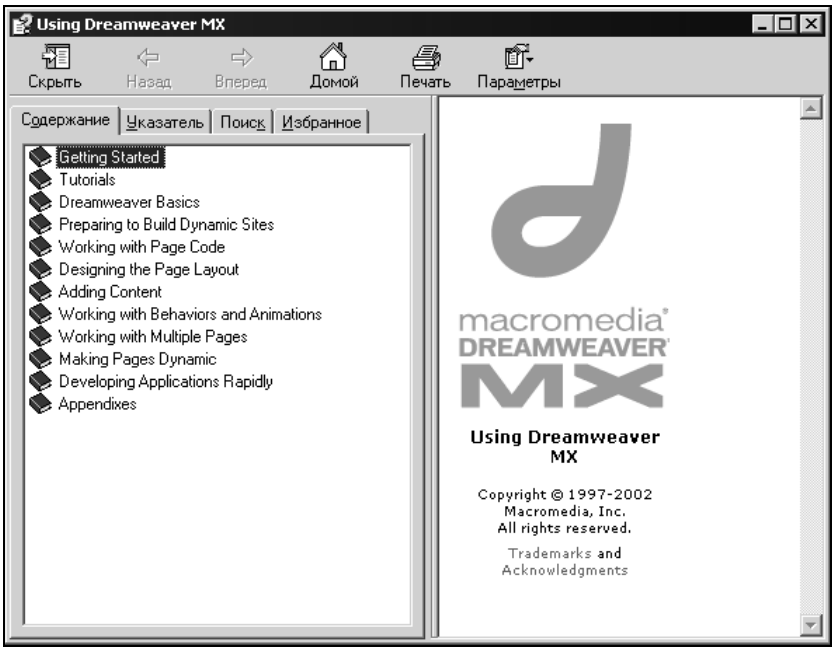


Рис. 2.32. Окно справочной системы Dreamweaver

В составе Dreamweaver также поставляется полное руководство по HTML от издательства "O'Relly". Его содержимое отображается в особой панели **Reference** (рис. 2.33). Поищите эту панель в доке — она обычно открыта по умолчанию. Если же вы успели ее закрыть, нажмите кнопку **Reference** в инструментарии документа (рис. 2.34). Вы также можете выбрать пункт **Reference** меню **Window** или нажать комбинацию клавиш <Shift>+<F1>. При этом в окне руководства будет отображена справка по тегу, на котором находится курсор в окне редактирования HTML-кода.

Пользуясь панелью **Reference**, вы можете получить справку о каждом атрибуте любого тега HTML. Например, давайте узнаем побольше о теге `<B>` (как помните, он делает заключенный в него текст жирным). Выберите в раскрывающемся списке **Tag** название этого тега — и в панели появится его описание.

Справа от раскрывающегося списка **Tag** находится другой раскрывающийся список, в котором выбираются атрибуты. Сейчас там вы видите слово "Description". Это знак того, что на панели **Reference** отображаются сведения о самом теге. Если вы хотите просмотреть сведения об атрибуте, выберите в списке атрибутов нужный.

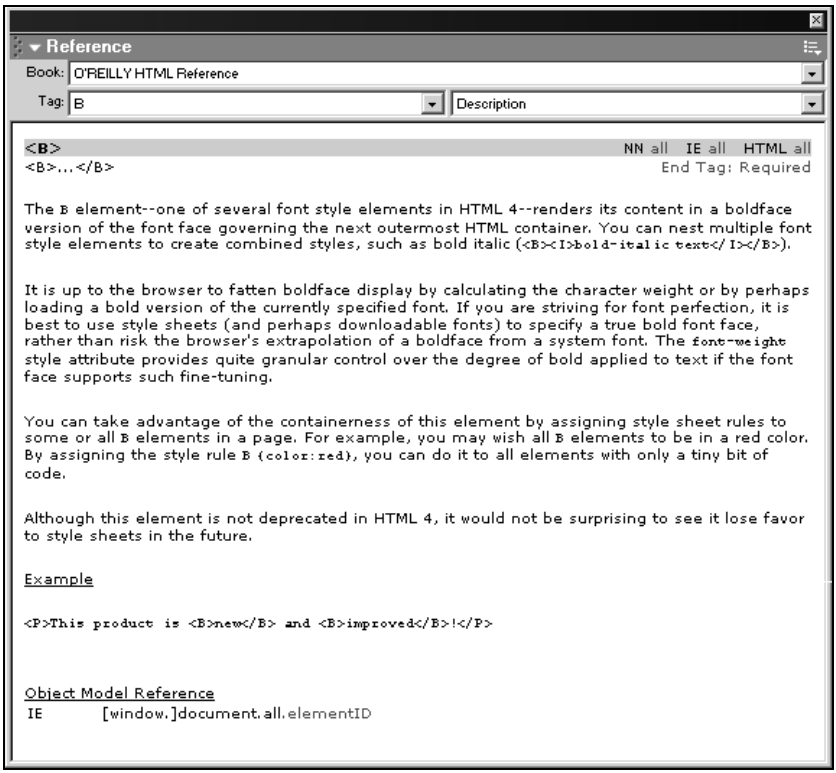


Рис. 2.33. Панель **Reference**

Рис. 2.34. Кнопка **Reference**

Настройка Dreamweaver

А сейчас самое время настроить наш Dreamweaver так, чтобы он нормально понимал русский язык. Заодно мы расскажем о других настройках, которые могут вам пригодиться в дальнейшем.

Вся работа будет происходить в многофункциональном окне настройки программы, состоящем из множества вкладок с разными элементами управления. Чтобы вызвать его, выберите пункт **Preferences** меню **Edit** или нажмите комбинацию клавиш <Ctrl>+<U>. В левой части окна настройки отображен список вкладок, а в правой появляется само содержимое выбранной вкладки.

Учим русский

Выберем в списке вкладок пункт **New Document**. Окно настройки примет вид, показанный на рис. 2.35. Итак, что же здесь изображено?

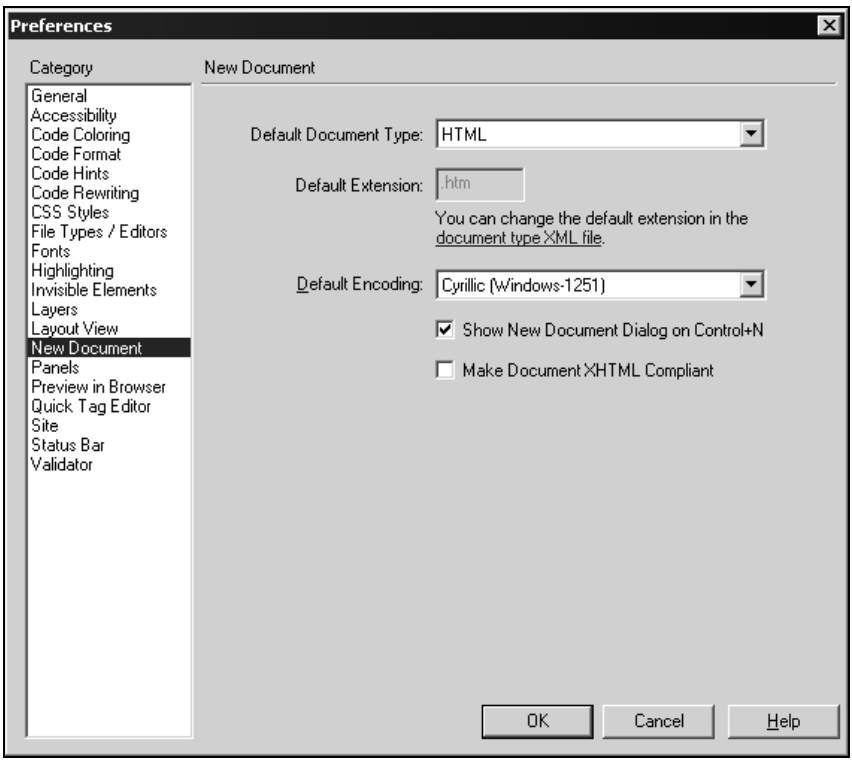


Рис. 2.35. Вкладка **New Document** диалогового окна **Preferences**

Но прежде чем начать разговор о *русификации* Dreamweaver, немного поговорим об особенностях национального Web-творчества. А именно о кодировках русского языка и борьбе с ними.

Вероятно, вы знаете, что каждый символ, который может быть введен с клавиатуры и отображен на экране, имеет уникальный номер, называемый *кодом символа*. Совокупность таких кодов вместе с описанием, какой код какому символу соответствует, образует *кодировку*. Каждая кодировка имеет свое наименование, например 1251 или КОИ-8.

Поскольку любой язык использует свой набор символов, для каждого языка кодировки, как правило, различны. (Исключение — некоторые западноевропейские языки.) Но на этом путаница с кодировками не кончается. Дело в том, что разные операционные системы используют различные кодировки. Например, западноевропейская версия Windows использует кодировку

1250, русская — 1251, американская версия MS-DOS — 437, а русская — 866 (она же ISO-8859-5). Ну, американская с западноевропейской — бог с ними, обойдемся без иноземцев! Однако русских кодировок, как видите, уже две. А если добавить сюда еще кодировку, используемую русской версией операционной системы UNIX — КОИ-8, и русской версией компьютеров Macintosh — MacCyrillic, кодировок станет уже четыре. И это только главные, на памяти автора существовали еще штуки четыре менее распространенных кириллических кодировок ("основная" кодировка ГОСТ, "болгарская", "американская", "югославская" и еще какие-то). Кроме того, в последнее время появилась кодировка Unicode, поддерживающая ВСЕ имеющиеся на Земле языки. Настоящая тирания кодировок!..

Чем все это грозит? А вот чем. Вы, наверно, пытались открыть текстовый документ, созданный в Блокноте, в Norton Commander. Видели, что при этом получается — текст абсолютно нечитаем. А все потому, что русские кодировки 866 (MS-DOS) и 1251 (Windows) не совпадают! В них одному и тому же символу присвоены разные коды!!!

Каков же выход?

Выхода нет. Можно надеяться только на то, что какая-то из кодировок станет стандартом и постепенно вытеснит конкурентов. Пока что на роль такого (негласного) стандарта претендует 1251, хотя интернетчики старого поколения, пользующиеся UNIX-совместимыми системами, "пропихивают" КОИ-8. Во всяком случае, сейчас большинство Web-страниц, имеющихся в русском сегменте Сети, написано в кодировке 1251.

Здесь стоит упомянуть еще два момента. Современные программы Web-обозревателей поддерживают все доступные сейчас кодировки и корректно их распознают. Это первое. Второе: Web-сервер (точнее, его администратор) может потребовать, чтобы публикуемые вами странички были закодированы в какой-либо конкретной кодировке, например в КОИ-8. Это стоит иметь в виду, когда вы будете выбирать кодировку для своего Web-творения.

Когда вы создаете в Dreamweaver Web-страницу, используемая в ней кодировка прописывается в ее заголовке с помощью особого тега <META>. Например, так:

```
<META HTTP-EQUIV="Content-Type" CONTENT="text/html;
↳charset=windows-1251"></HEAD>
```

Как вы поняли, эта страница создана с использованием кодировки Windows, т. е. 1251. Подробнее о теге <META> мы поговорим далее в этой книге.

Итак, какие же кодировки поддерживает Dreamweaver? (Имеются в виду, конечно же, русские кодировки.) Все они перечислены в табл. 2.4 и задаются с помощью раскрывающегося списка **Default Encoding**.

Таблица 2.4. *Кодировки русского текста, поддерживаемые Dreamweaver*

Обозначение	Описание
ISO-8859-5	Русская версия MS-DOS
КОИ8 (KOI-8R)	Русские версии UNIX-совместимых систем
MacCyrillic	Русская версия ОС Macintosh
Windows-1251	Русские версии Windows, начиная от версии 3.0
Western (Latin1)	Это не русская кодировка, она включена в этот список только для справки. Поддерживает западноевропейские языки

Какую же кодировку выбрать? Ответ прост. Если вы не связаны какими-либо специфическими требованиями администратора Web-сервера, на котором будет опубликован ваш сайт, смело выбирайте пункт **Windows-1251**. В противном случае выберите ту кодировку, которую требует сервер. Если вы создаете странички на английском языке, ваш выбор — **Western (Latin1)**.

Теперь переключитесь на вкладку **Fonts** (рис. 2.36). На этой вкладке вы сможете настроить шрифты, которыми будет отображаться текст вашей страницы. В списке **Font Settings** выберите шрифтовой набор, который будет использован для отображения ваших Web-страниц. Здесь альтернатива еще проще: если текст русский — выбирайте **Cyrillic**, если английский — **Western (Latin1)**.

Что касается начертаний и размеров шрифтов, используемых для отображения текста, автор может только посоветовать, но никак не порекомендовать. Автор предпочитает в качестве пропорционального шрифта (раскрывающийся список **Proportional Font**) Arial, в качестве моноширинного (**Fixed Font**) — Lucida Console, а для отображения исходного HTML-кода в редакторе кода (**Code Inspector**) — тоже Lucida Console. Размеры шрифтов (раскрывающийся список **Size**) автор обычно ставит равным 10 пунктам (малый размер, **Small**). Но, еще раз повторим, что это дело вкуса.

А теперь еще одна важная деталь. К сожалению, все программы имеют ошибки, даже самые лучшие из них. Dreamweaver в этом случае не исключение. Из-за ошибки он некорректно открывает Web-страницы, в которых не прописана с помощью тега `<META>` используемая в них кодировка. Для того чтобы вразумить его, нам придется сделать следующее.

Прежде всего, закройте Dreamweaver. Далее откройте в Проводнике или в другом диспетчере файлов папку, в которой у вас установлен Dreamweaver. Обычно это папка Program Files/Macromedia/Dreamweaver MX. В ней вы увидите папку Configuration. Откройте в ней подпапку Encodings. В этой подпапке находится файл EncodingMenu.xml. В этом файле перечислены все поддерживаемые Dreamweaver кодировки.

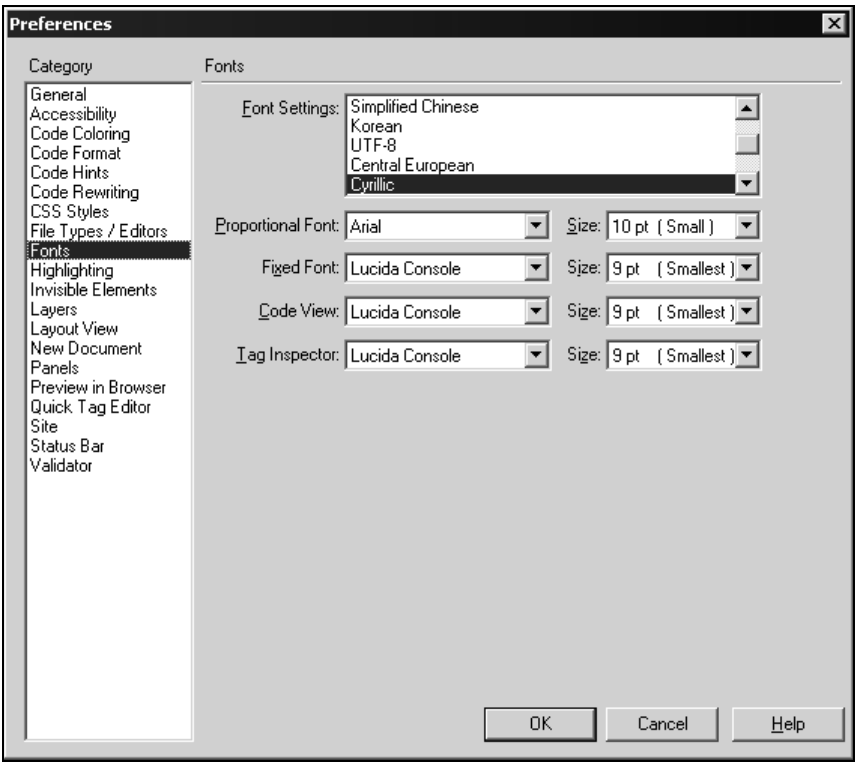


Рис. 2.36. Вкладка **Fonts** диалогового окна **Preferences**

Ниже приведен фрагмент этого файла, в котором перечисляются русские кодировки, интересующие нас:

```
<mm:encoding name="Cyrillic (ISO-8859-5)" charset="iso-8859-5"
fontgroup="Cyrillic" winfontcharset=204 macfontscript=7
filename="iso88595.xml"/>
<mm:encoding name="Cyrillic (KOI8-R)" charset="KOI8-R"
fontgroup="Cyrillic" winfontcharset=204 macfontscript=7
filename="KOI8R.xml"/>
<mm:encoding name="Cyrillic (MacCyrillic)" charset="x-mac-cyrillic"
fontgroup="Cyrillic" winfontcharset=204 macfontscript=7
filename="MacCyrillic.xml"/>
<mm:encoding name="Cyrillic (Windows-1251)" charset="windows-1251"
fontgroup="Cyrillic" winfontcharset=204 macfontscript=7
filename="Win1251.xml"/>
```

Кстати, данные в этом файле записаны в формате XML. Dreamweaver понимает этот формат и очень часто использует его для сохранения конфигурационных данных.

Дело в том, что из-за ошибки Dreamweaver использует для представления текста страниц с непрописанной кодировкой ту, которая встретится ему первой. В данном случае это кодировка MS-DOS — ISO-8859-5. Нам нужно поместить на первое место кодировку 1251. Для этого исправьте файл EncodingMenu.xml так:

```
<mm:encoding name="Cyrillic (Windows-1251)" charset="windows-1251"
fontgroup="Cyrillic" winfontcharset=204 macfontscript=7
filename="Win1251.xml"/>
<mm:encoding name="Cyrillic (ISO-8859-5)" charset="iso-8859-5"
fontgroup="Cyrillic" winfontcharset=204 macfontscript=7
filename="iso88595.xml"/>
<mm:encoding name="Cyrillic (KOI8-R)" charset="KOI8-R"
fontgroup="Cyrillic" winfontcharset=204 macfontscript=7
filename="KOI8R.xml"/>
<mm:encoding name="Cyrillic (MacCyrillic)" charset="x-mac-cyrillic"
fontgroup="Cyrillic" winfontcharset=204 macfontscript=7
filename="MacCyrillic.xml"/>
```

После этого сохраните этот файл и закройте его. Теперь можете запускать Dreamweaver — он станет корректно открывать все Web-страницы.

Настраиваем скорость интернет-соединения

Выше было сказано, что строка статуса окна документа содержит секцию, где показывается примерный размер открытой в окне Web-страницы и время ее загрузки. И еще упоминалось, что это время считается с учетом скорости 28,8 Кбит/с, заданной по умолчанию в настройках Dreamweaver. Вы можете изменить эту скорость.

Выберите в окне настроек вкладку **Status Bar**. На экране появится вкладка настройки строки статуса (рис. 2.37). Среди всех элементов управления, размещенных на ней, нас интересует только раскрывающийся список **Connection Speed**, в котором выбирается скорость соединения в килобитах в секунду. Выберите нужную вам скорость или введите ее вручную.

Добавляем программы просмотра Web-страниц

Если на вашем компьютере установлена одна программа Web-обозревателя, настройка закончена, и вы можете нажать кнопку **OK**. Если же таких программ у вас несколько (например, Microsoft Internet Explorer и Netscape Navigator), то вы, скорее всего, захотите добавить вторую из них в список Dreamweaver, чтобы впоследствии просматривать в ней разрабатываемые Web-страницы. Как это сделать, сейчас объясним.

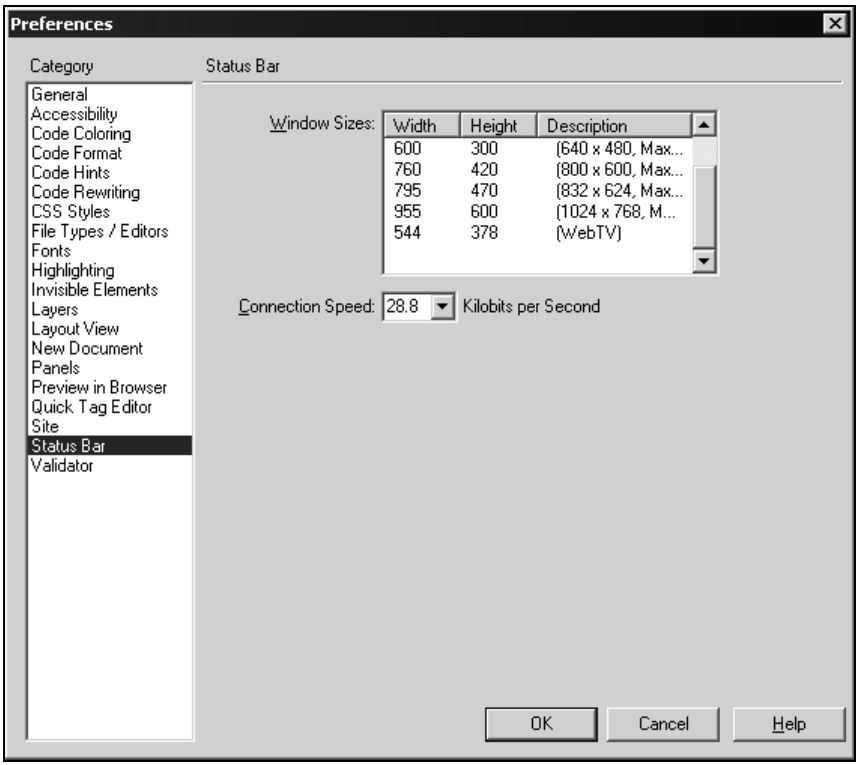


Рис. 2.37. Вкладка **Status Bar** диалогового окна **Preferences**

Выберем вкладку **Preview in Browser**. То, что вы увидите, показано на рис. 2.38.

В списке **Browsers** перечислены все программы просмотра, которые смог найти Dreamweaver при установке. Сейчас там всего одна строка: **iexplore**. Но мы исправим это положение.

Нажмите кнопку со знаком "плюс", расположенную выше списка. На экране появится окно ввода сведений о программе просмотра, показанное на рис. 2.39.

Здесь все просто. В поле ввода **Name** вводится имя программы, которое будет появляться в списке и меню Dreamweaver. Это имя желательно сделать более вразумительным, чем малопонятное "iexplore". В поле ввода **Application** вводится путь доступа к исполняемому файлу программы. Но т. к. вы вряд ли помните его наизусть, нажмите кнопку **Browse**, выберите нужный файл в стандартном диалоговом окне открытия файла и нажмите кнопку открытия.

А что делают флажки **Primary Browser** и **Secondary Browser**? Вот на них стоит остановиться подробно.

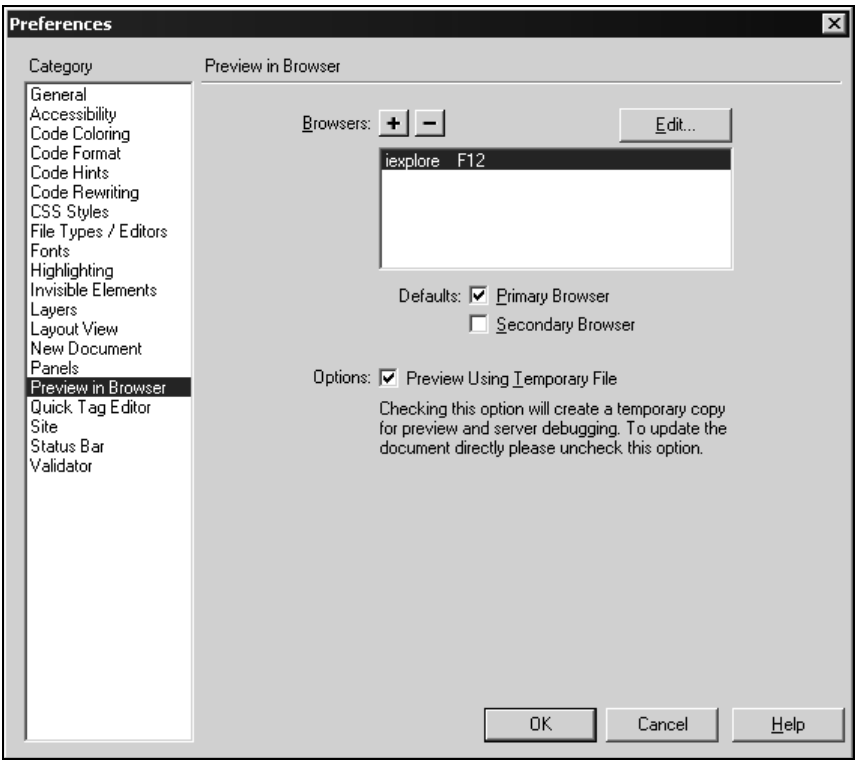


Рис. 2.38. Вкладка **Preview in Browser** диалогового окна **Preferences**

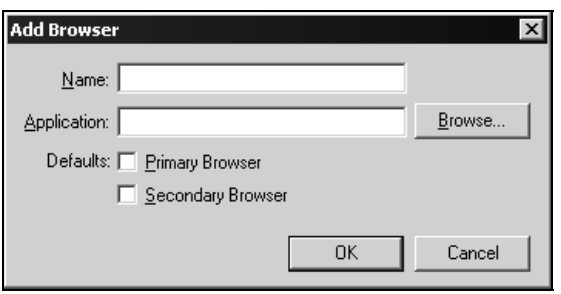


Рис. 2.39. Окно сведений о программе просмотра

Dreamweaver позволяет вам из всех занесенных в его список программ просмотра выбрать двоих "любимчиков". Этих "любимчиков" вы сможете впоследствии вызывать нажатием одной клавиши или комбинацией клавиш. Один из них станет первичным (primary) и будет вызываться клавишей <F12> (по умолчанию это Internet Explorer), а второй — вторичным (secondary), и "отвечать" за него будет комбинация клавиш <Ctrl>+<F12>. Как вы

уже поняли, флажок **Primary Browser** задает первичного "любимчика", а **Secondary Browser** — вторичного. Остается только добавить, что из этих флажков может быть включен только один, т. е. программа просмотра может быть только первичным или только вторичным "любимчиком" либо не является таковым вообще.

Закончив ввод данных о новой программе просмотра, нажмите кнопку **OK**. Внесенная вами программа добавится в список **Browsers**.

Как вы уже заметили, флажки **Primary Browser** и **Secondary Browser** имеются также в окне настройки. Это позволит вам изменить статус программы просмотра, не открывая окна параметров: просто выделите в списке необходимую строку и включите или отключите нужный флажок. Кнопка **Edit** позволит вам изменить сведения о программе просмотра; при ее нажатии на экране появится окно параметров. Кнопка со знаком "минус", расположенная выше списка **Browsers**, позволит вам удалить ненужную программу просмотра. Но будьте осторожны: программа удаляется из списка сразу же, без всякого предупреждения!

Добавляем внешний HTML-редактор

Всем хорош Dreamweaver. И Web-страницы позволяет редактировать, и до HTML-кода добраться несложно. Однако иногда возникает необходимость во внешнем HTML-редакторе. Например, если нужно сделать что-то такое, чего Dreamweaver не поддерживает (да-да, может быть и так), или просто немного поработать в привычном HTML-редакторе. Как это можно сделать?

Во-первых, просто запустить необходимую программу из меню **Пуск** и открыть в ней нужный файл. Это удобно, если вы прибегаете к внешнему HTML-редактору изредка.

Во-вторых, можно интегрировать внешний редактор в среду Dreamweaver. Эта возможность незаменима для тех, кто часто пользуется подобными программами.

Переключитесь на вкладку **File Types/Editors** диалогового окна настроек (рис. 2.40). В поле ввода **External Code Editor** введите путь и имя файла программы внешнего редактора. Но проще всего будет щелкнуть кнопку **Browse** и выбрать нужный файл в появившемся на экране диалоговом окне открытия файла Windows.

Какую программу лучше всего использовать в качестве внешнего HTML-редактора? Здесь все зависит от ваших вкусов. Можно использовать какой-либо простейший текстовый редактор для редактирования "сырого" HTML-кода: Блокнот или его более мощный аналог — Notepad+, доступный по адресу **http://dimonius.da.ru**, а можно пользоваться старым добрым Arachnophilia. Также можно работать со сложным визуальным Web-редактором, например FrontPage Express, поставляемым стандартно с Microsoft Internet

Explorer. (Кстати, некоторые вещи, специфичные для Internet Explorer, лучше делать во FrontPage Express.) В общем, никто не ограничивает вас в выборе внешнего HTML-редактора по своему вкусу.

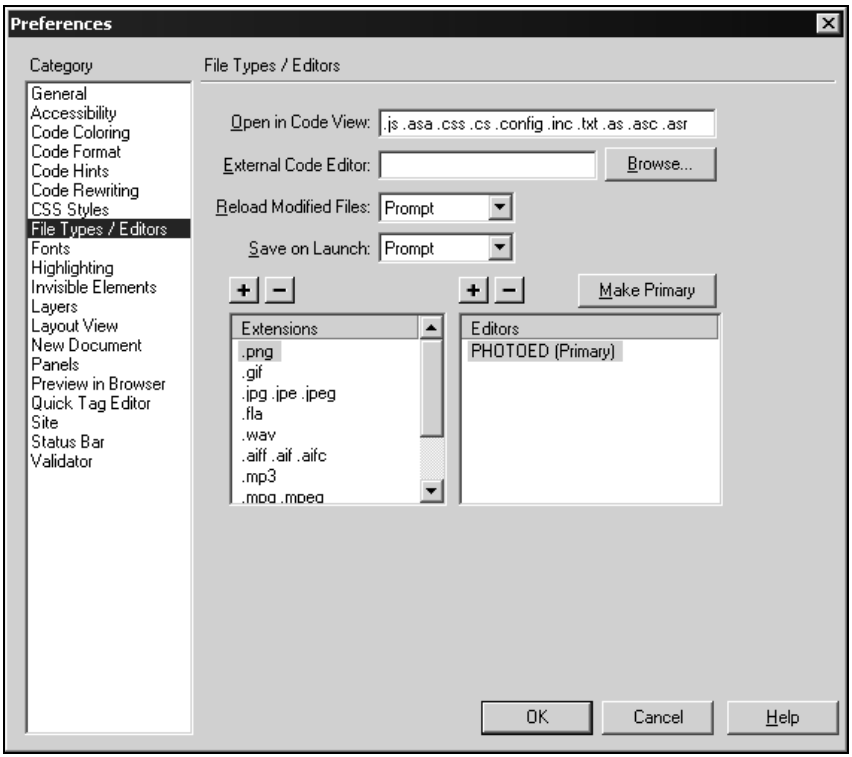


Рис. 2.40. Вкладка **File Types / Editors** диалогового окна **Preferences**

По окончании настройки Dreamweaver нажмите кнопку **OK**, чтобы сохранить сделанные установки. Конечно, возможности настройки, предлагаемые этой программой, очень велики, но мы не будем о них рассказывать. Все необходимые сведения вы можете получить из интерактивной справки. Мы рассказали только о тех настройках, без которых вам будет трудно работать с Dreamweaver.

Что дальше?

Выяснив, что есть что в Dreamweaver, можно приступать к созданию наших первых Web-страничек.

В следующей главе мы создадим первую, главную и самую простенькую Web-страницу. На ней не будет ничего, кроме форматированного текста и нескольких гиперссылок. Итак, начнем!

ГЛАВА 3

Начинаем с текста

В этой главе мы, собственно, и начнем работу с Dreamweaver. На первый раз мы не будем усложнять себе жизнь и ограничимся простейшей Web-страницей с краткими сведениями об авторе и набором ссылок на другие, пока еще не существующие страницы Web-сайта.

Здесь мы будем работать с текстом — самым важным составляющим современных Web-страниц. Представьте, что вы работаете в вашем любимом текстовом редакторе, и — вперед!

Создание новой Web-страницы

Но сначала нужно создать новую Web-страницу.

Как вы помните, для этого нужно выбрать пункт **New** в меню **File** или нажать комбинацию клавиш <Ctrl>+<N>. После этого появится диалоговое окно **New Document** (рис. 3.1).

Dreamweaver MX поддерживает создание Web-страниц на основе шаблонов. *Шаблон* — это особым образом подготовленная и сохраненная Web-страница, на которой уже имеются некоторые элементы, которые вам могут понадобиться (сведения об авторских правах, название и т. п.). Вам останется только добавить основной текст и немного исправить оформление — и готовая профессионально оформленная страница у вас в руках. Если вам нужно быстренько "сляпать" страничку с каким-то текстом, шаблоны — настоящая находка.

Однако настоящее творчество не терпит суеты. Поэтому давайте начнем с чистого листа и, не торопясь, создадим свой Web-шедевр.

Взгляните на рис. 3.1. Вы видите, что слева расположен список **Category**, где вы выбираете категорию шаблонов. Сами шаблоны, относящиеся к выбранной категории, перечислены в списке **Basic Page**. А внешний вид вы-

бранного шаблона во многих случаях можно просмотреть в расположенной справа панели предварительного просмотра.

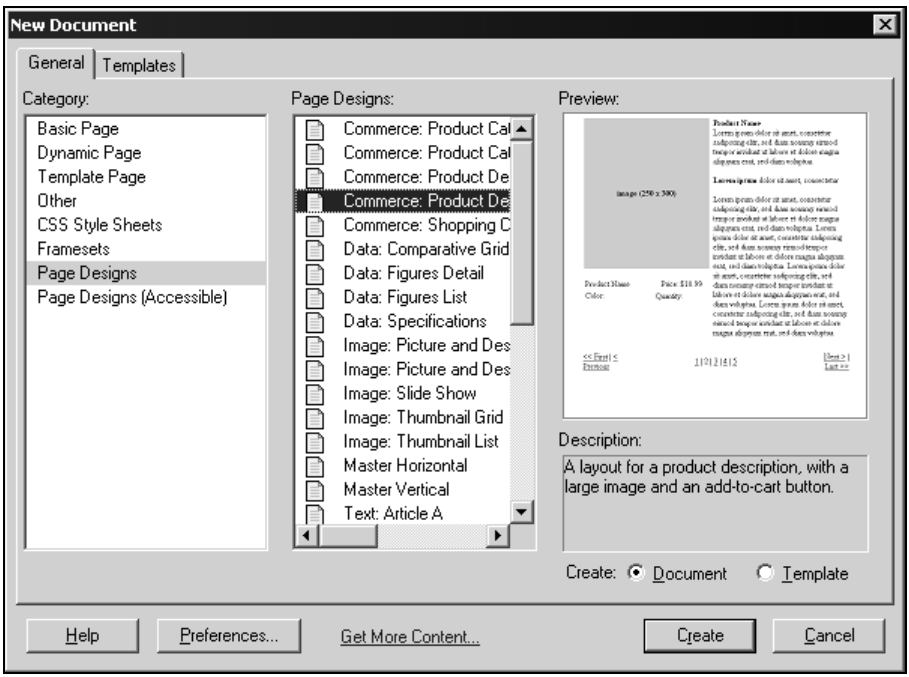


Рис. 3.1. Диалоговое окно **New Document**

Итак, нам нужен "пустой" шаблон. Выберем в списке **Category** пункт **Basic Page**, а в списке **Basic Page** — пункт **HTML**, после чего нажмем кнопку **OK**. На экране появится пустое окно документа. Новая Web-страница создана.

Работа с текстом

Здесь мы опишем работу с текстом в окне документа Dreamweaver на примере нашей первой Web-страницы.

Ввод текста

Вот и наступила торжественная минута! Сейчас мы наберем наш первый текст в окне документа Dreamweaver (рис. 3.2).

Набрав текст, сразу же сохраним его в файле под именем 3.1.htm. И поместите ее в отдельную папку — так будет проще управляться со всем нашим Web-хозяйством.

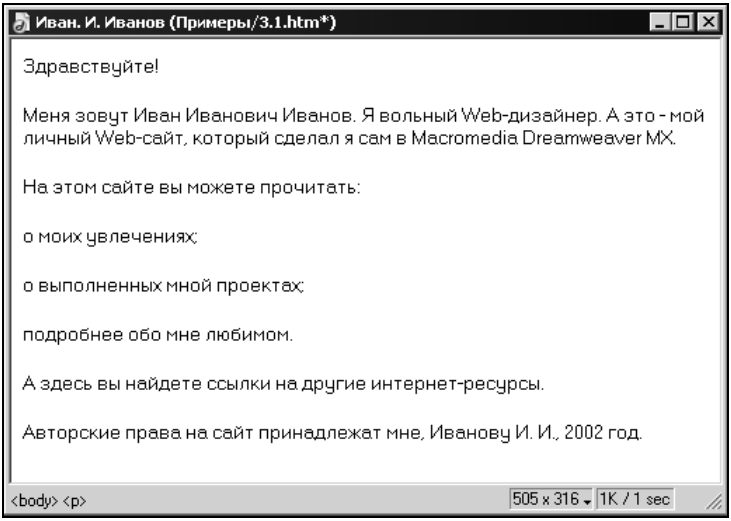


Рис. 3.2. Наш первый текст

Вы уже знаете, что одна из Web-страниц на Web-сервере задается в качестве страницы по умолчанию. Также вы знаете, что такая страница обычно носит имя default или index (и расширение htm или html). Но в данном случае мы пренебрегли этим соглашением, хотя наша первая страница так и просится быть страницей по умолчанию. Давайте назовем ее 3.1.htm — таким образом, мы не запутаемся в дальнейшем во множестве разнокалиберных страниц, названных как бог на душу положит. Не беспокойтесь по поводу нарушения соглашений — это ненадолго.

Итак, мы создали первую нашу Web-страницу... Ах да, мы же совсем забыли о названии! Помните название, задаваемое тегом `<TITLE>`, которое не отображается в окне Web-обозревателя, но выводится в его заголовке. Кроме того, это название помещается в списке *"истории"* Web-обозревателя, где содержатся названия и адреса всех посещенных в течение некоторого времени страниц. Давайте зададим его для удобства будущих посетителей сайта. Тем более что это делается очень просто: наберите его в поле ввода, находящемся в инструментарии документа (рис. 3.3).

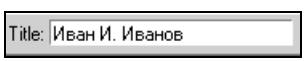


Рис. 3.3. Поле ввода названия Web-страницы в инструментарии документа

Сохраним нашу страницу еще раз. И посмотрим на нее критически.

Чего ей не хватает?

Конечно, вы уже посетили много Web-сайтов. И вы видели, как красочно они оформлены. По сравнению с ними наша первая страничка выглядит не

просто бледно — она совсем никак не выглядит. Давайте приведем ее в пристойный вид.

Форматирование абзацев

Прежде всего, давайте сделаем нашей странице нормальный "кричащий" заголовок. (Имеется в виду не HTML-заголовок, а обычный заголовок, как у газеты.) Первой строкой как раз набрано "Здравствуйте" — она прекрасный кандидат в заголовки.

Чтобы отформатировать текстовый абзац как заголовок, нужно воспользоваться редактором свойств, точнее, раскрывающимся списком форматов абзаца. Он показан на рис. 3.4.

Пункт **Paragraph** этого списка форматирует текст как обычный абзац (отмечаемый тегом <P>). Пункты **Heading 1**, ..., **Heading 6** позволяют превратить его в заголовок, соответственно, первого, второго уровня и т. д. Пункт **Preformatted** превращает абзац в текст *фиксированного формата*, где форматирование задано не тегами HTML, а как в обычном тексте: отступы создаются пробелами, а разбиение на абзацы — символами возврата каретки и перевода строки. Это может быть очень полезно при выкладывании в Сеть больших текстовых документов, созданных в старых текстовых редакторах, без сложного HTML-форматирования.

Рис. 3.4. Раскрывающийся список форматов абзаца, находящийся в редакторе свойств

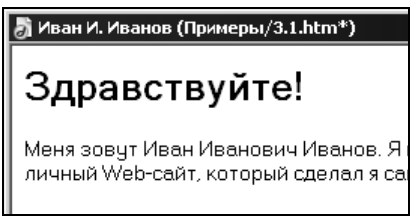


Рис. 3.5. Созданный нами заголовок

Примечание

Заголовки создаются парными тегами <H*n*>, где *n* — число от 1 до 6. Текст фиксированного формата задается парным тегом <PRE>.

Поставим текстовый курсор на строку "Здравствуйте" и выберем в меню форматов пункт **Heading 1** — это наилучшим образом подходит для приветствия. Получившийся результат показан на рис. 3.5.

Если вы закрыли редактор свойств, не беспокойтесь — то же самое вы можете сделать, используя главное меню или контекстное меню окна документа. В первом случае выберите в меню **Text** пункт **Paragraph Format** и в по-

явившемся на экране подменю — нужный формат абзаца. Во втором случае щелкните правой кнопкой мыши на строке приветствия, также выберите пункт **Paragraph Format** и в подменю — формат абзаца.

Если же вы предпочитаете не отрывать пальцы от клавиатуры, можете задать формат заголовка комбинацией клавиш <Ctrl>+<*n*>, где <*n*> — одна из клавиш <1>, ..., <6>. Формат обычного абзаца можно задать, нажав комбинацию клавиш <Ctrl>+<Shift>+<P>.

Теперь давайте выровняем заголовок по центру. В этом нам помогут кнопки выравнивания редактора свойств (рис. 3.6).

Рис. 3.6. Кнопки выравнивания, расположенные в редакторе свойств

Эти кнопки задают выравнивание соответственно (порядок перечисления слева направо):

- по левому краю;
- по центру;
- по правому краю;
- выравнивание по обоим краям (полное выравнивание).

Внимание!

Полное выравнивание поддерживается только новыми версиями Web-обозревателей.

Одновременно может быть нажата (включена) только одна из этих кнопок либо ни одной. (В последнем случае задается так называемое выравнивание по умолчанию, как правило, по левому краю.) Такие кнопки называются *кнопками-переключателями*, по аналогии с пунктами-переключателями меню.

Итак, снова установим текстовый курсор на заголовок и нажмем кнопку выравнивания по центру. На рис. 3.7 показано, что у нас получится в этом случае.

Рис. 3.7. Центрированный заголовок

Аналогичного результата вы можете достигнуть, воспользовавшись подменю **Align** меню **Text** или аналогичным подменю контекстного меню. Пункт-переключатель **Left** задает выравнивание по левому краю, **Center** — по центру, **Right** — по правому краю, а **Justify** — полное выравнивание. Также вы можете нажать комбинацию клавиш <Ctrl>+<Alt>+<Shift>+<клавиша>, где <клавиша> — <L>, <C>, <R> или <J>, соответственно.

Примечание

Выравнивание задается значением атрибута `ALIGN` тегов `<P>` и `<Hn>`. Доступны значения `left`, `center`, `right` или `justify`. Что они означают, вы уже знаете.

Теперь давайте как-нибудь выделим последнюю строку со сведениями об авторских правах. Можно поставить текстовый курсор в ее начало и нажать клавишу <Enter>, чтобы вставить между этой строкой и остальным текстом дополнительную пустую строку. А можно поступить по-другому.

Взгляните на кнопки, показанные на рис. 3.8. Эти кнопки позволяют установить или убрать *отступ* слева у абзаца. Левая кнопка уменьшает (убирает) отступ, а правая — увеличивает.

Рис. 3.8. Кнопки задания отступа абзаца, находящиеся в редакторе свойств

Давайте поставим курсор мыши на последнюю строку текста и нажмем кнопку увеличения отступа. Результат показан на рис. 3.9. Вы можете еще раз нажать эту кнопку для увеличения отступа или левую кнопку, чтобы уменьшить отступ.

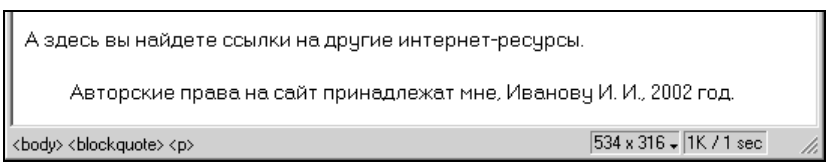


Рис. 3.9. Строка с отступом

Аналогичный результат дают пункты **Outdent** и **Indent** меню **Text** или подменю **List** контекстного меню. Вы также можете нажать комбинацию клавиш <Ctrl>+<Shift>+<]> для увеличения отступа и <Ctrl>+<Shift>+<[> — для его уменьшения.

Примечание

Отступ задается парным тегом `<BLOCKQUOTE>`.

Теперь обратим внимание на список тем, доступных на сайте. (Из строк этого списка потом будут сделаны гиперссылки, указывающие на соответствующие страницы.) И вспомним, что "обычные" текстовые редакторы поддерживают создание нумерованных и маркированных списков. Пункты *нумерованных* (упорядоченных) списков, как вы знаете, обозначаются порядковыми номерами, а пункты *маркированных* (неупорядоченных) списков — какими-либо значками. Давайте и мы создадим такой список.

Для этого, прежде всего, выделите все строки, которые войдут в список. Всего их три.

Для того чтобы преобразовать выделенные строки в список, опять воспользуемся кнопками редактора свойств. Они показаны на рис. 3.10. Левая кнопка создает маркированный список, а правая — нумерованный. В данный момент времени может быть нажата только одна кнопка либо ни одной; в последнем случае абзац возвращается к своему обычному виду.

Рис. 3.10. Кнопки создания списков, расположенные в редакторе свойств

Нажмите левую кнопку. Созданный нами маркированный список показан на рис. 3.11.

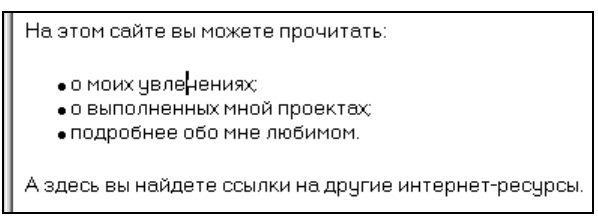


Рис. 3.11. Маркированный список

В меню **Text** и в контекстном меню предусмотрено специальное подменю **List**, выполняющее те же действия. В табл. 3.1 описаны все пункты-переключатели этого подменю. Каких-либо комбинаций клавиш для них не предусмотрено.

Таблица 3.1. *Пункты подменю* ***List***

Пункт	Описание
None	Обычный абзац текста
Unordered List	Маркированный список
Ordered List	Нумерованный список

Таблица 3.1 *(окончание)*

Пункт	Описание
Definition List	*Список определений.* Каждая нечетная строка становится как бы термином, требующим разъяснения, а каждая четная — этим самым разъяснением

Кроме того, вы получаете возможность изменить некоторые дополнительные параметры списка. Для этого нажмите кнопку **List Item** на панели редактора свойств (рис. 3.12). На экране появится диалоговое окно **List Properties** (рис. 3.13). Давайте рассмотрим это окно подробнее.

Рис. 3.12. Кнопка доступа к дополнительным свойствам списков

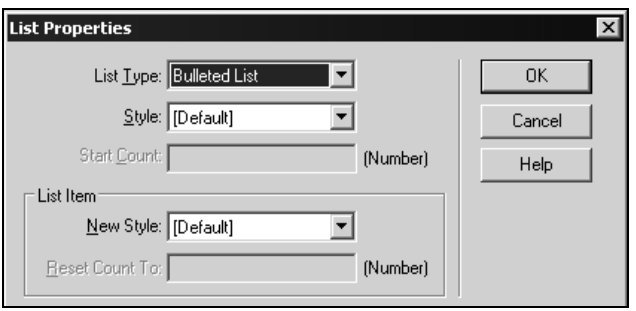


Рис. 3.13. Диалоговое окно **List Properties**

Раскрывающийся список **List Type** позволяет задать *тип* списка. Здесь доступны следующие позиции:

- **Bulleted List** — маркированный список;
- **Numbered List** — нумерованный список;
- **Directory List** — *список папок*, визуально похож на маркированный список;
- **Menu List** — *список-меню*, визуально опять же похоже на маркированный список.

Если выбран первый или второй пункт, становятся доступными другие элементы управления.

Раскрывающийся список **Style** позволяет задать *стиль* маркера или нумерации списка. Если выбран маркированный список, доступны следующие значения:

- **Bullet** — кружок с заливкой;
- **Square** — квадратик;
- **[Default]** — маркер по умолчанию, обычно кружок с заливкой.

Если же выбран нумерованный список, доступны:

- **Number** — арабские цифры;
- **Roman Small** — малые римские цифры;
- **Roman Large** — большие римские цифры;
- **Alphabet Small** — малые латинские буквы;
- **Alphabet Large** — большие латинские буквы;
- **[Default]** — нумерация по умолчанию, обычно арабские цифры.

Кроме того, если выбран нумерованный список, доступно поле ввода **Start Count**, где вы сможете ввести номер, с которого начнется нумерация строк списка.

Все эти настройки относились к списку в целом. В нижней половине окна расположена группа элементов управления **List Item**, затрагивающая только ту строку списка, на которой находится текстовый курсор. В эту группу входят раскрывающийся список **New Style** и поле ввода **Reset Count To**, аналогичные раскрывающемуся списку **Style** и полю ввода **Start Count** верхней части окна.

Закончив настройку свойств, нажмите кнопку **OK** для сохранения установок или **Cancel** — для отказа от них.

Вы можете немного "поиграться" с этими дополнительными настройками, чтобы узнать их получше. Лишняя практика никогда не повредит.

Примечание

Списки задаются парными тегами <UL> (маркированный), <OL> (нумерованный), <DIR> (список папок) и <MENU> (меню). Между этими тегами помещаются отдельные строки, задаваемые тегом <LI>. Удивительно, но у тега <LI> нет пары!

Форматирование отдельных символов

Итак, мы научились форматировать целиком абзацы текста. Теперь поговорим, что можно сотворить с отдельными символами текста.

Все текстовые редакторы, например Microsoft Word или поставляемый в составе Windows WordPad, позволяют менять начертание и размер шрифта текста, делать его жирным, курсивным или подчеркнутым, а также менять его цвет. Все это доступно и в Dreamweaver. Нужно только иметь в виду, что возможности HTML по форматированию текста сильно ограничены существующими стандартами.

Ладно, к делу.

Начнем с самого простого. Попытаемся сделать некоторые фрагменты текста нашей страницы жирными и курсивными. И помогут нам в этом две кнопки изменения *начертания*, показанные на рис. 3.14.

Рис. 3.14. Кнопки изменения начертания шрифта

Соответственно, левая (**B**) кнопка позволяет сделать шрифт жирным, а правая (***I***) — курсивным. Обе эти кнопки могут быть нажаты или отжаты независимо, т. е. текст может быть простым, жирным, курсивом или жирным курсивом.

Выделим слова "Иван Иванович Иванов" и нажмем кнопку **B**. Текст станет жирным. Теперь выделим полностью последнюю строку (со сведениями об авторских правах) и нажмем кнопку ***I***. Последняя строка станет курсивной. Полученный результат можно увидеть на рис. 3.15.

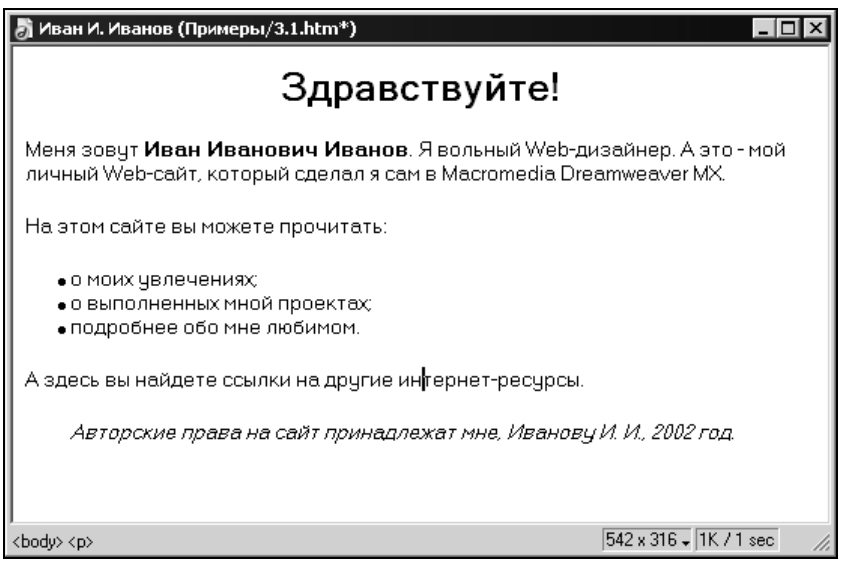


Рис. 3.15. Жирный текст и курсивная строка

Вместо нажатия кнопки **B** вы можете выбрать пункт-выключатель **Bold** в подменю **Style** меню **Text** или контекстного меню. В том же подменю существует пункт **Italic** — аналог кнопки ***I***. Вы также можете нажимать комбинации клавиш <Ctrl>+<B> и <Ctrl>+<I> соответственно.

А вот чтобы включить или отключить подчеркивание текста линией, вам все равно придется воспользоваться пунктом-выключателем **Underline** подменю **Style** меню **Text** или контекстного меню. Ни кнопки, ни комбинации клавиш для этого не предусмотрено.

Вы могли заметить, что в подменю **Style** много пунктов. Давайте их рассмотрим.

Здесь нужно сказать, что все теги HTML, предназначенные для оформления текста, делятся на две большие группы. *Теги физического форматирования* просто говорят Web-обозревателю: "Сделай текст вот таким и не задавай лишних вопросов". Таким образом, если вы сделали текст жирным, то Web-обозреватель просто выведет его жирным шрифтом, не выполняя при этом никакой дополнительной обработки. Такие теги хороши тогда, когда вы не даете оформляемому с их помощью тексту никакого особого значения.

В отличие от них, *теги логического форматирования* дают тексту, составляющему их содержимое, какое-либо дополнительное значение. Например, вы можете превратить с помощью одного из этих тегов фрагмент текста в цитату. Web-обозреватель выведет ее особым шрифтом (как правило, курсивом, хотя, может и вообще никак не выделить) и, вместе с тем, может выполнить какую-то дополнительную обработку этого текста, например, вывести все цитаты, встретившиеся в странице, в отдельное окно. (Правда, ни один современный Web-обозреватель так не делает, но кто знает, что будет дальше...)

Выделение текста жирным и курсивным шрифтом, рассмотренное нами выше, выполняется с помощью тегов физического форматирования. В самом деле, мы просто по-другому выделили некоторые фрагменты текста нашей странички, не давая им какого-то особого значения. И Web-обозреватель нас прекрасно поймет.

Давайте же продолжим изучение подменю **Style**. В нем мы видим пункт-выключатель **Strikethrough**, включив который мы сделаем выделенный фрагмент текста зачеркнутым. Зачеркивание текста выполняется опять же с помощью тега физического форматирования, который мы рассмотрим чуть ниже.

А остальные пункты подменю **Style** задают именно логическое форматирование. В табл. 3.2 приведены все эти пункты и их краткие описания, а на рис. 3.16 — примеры текста, отформатированные с использованием соответствующих пунктов этого подменю. Вы можете сами сделать такой пример и поэкспериментировать с разными стилями текста; сохраните его в файле 3.2.htm.

Таблица 3.2. *Пункты логического форматирования подменю* ***Style***

Пункт	Описание
Teletype	Текст, выведенный устройством вывода компьютера ("телетайп")
Emphasis	Важный текст
Strong	Очень важный текст. Имеет большее значение, чем текст, отформатированный через **Emphasis**

Таблица 3.2 *(окончание)*

Пункт	Описание
Code	Используется для обозначения в тексте фрагментов исходного кода программы на каком-либо языке программирования (команд, имен переменных, ключевых слов и т. п.)
Variable	Используется для обозначения в тексте имен переменных программы на каком-либо языке программирования
Sample	Вывод какой-либо программы
Keyboard	Текст, который пользователь должен ввести с клавиатуры
Citation	Цитата
Definition	Термин, встречающийся в тексте первый раз

Рис. 3.16. Примеры различного форматирования текста с помощью пунктов подменю **Style**

В табл. 3.3 перечислены теги, вставляемые в HTML-код пунктами подменю **Style**.

Таблица 3.3. *Теги, вставляемые в HTML-код пунктами подменю* ***Style***

Пункт	Пара тегов
Bold	`<B>...</B>`
Italic	`<I>...</I>`

Таблица 3.3 *(окончание)*

Пункт	Пара тегов
Underline	`<U>...</U>`
Strikethrough	`<S>...</S>`
Teletype	`<TT>...</TT>`
Emphasis	`<EM>...</EM>`
Strong	`<STRONG>...</STRONG>`
Code	`<CODE>...</CODE>`
Variable	`<VAR>...</VAR>`
Sample	`<SAMP>...</SAMP>`
Keyboard	`<KBD>...</KBD>`
Citation	`<CITE>...</CITE>`
Definition	`<DFN>...</DFN>`

Теперь давайте зададим шрифт, которым набран наш текст. Делается это с помощью двух раскрывающих списков, расположенных в редакторе свойств. Разумеется, Dreamweaver предоставляет в ваше распоряжение также соответствующие пункты меню и комбинации клавиш. Для смены шрифта используется раскрывающийся список, показанный на рис. 3.17, а для смены размера шрифта — список, показанный на рис. 3.18. Но для того, чтобы понять, что же там отображается, нужно дать некоторые разъяснения.

Рис. 3.17. Раскрывающийся список смены шрифта

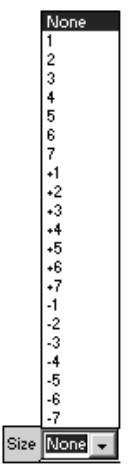


Рис. 3.18. Раскрывающийся список задания размера шрифта

Вспомним, зачем создавалась Всемирная паутина WWW. А именно, чтобы каждый человек, пользуясь любой компьютерной платформой, находясь в любой части света, смог прочитать любой HTML-документ. А т. к. разные компьютерные платформы имеют разные параметры видеоподсистемы, в частности разные стандартные шрифты, то задача одинакового отображения сильно усложняется. Поэтому HTML определяет несколько *стандартных шрифтов*, которые обязательно должны быть установлены на компьютере клиента, и несколько, а именно семь, размеров этих самых шрифтов. Конечно, вы можете использовать в своих Web-страницах другие шрифты, но тогда потрудитесь, чтобы они были установлены у пользователя. Впрочем, если на клиентском компьютере не будет того или иного шрифта, операционная система и Web-обозреватель используют ближайший максимально похожий из уже установленных шрифтов.

Наименования шрифтов в стандарте HTML могут записываться сразу несколько, через запятую. При этом если первого в списке шрифта на клиентском компьютере нет, Web-обозреватель ищет второй, третий и т. д., пока не доберется до стандартного шрифта, который уж точно должен там быть.

Посмотрите еще на рис. 3.17. В этом раскрывающемся списке присутствуют все стандартные шрифты, определенные HTML. Вы можете выбрать любой из них. Если же вас не устраивает ни один, то вы можете просто ввести туда название любого другого шрифта, установленного на вашем компьютере, например Impact, и нажать клавишу <Enter>. Естественно, вся ответственность за правильное отображение текста на клиентском компьютере в данном случае ляжет на вас.

Пункт **Default Font** этого списка сбрасывает шрифтовые установки, точнее, возвращает их к установкам родительского элемента.

Соответственно, в раскрывающемся списке размеров шрифта (рис. 3.18) отображены все доступные размеры шрифтов. Их семь, как уже и говорилось. Кроме того, в этом меню присутствуют пункты вида **+n** и **−n**, где **n** — число от 1 до 7. Выбор этих пунктов позволяет соответственно увеличить или уменьшить размер шрифта на *n* ступеней относительно шрифта родительского элемента. Пункт **None** позволит вернуть размер шрифта к установкам родительского элемента.

Перед тем как изменять наименование или размер шрифта, нужно выделить соответствующий фрагмент текста. Это вам уже знакомо: все уже известные вам команды форматирования текста работают таким же образом.

Давайте выделим имя нашего героя и увеличим его на два размера. Для этого выберем в раскрывающемся списке размера шрифта пункт **+2**. Полученный результат показан на рис. 3.19. Не правда ли, гипотетический Иван Иванович страдает манией величия?

Теперь сменим шрифт. Но не какого-то выделенного фрагмента, а ВСЕГО текста. Для этого выделим весь текст в окне. Конечно, для этого можно

выбрать пункт **Select All** меню **Edit** или нажать комбинацию клавиш <Ctrl>+<A>. Но поступим по-другому. Вы, конечно, помните, что окно документа содержит так называемую секцию тегов (см. рис. 2.12). Давайте щелкнем мышью по кнопке **<body>**, чтобы выделить все содержимое страницы (содержимое тега <BODY>, как вы уже догадались).

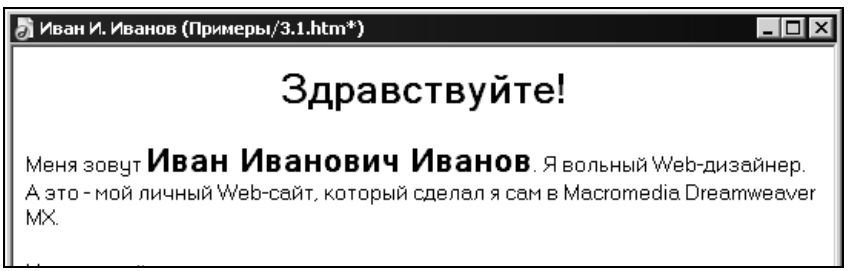


Рис. 3.19. Увеличенный шрифт

Теперь, когда все содержимое документа выделено, выберите нужный шрифт. Для примера выберем пункт **Verdana, Arial, Helvetica, sans-serif** — любимый шрифт автора. (Конечно, автор не собирается навязывать вам свои эстетические вкусы — это просто для примера.) Результат показан на рис. 3.20.

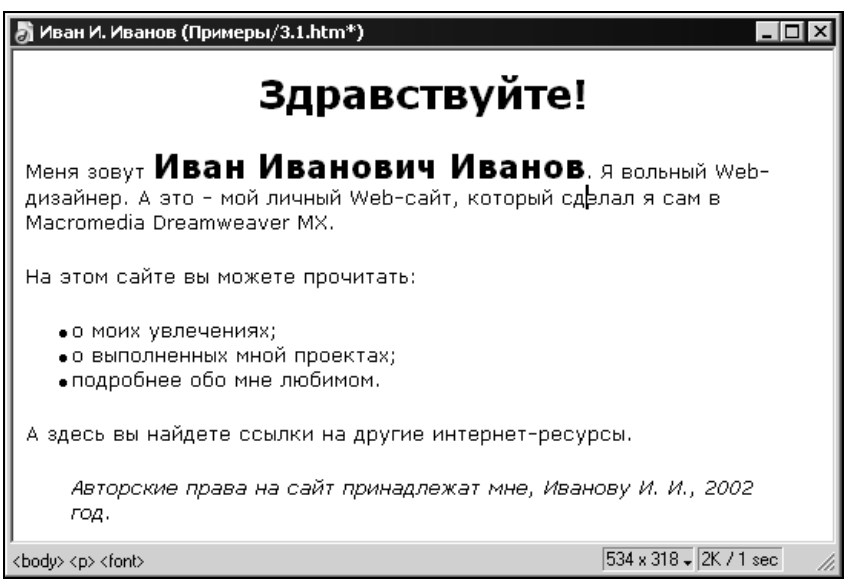


Рис. 3.20. Web-страница после изменения шрифта

Задать шрифт вы также можете с помощью подменю **Font**, расположенного в меню **Text** и контекстном меню. Для смены размера шрифта Dreamweaver предоставляет целых два подменю: **Size** и **Size Change**. Оба они располагаются в меню **Text**. В контекстном меню они объединены в одно подменю **Size**. Никаких клавишных сочетаний для выполнения этих операций с клавиатуры не предусмотрено.

Теперь давайте выделим цветом название программного продукта, с помощью которого делается этот замечательный сайт. Снова обратим внимание на редактор свойств. На рис. 3.21 показан инструмент для задания цвета — так называемый *селектор цвета*.

Рис. 3.21. Селектор цвета Dreamweaver

Селектор цвета состоит из двух частей. Справа находится поле ввода, где вы можете ввести шестнадцатеричный код нужного цвета. Но вряд ли вы помните наизусть шестнадцатеричные коды ваших любимых цветов. Поэтому слева находится кнопка вызова *окна выбора цвета*, которое показано на рис. 3.22.

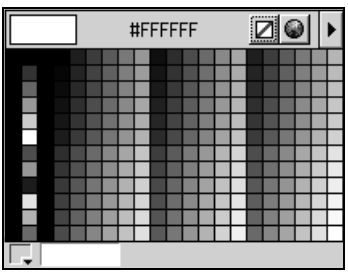


Рис. 3.22. Окно выбора цвета селектора цветов

Большую часть этого окна занимает *палитра*, где, собственно, вы и ищете нужный цвет. При этом в небольшом текстовом поле, расположенном в центре верхней серой полосы окна, высвечивается шестнадцатеричный код цвета, над которым вы в данный момент держите курсор мыши. Это очень удобно: впоследствии вам уже не надо будет долго выбирать цвет из палитры — вы просто введете в поле ввода селектора его код. Если же ни один цвет из палитры вам не подходит, вы можете с помощью того же курсора мыши (имеющего вид пипетки) "взять" необходимый цвет откуда угодно: с рабочего стола, из любого окна, из любого элемента управления. Чтобы закрыть окно выбора цвета, нажмите клавишу <Esc>.

В правом верхнем углу расположены три кнопки. Самая левая из них позволит вам быстро выбрать цвет, заданный по умолчанию (обычно это цвет

родительского элемента). Средняя кнопка вызывает на экран стандартное диалоговое окно выбора цвета. А правая позволит вам переключить режимы палитры выбора цвета. При нажатии на нее на экране появляется дополнительное меню, предлагающее пять режимов:

- **Color Cubes** и **Continuous Tone** — просто меняют внешний вид палитры;
- **Windows OS** и **Mac OS** — позволяют выбрать палитру, специфичную для какой-либо из операционных систем, в данном случае Microsoft Windows и Apple Macintosh;
- **Grayscale** — выбирает черно-белую палитру.

Кроме того, это меню содержит еще один, шестой, изначально включенный пункт-выключатель **Snap to Web Safe**. Зачем он нужен?

Чтобы ответить на этот вопрос, снова поговорим о проблеме межплатформенной совместимости, которую пришлось решать создателям языка HTML и WWW.

Как вы уже знаете, разные компьютерные платформы... да что там! — даже разные компьютеры имеют различные параметры видеоподсистемы. Одни могут отображать всего шестнадцать цветов, а другие — все 16,7 миллиона, что с лихвой перекрывает цветовую разрешающую способность человеческого глаза. Разумеется, при таком богатом множестве компьютерных платформ Web-дизайнеру не стоит и рассчитывать, что все его цвета и оттенки будут отображены правильно везде. Поэтому стандарт HTML определяет так называемую *безопасную палитру* цветов, которая гарантированно должна отображаться правильно всеми программами на всех компьютерах. Web-дизайнерам рекомендуется придерживаться этой безопасной палитры (хотя никто им не запрещает ее игнорировать). Так вот, пункт **Snap to Web Safe**, включенный в дополнительное меню окна выбора цвета, и заставляет селектор цветов Dreamweaver "уважать" безопасную палитру и подгонять под нее все выбираемые пользователем цвета.

Примечание

Все цвета из палитр **Color Cubes** и **Continuous Tone** являются безопасными, цвета из других палитр — нет. Естественно, если вы выбрали какой-то цвет извне окна выбора цветов, он совсем не обязательно уложится в безопасную палитру.

Название изучаемой нами программы уже выделено, так что щелкнем по кнопке селектора цветов и выберем любой понравившийся цвет. Автор выбрал синий — он хорошо успокаивает нервы. Рисунок мы приводить не будем — все равно черно-белая иллюстрация не может передать цвет.

Если вы предпочитаете работать с меню, то для смены цвета шрифта воспользуйтесь пунктом **Color** меню **Text**. К сожалению, в этом случае вы не сможете обратиться к селектору цветов — Dreamweaver выдаст вам стандартное диалоговое окно выбора цвета Windows.

Примечание

Задание шрифта и цвета текста достигается с помощью тега <FONT>...</FONT>. Этот тег должен содержать один из атрибутов: COLOR, FACE или SIZE, задающих, соответственно, цвет, наименование шрифта и его размер. Как вы уже поняли, это тег физического форматирования.

Совет

Если вы посмотрите на секцию тегов, то увидите, что теги <FONT> вложены в теги <P> и <LI>, т. е. являются дочерними тегами самого последнего уровня. Это из особенностей HTML: некоторые теги могут быть вложены друг в друга только в определенном порядке.

Что теперь делать? Давайте посмотрим на наш текст и подумаем. Скажем, не слишком ли выделяются слова "Иван Иванович Иванов", набранные огромным жирным шрифтом? Не нужно ли вернуть их к обычному шрифту? Щелкните правой кнопкой мыши где-нибудь на этих словах и выберите в контекстном меню пункт **Remove Tag *<font>***. Dreamweaver тотчас удалит тег, задающий размер шрифта, но оставит его содержимое. Мы уже описывали этот пункт меню в *главе 2*, так что он должен быть вам знаком.

Вставка специальных символов

Чего нам еще не хватает для полного счастья?

Давайте отделим первый и второй абзацы от остального текста. Конечно, можно вставить пустую строку, но мы поступим лучше. И HTML (вкупе с Dreamweaver) нам в этом поможет.

Поставим текстовый курсор в начало третьего абзаца и нажмем кнопку **Horizontal Rule** (рис. 3.23), расположенную на вкладке **Common** панели объектов. Как показано на рис. 3.24, между вторым и третьим абзацами появится разделяющая их *горизонтальная линия*.

Вы также можете использовать пункт **Horizontal Rule** меню **Insert**.

Рис. 3.23. Кнопка **Horizontal Rule** панели объектов

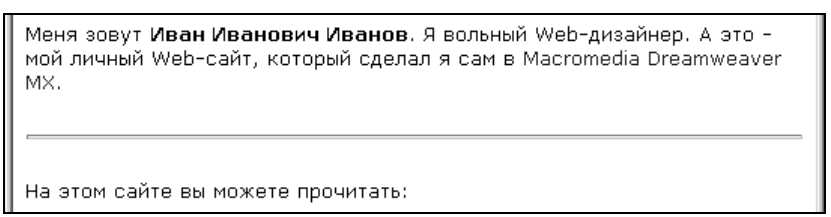


Рис. 3.24. Горизонтальная линия

Щелкните мышью по горизонтальной линии — она будет выделена целиком. После этого обратите внимание на редактор свойств. Вместо свойств текста и шрифта там будут отображены свойства горизонтальной линии, которые вы, конечно же, можете изменить (рис. 3.25).

Рис. 3.25. Вид редактора свойств при выделенной горизонтальной линии

Давайте подробно опишем все элементы управления редактора свойств, доступные в этом случае:

- поле ввода **W** позволяет задать ширину горизонтальной линии в пикселах или процентах от всей ширины родительского элемента, в данном случае страницы. По умолчанию горизонтальная линия занимает всю ширину страницы. Единица измерения задается в раскрывающемся списке, расположенном справа от поля ввода; доступны пункты **pixels** (пикселы; этот вариант указан по умолчанию) и **%**;
- поле ввода **H** позволяет задать толщину горизонтальной линии в пикселах. Значение по умолчанию — 2;
- раскрывающийся список **Align** задает выравнивание линии. Эти установки имеют смысл только в том случае, если ширина линии меньше ширины родительского элемента. Доступны значения: **Default** (выравнивание по умолчанию; такое же, как у родительского элемента), **Left** (по левому краю), **Center** (по центру) и **Right** (по правому краю);
- флажок **Shading** позволяет включить или отключить "трехмерность" линии. По умолчанию он включен.

Давайте зададим ширину линии, равную 50%. Для этого выберите в раскрывающемся списке единиц измерения проценты (пункт **%**), введите в поле ввода **W** число 50 и нажмите клавишу <Enter>. Результат показан на рис. 3.26.

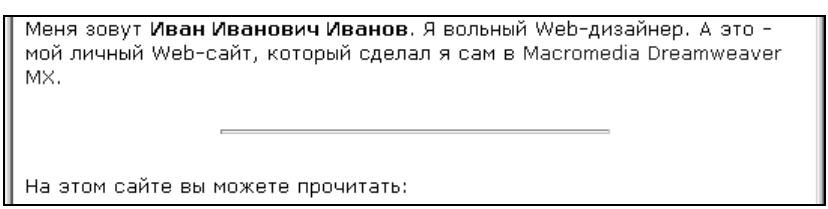


Рис. 3.26. Новый вид горизонтальной линии

Примечание

Горизонтальная линия вставляется в текст с помощью одинарного тега <HR>. Этот тег может содержать атрибуты ALIGN, COLOR, NOSHADE, SIZE и WIDTH, задающие, соответственно, выравнивание, цвет, "трехмерность", толщину и ширину. Заметьте, что, хотя тег <HR> и поддерживает атрибут COLOR, Dreamweaver не позволит вам задать цвет горизонтальной линии в редакторе свойств или как-либо еще. Хотя вы всегда можете отредактировать HTML-код вручную.

Внимание!

Среди атрибутов тега <HR> особняком стоит NOSHADE. Этот атрибут не имеет значения (так называемый *атрибут без значения*), а просто либо присутствует, либо не присутствует в теге. В первом случае он отменяет "трехмерность" линии, а во втором — разрешает.

Теперь посмотрим на последнюю строку нашего текста, содержащую сведения об авторских правах. Общепринятым стандартом стало использование значка "©" для обозначения авторских прав. У нас же — длинный невразумительный текст. Давайте заменим его коротким и приметным значком.

Сначала выделим слова "Авторские права принадлежат мне," (не забыв и запятую), которые мы заменим значком "©", и удалим. После этого переключимся на страницу **Characters** панели объектов. Нас интересует кнопка **Copyright** (рис. 3.27).

Рис. 3.27. Кнопка **Copyright** панели объектов

Поставьте на нужное место текстовый курсор и щелкните эту кнопку. Dreamweaver выведет небольшое предупреждение (рис. 3.28), говорящее о том, что данный символ может не отобразиться корректно в случае использования некоторых кодировок. Закройте его, нажав кнопку **OK**. Если вы больше не хотите, чтобы оно выводилось на экран в дальнейшем, перед закрытием включите флажок **Don't show me again**.

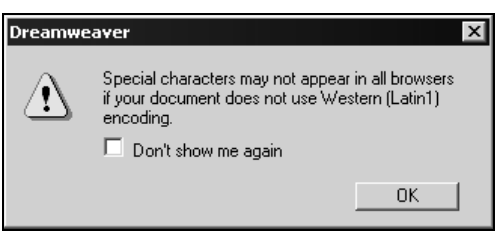


Рис. 3.28. Предупреждение о возможном некорректном отображении специального символа

На рис. 3.29 показан вставленный символ "©". Не забудьте также исправить фамилию нашего персонажа.

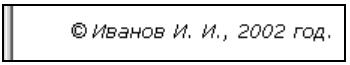


Рис. 3.29. Символ "©"

Здесь мы столкнулись с так называемыми *специальными символами* HTML. Эти символы из соображений все той же совместимости не могут быть просто так вставлены в код HTML, а заменяются либо числовыми кодами, либо мнемоническими обозначениями. В частности, символ "©" обозначается в HTML-коде как `©`.

Для вставки знака "©" вы также можете воспользоваться пунктом **Copyright** подменю **Special Characters** меню **Insert**.

Остается добавить в текст нашей страницы грозную фразу, предупреждающую всех желающих процитировать что-либо с настоящего сайта о том, что перед этим нужно справиться у владельца авторских прав на сайт. Допишем эту фразу в конце последней строки...

Стоп! Необходимо не просто дописать такую фразу, а как-то отделить ее от собственно сведений об авторских правах. Но как? Можно вставить горизонтальную линию, но тогда последняя строка растянется по вертикали, что будет выглядеть громоздко и некрасиво. Можно нажать клавишу <Enter> и сделать второй абзац, но получается все равно слишком громоздко (видели расстояние между абзацами?). Хотелось бы дописать эту фразу второй строкой, но в том же самом абзаце.

И на этот случай HTML (и Dreamweaver) предлагает решение. Речь идет о так называемом *разрыве строк* — особом теге, заставляющем все следующие за ним символы переноситься на новую строку. При этом он не создает нового абзаца, что нам и надо.

Поставьте курсор в конец строки со сведениями об авторском праве. После этого нажмите кнопку **Line Break** (рис. 3.30) на вкладке **Characters** панели объектов. Курсор тотчас перескочит на новую строку. Допишите нужную фразу. У вас должно получиться то, что изображено на рис. 3.31.

Рис. 3.30. Кнопка **Line Break** панели объектов

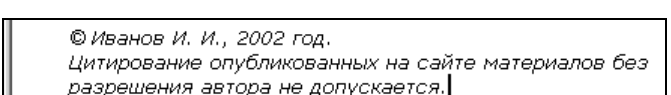


Рис. 3.31. Использование разрыва строк

Вы также можете для вставки разрыва строк использовать пункт **Line Break** подменю **Special Characters** меню **Insert** или — что проще всего — комбинацию клавиш <Shift>+<Enter>.

И последний штрих. Во втором абзаце нашего текста есть такая фраза: "А это — мой личный Web-сайт". Здесь используется тире. А по правилам полиграфии тире нельзя переносить на другую строку. Web-обозреватель об этом не знает и при недостатке свободного места все же перенесет тире, в результате чего текст будет выглядеть очень неаккуратно.

Чтобы этого не было, между словом "это" и тире нужно вставить *неразрывный пробел*. Что бы ни случилось, Web-обозреватель никогда не будет переносить строку по этому пробелу.

Прежде всего, уберите пробел между словом "это" и тире и поставьте там текстовый курсор. После этого нажмите на вкладке **Characters** панели объектов кнопку **Non-Breaking Space** (рис. 3.32). Вот и все.

Рис. 3.32. Кнопка **Non-Breaking Space** панели объектов

Для вставки неразрывного пробела вы также можете использовать пункт **Non-Breaking Space** подменю **Special Characters** меню **Insert** или сочетание клавиш <Ctrl>+<Shift>+<Пробел>.

Примечание

Неразрывный пробел — это специальный символ, обозначаемый кодом ` `.

Взгляните снова на вкладку **Characters** панели объектов. Мы описали не все специальные символы, которые можно вставить в документ с ее помощью. В табл. 3.4 описаны все остальные символы.

Таблица 3.4. *Специальные символы, доступные на вкладке* ***Characters*** *панели объектов*

Символ	Описание	Пункт подменю Special Characters меню Insert
"	Левая кавычка	**Left Quote**
"	Правая кавычка	**Right Quote**
—	Длинное тире	**Em-Dash**
£	Фунт стерлингов	**Pound**
¥	Иена	**Yen**

Таблица 3.4 *(окончание)*

Символ	Описание	Пункт подменю Special Characters меню Insert
€	Евро	**Euro**
®	Зарегистрированная торговая марка	**Registered**
™	Торговая марка	**Trademark**

С помощью кнопки **Other Characters** (рис. 3.33) можно вставить в документ любой доступный в HTML специальный символ. При нажатии на нее на экране появится диалоговое окно **Insert Other Character**, показанное на рис. 3.34. Вам необходимо будет щелкнуть по кнопке, соответствующей нужному символу, и нажать кнопку **OK** для его вставки или **Cancel** — для отказа от этого.

Рис. 3.33. Кнопка **Other Characters** панели объектов

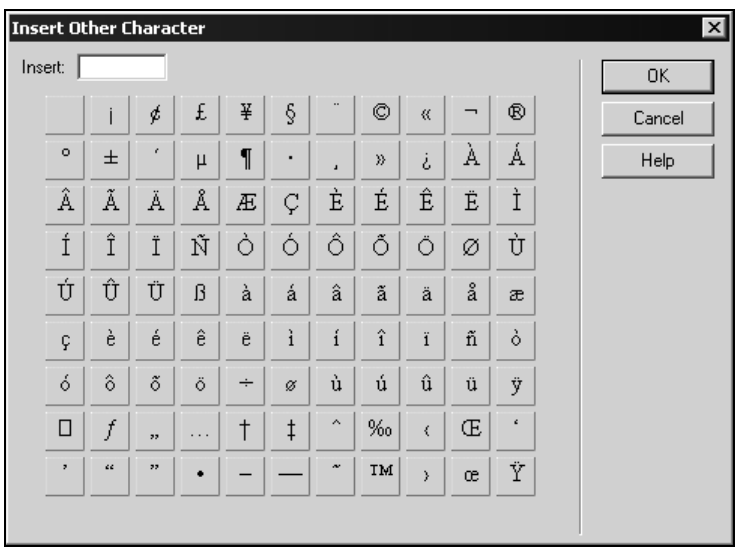


Рис. 3.34. Диалоговое окно **Insert Other Character**

Для вызова диалогового окна вставки символа вы также можете использовать пункт **Other** подменю **Special Characters** меню **Insert**.

Использование HTML-стилей

Очень и очень многие мощные текстовые редакторы предлагают пользователю так называемые *стили*, т. е. описания формата фрагментов текста, имеющие имя и сохраненные отдельно от редактируемого документа. Выделив фрагмент текста и выбрав стиль, пользователь может мгновенно применить к этому фрагменту все стилевые настройки: шрифт, выравнивание, цвет текста и т. п. Если же пользователю потребуется быстро изменить форматирование текста, он может просто отредактировать стиль — и все его настройки будут мгновенно применены к тем фрагментам, к которым ранее был присвоен этот стиль.

К сожалению, HTML не поддерживает стили. Dreamweaver выходит из этого положения, задавая свои стили, называемые *HTML-стилями*. Со стилями, скажем, Microsoft Word они имеют мало общего, в частности, фрагмент текста не может хранить имя стиля, который был к нему применен. Вдобавок, с помощью HTML-стилей можно задать только шрифтовые настройки, цвет текста и выравнивание абзаца. Однако HTML-стили во многих случаях помогут быстро отформатировать текст.

Поищите в доке панель под названием **HTML Styles** (рис. 3.35). Если ее нет, выберите пункт **HTML Styles** меню **Window** или нажмите комбинацию клавиш <Ctrl>+<F11>.

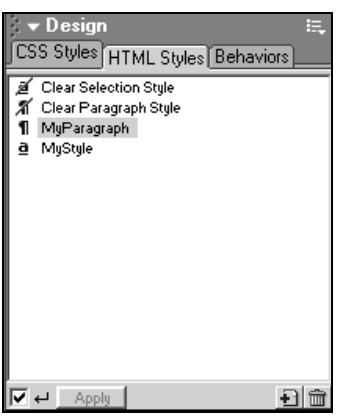


Рис. 3.35. Панель **HTML Styles**

Большую часть этой панели занимает собственно список HTML-стилей. Кроме пунктов, задающих определенные вами HTML-стили, там есть еще два: **Clear Selection Style** и **Clear Paragraph Style**. Они выполняют одинаковую задачу — удаляют всякое HTML-форматирование фрагмента текста. Но почему их два? И чем они различаются?

Прежде чем дать ответ на этот вопрос, еще немного поговорим о HTML-стилях.

Как и в обычных текстовых редакторах, HTML-стиль Dreamweaver может применяться либо к выделенному фрагменту текста, либо к абзацу целиком; в первом случае он называется *стилем символов*, а во втором — *стилем абзаца*. При создании HTML-стиля вы задаете, к чему он будет применен. Так вот, пункт **Clear Selection Style** удаляет HTML-форматирование выделенного текста, а **Clear Paragraph Style** — всего абзаца.

Вы можете испытать эти пункты списка в действии. Для этого выделите, скажем, имя нашего героя и выберите пункт **Clear Selection Style**. Вы увидите, что жирное выделение имени сразу же пропадет. Теперь поставьте текстовый курсор на какой-либо абзац, например на последнюю строку со сведениями об авторских правах, и выберите пункт **Clear Paragraph Style**. Произойдет то же самое, за тем исключением, что останется отступ. Сделайте дважды операцию отмены (либо выберите пункт **Undo** меню **Edit**, либо нажмите комбинацию клавиш <Ctrl>+<Z>), чтобы вернуть все, как было.

Теперь взглянем снова на панель **HTML Styles**. В ее левом нижнем углу находится флажок автоматического применения стилей **Auto Apply**. Когда он включен (а он включен по умолчанию), выбираемые в списке HTML-стили применяются сразу же после выбора. Если вы его отключите, то для того, чтобы применить стиль, вам нужно после его выбора в списке щелкнуть кнопку **Apply**, находящуюся правее этого флажка, выбрать пункт **Apply** контекстного меню списка стилей или дополнительного меню панели или просто нажать клавишу <Enter>. Это может быть полезно, например, когда вы часто редактируете HTML-стили и не хотите, чтобы выбранный стиль применялся сразу же.

Так как же создавать HTML-стили? Очень просто.

Здесь есть две возможности: вы можете создать стиль "с нуля" и скопировать его с уже отформатированного и выделенного фрагмента текста. Второй способ намного проще первого, поэтому мы рассмотрим его в первую очередь.

Давайте создадим HTML-стиль под названием `Имя`, представляющий имя автора сайта. Для этого выделим в окне документа текст имени и нажмем в окне HTML-стилей кнопку **New Style** (рис. 3.36), находящуюся в правом нижнем углу панели. Вы также можете либо выбрать пункт **New** в дополнительном меню панели, либо выбрать этот же пункт в контекстном меню списка стилей панели. Так или иначе на экране появится диалоговое окно **Define HTML Style**, показанное на рис. 3.37.

Рис. 3.36. Кнопка **New Style** панели **HTML Styles**

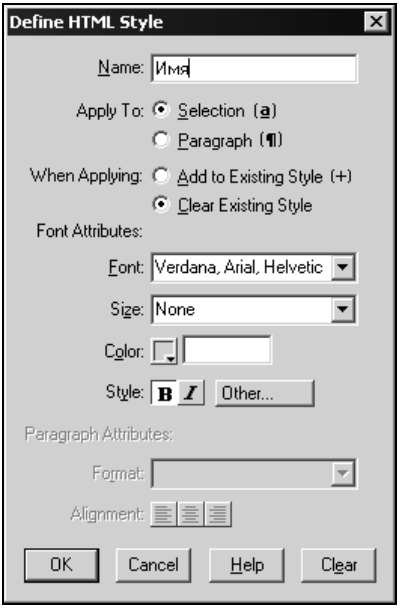


Рис. 3.37. Диалоговое окно **Define HTML Style**

В поле ввода **Name** вводится имя стиля. Введем сюда слово Имя.

С помощью группы переключателей **Apply To** задается тип HTML-стиля: **Selection** задает стиль символов, а **Paragraph** — стиль абзаца. Сейчас включен переключатель **Selection**, оставим его включенным.

Группа переключателей **When Applying** позволяет определить, что произойдет, если этот стиль будет применен к фрагменту, уже отформатированному средствами HTML: **Add to Existing Style** добавляет новое форматирование к существующему, **Clear Existing Style** удаляет старое форматирование перед добавлением нового. Оставим переключатель **Clear Existing Style** включенным.

Группа элементов управления **Font Attributes** задает параметры шрифта. Раскрывающийся список **Font** задает наименование шрифта, **Size** — размер, селектор цвета **Color** позволяет выбрать цвет текста, а кнопки-выключатели **B** и ***I*** делают шрифт жирным и курсивным соответственно. Также доступна кнопка **Other**, при нажатии которой появляется меню, из которого вы можете выбрать дополнительные виды HTML-форматирования текста (в основном, это теги логического форматирования). Здесь мы также ничего не трогаем.

Закончив, нажимаем кнопку **OK**, чтобы сохранить вновь созданный стиль. Кнопка **Cancel** позволит отказаться от этого, а кнопка **Clear** сбрасывает все настройки стиля в состояние по умолчанию. Так или иначе, мы создали наш первый HTML-стиль. После этого в списке стилей появится новая

строка "Имя". Левее имени стиля мы увидим букву a, обозначающую стиль символов.

Создадим теперь еще один стиль, но на этот раз — стиль абзаца, а именно абзаца со сведениями об авторских правах. И на этот раз, "с нуля". Для этого уберем всякое выделение с текста в окне документа и снова нажмем кнопку **New Style** панели стилей. В появившемся диалоговом окне **Define HTML Style** в поле **Name** введем "Права", включим переключатель **Paragraph**, включим кнопку-переключатель ***I*** и отключим **B**, а в раскрывающемся списке **Size** выберем пункт **−1**.

Как видим, в окне **Define HTML Style** стала доступна группа элементов управления **Paragraph Attributes**, задающая форматирование всего абзаца. Раскрывающийся список **Format** задает формат абзаца (**Paragraph**, **Heading 1** и т. д.), а три кнопки-переключателя **Alignment** — выравнивание. Оба эти элементы управления полностью подобны своим "коллегам" из редактора свойств. Ничего здесь трогать пока не будем.

Закончив, нажмем кнопку **OK**, и в списке стилей появится новое имя, помеченное знаком абзаца. Теперь выберем последний абзац нашего текста и щелкнем новоиспеченный стиль. Формат сведений об авторских правах тотчас изменится: размер шрифта станет меньше.

Если вам нужно изменить уже созданный стиль, щелкните по нему правой кнопкой мыши и в появившемся контекстном меню выберите пункт **Edit**. Если флажок **Auto Apply** отключен, вы также можете выбрать необходимый стиль и выбрать пункт **Edit** в дополнительном меню панели или просто дважды щелкнуть по нужному стилю. После этого на экране появится уже знакомое вам диалоговое окно **Define HTML Style**, в котором вы сможете изменить все, что хотите.

Для удаления ненужного стиля также щелкните по нему правой кнопкой мыши и выберите в контекстном меню пункт **Delete**. Dreamweaver предупредит вас в особом окне, что стили удаляются безвозвратно; нажмите кнопку **OK**, если вы преисполнены решимости избавиться от этого стиля, и кнопку **Cancel**, если передумали его удалять. Если флажок **Auto Apply** отключен, вы также можете выбрать нужный (точнее, ненужный) стиль и выбрать пункт **Delete** в дополнительном меню панели или щелкнуть кнопку **Delete Style** (рис. 3.38).

Рис. 3.38. Кнопка **Delete Style** панели **HTML Styles**

Пункт **Duplicate** контекстного меню списка стилей и дополнительного меню панели позволит вам создать новый стиль третьим возможным способом: скопировав и слегка изменив существующий стиль. Если вы укажете какой-либо стиль в списке и выберите пункт **Duplicate**, на экране появится диало-

говое окно **Define HTML Style**, в котором будут подставлены параметры выбранного стиля. Задайте имя для нового стиля, измените нужные параметры и не забудьте нажать кнопку **OK**.

Работа с гиперссылками

В *главе 1* мы уже говорили о гиперссылках. Пора научиться создавать их в среде Dreamweaver.

Для того чтобы описать личность нашего незабвенного Ивана Ивановича Иванова, нам нужно пять Web-страниц: главная, список увлечений, список проектов, более подробные сведения об авторе и любимые ссылки. В данный момент у нас готова только главная страница. И прежде, чем мы приступим к созданию других страниц, узнаем, как же создаются гиперссылки.

Создание обычных гиперссылок

А создаются они очень просто. И в этом — большая заслуга разработчиков Dreamweaver, сделавших продукт, реализующий просто даже весьма сложные задачи.

Но хватит петь дифирамбы Dreamweaver. Давайте попробуем его в настоящем деле! Вставим в текст нашей пока что единственной Web-страницы две гиперссылки, указывающие "вовне" нашего сайта: на "домашний" сайт фирмы Macromedia — создателя Dreamweaver и на почтовый адрес гипотетического Ивана Ивановича. Щелкнув по первой из них, пользователь сможет попасть на сайт **http://www.macromedia.com**, а щелкнув на второй — написать письмо Ивану Ивановичу. Второй ссылкой мы займемся чуть позже, а пока...

Выделим название фирмы Macromedia и обратим внимание на редактор свойств. Мы сразу заметим большое поле ввода интернет-адреса для гиперссылки (рис. 3.39).

Рис. 3.39. Поле ввода интернет-адреса в редакторе свойств

Все, что нам нужно сделать, — это ввести в него интернет-адрес сайта Macromedia и нажать клавишу <Enter>. Результат появится сразу (рис. 3.40). Теперь при просмотре в Web-обозревателе пользователь может просто щелкнуть мышью по слову "Macromedia", чтобы перейти на "домашний" сайт этой фирмы.

Вы уже заметили, что после вставки гиперссылки слово "Macromedia" изменило свой цвет на синий и стало подчеркнутым, хотя мы явно не задавали

для него никакого HTML-форматирования. Дело в том, что, согласно стандарту HTML, текст-содержимое гиперссылок форматируется особым образом, чтобы его всегда можно было отличить от остального текста. И по умолчанию все ссылки, еще не посещенные пользователем, выделяются синим цветом.

Рис. 3.40. Гиперссылка, указывающая на сайт Macromedia

Для создания гиперссылки из выделенного текста вы также можете воспользоваться пунктом **Make Link** меню **Modify** или контекстного меню или нажать комбинацию клавиш <Ctrl>+<L>. После этого на экране появится диалоговое окно **Select File**, показанное на рис. 3.41. Нужный адрес вводится в поле ввода **URL** этого окна. Остальные элементы управления лучше пока не трогать — мы рассмотрим их позже, когда будем связывать воедино все Web-страницы нашего сайта. Не забудьте нажать кнопку **OK**, чтобы Dreamweaver создал эту гиперссылку.

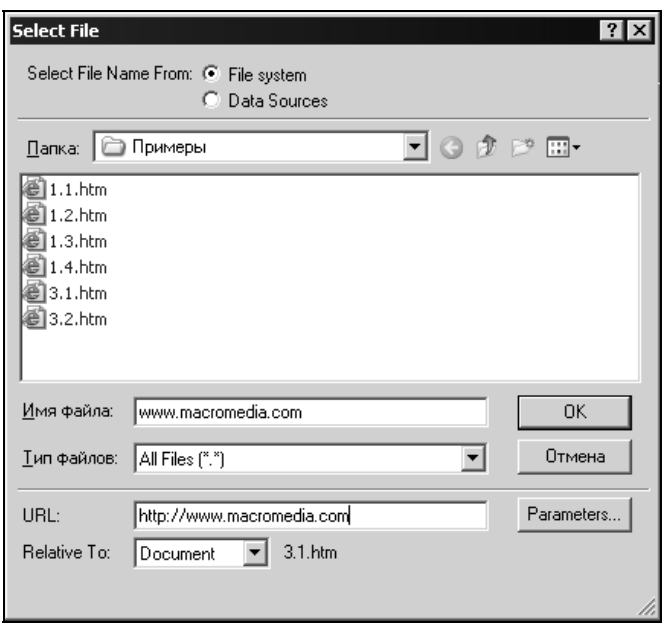


Рис. 3.41. Диалоговое окно **Select File**

Просматривая рис. 3.39, вы, вероятно, обратили внимание, что справа от поля ввода интернет-адреса находятся две небольшие кнопки. Кнопка в ви-

де мишени, находящаяся левее, нам пока бесполезна — мы рассмотрим ее, когда будем говорить об управлении сайтом, в *главе 6*. Кнопка в виде папки, находящаяся правее, позволит вам открыть то же самое диалоговое окно **Select File**.

Впоследствии вы можете изменить интернет-адрес вашей гиперссылки. Для этого поставьте текстовый курсор на текст гиперссылки, измените адрес и нажмите кнопку <Enter>. Вы также можете воспользоваться пунктом **Change Link** меню **Modify** или контекстного меню или нажмите комбинацию клавиш <Ctrl>+<L>. После этого на экране появится все то же незаменимое диалоговое окно **Select File**, в котором вы сможете изменить интернет-адрес.

Для удаления гиперссылки и превращения ее содержимого в обычный текст воспользуйтесь пунктом **Remove Link** меню **Modify** или контекстного меню или нажмите комбинацию клавиш <Ctrl>+<Shift>+<L>.

Отлично! Мы имеем гиперссылку для перехода на сайт Macromedia. Чего еще желать?

Много чего. Например, мы можем сделать так, что при переходе на сайт Macromedia пользователь будет иметь возможность просматривать также и сайт Иванова. Как это сделать? Очень просто: мы откроем сайт Macromedia в новом окне Web-обозревателя.

Поставьте текстовый курсор где-нибудь в тексте гиперссылки и снова посмотрите на редактор свойств. На рис. 3.42 показан раскрывающийся список задания *цели гиперссылки*.

Рис. 3.42. Раскрывающийся список задания цели гиперссылки, расположенный в редакторе свойств

Этот список содержит четыре пункта, но сейчас полезными для нас будут только два: **_blank** и **_self**. Первый из них позволит вывести страницу, на которую указывает гиперссылка, в новом окне Web-обозревателя, а второй — в том же окне. (Вместо того чтобы выбирать второй пункт списка, вы можете просто оставить его содержимое пустым.) В данном случае вам нужно выбрать пункт **_blank**.

Для задания цели гиперссылки вы также можете воспользоваться подменю **Link Target** меню **Modify** или подменю **Target Frame** контекстного меню. Для этого выберите нужный пункт-переключатель. Пункт **Default Frame** задает цель по умолчанию.

Примечание

Гиперссылка создается с помощью парного тега <A>, внутри которого помещается текст. Этот тег содержит обязательный атрибут HREF, задающий интернет-адрес назначения. Кроме того, данный тег может включать атрибут TARGET, задающий цель гиперссылки.

Вот мы и сделали первую нашу гиперссылку. Теперь примемся за вторую.

Создание почтовых гиперссылок

Вторая гиперссылка будет слегка отличаться от первой. Хотя бы потому, что она указывает не на Web-страницу, а на почтовый адрес, и при щелчке на ней пользователь попадет в окно открывшейся программы почтового клиента, установленного в системе по умолчанию. И еще: гиперссылку, указывающую на почтовый адрес, нужно записать особым образом.

Давайте вспомним интернет-адрес сайта Macromedia:

http://www.macromedia.com

и придумаем почтовый адрес, который может быть у нашего гипотетического Web-дизайнера Ивана Иванова:

ivanov@somemail.ru

Немного теории. Почтовый адрес состоит из имени почтового ящика и имени *сервера почты*, разделенных символом "@" ("коммерческое эт", также известен как "собака", "ухо" или "улитка"). Вы можете провести аналогии между номером абонементного ящика и адресом почтового отделения, где этот ящик находится. Значок "коммерческое эт" — первый признак, что перед нами почтовый адрес.

Согласно стандарту HTML, этот почтовый адрес должен быть записан так:

mailto:ivanov@somemail.ru

причем между двоеточием после "mailto" и собственно адресом не должно быть пробела.

В последнем абзаце, где говорится об авторских правах, есть имя автора. Из него-то мы и сделаем гиперссылку, указывающую на почтовый адрес. Для этого выделим имя автора и введем почтовый адрес в приведенном выше формате в поле адреса редактора свойств. После нажатия клавиши <Enter> мы получим результат, показанный на рис. 3.43.

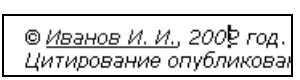


Рис. 3.43. Гиперссылка, указывающая на почтовый адрес

Как видите, эта гиперссылка внешне ничем не отличается от сделанной нами ранее. Фактически, гиперссылки, указывающие на Web-страницы и почтовые адреса, ничем не отличаются друг от друга. И чтобы не обескуражить пользователя, вам придется самим дать понять ему, что он получит в результате. Вы можете написать где-то в тексте, что это почтовый адрес, или сделать что-то еще.

К несчастью, из-за ошибки в Dreamweaver невозможно отредактировать почтовый адрес, помещенный в гиперссылку. Если вы выберете пункт **Change Link** в меню **Modify** (или нажмете комбинацию клавиш <Ctrl>+<L>), то вместо полного адреса в диалоговом окне **Select File** появится только адрес почтового сервера. Так что вам для этих целей лучше воспользоваться редактором свойств.

Есть еще один способ создать гиперссылку, указывающую на почтовый адрес, более элегантный. Для этого необходимо либо переключиться на страницу **Common** панели объектов и щелкнуть кнопку **Email Link** (рис. 3.44), либо выбрать пункт **Email Link** в меню **Insert**. В результате этих действий на экране появится небольшое диалоговое окно **Email Link**, показанное на рис. 3.45.

Рис. 3.44. Кнопка **Email Link** панели объектов

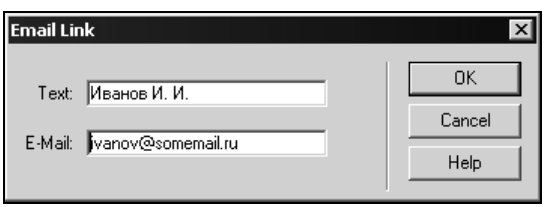


Рис. 3.45. Диалоговое окно **Email Link**

В поле **Text** самим Dreamweaver подставляется текст, который будет являться содержимым гиперссылки — имя, выделенное нами в окне документа. К сожалению, при этом Dreamweaver портит русский текст, превращая его в набор невразумительных черточек и квадратиков, так что вам, скорее всего, придется ввести его заново. В поле **E-Mail** вам нужно будет ввести почтовый адрес, причем ввести его нужно без "mailto:". После этого остается нажать кнопку **OK** для создания гиперссылки или **Cancel** — для отказа от этого.

Другие гиперссылки

А теперь рассмотрим создание в среде Dreamweaver гиперссылок, указывающих на другие интернет-ресурсы. Вам, должно быть, известно, что, кроме WWW и электронной почты, Интернет предлагает и другие *сервисы*:

FTP, группы новостей и др. И на каждый из этих ресурсов вы также можете сделать гиперссылку, воспользовавшись Dreamweaver, или по-старинке, в Блокноте.

Мы не будем рассматривать все сервисы, предлагаемые Интернетом. Ограничимся только теми, с которыми вы наверняка столкнетесь.

Создание гиперссылок на FTP-серверы

FTP (File Transfer Protocol — протокол передачи файлов) — один из старейших сервисов Интернета. Он служит для распространения файлов в Сети. Файлы помещаются на диски *FTP-сервера*, а пользователи с помощью специальных программ *FTP-клиентов* подключаются к этому серверу и забирают файлы. Примеры FTP-клиентов: популярнейший CuteFTP, WS_FTP и др. Файлы от сервера клиенту передаются по протоколу FTP, давшему название самому этому сервису.

Для подключения к FTP-серверу пользователю нужно указать *имя* и *пароль* пользователя. В подавляющем большинстве случаев пользователь использует имя "anonymous" (анонимный доступ) и пароль либо тоже "anonymous", либо свой почтовый адрес, либо *пустой* пароль (пустая строка ""). Вследствие этого интернет-адрес файла, находящегося на сервере FTP, будет иметь вид:

ftp://username:userpassword@ftp.someserver.ru/pub/utility/util23.zip

Вместо слов "username" и "userpassword" подставляются реальные имя и пароль пользователя. Имя и пароль разделяются двоеточием, а пароль и собственно адрес сервера — уже знакомым вам символом "коммерческое эт". Все это вам нужно набрать опять же в поле ввода интернет-адреса, расположенном в редакторе свойств.

Новости (USENET)

Новости (часто их называют USENET) — довольно молодой сервис Интернета, чем-то похожий на электронную почту. Отличие заключается в том, что все посланные вами в определенную группу новостей сообщения будут доступны всем ее подписчикам, которые смогут присылать на них свои ответы, опять же, доступные всем. Сами группы новостей со всеми их сообщениями организуются на *сервере новостей*, а передаются они по протоколу *NNTP* (Network News Transfer Protocol — протокол передачи сетевых новостей). Для чтения новостей служат *клиенты новостей*, например, Agent; кроме того, новости поддерживаются некоторыми почтовыми клиентами, например, Microsoft Outlook Express.

Каждое сообщение группы новостей имеет свой уникальный номер, по которому к нему можно обратиться. Например:

nntp://news.someserver.ru/alt.somegroup/16718

Здесь после адреса самого сервера новостей через обратную косую черту указывается имя группы новостей, а после нее опять же через обратную косую черту — номер сообщения.

Использование "якорей"

Теперь самое время узнать еще об одном виде гиперссылок, весьма специфичном. Это так называемые *"якоря"* (по-английски — anchors). В отличие от других гиперссылок, они не указывают на другую страницу (файл, адрес электронной почты), а помечают некоторый участок текста текущей страницы, чтобы другая гиперссылка могла на него сослаться.

Зачем это нужно? Ну, например, если ваша Web-страница содержит большой текст, разбитый на главы (кто знает, может, вы написали роман и выложили его в Сети), вы можете поместить в начале нее оглавление. При щелчке на пункте оглавления пользователь тотчас перескакивает на соответствующую главу вашего труда. Очень удобно!

"Якорь" не отображается Web-обозревателем. Однако Dreamweaver, чтобы помочь Web-дизайнеру, показывает на странице специальный значок.

Для того чтобы научиться работать с "якорями", создадим небольшую учебную страничку. Ничего особенного на ней нет: просто небольшой фрагмент текста из этой книги, разумеется, отформатированный. После этого напишем в начале страницы небольшое оглавление. И, наконец, сохраним ее в файле 3.3.htm. Получившаяся страничка показана на рис. 3.46.

Теперь вставим на эту страницу первый "якорь", для чего поместим текстовый курсор в начало первого абзаца, после заголовка, и переключимся на страницу **Common** панели объектов. Кнопка вставки "якоря" **Named Anchor** показана на рис. 3.47. Нажмите ее. Вы также можете воспользоваться пунктом **Named Anchor** меню **Insert** или просто нажать комбинацию клавиш <Ctrl>+<Alt>+<A>.

В результате этих действий на экране появится диалоговое окно **Named Anchor**, показанное на рис. 3.48. В поле ввода **Anchor Name** введите имя вставляемого "якоря"; в нашем случае это будет `par1`. Имена "якорей" должны состоять из латинских букв и цифр, причем первым символом должна быть буква. После этого нажмите кнопку **OK**, чтобы вставить "якорь", или **Cancel** — для отказа от этого.

Dreamweaver отобразит значок "якоря" (рис. 3.49) в позиции, где тот был вставлен. Это делается только для удобства Web-дизайнера, Web-обозреватель, как вы помните, "якоря" на страницах вообще не отображает.

Если вы выделите значок "якоря", редактор свойств изменит свой вид. На нем появится единственное поле ввода **Name**, в котором вы сможете изменить имя "якоря".

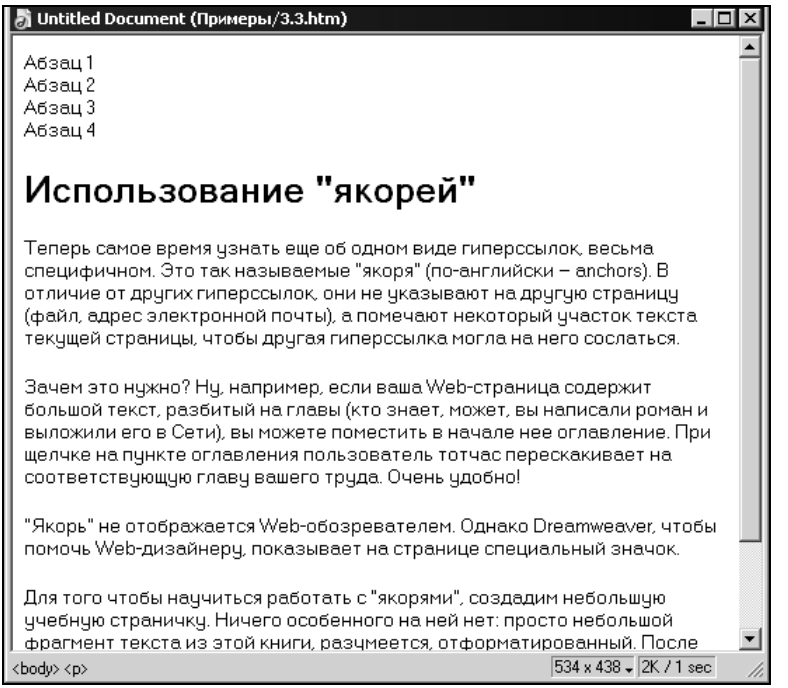


Рис. 3.46. Учебная Web-страница, где будут использованы "якоря"

Рис. 3.47. Кнопка **Named Anchor** панели объектов

Рис. 3.48. Диалоговое окно **Named Anchor**

Рис. 3.49. Значок "якоря" в окне документа Dreamweaver

Теперь создадим гиперссылку, ссылающуюся на "якорь". Выделим строку "Абзац 1" в оглавлении. Адрес, на который она ссылается, будет выглядеть так: **#par1**.

Значок "решетки" (#) означает, что перед нами имя "якоря". Введите его в поле адреса редактора свойств и нажмите клавишу <Enter>. Полученная нами гиперссылка ничем не будет отличаться от обычной, указывающей на другую страницу.

Из другой Web-страницы на этот "якорь" можно сослаться, написав перед "решеткой" имя файла этой страницы: **3.3.htm#par1**.

Вот и все! Нам теперь только осталось разместить остальные три "якоря", помечающие второй, третий и четвертый абзацы, и три гиперссылки, ссылающиеся на них. Не будем описывать это — вы уже знаете, как это делается.

Закончив страницу, сохраните ее и вызовите для предварительного просмотра в Web-обозревателе. Теперь уменьшите его окно так, чтобы в нем не помещалось текста больше одного абзаца, и щелкните по гиперссылке, указывающей на третий абзац текста. Содержимое окна Web-обозревателя будет тотчас прокручено вверх так, что в нем появится третий абзац, т. е. вставленные нами "якоря" и гиперссылки, составляющие оглавление, работают.

Общие свойства Web-страницы

Ну вот. Теперь у нас есть практически готовая главная Web-страница нашего сайта. Осталось только сделать несколько завершающих штрихов — и работа над ней будет завершена.

Выше мы создали две гиперссылки. И выяснили, что Web-обозреватель форматирует их по-своему, отлично от основного текста. Это, собственно, правильно: гиперссылки должны сразу бросаться в глаза, иначе пользователь будет вынужден едва ли не в лупу просматривать весь ваш текст в поисках гиперссылок, ведущих на другие страницы. И в конце концов уйдет с вашего сайта, что вам совсем не нужно.

Может быть и так. Вы тщательно разработали дизайн Web-страницы, подобрали цветовую гамму, а гиперссылки с их фиксированными цветами выбиваются из общего ряда и просто режут глаз. Ведь Web-обозреватель не обращает ни малейшего внимания на цветовые настройки родительского элемента (в данном случае, текста), когда расцвечивает их. Что делать в этом случае?

Такая проблема решается очень просто. Нужно всего лишь изменить *общие свойства* вашей Web-страницы, т. е. настройки, затрагивающую саму страницу, а не ее элементы. Давайте выясним, как это сделать.

Общие свойства страницы задаются в диалоговом окне **Page Properties**. Вызвать его можно, выбрав пункт **Page Properties** в меню **Modify** или контекст-

ном меню или нажав комбинацию клавиш <Ctrl>+<J>. Это окно показано на рис. 3.50.

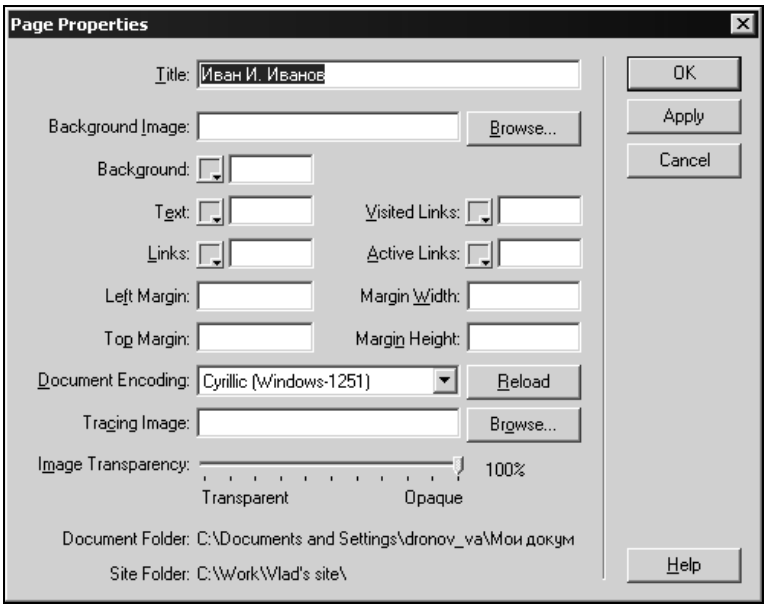


Рис. 3.50. Диалоговое окно **Page Properties**

В поле ввода **Title** вводится название Web-страницы. Вы можете пользоваться полем ввода, расположенным на инструментарии документа (см. рис. 3.3), либо вводить его в названном окне.

Селектор цвета **Background** позволяет вам выбрать цвет фона страницы. По умолчанию он белый, но вы можете выбрать какой угодно, лишь бы после этого текст на вашей странице нормально читался. Например, давайте для нашей страницы выберем светло-желтый цвет под кодом `#FFFFCC`.

Аналогично селектор цвета **Text** позволяет выбрать цвет текста (по умолчанию он черный). Опять же здесь действует правило: текст должен читаться нормально, без излишнего напряжения глаз. Для нашей страницы выберем темно-синий цвет `#000066`.

Теперь обратим внимание на три селектора **Links**, **Visited Links** и **Active Links**. Они позволяют выбрать цвет гиперссылок, соответственно, непосещенных (по умолчанию синий), посещенных (темно-красный) и активной гиперссылки (ярко-красный), по которой в данный момент щелкает пользователь. И еще раз повторим то же самое правило: пользователю должно быть удобно читать ваш текст. Но в случае гиперссылок добавляется еще одно правило: обычные гиперссылки должны быть хорошо заметны на фоне

текста, посещенные ссылки должны выглядеть чуть менее заметно (но все равно выделяться на фоне текста), а активная гиперссылка — максимально ярко. Исходя из этого правила, выберем, соответственно, ярко-синий (#0066FF), тускло-синий (#006699) и ярко-красный (#FF0000) цвета.

Мы можем нажать кнопку **OK**, чтобы сохранить сделанные установки. Но в данном случае лучше всего воспользоваться кнопкой **Apply**, которая применяет заданные вами свойства, не закрывая окна **Page Properties**. Отодвинем окно в сторону и посмотрим на нашу страничку. А что, получилось неплохо!

Поля ввода **Left Margin** и **Top Margin** задают расстояния между рамкой окна Web-обозревателя и содержимым Web-страницы, соответственно, слева и сверху. По умолчанию эти значения равны 10 и 15 пикселов. Вы можете задать эти значения равными 0, если хотите вообще убрать свободное пространство между рамкой окна и содержимым страницы, но мы не советуем вам это делать — страница будет выглядеть очень плохо.

Внимание!

Значения, введенные в поля ввода **Left Margin** и **Top Margin**, не воспринимаются Navigator.

Поля ввода **Margin Width** и **Margin Height** задают расстояние между рамкой окна Web-обозревателя и содержимым Web-страницы, соответственно, по горизонтали и по вертикали. Они поддерживаются и Internet Explorer, и Navigator. Поэтому лучше всего задавать именно их из соображений совместимости.

Раскрывающийся список **Document Encoding** задает кодировку текста вашей Web-страницы. (О кодировках и их выборе см. *главу 2*.) По умолчанию там выбрана кодировка, которую вы задали в настройках Dreamweaver, и менять ее без уважительных причин не стоит.

Закройте диалоговое окно задания свойств Web-страницы, нажав кнопку **OK** или **Cancel**. С окном **Page Properties** мы закончили.

Примечание

Все вышеописанные настройки задаются атрибутами тега <BODY>, перечисленными в табл. 3.5.

Таблица 3.5. *Атрибуты тега <BODY>*

Атрибут	Описание	Где задается
ALINK	Цвет активной гиперссылки	**Active Links**
BGCOLOR	Цвет фона страницы	**Background**

Таблица 3.5 *(окончание)*

Атрибут	Описание	Где задается
BOTTOMMARGIN	Расстояние между нижней рамкой окна и содержимым страницы. По умолчанию 0	—
LEFTMARGIN	Расстояние между левой рамкой окна и содержимым страницы	**Left Margin**
LINK	Цвет непосещенных гиперссылок	**Links**
MARGINHEIGHT	Расстояние между рамкой окна и содержимым страницы по вертикали	**Margin Height**
MARGINWIDTH	Расстояние между рамкой окна и содержимым страницы по горизонтали	**Margin Width**
RIGHTMARGIN	Расстояние между правой рамкой окна и содержимым страницы. По умолчанию 10	—
TEXT	Цвет обычного текста	**Text**
TOPMARGIN	Расстояние между верхней рамкой окна и содержимым страницы	**Top Margin**
VLINK	Цвет посещенных гиперссылок	**Visited Links**

Примечание

Обратите внимание, что значения атрибутов BOTTOMMARGIN и RIGHTMARGIN в среде Dreamweaver вы установить не можете. Для этого вам придется редактировать сам HTML-код.

Внимание!

Кодировка документа задается с помощью одинарного тега <META>, размещающегося в HTML-заголовке страницы. Подробнее о теге <META> мы поговорим в *главе 14*.

Дополнительные возможности Dreamweaver

Здесь мы рассмотрим некоторые дополнительные, но отнюдь не второстепенные возможности Dreamweaver, которые обязательно вам пригодятся.

Использование цветовых схем

Только что мы рассмотрели, как с помощью диалогового окна **Page Properties** можно быстро поменять всю расцветку Web-страницы. Но Dreamweaver

предлагает большее. Вместо того чтобы скрупулезно, методом проб и ошибок, подбирать цвета для страницы, вы можете воспользоваться одной из определенных в программе *цветовых схем* — комбинацией цветовых настроек для текста, фона и гиперссылок, подобранной профессиональными дизайнерами.

Цветовые схемы выбираются в диалоговом окне **Set Color Scheme Command**. Чтобы вызвать его, выберите пункт **Set Color Scheme** в меню **Command**. Это окно показано на рис. 3.51.

Рис. 3.51. Диалоговое окно **Set Color Scheme Command**

Процесс выбора цветовой схемы состоит из двух шагов. Сначала вы должны выбрать одну из категорий цветовых схем, называемых по цвету фона страницы: **Black**, **Blue**, **Brown**, ..., **Yellow**. Категория выбирается в списке **Background**, после чего в списке **Text and Links** появляются имена собственно категорий, из которых вы можете выбрать любую.

Оценить выбранную цветовую схему вы можете на панели предварительного просмотра, расположенной в центре окна **Set Color Scheme Command**. Но т. к. разработчики Dreamweaver уверяют, что все цветовые схемы разработаны профессиональными дизайнерами, они будут, по крайней мере, достойно выглядеть. Однако окончательный выбор схемы, наиболее подходящей вашему сайту, зависит только от вас.

Нажав кнопку **Apply**, вы можете применить выбранную вами цветовую схему без закрытия диалогового окна. Чтобы закрыть это окно, нажмите кнопку **OK** для сохранения выбранных установок или **Cancel** — для отказа от них.

Попробуйте поэкспериментировать с предлагаемыми Dreamweaver цветовыми схемами. Может быть, какая-то из них вам понравится.

Вставка и чтение комментариев

Вот и все. На этом этапе мы закончили. Наша Web-страничка уже полностью функциональна; осталось только написать остальные четыре страницы и связать их с главной.

Но память человеческая недолговечна... Вы можете замотаться с работой и забыть, что хотели доделать сайт. Неплохо было бы записать все, что вы хотели сделать, на бумажку. Но, имея перед собой компьютер, как-то несолидно оклеивать стены и набивать карманы бумажками-"незабудками". В конце концов, мы в компьютерную эпоху живем или нет?!

К счастью, стандарт HTML определяет так называемые *комментарии* — фрагменты текста, хранимые в коде Web-страницы, но не обрабатываемые и не отображаемые Web-обозревателем. Вы можете написать в комментарии все, что угодно, но обычно разработчики страниц пишут короткие примечания или напоминания для самих себя.

Прежде всего, решите, где вы вставите комментарий. Логично поместить его как можно ближе к тому фрагменту страницы, который вы хотите изменить в дальнейшем и к которому, собственно, и будет относиться комментарий. В нашем случае поместите текстовый курсор в конце третьего абзаца ("На этом сайте вы можете прочитать:").

Вставить комментарий проще всего из панели объектов. Для этого переключитесь на страницу **Common** и нажмите кнопку **Comment** (рис. 3.52). На экране появится диалоговое окно **Comment**, показанное на рис. 3.53.

Рис. 3.52. Кнопка **Comment** панели объектов

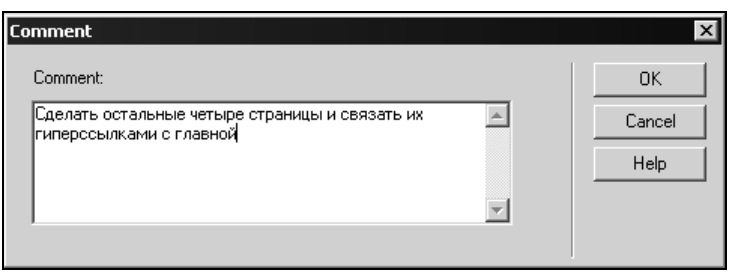


Рис. 3.53. Диалоговое окно **Comment**

Это окно устроено очень просто. Вы вводите текст комментария в область редактирования **Comment** и нажимаете кнопку **OK** для вставки комментария или **Cancel** — для отмены. После этого в конце третьего абзаца появится значок комментария, отображаемый в таких случаях Dreamweaver (рис. 3.54).

Повторим, что в программе Web-обозревателя комментарий не будет виден совсем.

Рис. 3.54. Значок комментария в окне документа Dreamweaver

Для вставки комментария вы также можете выбрать пункт **Comment** подменю **Text Objects** меню **Insert**.

Теперь щелкните мышью на значке вставленного нами комментария. Значок будет выделен, а окно редактора свойств изменится (рис. 3.55). В области редактирования, занимающей большую его часть, вы сможете просмотреть и изменить текст комментария.

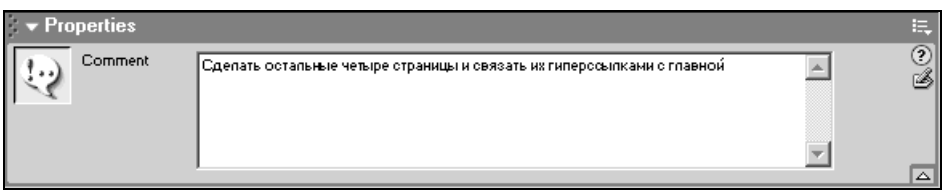


Рис. 3.55. Вид редактора свойств при выделенном комментарии

Чтобы впоследствии удалить ненужный комментарий, просто выделите мышью его значок и нажмите клавишу <Del>.

Примечание

Комментарии вставляются в HTML-код с помощью пары тегов `<!--...-->` или `<COMMENT>...</COMMENT>`.

Вставка даты

Хорошим тоном в Web-дизайне считается указание даты последнего обновления Web-страницы. В самом деле, дату вставить очень легко — достаточно просто написать число, месяц и год в соответствующем формате. Но что делать, если вы не в состоянии вспомнить, какое сегодня число (как, например, автор этой книги)?

Dreamweaver может вставить дату за вас. Вы просто задаете формат даты и времени и указываете, нужно ли обновлять вставленную дату всякий раз при сохранении страницы. Видите, Dreamweaver может даже вместо вас следить за календарем!

Поскольку создаваемая нами Web-страница очень проста, мы не будем ставить на ней дату обновления. Вместо этого создадим пустую Web-страницу

и вставим дату в нее, для чего переключимся на страницу **Common** панели объектов и нажмем кнопку **Date** (рис. 3.56). Вы также можете выбрать пункт **Date** меню **Insert**. После этого на экране появится диалоговое окно **Insert Date**, показанное на рис. 3.57.

Рис. 3.56. Кнопка **Date** панели объектов

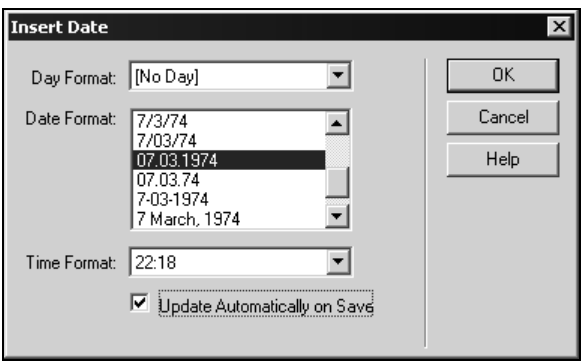


Рис. 3.57. Диалоговое окно **Insert Date**

Раскрывающийся список **Day Format** для нас бесполезен, т. к. представляет все имена дней недели по-английски. Поэтому оставьте выбранным пункт **[No Day]**.

В списке **Date Format** выбирается формат даты. Выберите один из двух форматов "число.месяц.год", которые там есть.

Раскрывающийся список **Time Format** задает формат времени. Выберите пункт **22:18**, если вы хотите отображать время в 24-часовом формате, пункт **10:18 PM**, если хотите видеть время в 12-часовом формате, или пункт **[No Time]**, если время вам вообще не нужно.

Флажок **Update Automatically on Save** включает или отключает режим обновления даты при сохранении страницы.

Как обычно, кнопка **OK** выполняет вставку даты, а кнопка **Cancel** — отменяет.

Для нашего примера выберите один из вышеуказанных форматов даты в списке **Date Format**, пункт **22:18** — в раскрывающемся списке **Time Format** и включите флажок **Update Automatically on Save**. Не забудьте нажать кнопку **OK**! После этого на страницу будет помещена сегодняшняя дата.

Если вы при вставке даты включили флажок **Update Automatically on Save**, Dreamweaver позволит вам изменить формат даты и (или) времени. Выдели-

те вставленную дату — и редактор свойств изменит свой вид, предоставив вам кнопку **Edit Date Format**. При нажатии на нее на экране появится диалоговое окно **Insert Date** (см. рис. 3.57), в котором вы сможете изменить формат даты и времени.

Теперь испытаем Dreamweaver. Для этого сохраним страницу, скажем, под именем 3.4.htm. После этого вы, наверно, заметите, что время, которое вы поместили на страницу вместе с датой, изменится. (Если этого не случится, подождите минуты две и потом снова сохраните страницу.) То есть, таким образом, Dreamweaver действительно обновляет дату и время при сохранении страницы.

"Чистка" HTML-кода

В процессе работы над Web-страницей Dreamweaver вставляет в текст и убирает из него множество различных HTML-тегов. Хоть он и очень "интеллектуален" и никогда не вставит и не удалит лишнего, все-таки стоит время от времени "чистить" получившийся код, чтобы сделать его яснее и компактнее. Эта "чистка" включает в себя:

- удаление парных тегов, "потерявших" свою пару;
- удаление лишних (избыточных) тегов. Такие теги могут включаться в код для того, чтобы сделать его совместимым со старыми программами Web-обозревателей;
- объединение тегов, где это возможно;
- удаление комментариев.

За "чистку" HTML-кода "отвечает" пункт **Clean Up HTML** меню **Command**, выбрав который, вы откроете диалоговое окно **Clean Up HTML / XHTML**, показанное на рис. 3.58.

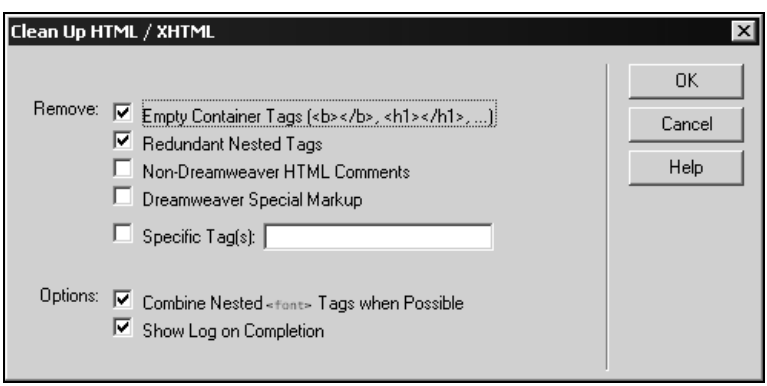


Рис. 3.58. Диалоговое окно **Clean Up HTML / XHTML**

Группа флажков **Remove** поможет вам задать, что именно следует удалять при "чистке":

- **Empty Container Tags** — "пустые" теги, т. е. парные теги без содержимого;
- **Redundant Nested Tags** — ненужные (избыточные) теги;
- **Non-Dreamweaver HTML Comments** — HTML-комментарии, вставленные пользователем;
- **Dreamweaver Special Markup** — HTML-комментарии, вставленные самим Dreamweaver для своих служебных целей, например, чтобы пометить вставленную дату с возможностью обновления;
- **Specific Tag(s)** — теги, перечисленные в поле ввода, находящемся правее.

Группа флажков **Options** поможет вам задать дополнительные параметры "чистки":

- **Combine Nested <*font*> Tags when Possible** — по возможности комбинирует вложенные теги `<FONT>` с разными параметрами;
- **Show Log on Completion** — показывает статистику по окончании "чистки".

Задав нужные параметры (в подавляющем большинстве случаев можно оставить заданные по умолчанию), нажмите кнопку **OK** для запуска процесса "чистки" или **Cancel** — для отказа от нее. "Чистка" может занять довольно много времени, особенно если Web-страница сложна, а компьютер, на котором установлен Dreamweaver, маломощен. По окончании, если был включен флажок **Show Log on Completion**, на экране появится окно статистики, где будет описано, что сделал Dreamweaver в процессе "чистки". В нашем случае (рис. 3.59) ничего сделано не было — HTML-код нашей страницы и так был "чист". Закройте это окно нажатием кнопки **OK**.

Рис. 3.59. Окно статистики "чистки" HTML-кода

Проверка совместимости HTML-кода

В мире эксплуатируется много разных программ Web-обозревателей, сильно различающихся как функциональными возможностями, так и степенью совместимости с различными интернет-стандартами. Более того, разные версии двух самых распространенных Web-обозревателей также имеют очень

большие отличия. И Web-страница, прекрасно отображающаяся в одной программе, совершенно не работает в другой.

Как избежать проблем с совместимостью? Что нужно для этого делать?

Прежде всего, конечно, соблюдать все стандарты HTML. Не кидаться сломя голову за новомодными, свежевведенными расширениями этих самых стандартов и, тем более, нестандартными, "фирменными" возможностями той или иной программы. Иначе вы можете сильно сузить свою аудиторию. Кроме того, фирмы Microsoft и Netscape разработали ряд рекомендаций для Web-дизайнеров, которые позволят им создавать специальные, "совместимые" страницы, одинаково хорошо (или, по крайней мере, сносно) отображаемые в большинстве программ. Рекомендуем вам ознакомиться с этими рекомендациями, хотя в конкретных случаях помощи от них не так уж и много, и вам придется изрядно поломать голову.

Но как узнать, будет ли данная Web-страница нормально отображаться в разных программах Web-обозревателей? Вам поможет в этом Dreamweaver.

Выберите пункт **Check Target Browsers** в подменю **Check Page** меню **File**. На экране появится диалоговое окно **Check Target Browsers** (рис. 3.60).

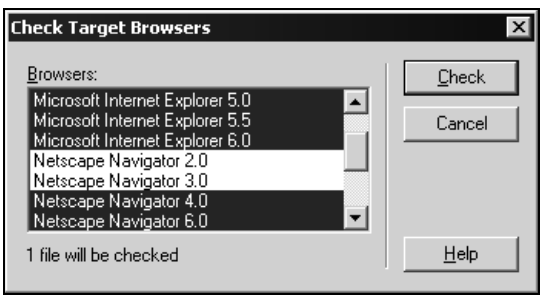


Рис. 3.60. Диалоговое окно **Check Target Browsers**

В списке **Browsers** представлены все программы, о которых "знает" Dreamweaver. Версия MX поддерживает разные версии Internet Explorer, Navigator и Opera; новейший Web-обозреватель Mozilla там отсутствует. Выберите нужные программы, щелкая мышью на соответствующих строках списка, после чего нажмите кнопку **Check**.

После довольно долгого ожидания Dreamweaver откроет особую панель **Target Browser Check** и выведет в ней результаты проверки страницы на совместимость (рис. 3.61) в виде списка. В колонке **Description** этого списка будет дано краткое описание найденной несовместимости, увы, на английском. Дважды щелкните мышью на нужной строке списка — и Dreamweaver, переключив окно документа в режим показа страницы и кода, подсветит проблемный HTML-тег.

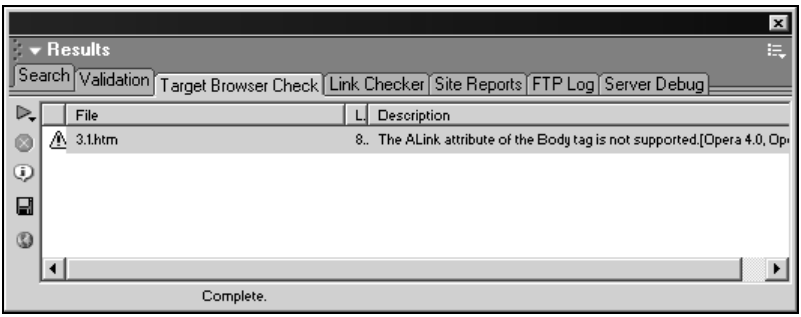


Рис. 3.61. Панель **Target Browser Check**

Здесь сразу видно, что атрибут `ALINK` (цвет активной гиперссылки) тега `<BODY>` не поддерживается Opera. Конечно, это не смертельно — в целом наша первая Web-страничка совместима со всеми наиболее распространенными программами Web-обозревателей. Но эта функция, встроенная в Dreamweaver, позволит вам в будущем избежать серьезных проблем с совместимостью.

Что дальше?

Итак, мы создали первую, главную страницу нашего сайта. В процессе ее создания мы научились работать с текстом и гиперссылками, используя для этого стандартные инструменты Dreamweaver. Следующим шагом будет работа с графикой и мультимедийными файлами, иными словами, внедренными объектами.

В *главе 4* мы научимся вставлять на страницы графические изображения, звуковые и видеоклипы. А заодно создадим страничку с более подробными сведениями о гипотетическом Иване Ивановиче и его портретом.

ГЛАВА 4

Рисунки, звуки, фильмы

В *главе 3* мы научились работать с текстом. В этой главе мы разберемся со всем, что относится к внедренным объектам: графическими изображениями и мультимедийными данными (прежде всего, аудио- и видеоклипами). Мы научимся помещать их на Web-страницы и использовать для специальных целей, например, для создания изображений-гиперссылок и изображений-карт. И Dreamweaver нам в этом поможет.

Следующая по очереди Web-страница нашего сайта — сведения об авторе. А какая страница об авторе обойдется без его портрета. Но прежде чем помещать на Web-страницу портрет (да и любое изображение), неплохо было бы создать эту самую страницу. Этим мы сейчас и займемся.

Сначала создадим в Dreamweaver новую пустую Web-страницу. Вы уже знаете, как это делается. Не будем ничего особо выдумывать: достаточно сочинить немного текста. Единственное: цвета и шрифты на новой странице должны быть такими же, как и на ранее созданной. Это необходимо, чтобы соблюсти единообразное оформление, чтобы пользователь знал, что пока еще находится на одном сайте.

На рис. 4.1 показана наша новая Web-страница. Опустим рассказ о ее создании, т. к. в этой главе речь пойдет совсем о другом.

Как видите, мы не слишком утруждали себя придумыванием деталей биографии нашего героя. Хватило нескольких строк текста, чтобы получилась Web-страница. Сохраним ее в файле 4.1.htm.

Теперь откроем первую нашу страницу (файл 3.1.htm). Сейчас мы создадим первую "внутрисайтовую" гиперссылку, т. е. гиперссылку, ведущую на другую страницу нашего сайта. Для этого сначала выделим строку "подробнее обо мне, любимом" (последняя строка списка). И обратим внимание на редактор свойств.

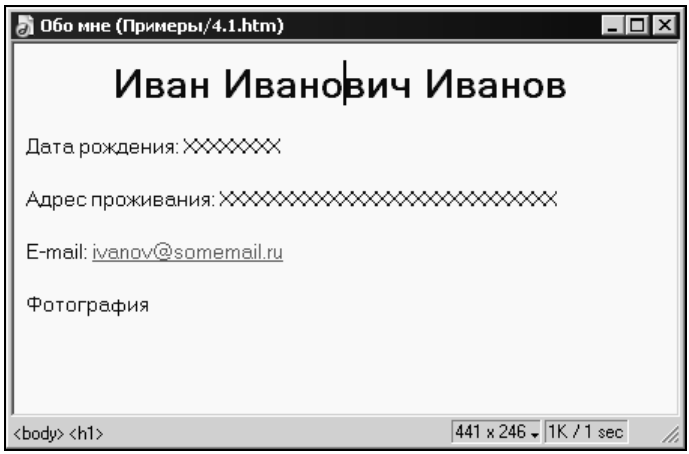


Рис. 4.1. Web-страница сведений об авторе

Вы уже заметили — справа от поля ввода адреса находятся две кнопки. Одна из них имеет вид мишени и расположена левее; она нам сейчас бесполезна. Вторая гораздо интереснее: она имеет вид папки и при нажатии открывает диалоговое окно **Select File**, показанное на рис. 3.41. В верхней части этого окна находится список файлов текущей папки и раскрывающийся список выбора папок, совсем как в стандартном диалоговом окне открытия файлов Windows. Вы должны будете выбрать нужный файл Web-страницы, чтобы создать на нее гиперссылку. Так мы и сделаем. И не забудьте нажать кнопку **OK**. Вот и все — гиперссылка создана.

И обязательно сохраните нашу главную страницу. Частое сохранение документов (любых) — важнейшая и надежнейшая гарантия от потери данных. После этого главную страницу можно закрыть и сосредоточиться на сведениях об авторе.

Работа с графическими изображениями

Графические изображения (а также аудио- и видеоклипы), как вы помните, являются внедренными элементами, т. к. они не помещаются в сам HTML-код Web-страницы, а хранятся в отдельных файлах. Дело в том, что графика принципиально отличается от текста, кодируется и хранится иначе, чем текст. Поэтому совместить в одном файле HTML-текст и графику невозможно.

Примечание

Здесь нужно сделать оговорку. Существует язык *VRML* (Virtual Reality Markup Language — язык разметки виртуальной реальности), с помощью которого создаются целые виртуальные миры, которые можно исследовать. Эти миры запи-

сываются в виде текстовых тегов, аналогичных тегам HTML, встраиваемым прямо в HTML-код. Также фирмой Microsoft разработан язык *VML* (Vector Markup Language — язык разметки векторов), служащий для создания графических изображений, тоже в виде текстовых тегов. Но этот язык распространен очень слабо и поддерживается пока только Internet Explorer, начиная с версии 5.0.

Два вида графических изображений

Все графические изображения делятся на две большие группы по принципу кодирования, хранения в файлах и отрисовки на устройствах вывода (экран, принтер и т. п.). Давайте их рассмотрим.

Растровые изображения представляют собой набор точек; каждая такая точка может иметь какой-либо цвет, от белого до черного. Цвета всех точек, составляющих подобное изображение, записываются в массив, который, в свою очередь, вместе с некоторой служебной информацией сохраняется в файле. Такой массив называется *растром*.

К растровым относятся хорошо вам известные изображения форматов *GIF* (Graphic Interchange Format — формат обмена графикой) и *JPEG* (Joint Pictures Encoding Group — группа кодирования неподвижных изображений), которые вы загружаете из Интернета, чтобы полюбоваться творениями Бориса Вальехо. К растровым относится также стандартный формат хранения изображений в Windows — *BMP* (BitMaP — битовая матрица). Еще стоит упомянуть весьма напористого новичка — формат *PNG* (Portable Network Graphics — перемещаемая сетевая графика). Этот пока еще малораспространенный формат может стать преемником формата GIF, если недавние попытки сделать последний платным все-таки увенчаются успехом.

На самом деле, растровых форматов огромное множество, но подавляющее большинство из них — либо "фирменные", поддерживаемые одной программой, либо малораспространенные в силу каких-то причин, либо просто не используемые в Интернете.

Достоинствами растровых изображений являются исключительно простая обработка и вывод на устройство отображения. Действительно, что может быть проще, чем считать массив значений цветов точек и эти самые точки показать пользователю. Недостатки: большой размер (особенно, качественных изображений с большим количеством цветов) и резкая потеря качества при масштабировании.

Хотя большой размер — не такой уж и непреодолимый недостаток. Форматы GIF, JPEG и PNG стандартно поддерживают *сжатие* массива данных с помощью специальных алгоритмов. Однако тут кроется другая опасность. Дело в том, что эти алгоритмы реализуют так называемое *сжатие с потерями*, при которых часть информации, не очень существенная для отображе-

ния, отбрасывается, в результате чего объем массива сильно уменьшается. ("Обычные" программы архиваторов, которыми вы пользуетесь, например Zip или Rar, реализуют *сжатие без потерь*.) При этом, если задать слишком сильное сжатие такого изображения, может пострадать его качество из-за того, что слишком большая часть информации будет отброшена. Впрочем, это уже проблема Web-художника, который будет готовить подобное изображение к публикации в Интернете.

Какие форматы растровых изображений используются в Web-дизайне? В основном, GIF и JPEG. Угадайте, почему? Конечно, потому, что они поддерживают сжатие. Причем, изображения GIF применяют для элементов оформления страниц (линии, маркеры списков и т. п.) и штриховых рисунков, а JPEG — для полутоновых рисунков с большим количеством цветов. Это вызвано тем, что для сжатия изображений этих форматов применяются разные алгоритмы, имеющие свои сильные стороны и свои недостатки.

Формат GIF имеет особое свойство, за что его любят Web-художники. Дело в том, что из-за особенностей этого формата в один GIF-файл можно записать последовательность графических изображений, фактически настоящий фильм. Многочисленные "живые" (даже слишком) рекламные картинки — баннеры — которые в последнее время просто залепили Web-страницы, сделаны в формате *"анимированный GIF"*. Иногда такими картинками злоупотребляют, и ничего хорошего из этого не получается.

Другое достоинство формата GIF — возможность задать *"прозрачный" цвет*. Впоследствии Web-обозреватель вместо точек, имеющих этот цвет, будет подставлять точки "фона" родительского элемента. Говоря просто, можно сделать рисунок с "дырками", сквозь которые "просвечивает" то, что находится под рисунком.

Формат PNG, как говорят его создатели, объединяет возможности GIF и JPEG, не "прихватывая" заодно с собой их недостатки. Но пока он что-то не очень популярен в Сети. Однако если GIF все-таки сделают платным...

Internet Explorer также поддерживает формат BMP. Непонятно, зачем это нужно, ведь другие Web-обозреватели его не поддерживают.

Осталось только привести расширения, под которыми сохраняются файлы того или иного формата. Файлы GIF, PNG и BMP имеют "говорящие" расширения gif, png и bmp, а файлы JPEG — jpeg, jpg или jpe.

Вторая разновидность графических изображений — *векторные*. В отличие от растровых, состоящих из точек, они состоят из линий, называемых *примитивами*. Каждый такой примитив описывается определенной формулой, имеющей конкретный набор параметров. Вот эти параметры и сохраняются в массиве данных. К векторным относится формат изображений, созданных в популярнейшей программе Macromedia Flash, а также второй стандартный формат хранения изображений в Windows — WMF (Windows MetaFile — ме-

тафайл Windows). Вообще, существует много форматов векторных изображений, здесь перечисляться они не будут.

Что предлагают нам векторные изображения? Во-первых, небольшой объем, т. к. параметры какой-нибудь загогулины занимают значительно меньше места, чем весь набор составляющих ее точек. Во-вторых, исключительно простое масштабирование: Web-обозревателю достаточно перевычислить формулы с новыми параметрами размеров и нарисовать новую картинку на основе результатов этих вычислений. Недостаток: очень сложные алгоритмы вывода, включающие вычисление сложных формул. К тому же, чем векторное изображение сложнее, тем оно больше и тем дольше выводится на экран.

В Web-дизайне широко используется только один формат векторной графики — Flash. Но назвать это чудо программистского искусства графической программой не поворачивается язык — это мощнейший мультимедийный комплект, позволяющий создавать целые фильмы со звуковым сопровождением, причем, фильмы интерактивные, реагирующие на действия пользователя. При этом файлы, имеющие расширение swf, отличаются небольшими размерами и быстро грузятся даже по относительно слабым каналам. Все Web-обозреватели имеют в своем составе дополнительные модули, позволяющие просматривать фильмы Flash, а те, кто их не имеет, могут без проблем загрузить с сайта Macromedia.

Примечание

Формат VML, о котором говорилось выше, также растровый.

Вставка графического изображения

Ну вот, теория закончена. Пора переходить к практике.

Прежде всего, запасемся портретом, который впоследствии поместим на нашу страницу. Только вот откуда его взять? Ведь нашего Ивана Ивановича Иванова не существует в природе. Автор нашел выход... хоть художник из него не бог весть какой. Готовый "портрет", если можно так выразиться, сохранен в файле Ivanov.gif.

Посмотрим на страницу сведений об авторе. Портрет мы вставим сразу же после слова "Фотография". Поместим текстовый курсор в конце этого слова и посмотрим на вкладку **Common** панели объектов — там находится кнопка **Image** (рис. 4.2). Вы также можете воспользоваться пунктом **Image** меню **Insert** или нажать комбинацию клавиш <Ctrl>+<Shift>+<I>.

Рис. 4.2. Кнопка **Image** панели объектов

Нажмем эту кнопку. На экране появится диалоговое окно **Select Image Source**, показанное на рис. 4.3.

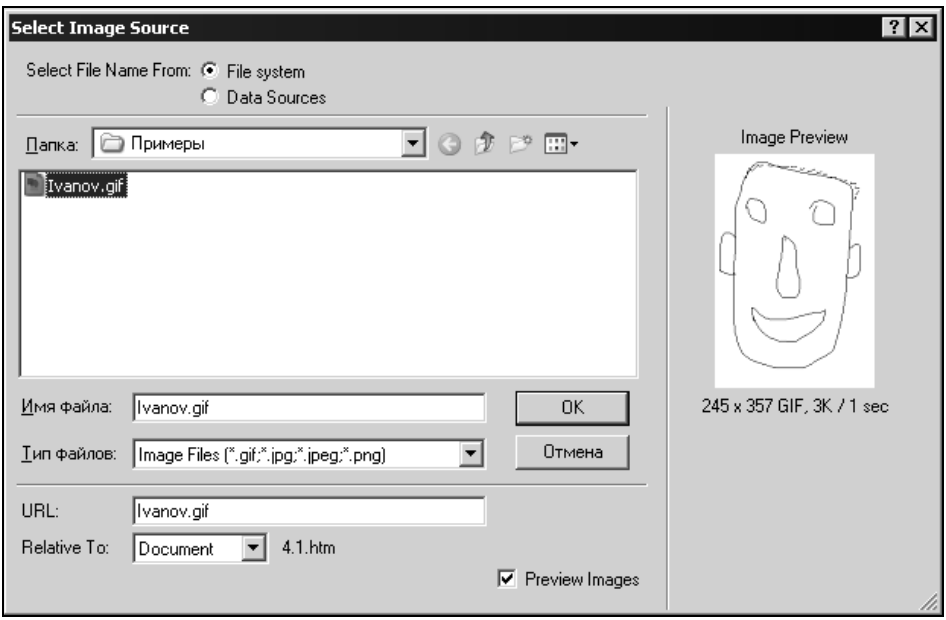


Рис. 4.3. Диалоговое окно **Select Image Source**

Да-а-а... Взглянуть страшно. Но вас предупреждали, что автор не дружит с живописью!

В общем, здесь описывать особо нечего. Раскрывающийся список папок и список файлов позволят вам выбрать нужную папку и файл. В поле ввода **Имя файла** появится имя выбранного файла (или вы можете вручную ввести его туда). А раскрывающийся список **Тип файлов** позволит вам выбрать, какой тип файлов вы хотите найти. Все это знакомо вам по стандартным диалоговым окнам открытия и сохранения файлов Windows. Единственное отличие — справа находится панель предварительного просмотра, где вы в данный момент видите физиономию Ивана Ивановича. А если она вам настолько противна, что вы хотите убрать ее с глаз долой, просто отключите флажок **Preview Images**.

Итак, вы выбрали файл, где находится портрет нашего героя. Осталось нажать кнопку **OK**. После этого наша страница примет такой вид, как на рис. 4.4.

Ну вот! Мало того, что этот портрет страшен, как семь смертных грехов, так он еще и занял почти все окно! Давайте его уменьшим, а то пользователь испугается.

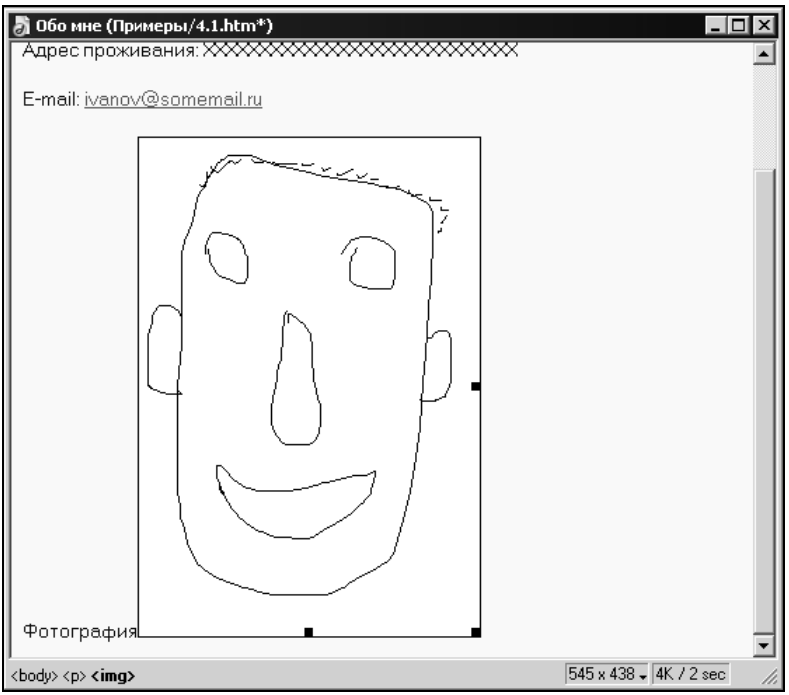


Рис. 4.4. Страница сведений об авторе с его портретом

Обратите внимание, что на правой и нижней границе портрета имеются небольшие черные квадратики. Это так называемые *маркеры изменения размера*. Вы можете "захватить" мышью любой маркер и перетащить его на новое место, изменив тем самым горизонтальный или вертикальный размер изображения. А если вы хотите, чтобы оба размера изменялись пропорционально, перетащите мышью маркер, находящийся в правом нижнем углу изображения при нажатой клавише <Shift>. Вот так стало лучше (рис. 4.5).

Теперь сохраним получившуюся страницу и посмотрим на редактор свойств. Проверьте, выделен ли портрет: признаком этого служат маркеры изменения размеров. Если их не видно, щелкните мышью по портрету — и они появятся. То, что вы увидите после этого, показано на рис. 4.6.

Поля ввода **W** и **H** позволяют вам ввести вручную соответственно ширину и высоту изображения. Это может быть полезно, если выделенное изображение — часть оформления сайта и должно плотно "прилегать" к другим таким же элементам оформления. В остальных случаях удобнее задавать размеры изображения, перетаскивая мышью маркеры размера.

При вставке графического изображения Dreamweaver сам заполняет эти поля. Мы рекомендуем вам не удалять эти значения. Дело в том, что Web-обозреватель после загрузки страницы отобразит еще не загруженные изо-

бражения в виде пустых рамок. Если размеры изображений были явно заданы, они будут сразу же применены к рамкам, и оформление страницы не нарушится. В противном случае Web-обозреватель отобразит рамки некоего размера по умолчанию, и при последующей загрузке изображений их размеры будут меняться, что вызовет изменение самой страницы. А это очень неприятно.

Поля ввода **V Space** и **H Space** задают соответственно вертикальное и горизонтальное расстояние от края изображения до обтекающего его текста. По умолчанию оба они равны нулю.

Поле ввода **Src** задает имя файла, где хранится графическое изображение. Справа от него вы видите уже знакомые вам две кнопки. Нажав на правую из них (с изображением папки), вы откроете диалоговое окно **Select Image Source**, показанное на рис. 4.3.

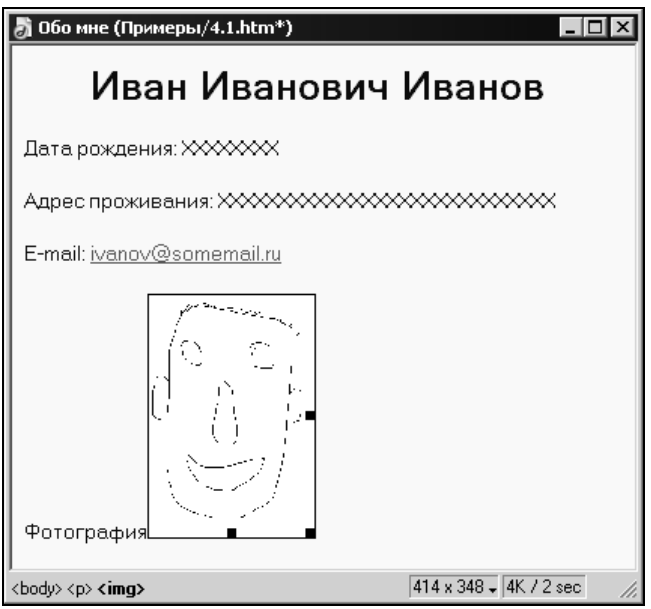


Рис. 4.5. Страница сведений об авторе после изменения размера портрета

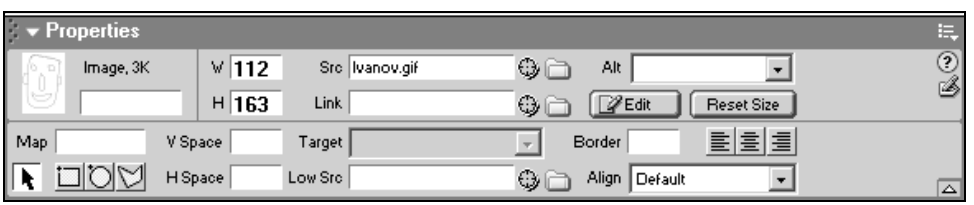


Рис. 4.6. Вид редактора свойств при выделенном изображении

Вы также можете изменить имя файла изображения, выбрав пункт **Source File** в контекстном меню или просто дважды щелкнув по изображению мышью. После этого на экране появится диалоговое окно **Select Image Source**.

Поле ввода **Low Src** аналогично полю **Src**, за тем исключением, что задает имя файла, где сохранено так называемое *"черновое" изображение*. "Черновое" изображение имеет меньший размер, как правило, за счет большего сжатия и низкого качества. Когда пользователь соединяется с Интернетом по низкоскоростному каналу, Web-обозреватель первым делом загружает "черновик", т. к. он имеет значительно меньший размер, и выводит его на странице. А уже потом, пока пользователь просматривает готовую страницу, постепенно загружается полноценное изображение и подменяет собой "черновик".

Рекомендуется изготавливать "черновик" только тогда, когда оригинальное изображение слишком велико, чтобы быстро загрузиться. В частности, это подойдет, если вы делаете сайт с художественной графикой.

Dreamweaver предоставляет вам другую возможность задать имя файла "черновика". Для этого выберите пункт **Low Source** в контекстном меню и укажите нужный файл в появившемся на экране диалоговом окне **Select Image Source**.

Поле ввода **Border** позволяет задать толщину рамки, отображаемой вокруг изображения. По умолчанию она равна нулю, т. е. рамки нет.

Поле ввода **Alt** задает так называемый *"альтернативный текст"*. Это придумано опять же для пользователей медленных каналов связи. После того как Web-обозреватель загрузит HTML-файл с Web-страницей, он вместо изображений, помещенных на ней, отобразит пустые рамки соответствующих размеров. Когда пользователь поместит курсор мыши над пустой рамкой рисунка, Web-обозреватель отобразит небольшую подсказку, содержащую этот самый "альтернативный текст". Поэтому мы рекомендуем вам всегда заполнять это поле ввода.

А теперь обратимся к раскрывающемуся списку **Align**. Оно позволяет вам задать выравнивание, а фактически — относительное местоположение изображения и обтекающего его текста. Но сначала поговорим о том, как графические изображения размещаются на Web-странице.

Дело в том, что изображение, вставленное в текст Web-страницы, ведет себя как обычный символ. Он находится в так называемом *"потоке" текста*, "льющемся" от начала до конца страницы и заполняющем ее целиком, и всецело подчиняется ему. В этом случае Web-дизайнер жестко ограничен "потоком" текста и не сможет переместить графический элемент туда, куда он хочет. И специально для такого случая был предусмотрен параметр относительного расположения графического изображения и обтекающего его текста.

Раскрывающийся список **Align** предоставляет Web-дизайнеру следующие пункты:

- **Browser Default** — расположение по умолчанию, обычно аналогично пункту **Baseline**;
- **Baseline** — низ изображения совпадает с *базовой линией текста* (воображаемой линией, по которой пишется строка текста);
- **Top** — верх изображения совпадает с верхом текста;
- **Middle** — середина изображения совпадает с базовой линией текста;
- **Bottom** — низ изображения совпадает с низом текста (обычно не то же самое, что **Baseline**);
- **TextTop** — верх изображения совпадает с верхом самого высокого символа текста (обычно не то же самое, что **Top**);
- **Absolute Middle** — середина изображения совпадает точно с *центральной линией текста* (линией, проходящей через центр строки);
- **Absolute Bottom** — низ изображения совпадает с низом самого низко сидящего символа текста;
- **Left** — изображение "прижимается" к левому краю страницы;
- **Right** — изображение "прижимается" к правому краю страницы;
- **Default** — расположение по умолчанию, обычно аналогично пункту **Baseline**.

В последних двух случаях изображение становится *"плавающим"*, т. е. не привязанным жестко к "потоку" текста. "Плавающее" изображение может быть смещено Web-обозревателем влево или вправо, при этом текст, в который оно было вставлено, может быть раздвинут. А в точке, где оно было вставлено, Dreamweaver отображает специальный *маркер "плавающего" изображения*, показанный на рис. 4.7. Этот маркер выводится только для удобства Web-дизайнера; в окне Web-обозревателя он виден не будет.

Рис. 4.7. Маркер "плавающего" изображения

Задать выравнивание изображения вы также можете в подменю **Align** контекстного меню.

Итак, с редактором свойств мы разобрались. Теперь давайте зададим свойства портрета нашего героя. Таким образом, расстояния от текста — 5 пикселов с обеих сторон, выравнивание — по левому краю, альтернативный текст — "Это я, Иван Иванович Иванов". Сохраним страницу.

Примечание

Изображение вставляется в текст с помощью одинарного тега `<IMG>`. Имя файла задается посредством атрибута `SRC`. Кроме него, этот тег имеет множество других атрибутов, соответствующих описанным нами параметрам. Получить подробное описание каждого из них вы можете, воспользовавшись поставляемым с Dreamweaver руководством по HTML.

Теперь опишем еще несколько возможностей, предлагаемых Dreamweaver для работы с изображениями. Сразу оговорюсь, что они весьма невелики — если вы хотите что-то подправить, лучше воспользуйтесь специализированным графическим редактором.

Кнопка **Reset Size** редактора свойств и одноименный пункт контекстного меню позволят вам сбросить размеры изображения в их значения по умолчанию. Это полезно, если вы сильно исказили размеры изображения и хотите начать все сначала.

Кнопка **Edit** редактора свойств и пункт **Edit With** ***<имя программы>*** контекстного меню позволят вам открыть выделенное изображение в программе, установленной в системе как программа для открытия этих файлов по умолчанию. Это та программа, в которой графические файлы будут открыты при двойном щелчке на них в окне Проводника. Вполне возможно, вам от этой функции не будет особой пользы (у автора, например, некоторые изображения открываются в Internet Explorer, в котором много не наредактируешь). В этом случае воспользуйтесь подменю **Edit With** контекстного меню; выберите в нем пункт **Browse**, затем выберите в появившемся на экране диалоговом окне открытия файла Windows исполняемый файл нужной программы и нажмите кнопку открытия. Изображение откроется в этой программе, и вы сможете сделать с ним все, что хотите.

Изображения-гиперссылки

В предыдущей главе мы научились преобразовывать фрагменты текста в гиперссылки. Но, кроме текста, гиперссылкой можно сделать и графическое изображение. Такие *изображения-гиперссылки* часто встречаются на Web-страницах.

Сделать изображение-гиперссылку очень просто, так же просто, как и гиперссылку текстовую. Вам необходимо только выделить нужное изображение и ввести в поле ввода **Link** редактора свойств нужный адрес.

Давайте, чтобы проиллюстрировать вышесказанное, вставим на нашу страницу со сведениями об авторе еще одно изображение. И преобразуем его в почтовую гиперссылку.

Но, прежде всего, заготовим само изображение. Это будет значок "коммерческое эт", часто использующийся в качестве символа электронной почты.

Создайте его в графическом редакторе или найдите готовый. Назовите файл с этим изображением Email.gif.

После этого поставьте курсор в начале строки "E-mail: ivanov@somemail.ru", сотрите символы "E-mail" и нажмите кнопку **Image** вкладки **Common** панели объектов. Выберите файл, где сохранено ваше "коммерческое эт", и нажмите кнопку **OK**. Измените размеры свежевставленного изображения, чтобы оно не сильно отличалось от размера шрифта текста, и установите для него выравнивание **Absolute Middle**. Результат вы можете видеть на рис. 4.8.

Рис. 4.8. Значок "коммерческое эт" — символ электронной почты

Теперь преобразуем этот значок в почтовую гиперссылку. Для этого выделим его, наберем в поле ввода **Link** редактора свойств почтовый адрес нашего героя

mailto:ivanov@somemail.ru

и нажмем клавишу <Enter>.

Внешне изображение-гиперссылка ничем не отличается от обычного изображения. Однако если вы вызовете Web-обозреватель для предварительного просмотра Web-страницы, то при наведении курсора мыши на это изображение он изменит свою форму на "указующий перст". А при щелчке на этом изображении откроется почтовый клиент.

Если параметр **Border** изображения-гиперссылки имеет значение, отличное от нуля, то рамка, рисуемая вокруг изображения, будет иметь такой же цвет, какой был установлен для гиперссылок в общих свойствах данной страницы. В нашем случае, такая рамка будет иметь светло-синий цвет для непосещенных, темно-синий для посещенных и ярко-красный для активной гиперссылок. Такой эффект можно специально использовать, чтобы сделать оригинальный дизайн Web-страницы, но обычно им не пользуются.

Ну и, конечно же, изображение-гиперссылка может указывать на обычную Web-страницу.

Примечание

HTML-код, использующийся для вставки изображения гиперссылки, имеет вид: `<A HREF="<Интернет-адрес>"><IMG SRC="<Адрес файла изображения>"></A>`.

Активные изображения

Изображение-гиперссылка при наведении на нее курсора мыши никак не проявляет себя. Конечно, можно выставить параметр **Border** равным пяти

или десяти, но это во многих случаях будет не совсем эстетично. Вот сделать бы так, чтобы изображение менялось при наведении на него курсора мыши, чтобы пользователь сразу видел, что это гиперссылка...

Оказывается, это можно сделать. Такие изображения, реагирующие на действия пользователя, называются *активными*. И Dreamweaver позволяет делать их несколькими щелчками мыши. Единственное, о чем вам нужно позаботиться, так это о втором изображении, которое будет появляться на месте исходного, когда пользователь наведет на него курсор мыши (так называемое *перекрывающее изображение*).

Давайте рассмотрим, как поместить на страницу активное изображение. Для этого используем страницу сведений об авторе 4.1.htm, куда мы уже вставили изображение-гиперссылку. Согласитесь, она выглядит не очень эстетично. Давайте заменим ее активным изображением. Создадим в графическом редакторе два изображения со значком "коммерческое эт": одно — обычное, а второе — инвертированное, которое будет перекрывать первое. Первое изображение сохраним в файле Email.gif, а второе — в файле Email2.gif. И, конечно, сотрем всю строку с адресом электронной почты так, чтобы на ее месте остался пустой параграф. Текстовый курсор оставим на получившемся пустом параграфе.

Активное изображение вставляется с помощью кнопки **Rollover Image** (рис. 4.9) вкладки **Common** панели объектов или пункта **Rollover Image** подменю **Interactive Images** меню **Insert**. При этом на экране появляется диалоговое окно **Insert Rollover Image**, показанное на рис. 4.10.

Рис. 4.9. Кнопка **Rollover Image** панели объектов

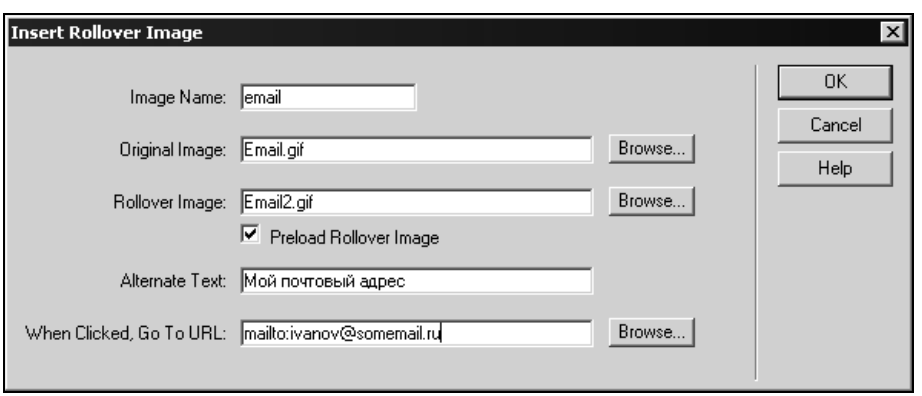


Рис. 4.10. Диалоговое окно **Insert Rollover Image**

В поле ввода **Image Name** вводится уникальное имя вставляемого активного изображения. Имя должно содержать только латинские буквы, цифры и знаки подчеркивания, причем начинаться должно с буквы. Проверьте только, действительно ли это имя уникально.

В поле ввода **Original Image** вводится имя файла оригинального изображение. Если вы не хотите вводить его вручную, нажмите кнопку **Browse** справа от этого поля ввода. После этого на экране появится диалоговое окно **Select Image Source** (см. рис. 4.3), где вы сможете выбрать нужный файл.

В поле ввода **Rollover Image** вводится имя файла перекрывающего изображения. Также, если вам неохота водить его вручную, на помощь всегда придет кнопка **Browse**.

Флажок **Preload Rollover Image** заставляет Dreamweaver сгенерировать и вставить в HTML-код вашей страницы небольшую программу-сценарий, которая заранее загрузит перекрывающее изображение. Этот флажок включен по умолчанию, и отключать его не стоит. Если же его отключить, то перекрывающее изображение будет загружено тогда, когда в нем возникнет нужда, т. е. когда пользователь поместит над изображением курсор мыши.

В поле ввода **Alternate Text** вводится "альтернативный текст". Лучше его ввести.

Последнее поле ввода **When Clicked, Go To URL** задает интернет-адрес, по которому произойдет переход, если пользователь щелкнет на изображении. Спасительная кнопка **Browse** придет на помощь тем, кто не любит стучать по клавиатуре.

На рис. 4.10 в соответствующие поля ввода подставлены все нужные значения. Вам остается ввести их и нажать кнопку **OK**, после чего активное изображение будет создано. Измените его размер, если хотите.

Теперь остается проверить созданное нами активное изображение в действии. Для этого выведите его в Web-обозревателе для предварительного просмотра и поместите курсор мыши над активным изображением. Вы увидите, как оно изменится. Если вы щелкнете по нему, откроется почтовый клиент.

Карты-изображения

Изображение-гиперссылка — это простейший случай графических гиперссылок, применяемых сейчас в Web-дизайне. Более сложным случаем являются активное изображение и *карта-изображение*, сложная графическая гиперссылка, представляющая собой изображение, разбитое на части, причем каждая часть является гиперссылкой и указывает на свой интернет-адрес. Например, можно сделать красивую заставку и разместить на ней надписи, соответствующие разделам сайта. Такие части, представляющие собой отдельные гиперссылки, называются *"горячими областями"*.

Для создания карты-изображения используются обычные графические изображения, созданные в любом графическом редакторе. Информация о местонахождении, форме, размерах "горячих областей" и интернет-адресах, на которые они указывают, хранится в HTML-коде страницы.

Dreamweaver позволяет очень просто создать карты-изображения. Нужные "горячие области" рисуются поверх изображения, после чего остается только ввести интернет-адреса, на которые они будут указывать.

В данный момент давайте не будем использовать на своих Web-страницах карты-изображения. Однако пример создания привести необходимо, поэтому мы создадим специальную Web-страничку и специальное изображение для учебных нужд. Файл изображения называется Map.gif, а файл странички — 4.2.htm. Эта страничка не будет содержать ничего, кроме изображения (рис 4.11).

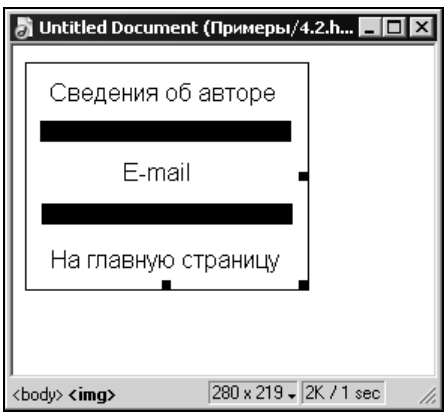


Рис. 4.11. Учебная Web-страница с картой-изображением

Разместить на изображении "горячие области" нам помогут несколько элементов управления, находящиеся в левом нижнем углу редактора свойств и пока еще нами не рассмотренные. Теперь пришла пора разобраться с ними (рис. 4.12).

Рис. 4.12. Элементы управления редактора свойств, использующиеся для создания "горячих областей"

Прежде всего, рассмотрим четыре кнопки, расположенные вдоль нижней границы окна редактора свойств. Они поделены на две неравные группы. Группа, находящаяся правее и состоящая из трех кнопок, позволяет размес-

тить на изображении соответственно прямоугольную, круглую и многоугольную "горячую область". Единственная кнопка, расположенная левее, позволяет манипулировать уже созданными "горячими областями".

Так как же разместить "горячую область"? Очень просто. Давайте начнем с прямоугольной. Щелкните на кнопке с изображением прямоугольника, затем щелкните по графическому изображению и, не отпуская левой кнопки мыши, протащите ее, пока не захватите всю область изображения, которую хотите сделать "горячей". Вы заметите, что за курсором мыши "тянется" светло-синяя линия. "Охватив" нужную область (в нашем случае, надпись "Сведения об авторе"), отпустите кнопку мыши, и выбранная вами область тотчас закрасится светло-синим.

После этого, как вы заметили, Dreamweaver подставит в поле ввода **Map** строку "Map". Что она значит? Это уникальное имя изображения-гиперссылки, которое будет использоваться Web-обозревателем. Если вы планируете на своей странице только одну карту-изображение, можете оставить это имя по умолчанию, в противном случае дайте ей уникальное имя, хотя бы "Map1". Заметьте, что в таких именах допускаются только латинские буквы, цифры и знаки подчеркивания, причем начинаться имя должно с буквы.

Но Dreamweaver не только сгенерировал уникальное имя для карты-изображения. Он еще и в очередной раз изменил вид редактора свойств (рис. 4.13). Теперь вы можете задать параметры только что созданной "горячей области".

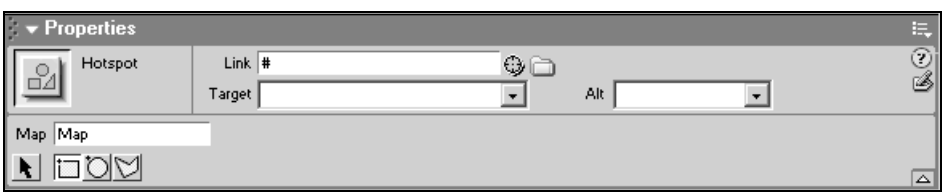


Рис. 4.13. Вид редактора свойств для выделенной "горячей области"

Все элементы управления нового редактора свойств вам уже знакомы. Это поле ввода **Src**, используемое для ввода интернет-адреса, раскрывающийся список **Target**, задающий цель гиперссылки, и поле ввода **Alt**, где вводится "альтернативный" текст. Здесь все понятно.

Разобравшись с тем, как вводятся параметры "горячей области", снова обратим внимание на изображение и саму "горячую область". Вы уже, наверно, заметили, что по углам прямоугольника, представляющего собой эту самую "горячую область", расположены небольшие синие квадратики — маркеры изменения размера. "Захватив" любой такой маркер мышью, вы можете изменить размеры "горячей области". А чтобы переместить ее на другое место,

просто перетащите ее мышью. Однако перед всеми этими операциями проверьте, что из кнопок, показанных на рис. 4.12, нажата та, на которой изображена стрелка, и, если нет, нажмите ее.

Но пора двигаться дальше. Введите в поле **Src** имя файла Web-страницы сведений об авторе (4.1.htm) и нажмите клавишу <Enter>.

Теперь добавим на наше изображение круглую "горячую область". Для этого нажмем кнопку с изображением круга и точно так же "протащим" мышь по изображению, охватив надпись "E-mail". Dreamweaver поместит круглую область там, где мы ему указали, и активизирует ее, разместив по четырем сторонам круга маркеры изменения размеров.

Вы можете изменять размеры и местоположение круглой "горячей области" так же, как и прямоугольной. Редактор свойств имеет в этом случае такой же вид (см. рис. 4.13). Поэтому мы не будем здесь останавливаться. Введите в поле ввода **Src** почтовый адрес Ивана Ивановича и нажмите клавишу <Enter>.

Многоугольная "горячая область" создается несколько сложнее. Нажав кнопку с изображением многоугольника, прежде всего, щелкните мышью в месте, где должна располагаться первая из *угловых точек* многоугольной области. После этого там появится синяя точка — начало нашего многоугольника. Затем щелкните мышью там, где должна быть вторая угловая точка, и Dreamweaver проведет между этими двумя точками линию. Далее вам останется щелкнуть в местах расположения остальных угловых точек многоугольника, а Dreamweaver сам проведет между ними линии, которые и образуют нужную нам многоугольную "горячую" область. Впоследствии вы можете перетащить мышью любую точку этой области. Вот только вставить новую или удалить старую угловую точку вы уже не сможете — вам придется удалить созданную "горячую" область и создать новую.

Выделите многоугольную "горячую область", если она еще не выделена, введите в поле ввода **Src** имя файла 3.1.htm — нашей главной страницы и нажмите клавишу <Enter>. Наша карта-изображение закончена. То, что получилось, показано на рис. 4.14.

После этого можно вызвать Web-обозреватель для просмотра нашей страницы и испытать нашу карту-изображение в действии. Сделайте так — она действительно работает!

Осталось описать совсем немного дополнительных функций, предлагаемых Dreamweaver для работы с картами-изображениями.

С помощью пункта **Image Map Name** контекстного меню вы можете изменить имя "горячей области", если редактор свойств недоступен. При выборе этого пункта на экране появится диалоговое окно **Change Attribute**, показанное на рис. 4.15. Введите новое значение атрибута (в нашем случае — имени "горячей области") и нажмите кнопку **OK**.

Рис. 4.14. Готовая карта-изображение

Пункт **Link** контекстного меню выводит на экран диалоговое окно **Select File**, где вы можете выбрать файл, на который будет ссылаться "горячая область".

Пункт **Alt** контекстного меню позволит вам с помощью диалогового окна **Change Attribute** задать "альтернативный" текст.

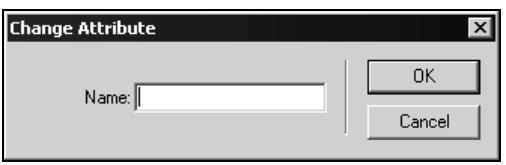


Рис. 4.15. Диалоговое окно **Change Attribute**

А о пунктах **Bring To Front** и **Send To Back** все того же неисчерпаемого контекстного меню поговорим подробнее.

Иногда случается так, что две "горячие области" перекрывают друг друга, т. е. имеют общий фрагмент. При этом при щелчке на такой "спорной территории" активизируется "горячая область", расположенная сверху, а сверху всегда появляется та область, что была создана последней. Но не всегда получается разместить "горячие области", как было предусмотрено по плану. Так вот, чтобы управлять их перекрытием, используются пункты **Bring To Front** и **Send To Back** контекстного меню. Первый переносит выделенную "горячую область" наверх, а второй — "загоняет" ее вниз.

В последнем случае вы также можете воспользоваться пунктами **Bring To Front** и **Send To Back** подменю **Arrange** меню **Modify**.

И еще. Иногда бывает нужно избавиться от показываемых Dreamweaver "горячих областей", например, чтобы оценить, как будет выглядеть карта-

изображение в окне Web-обозревателя. Конечно, можно загрузить его в самом Web-обозревателе для предварительного просмотра. Но можно сделать иначе. Отключите пункт-выключатель **Image Maps** в подменю **Visual Aids** меню **View**. При этом "горячие области" пропадут. Чтобы вернуть их назад, просто включите этот пункт.

Графика Macromedia Flash

Выше мы уже упоминали мощнейший пакет векторной Web-графики и анимации Macromedia Flash, "родного брата" Dreamweaver. "Братство" это проявляется не только в общем "родителе" — фирме Macromedia, но и в тесной интеграции двух пакетов. В частности, если у вас на компьютере установлены и Dreamweaver, и Flash, вы можете запустить Flash из Dreamweaver, нарисовать изображение и тут же вставить его в Web-страницу, открытую в окне Dreamweaver. Также Dreamweaver предлагает пользователю несколько "заготовок" различных надписей и кнопок, выполненных в формате Flash, которые вы также можете поместить на страницу.

Но даже если вы не имеете установленного на компьютере пакета Flash, вы все равно сможете размещать на своих страницах надписи и кнопки в формате Flash. Надписи Flash — это уже знакомые вам изображения-гиперссылки, но выполненные в формате Flash. Что касается кнопок Flash, то это просто активные изображения, имеющие вид кнопок и "нажимающиеся" при щелчке на них.

Если у вас установлена достаточно новая программа Web-обозревателя, то необходимый модуль расширения уже находится в ее составе. В противном случае вам придется посетить Web-сайт Macromedia, чтобы найти соответствующую версию проигрывателя. Так или иначе, вам время от времени все равно придется туда заходить, чтобы найти "свежий" проигрыватель, т. к. Flash постоянно обновляется, "обрастает" новыми возможностями, которые не поддерживаются старыми версиями проигрывателя.

Сейчас мы рассмотрим, какие возможности работы с Flash-изображениями предлагает нам Dreamweaver.

Для такого случая мы снова создадим небольшую учебную страничку. Страница со сведениями об авторе уже готова (или почти готова), так что не стоит пока над ней издеваться. Для наших экспериментов будет достаточно обычной пустой странички. Но, прежде чем помещать на нее Flash-изображения, сохраните ее в файле 4.3.htm, в противном случае Dreamweaver сам напомнит вам об этом. И закройте эту страницу.

Увы, увы, опять приходится говорить об ошибках, как бы нам этого не хотелось... Конечно, иногда ошибки бывают маленькими и незаметными, но Dreamweaver MX — не тот случай. Из-за досаднейшей промашки разработчиков он не всегда корректно воспринимает имена файлов и папок, на-

бранные русскими буквами. Поэтому проверьте имя папки, в которой вы сохранили Web-страницу 4.3.htm, и если ее имя или имя одной из папок верхнего уровня имеет русские имена, перенесите файл страницы в другую папку, имеющую в имени только английские имена и цифры. Теперь можете открывать вашу страницу 4.3.htm.

Сначала рассмотрим вставку *надписи Flash* — изображения в формате Flash, содержащего строку текста и работающего как изображение-гиперссылка.

Надпись Flash создается с помощью кнопки **Flash Text** (рис. 4.16) страницы **Media** панели объектов. Вы также можете воспользоваться пунктом **Flash Text** подменю **Interactive Images** меню **Insert**. Параметры создаваемой надписи Flash задаются в диалоговом окне **Insert Flash Text**, показанном на рис. 4.17.

Рис. 4.16. Кнопка **Flash Text** панели объектов

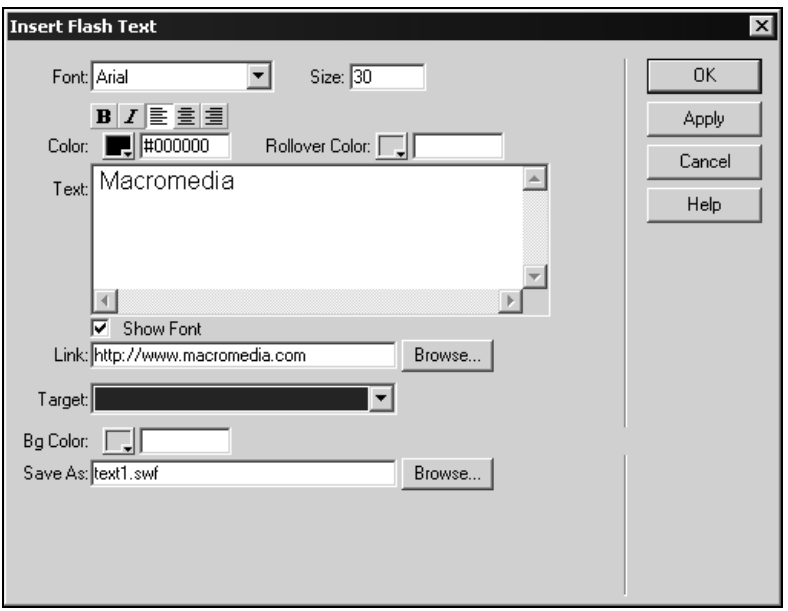


Рис. 4.17. Диалоговое окно **Insert Flash Text**

В раскрывающемся списке **Font** выбирается шрифт, которым будет написан текст надписи. В поле ввода **Size** задается его размер в пунктах. Кнопки-выключатели **B** и ***I*** позволяют сделать шрифт текста жирным или курсивным. Три кнопки с изображением выровненных влево, по центру и вправо

абзацев задают соответствующее выравнивание текста. Селектор цвета **Color** задает цвет текста, а **Rollover Color** — цвет, который примет текст, когда пользователь поместит над ним курсор мыши. В области редактирования **Text** вводится собственно текст надписи. Если флажок **Show Font** включен (а он включен по умолчанию), содержимое этой области редактирования показывается так, как оно будет выглядеть на Web-странице.

Поле ввода **Link** служит для задания интернет-адреса, по которому произойдет переход, если пользователь щелкнет на надписи. Также предусмотрена кнопка **Browse**. Цель гиперссылки задается с помощью раскрывающегося списка **Target**.

Селектор цвета **Bg Color** задает цвет фона надписи.

Последнее поле ввода **Save As** позволяет задать имя файла, где будет сохранена созданная надпись. По умолчанию Dreamweaver подставляет туда автоматически созданное имя файла, которое вы можете изменить, если хотите. Кнопка **Browse** вызывает на экран стандартное диалоговое окно сохранения файла Windows.

Наберите в области редактирования какой-либо текст, например, "Macromedia". Введите в поле ввода **Link** какой-нибудь интернет-адрес, например `http://www.macromedia.com`. Выберите шрифт и цвета по вкусу. И нажмите кнопку **OK**. То, что у вас получится, показано на рис. 4.18.

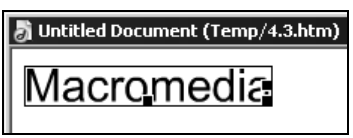


Рис. 4.18. Только что созданная надпись Flash

Редактор свойств позволяет отредактировать множество параметров надписи Flash (рис. 4.19). Как и у обычного изображения, доступно изменение геометрических размеров (поля ввода **W** и **H**), выравнивания (раскрывающийся список **Align** и одноименное подменю в контекстном меню), горизонтальное и вертикальное расстояние до текста (поля ввода **H Space** и **V Space**) и сброс размеров в изначальное состояние (кнопка **Reset Size**).

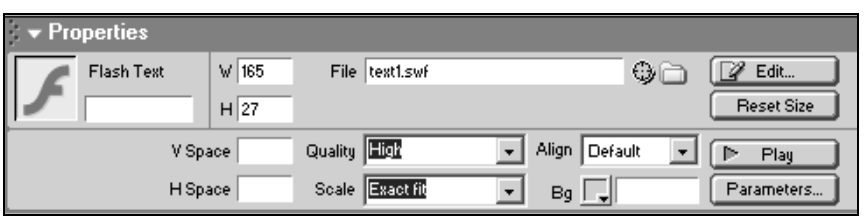


Рис. 4.19. Вид редактора свойств при выбранном изображении Flash

Остальные элементы управления задают специфичные для изображений Flash параметры. Кнопка **Edit** (а также одноименный пункт контекстного меню) выводит на экран диалоговое окно создания надписи Flash, где вы сможете ее отредактировать. В поле ввода **File** задается имя файла, где хранится Flash-изображение; если у вас есть другие файлы надписей Flash, оно вам может пригодиться. Селектор цвета **Bg** позволяет быстро сменить цвет фона надписи Flash.

Теперь обратим внимание на кнопку **Play**. Как вы помните, чтобы протестировать активное изображение, созданное нами выше, мы были вынуждены загрузить страницу в Web-обозревателе, т. к. Dreamweaver не поддерживает выполнение сценариев. Изображения формата Flash не используют сценариев и поэтому могут быть просмотрены в самом Dreamweaver. Для того чтобы запустить просмотр, щелкните на кнопке **Play**. После этого рамка выделения с маркерами изменения размеров пропадет и вы, если поместите курсор мыши над надписью, увидите, что ее цвет изменился. Вот только на щелчки она не реагирует.

Заметьте, что при этом кнопка **Play** изменит название на **Stop**. Чтобы вернуть надпись Flash в режим редактирования из режима просмотра, щелкните по ней.

Вместо нажатия кнопки **Play** вы можете выбрать пункт **Play** подменю **Plugins** меню **View** или нажать комбинацию клавиш <Ctrl>+<Alt>+<P>. Пункт **Play All** того же подменю (или комбинация клавиш <Ctrl>+<Alt>+ +<Shift>+<P>) запускают просмотр всех изображений Flash, имеющихся на странице. Соответственно, пункт **Stop** (или комбинация клавиш <Ctrl>+ +<Alt>+<X>) останавливает просмотр текущего изображения Flash, а **Stop All** (или комбинация клавиш <Ctrl>+<Alt>+<Shift>+<X> — тренируйте свои пальцы) — всех изображений Flash на данной странице.

Раскрывающийся список **Quality** позволяет задать качество отображения изображения Flash. По умолчанию выбран пункт **High** — высокое качество. Однако в этом случае модуль просмотра изображений Flash будет требовать слишком много системных ресурсов, и если вы планируете распространять свои Flash-творения среди владельцев маломощных компьютеров, то стоит выбрать пункт **Low**. Еще два пункта позволят достичь компромисса: **Auto High** сначала устанавливает высокое качество, а потом при необходимости понижает, а **Auto Low** — наоборот.

Раскрывающийся список **Scale** задает, как изображение Flash отображается в границах, установленных параметрами **W** и **H**. Выбранный по умолчанию пункт **Default (Show all)** заставляют проигрыватель Flash изменять масштаб изображения, "впихивая" его в нужные размеры, и предотвращать его искажение, соблюдая пропорции. Пункт **No border** выводит изображение в оригинальном масштабе; если при этом размеры **W** и **H** слишком малы, часть изображения может не быть видна. И, наконец, пункт **Exact fit** подобен **Default (Show all)**, но не предотвращает искажение масштаба.

Манипулируя этими параметрами, вы можете получить весьма интересные эффекты. Поэкспериментируйте с полученным изображением Flash и посмотрите, что выйдет из ваших экспериментов.

Теперь поговорим о *кнопках Flash*. Создайте новую Web-страницу и сохраните ее под именем 4.4.htm. (Впрочем, вы можете стереть надпись Flash, вставленную нами ранее на страницу 4.3.htm, очистив ее.)

Кнопка Flash создается с помощью кнопки **Flash Button** страницы **Media** панели объектов (см. рис. 4.20). Вы также можете воспользоваться пунктом **Flash Button** подменю **Interactive Images** меню **Insert**. Параметры создаваемой кнопки Flash задаются в диалоговом окне **Insert Flash Button**, показанном на рис. 4.21.

Рис. 4.20. Кнопка **Flash Button** панели объектов

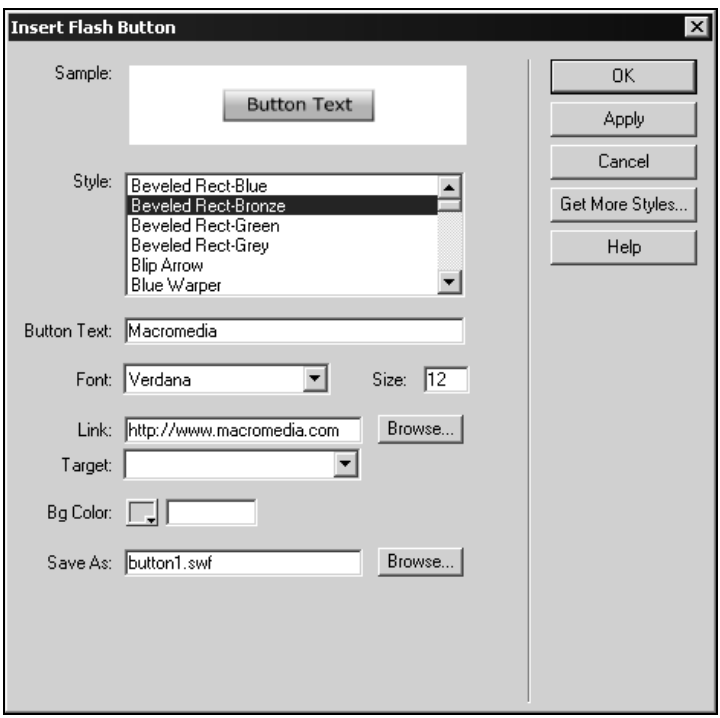


Рис. 4.21. Диалоговое окно **Insert Flash Button**

Здесь вы видите, прежде всего, большой список **Style**, где представлены все доступные в Dreamweaver стили кнопок Flash. Мы не будем их все рассматривать, т. к. описание получится большое и маловразумительное. Лучше

всего вам будет просто посмотреть на них воочию, ведь правду говорят, что лучше один раз увидеть, чем сто раз услышать. Поможет вам в этом панель предварительного просмотра, расположенная выше этого списка.

Текст кнопки вводится в поле ввода **Button Text**. Остальные элементы управления знакомы вам по диалоговому окну **Insert Flash Text** (см. рис. 4.17), и выполняют они те же самые функции.

Давайте введем в поле ввода **Button Text** строку `Macromedia`, в поле ввода **Link** — `http://www.macromedia.com`, выберем какой-нибудь приглянувшийся стиль кнопки и нажмем кнопку **OK**. Получившаяся кнопка может выглядеть, например, так — см. рис. 4.22.

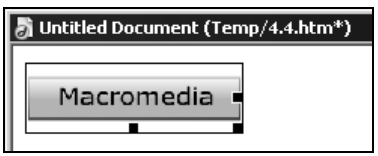


Рис. 4.22. Только что созданная кнопка Flash

Дизайнеры фирмы Macromedia приготовили для вас прекрасные шаблоны для создания Flash-кнопок. Не верите? Нажмите кнопку **Play** на редакторе свойств, поднесите курсор мыши к кнопке и посмотрите, что получится. К сожалению, печатная иллюстрация не может передать движение...

Редактор свойств позволит вам изменить различные параметры кнопки Flash. Все они аналогичны параметрам надписи Flash, да и всем изображениям Flash.

Примечание

Изображения Flash помещаются на страницу с помощью парного тега `<OBJECT>`. Этот тег служит для размещения не только изображений Flash, но и любых внедренных объектов, для которых в системе установлены модули просмотра. Таким образом можно, например, разместить на Web-странице документ MS Word или видеофильм. Тег `<OBJECT>` имеет множество атрибутов, которые служат для задания типа внедренного объекта, интернет-адреса дистрибутивного файла модуля просмотра, служащего для его обработки, геометрических размеров внедренного объекта и т. п. Мы не будем рассматривать эти атрибуты — вы можете найти их описания в электронном руководстве по HTML, поставляемом в составе Dreamweaver.

Выше говорилось, что тег `<OBJECT>` парный. Внутри этого тега помещаются описания различных параметров внедренного объекта (не самого тега `<OBJECT>`). Каждый параметр описывается с помощью одинарного тега `<PARAM>`, имеющего атрибуты `NAME` (имя параметра) и `VALUE` (значение параметра).

В дальнейшем мы вернемся к тегу `<OBJECT>`, когда будем рассматривать внедренные объекты.

Ну и, разумеется, вы можете поместить на Web-страницу любое изображение Flash, созданное вами или кем-то другим. Это выполняется очень просто. Нажмите кнопку **Flash** (рис. 4.23) вкладки **Common** панели объектов (эта же кнопка находится на вкладке **Media**). Вы также можете выбрать пункт **Flash** подменю **Media** меню **Insert** или нажать комбинацию клавиш <Ctrl>+<Alt>+<F>. На экране появится уже знакомое вам диалоговое окно **Select File**, где вы сможете выбрать нужный файл SWF.

Рис. 4.23. Кнопка **Flash** панели объектов

Графика Macromedia Shockwave

Осталось сказать чуть-чуть о других форматах поддерживаемых данных.

Macromedia Shockwave — непосредственный предшественник Flash. Чтобы поместить файл в этом формате на Web-страницу, нажмите кнопку **Shockwave** (рис. 4.24) вкладки **Media** панели объектов, выберите пункт **Shockwave** подменю **Media** меню **Insert** или нажмите комбинацию клавиш <Ctrl>+<Alt>+<D>. После этого на экране появится диалоговое окно **Select File**, где вы сможете найти и выбрать нужный файл.

Рис. 4.24. Кнопка **Shockwave** панели объектов

Внешний вид редактора свойств при выделенном изображении Shockwave показан на рис. 4.25. Как видите, свойства изображения Shockwave аналогичны соответствующим свойствам изображения Flash за исключением отсутствия некоторых параметров.

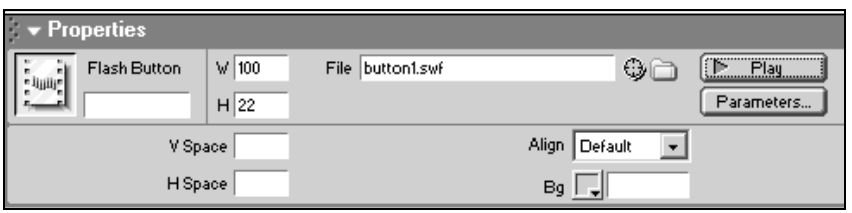


Рис. 4.25. Вид редактора свойств при выделенном изображении Shockwave

Фоновые изображения

Выше мы рассмотрели, как размещаются на Web-странице всевозможные изображения в разных форматах. Все эти изображения представляли собой

полезное содержание страницы и находились в "потоке" текста либо жестко к нему привязанные, либо "плавающие" в нем. Однако стандарт HTML определяет еще и так называемые *фоновые изображения*, помещаемые под текстом и "просвечивающие" сквозь "прозрачный" цвет графических элементов страницы. В этом смысле фоновые изображения похожи на водяные знаки на гербовой бумаге.

Задать фоновое изображение для Web-страницы очень просто. Для этого вызовите на экран диалоговое окно **Page Properties** (см. рис. 3.50), для чего выберите пункт **Page Properties** меню **Modify** или контекстного меню или нажмите клавиши <Ctrl>+<J>. В поле ввода **Background Image** введите имя файла фонового изображения или нажмите кнопку **Browse** и выберите нужный файл в диалоговом окне. После этого нажмите кнопку **OK**. Получившийся результат может выглядеть, например, так — см. рис. 4.26. (Автор создал пустую Web-страницу, поместил на нее фрагмент текста этой книги, задал для него в качестве фонового изображения файл J0143756.gif, поставляемый в составе Microsoft Office 2000, и сохранил результат в файле 4.5.htm.)

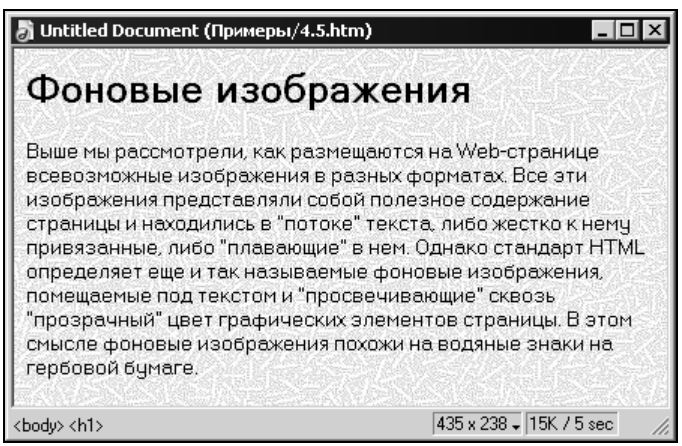


Рис. 4.26. Web-страница с фоновым изображением

Удачно подобранное фоновое изображение может заметно оживить Web-страницу. Но не следует выбирать в качестве фона слишком яркое или пестрое изображение. Лучше всего сразу после задания фона попробуйте сами почитать текст, выведенный на странице, и оцените, легко ли он читается. Если нет, замените фоновое изображение другим или совсем откажитесь от него. Можно также попытаться подобрать соответствующий цвет текста, чтобы он нормально читался, но не резал глаза. Снова посмотрите на рис. 4.26 — там текст выглядит на фоне достаточно контрастно, чтобы нормально читаться.

Фоновое изображение ведет себя точно так же, как обычные изображения, помещенные на странице: Web-обозреватель сначала загружает HTML-файл

страницы, а потом — все сопутствующие файлы, в том числе изображения. Это значит, что какое-то время фон на вашей странице будет отсутствовать. И вам нужно позаботиться о том, чтобы пользователь хотя бы смог прочитать текст вашей страницы. Для этого подберите цвет ее фона, аналогичный общему тону фонового изображения. Так, если фоновое изображение имеет темный тон, задайте для страницы темно-серый, темно-синий или черный цвет фона. И не следует в этом случае оставлять фон страницы белым, т. к. ваш текст тоже, скорее всего, будет белого цвета.

Мультимедиа

Мультимедиа (от англ. *multimedia* — многосредность) — это сокращенное обозначение информации, отличающейся от простых текста и графики. Мультимедиа — это звуки, музыка, видео, анимация, в последнее время оккупировавшие диски наших компьютеров, загромоздившие их гигабайтами полезной и бесполезной информации. Мультимедиа — это мощнейшие трехмерные ускорители на видеокартах, это звуковые контроллеры, ставшие (наконец-то) стандартом, CD- и DVD-диски и соответствующие им приводы. Мультимедиа — это Web-страницы, пестрящие анимацией, оглушающие каждого встречного приветственной музыкой. Мультимедиа — это символ современного компьютерного мира.

Разумеется, мы не можем пройти мимо такой насущной темы. Да и сами создатели Dreamweaver не могли ее проигнорировать. Dreamweaver оснащен впечатляющими средствами помещения на Web-страницы мультимедийного содержания, да и вообще любых внедренных объектов. И мы их сейчас рассмотрим.

Но, прежде всего, решим, куда же помещать наше мультимедийное содержание. Как правило, его помещают на главную страницу, и оно будет проигрываться либо при ее (страницы) открытии, либо после того, как пользователь нажмет на кнопку, запускающую проигрывание. Так поступим и мы. И еще: это мультимедийное содержание не должно быть очень большим, иначе пользователь устанет ждать, пока оно загрузится, и уйдет с вашего сайта.

Не будем говорить о проблеме соблюдения авторских прав — это и так понятно, ведь практически любое авторское произведение защищено законодательством от кражи и незаконного копирования (что, собственно, одно и то же). В качестве решения проблемы вы можете поместить на свою страницу небольшой кусочек, скажем, музыкального произведения или фильма, сопроводив его текстом, предлагающим пользователю купить соответствующее произведение на кассете или компакт-диске, если оно ему понравится. Это обычная практика музыкальных сайтов, соблюдающих (или пытающихся соблюдать) законы.

Есть, конечно, и второй выход: написать музыку или снять фильм самому...

Поддержка мультимедийных данных

Поговорим о том, каким же образом Web-обозреватель обрабатывает мультимедийное содержание.

Некоторые типы мультимедийных данных поддерживаются Web-обозревателем непосредственно. (Хотя мультимедийными данными это назвать сложно.) Как вы поняли, речь идет об обычных растровых изображениях в формате GIF, JPEG или PNG (они помещаются на страницы с помощью тега `<IMG>`). Internet Explorer также поддерживает свой собственный тег `<BGSOUND>` для привязки к странице фонового музыкального сопровождения и атрибут `DYNSRC` тега `<IMG>`, с помощью которого на страницу можно поместить фильм. Но эти возможности не являются стандартными, поэтому мы не будем их рассматривать. Если вы хотите узнать о них побольше, обратитесь к электронному руководству по HTML.

Но поддержка очень и очень многих форматов данных в Web-обозреватель не заложена — форматов так много, что охватить все просто невозможно. Проблема решается использованием дополнительных программ. Каждая такая программа "отвечает" за свой формат данных. И, когда Web-обозреватель получает какие-либо данные, которые он не может обработать непосредственно, он загружает соответствующую программу.

Мультимедийные данные, не поддерживаемые Web-обозревателем, помещаются на страницу с помощью особого тега. С помощью соответствующего атрибута задается имя *файла данных*, который и будет обрабатываться дополнительной программой. Программа читает данные из файла, обрабатывает их и генерирует на их основе какой-то экранный вывод, либо проявляет себя каким-то иным образом.

Но как Web-обозреватель определяет, какая программа нужна для обработки данных того или иного формата? Дело в том, что каждый формат данных имеет свой уникальный идентификатор, называемый *типом MIME* (Multipurpose Internet Mail Extensions — многоцелевые расширения почты Интернета). А дополнительная программа при установке сообщает системе, какой тип MIME ей "по зубам". Соответствия типов MIME форматам данных и обрабатывающим их программам записываются в Реестре Windows.

В табл. 4.1 приведены некоторые типы MIME и соответствующие им форматы данных.

Таблица 4.1. *Типы MIME*

Тип файлов	Тип MIME
Архив RAR	`application/x-tar`
Архив ZIP	`application/x-zip-compressed`

Таблица 4.1 *(окончание)*

Тип файлов	Тип MIME
Аудио- или видеозапись ASF	`video/x-ms-asf`
Аудио- или видеозапись WMV	`video/x-ms-wmv`
Аудиозапись AIFF	`audio/aiff`
Аудиозапись AU	`audio/basic`
Аудиозапись MIDI	`audio/mid`
Аудиозапись MP3	`audio/mpeg`
Аудиозапись WAV	`audio/wav`
Аудиозапись WMA	`audio/x-ms-wma`
Видеозапись AVI	`video/avi`
Видеозапись Indeo (IVF)	`video/x-ivf`
Видеозапись MPEG	`video/mpeg`
Визитная карточка, используемая почтовыми программами для хранения данных об адресате	`text/x-vcard`
Графический файл ART	`image/x-jg`
Графический файл BMP	`image/bmp`
Графический файл GIF	`image/gif`
Графический файл JPEG	`image/jpeg`
Графический файл Macromedia Flash	`application/futuresplash`
Графический файл TIFF	`image/tiff`
Документ Adobe Acrobat	`application/pdf`
Документ HTML	`text/html`
Документ Microsoft Excel	`application/x-msexcel`
Документ Microsoft Word	`application/msword`
Документ RTF	`application/msword`
Документ XML	`text/xml`
Приложение	`application/x-msdownload`
Приложение HTML (HTA)	`application/hta`
Таблица стилей HTML	`text/css`
Текстовый документ	`text/plain`

Дополнительные программы, расширяющие возможности Web-обозревателя, делятся на две разновидности, различающиеся принципом работы:

1. *Модули расширения* Web-обозревателя (по-английски — plugins). Впервые их начал поддерживать Netscape 2.0; в Internet Explorer поддержка их появилась в версии 3.0. Это небольшие специализированные программы, выполненные в виде динамических библиотек Windows DLL. Они загружаются самим Web-обозревателем, чтобы обработать тот или иной файл. Если нужный модуль расширения не установлен, то Web-обозреватель может сам загрузить и установить его.
2. *Элементы ActiveX.* Общий стандарт расширения для операционной системы Windows, продвигаемый фирмой Microsoft, может быть использован и для "оживления" Web-страниц. Обладая всеми возможностями модулей расширения, элементы ActiveX могут поддерживаться не только Web-обозревателями, но и всеми программами, установленными в системе. Элемент ActiveX однозначно идентифицируется с помощью *GUID* (Global Unique IDentifier — глобальный уникальный идентификатор), который вместе с поддерживаемыми типами данных MIME записывается в Реестре. Элементы ActiveX поддерживаются Internet Explorer 3.0, Navigator 6.0, Mozilla 1.0 и более новыми версиями; о поддержке их Opera данных нет.

Модули расширения

Если вы хотите поместить на Web-страницу какое-либо мультимедийное (и не только) содержание, обрабатываемое с помощью модуля расширения, вы, прежде всего, должны удостовериться, что этот модуль расширения установлен у вас на компьютере. В противном случае он не будет появляться в списке установленных модулей расширения, и вы не сможете его выбрать. Кроме того, у вас должен быть установлен Netscape Navigator версии 4.7*. Дело в том, что модули расширения считаются устаревшей технологией, и для их поддержки необходимо устаревшее программное обеспечение.

Давайте поместим на нашу Web-страницу небольшой аудиоклип в формате *WAV* (WAVe — "волна", стандартный формат хранения звуков в Windows). Откроем новое окно Dreamweaver и сохраним пустую страницу в файле 4.6.htm. В качестве файла данных мы используем небольшой аудиоклип из папки Windows\Media, например, ding.wav. Скопируйте его в папку, где хранится Web-страница 4.6.htm.

Для вставки на Web-страницу модуля расширения служит кнопка **Plugin** (рис. 4.27), расположенная на вкладке **Media** панели объектов. Вы также можете использовать пункт **Plugin** подменю **Media** меню **Insert**. После этого на экране появится диалоговое окно **Select File**. Найдите аудиофайл, который вы хотите поместить на страницу, выберите его в списке файлов и нажмите кнопку **OK**.

Рис. 4.27. Кнопка **Plugin** панели объектов

После этого в окне документа появится значок модуля расширения. Измените его размеры, как вы проделывали это с изображением. Результат показан на рис. 4.28.

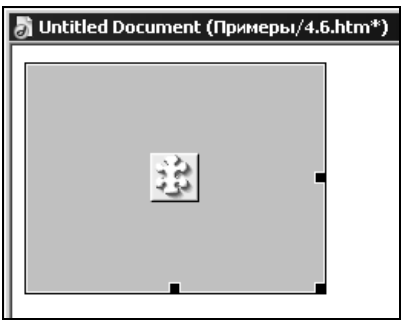


Рис. 4.28. Модуль расширения

Теперь обратим внимание на редактор свойств (рис. 4.29).

Поля ввода **W** и **H**, **H Space**, **V Space** и **Border** и раскрывающийся список **Align** должны быть вам уже знакомы. Также должна быть вам знакома кнопка **Play**/**Stop**.

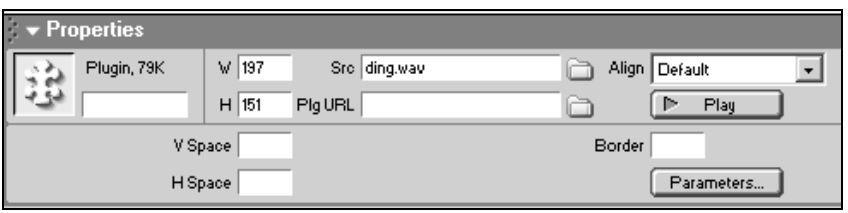


Рис. 4.29. Редактор свойств при выделенном модуле расширения

В поле ввода **Src** вводится имя файла данных. Справа от него расположена уже знакомая нам кнопка с изображением папки. Щелкнув по ней, вы получите на экране диалоговое окно открытия файла данных. С тем же успехом вы можете выбрать пункт **Source File** контекстного меню.

Поле ввода **Plg URL** используется для задания интернет-адреса файла, содержащего дистрибутивный комплект этого модуля расширения.

Кнопка **Parameters** служит для задания дополнительных параметров модуля расширения. Вместо нажатия кнопки **Parameters** вы также можете использовать одноименный пункт контекстного меню. В большинстве случаев это

не нужно, но если вы хотите, чтобы модуль расширения вел себя особым образом, вам придется задать параметры. Само собой, предварительно вам необходимо будет выяснить, какие параметры поддерживает этот модуль расширения, для чего вам придется обратиться к его документации.

При нажатии кнопки **Parameters** на экране появляется диалоговое окно **Parameters**, показанное на рис. 4.30. Большую его часть занимает список-таблица параметров и их значений. Параметры вводятся простым набором нужных значений в соответствующих колонках таблицы: имени параметра в графе **Parameter** и значения параметра в графе **Value**. Кнопка со знаком "плюс" добавляет новую строку таблицы, кнопка со знаком "минус" — удаляет текущую строку. Кнопки ↑ и ↓ сдвигают текущую строку выше или ниже.

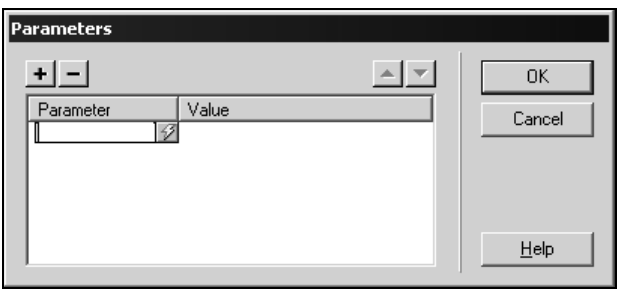


Рис. 4.30. Диалоговое окно **Parameters**

Теперь посмотрим, как работает наш модуль расширения. Для этого закроем окно задания параметров, если оно открыто, выделим модуль расширения в окне документов, если он не выделен, и нажмем кнопку **Play** редактора свойств. Модуль расширения примет такой вид — см. рис. 4.31.

Рис. 4.31. Запущенный модуль расширения

Как видите, это привычный интерфейс проигрывателя мультимедийных файлов. Подавляющее большинство программ-проигрывателей имеют именно такой вид. Здесь мы видим кнопки (перечислены в порядке слева напра-

во) остановки, запуска и приостановки проигрывания и движок-регулятор громкости воспроизведения звука. Если вы нажмете кнопку запуска проигрывания, то услышите аудиоклип.

Если вы щелкнете по модулю расширения, когда он работает, появится контекстное меню, предлагающее вам запустить (пункт **Play**), приостановить (**Pause**) или остановить совсем (**Stop**) проигрывание файла. Пункт **Save As** позволит вам сохранить файл данных под другим именем, а пункт **About** выведет небольшое диалоговое окно со сведениями о модуле расширения и его разработчике. Заметьте, что это меню выводит не Dreamweaver, а сам модуль расширения.

Внимание!

Здесь мы описали интерфейс модуля расширения, предназначенного для проигрывания мультимедийных файлов. Интерфейс других модулей расширения, разумеется, будет иным.

Закончив экспериментировать с модулем расширения, нажмите кнопку **Stop**. Вот и все! Теперь вы можете загрузить эту страничку в Web-обозревателе и проверить ее по полной программе.

Вы также можете задать в качестве файла данных видеоклип. В каталоге Windows (WinNT) имеется файл clock.avi в формате *AVI* (Audio and Video Interleaved — чередующиеся аудио и видео) — стандартном формате хранения видеоданных в Windows, который можно задать в качестве файла данных. Правда, для этого придется подогнать размеры модуля, чтобы клип влез в него целиком. Результат показан на рис. 4.32.

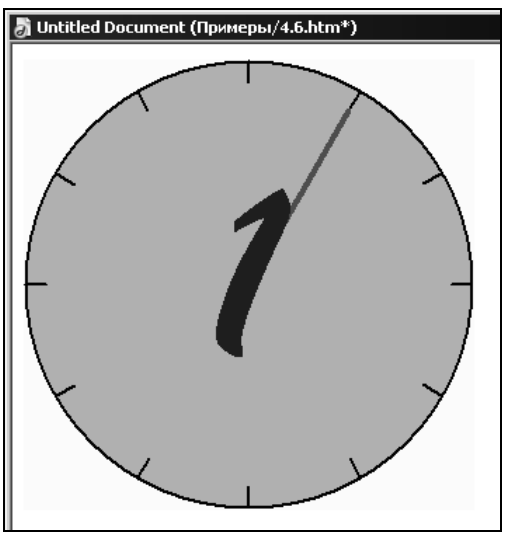


Рис. 4.32. Видеоклип на Web-странице

В этом случае в контекстном меню доступны следующие пункты:

- **Play** — запуск проигрывания клипа;
- **Rewind (Start of movie)** — перемотка в начало;
- **Forward (End of movie)** — перемотка в конец;
- **Frame Back** — перемотка на кадр к началу;
- **Frame Forward** — перемотка на кадр к концу.

Ну что ж, выберем пункт **Play** и посмотрим, что получится...

Примечание

Модуль расширения помещается на Web-страницу с помощью парного тега `<EMBED>`. Этот тег содержит множество атрибутов; в частности, атрибут `SRC` задает интернет-адрес файла данных, атрибут `PLUGINSPAGE` — интернет-адрес дистрибутива модуля расширения, атрибут `TYPE` — тип MIME данных и т. д. Дополнительные параметры задаются внутри тега `<EMBED>` набором уже знакомых вам тегов `<PARAM>`.

Элементы ActiveX

Теперь рассмотрим элементы ActiveX и их размещение на Web-страницах.

Для начала откроем новое окно Dreamweaver и сохраним новую пустую страницу в файле под именем 4.7.htm. После этого найдем файл данных. В одной из папок Microsoft Office 2000 хранится аудиофайл формата *MIDI* (Musical Instruments Digital Interface — цифровой интерфейс музыкальных инструментов) по имени Htech_01.mid, который мы и используем в качестве файла данных. Если вы не найдете этого файла, можете использовать любой другой аудио- или видеофайл, например, ding.wav или clock.avi.

Для помещения на Web-страницу элемента ActiveX можно использовать либо кнопку **ActiveX** (рис. 4.33), расположенную на вкладке **Media** панели объектов, либо пункт **ActiveX** подменю **Media** меню **Insert**. После этого в окне документа вы увидите значок элемента ActiveX (рис. 4.34). Увеличьте его размеры, как вы проделывали это с изображениями и модулем расширения.

Рис. 4.33. Кнопка **ActiveX** панели объектов

Заметьте, что запроса на открытие файла данных Dreamweaver при этом не выдает. Как привязать данные к элементу ActiveX, мы сейчас рассмотрим.

Взгляните на редактор свойств (рис. 4.35). В раскрывающемся списке **ClassID** выберите нужный элемент ActiveX. В нашем случае это будет из-

вестная программа мультимедийного проигрывателя RealPlayer — именно с ее помощью мы будем проигрывать MIDI-файл.

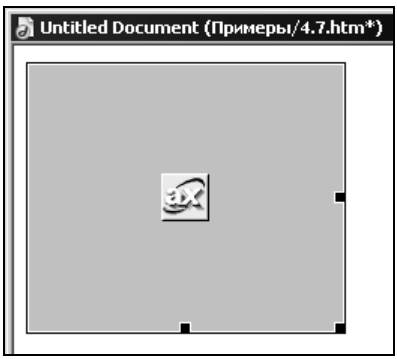


Рис. 4.34. Значок элемента ActiveX

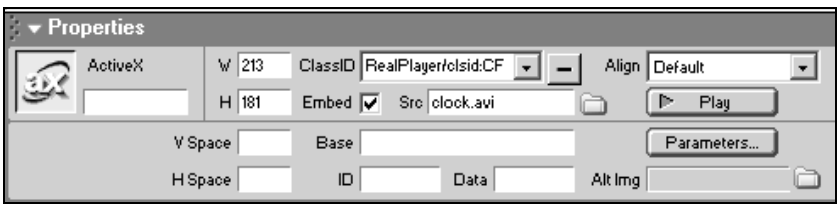


Рис. 4.35. Редактор свойств при выделенном элементе ActiveX

Найдите в редакторе свойств кнопку **Parameters**. Вы помните, зачем она нужна?

Все дело в том, что у элемента ActiveX нет собственного параметра для задания файла данных, как это было у модуля расширения. Файл данных задается в списке дополнительных параметров, вызываемых с помощью кнопки **Parameters** или одноименного пункта контекстного меню. Для этого служит параметр под названием `src`.

Откройте уже знакомое вам диалоговое окно **Parameters** (см. рис. 4.30). В единственной строке таблицы параметров в графе **Parameter** введите имя параметра — `src`, а в графе **Value** — имя MIDI-файла. И нажмите кнопку **OK**.

Теперь можно проверить в работе получившийся элемент ActiveX. Нажмите кнопку **Play**, щелкните по кнопке воспроизведения — и вы услышите музыку.

Внешний вид запущенного элемента ActiveX изображен на рис. 4.31. Не правда ли, он поразительно напоминает модуль расширения, с которым мы познакомились выше. Почему? Дело в том, что многие программы имеют несколько пользовательских интерфейсов. Один из этих интерфейсов —

обычная программа, которую пользователь запускает из меню **Пуск** или щелчком по файлу в окне Проводника. Второй и третий — соответственно модуль расширения Web-обозревателя и элемент ActiveX. И зачастую эти три интерфейса очень похожи друг на друга; это делается для того, чтобы пользователю было удобно. Программа RealPlayer, используемая нами для размещения на Web-страницах аудиоклипов, как раз имеет все эти три интерфейса.

Давайте посмотрим, что еще нам предлагает редактор свойств. Конечно, поля ввода **W**, **H**, **H Space** и **V Space** и раскрывающийся список **Align** вам уже знакомы. Рассмотрели также мы и раскрывающийся список **ClassID**. А вот остальные элементы управления пока еще не знакомы нам.

Поле ввода **Base** служит для задания интернет-адреса дистрибутивного файла элемента ActiveX. И, если нужного элемента ActiveX на компьютере пользователя не установлено, Web-обозреватель самостоятельно загрузит дистрибутив с указанного адреса и установит его. Единственное: вам нужно точно знать этот интернет-адрес.

Поле ввода **Data** служит для задания имени *файла дополнительных данных* для элемента ActiveX. (Не путайте его с файлом данных.) В настоящее время мало кто использует этот параметр.

В поле ввода **Alt Img** задается имя файла изображения, которое будет отображаться вместо элемента ActiveX, если Web-обозреватель не сможет загрузить файл или не в полной мере поддерживает технологию ActiveX. Правила хорошего тона и законы рынка требуют написания Web-страниц, совместимых с максимальным числом программ, поэтому лучше предусмотреть такое "альтернативное" изображение. Впрочем, если вы уверены, что нужды в нем не возникнет, можете не задавать — подавляющее большинство Web-дизайнеров так и делают.

Теперь обратимся к флажку **Embed** и полю ввода **Src**.

Выше мы выяснили, что элементы ActiveX поддерживаются не всеми программами Web-обозревателей. Но что делать тем, кто предпочитает пользоваться теми программами, которые как раз их не поддерживают? Выходит, те, кто помещают на свои Web-страницы элементы ActiveX, ограничивают свою аудиторию? Отчасти, да. Но Dreamweaver предлагает изящное решение этой проблемы.

Все дело в стандартах HTML, определяющих поведение Web-обозревателя, встретившего тег, которого он не "знает" *(неизвестный тег)*. Web-обозреватель должен такой тег игнорировать. Тогда можно сделать так: в код Web-страницы помещается тег элемента ActiveX, а внутри него — тег модуля расширения. Поэтому "знакомая" с элементами ActiveX программа прочитает и обработает первый тег и проигнорирует второй, а "не знакомая", наоборот, проигнорирует первый, т. к. его не "знает", зато обработает второй. Как говорится, и овцы сыты, и волки целы...

Dreamweaver автоматически создает такой "двойной" тег, если в редакторе свойств был включен флажок **Embed**. В этом случае доступно поле ввода **Src**, где задается имя файла данных для модуля расширения. Это уже нам знакомо. И еще: если флажок **Embed** включен, то поле ввода **Alt Img**, где задается "альтернативное" изображение, выводимое, если элемент ActiveX почему-то не удается запустить, становится недоступным. (Фактически, "альтернативное" изображение задается точно так же, как модуль расширения — расположение внутри тега элемента ActiveX тега `<IMG>`.)

Выше мы рассмотрели, как можно поместить на Web-страницы изображения в формате Flash — надписи и кнопки. Dreamweaver при этом решал проблему совместимости точно так же, т. е. помещением модуля расширения внутри описания элемента ActiveX. Поэтому наши надпись и кнопка будут одинаково хорошо работать в обеих популярных программах Web-обозревателей.

Примечание

Элемент ActiveX задается уже знакомым вам тегом `<OBJECT>`. Этот тег мы рассмотрели, когда говорили о помещении на Web-страницы изображений Flash. Как видите, изображения формата Flash на самом деле обрабатываются элементом ActiveX.

Что дальше?

В этой главе мы рассмотрели помещение на Web-страницы различных нетекстовых элементов. Теперь наш Иван Иванович обзавелся лицом, правда, не бог весть каким. А страничка со сведениями об авторе украсилась его портретом.

Но, кроме рисунков, важным средством представления данных являются таблицы. В самом деле, таблица — лучший способ поместить максимальное количество данных на минимальной площади. И в следующей главе мы поговорим о таблицах. А заодно создадим страницы со списками увлечений нашего воображаемого героя и завершенных им проектов.

ГЛАВА 5

Таблицы

Если вам нужно поместить на ограниченном пространстве Web-страницы множество цифровых (и не только цифровых) данных, нет лучшего средства, чем таблица. Если вам необходимо создать красивый список, снова на помощь приходит таблица. Если вам требуется точно позиционировать текст и графику относительно друг друга, опять же незаменима таблица. Таблицы заполонили Web-документы. И немудрено: при нескольких не слишком значительных недостатках они обладают массой достоинств.

Без таблиц в Web-дизайне никуда. Таблицы с разноцветными ячейками, с границами и без границ, таблицы, вложенные друг в друга, таблицы с невидимыми границами приходят на помощь, когда Web-дизайнеру нужно сделать что-либо отличное от простого "потока" текста, "разбавленного" рисунками. Но — странная ирония судьбы и высоколобых разработчиков языка HTML — стандартизированы они были совсем недавно. А это значит, что раньше, если Web-дизайнеру требовалось поместить на страницу табличные данные, он прибегал к тексту фиксированного формата. Помните старинные текстовые редакторы для MS-DOS: Лексикон, "Слово и дело", MiltiEdit? Помните, как там делались таблицы? Вот это и есть текст с фиксированным форматированием.

Но эти времена давно в прошлом. Сейчас же стандарт HTML предлагает Web-дизайнерам мощнейшие средства создания таблиц любой сложности. Главное — не запутаться в нагромождении строк и ячеек, но это уже проблема не HTML.

Создавать таблицы, вводя HTML-код вручную, очень трудоемко. Dreamweaver значительно облегчает труд Web-дизайнера, позволяя вставить в текст таблицу несколькими щелчками мыши. Точно так же просто изменить свойства как самой таблицы, так и любой ее строки или ячейки. И уж совсем просто размещать в ячейках таблицы нужное содержимое: текст, графику, другие таблицы и все, чем богат HTML и что придет в голову Web-дизайнеру.

В этой главе мы рассмотрим основные моменты работы с таблицами. Мы научимся их создавать, форматировать, размещать в них полезное содержимое, работать с ячейками и рассмотрим все их достоинства и недостатки. Но сначала давайте познакомимся с доисторическим, т. е. "дотабличным" периодом в истории HTML. А именно, выясним (или вспомним), что такое текст фиксированного формата и чем он может быть нам полезен.

Текст фиксированного формата

Как Web-обозреватель выводит на Web-страницу текст? На самом деле, здесь нет ничего сложного. Он придерживается нескольких простых правил, которые вам стоит узнать:

- символы возврата каретки преобразуются в пробелы. Строка обрывается только тогда, когда встречается тег конца абзаца `</P>` или разрыва `<BR>`;
- любое количество последовательно стоящих пробелов преобразуется в один пробел;
- текст заполняет по ширине все доступное пространство родительского элемента (страницы, абзаца и т. п.). Внутри одного абзаца строки будут иметь примерно одинаковую длину с учетом переносов.

Эти правила, которым следует каждый Web-обозреватель и которые закреплены в стандартах HTML, помогают устранить последствия некоторых ошибок Web-дизайнера. Например, следуя им, Web-обозреватель игнорирует дублирующиеся пробелы — настоящий бич современной полиграфии. Однако, если вам нужно поместить на странице текст фигурной формы, отформатированный с помощью пробелов и возвратов каретки, лучше сразу откажитесь от этой идеи.

Или прибегните к тексту фиксированного формата.

Рассмотрим такой HTML-код:

```
<HTML>
<HEAD>
<TITLE>Фигурный текст</TITLE>
</HEAD>
<BODY>
<P>
\XXXXXX\
 \XXXXXX\
  \XXXXXX\
   \XXXXXX\
    \XXXXXX\
     \XXXXXX\
```

```
</P>
</BODY>
</HTML>
```

Наберите этот текст в любом текстовом редакторе (можно и в Dreamweaver в режиме отображения кода), сохраните его в файле под именем 5.1.htm и откройте в Web-обозревателе. То, что вы увидите, показано на рис. 5.1.

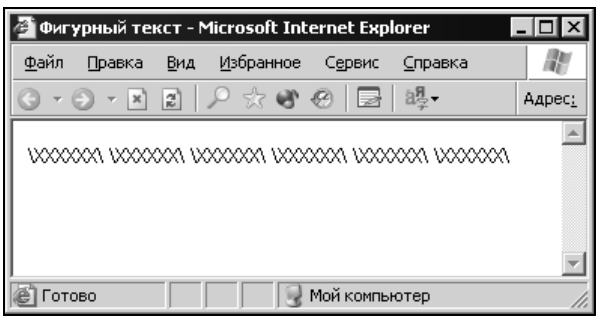


Рис. 5.1. Фигурный текст неправильно отображается Web-обозревателем

Да уж, мягко говоря, не то... Но не надо ругать Web-обозреватель: он всего лишь программа, следующая заложенному в ней алгоритму. Давайте сделаем лучше. Заменим теги <P> и </P> на <PRE> и </PRE>, соответственно. Полученный файл сохраним под именем 5.2.htm и откроем в Web-обозревателе. Посмотрите на рис. 5.2 — совсем другое дело!

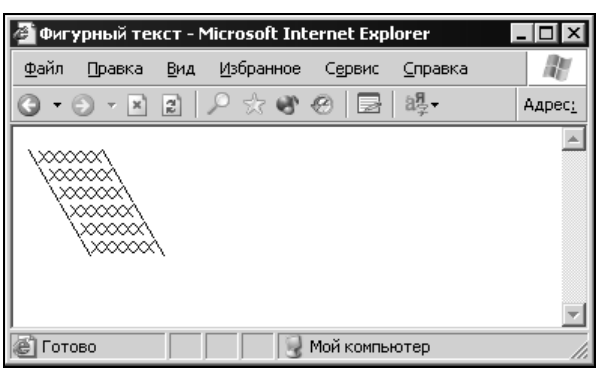


Рис. 5.2. Текст фиксированного формата отображается правильно

Web-обозреватель прекрасно знает свое дело. Ему нужно только правильно поставить задачу — и он ее решит.

Так вы поняли, что такое текст фиксированного формата? Все, что находится внутри парного тега <PRE>, Web-обозревателем переносится на страницу

без всяких изменений. Кроме того, такой текст выводится *моноширинным шрифтом* (шрифтом, символы которого имеют одинаковую ширину), а не *пропорциональным*, как обычно (символы пропорционального шрифта имеют разную ширину). Поэтому он выглядит, как в старые добрые времена MS-DOS. (Если вы, конечно, помните эти времена...)

А можно ли сделать такое в Dreamweaver? Конечно! Для этого просто наберите первую строку текста, который вы хотите превратить в текст фиксированного формата, и:

- либо выберите пункт **Preformatted** в раскрывающемся списке **Format** редактора свойств;
- либо выберите пункт **Preformatted Text** в подменю **Paragraph Format** меню **Text**;
- либо выберите пункт **Preformatted Text** в подменю **Paragraph Format** контекстного меню.

Вот и все.

В тексте с фиксированным форматированием действуют все теги HTML. Например, вы можете раскрасить такой текст в разные цвета, поместить в нем изображение или гиперссылку, как и в обычном тексте.

Выше мы говорили о таблицах. С помощью текста фиксированного формата они создаются очень просто, хотя и трудоемко. Взгляните, например, на такой код:

```
<PRE>
+----+---------------------------+----------------+
! №№ !          Проект           !    Закончен?   !
+----+---------------------------+----------------+
!  1 ! Свой домашний сайт        !       Да       !
+----+---------------------------+----------------+
!  2 ! Сайт своей любимой кошки  !   Нет, кошка   !
!    !                           !     убежала    !
+----+---------------------------+----------------+
</PRE>
```

Вы можете сохранить его в HTML-файле и загрузить в Web-обозревателе. Даже несмотря на отсутствие тегов <HTML>, <HEAD> и <BODY>, он будет нормально отображаться. Да, это таблица, хотя и корявая. Именно такие таблицы были в ходу, когда HTML еще не имел инструментов для построения настоящих таблиц, и Web-дизайнерам приходилось ломать голову, как сделать "таблицу без таблицы".

Зачем мы все это рассматриваем? Ведь это, фактически, устаревший подход. Может быть и устаревший, но в некоторых случаях весьма актуальный. В частности, если вы имеете много текстовых файлов и хотите опубликовать их в Интернете, но не желаете заниматься трудоемким HTML-формати-

рованием, то самый лучший вариант — заключить их содержимое в теги <PRE>. Так поступает Максим Мошков, когда выкладывает в свою интернет-библиотеку новые книги.

Простые таблицы

Ну все, пора переходить к настоящим таблицам. Хватит с нас корявого текста фиксированного формата!

Создание таблиц

Сначала, как обычно, создадим новую Web-страницу.

Пустую таблицу создать проще всего, нажав кнопку **Insert Table** (рис. 5.3) вкладки **Common** панели объектов (см. рис. 4.3). Также вы можете выбрать пункт **Table** меню **Insert** или нажать комбинацию клавиш <Ctrl>+<Alt>+<T>. В любом случае на экране появится диалоговое окно **Insert Table**, показанное на рис. 5.4.

Рис. 5.3. Кнопка **Insert Table** панели объектов

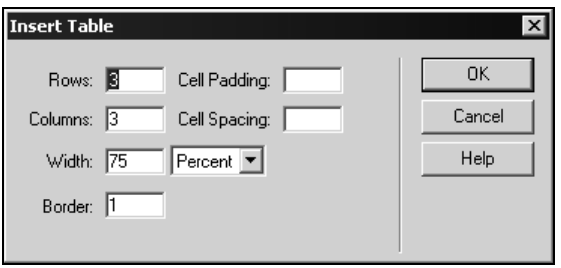


Рис. 5.4. Диалоговое окно **Insert Table**

В полях ввода **Rows** и **Columns** вводится, соответственно, количество строк и столбцов создаваемой таблицы. Если вы ошибетесь и введете большее или меньшее количество строк или столбцов, не беда — вы всегда сможете добавить их или удалить.

В поле ввода **Width** задается ширина таблицы. Возможно задание ширины как в пикселах, так и в процентах от ширины родителя. В раскрывающемся списке, расположенном справа от этого поля ввода, вы должны будете выбрать пункт **Percent** (проценты) или **Pixels** (пикселы).

В поле ввода **Border** задается толщина границ таблицы в пикселах. По умолчанию она равна 1; вы можете ввести 0, чтобы убрать границы совсем.

В поле ввода **Cell Padding** задается расстояние между границей ячейки таблицы и ее содержимым в пикселах. По умолчанию оно равно 1.

Аналогично, поле ввода **Cell Spacing** служит для задания расстояния между границами отдельных ячеек. По умолчанию оно равно 2.

Изменяя значения полей ввода **Cell Padding** и **Cell Spacing**, можно получить интересные эффекты, например огромные промежутки между границами ячеек или полупустые ячейки, в самом центре которых съежился небольшой текст. Обычно такие эффекты используют в декоративных таблицах.

Задав значения в полях ввода, нажмите кнопку **OK**. Мы не будем говорить, что именно вам нужно вводить — поэкспериментируйте сами. Так или иначе, наша первая таблица будет чисто учебной. Во всяком случае, у вас должно получиться что-то похожее на рис. 5.5. Сохраните эту таблицу в файле 5.3.htm.

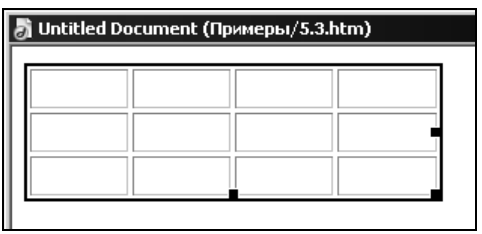


Рис. 5.5. Наша первая таблица

Рис. 5.6. Таблица с заполненными ячейками

Теперь поставьте текстовый курсор в любую ячейку таблицы и наберите какой-нибудь текст. Повторите то же самое с любыми другими ячейками.

В одну из ячеек можете вставить графическое изображение из тех, что мы использовали в предыдущих занятиях. У вас получится нечто, похожее на рис. 5.6.

Вы даже можете вставить в ячейку таблицы другую таблицу. Попробуйте — и это получится!

Кстати, даже если вы задали толщину границ таблицы равной нулю, Dreamweaver все равно будет отображать тонкую штриховую линию, чтобы помочь вам не потерять таблицу. Если эта граница вам мешает, и вы уверены, что справитесь без нее, отключите пункт-выключатель **Table Borders** подменю **Visual Aids** меню **Views**. Чтобы вернуть границы назад, просто включите этот пункт.

Работа с таблицами

Измените размер окна документа Dreamweaver, в котором находится наша таблица. Заметьте, как изменяется ширина таблицы, а все потому, что она задана относительно ширины родителя, в нашем случае окна. Если бы мы задали фиксированную ширину таблицы в пикселах, она бы не изменилась. Также вы, наверное, уже заметили, что при вводе текста в ячейки их ширина и высота изменяются, чтобы вместить текст полностью. Такие действия выполняют и Dreamweaver при создании страниц, и программы Web-обозревателей при их отображении; если размер ячейки не был жестко задан, он всегда устанавливается самой программой.

Поместите курсор на вертикальной границе между двумя ячейками. Вы увидите, что он примет форму двунаправленной стрелки, и вы сможете захватить мышью границу и переместить ее по горизонтали. Одна из ячеек станет шире, а другая — у́же. При этом им будет присвоено фиксированное значение ширины; Web-обозреватель будет изменять ширину таких ячеек только в крайнем случае, когда ему не будет хватать места для содержимого этих ячеек.

То же самое вы можете сделать и с горизонтальной границей между ячейками. В этом случае строкам, границу между которыми вы двигаете, будет присвоено фиксированное значение высоты.

Точно таким же образом вы можете изменить общую ширину таблицы. Для этого вам будет достаточно перетащить в нужную сторону крайнюю правую границу таблицы. Ну и, разумеется, Dreamweaver позволит вам изменить высоту таблицы, просто перетащив ее крайнюю нижнюю границу.

Имейте, однако, в виду, что во всех этих случаях Dreamweaver присвоит размерам таблицы, строк и ячеек (смотря, чьи размеры вы изменяли) фиксированные значения. Но, опять же, если Web-обозревателю не будет хватать места в этих ячейках, он их увеличит.

Вообще, заданные вами размеры строк и столбцов, да и размеры самой таблицы, — не более чем рекомендация Web-обозревателю. Вы говорите ему: "Если получится, сделай размер такой-то ячейки таким-то, а если не получится — задавай его на свое усмотрение". Это стандартное поведение Web-обозревателя.

Итак, с размерами разобрались. Теперь поговорим, как можно добавить новые или удалить лишние строки или столбцы.

Предположим, вам необходимо добавить новую строку или новый столбец. Для этого поместите текстовый курсор в ячейку строки, над которой будет добавлена новая, и выберите пункт **Insert Row** в подменю **Table** меню **Modify** или контекстного меню. Также вы можете нажать комбинацию клавиш <Ctrl>+<M>. Для добавления столбца поставьте текстовый курсор в ячейку, справа от которой будет добавлен новый столбец, и выберите пункт **Insert Column** в подменю **Table** меню **Modify** или контекстного меню. Если вы привыкли работать с клавиатурой, нажмите комбинацию клавиш <Ctrl>+ +<Shift>+<M>. Это простейшие и наиболее быстрые команды вставки строки или столбца.

Также вы можете воспользоваться пунктом **Insert Rows or Columns** подменю **Table** меню **Modify** или контекстного меню. После его выбора на экране появится диалоговое окно **Insert Rows or Columns**, показанное на рис. 5.7.

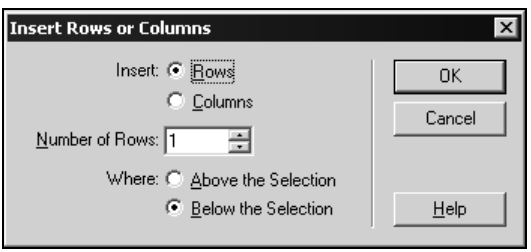


Рис. 5.7. Диалоговое окно **Insert Rows or Columns** (включен переключатель **Rows**)

С помощью переключателей группы **Insert** вы задаете тип объекта, который хотите вставить в таблицу. Переключатель **Rows** задает вставку строк, а переключатель **Columns** — столбцов.

В случае если выбран переключатель **Rows**, в поле счетчика **Number of Rows** задается количество вставляемых строк, а с помощью группы переключателей **Where** выбирается, где они будут вставляться. Переключатель **Above the Selection** вставит новые строки над *текущей строкой* (той, где стоит текстовый курсор), а переключатель **Below the Selection** — под текущей строкой.

В случае если выбран переключатель **Columns**, диалоговое окно принимает вид, показанный на рис. 5.8. При этом в поле счетчика **Number of Columns** задается количество вставляемых столбцов, а с помощью двух переключателей **Where** выбирается, где они будут вставляться. Переключатель **Before**

current Column вставит новые столбцы перед *текущим столбцом* (тем, где стоит текстовый курсор), а переключатель **After current Column** — за текущим столбцом.

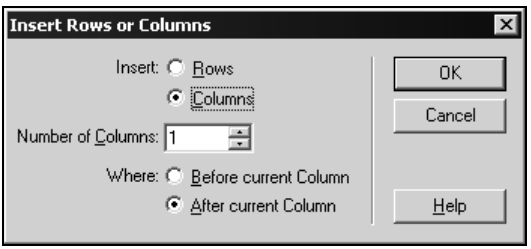


Рис. 5.8. Диалоговое окно **Insert Rows or Columns** (включен переключатель **Columns**)

Строки или столбцы будут вставлены сразу после нажатия кнопки **OK**.

У вас есть еще одна возможность вставить в таблицу новую строку или столбец — разделить текущую ячейку по горизонтали или по вертикали. В первом случае текущая строка делится на две или несколько строк, а во втором текущий столбец делится на два или несколько столбцов. Для того чтобы разделить ячейку таблицы, нажмите кнопку разделения ячейки, находящуюся в нижнем левом углу редактора свойств (рис. 5.9). Вы также можете выбрать пункт **Split Cell** в подменю **Table** меню **Modify** или контекстного меню либо нажать комбинацию клавиш <Ctrl>+<Alt>+<S>. В любом из этих случаев на экране появится диалоговое окно **Split Cell**, показанное на рис. 5.10.

Рис. 5.9. Кнопка разделения ячейки редактора свойств

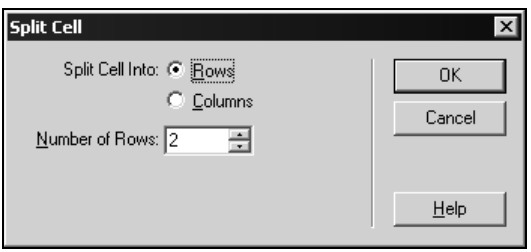


Рис. 5.10. Диалоговое окно **Split Cell**

Группа переключателей **Split Cell Into** задает, как будет делиться ячейка. Переключатель **Rows** задает разделение текущей строки на несколько строк, количество которых устанавливается в поле счетчика **Number of Rows**. Если

же выбран переключатель **Columns**, текущий столбец будет делиться на несколько столбцов, количество которых задается в поле счетчика **Number of Columns**. После нажатия кнопки **OK** текущая ячейка будет разделена.

Ненужные строки и столбцы удаляются еще проще. Для удаления текущей строки либо выберите пункт **Delete Row** в подменю **Table** меню **Modify** или контекстного меню, либо нажмите комбинацию клавиш <Ctrl>+<Shift>+<M>. Для удаления текущего столбца либо выберите пункт **Delete Column** в подменю **Table** меню **Modify** или контекстного меню, либо нажмите комбинацию клавиш <Ctrl>+<Shift>+<->.

Формирование таблиц

Обычно после рассмотрения создания какого-либо элемента страницы мы рассказываем о тегах HTML, с помощью которых он создается. Теперь пришла пора выяснить, с помощью каких тегов формируется таблица HTML. Итак, она формируется в четыре этапа. Сейчас мы их рассмотрим.

На первом этапе создается сама таблица и задаются ее параметры (ширина, толщина границы и т. п.). Это уже нам знакомо — посмотрите на рис. 5.4, где изображено диалоговое окно **Insert Table**. Все задаваемые в нем параметры и есть параметры таблицы.

Примечание

Таблица создается с помощью парного тега `<TABLE>`, имеющего множество атрибутов, из которых мы выделим `WIDTH`, задающий ширину таблицы, и `HEIGHT`, задающий ее высоту.

На втором этапе внутри таблицы создается набор описаний строк таблицы. Собственно, таблица состоит из множества строк. Каждая строка может иметь свои параметры, в частности высоту.

Примечание

Строка таблицы создается с помощью парного тега `<TR>`. Высота строки задается атрибутом `HEIGHT`. Тег `<TR>` может появляться только внутри тега `<TABLE>`.

На третьем этапе внутри каждой строки создаются описания ячеек. Каждая ячейка также может иметь параметры, например ширину. Каждая строка таблицы, таким образом, состоит из набора ячеек.

Примечание

Ячейка таблицы создается с помощью парного тега `<TD>`. Ширина ячейки задается атрибутом `WIDTH`. Тег `<TD>` может появляться только внутри тега `<TR>`.

Последний, четвертый, этап — это помещение в ячейки полезного содержимого. Запомните, что это содержимое может находиться только в ячейках

таблицы (тег <TD>), но никак не в строках (<TR>) или самой таблице (<TABLE>). Если вы нарушите это правило, HTML-код может отобразиться Web-обозревателем неверно.

Вам все это может показаться очень сложным и громоздким. Но на самом деле ничего сложного в этом нет — наоборот, все очень просто, если понять принцип. Более того, подобный способ формирования таблиц исключительно гибок, и именно в жертву гибкости была принесена компактность HTML-кода таблицы.

Давайте приведем HTML-код таблицы. Он послужит нам хорошей иллюстрацией к вышесказанному.

```
<TABLE>
  <TR>
    <TD>
      Содержимое ячейки
    </TD>
    <TD>
      Содержимое ячейки
    </TD>
  </TR>
  <TR>
    <TD>
      Содержимое ячейки
    </TD>
    <TD>
      Содержимое ячейки
    </TD>
  </TR>
</TABLE>
```

Форматирование таблиц

Ну вот, как работать с таблицами, мы разобрались. Теперь пора изучить средства, предлагаемые HTML для форматирования таблиц. Но сначала нужно выяснить, как выделить в окне документа Dreamweaver нужный элемент нашей таблицы.

Выделение элементов таблиц

Обратимся к Dreamweaver, а именно к нашей учебной страничке 5.3.htm. Откроем ее, поместим текстовый курсор в одну из ячеек и посмотрим на секцию тегов. Мы увидим там что-то, похожее на

```
<BODY><TABLE><TR><TD><P>
```

Тега <P> может и не быть (это содержимое ячейки), а на тег <BODY> мы отвлекаться не будем. Обратим внимание на уже знакомые нам теги, задающие отдельные части таблицы.

Щелкнем мышью по тегу <TD>. Ячейка, где стоит текстовый курсор, будет выделена толстой черной рамкой. Теперь вы можете производить какие-либо манипуляции с этой ячейкой, в частности задать значения ее параметров. Теперь щелкните по тегу <TR> в секции тегов. В этом случае будет выделена вся строка, в которой находится текущая ячейка. Ну и нажатие на тег <TABLE> выделяет всю таблицу.

Есть и другие способы выделить таблицу, строку или целый столбец сразу.

Чтобы выделить строку, поместите курсор мыши точно на левую границу самой левой ячейки этой строки. Курсор при этом примет вид небольшой черной стрелки, направленной вправо. Если вы щелкнете в этот момент левой кнопкой мыши, строка, напротив которой стоит курсор, будет выделена. А если вы нажмете левую кнопку мыши и, не отпуская ее, протащите мышь по вертикали, то сможете выделить сразу несколько строк.

Столбец или несколько столбцов можно выделить точно так же, только для этого курсор мыши помещается на верхнюю границу самой верхней строки таблицы. И принимает он вид черной стрелки, направленной вниз.

Вы можете также выделить одновременно несколько ячеек. Для этого щелкните мышью на ячейке, где начнется ваше выделение, и, не отпуская левой кнопки мыши, протащите ее, пока все нужные ячейки не будут выделены.

Таблицу выделить несколько сложнее. Поместите курсор на внешнюю границу таблицы и немного сдвиньте его "вовне". При этом курсор примет вид четырехконечной стрелки. Если после этого вы щелкнете мышью, таблица будет выделена целиком.

Также разработчики Dreamweaver предусмотрели еще один, самый быстрый способ выделить всю таблицу. Поставьте текстовый курсор в любую ячейку таблицы и выберите пункт **Select Table** подменю **Table** меню **Modify** или контекстного меню или просто нажмите комбинацию клавиш <Ctrl>+<A>.

После выделения таблицы на ее границе вы увидите уже знакомые вам маркеры изменения размеров. С их помощью вы можете изменить размеры таблицы.

Таким образом вы можете добраться до любой из составных частей вашей таблицы.

А теперь поговорим, что нам предлагает редактор свойств Dreamweaver.

Параметры ячейки

Чтобы добраться до параметров ячейки, совсем не обязательно ее выделять. Просто поставьте в нее текстовый курсор, и редактор свойств тотчас предложит вам задать ее параметры (рис. 5.11).

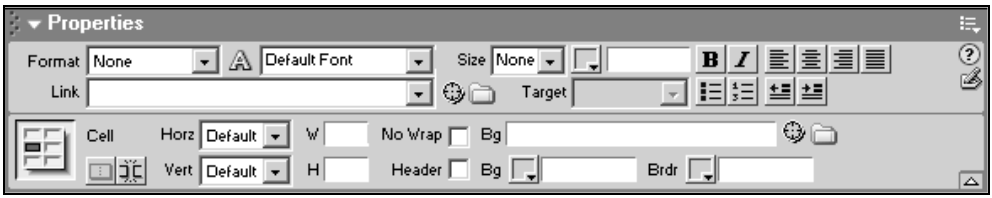


Рис. 5.11. Вид редактора свойств при выделенной ячейке таблицы

Как видите, уже знакомые нам элементы управления, находящиеся в верхней части редактора свойств, позволяют задать параметры текста. Обратимся к нижней части редактора свойств.

Раскрывающийся список **Horz** позволяет задать горизонтальное выравнивание содержимого ячейки. Здесь доступны четыре пункта:

- **Default** — выравнивание по умолчанию, обычно по левому краю, в некоторых случаях — по центру;
- **Left** — выравнивание по левому краю;
- **Center** — по центру;
- **Right** — по правому краю.

Аналогично раскрывающийся список **Vert** задает вертикальное выравнивание содержимого ячейки. Здесь доступно пять пунктов:

- **Default** — выравнивание по умолчанию, обычно посередине;
- **Top** — выравнивание по верху;
- **Middle** — посередине;
- **Bottom** — по низу;
- **Baseline** — по базовой линии.

В полях ввода **W** и **H** вводятся, соответственно, величины ширины ячейки и высоты строки, в которой она находится. Вы можете задать их в пикселах или процентах от ширины таблицы. Во втором случае просто добавьте после цифр знак процента. Если вы ничего не введете в это поле ввода, то ячейка займет все свободное пространство.

Флажок **No Wrap** запрещает перенос строк внутри ячейки. В этом случае ячейка растягивается по ширине, чтобы вместить весь имеющийся в ней текст, который в этом случае будет "вытянут" в одну строку. Обычно Web-обозреватель растягивает ячейку по ширине только в самом крайнем случае,

если в тексте встретится очень длинное слово, а вместо этого растягивает ее по вертикали и переносит текст по строкам. Используйте этот параметр только для декоративных целей и с большой осторожностью, иначе ваша таблица может оказаться столь широкой, что пользователю придется прокручивать ее по горизонтали. А слишком широкие таблицы, не помещающиеся в окне Web-обозревателя, — очень плохой стиль в Web-дизайне (рис. 5.12).

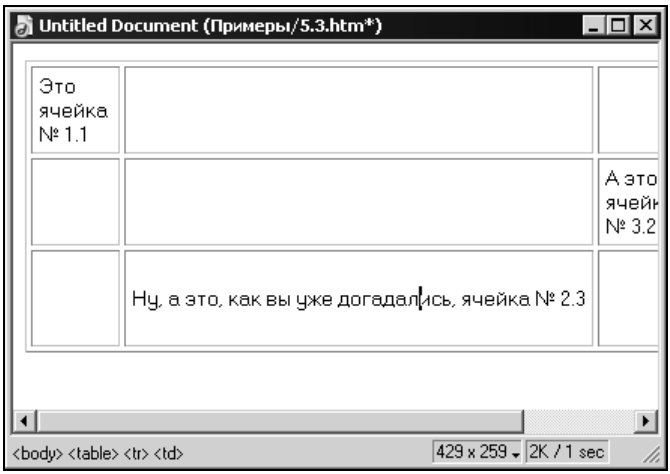


Рис. 5.12. Вид таблицы при включенном флажке **No Wrap**

Флажок **Header** позволит отформатировать ячейку, в которой находится текстовый курсор, как *ячейку заголовка* таблицы. Ячейка заголовка отличается от обычной ячейки тем, что ее содержимое выравнивается по центру и отображается жирным шрифтом. Используйте этот параметр только для ячеек первой строки таблицы, содержащей ее "шапку"; в этом случае лучше всего отформатировать таким образом все ячейки строки (например, как показано на рис. 5.13).

Примечание

Ячейка заголовка таблицы создается с помощью тега `<TH>`, аналогичного `<TD>`.

Поле ввода **Bg** служит для задания имени файла фонового изображения ячейки. Селектор цвета **Bg** (не путать с одноименным полем ввода!) позволяет задать цвет фона ячейки таблицы. Если ни то, ни другое не заданы, будет использован фон страницы.

Как видите, ячейка таблицы может иметь свое фоновое изображение и свой цвет фона, отличные от фонового изображения страницы. Это позволяет Web-дизайнерам создавать очень красивые графические эффекты: текстовые плашки, линейки и т. п. Однако, если вы собираетесь задать фоновое изо-

бражение и цвет фона для ячейки таблицы, следуйте рекомендациям, изложенным в *главе 3*.

Селектор цвета **Brdr** позволяет задать цвет границы ячейки.

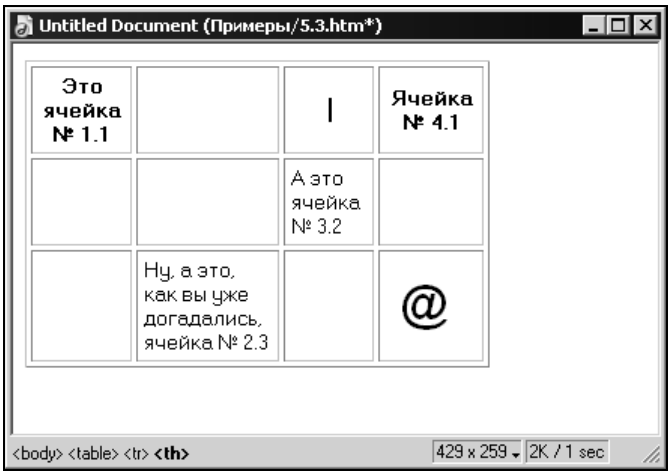


Рис. 5.13. Таблица со строкой заголовка

Параметры строки

Для строк предлагается такой же набор параметров, как и для ячеек (см. рис. 5.11). Естественно, все эти свойства применяются не к отдельной ячейке, а ко всей строке. Мы с вами не будем их подробно рассматривать, обратим внимание только на следующие замечания.

- Если ввести в поле ввода **W** какое-либо значение и нажать клавишу <Enter>, данное значение будет применено ко всем ячейкам таблицы. Используйте эту особенность, если хотите создать таблицу с ячейками одинаковой ширины. Но если вы зададите некорректное значение (например, 50% для таблицы из десяти столбцов), Web-обозреватель сам решит, какой ширины будет та или иная ячейка.
- Если включить флажок **No Wrap**, ваша таблица может расшириться до невообразимых пределов. Хорошо подумайте, прежде чем включить этот флажок.
- Если вы задали фоновое изображение (поле ввода **Bg**), цвет фона (селектор цвета **Bg**) или цвет границы (селектор цвета **Brdr**), значения этих параметров будут применены ко всем ячейкам выделенной строки. Естественно, вы можете изменить любой из названных параметров у любой ячейки. Посмотрите на рис. 5.14 — там для всей нижней строки был задан темно-серый фон, а для ячейки № 2.3 — белый.

Рис. 5.14. Результат установки разных цветов фона для всей нижней строки и ячейки № 2.3. Видно, как параметры ячейки перекрывают параметры строки

Однако разные Web-обозреватели поддерживают не все параметры, предлагаемые Dreamweaver для строк и ячеек. И ваша красивая таблица может отобразиться в них совсем не так, как вы планировали. Поэтому чем чаще вы будете справляться в электронном руководстве по HTML, какой атрибут каким тегом поддерживается, и чем чаще будете просматривать вашу страницу в разных Web-обозревателях, тем меньше у вас будет возникать проблем с совместимостью.

Параметры таблицы

Вид редактора свойств при выделенной таблице показан на рис. 5.15. Как видите, набор параметров здесь сильно отличается от набора параметров ячеек и строк. И это не удивительно.

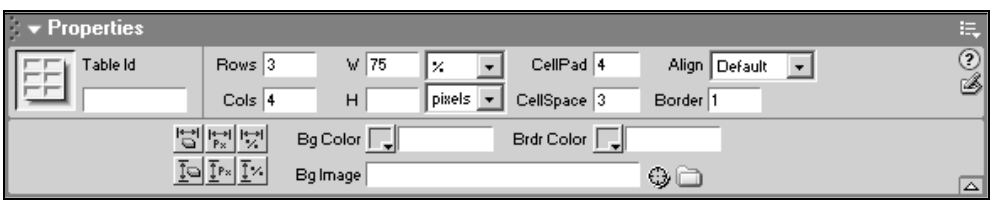


Рис. 5.15. Вид редактора свойств при выделенной таблице

Сначала поговорим о знакомом.

Поля ввода **Rows** и **Cols** служат для быстрого изменения количества, соответственно, строк и столбцов таблицы.

Внимание!

Если вы задали количество строк или столбцов меньше существующего, лишние строки или столбцы будут удалены без предупреждения.

Поля ввода **W** и **H** служат для задания, соответственно, ширины и высоты таблицы. Эти значения могут быть заданы как в пикселах, так и в процентах относительно ширины или высоты родительского элемента страницы (элемента, в котором находится наша таблица; это может быть сама Web-страница или ячейка другой таблицы). Вам нужно просто выбрать в раскрывающихся списках, находящихся справа от соответствующего поля ввода, значение **pixels** (пикселы) или **%**.

Поле ввода **Bg Image** служит для задания имени файла фонового изображения для всей таблицы. Селектор цветов **Bg Color** задает цвет фона таблицы, а селектор цветов **Brdr Color** — цвет границ таблицы. Все эти установки перекрываются аналогичными установками для ячейки (см. пример на рис. 5.16).

Рис. 5.16. Перекрытие параметров таблицы параметрами ячеек. Для ячейки № 3.2 задан светло-серый фон, для ячейки № 4.1 — белая граница, для всей таблицы — белый фон и черная граница

Поле ввода **CellPad** служит для задания расстояния между границей ячейки и ее содержимым в пикселах. Поле ввода **CellSpace** задает расстояние между границами находящихся рядом ячеек в пикселах. Поле ввода **Border** служит для задания толщины границы ячейки в пикселах. Все эти параметры мы задавали в диалоговом окне **Insert Table** (см. рис. 5.4).

Теперь рассмотрим параметры, с которыми мы еще не встречались, а если и встречались, то в приложении к другим элементам страницы.

Раскрывающийся список **Align**, как вы поняли, служит для задания горизонтального выравнивания таблицы.

Доступны четыре значения:

- **Default** — выравнивание по умолчанию, обычно влево;
- **Left** — выравнивание влево;
- **Center** — по центру;
- **Right** — вправо.

Заметьте, что с помощью этого списка задается не выравнивание содержимого таблицы, а выравнивание самой таблицы, т. е. будет ли таблица прижата к левому краю окна или центрирована в нем.

В левом нижнем углу редактора свойств есть группа из шести кнопок, управляющих значениями ширины и высоты ячеек и строк таблицы. Эта кнопки так важны, что мы приведем их на рис. 5.17 отдельно.

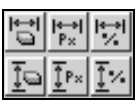


Рис. 5.17. Кнопки управления значениями ширины и высоты ячеек и строк таблицы

Часто бывает так, что после многочисленных экспериментов с установкой размеров ячеек таблицы нужно вернуться к значениям по умолчанию. В самом деле, жестко задавать размеры ячеек стоит лишь в том случае, если вы точно знаете, какого эффекта хотите достичь. В основном, это нужно лишь для дизайнерских изысков, но не для простой публикации таблично организованных данных. В последнем случае лучше все оставить как было — пусть сам Web-обозреватель разбирается, какой ширины сделать каждую ячейку, чтобы таблица поместилась в его окно.

Итак, мы уже порядком намучались, задавая фиксированные размеры ячеек, и теперь хотим стереть их, вернувшись к значениям по умолчанию. Для этого предназначены левая верхняя и левая нижняя кнопки группы. Верхняя удаляет значения ширины ячеек, а нижняя — значения высоты строк. Попробуйте выделить нашу учебную таблицу и нажать эти кнопки.

Внимание!

При этом будут сброшены также значения ширины и высоты самой таблицы.

Если вы закрыли редактор свойств или предпочитаете пользоваться меню, вы можете достичь того же эффекта, выбрав пункты **Clear Cell Heights** (стереть высоты строк) или **Clear Cell Widths** (стереть значения ширины ячеек) подменю **Table** меню **Modify**.

Теперь представим такой случай. Вы после долгих мучений выставили значения ширины ячеек и теперь довольны результатом. Но — все эти величины заданы в процентах! И теперь, если размеры таблицы изменятся, все

значения ширины ячеек "поплывут", ведь они относительны, а не абсолютны. Как быть в этом случае?

Для этого служит средняя верхняя и средняя нижняя кнопки (рис. 5.17). Верхняя из них позволяет преобразовать относительные значения ширины ячеек в абсолютные, фиксированные. А нижняя то же самое выполняет с высотами строк. Аналогичного эффекта вы можете достичь, выбрав пункты **Convert Widths to Pixels** (преобразовать ширину ячеек в пикселы) или **Convert Heights to Percents** (преобразовать высоты в пикселы) подменю **Table** меню **Modify**.

Правая верхняя и правая нижняя кнопки (рис. 5.17) выполняют обратное преобразование — абсолютных значений (пикселы) в относительные (проценты). Опять же, верхняя кнопка проделывает это с шириной ячеек, а нижняя — с высотами строк. Также вы можете воспользоваться пунктами **Convert Widths to Percent** (преобразовать ширину ячеек в проценты) или **Convert Heights to Percent** (преобразовать высоты в проценты) подменю **Table** меню **Modify**.

Предопределенные форматы таблиц

Ко всему вышесказанному можно добавить, что Dreamweaver предлагает на выбор множество *предопределенных форматов* таблиц и возможность быстро "подогнать" ту или иную схему под свои нужды. Концепция форматов таблиц схожа с концепцией предопределенных цветовых схем, рассмотренных нами в *главе 3* — они тоже созданы профессиональными дизайнерами с целью помочь начинающим пользователям.

Формат таблиц выбирается с помощью диалогового окна **Format Table**, показанного на рис. 5.18. Вызвать его можно, выбрав пункт **Format Table** в меню **Command**. Этот пункт меню доступен только тогда, когда вы выделите таблицу или хотя бы поставите текстовый курсор в ее ячейку.

В левом верхнем углу находится список, в котором представлены все доступные предопределенные форматы таблиц. Вы просто выбираете нужную строку и просматриваете получившийся результат в панели предварительного просмотра, находящемся правее списка. Расположенные ниже группы элементов управления помогут вам немного изменить предопределенный формат, если он вас не вполне устраивает.

Группа **Row Colors** позволяет задать цвета строк. Dreamweaver в этом случае предлагает вам задать цвета групп строк с чередованием (нечетная группа строк — одного цвета, четная — другого и т. п.) для лучшей читаемости больших таблиц с множеством строк. Селектор цвета **First** задает цвет нечетных групп строк, а **Second** — четных. Раскрывающийся список **Alternate** задает, сколько строк будет содержаться в каждой группе. Здесь доступно пять пунктов:

- **<do not alternate>** — все строки имеют один цвет, заданный селектором цвета **First**;

- **Every Other Row** — группы по одной строке;
- **Every Two Rows** — по две строки;
- **Every Three Rows** — по три строки;
- **Every Four Rows** — по четыре строки.

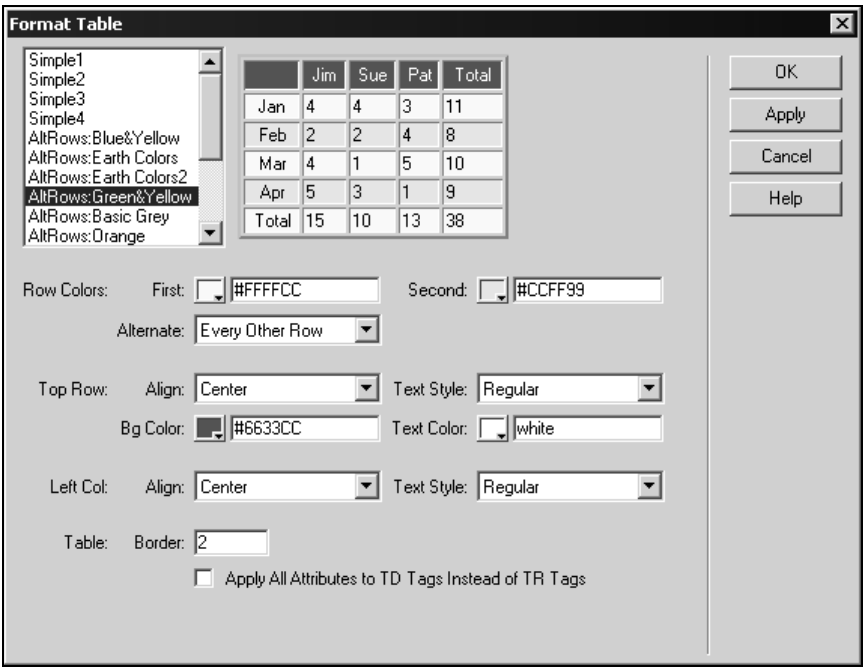


Рис. 5.18. Диалоговое окно **Format Table**

Группа **Top Row** позволяет установить параметры первой строки таблицы, ее "шапки". Раскрывающийся список **Align** задает выравнивание текста: по левому краю (пункт **Left**), по центру (**Center**), по правому краю (**Right**) или по умолчанию (**None**). Раскрывающийся список **Text Style** задает стиль текста заголовка: обычный (пункт **Regular**), жирный (**Bold**), курсив (**Italic**) или жирный курсив (**Bold Italic**). Селекторы цвета **Bg Color** и **Text Color** служат для задания цветов, соответственно, фона и текста первой строки.

С помощью группы **Left Col** устанавливаются параметры левого столбца таблицы. Раскрывающиеся списки **Align** и **Text Style** вам уже знакомы.

Поле ввода **Border** задает толщину границы таблицы в пикселах.

Обычно Dreamweaver применяет все заданные здесь настройки к строкам и ячейкам таблицы. Но если вы включите флажок **Apply All Attributes to TD Tags Instead of TR Tags**, он будет применять их только к ячейкам.

Закончив подбор формата таблицы, нажмите кнопку **OK**, чтобы закрыть окно и применить заданные параметры. Если вы нажмете кнопку **Apply**, то заданные вами параметры будут применены к таблице без закрытия окна.

Сортировка таблицы

Очень часто бывает необходимо отсортировать таблицу по значению одного или двух столбцов (более сложные сортировки по множеству столбцов встречаются значительно реже). Это случается, когда пользователь создает таблицу и вводит в нее данные из какого-то небрежно составленного документа или даже множества таких документов. А делопроизводство в большинстве наших государственных учреждений поставлено так, что навести порядок в бумажках бывает очень трудно. Если вообще возможно...

Разработчики Dreamweaver значительно облегчили операцию сортировки таблиц. Все, что вам нужно сделать, — это выделить таблицу (или просто поставить в любую ячейку текстовый курсор) и выбрать пункт **Sort Table** в меню **Commands**. На экране появится диалоговое окно **Sort Table**, показанное на рис. 5.19.

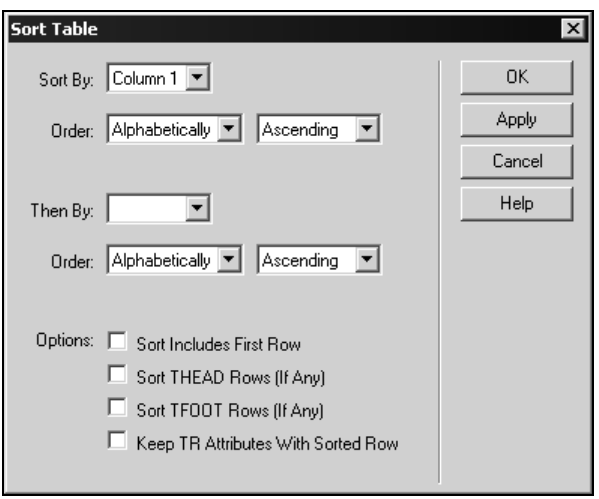


Рис. 5.19. Диалоговое окно **Sort Table**

Номер первого столбца, по значениям которого будет производиться сортировка, задается в раскрывающемся списке **Sort By**. Все имеющиеся в нем пункты имеют вид **Column** <***номер столбца***>, так что просто выберите столбец с нужным номером.

А вот группу элементов **Order**, состоящую из двух раскрывающихся списков, стоит рассмотреть подробнее. Давайте предположим, что в столбце, по

которому должна производиться сортировка, находятся одни цифры, например номера строк:

```
4
2
6
...
1
12
...
34
27
```

Как видите, они расположены в беспорядке, т. е. таблица не отсортирована. Попробуем отсортировать ее. У нас получится следующее:

```
1
10
11
...
2
20
21
...
3
```

Порядок сортировки более чем странный и, уж конечно, неправильный. Но почему?

Дело в том, что сортировка велась по *буквенному значению* символов. В таком случае каждое число Dreamweaver рассматривает как набор символов. Внимательнее присмотритесь к отсортированной последовательности — ведь с этой точки зрения она правильная.

Но мы-то ожидали, что Dreamweaver отсортирует числа по их *числовому значению*. Однако прежде ему нужно дать соответствующие указания, для чего и предназначена группа элементов **Order**. Пункт **Alphabetically** левого раскрывающегося списка задает сортировку по буквенному значению символов, а пункт **Numerically** — по числовому значению. И если мы выберем второй пункт, наша гипотетическая таблица будет отсортирована, как и требовалось:

```
1
2
3
...
10
```

```
11
...
20
21
```

Однако помните, что порядок сортировки по числовому значению следует задавать только тогда, когда в соответствующем столбце вашей таблицы находятся одни числа.

Второй раскрывающийся список группы элементов **Order** задает порядок сортировки. Пункт **Ascending** позволяет отсортировать столбец по возрастанию его значений, пункт **Descending** — по убыванию.

Раскрывающийся список **Then By** задает номер второго столбца, по которому будет производиться сортировка, если первый столбец содержит одинаковые значения. Ниже этого списка находится уже знакомая вам группа элементов **Order**.

По умолчанию Dreamweaver при сортировке таблицы не затрагивает первую строку, где обычно находится ее заголовок, иначе говоря, "шапка". Но если вы включите флажок **Sort Includes First Row**, сортировка затронет и первую строку. Включите этот флажок, если ваша таблица не имеет "шапки", а полезные данные начинаются сразу с первой строки.

Флажки **Sort THEAD Rows (If Any)** и **Sort TFOOT Rows (If Any)** позволяют отсортировать заодно и строки, находящиеся в секции *заголовка* и *"поддона"* таблицы. Эти секции создаются с помощью парных тегов <THEAD> и <TFOOT> соответственно и могут, в принципе, обрабатываться Web-обозревателем особым образом. К сожалению, Dreamweaver не поддерживает создание этих секций — вам придется вставить нужные теги вручную.

Примечание

Теги <THEAD> и <TFOOT> объединяют строки (теги <TR>), которые должны находиться в нужной секции. Для создания секции *"тела"* таблицы, объединяющей строки, содержащие полезную информацию, служит аналогичный тег <TBODY>.

Выше мы рассмотрели предопределенные форматы, позволяющие быстро форматировать таблицы. Практически во всех этих форматах нечетные группы строк окрашиваются в один цвет, четные — в другой. Это достигается заданием соответствующих параметров для строк таблицы (соответствующих атрибутов тега <TR>). При сортировке таких таблиц параметры форматирования строк (в частности, цвет фона и текста) не будут переноситься вместе с данными строк, т. е. вся их расцветка останется на месте. Это очень удобно, т. к. пользователю не нужно будет переформатировать таблицу после каждой сортировки.

Однако может случиться так, что пользователь захочет переместить вместе с данными строк их параметры форматирования. Это может быть полезно,

например, если пользователь выделяет какие-то данные в таблице отдельным цветом. Для такого случая предусмотрен флажок **Keep TR Attributes With Sorted Row**. Достаточно его включить, чтобы Dreamweaver начал переносить данные строк вместе с их форматированием.

Чтобы выполнить сортировку, нажмите кнопку **OK**. Также вы можете нажать кнопку **Apply**, чтобы выполнить сортировку, не закрывая окно **Sort Table**.

Вставка табличных данных

Да, конечно, создавать "с нуля" таблицы в Dreamweaver проще простого. Но что делать, если нужно поместить на Web-страницу таблицу, созданную в другой программе? Как Dreamweaver поможет нам в данном случае?

Поможет, и еще как! Ведь он содержит довольно мощные средства помещения на страницы табличных данных, сохраненных в текстовом формате. (Правда, при этом он коверкает русский текст, преобразуя его в нечитаемую абракадабру. Но все равно, его возможности помогут вам при помещении на Web-страницу числовых данных.) Сейчас мы их рассмотрим. Но сначала поговорим о текстовых форматах таблиц.

Текстовые форматы записи табличных данных — это довольно старая идея. Заключается она в том, что данные записываются в виде строк, а значения отделяются друг от друга каким-либо разделителем: запятой, знаком табуляции и т. п. Скажем, если мы переведем небольшую табличку, приведенную в начале этой главы, в текстовый формат, она будет иметь такой вид:

```
№№,Проект,Закончен?
1,Свой домашний сайт,Да
2,Сайт своей любимой кошки,"Нет, кошка убежала"
```

Здесь мы выбрали в качестве разделителя запятую. Но т. к. последнее значение в последней строке имеет внутри себя запятую, нам пришлось взять это значение в кавычки.

Файлы такого формата используются довольно часто и даже имеют свое название — *CSV* (Comma Separated Values — значения, разделенные запятыми). Они сохраняются с расширением csv и могут порождаться многими программами процессоров электронных таблиц и СУБД (систем управления базами данных), в частности Microsoft Excel и Microsoft Access.

Сохраним и мы нашу маленькую табличку в таком формате. Файл назовем 5.1.csv. Теперь создадим новую Web-страницу и попытаемся поместить сюда данные, только что сохраненные в CSV-файле.

Нажмите кнопку **Tabular Data** (рис. 5.20), находящуюся на закладке **Common** панели объектов. Вы также можете выбрать пункт **Import Tabular Data** под-

меню **Table Objects** меню **Insert**. При этом на экране появится диалоговое окно **Import Tabular Data**, показанное на рис. 5.21.

Рис. 5.20. Кнопка **Tabular Data** панели объектов

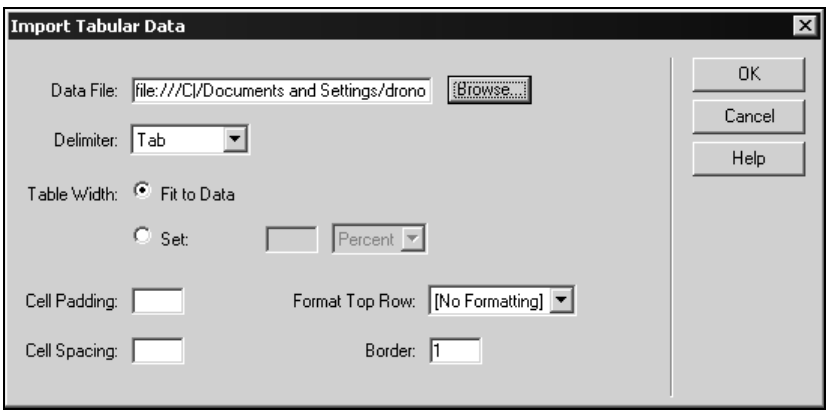


Рис. 5.21. Диалоговое окно **Import Tabular Data**

В поле ввода **Data File** вводится имя файла данных. Вы также можете нажать кнопку **Browse**, расположенную справа от этого поля ввода, и выбрать нужный файл в стандартном диалоговом окне открытия файла Windows.

Раскрывающийся список **Delimiter** позволит вам выбрать разделитель значений. В нем доступны следующие пункты: **Tab** — знак табуляции, **Comma** — запятая, **Semicolon** — точка с запятой, **Colon** — двоеточие и **Other** — другой знак. Если вы выберете последний пункт, справа от списка появится небольшое поле ввода, где вы сможете ввести символ — разделитель значений.

Группа переключателей **Table Width** позволит вам задать ширину результирующей таблицы. Установив переключатель **Fit to Data**, вы предоставляете Web-обозревателю право самому задать ширину таблицы. Переключатель **Set** позволяет вам жестко задать ширину таблицы. При этом в поле ввода, расположенном справа от этого переключателя, вы сможете ввести значение ширины. А в раскрывающемся списке, находящемся правее, задается единица измерения ширины: пикселы (пункт **Pixels**) или проценты (**Percent**).

Назначение полей ввода **Cell Padding**, **Cell Spacing** и **Border** уже вам известно. Мы рассматривали их ранее.

Раскрывающийся список **Format Top Row** позволяет вам задать, как будет форматироваться первая строка результирующей таблицы (обычно там на-

ходится заголовок таблицы). Здесь доступны четыре пункта: **[No Formatting]** — никак не форматируется, **Bold** — выводится жирным шрифтом, **Italic** — курсивом, **Bold Italic** — жирным курсивом.

Закончив установку параметров, нажмите кнопку **OK**. И посмотрите, что получится. Как видите, Dreamweaver вполне сносно справился со своей задачей. Правда, как говорилось ранее, он изуродовал русский текст и вдобавок некорректно обработал значение в кавычках — воспринял запятую внутри него как разделитель и создал еще один столбец в таблице. Но все-таки, это лучше, чем ничего. И если вам нужно быстренько "перегнать" множество таблиц с числами в формат HTML, воспользуйтесь данной функцией.

Слияние ячеек таблиц

Мы научились работать с таблицами и их составными частями. Вы можете подумать, что это все. Отнюдь! Мы еще не рассмотрели такое мощное средство, как *слияние ячеек*.

Что это такое, проще всего объяснить на примере.

Предположим, у нас есть следующая таблица (рис. 5.22). Ничего особенного, обычная таблица, какие вы уже умеете создавать в Dreamweaver. Простейший набор строк и ячеек, даже без форматирования. Мы пронумеровали ячейки таблицы, чтобы они не пустовали и чтобы нам самим в дальнейшем было легче.

1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20

Рис. 5.22. Простая таблица

Все это очень просто и прекрасно вам знакомо. Теперь давайте рассмотрим более сложную таблицу (рис. 5.23).

Здесь вы видите, что некоторые ячейки объединены, слиты в одну (об этом говорит знак "плюс" между их номерами). Такой прием называется *слиянием ячеек*. Как видите, ячейки могут быть слиты по горизонтали и по вертикали. И количество ячеек, соединяемых в одну, не ограничено.

<table>
<tr><td rowspan="2">1+6</td><td colspan="2">2+3</td><td colspan="2">4+5</td></tr>
<tr><td>7</td><td>8</td><td>9</td><td>10</td></tr>
<tr><td>11</td><td colspan="4">12+13+14+15</td></tr>
<tr><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td></tr>
</table>

Рис. 5.23. Более сложная таблица

Но как это сделать на Web-странице? Поддерживает ли HTML слияние ячеек? Конечно, поддерживает, иначе мы не стали бы вам об этом рассказывать.

Давайте откроем новое окно Dreamweaver и поместим в него простую таблицу в пять столбцов и четыре строки, аналогичную рис. 5.22. Не будем мудрить с форматированием и просто пронумеруем ячейки таблицы, чтобы было легче в ней ориентироваться (как в вышеприведенном примере). Сохраним полученную таблицу под именем 5.4.htm (рис. 5.24).

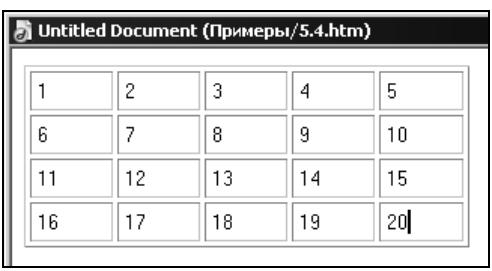


Рис. 5.24. Изначальная таблица, готовая к слиянию ячеек

Теперь давайте скажем Dreamweaver, какие ячейки мы хотим соединить. Пусть, например, это будут ячейки 2 и 3, как на рис. 5.23. Выделите их. И обратите внимание на редактор свойств. В его левом нижнем углу находится кнопка, изображенная на рис. 5.25.

Рис. 5.25. Кнопка слияния ячеек редактора свойств

С помощью данной кнопки и производится слияние ячеек. Нажмите ее. На рис. 5.26 показано, что получится в этом случае.

Рис. 5.26. Таблица, получившаяся после слияния ячеек 2 и 3

Вместо того чтобы нажимать эту кнопку, вы можете выбрать пункт **Merge Cells** подменю **Table** меню **Modify** или контекстного меню либо нажать комбинацию клавиш <Ctrl>+<Alt>+<M>.

Теперь поместите текстовый курсор в ячейку 4. Мы сольем ее с ячейкой 5. Для этого выберите пункт **Increase Column Span** подменю **Table** меню **Modify** или контекстного меню либо нажмите комбинацию клавиш <Ctrl>+ +<Shift>+<]>. Ячейки 4 и 5 сольются в одну. Это еще один способ слияния ячеек по горизонтали, самый быстрый, но позволяющий слить за один раз только две ячейки.

Остается соединить четыре ячейки — 12, 13, 14 и 15 — в одну. Выполните это одним из только что рассмотренных способов.

Как видите, все очень просто. И это заслуга Dreamweaver.

Итак, мы узнали, как выполняется слияние ячеек по горизонтали. Осталось рассмотреть, как ячейки таблицы сливаются по вертикали. В частности, нам нужно слить ячейки 1 и 6. Опять же вы можете выделить их и нажать кнопку слияния, показанную на рис. 5.25. Либо вы можете поставить текстовый курсор в ячейку 1 и выбрать пункт **Increase Row Span** подменю **Table** меню **Modify** или контекстного меню. Здесь тоже все просто.

Но что делать, если вы хотите разъединить слитые ячейки? Dreamweaver также максимально облегчает вам эту задачу. Достаточно поместить текстовый курсор в ячейку, в которую были слиты несколько ячеек таблицы, и сделать следующее:

- если выполнялось слияние по горизонтали, выбрать пункт **Decrease Column Span** подменю **Table** меню **Modify** или контекстного меню либо нажать комбинацию клавиш <Ctrl>+<Shift>+<[>;
- если выполнялось слияние по вертикали, выбрать пункт **Decrease Row Span** подменю **Table** меню **Modify** или контекстного меню.

Примечание

Слияние ячеек задается с помощью атрибутов `COLSPAN` и `ROWSPAN` тега `<TD>`. Атрибут `COLSPAN` задает слияние по горизонтали, а атрибут `ROWSPAN` — по вертикали. В качестве значения атрибута задается количество соединяемых ячеек.

Давайте приведем HTML-код нашей таблицы, чтобы понять, как работают атрибуты COLSPAN и ROWSPAN. После определений ячеек, использующих их, приведем подробные комментарии. Это пригодится, если вы захотите отредактировать полученный код вручную, да и просто для понимания принципов работы Dreamweaver:

```
<table width="75%" border="1">

  <tr>
<!-- Определяем ячейки первой строки -->
<!-- Соединяем ячейки 1 и 6 по вертикали -->
    <td rowspan="2">1+6</td>
<!-- Соединяем ячейки 2 и 3 по горизонтали -->
    <td colspan="2">2+3</td>
<!-- Соединяем ячейки 4 и 5 по горизонтали -->
    <td colspan="2">4+5</td>
  </tr>

  <tr>
<!-- Определяем ячейки второй строки. Заметьте, что здесь нет
определения ячейки 6, т. к. она уже соединена с первой
(см. определение ячеек первой строки). -->
    <td>7</td>
    <td>8</td>
    <td>9</td>
    <td>10</td>
  </tr>

  <tr>
<!-- Это третья строка. Здесь всего две ячейки: одиннадцатая и большая
ячейка, образованная слиянием ячеек 12-15. -->
    <td>11</td>
    <td colspan="4">12+13+14+15</td>
  </tr>

  <tr>
<!-- Последняя строка самая простая. Здесь вообще нет никаких
слияний -->
    <td>16</td>
    <td>17</td>
    <td>18</td>
    <td>19</td>
    <td>20</td>
  </tr>
</table>
```

Как вы уже заметили, слияния ячеек делают HTML-код таблицы крайне запутанным, а ошибки в нем — исключительно трудновыявляемыми. И куда делась былая стройность HTML-таблиц, воскликните в ужасе вы! Можно ли как-нибудь избежать этого хаоса тегов и атрибутов, определяющих слитые ячейки, этих ужасных ошибок, которые в нем таятся?

Можно. Достаточно просто обойтись без таблиц или, по крайней мере, не создавать в них слитых ячеек. Ну а уж если вам понадобится таблица, то используйте для ее построения Dreamweaver, ведь он, в отличие от Web-дизайнеров, никогда не делает ошибок в HTML-коде. Именно так поступают опытные Web-разработчики, когда им нужна сложная и громоздкая таблица. Автор настоятельно рекомендует поступать так и вам.

Использование таблиц

Выше мы говорили, что способ, принятый в HTML для формирования таблиц, исключительно гибок. Добавление строк с новыми данными в таблицу выполняется исключительно просто, даже если вы работаете непосредственно с HTML-кодом; точно так же просто можно удалить ненужные строки. Заметно сложнее добавить или удалить столбец, но здесь вам на помощь придет Dreamweaver. Платой за эту гибкость является некоторая громоздкость получающегося HTML-кода.

Стандарт HTML также позволяет вам поместить в ячейку таблицы все, что пожелаете, в том числе и другую таблицу. Поверьте, но Web-дизайнеры такое вытворяют с таблицами, что диву даешься. Многократная вложенность, всевозможные хитрости с параметрами, сложнейшие цветовые настройки, невидимые границы — просто высший пилотаж таблицестроения. В результате Web-страница становится больше похожей на газету: несколько колонок, оглавление, текст и изображения в рамках, линейки, красивый заголовок, сделанный без применения графики. И ведь все это делается довольно просто — достаточно только понять принцип.

Чтобы достичь "высшего пилотажа" таблицестроения, вам пока что не хватает знаний. В частности, знаний о некоторых параметрах и особенностях таблиц. Приготовьтесь — сейчас вы узнаете о некоторых приемах, к которым прибегают Web-дизайнеры, чтобы пустить вам пыль в глаза, и которые не очень-то раскрывают непосвященным.

Текст в рамке

Пожалуй, текст в рамке — самое простое, что можно сделать с использованием таблиц. С помощью таблицы Web-дизайнеры с легкостью обходят ограничения HTML в форматировании текстовых абзацев. (Во всяком случае, рамку вокруг абзаца штатными средствами HTML вы не сделаете.)

Итак, для страницы со списком увлечений мы хотим создать красивый заголовок с рамкой вокруг него, не используя при этом графику. Прежде всего, создадим новую Web-страницу — она и станет списком увлечений нашего гипотетического Ивана Ивановича. Сохраним ее в файле 5.5.htm. И перечислим параметры заголовка:

- собственно, текст, который мы поместим в рамку (пусть это будет просто заголовок "Список увлечений");
- расстояние между рамкой и помещенным в ней текстом (4 пиксела);
- толщина рамки (8 пикселов);
- цвет рамки (темно-синий);
- размеры рамки (ширина — 100%, высота — 50 пикселов).

Итак, казалось бы, все предельно просто. Ведь что такое, в самом деле, текст, обведенный рамкой? Фактически это таблица, состоящая из одной ячейки, где и содержится нужный нам текст. Толщина ее границы равна толщине нашей рамки, то же самое с цветом. А расстояние между рамкой и содержимым единственной ячейки и даст нам промежуток между рамкой и текстом.

Просто, не спорим. Но это как раз та простота, которая хуже воровства. Если бы все Web-обозреватели обрабатывали атрибуты тегов <TABLE>, <TR> и <TD> одинаково, мы бы на этом и остановились. Но давайте все-таки доделаем наш текст в рамке и посмотрим, что получится.

Начнем с самой рамки, т. е. с таблицы. Поместите в нашу пустую Web-страницу таблицу шириной 100% и высотой 50 пикселов с темно-синей рамкой толщиной 8 пикселов. Параметр **Cell Padding** должен быть равен 8 пикселам, а **Cell Spacing** — нулю. Эта таблица должна содержать одну строку и один столбец. То, что у вас должно получиться, показано на рис. 5.27.

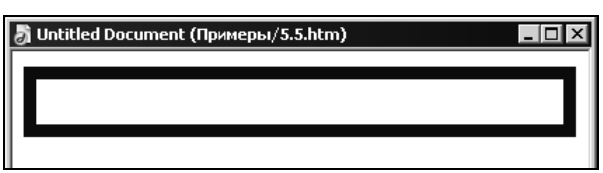


Рис. 5.27. Рамка без текста

Теперь нам осталось вписать в получившуюся рамку нужный текст. Поставьте текстовый курсор в единственную ячейку таблицы и наберите требуемый текст. Отформатируйте его как заголовок первого уровня и выровняйте по центру. Возможно, таблица немного растянется по вертикали, чтобы вместить весь текст. Можете включить флажок **No Wrap**, чтобы запретить перенос текста ячейки на другую строку. Получится нечто, напоминающее рис. 5.28.

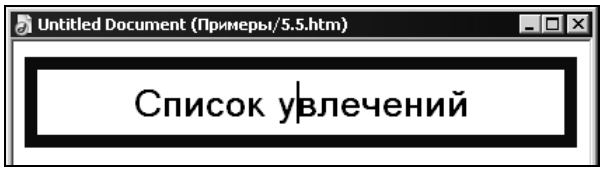


Рис. 5.28. Готовый заголовок — текст в рамке

Вот и все, вроде бы. Сохраните получившуюся страницу и откройте ее в Internet Explorer. Кажется, все нормально: Internet Explorer показывает ее так же, как и Dreamweaver. Наша работа закончена?

Нет. Уже говорилось, что разные программы Web-обозревателей обрабатывают один и тот же код HTML по-разному. И если вы откроете нашу новую страницу в старой версии Navigator... Да что говорить — смотрите сами (рис. 5.29). Рамка состоит из двух половин: одна светлая, "освещенная", другая темная, "укрывшаяся в тени". Разве мы это хотели получить?! Нет, конечно, мы ничего не имеем против таких рамок — они даже по-своему оригинальны, но мы-то хотели получить однотонную рамку!

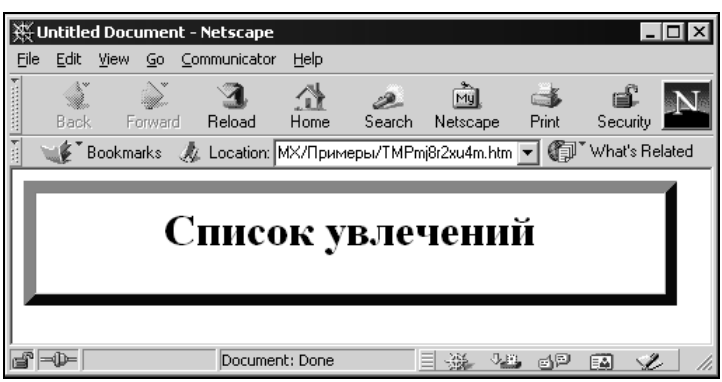


Рис. 5.29. Текст в рамке в окне Navigator 4.74.
Видно, что рамка разделена на две половины: светлую и темную

Что делать? Как заставить Navigator 4.74 отображать рамку нормально? Какой параметр таблицы или ячейки выставить?

К сожалению, ничто нам не поможет — таков уж этот Web-обозреватель... Нужно искать другой путь.

Давайте получше присмотримся к нашей рамке. В самом деле, что это такое? Текст заключен в белый прямоугольник, который находится внутри другого прямоугольника, но уже темно-синего цвета. Естественно, темно-синий прямоугольник больше белого...

Так ведь это две вложенные таблицы!

В самом деле, это так. Попробуем исправить ситуацию. Внешняя таблица имеет невидимую границу (ее толщина равна нулю), темно-синий цвет фона и параметр **Cell Padding**, равный толщине рамки, которую нам необходимо получить. Внешняя таблица имеет белый фон (это обязательно, т. к. содержимое вложенной таблицы "унаследует" фон от внешней), опять же невидимую границу и параметр **Cell Padding**, равный нужному нам промежутку между рамкой и текстом. Таким образом, мы обходим проблему с отображением границ таблицы в некоторых программах Web-обозревателях.

Давайте выполним вышесказанное. Но прежде сотрем все содержимое страницы 5.5.htm. Для этого быстрее всего щелкнуть тег `<BODY>` в секции тегов и нажать клавишу <Del>.

Создадим новую таблицу шириной 100% и высотой 50 пикселов, с темно-синим фоном и толщиной рамки, равной нулю. Параметр **Cell Padding** должен быть равен 8 пикселам, а **Cell Spacing** — нулю. Эта таблица должна содержать одну строку и один столбец. Результат показан на рис. 5.30.

Рис. 5.30. Внешняя таблица

Теперь помещаем текстовый курсор в единственную ячейку этой таблицы и вставляем в нее вторую, внутреннюю, таблицу. Ее ширина и высота должны быть равны 100% (чтобы заполнить всю ячейку), параметр **Cell Padding** — 4 пикселам, толщина границы и **Cell Spacing** — нулю. Обязательно зададим белый цвет фона. Эта таблица содержит опять-таки одну ячейку. Готовая комбинация из двух таблиц — внешней и внутренней — показана на рис. 5.31.

Рис. 5.31. Готовая рамка для заголовка, состоящая из двух вложенных одна в другую таблиц

Теперь введем и отформатируем текст заголовка — и опять получим то, что показано на рис. 5.28. Но на этот раз наше творение будет нормально показываться даже в самых капризных Web-обозревателях. Можете проверить!

Это лишь один из фокусов, к которым прибегают опытные Web-дизайнеры. К сожалению, Web-дизайн — пока что еще не столько технология и даже не

столько искусство, сколько шаманизм. А действия Web-дизайнера иной раз здорово смахивают на шаманские камлания... разве что без бубна.

Текст в графической рамке

Аппетит приходит во время еды. Нам уже кажется, что текст в обычной цветной рамке — слишком простенькое украшение для нашей странички. Мы хотим чего-нибудь покрасивее, посложнее, позабавнее.

Давайте сделаем все тот же текст в рамке. Но не в простой рамке, а в графической, т. е. для заполнения этой рамки мы используем не сплошной цвет, а графическое изображение. Это и красиво, и оригинально. Единственное: вам придется сделать рамку потолще, чтобы пользователи смогли разглядеть ваш графический фон.

Перед тем как начинать работу, необходимо приготовить заранее файл фонового изображения. Если же говорить о параметрах текста в графической рамке, то:

- текст, который мы поместим в рамку, не изменился — "Список увлечений";
- расстояние между рамкой и помещенным в нее текстом — 4 пиксела;
- толщина рамки — 40 пикселов;
- цвет рамки совпадает с общим тоном фонового изображения (в нашем случае светло-серый);
- фоновое изображение — файл Wb02049_.gif, поставляемый в комплекте Microsoft Office 2000;
- размеры рамки: ширина — 100%, высота не определена и устанавливается автоматически.

Графическая рамка для текста создается точно так же, как и цветная. За тем исключением, что для внешней таблицы мы зададим графический фон.

Сначала удалим содержимое страницы 5.5.htm. Если мы решили создать графическую рамку на странице со списком увлечений, пусть она станет окончательным вариантом ее заголовка.

Создадим внешнюю таблицу с шириной 100%, светло-серым фоном и толщиной рамки, равной нулю. Значение параметра **Cell Padding** устанавливаем 40 пикселов, а параметр **Cell Spacing** — ноль. Эта таблица должна содержать одну строку и один столбец. В качестве фонового изображения задаем подобранный файл. Полученный результат показан на рис. 5.32.

Теперь помещаем текстовый курсор в единственную ячейку этой таблицы и вставляем в нее вторую таблицу, также состоящую из одной строки и одного столбца. Ее ширина и высота должны быть равны 100% (чтобы заполнить всю ячейку), толщина границы и параметр **Cell Spacing** — 0, параметр **Cell**

Padding — 4 пиксела. Не забудьте задать для нее белый фон. Полученный результат показан на рис. 5.33.

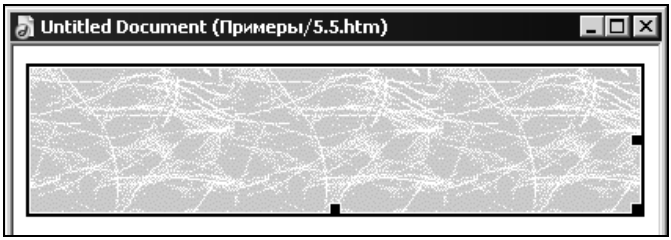


Рис. 5.32. Внешняя таблица с графическим фоном

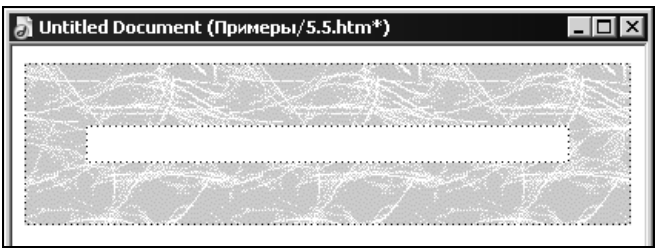


Рис. 5.33. Графическая рамка, состоящая из двух вложенных таблиц

Остается только вписать в ячейку внутренней таблицы нужный текст, отформатировать его и насладиться результатом (рис. 5.34).

Рис. 5.34. Готовый заголовок в графической рамке

Вот так мы создали красивый заголовок для нашей Web-страницы. Сохраните страницу и задайте ее параметры (цвет фона, текста и т. п.), как и у ранее созданных нами двух страниц. Ведь мы должны соблюдать единство стиля, помните?

К сожалению, здесь мы столкнемся с еще одной причудой старых версий Navigator. Давайте откроем в нем нашу Web-страницу и посмотрим, как он ее отобразит. Результат показан на рис. 5.35.

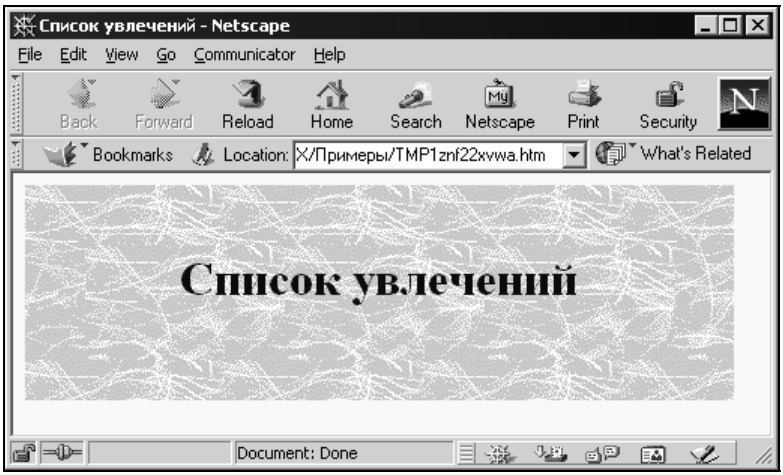


Рис. 5.35. Еще одна ошибка в старых версиях Navigator приводит к тому, что заголовок в графической рамке отображается некорректно

Из-за ошибки в программе неправильно отображается фон внутренней таблицы. Это происходит оттого, что Navigator некорректно обрабатывает установки фона вложенных таблиц. Если для вложенной таблицы задан только цвет фона, а не фоновое изображение, в то время как для внешней таблицы фоновое изображение задано, установки внутренней таблицы игнорируются.

Решение данной проблемы таково: найдите для внутренней таблицы соответствующее фоновое изображение. После этого Navigator отобразит внутреннюю таблицу правильно. Вы также можете найти или создать универсальный инструмент решения почти всех проблем Web-дизайнера — прозрачное графическое изображение размером 1×1 пиксел, сохраненное в формате GIF, так называемый *"однопиксельный GIF"*. Если вы зададите его в качестве фона внутренней таблицы, все будет отображаться в Navigator без искажений.

Но если у вас нет прозрачного GIF-изображения такого размера, не унывайте. Для того чтобы Navigator нормально отобразил внутреннюю таблицу, в режиме HTML-кода допишите в тег `<TABLE>` атрибут `BACKGROUND` с пустой строкой в качестве значения:

```
<TABLE BACKGROUND="">
  ...
</TABLE>
```

Странный, даже глупый прием, но он на самом деле действует.

Текст с отступами

Следующий интересный прием, помогающий немного оживить Web-страницу, — это размещение текста с *отступами* слева и (или) справа. Иногда слева, справа, сверху и (или) снизу помещают линейки, черные или цветные. Дизайнеры, работающие с пакетами настольных издательских систем, используют отступы и линейки так часто, что разработчики таких программ даже включили в них средства быстрого создания подобных эффектов. Однако HTML такого не поддерживает...

Хотя, почему не поддерживает? Ведь в *главе 3* мы поместили текст с авторскими правами с отступом слева. И там же мы поместили горизонтальную линию, разбив текст на логические части. Как вы помните, это достигается, соответственно, помещением текста внутрь парного тега <BLOCKQUOTE> и вставкой одинарного тега <HR>. И все это прекрасно работает!

Да, работает. Но не всегда так, как нам хотелось бы. Например, тегом <HR> вам никогда не удастся вставить в текст вертикальную линию. А величину отступа, задаваемого тегом <BLOCKQUOTE>, вы никак не сможете отрегулировать — Web-обозреватель сам выберет ее как бог на душу положит. И уж никак вы не сможете задать этим тегом отступ справа, а не слева.

Что делать? Продолжать завидовать пользователям издательских пакетов?

Нет. Есть способ лучше. Давайте подумаем, не смогут ли нам и в этом помочь таблицы.

Представим себе таблицу из одной строки и трех столбцов; таким образом, она содержит три ячейки. В средней ячейке поместим наш текст. Тогда, варьируя ширину боковых ячеек, мы сможем управлять величинами отступов слева и справа, т. е. получим то, что не может нам дать тег <BLOCKQUOTE>.

Давайте откроем страницу 5.5.htm и добавим ниже сделанного нами ранее заголовка вступление, описывающее, что находится на этой странице. Поставьте текстовый курсор правее заголовка и дважды нажмите клавишу <Enter>. После этого под заголовком будут созданы два новых абзаца, в нижнем из которых мы и поместим наш текст с отступом. Два абзаца нужны для того, чтобы между заголовком и текстом вступления было пустое пространство.

Теперь поместите на место, где сейчас находится текстовый курсор, таблицу. Параметры ее будут таковы: ширина — 100%, высота не определена, одна строка и три столбца, цвет фона не задан (будет использован фон страницы), толщина границы — 0. В среднюю ячейку впишем нужный текст и отформатируем его. Ширина боковых ячеек — 100 пикселов, а средняя ячейка пусть займет все остальное пространство (в поле ввода **W** ничего не вводите). То, что у вас должно получиться, показано на рис. 5.36. Сохраните страницу и выведите ее в окне Web-обозревателя для предварительного просмотра.

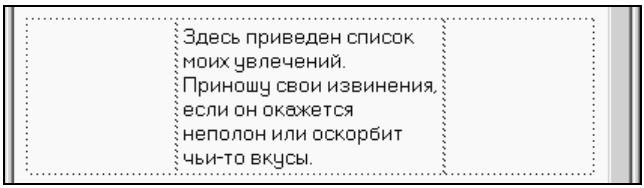


Рис. 5.36. Готовый текст с отступами

Это простейший случай текста с отступами. Теперь давайте рассмотрим случаи посложнее.

Как уже говорилось, в типографских документах часто применяются всевозможные линейки и плашки. *Линейки* — это черные или цветные линии, ограничивающие блок текста с одной или нескольких сторон. Если такие линейки ограничивают текст со всех сторон, они называются *рамками*. *Плашка* — это блок текста, чей фон отличается от белого. Верно располагая текстовые блоки, изображения, линейки и плашки и манипулируя шрифтами, дизайнер получает правильно сверстанную публикацию, не важно, типографский это дизайнер или его Web-коллега.

Как получить рамку или плашку, используя средства HTML, в основном понятно. Для плашки нужно использовать фон ячейки таблицы, где помещен текст. Чтобы получить рамку, трудно обойтись без вложенных таблиц; это мы уже рассмотрели, когда создавали заголовок для страницы со списком увлечений. Но как получить линейки?

Если вам нужна горизонтальная линейка, вы можете вставить в ячейку таблицы обычную горизонтальную линию HTML, создаваемую с помощью тега `<HR>`. Например, на рис. 5.37 мы поместили две такие линии в ячейку таблицы, где находится вступление. Мы убрали у них тень, выключив флажок **Shading** редактора свойств, — так красивее.

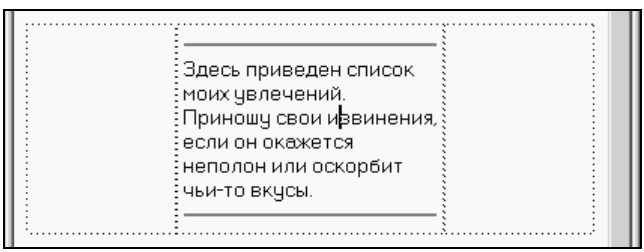


Рис. 5.37. Горизонтальные линейки в ячейке таблицы

Но что делать, если нужна вертикальная линейка? В этом случае не обойтись без модификации таблицы.

Давайте подумаем, что такое линейки. Это тонкие полосы черного (или другого) цвета, отделенные от текста некоторым пустым пространством.

А что, если в таблицу слева и справа добавить еще по две ячейки? Они должны быть достаточно узкими — всего несколько пикселов. Одна из них должна быть закрашена черным (или тем цветом, каким вы хотите сделать свои линейки), а другая вообще не должна иметь фона. Первая ячейка будет линейкой, а вторая — пустым пространством между линейкой и текстом.

Итак, пусть наши линейки имеют толщину 4 пиксела и закрашены черным. А пространство между линейкой и текстом пусть будет равно 6 пикселам.

Поместим текстовый курсор в левую ячейку нашей таблицы и нажмем кнопку разделения ячеек. В появившемся на экране диалоговом окне разделения ячеек выберем переключатель **Columns** и в поле счетчика **Number of Columns** введем 3 (т. к. хотим разделить ячейку по вертикали натрое). Поместим текстовый курсор в самую правую ячейку (она станет пустым пространством) и зададим ее ширину — 6 пикселов. Далее поместим текстовый курсор в ячейку, что находится левее (она станет линейкой), зададим ширину — 4 пиксела и черный цвет фона. Остается поставить текстовый курсор в самую левую ячейку (отступ) и вновь задать ее ширину — 100 пикселов, т. к. Dreamweaver изменил ее, когда делил одну ячейку на три. Но лучше всего задать ширину этой ячейки не 100, а 90 пикселов, с учетом толщины линейки и пространства между ней и текстом. Не стоит ужимать текст за счет декоративных элементов, пусть даже и очень красивых.

Сделали? Теперь повторите то же самое справа от текста, после чего сверьтесь с рис. 5.38, все ли правильно вы сделали.

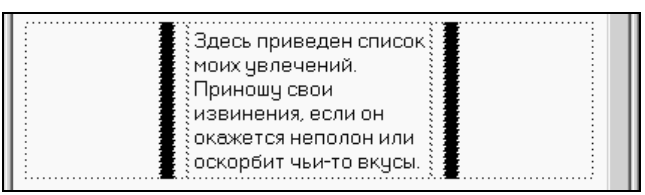


Рис. 5.38. Текстовый абзац с вертикальными линейками

Вы можете открыть эту страничку в Web-обозревателе и посмотреть, как он ее отобразит. Измените размеры окна Web-обозревателя и посмотрите, что произойдет.

Если же вы хотите сделать горизонтальную линейку, то вам придется добавить в таблицу уже не дополнительные ячейки, а дополнительные строки. И это единственное отличие — все остальное делается точно так же.

Если хотите, вы можете изменить цвет фона ячейки, в которой содержится текст, создав тем самым плашку. Вы также можете, используя вложенные таблицы, поместить этот текст в рамку. Вы уже знаете, как это делается, так что нет смысла об этом рассказывать.

Сложные таблицы

Уже было сказано, что работа Web-дизайнера — по большей части не наука (здесь вернее сказать — технология), а искусство (а зачастую — просто шаманизм). И, как в любом искусстве, здесь нет готовых путей решения той или иной проблемы. Есть только общие рекомендации и толстенные руководства по HTML и Web-обозревателям, не всегда и не во всем помогающие.

В процессе работы вы выясните, что стандарты HTML понимаются разработчиками Web-обозревателей по-своему, а сами программы Web-обозревателей содержат уйму ошибок, из-за которых правильный код отображается совсем не так, как должен. Мало того, сами средства HTML весьма ограничены и вам придется изрядно поломать голову над некоторыми особо сложными элементами дизайна.

Выход из этого положения? Вряд ли он существует... Конечно, программы Web-обозревателей постепенно приближаются к стандартам HTML, количество ошибок и несовместимостей в них уменьшается, а программы Web-редакторов становятся все "умнее" и проще в использовании. Но это не заменит обычной человеческой изобретательности. И Web-дизайн еще долго будет искусством (а зачастую — просто шаманизмом) и вряд ли когда-нибудь, даже в отдаленном будущем, полностью превратится в технологию.

Конечно, со временем к вам придет опыт, появятся свои наработки, свой стиль. Это вам сильно поможет в работе. Также очень помогает общение с другими, даже не обязательно более опытными Web-дизайнерами. Ведь давно ясно, что все сложности лучше решать сообща.

Посмотрите, например, на рис. 5.39. Это список увлечений нашего гипотетического героя. Давайте воплотим его в HTML.

Вы сразу скажете, что HTML не позволяет делать такие сложные таблицы. Да, напрямую не позволяет. Но сделать ее все-таки можно, и для этого нам придется воспользоваться все теми же вложенными таблицами.

Примечание

Вообще-то такую таблицу можно сделать стандартными средствами HTML, воспользовавшись уже знакомым вам слиянием ячеек. Но в этом случае вам не удастся сделать такую рамку, какую вы видите на рис. 5.39.

Посмотрите хорошенько на рис. 5.39. Что вы видите? В первую очередь, большую таблицу из четырех ячеек. В этих ячейках располагаются четыре вложенные таблицы, отформатированных различным образом. Две из них представляют собой секции таблицы: "Компьютеры" и "Прочие", а две другие — заголовки названных секций. Это логическая структура таблицы, и ее тоже нужно хорошо себе представлять. Но нас в данный момент больше

интересует физическая структура, описывающая, как отформатированы все эти вложенные таблицы.

Компьютеры		
Dreamweaver	Dreamweaver - мой любимый Web-редактор. Я готов сидеть за ним по двенадцать часов подряд	
Таблицы	Больше всего на свете мне нравится создавать в Dreamweaver таблицы	*Я даже детективы не читаю - в них нет таблиц*
Интернет	Ну и, конечно, без Интернета я никуда. Куда же без Интернета!	*Я сказал, что сижу за Dreamweaver по двенадцать часов в сутки. Остальные двенадцать часов я провожу в Интернете*
Прочее		
Кошка	Моя кошка - единственное живое существо, которое меня понимает	
Тараканы	Я развожу дома тараканов	*Породистых!*

Рис. 5.39. Список увлечений Ивана Ивановича Иванова

Чтобы лучше понять физическую структуру, давайте пока отложим в сторону клавиатуру и мышь и возьмем лист бумаги. Нарисуем на нем нашу большую таблицу и в каждой ее ячейке напишем параметры форматирования вложенных таблиц. У нас получится такой список:

- одна строка, один столбец, черный фон;
- три строки, три столбца, белый фон, невидимая граница;
- одна строка, один столбец, черный фон;
- две строки, три столбца, белый фон, невидимая граница.

Заметьте, что здесь перечислены только главные параметры. Остальное будет определено позднее, в самом конце. Сейчас же нам нужно выяснить для себя то, без чего настоящая таблица у нас не получится.

Итак, параметры форматирования вложенных таблиц мы выяснили. Осталось определить параметры внешней таблицы. А они таковы (напишите их на том же листе бумаги):

- четыре строки и один столбец;
- толщина границы должна быть равна нулю;

- параметр **Cell Padding** равен двум пикселам. Если мы хотим сделать видимую границу, то должны прибегнуть к приему, описанному ранее;
- цвет фона — черный.

Опять же, мы не стали выяснять малозначительные параметры. Выясним их позднее.

Казалось бы, все ясно. Пора приступать к работе над таблицей. Но не стоит торопиться. Давайте еще раз хорошенько посмотрим на нашу таблицу и перечитаем ее (и вложенных таблиц) параметры. Подумаем, что можно улучшить.

Во-первых, можно вообще не задавать для вложенных таблиц границу, т. е. приравнять ее нулю. Этим мы немного уменьшим размер HTML-кода таблиц и устраним возможные проблемы с отображением.

Во-вторых, для первой и третьей вложенных таблиц можно вообще не задавать цвет фона. В самом деле, если для содержимого таблицы не задан цвет фона, при его отображении будет использован цвет фона таблицы (или ее ячейки).

Такой процесс тонкой отладки и подгонки HTML-кода называется *оптимизацией*. Оптимизация ставит своей целью уменьшение размера кода *(оптимизация по размеру)*, уменьшение времени вывода страницы Web-обозревателем *(оптимизация по скорости)* либо то и другое одновременно. (Оптимизация применяется не только в Web-дизайне, но и в обычном программировании, однако разговор сейчас не об этом.) Если вы всерьез собираетесь заниматься Web-дизайном, старайтесь всегда найти минутку, чтобы "поколдовать" над HTML-кодом. Как правило, после оптимизации код действительно становится компактнее и быстрее.

И еще. Оптимизация — процесс творческий. Его невозможно автоматизировать, как нельзя переложить на плечи машин, скажем, стихосложение или зодчество. Конечно, "умный" Dreamweaver создает достаточно компактный код, но вмешательства человека не заменит никакая, даже самая совершенная программа. Если только, конечно, всех Web-дизайнеров не переведут в виртуальную реальность...

А вот теперь можно браться за мышь. Откроем страничку 5.5.htm, если она закрыта, и поставим текстовый курсор после текста вступления. Если вы хотите сделать промежуток между таблицей и текстом вступления больше, создайте еще один пустой абзац, нажав клавишу <Enter>. Разверните окно документа Dreamweaver на весь экран — у нас получится довольно большая таблица.

Прежде всего, разумеется, создадим внешнюю таблицу. Она будет состоять из четырех строк и одного столбца. Задайте ей черный цвет фона, невидимую границу (если вы не помните, граница становится невидимой в том случае, когда ее толщина равна нулю), параметр **Cell Padding**, равный двум

пикселам, и **Cell Spacing**, равный нулю. Ширина таблицы пусть останется 100%. После всего этого у вас должно получиться что-то, похожее на рис. 5.40.

Рис. 5.40. Внешняя таблица, которая потом станет списком увлечений

Теперь давайте создадим заголовок первой секции (первую вложенную таблицу). Поместите текстовый курсор в первую ячейку. Вставьте в нее таблицу, имеющую следующие параметры: одна строка и один столбец, невидимая граница. Ширина и высота таблицы пусть будут равны 100% — вложенная таблица должна заполнить ячейку целиком. Параметр **Cell Padding** оставьте равным двум пикселам, чтобы текст ячейки не прижимался вплотную к границе, а параметр **Cell Spacing** пусть будет равен нулю. Цвет фона не задавайте. Поместите в единственную ячейку этой таблицы текст и отформатируйте его, как вам нравится, но при этом обязательно задайте для него белый цвет, иначе пользователь ничего не увидит. Результат показан на рис. 5.41.

Рис. 5.41. Заголовок первой секции списка увлечений

Теперь очередь за первой секцией (вторая вложенная таблица). Поставьте текстовый курсор во вторую ячейку и поместите туда таблицу. Ее параметры: три строки и три столбца, опять же невидимая граница, белый фон. Ширину и высоту установите в 100%, параметр **Cell Padding** — 2 пиксела, а **Cell Spacing** — ноль. После этого заполните таблицу текстом, измените размеры ячеек так, чтобы таблица имела минимальный размер. Задайте для всех ячеек вертикальное выравнивание по верху — так ваша таблица будет больше похожа на печатный документ. Полученный результат можно видеть на рис. 5.42.

Осталось сделать заголовок второй секции и саму вторую секцию (третья и четвертая вложенные таблицы). Вы можете повторить все шаги, описанные выше, когда мы создавали первую и вторую вложенные таблицы, но это слишком трудоемко. Давайте применим другой способ, который изобрели

ленивые программисты, не желающие переписывать вручную похожие фрагменты кода.

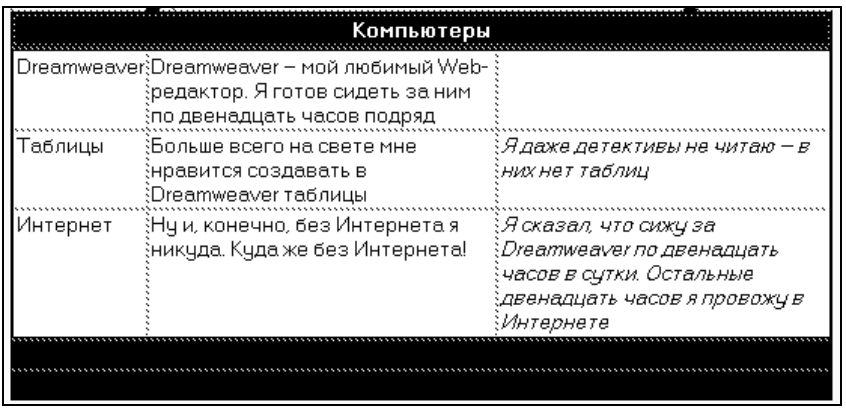


Рис. 5.42. Готовая первая секция списка увлечений

Поместите текстовый курсор в единственную ячейку первой вложенной таблицы и щелкните тег `<TABLE>` в секции тегов. Первая вложенная таблица будет выделена. Нажмите комбинацию клавиш <Ctrl>+<C>, чтобы скопировать ее в буфер обмена Windows (или сделайте это другим образом, благо Dreamweaver предоставляет целых три способа). Теперь поместите текстовый курсор в третью ячейку внешней таблицы и нажмите комбинацию клавиш <Ctrl>+<V>, чтобы вставить сюда содержимое буфера обмена. В третью ячейку будет вставлена точная копия первой вложенной таблицы; вам нужно будет только изменить находящийся в ней текст. Точно так же поместите в четвертую ячейку копию второй вложенной таблицы, удалите лишнюю строку и впишите в ее ячейки нужный текст. Окончательный вид страницы со списком увлечений представлен на рис. 5.43.

Отлично! Мы получили то, что хотели. Можете просмотреть готовую страницу в Web-обозревателе. И не забудьте сохранить плоды ваших трудов на диске.

Однако что-то в этой странице нам не нравится. Вы, наверно, знаете, что именно. Допущена эта ошибка специально, чтобы вас проверить. Если вы собираетесь серьезно заниматься Web-дизайном, вы должны видеть такие ошибки сразу.

Собственно, это даже не ошибка. Так, просто неоптимальная структура таблиц. Точнее, две лишние таблицы. Догадались, какие? Правильно, это первая и третья вложенные таблицы — заголовки секций. Посмотрите на них внимательнее — обе содержат только одну ячейку. И содержимое этой ячейки можно просто перенести в ячейку внешней таблицы без малейшего изменения внешнего вида таблицы.

Это тоже оптимизация. И на этот раз более серьезная.

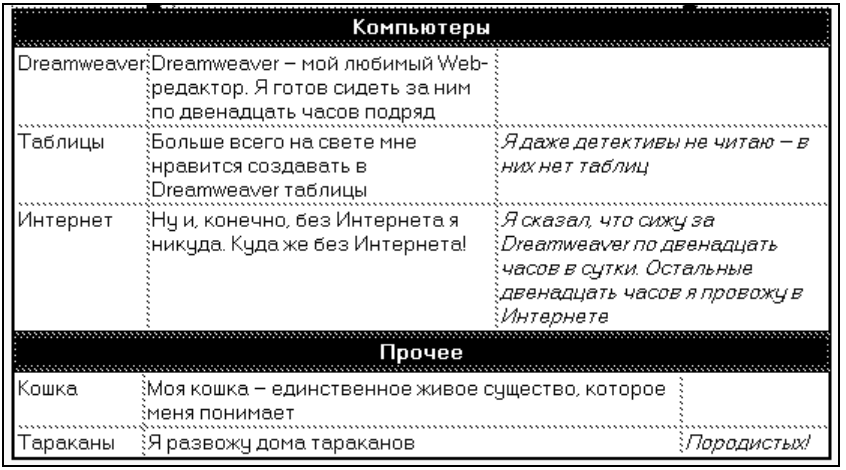


Рис. 5.43. Готовый список увлечений

Вместо того чтобы "тыкать" мышью в окно документа, мы отредактируем HTML-код. Откройте окно кода или переключитесь в режим отображения кода. Найдите следующий фрагмент:

```
<table width="100%" border="0" cellspacing="0" cellpadding="2"
↳bgcolor="#000000">
<!-- Это определение внешней таблицы -->
  <tr>
    <td>
<!-- Это первая строка внешней таблицы и ее единственная ячейка -->
      <table width="100%" border="0" cellspacing="0"
       ↳cellpadding="2" height="100%">
<!-- Это внутренняя таблица -->
        <tr>
          <td>
<!-- Это единственная строка и ячейка внутренней таблицы -->
            <div align="center">
              <font color="#FFFFFF">
                <strong>Компьютеры</strong>
              </font>
            </div>
<!-- Это текст, содержащийся в единственной ячейке внутренней таблицы -->
          </td>
        </tr>
      </table>
<!-- Здесь кончается определение внутренней таблицы -->
```

```
    </td>
  </tr>
<!-- А здесь кончается определение первой строки внешней таблицы -->
```

Сейчас мы вырежем несколько строк этого кода. Вот что получилось:

```
<table width="100%" border="0" cellspacing="0" cellpadding="2"
↳bgcolor="#000000">
  <tr>
    <td>
      <div align="center"><font
      ↳color="#FFFFFF"><strong>Компьютеры</strong></font></div>
    </td>
  </tr>
```

Как видите, мы удалили все определение внутренней таблицы, оставив только ее содержимое. Теперь переключитесь в режим отображения страницы и посмотрите, что получилось. Замечательно, нам даже ничего не нужно подправлять! Аналогично измените HTML-код, связанный с третьей ячейкой и третьей вложенной таблицей. Сделайте это самостоятельно — это будет лучшей практикой.

Все, наша оптимизация закончена! А страница списка увлечений Ивана Ивановича полностью готова.

Составные изображения

Теперь поговорим еще об одном применении таблиц. А именно о составных изображениях.

Предположим, вы создали, получили в подарок от знакомого художника или нашли в Сети красивую картину. И теперь хотите поместить ее на своей странице. Конечно, вы уже узнали, как это делается, в *главе 4*. А теперь предположим, что это не картина, а элемент оформления страницы, скажем, красивый заголовок. И этот заголовок содержит в себе несколько повторяющихся элементов. (А среди элементов оформления такое случается очень часто.)

Можно сделать так, как вы бы поступили с обычным графическим изображением, т. е. просто поместить его на страницу. А можно поступить иначе, как это часто делают опытные Web-дизайнеры: сделать составное изображение.

Что такое *составное изображение*? Ничего особенного, просто обычное изображение, но "разрезанное" на части, каждая из которых сохраняется в отдельном файле. Такие функции предлагаются всеми современными графическими пакетами, такими как Adobe Photoshop или Macromedia Fireworks.

В результате получается набор файлов — набор графических изображений-фрагментов.

Ну и что, скажете вы? Но дело в том, что изображение это режется не наобум, а таким образом, чтобы количество этих самых файлов-фрагментов было минимально. Возьмем какой-нибудь фрагмент, повторяющийся в изображении несколько раз; если изображение умело разрезать, вместо множества файлов мы будем иметь один-единственный — с этим самым фрагментом. За счет этого размеры загружаемого изображения сильно уменьшатся.

Это хорошо заметно на примере, показанном на рис. 5.44. Данный рисунок схематически показывает некое изображение, разрезанное на 30 частей. Однако, как видите, некоторые его фрагменты повторяются (их номера проставлены в ячейках), а некоторые фрагменты вообще можно исключить. Если бы мы не использовали составного изображения, общий размер загружаемой Web-обозревателем графики вырос бы в два раза. Так что выигрыш во времени загрузки налицо.

1	2	3	4	4
5	6	7	8	9
10	11	12	12	12
13	14	15	Пусто	Пусто
13	14	Пусто	Пусто	Пусто
13	15	Пусто	Пусто	Пусто

Рис. 5.44. Схема формирования составного изображения

Чтобы "собрать" воедино фрагменты изображения, используются особые таблицы, размеры ячеек которых жестко заданы и равны размерам соответствующих фрагментов. В каждую ячейку таблицы загружается свой фрагмент. Таким образом и формируется полное изображение.

В случае использования составного изображения можно организовать сложную графическую гиперссылку, аналогичную карте-изображению. Для этого в нужные ячейки таблицы просто помещаются соответствующие гиперссылки. Естественно, сначала вам необходимо нарисовать соответствующее изображение.

Проблемы с таблицами и их решение

Напоследок давайте поговорим о том, какие проблемы подстерегают Web-дизайнера, решившего использовать таблицы, и как они преодолеваются.

Общие недостатки таблиц и их преодоление

Итак, поговорим о недостатках таблиц... Да, у HTML-таблиц есть серьезные недостатки. Что ж, в мире нет ничего совершенного!..

О первом недостатке уже говорилось. Это громоздкость и запутанность HTML-кода, используемого для создания таблиц. Но это уже проблемы конкретного Web-дизайнера; в самом деле, необходимо просто аккуратно писать и тщательно документировать HTML-код, чтобы не запутаться в нем. К тому же, для создания таблиц опытные Web-дизайнеры прибегают к помощи визуальных Web-редакторов. Последние, конечно, не столь интеллектуальны и артистичны, как человек, но работают не в пример внимательнее и аккуратнее.

Второй, и самый существенный, недостаток — очень медленные загрузка и отображение. Дело в том, что многие Web-обозреватели, в частности популярнейший Internet Explorer, просто физически не в состоянии вывести таблицу на экран, пока не загрузит ее целиком. А если таблица велика, то... да что вам рассказывать?! Неужели вы не видели в Сети страничек, содержащих таблицы совершенно диких размеров? И неужели вам никогда не надоедало ждать, пока этот монстр соизволит доползти до вас полностью и явится пред ваши очи?

Примечание

Разработчики Web-обозревателей Opera и Mozilla утверждают, что эти программы могут выводить таблицы на экран по мере загрузки.

Ох уж эти гигантские таблицы... Но замедление загрузки могут вызвать и не размеры таблиц. Есть еще целый ряд причин, из-за которых Web-обозревателю может потребоваться много времени, чтобы вывести таблицу на экран. Сейчас мы все их рассмотрим. И заодно приведем кое-какие несложные рекомендации, призванные сделать загрузку и отображение ваших таблиц более быстрыми.

- Ранее уже говорилось о гигантах в мире таблиц. А рекомендации будут следующими. Прежде всего, не делайте больших таблиц. Если без этого не обойтись, то разбивайте каждую большую таблицу на несколько маленьких. Маленькие таблицы будут загружаться быстрее, и пользователь будет видеть, что Web-страница нормально обрабатывается, и сможет прочитать хоть какой-нибудь текст.

- Отображение таблицы может сильно замедлиться, если для нее или ее ячеек заданы относительные величины ширины и высоты. При этом Web-обозревателю сначала придется вычислить ширину и высоту родителя (самой Web-страницы или ячейки внешней таблицы). А точные значения ширины и высоты родителя он может получить только после того, как закончится обработка всей страницы, т. е. очень и очень нескоро. Здесь рекомендация может быть только одна: по возможности используйте абсолютные значения размеров таблиц и ячеек (но именно по возможности, а не всегда).
- Обработка и вывод Web-страницы могут сильно замедлиться, если в ней используются многократно вложенные таблицы. А если еще эти таблицы отформатированы с указанием относительных значений ширины и высоты, то... Рекомендации: тщательная проработка структуры таких таблиц и, естественно, оптимизация.

Ну вот, со скоростью загрузки и отображения таблиц закончили. Теперь давайте поговорим о причинах, по которым Web-обозреватель может отобразить таблицу неправильно. И, конечно, о решении этих проблем.

- Иногда Web-обозреватель не может корректно показать ячейку таблицы, если в ней ничего нет, т. е. ее код — `<TD></TD>`. Особенно этим страдает Navigator. Для решения данной проблемы достаточно поместить внутрь такой ячейки символ неразрывного пробела ` ` (как это делает по умолчанию Dreamweaver): `<TD> </TD>`. Правда, некоторые рекомендуют для этой же цели уже знакомый вам "однопиксельный GIF": `<TD><IMG SRC="1x1.gif"></TD>`, но об этом мы поговорим ниже.
- Если HTML-код таблицы содержит ошибки, будьте готовы, что Navigator выведет вам страницу без малейших следов этой самой таблицы. Вообще, Navigator исключительно чувствителен к ошибкам кода, в отличие от его "коллеги" Internet Explorer. Поэтому, если вы предпочитаете редактировать HTML-код таблиц вручную, тщательно проверяйте его на ошибки. А еще лучше: воспользуйтесь специальными программами проверки HTML-кода, которых сейчас довольно много на любом CD с программами. Ну и, конечно, радикальный выход из положения: воспользуйтесь для составления таблиц визуальным Web-редактором — он таких ошибок не сделает.

Проблемы с таблицами в старых версиях Navigator

А сейчас пора вплотную заняться Navigator.

Вы уже заметили, что старые версии этой программы отличаются весьма странным поведением. Мало того, что они зачастую некорректно отображают даже абсолютно правильный с точки зрения стандартов HTML-код, так еще и содержат множество ошибок, с некоторыми из них вы уже познакомились и научились их обходить. Сейчас мы продолжим разговор о не-

скончаемой войне Navigator и Web-дизайнеров, первыми жертвами которой становятся таблицы.

Примечание

Новые версии Navigator (6.0 и 7.0) выводят все Web-страницы корректно. Так что к ним это не относится. Все, что сказано ниже, относится только к старым версиям Navigator, а именно популярным до сих пор многочисленным версиям 4.7*.

Итак, какие сюрпризы продолжают преподносить нам старые версии Navigator?

Сюрприз первый. Когда мы говорили, что некоторые Web-обозреватели некорректно отображают ячейки таблиц, не содержащие данных, мы имели в виду и Navigator, и Internet Explorer. Но если Internet Explorer просто отображает такую ячейку без рамки, то Navigator может изуродовать таблицу так, что сам удивится. В качестве решения этой проблемы мы рекомендовали поместить в ячейку, которая должна оставаться пустой, символ неразрывного пробела ` `. Сам Dreamweaver делает так, когда помещает на Web-страницу пустую таблицу. Однако в случае с Navigator это не поможет — ячейка все равно будет отображена некорректно. Радикальный способ вразумить строптивую программу — "однопиксельный GIF", с которым вы уже знакомы.

Сюрприз второй. И более серьезный.

Когда вы задаете абсолютное значение ширины для ячейки, Internet Explorer и Navigator ведут себя по-разному. Internet Explorer принимает заданную Web-дизайнером ширину как руководство к действию и пытается ее соблюдать. Само собой, если содержимое такой ячейки никак не будет в нее помещаться (например, если вы включили параметр **No Wrap**, в результате чего Web-обозреватель не сможет разбить длинную строку текста на более короткие), Internet Explorer все-таки изменит ее ширину. Но это крайний случай — в остальных случаях Internet Explorer весьма послушен.

Navigator в таком случае ведет себя гораздо строптивее. Установленную Web-дизайнером ширину ячеек он принимает как рекомендации, следовать которым абсолютно не обязательно. Реальная ширина ячейки вычисляется им самим, исходя из размеров таблицы и ее остальных ячеек, и практически всегда не равна тому, что вы задали. Заложенный разработчиками беспокойный дух творчества не дает ему покоя.

Но что делать, если вы хотите, чтобы ячейки таблицы во что бы то ни стало имели нужную ширину? Есть решение, правда, довольно экстравагантное. Используется все тот же "однопиксельный GIF", из которого создается своеобразная "распорка", мешающая ячейке "сплющиться". При этом атрибут `WIDTH` тега `<TD>` (как вы помните, он задает ширину ячейки) не используется вообще, а ширина ячейки задается аналогичным атрибутом тега `<IMG>`.

Это лучше изучить на примере. Давайте рассмотрим небольшой фрагмент кода таблицы:

```
<TABLE>
  <TR>
    <TD WIDTH="50">Ячейка 1</TD>
    <TD WIDTH="100">Ячейка 2</TD>
    <TD WIDTH="200">Ячейка 3</TD>
  </TR>
</TABLE>
```

Перед вами код простейшей таблицы. Вы можете поместить его в файл, сохранить на диске и поочередно открыть в Internet Explorer и Navigator. Первый отобразит ее нормально, а второй, по своему обыкновению, будет своевольничать. Ну ничего, сейчас мы отобьем его охоту к экспериментам...

Посмотрите на этот код:

```
<TABLE>
  <TR>
    <TD>
      <IMG SRC="1x1.gif" HEIGHT="1" WIDTH="50">
      <BR>
      Ячейка 1
    </TD>
    <TD>
      <IMG SRC="1x1.gif" HEIGHT="1" WIDTH="100">
      <BR>
      Ячейка 2
    </TD>
    <TD>
      <IMG SRC="1x1.gif" HEIGHT="1" WIDTH="200">
      <BR>
      Ячейка 3
    </TD>
  </TR>
</TABLE>
```

Все очень просто! Мы поместили в каждую ячейку "однопиксельный GIF" и задали его ширину с помощью атрибута `WIDTH` тега `<IMG>`. Разумеется, ячейка таблицы растянется так, чтобы его вместить. Высоту "однопиксельного GIF" (атрибут `HEIGHT` тега `<IMG>`) мы задали в один пиксел, чтобы он не влиял на содержимое ячеек. В свою очередь, чтобы содержимое ячейки не повлияло на ширину "однопиксельного GIF", мы отделили их друг от друга тегом разрыва строки `<BR>`.

Можете проверить — все это работает!

Вообще, Navigator 4.7* славится своими странностями. Большинство из них, конечно же, давно выявлено и документировано. Если вы хотите узнать о них и научиться их обходить, посетите соответствующие Web-сайты или пообщайтесь с опытными Web-дизайнерами.

Конечно, Internet Explorer также имеет свои странности и ошибки. Но их значительно меньше, чем у Navigator, и затрагивают они, в основном, систему безопасности. Не зря же компания Microsoft при разработке Internet Explorer 6.0 во главу угла поставила именно защиту данных. И недаром Internet Explorer, согласно данным статистики, используют 95% пользователей Интернета.

Что дальше?

Осталось сделать последнюю страницу — список завершенных проектов. Ну, это уже совсем просто! Рис. 5.45 является тому примером. Не надо мудрить с оформлением, просто сделайте элементарнейшую страничку с простейшей табличкой. Но не забудьте соблюсти единство оформления и сохранить готовую страницу в файле 5.6.htm.

Рис. 5.45. Страница списка законченных проектов

Итак, почти все страницы нашего сайта готовы (осталось только сделать страницу ссылок, но это потом). Теперь самое время узнать, как соединить все эти страницы в Web-сайт и как поместить его во Всемирную сеть. И чем нам в этом может помочь Dreamweaver.

А Dreamweaver — просто мастер на все руки. Мало того, что он предоставляет нам мощные инструменты объединения Web-страниц в Web-сайт, так еще и помогает опубликовать его. А какие дополнительные инструменты дает он при этом!.. Но автор не будет сейчас раскрывать вам все его секреты — читайте следующую главу.

ГЛАВА 6

Работа с Web-сайтом

Итак, почти все страницы нашего Web-сайта созданы. Что дальше?

А дальше — последний этап работы. Что он в себя включает?

1. Связывание всех Web-страниц гиперссылками.
2. Проверку корректности ссылок.
3. Публикацию сайта в Сети.

Вот о чем мы и будем говорить в этой главе. А еще о том, чем нам в данном случае может помочь Dreamweaver.

Но сначала нужно создать еще одну Web-страницу — список ссылок на "дружественные" Web-сайты. Не будем особо мудрить и поместим на нее ссылку на "домашний" сайт Macromedia (рис. 6.1). Отойдем от нашей обычной системы именования файлов Web-страниц и сохраним полученный файл под именем Links.htm.

Теперь все страницы готовы. Пора приступать к работе над сайтом.

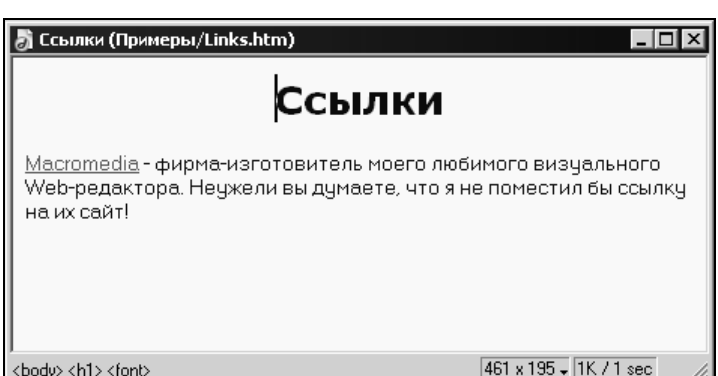


Рис. 6.1. Страница со списком ссылок

Начала сайтостроения

Но прежде чем мы начнем создавать наш первый настоящий сайт, выясним, что же такое Web-сайт и как он публикуется в Сети. В *главе 1* мы немного коснулись этих вопросов, а сейчас рассмотрим их более подробно.

Планирование сайта

Каждое дело начинается с планирования. В самом деле, прежде чем что-то делать, нужно твердо уяснить, что же мы хотим получить в результате. И спланировать это нужно в самом начале, перед тем, как приниматься за дело. Ведь когда дело сделано, менять что-либо значительно труднее, чем сразу делать все как надо.

Когда мы начинали создавать наши страницы, мы не задумывались о том, что когда-нибудь объединим их в сайт, и, соответственно, не спланировали его. Конечно, поскольку наш будущий сайт очень прост и состоит всего из пяти Web-страниц, для нас это не очень страшно. Но основные этапы планирования сайта никто не отменял. И сейчас мы их рассмотрим.

Основные этапы планирования сайта

Вообще, этап *планирования* — важнейший в разработке любого сайта. Это справедливо и для простейшей домашней странички, и для гигантского сайта транснациональной суперкорпорации, такой как Intel или Coca-Cola. Поэтому, начиная планирование будущего сайта, лучше закройте Dreamweaver и даже выключите компьютер, чтобы не возникло соблазна сразу же засесть за "ваяние". На этом этапе вам понадобятся только карандаш и бумага или, если вы все-таки предпочтете компьютер, программа текстового редактора или специальный пакет проектирования сайта, наподобие Microsoft Visio.

Итак, что же нужно решить для себя перед началом работы над сайтом?

1. Прежде всего, четко определить задачи сайта. Что он должен делать: рассказывать о ком-либо или о чем-либо, привлекать клиентов, помогать решать какие-то проблемы, просвещать или развлекать. В зависимости от задач структура сайта может сильно различаться.
2. Определить, какая конкретно информация должна на сайте присутствовать. Что вы хотите рассказать о себе или что вам нужно рассказать о заказчике. И ничего лишнего!
3. Собрать эту информацию, чтобы не искать ее потом, во время работы над сайтом, когда у вас будут совсем другие заботы. Все тексты, изображения, файлы, которые вы намерены выложить в Сеть, должны быть на вашем компьютере. Только так — и никаких ссылок на собственное разгильдяйство!

4. Решить, в каком ключе будет выполнен дизайн сайта. Будет ли он консервативным, строгим или затейливым. Соответственно, домашняя страничка должна отражать эстетические наклонности автора, рекламный сайт лучше сделать повеселее, а новостной — поскромнее, чтобы пестрота дизайна не заслоняла главное — информацию. На этом этапе лучше всего будет набросать на бумаге, как должна выглядеть та или иная страница.
5. Придумать логическую структуру сайта и — желательно — нарисовать ее. Здесь лучше не изобретать самому велосипед, а посетить какой-нибудь уже существующий и популярный Web-сайт и посмотреть, как он организован. Например, для домашнего сайта идеальна такая структура: начальная ("домашняя") страница с краткими сведениями о хозяине, приглашением посетить другие страницы сайта и набором ссылок на них; а на других страницах размещаются информация об увлечениях, проектах, разработках (если это программист, музыкант, художник, то список ссылок на файлы программ, аудиоклипов или картин), фотогалерея и странички с набором "дружественных" ссылок и более подробными сведениями об авторе, с почтовым адресом и фотографией.
6. Придумать физическую структуру сайта, т. е. как отдельные файлы, составляющие сайт, будут "раскиданы" по папкам.
7. Проверить, ничего ли вы не забыли. Это последний этап планирования сайта, но не менее важный, чем остальные.

Вот и все основные этапы проектирования сайта. Теперь осталось рассмотреть подробнее, что такое логическая и физическая структуры сайта.

Логическая структура Web-сайта

Итак, что мы сейчас имеем? Несколько разрозненных Web-страниц и сопутствующие им файлы графических изображений. Именно разрозненных страниц — они никак не связаны друг с другом. А важнейшим признаком Web-сайта является как раз тесная взаимосвязь отдельных его страниц. И не просто взаимосвязь, а взаимосвязь в соответствии с разработанной заранее структурой. Такая структура, называемая *логической*, описывает взаимосвязь различных страниц сайта.

Каким же образом связываются друг с другом страницы? Собственно, это уже зависит от того, как организована на сайте информация. То есть для каждого отдельного сайта нужно придумывать свою структуру. Конечно, есть общие принципы структуризации сайта, которым нужно следовать всегда. Сейчас мы их и рассмотрим. Вот примерный план хорошо продуманного сайта:

```
Заставка
Главная страница
```

```
Новости сайта
Раздел 1
  Страница 1
  Страница 2
  . . .
Раздел 2
  Страница 1
  Страница 2
  . . .
. . .
Сведения о разработчиках
Контактные данные
Карта сайта
```

Теперь поговорим о каждой составляющей сайта более подробно.

Заставка — это небольшая Web-страничка, появляющаяся на экране, как только пользователь набирает "домашний" адрес сайта, перед его главной страницей. Обычно на такой страничке помещается графический логотип сайта или его владельца. Часто такой логотип выполнен в виде фильма в формате Macromedia Flash. Поскольку такая заставка может загружаться очень долго, нужно предусмотреть небольшую, но заметную гиперссылку, щелкнув на которой посетитель сразу перейдет на главную страницу сайта.

Заставка — необязательный элемент сайта. Подавляющее большинство сайтов не содержат заставок, а те, которые их имеют, — обычно развлекательные или рекламные сайты. Подумайте, прежде чем создавать заставку для своего сайта, нужна ли она вам, подходит ли вашему сайту по концепции. Помните, что в большинстве случаев пользователь не дожидается окончания загрузки заставки и переходит дальше.

Главная страница отображается, когда пользователь набирает "домашний" адрес сайта без указания имени файла какой-либо страницы (например, **http://www.macromedia.com**). Она содержит краткую вводную информацию о сайте, новости (необязательно) и набор гиперссылок, ведущих на другие страницы сайта. Также иногда на главной странице помещаются сведения о разработчиках и их правах и сведения о контакте с разработчиками и другими лицами и организациями, упомянутыми на сайте.

Главная страница — обязательный элемент любого сайта. Как правило, главную страницу стремятся делать не слишком большой, чтобы посетитель не ушел с сайта, не дождавшись ее загрузки. Но не следует впадать в другую крайность — делать главную страницу настолько "спартанской" по содержанию, что посетитель не сможет даже понять, куда он попал. Запомните, что главная страница должна давать посетителю достаточно информации о сайте, но при этом не перегружать его излишними сведениями и не выводить из себя ожиданием окончания загрузки. А это довольно сложная задача, и

даже известные ресурсы часто страдают катастрофическим "раздутием" главной страницы.

Главная страница нашего сайта — 3.1.htm. Давайте переименуем ее в default.htm — такое название чаще всего дают главным страницам.

Новости сайта часто помещают на отдельной странице. Они представляют собой хронологический список всех дополнений и обновлений, сделанных на сайте. Как правило, выводятся только новости за некоторый период (месяц, квартал, год, в зависимости от того, насколько часто обновляется сайт). Для доступа к более старым новостям ("старостям") предусматривается так называемый *архив новостей*, на который ведет специальная гиперссылка.

Иногда, правда, новости помещают на главной странице. Этот подход тоже оправдан: посетитель сайта сразу видит, что на нем изменилось. Но, опять же, не стоит перебарщивать с объемом главной страницы, а новости зачастую занимают много места. В общем, решайте сами, куда поместить новости вашего сайта: на отдельную страницу или в особый отдел главной. Здесь уместно дать еще один совет: если вы поместили новости сайта на отдельной странице, каким-либо образом выделите гиперссылку, ведущую на эту страницу. В этом случае постоянному посетителю вашего сайта не надо будет долго ее искать, чтобы выяснить, какие изменения произошли на сайте во время его отсутствия.

Иногда, если сайт обновляется совсем редко или имеет небольшой объем, новости вообще не предусматриваются. Как, например, на нашем сайте.

Полезное содержимое сайта — это та информация, ради которой он был создан. Структурируется она так же, как в книге: отдельные абзацы, посвященные какой-либо теме, объединяются в главы, а главы в свою очередь — в более крупные единицы, разделы. Таким образом, посетитель сайта сразу сможет найти нужную информацию, двигаясь от разделов к главам, а от глав — к абзацам, пока не найдет то, ради чего сюда пришел.

Полезное содержимое нашего сайта — это страницы 5.4.htm, 5.5.htm и Links.htm. Переименуем первые две страницы в Passions.htm и Projects.htm соответственно.

Сведения о разработчиках могут помещаться как на отдельной странице, так и в особом отделе главной. Если разработчиков немного (или вообще один), более предпочтителен второй вариант. В таком случае сведения о них помещаются в самом низу главной страницы, рядом со сведениями об авторских правах. Если же разработчиков много или сведения о них достаточно объемные, лучше поместить их на отдельную страницу. Обязательно при этом укажите адрес электронной почты, по которому посетитель сайта сможет написать о проблемах, с которыми он столкнулся (незагружающиеся файлы, "пустые" изображения, "оборванные" ссылки, ошибки в тексте и т. п.). Иногда на странице вместе с остальными данными также поме-

щаются фотографии разработчиков, но это уже, как говорится, на вкус и цвет.

Сведения о разработчике нашего сайта помещены на странице 4.1.htm. Переименуем ее в About.htm.

Сведения о контакте с владельцем сайта нужны, если данный сайт преследует рекламные цели. Например, если это торговый сайт, необходимо обязательно указать контактные данные, иначе никто из потенциальных клиентов ничего не сможет купить. В этом случае необходимы адреса как обычной ("бумажной" или, как говорят американцы, snail mail — "улиточной почты"), так и электронной почты, а также телефон, факс и пейджер. В общем, все данные, по которым могли бы обратиться потенциальные клиенты.

Так как наш сайт не принадлежит к торговым, более того, он некоммерческий, сведений для контакта он не содержит. Хотя Иван Иванович мог бы разместить на нем список своих профессиональных достоинств, чтобы потенциальные работодатели смогли сразу их оценить. Но он не догадался сделать этого, а мы за него не будем этим заниматься.

Карта сайта — это страница, на которой изображена вся логическая структура сайта, показанная нами выше. Карта сайта служит для того, чтобы посетитель, точно знающий, что ему нужно, но не желающий продираться через иерархию ссылок, мог сразу добраться до необходимой информации. Карта помещается на всех достаточно больших сайтах.

Так как наш сайт невелик по размерам, карты у него нет.

Физическая структура Web-сайта

Физическая структура сайта описывает размещение файлов и папок на диске компьютера.

Когда мы сохраняли все файлы Web-страниц и связанных с ними графических изображений в одну папку, мы не думали о какой-то там физической структуре. И правильно! В подавляющем большинстве случаев такой подход приемлем для небольших сайтов, вроде нашего. В таких случаях можно не ломать себе голову, в какой папке поместить тот или иной файл, — лучше сосредоточиться на содержании. Все равно, если сайт состоит из двух-трех десятков файлов, от сложной физической структуры вы не получите никаких преимуществ.

Но когда количество файлов вырастает до сотни, лучше потратить время и разложить их по полочкам, то бишь по папкам.

Правильная организация файлов и папок файловой системы поможет вам в дальнейшем держать все это хозяйство в порядке. В самом деле, намного проще найти GIF-файл среди нескольких десятков других GIF-файлов, лежащих в отдельной папочке, чем рыться среди сотен разнородных файлов,

сваленных как попало в одну папку. К тому же, вам будет проще обновлять файлы сайта, если вы "разбросаете" их по отдельным папкам.

Существует своего рода типовая физическая структура, которой более или менее строго придерживаются все Web-дизайнеры. Она представлена в следующем листинге. Имена папок отличаются от имен файлов тем, что первые не имеют расширения. В квадратных скобках дается необязательный текст.

```
<Корневая папка вашего сайта>
  default.htm
  download[s]
    program1.exe
    program2.zip
    somefile.rar
  html[s]
    page1.htm
    page2.html
  media
    film1.avi
    film2.swf
    backgroundsound.mid
  pic[s]
    picture1.gif
    picture2.jpg
```

Вот такая структура. Как видите, файлы, в основном, организованы по типу: Web-страницы — в одной папке, графические файлы — в другой, мультимедийные — в третьей и т. д.

Корневой называется папка, в которой помещается ваш сайт. Запомните этот термин — мы часто будем использовать его в этой главе.

Файл default.htm, как вы поняли, — главная страница сайта. Этот файл почти всегда помещают в корневой каталог сайта.

В папке download[s] находятся файлы, не являющиеся ни Web-страницами, ни изображениями, ни мультимедийными данными. Это могут быть программы, файлы данных, архивные файлы и пр. Когда пользователь щелкает на гиперссылке, указывающей на такой файл, Web-обозреватель предлагает открыть его или сохранить на диске для последующего открытия. В любом случае, сам Web-обозреватель такие файлы не обрабатывает.

В папке html[s] помещаются сами Web-страницы.

В папке media находятся все мультимедийные файлы, используемые на страницах. Это фильмы в форматах AVI, Macromedia Flash и др., фоновая музыка в формате WAV, MIDI и т. п.

В папке pic[s] располагаются все графические изображения.

Иногда используется более сложная структура, когда внутри каждой папки создаются дополнительные подпапки. Это может выглядеть, например, таким образом:

```
html[s]
  chepter1
    page1.htm
    page2.htm
  chapter2
    page1.htm
    page2.htm
  . . .
pic[s]
  chapter1
    picture1.gif
    picture2.jpg
  chapter2
    picture1.gif
    picture2.fla
  . . .
```

Однако такая структура оправдана для достаточно больших сайтов. В противном случае вы рискуете заблудиться среди огромного количества подпапок, содержащих по одному файлу. Не стоит усложнять себе жизнь — организуйте физическую структуру сайта так, как вам удобно.

Иногда встречается и принципиально иной способ физического структурирования сайта. Во главу угла при этом ставится не тип файла, а логическая структура.

```
<Корневая папка вашего сайта>
  default.htm
  chapter1
    page1.htm
    page2.htm
    picture1.gif
    picture2.jpg
  chapter2
    page1.htm
    page2.htm
    picture1.gif
    picture2.fla
    program1.exe
  . . .
```

Может быть, вы найдете удобным структурировать ваш файл именно так. Но повторим, что такой способ весьма малоупотребителен.

Публикация Web-сайта

После того как сайт создан, его нужно опубликовать на Web-сервере. И сделать это можно тремя различными способами.

Первый способ самый простой, но далеко не всегда самый легкоосуществимый. Заключается он в том, что все файлы сайта переписываются на дискету или более емкий носитель информации (дискета Iomega Zip, магнитооптический диск, перезаписываемый диск CD-RW, лента стримера, обычный жесткий диск, в конце концов) и относится администратору Web-сервера. Администратор просто переписывает содержимое вашего носителя в соответствующую папку сервера и настраивает программное обеспечение. Вот и все.

Преимущество данного способа — исключительная простота. В самом деле, что может быть проще, чем переписать несколько папок и файлов на дискету и перенести ее в другое место. (А если Web-сервер находится в собственности вашей организации, можно просто скопировать информацию по сети или отправить по электронной почте.) Недостаток не менее серьезен, чем достоинство: вы должны иметь личный доступ к Web-серверу и его администратору. А для этого опять же нужно быть знакомым с администратором либо иметь свой Web-сервер.

Второй способ распространен очень мало. Некоторые бесплатные Web-серверы позволяют пользователю загрузить файлы его сайта через Web-обозреватель. Этот способ имеет всего одно довольно призрачное преимущество: простота загрузки. Пользователь вводит в поля ввода имена нужных файлов и нажимает кнопку **Submit** (Отправить). Недостатки: невозможность отправки количества файлов больше определенного лимита (обычно пяти), невозможность контролировать отправку файлов и крайняя ненадежность соединения при плохой связи.

Третий способ самый распространенный. На нем давайте остановимся подробнее.

Уже говорилось, что существует интернет-протокол FTP, используемый для передачи в сети файлов. Причем по FTP можно организовать передачу файлов как от FTP-клиента FTP-серверу, так и наоборот, от сервера клиенту. Особенностью протокола FTP является то, что клиент работает с FTP-сервером как с локальным диском своего компьютера; он может создавать, изменять и удалять файлы и папки, копировать и переносить файлы откуда угодно и куда угодно. (Правда, для этого нужно иметь разрешения на выполнение на дисках сервера каждой из этих операций.)

Эта особенность протокола FTP и легла в основу третьего и самого распространенного способа публикации Web-сайтов.

Заключается он в следующем. Администратор Web-сервера запускает и настраивает на компьютере программу FTP-сервера. После этого по запросу

автора, которому нужно опубликовать на сервере свой сайт, он создает корневую папку для этого сайта и дает автору права на доступ ТОЛЬКО к этой папке. Автор сайта с помощью программы FTP-клиента (например, CuteFTP) подключается к FTP-серверу и копирует во вновь созданную корневую папку файлы своего сайта. После этого администратор конфигурирует Web-сервер так, чтобы он "узнал" о существовании нового сайта. В дальнейшем, если автору сайта понадобится обновить какие-то файлы, он снова подключается к FTP-серверу и просто копирует на его диск обновленные файлы, затирая старые. (Реально этот процесс несколько сложнее, но мы опустим эти малозначительные сложности.)

Для того чтобы подключиться к FTP-серверу, пользователю нужно ввести свои имя и пароль. Пользователь получает их у администратора сервера. Для публикации сайта используется только именной вход — такая мера предосторожности не дает добраться до сайта злоумышленникам и просто посторонним людям.

Этот способ публикации сайтов используется на большинстве бесплатных и корпоративных Web-серверов. Сейчас давайте опишем последовательность действий для осуществления публикации сайта на сервере по FTP-протоколу.

1. Автор регистрируется на бесплатном сервере, задает имя и пароль, под которыми он будет заходить на FTP-сервер. Если сайт публикуется на корпоративном сайте, автор получает имя и пароль у администратора сервера, а корневая папка создается вручную. После этого на дисках сервера создается корневая папка его сайта, чье имя, как правило, совпадает с именем, под которым зарегистрирован автор.
2. Автор с помощью программы FTP-клиента подключается к серверу и, как правило, сразу попадает в свою корневую папку. Если подключение не произошло, нужно проверить, правильно ли введены имя и (или) пароль. Если все введено правильно, но подключения все равно не происходит, автор должен обратиться к администратору сервера.
3. Автор копирует на диски сервера файлы своего сайта. Если автору нужно всего лишь кое-что изменить, он копирует на сервер только измененные файлы, а также манипулирует файлами на сервере: удаляет, переименовывает, копирует и переносит из папки в папку. Если изменения, которые необходимо сделать, значительны по объему, проще удалить все содержимое корневой папки и скопировать все файлы заново.
4. Автор отключается от FTP-сервера. Это обязательно следует сделать, ведь FTP-сервер для поддержания соединения с клиентом забирает ресурсы компьютера, которые отнюдь не безграничны.
5. Автор запускает Web-обозреватель и тестирует свой сайт. Если обнаружатся ошибки, он вносит исправления и повторяет все шаги со 2 по 5.

Хорошая программа Web-редактора, обладающая функциями управления сайтом, сама предоставляет возможность публикации по протоколу FTP.

Dreamweaver — хорошая программа, поэтому все вышесказанное применимо и к нему. Более того, Dreamweaver сам ведет учет, какие файлы были изменены пользователем, и копирует на сервер только измененные файлы. Также он автоматически удалит ненужные файлы на сервере. Вам необходимо будет только правильно ввести данные о FTP-сервере.

Управление Web-сайтом в Dreamweaver

Ну все, хватит теории! Пора переходить к практике. Здесь мы пошагово рассмотрим публикацию сайта и возможности, предлагаемые Dreamweaver для управления сайтом.

Но сначала сделайте следующее. Найдите все файлы, составляющие наш небольшой сайт, и поместите их в отдельную папку. (Это файлы About.htm, default.htm, Links.htm, Passions.htm, Projects.htm, Email.gif, Email2.gif, Ivanov.gif и Wb02049_.gif.) Назовите эту папку, например, Sample1. Каждый сайт должен находиться в своей папке — таково требование Dreamweaver.

Регистрация сайта в Dreamweaver

Но прежде, чем начать управлять сайтом, его необходимо *зарегистрировать* в Dreamweaver. Пока что Dreamweaver не знает, что набор страниц, которые мы создали во время предыдущих занятий, — на самом деле сайт. И, соответственно, он ведет себя, будто никакого сайта нет.

Для того чтобы создать новый сайт, воспользуйтесь пунктом **New site** меню **Site**. После его выбора на экране появится диалоговое окно **Site Definition**, состоящее из двух вкладок. Если оно открыто на вкладке **Basic**, переключитесь на вкладку **Advanced** — она предоставляет больше возможностей по настройке вашего сайта. Вы увидите то, что показано на рис. 6.2.

Как видите, в левой части этого окна находится список вкладок второго уровня. Переключитесь на вкладку **Local Info**, где задается информация о файлах вашего сайта, находящихся на жестком диске вашего компьютера (*локальной копии* сайта).

В поле ввода **Site Name** вводится имя сайта. Оно служит только для того, чтобы вам самим было удобно с этим сайтом работать. Назовите сайт "Sample site1".

В поле ввода **Local Root Folder** указывается путь к корневой папке локальной копии сайта. Вы также можете щелкнуть по значку папки, расположенному справа от этого поля ввода, и выбрать нужную папку в появившемся на экране диалоговом окне.

Флажок **Refresh Local File List Automatically** включает или отключает автоматическое обновление списка файлов локальной копии сайта. Если вы оставите его включенным, список файлов сайта всегда будет обновляться авто-

матически, как только Dreamweaver становится активным. Это может быть полезным, если вы работаете с файлами сайта, используя Проводник Windows или другие программы управления файлами (например, FAR Commander), однако снижает быстродействие программы. Если же вы выключите вышеназванный флажок, вам самим придется обновлять список файлов сайта, но Dreamweaver будет активизироваться быстрее.

В поле ввода **Default Images Folder** вводится имя папки, в которой по умолчанию будут располагаться все графические изображения, помещаемые вами на Web-страницы сайта. Вы также можете щелкнуть по значку папки, расположенному справа от этого поля ввода и выбрать нужную папку в появившемся на экране диалоговом окне. Не вводите в это поле ничего.

В поле ввода **HTTP Address** вводится интернет-адрес вашего сайта. Это нужно, чтобы Dreamweaver смог проверить на правильность ссылки.

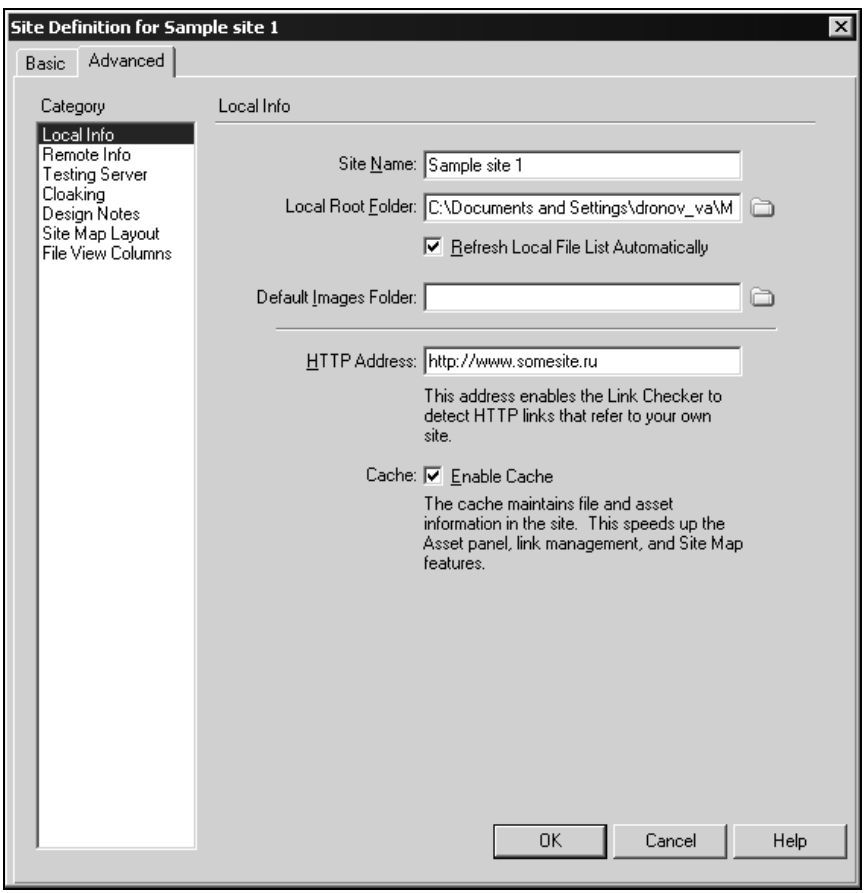


Рис. 6.2. Диалоговое окно **Site Definition** (вкладка **Local Info**)

Флажок **Enable Cache** включает или отключает кэширование файлов сайта. Кэширование позволяет ускорить операции с файлами сайта, но отнимает память компьютера. И все же лучше пожертвовать памятью и оставить кэширование включенным, иначе многие замечательные возможности Dreamweaver останутся для вас недоступными.

Покончив с вкладкой **Local Info**, перейдите к вкладке **Remote Info**, где задаются параметры *удаленной копии* сайта (файлов сайта, находящихся на дисках сервера). Эта вкладка показана на рис. 6.3.

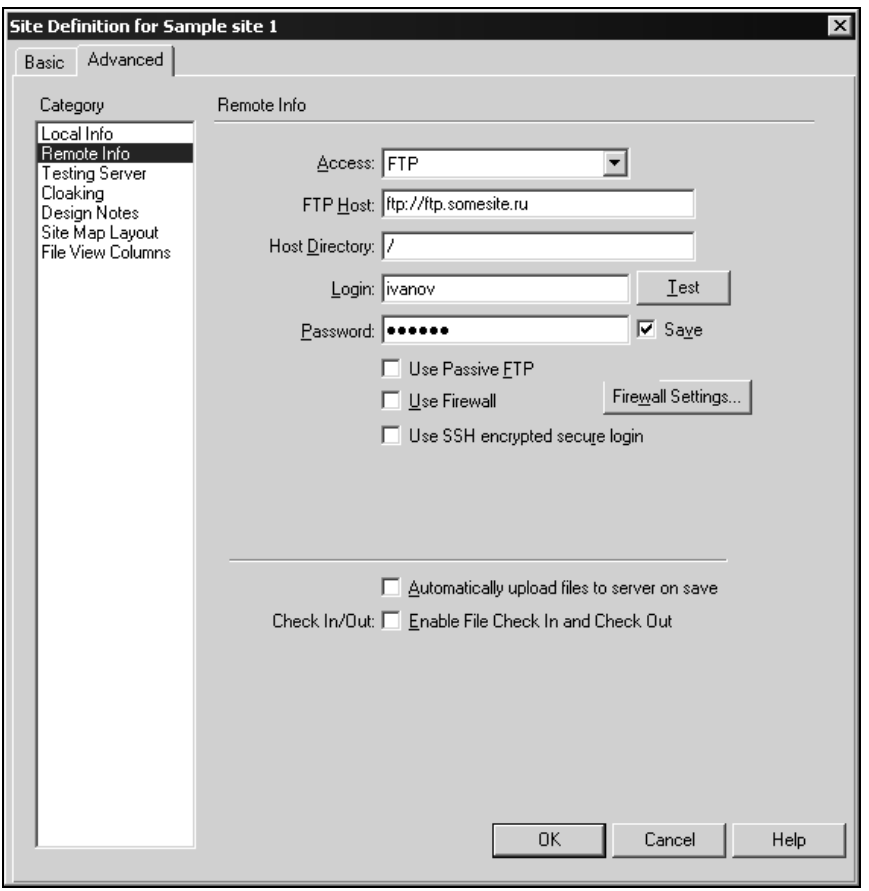


Рис. 6.3. Диалоговое окно **Site Definition**
(вкладка **Remote Info**, выбран режим отправки файлов по протоколу FTP)

Прежде всего необходимо выбрать режим отправки сайта на сервер. Выбирается он в раскрывающемся списке **Access**. Давайте рассмотрим, какие возможности предлагает нам Dreamweaver:

- **None** — удаленная копия сайта вообще не создается. Выберите этот пункт, если у вас пока нет доступа к Web-серверу, или вы создаете набор Web-страниц, которые будут распространяться, например, на дискетах или компакт-дисках;
- **FTP** — отправка сайта по протоколу FTP;
- **Local/Network** — отправка сайта по локальной сети. Также пригодится, если Web-сервер работает на вашем компьютере;
- **RDS**, **SourceSafe Database** и **WebDAV** — использование системы управления версиями. Пригодятся только в том случае, если вы — член большой группы разработчиков сайтов, использующих одну из этих систем.

Если вы выбрали пункт **None**, диалоговое окно **Site Definition** останется пустым. Никаких параметров удаленной копии вы ввести не сможете, т. к. никакой удаленной копии в этом случае создано не будет.

В случае выбора пункта **FTP** вам нужно будет ввести следующие параметры (см. рис. 6.3):

- **FTP Host** — адрес FTP-сервера вида **ftp://ftp.somesite.ru**;
- **Host Directory** — имя корневой папки вашего сайта (или "/", если корневая папка вашего сайта совпадает с корневой папкой FTP-сервера);
- **Login** — имя пользователя, под которым вы подключаетесь;
- **Password** — пароль (при его вводе в поле отображаются звездочки).

По умолчанию Dreamweaver сохраняет введенный вами пароль и автоматически вводит его при подключении к FTP-серверу. Если вы из соображений безопасности не желаете хранить этот пароль, а хотите вводить его при каждом подключении, отключите флажок **Save**.

Нажав кнопку **Test**, вы можете проверить, сможете ли подключиться к заданному FTP-серверу, используя заданные имя и пароль. После нажатия этой кнопки Dreamweaver выведет окно-предупреждение с текстом, сообщающим об удачной попытке подключения или описывающим возникшую проблему.

Если FTP-сервер, с помощью которого вы копируете файлы сайта на диски сервера, поддерживает только *пассивный протокол FTP*, включите флажок **Use Passive FTP**.

Если ваша организация для доступа в Интернет использует прокси-сервер или брандмауэр, включите флажок **Use Firewall**. О его настройке будет рассказано чуть ниже.

Включите флажок **Use SSH encrypted secure login**, если для входа на FTP-сервер используется защищенный протокол *SSH* (Secure Socket Layer — слой защищенных сокетов).

В случае если сайт будет отправляться по локальной сети, или если Web-сервер работает на вашем же компьютере, выберите пункт **Local/Network**

раскрывающегося списка **Access**. Окно **Site Definition** снова изменит свой вид (рис. 6.4).

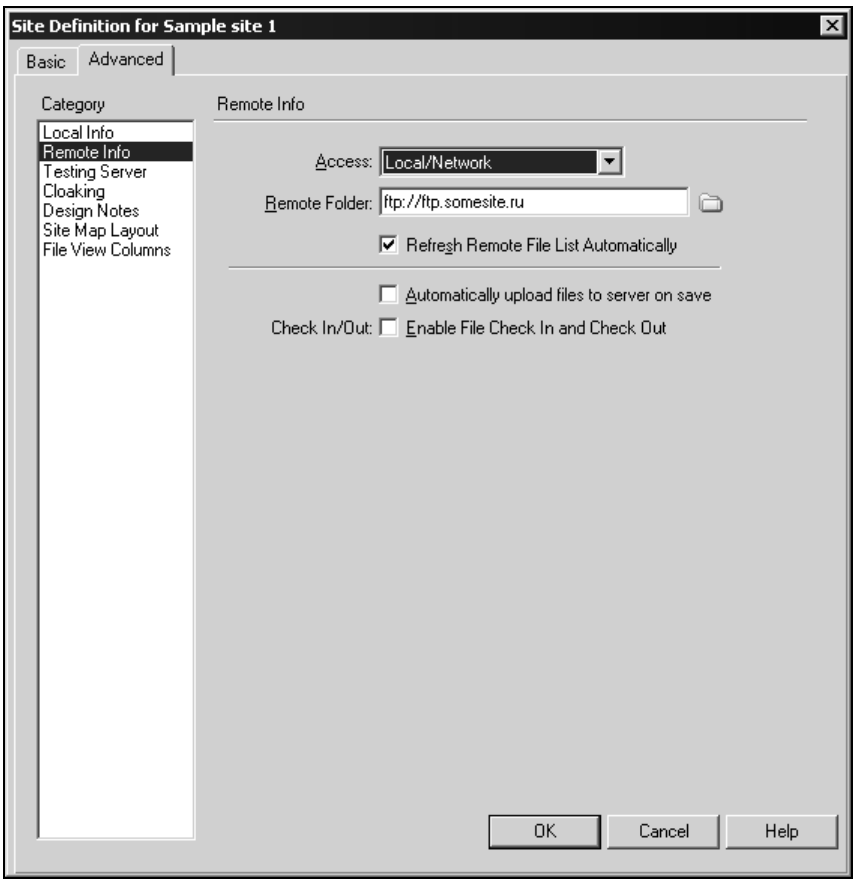


Рис. 6.4. Диалоговое окно **Site Definition**
(вкладка **Remote Info**, выбран режим отправки файлов по локальной сети)

В поле ввода **Remote Folder** вводится путь к корневой папке удаленной копии сайта (вы также можете щелкнуть значок папки, расположенный справа от поля ввода, и выбрать нужную папку в появившемся на экране диалоговом окне).

Флажок **Refresh Remote File List Automatically** включает или отключает автоматическое обновление списка файлов удаленной копии сайта. Отключите его, если хотите увеличить быстродействие системы, но в этом случае вам придется обновлять его вручную.

Если вы включите флажок **Automatically upload files to server on save**, Dreamweaver будет копировать файлы на сервер сразу же после их сохране-

ния. Вероятно, имеет смысл включать этот флажок, только если вы отправляете файлы на сервер по локальной сети или имеете постоянное подключение к Интернету. Если же вы подключаетесь к Интернету по телефонной сети, лучше отключить его и запускать копирование файлов вручную.

Вы уже заметили, что мы не стали рассматривать элементы управления, расположенные в нижней части окна. Они служат для работы в рабочей группе; описание их вы можете найти в интерактивной справке Dreamweaver. А пока что отключите флажок **Enable File Check In and Check Out**.

Остается переключиться на вкладку **Site Map Layout**, кое-что проверить и задать кое-какие дополнительные настройки. Эта вкладка показана на рис. 6.5.

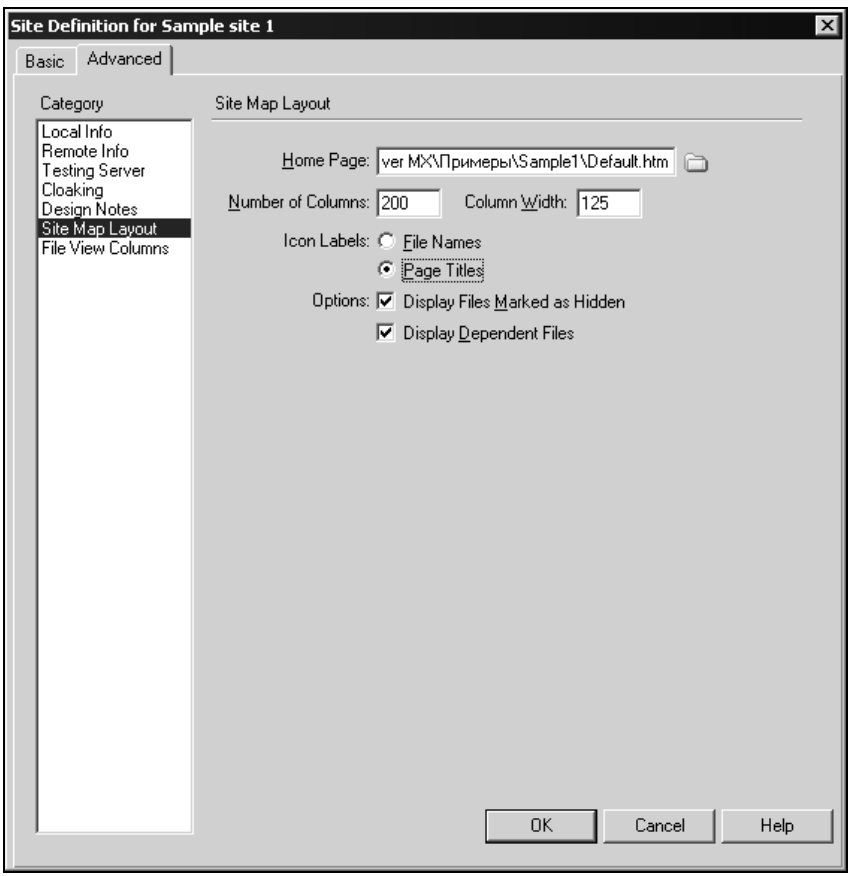


Рис. 6.5. Диалоговое окно **Site Definition** (вкладка **Site Map Layout**)

Прежде всего, проверьте, подставил ли Dreamweaver в поле ввода **Home Page** имя файла главной страницы сайта. Если нет (хотя должен был), вве-

дите его или щелкните по значку папки справа от поля ввода и выберите нужный файл в появившемся на экране диалоговом окне.

Остальные элементы управления вам на первых порах не очень-то пригодятся, а если вы ими заинтересуетесь, то сможете найти информацию о них в интерактивной справке. Рассмотрим только некоторые, которые могут вам пригодиться уже сейчас.

Прежде всего, скажем, что список файлов удаленной копии сайта может отображаться в двух режимах: как обычный список файлов и как "дерево" файлов. Во втором случае выводится иерархическая структура, показывающая как различные файлы, составляющие сайт, связаны друг с другом. Сами файлы изображаются в виде пиктограмм с подписями. Так вот, в качестве подписей могут выступать как имена файлов, так и названия Web-страниц (содержимое парного тега <TITLE>). Для выбора подписей служит группа переключателей **Icon Labels**; переключатель **File Names** задает вывод в качестве подписей имен файлов (включен по умолчанию), а **Page Titles** — названий Web-страниц. Вы можете включить любой из переключателей, в зависимости от ваших предпочтений.

Группа флажков **Options** позволяет задать следующие параметры:

- **Display Files Marked as Hidden** — включает или отключает вывод файлов, помеченных как скрытые. Если в вашем сайте содержатся такие файлы, лучше включите этот флажок;
- **Display Dependent Files** — включает или отключает вывод всех файлов, не являющихся Web-страницами (изображения, мультимедийные, программы, архивы и т. п.). Конечно, без этих файлов ваш список будет неполным, поэтому включите данный флажок. Но если ваш сайт очень велик, лучше его отключить, чтобы не загромождать список файлов.

Закончив ввод параметров вашего сайта, нажмите кнопку **OK** диалогового окна **Site Definition**. На этом регистрацию сайта можно считать законченной.

Настройка прокси-сервера или брандмауэра

А теперь давайте поговорим о настройке прокси-сервера или брандмауэра, если таковой используется в вашей организации. Если же вы не настроите Dreamweaver на использование прокси-сервера, вы не попадете в Интернет и не сумеете подключиться к FTP-серверу для отправки файлов сайта.

Сначала, конечно же, дадим определение прокси-сервера. *Прокси-сервер* (или *сервер полномочий*) — это программа (и одновременно компьютер, на котором она выполняется), в чьи задачи входит:

- предоставление доступа в Интернет нескольким пользователям по одному каналу (например, модему или высокоскоростному каналу);

- кэширование просмотренных Web-страниц на жестком диске с тем, чтобы при позднейшем просмотре можно быстро загрузить их оттуда, а не принимать повторно по медленному каналу;
- ограничение доступа из Интернета к локальной сети организации. В частности, прокси-сервер блокирует хакерские атаки и нежелательное содержимое Web-страниц (насилие, порнография и т. п.);
- разграничение прав доступа в Интернет для пользователей, в частности запрет некоторых сайтов для определенных пользователей, либо ограничение времени работы в Сети.

Как видите, прокси-сервер выполняет много функций. Однако часто используются программы, аналогичные прокси-серверу, но выполняющие не все его задачи, а только некоторые. В большинстве случаев, это обеспечение защиты локальной сети от доступа извне. Такие программы называются *брандмауэрами* (или *сетевыми экранами*, по-английски — firewall). В дальнейшем мы все такие программы будем называть прокси-серверами.

Если доступ в Интернет производится через прокси-сервер, все работающие с Интернетом программы должны быть соответствующим образом настроены.

А для того, чтобы их можно было настроить, они должны поддерживать работу через прокси-сервер. Большинство серьезных программ это поддерживают. А поскольку Dreamweaver — серьезная программа, то он тоже поддерживает прокси-сервер.

Но как его настроить?

Прежде всего, убедитесь, что в диалоговом окне **Site Definition**, на вкладке **Remote Info** (если выбран режим отправки сайта по протоколу FTP), включен флажок **Use Firewall**. Возможно, вам также придется включить флажок **Use Passive FTP**, если прокси-сервер этого требует (выяснить это можно у администратора прокси-сервера). Но это еще не все настройки — таким образом мы только дадим знать Dreamweaver, что для отправки такого-то сайта на сервер ему придется "общаться" с прокси-сервером.

А теперь — главное. Нажмите кнопку **Firewall Settings**. На экране появится знакомое вам диалоговое окно **Preferences**, уже переключенное на вкладку **Site**. Эта вкладка изображена на рис. 6.6.

Здесь нас интересуют в настоящий момент только поля ввода **Firewall Host** и **Firewall Port**. В первом вводится интернет-адрес прокси-сервера в вашей локальной сети, во втором — номер *порта TCP/IP*, своеобразного канала, по которому происходит пересылка данных того или иного протокола. Оба этих параметра вы можете выяснить у вашего сетевого администратора. Задав их, нажмите кнопку **OK**.

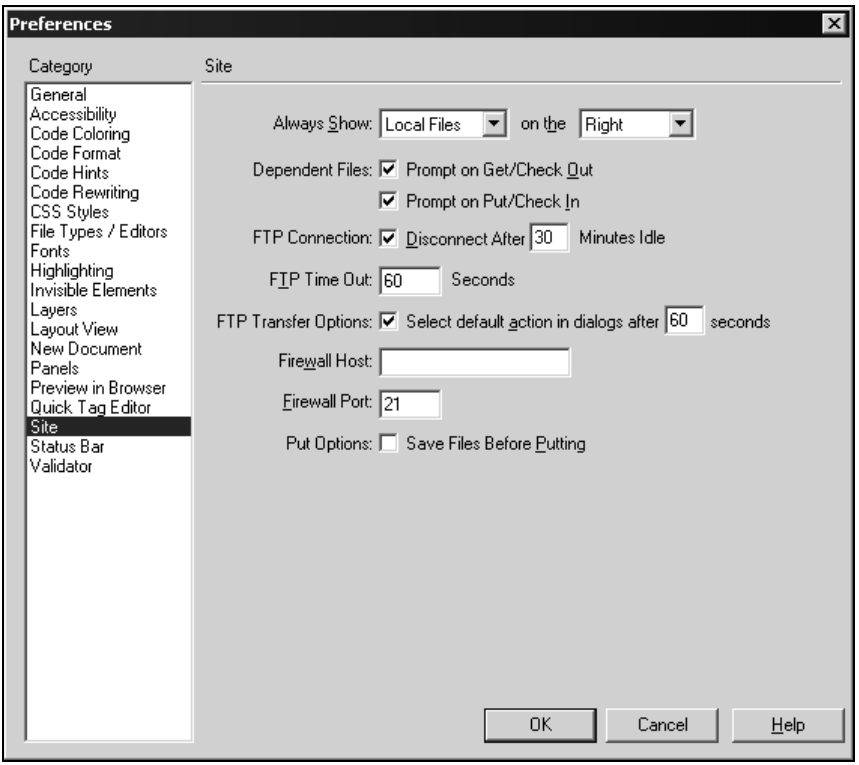


Рис. 6.6. Диалоговое окно **Preferences** (вкладка **Site**)

Панель *Site*

Все манипуляции с сайтом выполняются в панели **Site**. Поищите эту панель в доке. Если ее там нет, выберите пункт **Site** меню **Window** или нажмите клавишу <F8>. Также вы можете выбрать пункт **Site Map** меню **Site**.

Панель **Site** показана на рис. 6.7. Как видите, эта панель отличается от уже рассмотренных нами панелей Dreamweaver. Она имеет свой инструментарий с элементами управления, предоставляющими быстрый доступ к наиболее часто употребляемым командам, строку меню и строку статуса. Правда, строка статуса здесь не столь сложна, как у окна документа, и только показывает сведения о выбранном в списке файле.

Большую часть панели **Site** занимает список файлов. Изначально в ней отображается только список файлов локальной копии сайта. Файлы и папки, отображаемые в списке, организованы в виде "дерева", совсем как в Проводнике Windows. При этом узлами "дерева" служат папки: корневая папка сайта и вложенные в нее подпапки. Вы можете сворачивать и разворачивать

ветви "дерева", как в Проводнике, чтобы посмотреть содержимое той или иной папки.

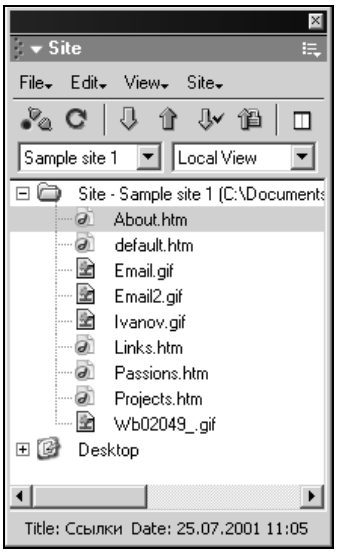


Рис. 6.7. Панель **Site**

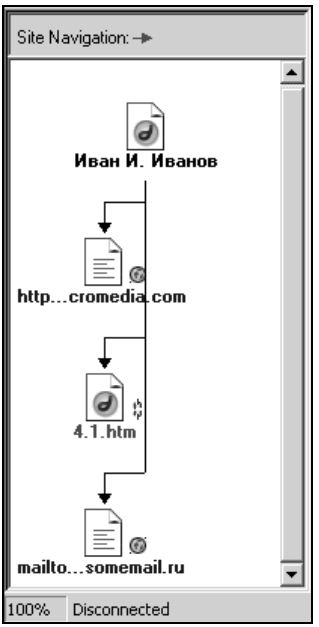


Рис. 6.8. "Дерево" файлов, отображаемое в панели **Site**

Вы можете выбрать любой из файлов в списке, щелкнув по нему мышью. Если вы хотите выбрать несколько файлов, щелкайте по ним, удерживая нажатой клавишу <Ctrl>. Если вы хотите выбрать все файлы списка, то воспользуйтесь пунктом **Select All** меню **Edit** панели **Site** или нажмите комбинацию клавиш <Ctrl>+<A>. Вы также можете воспользоваться пунктом **Invert Selection** того же меню; тогда все выделенные файлы станут невыделенными, а все невыделенные — выделенными.

Список файлов панели **Site** может работать в двух режимах. Первый режим — это показ обычного списка файлов *(режим списка)*, а второй — отображение иерархического "дерева" со всеми взаимозависимостями между страницами и связанными с ними файлами (*режим "дерева"*, показан на рис. 6.8). Для переключения этих режимов вы можете воспользоваться пунктами **Site Files** (показывает список файлов) и **Site Map** (показывает "дерево") меню **Window** главного окна Dreamweaver или одноименными пунктами меню **View** панели **Site**. Но проще всего нажать, соответственно, клавишу <F8> или комбинацию клавиш <Alt>+<F8>.

Если вы хотите просмотреть файлы удаленной копии сайта, вам нужно нажать кнопку-выключатель **Expand/Collapse** (рис. 6.9), расположенную в ин-

струментарии панели. После этого панель **Site** раздвинется и займет все главное окно Dreamweaver (рис. 6.10). В левой половине рабочей области появится список файлов удаленной копии сайта. Чтобы вернуть панель **Site** к своему обычному виду, снова нажмите кнопку **Expand/Collapse**.

Рис. 6.9. Кнопка **Expand/Collapse** инструментария панели **Site**

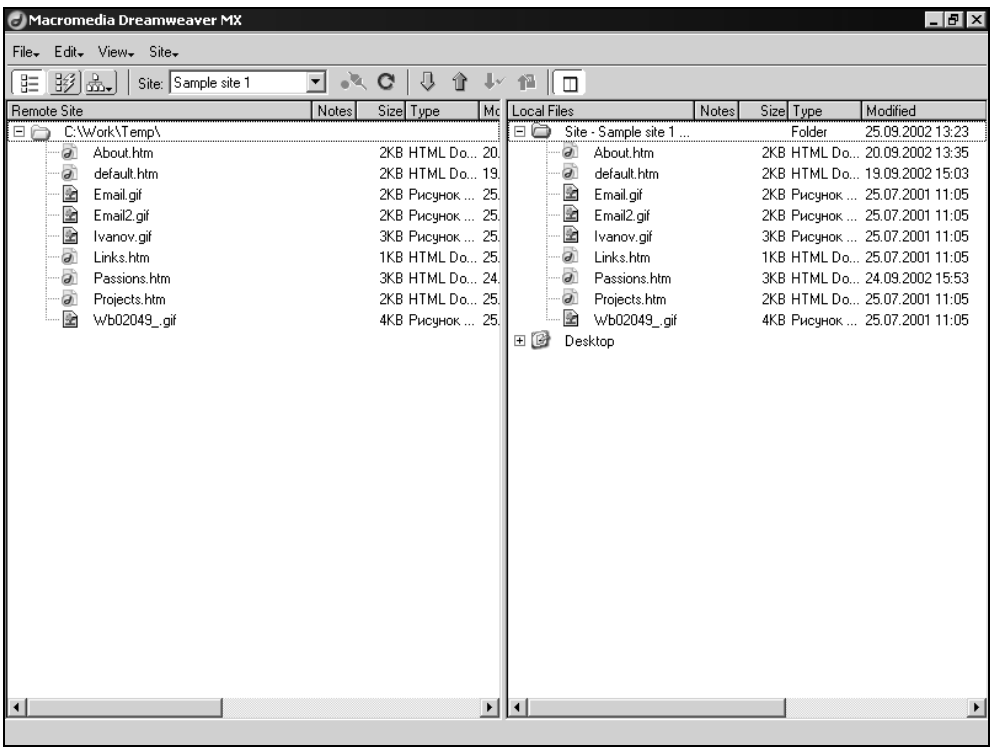


Рис. 6.10. Панель **Site** в расширенном виде

В данный момент может быть активен только один из списков. Активизировать необходимый список вы можете, щелкнув по нему мышью; вы также можете переключаться между списками, нажимая клавишу <Tab>.

Примечание

Dreamweaver можно настроить так, что список файлов удаленной копии будет выводиться справа, а список файлов локальной копии — слева. Для этого на вкладке **Site** диалогового окна **Preferences** в раскрывающемся списке **Always Show** выберите пункт **Remote Files**, а в раскрывающемся списке **on the** — пункт **Left**.

Вообще-то, если вы используете отправку файлов по протоколу FTP, список файлов удаленной копии изначально будет пустым. Чтобы просмотреть все файлы удаленной копии сайта, вам нужно будет подключиться к серверу FTP. Для этого нажмите кнопку-выключатель **Connect to remote host** (рис. 6.11), находящуюся в инструментарии панели **Site**. (Эта кнопка доступна, только если вы используете отправку файлов по протоколу FTP.) После этого Dreamweaver подключится к серверу FTP и загрузит список файлов. Чтобы отключиться от сервера FTP, еще раз щелкните эту кнопку.

Рис. 6.11. Кнопка **Connect to remote host** инструментария панели **Site**

Рис. 6.12. Кнопка **Refresh** инструментария панели **Site**

Если вы в диалоговом окне **Site Definition** отключили флажки **Refresh Local File List Automatically** и **Refresh Remote File List Automatically**, то, как мы уже говорили, вам придется обновлять эти списки вручную. Чтобы обновить активный список, выберите пункт **Refresh** меню **View** панели **Site** или нажмите клавишу <F5>. Если вы предпочитаете пользоваться мышью, щелкните кнопку **Refresh** инструментария панели **Site** (рис 6.12).

В инструментарии панели **Site** вы также можете увидеть два раскрывающихся списка. Они тоже могут быть полезны нам.

Раскрывающийся список, расположенный слева, позволяет быстро выбрать любой из зарегистрированных в Dreamweaver сайтов для отображения в панели. Это значит, что вы можете задать сколько угодно сайтов в Dreamweaver и с легкостью переходить от одного к другому.

Правый раскрывающийся список позволит вам выбрать режим отображения списка. В нем доступны четыре пункта:

- **Local View** — отображение списка файлов локальной копии;
- **Remote View** — отображение списка файлов удаленной копии;
- **Testing Server** мы пока рассматривать не будем;
- **Map View** — отображение "дерева" файлов локальной копии.

Пункт **New Site** меню **Site** панели позволяет ввести в Dreamweaver сведения о новом сайте. А на пункте **Edit Sites** этого же меню мы остановимся подробнее. При его выборе на экране появляется диалоговое окно **Edit Sites**, показанное на рис. 6.13. С помощью этого окна вы можете зарегистрировать в программе новые сайты, изменить сведения об уже существующих или совсем удалить ненужные сайты. Кстати, вызвать это окно вы можете, выбрав пункт **Edit Sites** раскрывающегося списка сайтов, находящегося в инструментарии панели.

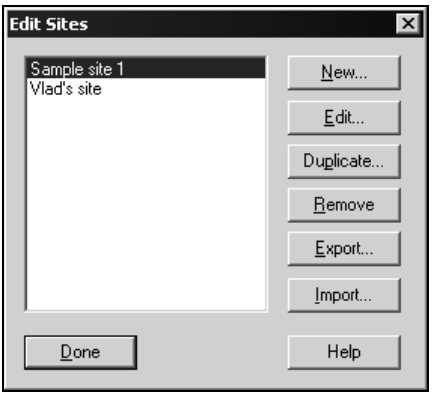


Рис. 6.13. Диалоговое окно **Edit Sites**

В списке, занимающем большую часть этого окна, перечислены все зарегистрированные в Dreamweaver сайты. Вы можете выбрать любой и произвести с ним различные манипуляции.

Нажав кнопку **Edit**, вы сможете отредактировать сведения о выделенном в списке сайте в диалоговом окне **Site Definition**.

Нажав кнопку **Remove**, вы получите возможность удалить сведения о ненужном сайте (но не сами файлы, входящие в этот сайт!). При этом Dreamweaver спросит вас, действительно ли вы хотите удалить эти сведения; нажмите кнопку **Yes** или **No**.

Нажав кнопку **New**, вы сможете ввести новый сайт в диалоговом окне **Site Definition**.

А при нажатии кнопки **Duplicate** в программе будет автоматически зарегистрирована полная копия выбранного сайта. После этого вы, если хотите, сможете изменить ее параметры. Это удобно, если вы собираетесь работать с двумя копиями одного и того же сайта.

Если же вы просто нажмете кнопку **Done**, закрывая диалоговое окно, Dreamweaver автоматически активизирует выбранный в списке сайт и выведет его содержимое в своем окне сайта.

Работа с файлами сайта

Итак, с панелью **Site** в общих чертах мы познакомились. Теперь пора приниматься за файлы, которые отображаются в списках. Давайте посмотрим, что мы можем с ними делать.

А Dreamweaver позволяет сделать с файлами многое. Фактически вы можете управлять ими так же, как в окне Проводника Windows или другой программы управления файлами. Давайте рассмотрим все файловые операции, поддерживаемые Dreamweaver.

Чтобы удалить выбранный в списке файл, нажмите клавишу <Del>. Вы также можете выбрать пункт **Delete** меню **File** или контекстного меню. После этого Dreamweaver спросит вас, действительно ли вы хотите удалить названный файл; нажмите кнопку **Yes** или **No**. Точно так же вы можете удалить любую папку.

Внимание!

Dreamweaver выполняет удаление файлов и папок с файлами в корзину Windows. Однако пустые папки удаляются бесследно.

Чтобы переименовать выбранный файл, щелкните по нему еще раз мышью или нажмите клавишу <F2>. Вы также можете выбрать пункт **Rename** меню **File** или контекстного меню. После этого на месте имени данного файла появится небольшое поле ввода, где будет подставлено его старое имя. Введите новое имя в это поле ввода или отредактируйте старое в нем же, после чего нажмите клавишу <Enter> для сохранения нового имени или клавишу <Esc> — для отказа от сохранения.

Если на Web-страницах присутствуют гиперссылки на переименованный файл, Dreamweaver предложит вам исправить их. На экране появится диалоговое окно **Update Files**, показанное на рис. 6.14. В списке **Update links in the following files?** находится список Web-страниц, где найдены гиперссылки, указывающие на переименованный файл. Нажмите кнопку **Update**, чтобы обновить их, или **Don't Update**, чтобы отказаться от обновления. В последнем случае вам придется сделать это вручную.

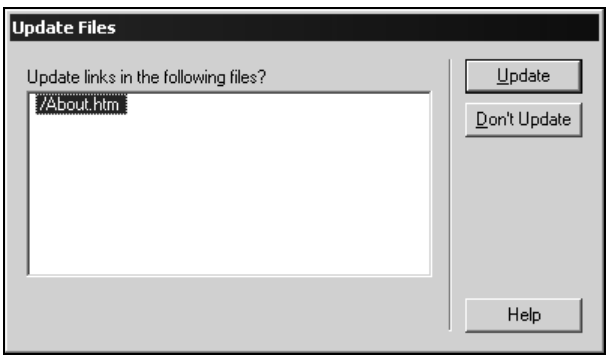


Рис. 6.14. Диалоговое окно **Update Files**

Чтобы открыть выбранный файл Web-страницы в окне документа Dreamweaver, дважды щелкните по нему или нажмите клавишу <Enter>. Вы также можете выбрать пункт **Open** меню **File** или контекстного меню. Если у вас установлена программа Web-графики Macromedia Fireworks, вы можете открыть не только файл Web-страницы, но и любой графический файл.

Чтобы создать новую Web-страницу, выберите пункт **New File** меню **File** или контекстного меню или нажмите комбинацию клавиш <Ctrl>+<Shift>+<N>. После этого в самом низу списка появится новый файл по имени untitled.htm, которое вы сразу же можете (и должны) изменить; изменив его, нажмите клавишу <Enter>, после чего вы можете открыть его в окне документа для того, чтобы ввести содержимое страницы.

Новая папка создается аналогично, только нужно выбрать пункт **New Folder** или нажать комбинацию клавиш <Ctrl>+<Shift>+<Alt>+<N>. После этого в самом низу списка появится новая папка по имени untitled, имя которое вы должны будете изменить; нажмите клавишу <Enter>. Запомните при этом, что названная папка будет вложена в ту, в которой вы сейчас находитесь. Например, если вы находитесь в корневой папке сайта, новая папка будет вложена в нее. А если вы выбрали какой-либо файл в папке второго уровня (или ее саму), новая папка будет вложена в папку второго уровня.

Создайте в корневой папке нашего сайта папки второго уровня HTMLs и Pics. Впоследствии мы поместим в них все Web-страницы, кроме главной, и все графические изображения.

Вы можете переносить файлы (и папки) из одной папки в другую, просто перетаскивая их мышью. Если же вы во время перетаскивания будете удерживать нажатой клавишу <Ctrl>, будет выполнено копирование файла или папки.

Другой способ переноса файлов — это использование буфера обмена. (Таким же способом можно копировать файлы.) Выделите нужные файлы и нажмите комбинацию клавиш <Ctrl>+<X> (можно также выбрать пункт **Cut** в меню **Edit** или контекстном меню). После этого выбранные файлы будут помещены в буфер обмена Windows. Чтобы вставить их в нужную папку, поместите курсор на один из файлов, содержащийся в ней, или на саму папку и нажмите комбинацию клавиш <Ctrl>+<V> (или выберите пункт **Paste** в меню **Edit** или контекстном меню). Файлы будут вставлены из буфера обмена в данную папку. Если вы хотите скопировать файлы в выбранную папку, вам нужно будет не переместить, а скопировать файлы в буфер обмена, нажав комбинацию клавиш <Ctrl>+<C> (или выбрав пункт **Copy** меню **Edit** или контекстного меню).

Когда вы перемещаете файлы из папки в папку, Dreamweaver стоит на страже корректности гиперссылок. Как только файл, на который ссылаются какие-либо Web-страницы, переносится в другую папку, на экране появляется уже знакомое вам диалоговое окно **Update Files**, показанное на рис. 6.14. Это относится как к страницам, так и к графическим и прочим файлам сайта.

Теперь сделайте следующее. Переместите все файлы Web-страниц, кроме главной (файл default.htm), в папку HTMLs, а все без исключения файлы графических изображений — в папку Pics. Если Dreamweaver спросит вас,

обновлять ли гиперссылки, указывающие на перемещенные файлы, нажмите кнопку **Update**. Теперь физическая структура нашего сайта примет более приличный вид.

Кроме того, любую Web-страницу вы сможете просмотреть в Web-обозревателе. Для этого выделите ее в списке файлов и нажмите клавишу <F12>, чтобы вызвать первичный Web-обозреватель, или комбинацию клавиш <Ctrl>+<F12>, чтобы вызвать вторичный Web-обозреватель. Вы также можете выбрать соответствующий пункт подменю **Previews in Browser** меню **File** или контекстного меню.

Если вы при регистрации нового сайта ошиблись в задании файла главной страницы (или если ошибся Dreamweaver — ведь он пытается автоматически определить ее), вы можете исправить эту ошибку. Для этого выделите нужный HTML-файл в списке файлов и выберите пункт **Set as Home Page** меню **Site** или контекстного меню.

Взаимодействие панели *Site* и окна документа

Однако публиковать сайт в Сети еще рано. Ведь полноценного Web-сайта у нас еще нет — просто набор Web-страниц, никак не связанных между собой. Ведь мы еще не связали страницы друг с другом, не расставили на главной странице гиперссылки, с помощью которых посетитель сможет перейти на другие страницы сайта. Давайте же сделаем это!

Вы уже знаете несколько способов создания гиперссылок. Сейчас мы рассмотрим с вами еще один способ. И в этом нам поможет панель **Site**.

Откройте главную страницу сайта default.htm в окне документа. Для этого достаточно дважды щелкнуть по названию файла в списке файлов локальной копии. Давайте сначала поместим на главной странице гиперссылку, указывающую на страницу списка увлечений — страницу Passions.htm.

Выделите в окне документа текст "о моих увлечениях" (первая строка списка) и посмотрите на редактор свойств, а именно на поле ввода **Link**, где указан интернет-адрес гиперссылки. Справа от него находятся два значка. Один, уже знакомый нам, находится справа и имеет вид папки, и при щелчке по нему открывает диалоговое окно **Select File**. А второй, находящийся слева и имеющий вид мишени, нам пока еще не знаком. Сейчас самое время с ним познакомиться.

Странный значок, не правда ли? На щелчок мышью он не откликается. Зачем он нужен?

Дело в том, что его не надо щелкать. Его необходимо перетаскивать на требуемый файл в панели **Site**. Давайте так и сделаем. Перетащим этот значок на файл Passions.htm в списке файлов этой панели и "бросим" его. В поле ввода **Link** редактора свойств тотчас появится адрес страницы Passions.htm.

Вы можете поступить и по-другому — перетащить файл Passions.htm из окна сайта прямо в поле ввода **Link** редактора свойств. Попробуйте сделать это с файлом Projects.htm, только прежде выделите вторую строку списка ("о выполненных мной проектах").

Осталось лишь создать последнюю гиперссылку, указывающую на страницу ссылок (Links.htm), и исправить ссылку, указывающую на страницу сведений об авторе (About.htm). Выделите слово "здесь" в предпоследнем абзаце и создайте гиперссылку одним из рассмотренных нами способов. Далее поставьте курсор на текст "подробнее обо мне любимом" и также выполните операцию создания гиперссылки — Dreamweaver поймет, что вы хотите сделать, и исправит адрес уже существующей гиперссылки.

Вот и все. Сайт готов. Закройте окно документа, не забыв сохранить его содержимое. Теперь сайт можно публиковать в Сети.

Но не спешите делать этого. Давайте поговорим еще кое о чем, что вам обязательно следует знать.

Абсолютные и относительные интернет-адреса

А поговорим мы об интернет-адресах. Точнее, о разных способах задания интернет-адресов.

Снова откройте главную страницу сайта default.htm. Поместите текстовый курсор на слово "Macromedia" в первом абзаце (если помните, это гиперссылка, указывающая на сайт фирмы). В поле ввода **Link** редактора свойств вы увидите следующее:

http://www.macromedia.com

Это Web-адрес сайта Macromedia. Теперь давайте рассмотрим еще один интернет-адрес, на этот раз воображаемый:

http://www.somesite.ru/folder1/folder2/page.html

Этот адрес указывает на файл page.html, находящийся на сервере http://www.somesite.ru в папке /folder1/folder2. Как видите, все основные части интернет-адреса здесь присутствуют. В результате адрес указывает прямо на нужный файл. Такой интернет-адрес называется *абсолютным*.

Теперь переместим текстовый курсор на первую строку списка ("о моих увлечениях"). В поле ввода **Link** мы увидим следующее:

HTMLs/Passions.htm.

Этот адрес также указывает на нужный файл, но не непосредственно, а относительно какого-то другого файла, а именно относительно адреса главной страницы нашего сайта. Такой адрес называется *относительным*.

Относительные интернет-адреса могут быть заданы относительно файла и корневой папки сайта. Проиллюстрируем это на примере. Откройте страни-

цу About.htm и щелкните мышью по фотографии Ивана Ивановича, чтобы ее выделить. В поле ввода **Src** вы увидите:

```
../Pics/Ivanov.gif
```

Этот интернет-адрес указан относительно файла Web-страницы. Две точки в начале обозначают ссылку на папку верхнего уровня.

А вот так будет выглядеть адрес, указанный относительно корневой папки сайта:

```
/Pics/Ivanov.gif
```

Способ задания относительных интернет-адресов выбирается с помощью раскрывающегося списка **Relative To** (рис. 6.15) диалогового окна **Select File**: пункт **Document** задает адрес относительно файла, а пункт **Site Root** — относительно корневой папки сайта. В таком случае в поле ввода **URL** появляется результирующий интернет-адрес.

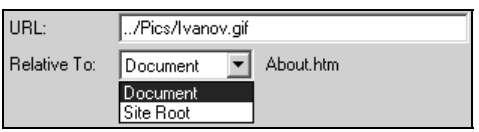


Рис. 6.15. Раскрывающийся список **Relative To** и поле ввода **URL** диалогового окна **Select File**

Как вы уже поняли, абсолютный интернет-адрес применяется, если нужный файл находится на другом сайте. Внутри сайта используются относительные адреса: они значительно компактнее и не создают лишней путаницы.

Но какой способ задания относительных адресов выбрать? Это зависит от того, где находится файл, на который вы хотите сослаться. Если он располагается в той же папке, что и файл текущей страницы, или в одной из подпапок этой папки, лучше задать адрес относительно файла, т. к. он будет компактнее. Если же нужный файл находится в другой папке, не вложенной в ту, где расположен файл текущей страницы, лучше задать адрес относительно корневой папки.

Проиллюстрируем вышесказанное на примере. Предположим, есть сайт с такой физической структурой:

```
<Корневая папка>
  default.htm
  folder1
    page1.htm
    folder2
      page2.htm
  folder3
    page3.htm
```

И предположим, что нужно разместить на странице page1.htm ссылки на страницы page2.htm и page3.htm. В таком случае правильно заданные интернет-адреса будут выглядеть следующим образом:

```
folder2/page2.htm
/folder3/page3.htm
```

т. е. адреса задаются относительно файла — для page2.htm и относительно корневой папки — для page3.htm. Если же мы поступим наоборот:

```
/folder1/folder2/page2.htm
../folder3/page3.htm
```

то смотрите сами, насколько громоздкими и запутанными стали адреса.

И еще. Если вы собираетесь публиковать сайт на Web-сервере, можете задавать относительные адреса как от файла, так и от корневой папки сайта. Но если ваш сайт будет существовать только в виде локальной копии (например, чисто учебный сайт, который никогда не будет опубликован в Сети, или сайт-прототип, предварительная версия), задавайте относительные адреса только от файла страницы. Адреса, заданные относительно корневой папки сайта, будут работать только под управлением программы Web-сервера; операционная система вашего компьютера просто не сможет их найти, поскольку будет отсчитывать адреса от корневой папки диска.

Для примера давайте сравним два наших интернет-адреса:

```
/folder3/page3.htm
../folder3/page3.htm
```

Предположим также, что локальная копия сайта находится в папке

```
c:\Projects\Sites\Sample1
```

Первый интернет-адрес задан относительно корневой папки сайта. Операционная система, однако, будет отсчитывать его от корневой папки диска, и результирующий путь будет таков:

```
c:\Folder3\page3.htm
```

что не соответствует действительности. Поэтому вы не сможете перейти на данную страницу, щелкнув по гиперссылке — операционная система не найдет этот файл, и Web-обозреватель выдаст сообщение об ошибке. В то же время, второй интернет-адрес, заданный относительно файла страницы, будет интерпретирован правильно:

```
c:\Projects\Sites\Sample1\Folder3\page3.htm
```

и операционная система без труда найдет нужный файл.

Проверка правильности ссылок и HTML-кода

Теперь-то, спросите вы, можно публиковать сайт? Нет, еще рано. Потерпите — осталось совсем чуть-чуть! Вот проверим корректность HTML-кода и гиперссылок — и опубликуем сайт в Сети.

При разработке Web-страниц в их HTML-коде постепенно накапливаются ошибки, для удаления которых в Dreamweaver существует функция "чистки" кода, рассмотренная нами в *главе 3*. Однако не все ошибки кода могут быть таким образом выявлены и исправлены. Скорее, функция "чистки" служит для оптимизации кода путем удаления и совмещения тегов, которые могут быть удалены и совмещены без ущерба для Web-страницы. Однако многие ошибки (в частности, типичные ошибки начинающего Web-дизайнера) Dreamweaver "вычистить" не может — чтобы их исправить, требуется вмешательство самого Web-дизайнера.

Но чтобы исправить код, нужно знать, что исправлять. И в этом Dreamweaver снова вам поможет.

Внимание!

Из-за ошибки Dreamweaver при проверке не может прочитать файлы, пути доступа к которым содержат русские буквы. Поэтому лучше всего помещать файлы ваших сайтов в папки, чьи имена состоят только из английских букв и цифр.

Итак, проверим правильность HTML-кода. Закройте все открытые окна документов, чтобы они вам не мешали. Выберите пункт **Reports** в меню **Site** панели **Site**. На экране появится диалоговое окно **Reports**, показанное на рис. 6.16.

Большую часть этого окна занимает иерархический список **Select Reports**, позволяющий выбрать данные, включаемые в отчет. Он состоит из двух "ветвей": **Workflow** и **HTML Reports**. Параметры, предлагаемые первой "ветвью", нужны для рабочей группы, так что мы сосредоточимся на второй "ветви", предлагающей нам задать параметры отчета по HTML. Это "дерево" содержит следующие пункты:

- **Combinable Nested Font Tags** — включает поиск вложенных тегов `<FONT>`, которые могут быть безболезненно объединены;
- **Accessibility** — включает проверку на корректность задания различных элементов страницы (все ли таблицы имеют заголовок, задана ли правильная кодировка и т. п.);
- **Missing Alt Text** — включает поиск незаполненных атрибутов `ALT` тегов `<IMG>`;
- **Redundant Nested Tags** — включает поиск ненужных вложенных тегов;
- **Removable Empty Tags** — включает поиск пустых тегов, которые могут быть безболезненно удалены;

- **Untitled Documents** — включает поиск Web-страниц без названия (без тега <TITLE>).

Каждый из этих пунктов представляет собой флажок, который вы можете установить или сбросить.

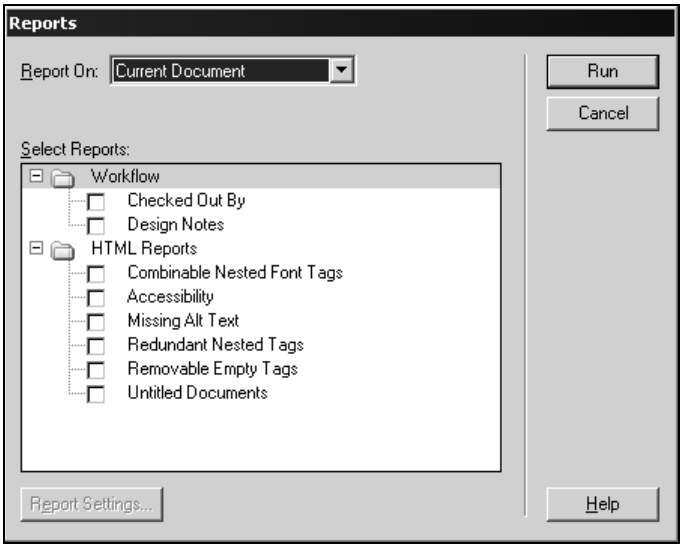


Рис. 6.16. Диалоговое окно **Reports**

Раскрывающийся список **Report On** позволяет задать, по каким страницам сайта будет выводиться отчет. Здесь доступны четыре пункта:

- **Current Document** — по странице, открытой в активном окне документа;
- **Entire Local Site** — по всему сайту;
- **Selected Files in Site** — по всем страницам, выделенным в списке файлов;
- **Folder** — по всем страницам, содержащимся в какой-либо папке.

Если вы выбрали последний пункт, ниже раскрывающегося списка **Report On** появится поле ввода, где вы должны будете ввести путь к нужной папке. Вы также можете щелкнуть мышью по значку папки справа от этого поля ввода и выбрать нужную папку в появившемся на экране диалоговом окне.

Задав необходимые параметры, нажмите кнопку **Run**. Через некоторое время Dreamweaver выведет панель **Site Reports**, показанную на рис. 6.17.

В списке, находящемся в этой панели, перечислены все найденные ошибки в HTML-коде. Вы можете выбрать любую из них. Дважды щелкнув по нужному пункту списка, вы откроете файл, в котором найдена ошибка; при этом окно документа откроется в режиме показа HTML-кода, причем ошибочный фрагмент будет выделен. Чтобы получить сведения о найденной

ошибке, нажмите кнопку **More Info** (с изображением восклицательного знака) — откроется окно справки Dreamweaver с нужными сведениями. А, нажав кнопку **Save Report** (с изображением дискеты), вы можете сохранить этот отчет в файле формата XML.

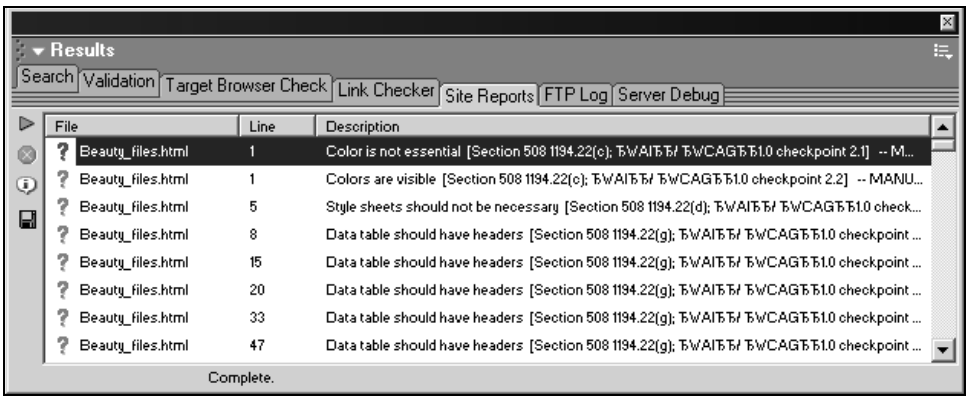


Рис. 6.17. Панель **Site Reports**

Исправьте найденные Dreamweaver ошибки, после чего закройте все окна документов, не забыв сохранить их содержимое. Не годится, чтобы наш сайт, даже и первый, содержал на своих страницах ошибки. Закройте панель **Site Reports**. Можете сделать это без сохранения.

Проверим теперь корректность гиперссылок.

Но что может случиться с гиперссылками? А случиться может, например, следующее. Предположим, вы поместили на свою Web-страницу гиперссылку, указывающую на другую страницу (неважно, принадлежит она тому же сайту или совсем другому). Теперь переименуйте или удалите файл той страницы, на которую указывает гиперссылка. Что получится, если посетитель вашего сайта щелкнет по такой гиперссылке? Web-сервер не сможет ее найти и отправит Web-обозревателю, запросившему несуществующую страницу, сообщение об ошибке. (Это так называемая *"ошибка 404"*, печально известная ошибка, чей код равен 404. Да вы и сами, наверное, не раз с ней сталкивались.) А гиперссылка, указывающая на несуществующий файл, называется *"оборванной"* или *"мертвой"*.

Конечно, Dreamweaver делает все, чтобы не допустить появления "оборванных" ссылок. Но, как говорится, и на старуху бывает проруха. Поэтому в него включены мощные средства выявления некорректных интернет-адресов, будь то адрес Web-страницы, изображения или файла архива.

Проверить корректность гиперссылок можно как на отдельной странице, так и во всем сайте. Чтобы проверить гиперссылки на отдельной странице сайта, сначала откройте ее. После этого выберите пункт **Check Links** подме-

ню **Check Page** меню **File** главного окна или нажмите комбинацию клавиш <Shift>+<F8>. Если вы хотите проверить ссылки на всех страницах сайта, просто выберите пункт **Entire Site** подменю **Check Links** контекстного меню панели **Site**. Также вы можете использовать пункт **Check Links Sitewide** меню **Site** панели или комбинацию клавиш <Ctrl>+<F8>. В любом случае после этого на экране появится панель **Link Checker**, показанная на рис. 6.18.

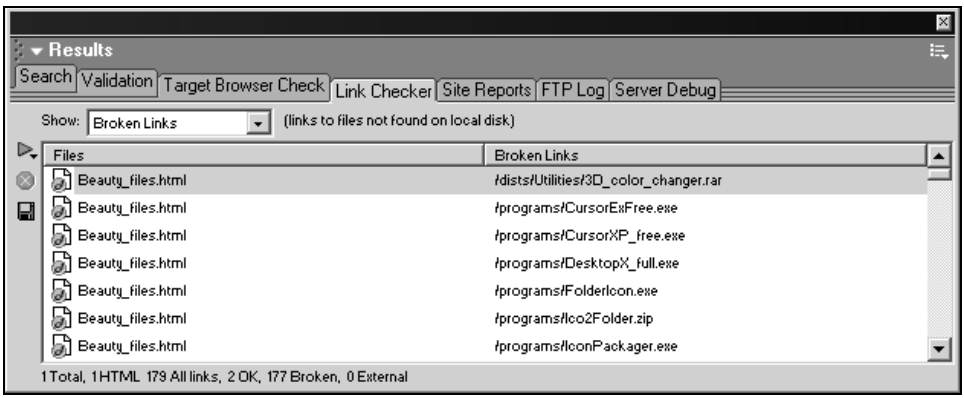


Рис. 6.18. Панель **Link Checker**

В верхней части этой панели находится раскрывающий список **Show**, где можно выбрать, что будет показано в списке, занимающем панель. В этом списке доступны три пункта:

- **Broken Links** — *"оборванные"* гиперссылки;
- **External Links** — *"внешние"* ссылки, т. е. ссылки, указывающие на другие сайты;
- **Orphaned Files** — *файлы-"сироты"*, т. е. файлы, на которые не указывает ни одна гиперссылка. В списке таких файлов должен быть только один пункт — default.htm (наша главная страница).

Как видите, каждый пункт списка содержит адрес страницы, на которой была найдена ссылка, и саму гиперссылку. Щелкнув по любой позиции этого списка, вы откроете соответствующую Web-страницу в окне документа, причем подсвечена будет как раз выбранная гиперссылка. Нажав кнопку **Save Report** (с изображением дискеты), вы сможете сохранить этот список в виде обычного текстового файла.

Также Dreamweaver предлагает мощное средство быстрой замены одной гиперссылки на другую по всему сайту. Это может быть полезно, если вы собираетесь изменить один адрес на другой, но не помните, на каких страницах он был указан. Для этого выберите пункт **Change Links Sitewide** в меню **Site** панели. На экране появится диалоговое окно **Change Link Sitewide**, показанное на рис. 6.19.

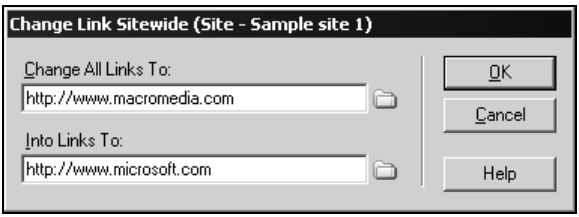


Рис. 6.19. Диалоговое окно **Change Link Sitewide**

В поле ввода **Change All Links To** введите адрес, который вы хотите заменить, в поле ввода **Into Links To** — адрес, на который вы меняете, и нажмите кнопку **OK**. Dreamweaver выведет на экран уже знакомое вам окно **Update Files**, показанное на рис. 6.14; вам будет нужно только щелкнуть кнопку **Update**.

Работа с сервером

Итак, настал тот желанный миг, когда наш сайт совсем готов! Мы создали все страницы, связали их друг с другом, исправили ошибки в HTML-коде и даже проверили на корректность интернет-адреса гиперссылок. То есть сделали все, что очень часто забывают сделать даже довольно опытные Web-дизайнеры. И, конечно же, помог нам в этом Dreamweaver.

Теперь наш сайт действительно можно публиковать в Сети.

Публикация сайта

Прежде чем публиковать сайт, проверьте его настройки. В частности, особое внимание обратите на настройки FTP-сервера — если вы допустите ошибку, выявить ее позднее, во время работы с удаленной копией сайта, будет очень трудно. Попробуйте, используя программу FTP-клиента, соединиться с FTP-сервером, отправить в корневую папку вашего сайта какой-нибудь ненужный файл и сразу же удалить его. Если все пройдет нормально, значит, у вас есть все необходимые права доступа; в противном случае обратитесь к администратору сервера.

Чтобы соединиться с FTP-сервером, нажмите кнопку **Connect to remote host**, расположенную на инструментарии панели **Site** (см. рис. 6.11). Вы также можете выбрать пункт **Connect** в меню **Site** панели **Site** или нажать комбинацию клавиш <Ctrl>+<Alt>+<Shift>+<F5>. Если в настройках сайта вы отказались от сохранения пароля (отключили флажок **Save** на вкладке **Remote Info** диалогового окна **Site Definition**), Dreamweaver выведет небольшое диалоговое окно с полем ввода, где вы должны будете ввести пароль, и кнопками **OK** (соединение) и **Cancel** (отказ от соединения).

В любом случае, после этого правый список панели заполнится. Поначалу он будет, правда, содержать одну-единственную строчку — название вашей удаленной корневой папки. Собственно, так оно и должно быть — ведь мы пока еще ничего туда не скопировали.

Простейший способ опубликовать наш сайт — это выбрать корневую папку в списке файлов локальной копии и нажать кнопку **Put File(s)** инструментария панели (рис. 6.20). Данная кнопка запускает процесс копирования файлов на FTP-сервер. Также вы можете выбрать пункт **Put** меню **Site** панели или контекстного меню или нажать комбинацию клавиш <Ctrl>+<Shift>+ +<U>.

Рис. 6.20. Кнопка **Put File(s)** инструментария панели **Site**

Так как вы выбрали корневую папку, т. е. фактически весь сайт, Dreamweaver переспросит вас, действительно ли вы хотите скопировать весь сайт на сервер. Нажмите кнопку **Yes**.

Во время отправки сайта на сервер Dreamweaver будет отображать ход отправки файлов в небольшом диалоговом окне. Вы можете прервать отправку, щелкнув по кнопке **Cancel** этого окна.

Вот мы и дождались окончания отправки! Наш сайт опубликован на сервере. Просто? Даже слишком.

Кстати, если вы перед публикацией сайта на сервере все-таки забыли к нему подключиться — не беда. Dreamweaver сам подключится к серверу. (Правда, отключаться вам придется все-таки вручную.)

Протестируйте свежеопубликованный сайт в Web-обозревателе. Посмотрите, все ли работает. Хотя, мы вроде бы исправили все ошибки...

Но давайте предположим, что вам что-то не понравилось, и вы все-таки внесли некоторые изменения и усовершенствования в некоторые Web-страницы. После этого вам будет необходимо опубликовать измененные файлы на сервере, перезаписав их удаленные копии. Как это сделать? Так же просто, как мы только что опубликовали сайт целиком. А именно, выделите нужные файлы и нажмите кнопку **Put file(s)**. Если вы открыли какие-либо страницы, изменили их и не сохранили, Dreamweaver предложит вам сохранить данные файлы; в этом случае нажмите кнопку **Yes** для сохранения соответствующего файла, кнопку **No** — для отказа от сохранения или кнопку **Cancel** — для отказа от его публикации.

Также возможно, что при этом Dreamweaver спросит вас, обновлять ли на сервере зависимые файлы, т. е. те, на которые ссылается публикуемая Web-страница. Нажмите кнопку **Yes** для публикации зависимых файлов или кнопку **No** — для отказа от их обновления (например, если на сервере уже

есть все необходимые файлы, а вы просто хотите обновить саму страницу) или кнопку **Cancel** — для отказа от публикации страницы со всеми ее файлами.

А можно опубликовать файлы на сервере еще проще, для чего достаточно перетащить их из списка файлов локальной копии в список копии удаленной. Точно так же, как вы перемещали файлы локальной копии из папки в папку.

Да, все это просто. Если вы точно знаете, какие файлы хотите опубликовать. Но что делать, если вы не помните, какие файлы изменяли? Публиковать заново весь сайт? А если сайт очень велик?

Дело в том, что операционная система Windows (как и многие другие операционные системы) хранит дату и время последнего изменения каждого файла. Сравнивая две даты, можно выяснить, какой файл новее. Скажем, если два файла на локальном диске и сервере имеют идентичные имена, но дата последнего изменения первого файла позднее, значит, этот файл новее. И, следовательно, его нужно скопировать на сервер, чтобы поддержать актуальность удаленной копии сайта.

Именно на таком принципе основан механизм *синхронизации* копий сайта. Dreamweaver проверяет даты разных копий файлов и принимает решение, какие из них нужно скопировать на сервер, а какие, наоборот, с сервера на локальный диск (бывает и так). Более того, основываясь на этом механизме, Dreamweaver копирует на сервер вновь созданные файлы и удаляет удаленные файлы локальной копии. Механизм прост и надежен, если, конечно, встроенные часы и календарь вашего компьютера установлены правильно.

Чтобы запустить синхронизацию файлов, выберите пункт **Synchronize** меню **Site** или контекстного меню. На экране появится диалоговое окно **Synchronize Files**, показанное на рис. 6.21.

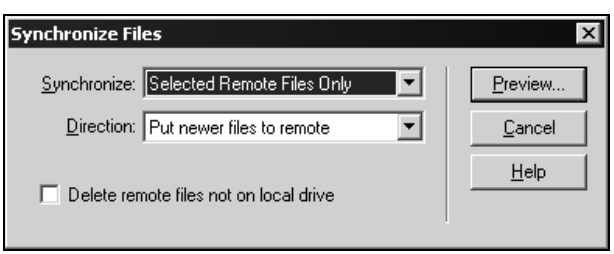


Рис. 6.21. Диалоговое окно **Synchronize Files**

Раскрывающийся список **Synchronize** позволяет задать, какие файлы вы хотите синхронизировать. Пункт **Selected Local File Only** позволяет синхронизировать только выбранные файлы из локальной копии. (Если в данный момент активна удаленная копия сайта, в списке показан пункт, который носит название **Selected Remote Files Only** и позволяет синхронизировать

выбранные файлы удаленной копии.) А пункт **Entire <*название сайта*> Site** позволяет синхронизировать весь сайт целиком.

А вот раскрывающийся список **Direction** мы рассмотрим подробнее. Здесь находятся три пункта, с помощью которых задается направление переноса файлов.

Пункт **Put newer files to remote** позволяет скопировать новые файлы на сервер, заменив ими устаревшие файлы удаленной копии. Если при этом включен флажок **Delete remote files not on local drive**, файлы удаленной копии, для которых не будут найдены соответствующие файлы локальной копии, будут удалены.

Пункт **Get newer files from remote** позволяет получить новые файлы с сервера, заменив ими устаревшие файлы локальной копии, это может быть полезно, если вы потеряли локальную копию сайта. Если при этом включен флажок **Delete local files not on remote server**, файлы локальной копии, для которых не будут найдены соответствующие файлы удаленной копии, будут удалены.

Пункт **Get and Put newer files** позволяет осуществить "взаимовыгодный обмен" новыми файлами. При этом будут синхронизированы и локальная и удаленная копии сайта. А если какому-либо файлу одной копии не будет найден соответствующий файл другой копии, он будет не удален, а скопирован в другую копию сайта.

Установив нужные параметры синхронизации (в основном, это будет, как мы полагаем, копирование новых файлов на сервер), нажмите кнопку **Preview**. Если вы хотите отказаться от синхронизации копий сайта, нажмите кнопку **Cancel**.

После запуска процесса синхронизации на экране появится окно списка синхронизируемых файлов, показанное на рис. 6.22. Здесь вы сможете указать, какие файлы необходимо синхронизировать, а какие — нет.

Как видите, большую часть этого окна занимает собственно список синхронизируемых файлов. Список организован в виде таблицы с тремя колонками.

- **Action** — действие, которое будет применено к файлу. Всего таких действий три:
  - **Put** — копирование на сервер;
  - **Get** — копирование с сервера на локальный диск;
  - **Delete** — удаление.

  Слева от названия действия находится флажок, по умолчанию включенный. Отключив его, вы сможете отменить действие над этим файлом (фактически, его синхронизацию).
- **File** — имя файла.
- **Status** — состояние синхронизации. Показывается после ее завершения.

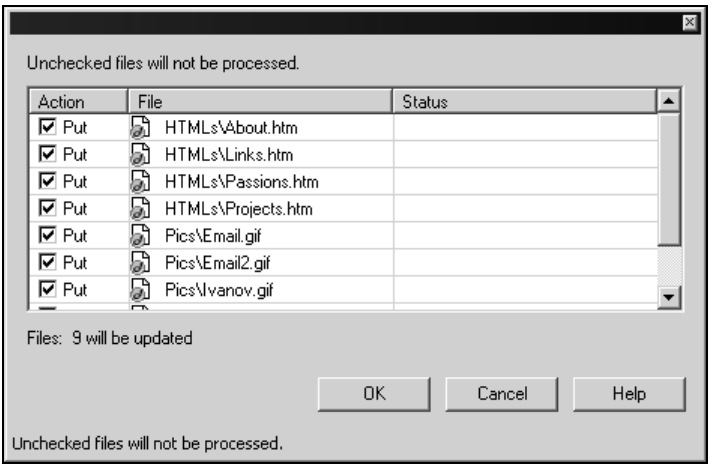


Рис. 6.22. Окно списка синхронизируемых файлов

Синхронизация запускается нажатием кнопки **OK**. Нажав кнопку **Cancel**, вы сможете от нее отказаться. Если какие-либо файлы в результате синхронизации должны быть удалены, Dreamweaver предупредит об этом; нажмите кнопку **OK** для удаления этих файлов и кнопку **Cancel** — для отказа от удаления.

По завершении синхронизации Dreamweaver выведет в то же самое окно результаты синхронизации, заполнив колонку **Status** (рис. 6.23). Закройте окно, нажав кнопку **Close**. Нажав кнопку **Save Log**, вы можете сохранить отчет о результатах синхронизации в виде обычного текстового файла.

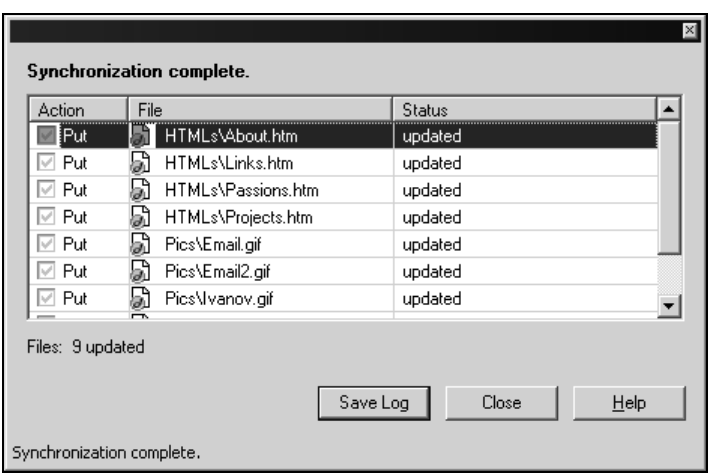


Рис. 6.23. Dreamweaver только что закончил синхронизацию

Закончив работу, не забудьте отключиться от FTP-сервера. Для этого просто щелкните еще раз кнопку **Connect to remote host**. Вместо этого вы можете выбрать пункт **Disconnect** в меню **Site** панели или нажать комбинацию клавиш <Ctrl>+<Alt>+<Shift>+<F5>.

Внимание!

Обязательно отключайтесь от FTP-сервера после того, как закончите работу с удаленной копией сайта! Помните, что на поддержание каждого соединения с клиентом FTP-сервер расходует ресурсы компьютера, на котором работает. А значит, другому клиенту этих ресурсов может не хватить. Уважайте других клиентов!

Работа с копиями сайта

Теперь давайте рассмотрим дополнительные возможности, предлагаемые Dreamweaver для работы с разными копиями сайта.

Выше говорилось, что можно не только публиковать отдельные файлы (и весь сайт) на сервере, но и, при необходимости, скопировать их с сервера на локальный диск. Для этого можно воспользоваться кнопкой **Get File(s)** (рис. 6.24); просто выберите нужные файлы в одном из списков и нажмите на данную кнопку. Кроме того, вы можете выбрать пункт **Get** меню **Site**, нажать комбинацию клавиш <Ctrl>+<Shift>+<D> или просто перетащить файлы из списка удаленной копии в список локальной.

Рис. 6.24. Кнопка **Get File(s)** инструментария панели **Site**

Dreamweaver также поможет вам узнать, какие файлы локальной или удаленной копии изменились после последней синхронизации. Если вы выберете пункт **Select Newer Local** меню **Edit** панели, будут выделены новые файлы локальной копии, если пункт **Select Newer Remote** — новые файлы удаленной копии. Вам остается только перенести эти файлы в другой (удаленный или локальный) список файлов.

Еще одна полезная возможность, предлагаемая Dreamweaver, — поиск выбранного в одном из списков файла в другом списке. Например, выбрав файл в списке файлов удаленной копии, вы можете найти соответствующий ему файл в списке файлов локальной копии. Достаточно выбрать пункт **Locate in Remote Site** для поиска выбранного файла в списке удаленной копии и пункт **Locate in Local Site** — для поиска в списке локальной копии. Оба эти пункта находятся в меню **Edit** и контекстном меню панели **Site**.

Дополнительные возможности работы с Web-страницами и Web-сайтами

Напоследок рассмотрим еще некоторые интересные возможности, предлагаемые Dreamweaver для работы со страницами и сайтами. Они помогут вам лучше организовать работу. Учтите, однако, что эти возможности доступны только для сайта и не доступны для простого набора страниц.

Примечания

В *главе 3* мы рассмотрели так называемые комментарии HTML, позволяющие разместить внутри HTML-кода страницы произвольный текст. Обычно Web-дизайнеры создают таким образом небольшие заметки, примечания, напоминания, что нужно сделать, и т. п. В частности, мы с вами поместили в тексте главной страницы default.htm небольшое напоминание ("незабудку") с тем, чтобы не забыть сделать гиперссылки на другие страницы нашего сайта. Такие комментарии не отображаются Web-обозревателем, а в окне Dreamweaver выводятся в виде небольшого значка, практически не влияющего на внешний вид страницы.

Всем хороши HTML-комментарии, но и у них есть недостатки. Первый недостаток: код такого комментария увеличивает размер Web-страницы, не принося при этом никакой пользы конечному пользователю (такой вот каламбур). Второй недостаток: после выполнения всех работ, о которых напоминает нам комментарий-"незабудка", последний нужно удалить. А поскольку Web-дизайнер — человек, а не программа, то он имеет досадное свойство забывать сделать элементарнейшие вещи. Как, например, забыли это сделать и мы.

Но Dreamweaver приходит на помощь забывчивым Web-дизайнерам. Он предоставляет возможность, называемую *примечаниями*. Вы можете создавать примечания как для отдельной страницы, так и для любого размещенного на ней внедренного элемента. Эти примечания сохраняются в файлах в специальных подпапках корневой папки сайта, поэтому вам перед тем, как использовать примечания, необходимо создать сам сайт.

Давайте посмотрим, что же это за чудо техники и как его можно использовать.

Но, прежде всего, давайте проверим настройки. Откройте диалоговое окно **Site Definition** для нашего сайта Sample site 1. (Выберите пункт **Edit Sites** меню **Site** панели **Site**, потом в списке сайтов появившегося диалогового окна **Edit Sites** выберите пункт **Sample site 1** и нажмите кнопку **Edit**.) Переключитесь на вкладку **Design Notes**. Ее содержимое показано на рис. 6.25.

Удостоверьтесь, что флажок **Maintain Design Notes** включен (он должен быть включен по умолчанию). Если он выключен, Dreamweaver не будет поддерживать примечания.

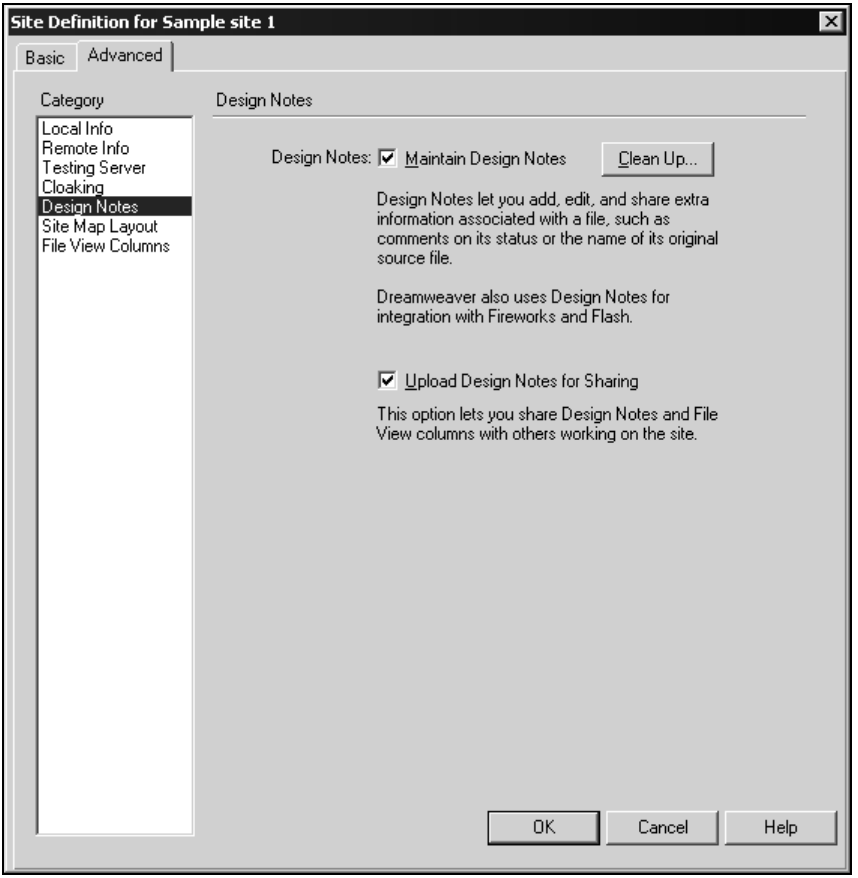


Рис. 6.25. Диалоговое окно **Site Definition** (вкладка **Design Notes**)

Теперь обратите внимание на флажок **Upload Design Notes for Sharing**. Если он включен (а он также включен по умолчанию), Dreamweaver будет отправлять файлы, где хранятся примечания, на сервер вместе с другими файлами сайта. В принципе, это может быть удобно: вам не нужно хранить в своем жестком диске файлы локальной копии сайта, при необходимости что-то исправить вы можете просто загрузить их с сервера. Однако если вы не хотите, чтобы кто-то рылся в ваших примечаниях, можете отключить этот флажок. (Хотя посетители вашего сайта и так не смогут добраться до примечаний. А если все-таки доберутся, виноват будет администратор Web-сервера.)

Если вы сделали какие-либо изменения в настройках сайта, нажмите кнопку **OK**; в противном случае нажмите кнопку **Cancel**. Закройте также окно **Edit Sites**, нажав кнопку **Done**.

Прежде всего, откройте страницу default.htm. Именно к ней мы и привяжем наши первые примечания.

Выберите пункт **Design Notes** меню **File** или аналогичный пункт **Design Notes for Page** контекстного меню страницы. На экране появится диалоговое окно **Design Notes**, показанное на рис. 6.26. Переключитесь на вкладку **Basic Info**.

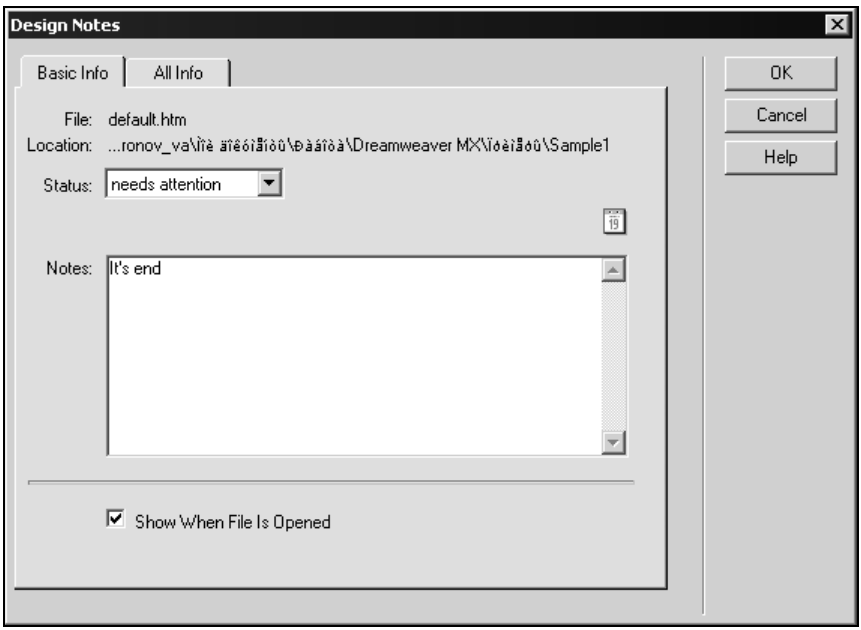


Рис. 6.26. Диалоговое окно **Design Notes** (вкладка **Basic Info**)

С помощью раскрывающегося списка **Status** выбирается состояние, на котором находится текущая страница. Здесь доступны следующие пункты:

- **draft** — черновик;
- **revision1-revision3** — ревизия 1—3;
- **alpha** — альфа-версия (предварительная версия, предназначенная для тестирования самими разработчиками сайта;
- **beta** — бета-версия (предварительная версия, предназначенная для широкого тестирования конечными пользователями);
- **final** — окончательная версия;
- **need attention** — странице следует уделить внимания.

Само примечание вводится в текстовое поле **Notes**. К сожалению, из-за ошибки Dreamweaver не поддерживает русские предложения с пробелами, так что придется писать по-английски или по-русски с транслитерацией.

Возле правого верхнего угла поля **Notes** находится небольшая кнопка, имеющая вид страницы календаря. Если вы щелкнете по ней мышью, в содержимое поля **Notes** будет вставлена сегодняшняя дата.

Если вы хотите, чтобы введенные вами примечания показывались при открытии этой Web-страницы, включите флажок **Show When File Is Opened**. Автор советует вам включить его, чтобы не пропустить важные примечания, введенные вами или вашими коллегами.

На вкладке **All Info** (рис. 6.27) вы можете ввести более подробную информацию о данной странице. Эта информация вводится в виде набора строк, имеющих формат:

```
<Имя>=<Значение>
```

В частности, несколько таких строк вы можете видеть в списке **Info** на рис. 6.27.

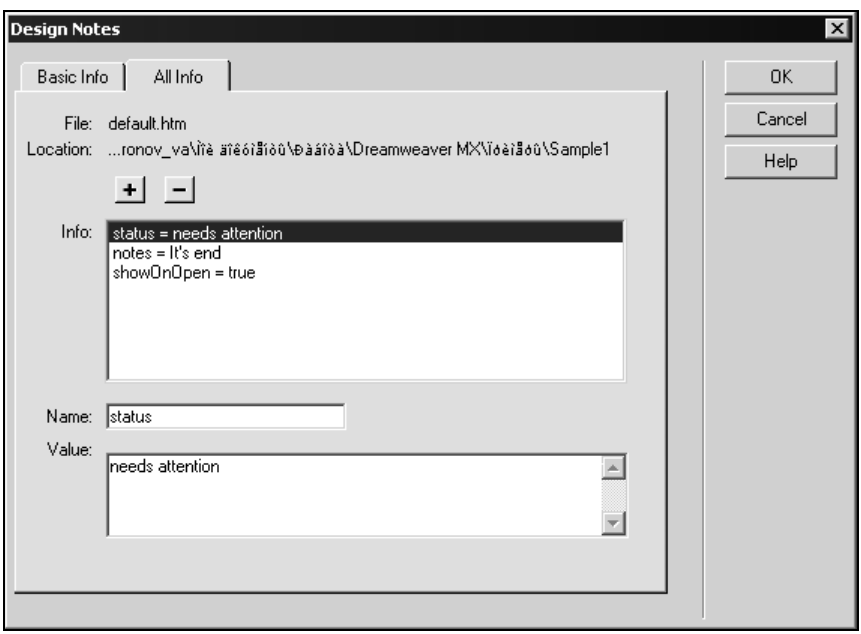


Рис. 6.27. Диалоговое окно **Design Notes** (вкладка **All Info**)

Чтобы ввести новую строку, нажмите кнопку со знаком "плюс", введите в поле ввода **Name** левую часть строки ("имя"), а в поле **Value** — правую ("значение").

Чтобы изменить существующую строку, выберите ее в списке **Info**, после чего в поле ввода **Name** измените ее левую часть (имя), а в поле **Value** — правую (значение).

Чтобы удалить ненужную строку, выберите ее в списке **Info** и нажмите кнопку со знаком "минус".

Закончив ввод примечаний, нажмите кнопку **OK**.

Теперь вы можете проверить примечания. Для этого снова откройте страницу default.htm. Если вы включили флажок **Show When File Is Opened**, окно **Design Notes** (см. рис. 6.26) появится сразу же после открытия этой страницы.

Теперь рассмотрим, как можно привязывать примечания к внедренным элементам страницы (изображениям, фильмам Flash и т. п.). Это важное условие — элемент страницы должен быть сохранен в отдельном файле. Привязать примечание, скажем, к таблице или гиперссылке вы не сможете. (Но вы можете привязать соответствующие примечания к самой Web-странице.)

Откроем страницу About.htm, где есть изображение, к которому можно добавить примечание, — портрет нашего мифического героя. Выделим портрет и... А вот теперь будьте внимательны: чтобы привязать к внедренному элементу примечание, нужно воспользоваться пунктом **Design Notes** контекстного меню графического изображения. Пункт **Design Notes** меню **File** позволяет привязать примечание только к самой странице.

Диалоговое окно **Design Notes** для внедренного элемента страницы ничем не отличается от окна, показанного на рис. 6.26. И это правильно: незачем лишний раз запутывать пользователя множеством разнокалиберных окон, выполняющих схожие задачи. К сожалению, флажок **Show When File Is Opened** в этом случае работать не будет — даже если вы его включите, окно **Design Notes** при открытии страницы не появится.

Введите что-нибудь в качестве примечания и нажмите кнопку **OK**. Закройте все окна документа Dreamweaver, разверните панель **Site** на весь экран и обратите внимание на список файлов локальной копии (рис. 6.28).

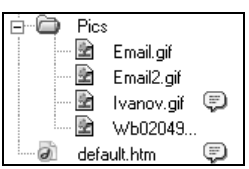


Рис. 6.28. Значки примечаний в списке файлов панели **Site**

Вы видите, что в столбце **Notes** в строках, описывающих файлы default.htm и Ivanov.gif, появились некие значки. Это значки примечаний; они сигнализируют, что к таким-то файлам привязаны примечания. Вы можете дважды щелкнуть по такому значку мышью, и на экране появится окно **Design Notes**.

Выше мы описали, как задаются примечания в окне документа. То же самое вы можете сделать и в панели **Site**. То есть вы можете либо выбрать пункт

Design Notes меню **File**, либо воспользоваться аналогичным пунктом контекстного меню панели. Но проще всего дважды щелкнуть по столбцу **Notes** списка файлов, даже если там и нет значка примечания.

Активы

Другой замечательной возможностью, предоставляемой Web-дизайнеру Dreamweaver, являются *активы*. Это список всех внедренных элементов, цветов и гиперссылок, использованных вами на всех страницах вашего сайта. Активы помогут вам быстро найти нужный элемент при одном условии: вы уже создали сайт и использовали этот элемент на ваших страницах.

Активы отображаются в панели **Assets** (рис. 6.29), которую вы можете найти в доке. Если же ее там нет, выберите пункт **Assets** меню **Window** или нажмите клавишу <F11>.

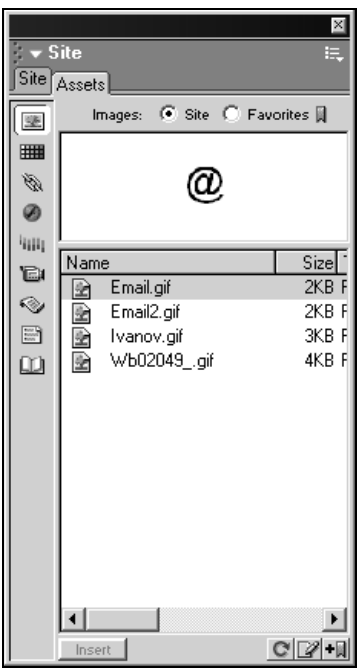


Рис. 6.29. Окно активов

Большую часть данного окна занимает список *элементов*, относящихся к той или иной *категории*. Вы можете выбрать любой элемент в списке; при этом он будет показан в расположенной над списком панели предварительного просмотра. На рис. 6.29 вы видите, что в списке выбран и в панели предварительного просмотра показан файл Email.gif. В нижней части окна

активов находятся четыре кнопки, выполняющие различные действия над выбранными в списке элементами.

Чтобы сменить категорию выводимых в списке элементов, необходимо щелкнуть по одному из значков, расположенных на вертикальной серой полосе слева от списка. Перечислим их сверху вниз и расскажем, за какую категорию какой из них отвечает:

- графические изображения (имеются в виду "классические" форматы Интернета — GIF и JPEG и "новичок" PNG);
- цвета;
- гиперссылки;
- изображения и фильмы Flash;
- изображения и фильмы Shockwave;
- фильмы "классических" форматов, в частности AVI и MOV;
- сценарии, написанные на языках JavaScript и VBScript (будут рассмотрены в *главе 13*);
- шаблоны (будут рассмотрены в *главе 9*);
- элементы библиотеки Dreamweaver (будет рассмотрена в конце этой главы).

Пощелкайте мышью на этих значках и посмотрите, что получится. Как видите, Dreamweaver ведет строжайший учет всему, что вы помещаете на свои страницы.

Но как же использовать это неописуемое богатство? Очень просто.

Графические изображения, фильмы и гиперссылки вы можете просто перетащить мышью на свою страницу, в то место, где они должны находиться. При этом Dreamweaver сам вставит в нужное место соответствующий HTML-код. Этого же можно достичь, выбрав пункт **Insert** (для цветов — пункт **Apply**) контекстного меню списка или дополнительного меню панели **Assets** или нажав кнопку **Insert** в левом нижнем углу этой панели.

Dreamweaver также предусматривает возможность изменить графическое изображение формата GIF, JPEG, PNG, Flash или Shockwave, если у вас установлено соответствующее приложение (а именно, Flash и Shockwave для "своих" форматов и Fireworks — для форматов GIF, JPEG и PNG). Для этого достаточно дважды щелкнуть по нужной позиции списка или выбрать пункт **Edit** контекстного или дополнительного меню. Также можно нажать кнопку **Edit** в правом нижнем углу панели **Assets** (рис. 6.30).

Рис. 6.30. Кнопка **Edit** панели **Assets**

Также для внедренных объектов поддерживается поиск соответствующего файла и копирование его в другой сайт. Чтобы найти файл, где сохранено изображение или фильм, выберите пункт **Locate in Site** контекстного меню, и Dreamweaver активизирует окно сайта с выделенным в списке файлом. Чтобы скопировать файл, соответствующий внедренному элементу, в другой сайт, выберите нужный пункт в подменю **Copy to Site** контекстного меню. Запомните, что при этом Dreamweaver просто копирует файл в другой сайт, но никак не связывает его с другими файлами — об этом вы должны позаботиться сами.

Цвета вы можете применить к любому тексту. Для этого достаточно просто перетащить необходимый цвет на выделенный текст. Но, вероятно, значительно удобнее будет выбрать пункт **Apply** в контекстном меню или нажать кнопку **Apply** внизу окна активов.

Как видите, иметь под рукой активы очень удобно. Не надо вспоминать, как называется файл графического изображения, вставленный вами на другую страницу неделю назад. Но если у вас достаточно большой сайт, обилие позиций в списке окна активов начинает раздражать. Хочется отобрать некоторые из них, чтобы работать с ними постоянно, а к остальным возвращаться только время от времени. Создать список избранных элементов, этаких "любимчиков" или, по-английски, "favorites".

Список избранных элементов

И такую возможность поддерживает Dreamweaver. А именно — вы можете создать список *избранных элементов*, своего рода "любимчиков".

Поместить нужный элемент активов в список избранных очень просто: достаточно выделить необходимую позицию в списке панели **Assets** и выбрать пункт **Add to Favorites** контекстного или дополнительного меню или щелкнуть кнопку **Add to Favorites** внизу окна активов (рис. 6.31). Dreamweaver выдаст вам небольшое предупреждение, сообщающее, что выбранный элемент будет добавлен в список ваших "любимчиков". И, предупреждая ваш вопрос, подскажет, где его искать. Закройте окно-предупреждение, нажав кнопку **OK**. Последуйте доброму совету: включите переключатель **Favorites** в верхней части панели **Assets**. Как видите, панель **Assets** после этого практически не изменится (рис. 6.32) — изменится только содержимое списка элементов. Также в ней останутся панель предварительного просмотра, значки категорий и набор кнопок, выполняющих разные задачи.

Рис. 6.31. Кнопка **Add to Favorites** панели **Assets**

Вы можете помещать на страницы выбранные элементы простым перетаскиванием, выбрав пункт **Insert** (или пункт **Apply** — для цветов) в контекст-

ном или дополнительном меню или нажав одноименную кнопку в левом нижнем углу панели. Вы можете открывать элементы для редактирования, дважды щелкнув нужную позицию в списке, выбрав пункт **Edit** в контекстном или дополнительном меню или нажав кнопку **Edit** в правом нижнем углу панели (см. рис. 6.30). Также вы можете искать соответствующие файлы в текущем сайте и копировать их в другие сайты — вы знаете, как это делается.

На этом сходство заканчивается. Далее начинаются различия.

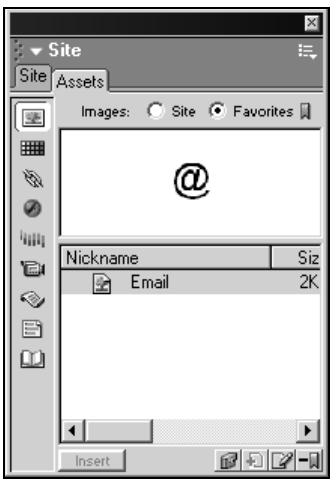


Рис. 6.32. Панель **Assets**, отображающая список избранных элементов

Элементам, помещаемым вами в "любимчики", присваиваются *"прозвища"* (по-английски — nicknames). Изначально они совпадают с именем файла без расширения, что не всегда удобно. Вы можете дать элементу вразумительное "прозвище", щелкнув по нему, введя новое "прозвище" в появившемся поле ввода и нажав клавишу <Enter>. Аналогичную функцию выполняет пункт **Edit Nickname** контекстного или дополнительного меню.

Список избранных элементов может быть организован наподобие файловой системы, т. е. в виде папок и вложенных в них файлов. Создать папку можно, выбрав в контекстном или дополнительном меню пункт **New Favorites Folder** или нажав кнопку **New Favorites Folder** в правом нижнем углу панели (рис. 6.33). В списке появится новая папка, причем вместо ее имени будет показано поле ввода, в которое вы сможете ввести имя новой папки, после чего не забудьте нажать клавишу <Enter>.

Рис. 6.33. Кнопка **New Favorites Folder** панели **Assets**
(режим отображения списка избранных элементов)

Создав папку, вы можете переместить в нее любые элементы простым перетаскиванием, как вы работаете с файлами в окне Проводника или в панели **Site**. Вы также можете вкладывать папки в другие папки и таким образом создавать многоуровневую структуру. Посмотрите, что получилось у нас — рис. 6.34.

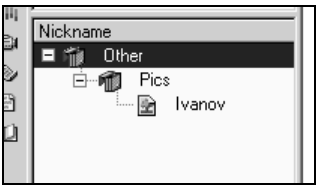


Рис. 6.34. Многоуровневая структура папок в списке избранных элементов

Ну и, разумеется, Dreamweaver предоставляет вам возможность удалить любой элемент, "впавший в опалу", из списка избранных. Для этого выделите элемент и нажмите клавишу <Del>, выберите пункт **Remove from Favorites** контекстного или дополнительного меню или нажмите на одноименную кнопку в правом нижнем углу панели (рис. 6.35). Элемент списка избранных удаляется без предупреждения, поэтому будьте внимательны. Таким же образом вы можете удалить любую папку, только учтите, что папка удаляется сразу со всем содержимым. Поэтому, если какие-либо файлы, входящие в эту папку, вам еще нужны, сначала переместите их в какое-нибудь другое место.

Рис. 6.35. Кнопка **Remove from Favorites** панели **Assets** (режим отображения списка избранных элементов)

Чтобы вернуться в список активов, включите переключатель **Site** в верхней части панели **Assets**.

Библиотека

Не кажется ли вам, что список избранных элементов — всего лишь полумера? Конечно, полезно иметь несколько "любимых" изображений, гиперссылок, цветов и фильмов, чтобы быстро помещать их на страницы. Но иногда очень нужно сохранить таким образом не отдельный элемент, а целый кусок текста. Может ли это Dreamweaver?

Конечно, может. Для таких случаев он предоставляет в ваше распоряжение так называемую *библиотеку*.

Библиотека доступна из панели **Assets**. Вызовите эту панель (проще всего нажать клавишу <F11>) и щелкните по самому нижнему из значков катего-

рий. Все — теперь вы просматриваете содержимое своей пока еще пустой библиотеки (рис. 6.36).

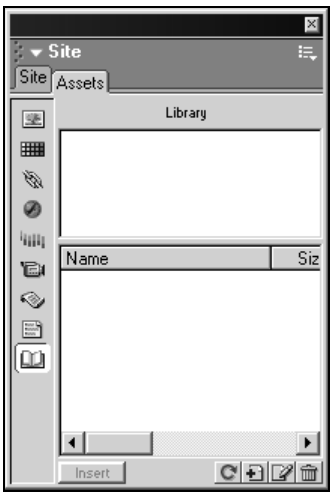


Рис. 6.36. Список содержимого библиотеки

Поместить новый элемент в библиотеку очень просто. Достаточно выделить нужный фрагмент Web-страницы и перетащить его в окно библиотеки. Откройте страницу default.htm, выделите заголовок, первый абзац и горизонтальную линию, отделяющую его от остального текста, и добавьте выделенный фрагмент в библиотеку. Вы также можете выбрать пункт **Add Object to Library** в подменю **Library** меню **Modify** или нажать комбинацию клавиш <Ctrl>+<Shift>+<B>. В списке появится новая позиция, вместо имени которой будет показано поле ввода. Введите туда имя вновь созданного элемента библиотеки, например "Header", и нажмите клавишу <Enter>. Вот и все.

Содержимое выбранного в списке элемента библиотеки вы можете просмотреть в уже знакомой вам панели предварительного просмотра. Правда, чтобы просмотреть в этой панели достаточно большой элемент (например, наш заголовок "Header"), вам придется прокручивать его во всех направлениях.

Вот мы и положили начало нашей библиотеке. Теперь вы можете вставить этот элемент куда угодно, перетащив его мышью на страницу, выбрав пункт **Insert** контекстного или дополнительного меню или нажав кнопку **Insert** в левом нижнем углу панели.

Но что нам может дать использование библиотеки, кроме облегчения вставки повторяющихся фрагментов текста? А вот что. Дело в том, что элемент библиотеки — это нечто большее, чем просто кусок текста, приправленный

HTML-тегами и внедренными элементами. Вы можете изменить любой элемент библиотеки — и Dreamweaver обновит все содержащие его Web-страницы. Это достигается тем, что Dreamweaver помечает HTML-код вставленного библиотечного элемента особыми комментариями.

Представляете, что это дает! Вы можете создавать страницы, скажем, с одинаковым заголовком, который будете вставлять из библиотеки. И при необходимости что-то в нем изменить вы просто меняете заголовок, находящийся в библиотеке, и даете Dreamweaver команду повторить это изменение на всех соответствующих страницах.

Давайте, например, изменим цвет шрифта слова "Здравствуйте!". Выделите наш единственный элемент библиотеки и либо выберите пункт **Edit** контекстного или дополнительного меню, либо нажмите на кнопку **Edit** в правом нижнем углу панели (см. рис. 6.30). На экране появится новое окно документа, содержащее элемент библиотеки.

Измените цвет слова "Здравствуйте!" на темно-красный и закройте окно. Dreamweaver спросит вас, сохранять ли этот элемент; нажмите кнопку **Yes**. После этого на экране появится диалоговое окно **Update Library Items**, в точности совпадающее с уже знакомым вам окном **Update Files** (см. рис. 6.14). Нажмите кнопку **Update** и закройте появившееся на экране другое диалоговое окно, нажав кнопку **Close**. Если вы теперь откроете страницу default.htm, то увидите, что цвет слова "Здравствуйте!" изменился.

Что еще вы можете сделать с элементами библиотеки? Да практически то же, что и с избранными элементами, т. е.:

- переименовать, щелкнув по имени элемента или выбрав пункт **Rename** контекстного или дополнительного меню;
- удалить, нажав клавишу <Del>, выбрав пункт **Delete** контекстного или дополнительного меню или нажав на кнопку **Delete** в правом нижнем углу панели (рис. 6.37). Dreamweaver выдаст предупреждение; нажмите кнопку **Yes** для удаления элемента и кнопку **No** для отказа от удаления При этом элементы, помещенные на страницы, будут сохранены;
- найти на сайте соответствующий файл, выбрав пункт **Locate in Site** контекстного меню. Элементы библиотеки сохраняются в файлах под введенными вами именами и расширением lbi в подпапке Library корневой папки сайта;
- скопировать этот файл в библиотеку другого сайта, выбрав соответствующий пункт подменю **Copy to Site** контекстного меню.

Рис. 6.37. Кнопка **Delete** панели **Assets** (режим отображения содержимого библиотеки)

Но, кроме уже знакомых вам действий над файлами, библиотека предоставляет и новые. Это обновление Web-страниц, содержащих библиотечные элементы, после их изменения.

Для того чтобы обновить открытую в активном окне документа страницу, выберите пункт **Update Current Page** контекстного меню элемента библиотеки, дополнительного меню панели или подменю **Library** меню **Modify** главного окна. Dreamweaver сразу же обновит ее.

Другое дело, если вам нужно обновить страницы во всем сайте. Для этого выберите пункт **Update Site**. На экране появится диалоговое окно **Update Pages**, показанное на рис. 6.38. Это окно уже вам знакомо — Dreamweaver выводит его при каждом удобном случае.

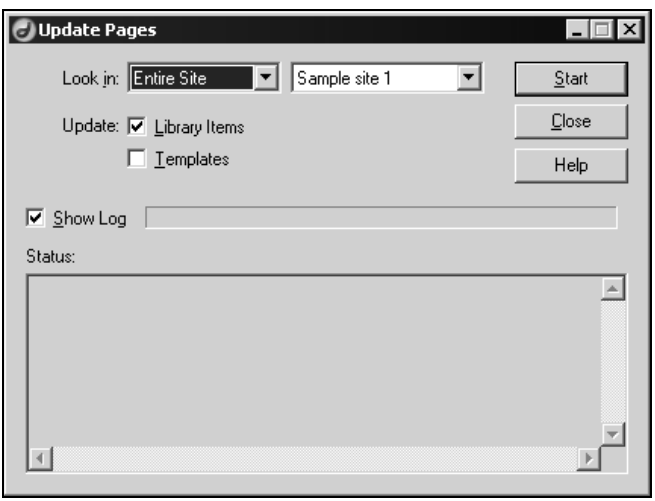


Рис. 6.38. Диалоговое окно **Update Pages**

Если же вам необходимо обновить все страницы сайта, выберите в раскрывающемся списке **Look in** пункт **Entire Site**, а в раскрывающемся списке, расположенном справа от него, — название сайта. Если требуется обновить только те страницы, на которых вы поместили тот или иной библиотечный элемент, выберите в том же раскрывающемся списке **Look in** пункт **Files That Use**, а в раскрывающемся списке, расположенном справа от него, — имя необходимого библиотечного элемента. Проверьте, включен ли флажок **Library Items** в группе **Update**. И нажмите кнопку **Start**.

Если вы оставили включенным флажок **Show Log**, Dreamweaver будет отображать ход процесса просмотра и обновления файлов в расположенном в нижней половине окна текстовом поле. После того как процесс обновления закончится, закройте это окно, нажмите кнопку **Close**.

Иногда бывает необходимо преобразовать элемент библиотеки, помещенный на Web-страницу, в обычный фрагмент кода HTML. Для этого выделите его в окне документа и выберите в контекстном меню пункт **Detach from Original**. После этого Dreamweaver выдаст вам небольшое окно-предупреждение; нажмите кнопку **Yes**, чтобы открепить элемент от библиотеки, и кнопку **No**, если вы почему-то передумали делать это.

Дополнительные возможности поиска и замены

В *главе 2* мы узнали о мощных возможностях текстового поиска и замены, предоставляемых Dreamweaver. Однако мы рассмотрели их не все.

Взгляните на рис. 2.27. Видите раскрывающийся список **Find In**? А ведь мы о нем так ничего и не сказали.

Этот раскрывающийся список задает файлы, где будет выполняться поиск и замена. Здесь доступны четыре пункта:

- **Current Document** — поиск и замена выполняются в тексте страницы, открытой в активном окне. Этот пункт выбран по умолчанию;
- **Entire Local Site** — поиск и замена осуществляются в файлах всего сайта;
- **Selected Files In Site** — поиск и замена производятся в выбранных в окне сайта файлах;
- **Folder** — поиск и замена в файлах, находящихся в заданной папке. Если выбран этот пункт, справа от раскрывающегося списка появляется поле ввода, в котором вы должны указать имя папки. Вы также можете щелкнуть на значке папки, расположенном справа от поля ввода, и выбрать нужную папку в появившемся на экране диалоговом окне.

При выполнении поиска Dreamweaver будет последовательно открывать страницы, где встретилась введенная подстрока. Вы также можете выполнить глобальный поиск, нажав кнопку **Find All**. Результаты поиска будут отображены в уже знакомой вам панели **Search** (см. рис. 2.30).

Другие возможности

Кроме примечаний, активов и библиотеки, Dreamweaver таит еще много приятных для Web-дизайнеров сюрпризов. И со временем мы их изучим. Однако будьте готовы к тому, что все они потребуют, чтобы вы прежде создали сайт, а уж потом приступали к разработке страниц.

В частности, в *главе 3* мы изучили так называемые HTML-стили — предопределенные наборы установок параметров текста, которые вы можете использовать для его форматирования. Так вот, HTML-стили требуют, чтобы вы сначала создали сайт. А все потому, что записи о заданных HTML-стилях сохраняются в виде файлов в корневой папке сайта.

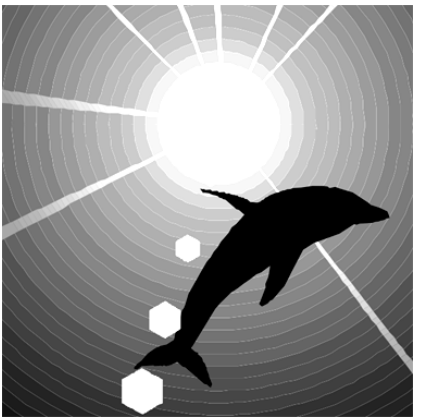
В дальнейшем, когда мы будем рассматривать табличный дизайн Web-страниц, вы познакомитесь с шаблонами. Чтобы использовать шаблоны, нам опять-таки понадобится создать сайт. Так что, в Dreamweaver без сайта никуда.

Что дальше?

Вот мы и создали свой первый Web-сайт. И не только создали, но и опубликовали в Сети. Прославились на весь мир, так сказать...

Казалось бы, чего еще желать. Однако правду говорят в народе, что первый блин — всегда комом. Ну, не сказать, чтобы наш первый сайт получился таким уж неудачным, но как-то он... не очень... И при взгляде на шикарные Web-творения, присутствующие в Интернете по соседству с нашим более чем скромным сайтиком, просто зависть берет. Ну, неужели мы так не можем?!

Можем, конечно. И сделаем! Но сначала изучим профессиональные способы Web-дизайна. И начнем с классических — фреймовых, использующих одну из самых интересных возможностей, предоставляемых HTML, — фреймы.



ЧАСТЬ II

Занимаемся профессиональным Web-дизайном

Глава 7. Фреймы

Глава 8. Табличный дизайн

Глава 9. Использование шаблонов

Глава 10. Каскадные таблицы стилей

ГЛАВА 7

Фреймы

Итак, простейший сайт мы создали. Первый шаг в Web-дизайне нами сделан. Пора двигаться дальше. В этой главе мы создадим новую версию нашего сайта с использованием фреймов. Поэтому прежде, чем мы начнем изучать фреймы и работу с ними, создайте для нового сайта новую папку. Назовите ее, скажем, Sample2. Внутри нее создайте подпапки HTMLs и Pics, как в старой версии сайта.

Но и старую папку со старой версией сайта пока оставьте — она нам еще пригодится. Мы позаимствуем из нее все графические изображения и большинство текстов.

Введение во фреймы

Так что же такое фреймы? Давайте выясним это.

Фрейм и набор фреймов

Откройте какую-нибудь из наших Web-страниц, созданных в предыдущих главах. Посмотрите на нее. Что она собой представляет? Правильно, текстовый документ. Обычный текстовый документ, возможно, "разбавленный" графикой и таблицами. Но, так или иначе, все это помещается в основном "потоке" текста и лишь изредка выбивается из него (например, изображения, выровненные влево или вправо; подробнее *см. главу 4*). Все, абсолютно все "течет" в этом могучем "потоке": и *основное (полезное) содержимое страницы*, и *сведения об авторских правах*, и *заголовок*, и *набор гиперссылок*.

А теперь представьте себе, что окно Web-обозревателя разбито на несколько меньших "форточек" и в каждой "форточке" загружается своя Web-страница. В одной — набор гиперссылок, в другой — заголовок, в третьей — сведения об авторских правах, а в самой большой — полезный текст, ради которого

и был создан этот сайт. Причем "форточки" эти ведут себя как независимые окна Web-обозревателя: у них даже есть свои полосы прокрутки.

Когда вы щелкаете по гиперссылке, нужная страница сразу загружается в соответствующую ей "форточку". При этом она содержит только полезный текст — набор гиперссылок, заголовок сайта и сведения об авторских правах находятся на страницах, отображаемых в других "форточках".

Схематично все это можно представить так, как показано на рис. 7.1.

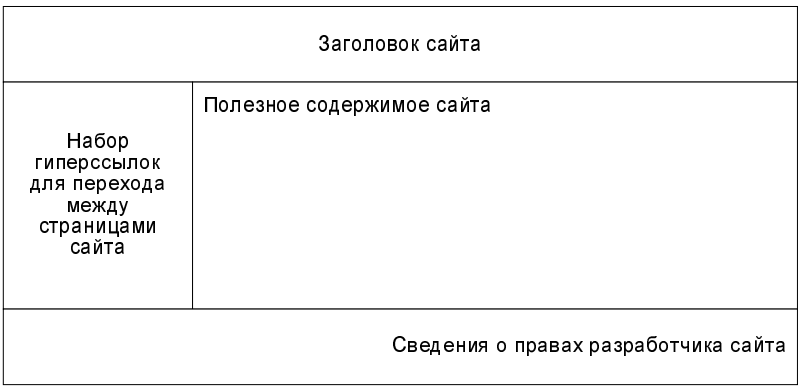


Рис. 7.1. Классический набор из четырех фреймов

Это очень удобно. Когда посетитель сайта щелкает по гиперссылке, обновляется только содержимое одной "форточки" (в которой отображается основное содержимое). Такой подход дает Web-дизайнеру два главных преимущества. Во-первых, страницы загружаются значительно быстрее, т. к. содержат только полезную информацию, без набора гиперссылок, заголовка, сведений об авторских правах и прочей вспомогательной информации. Во-вторых, не нужно всю эту дополнительную информацию дублировать на каждой странице сайта — достаточно создать специальные странички, содержащие ее части, а потом загрузить их в соответствующие "форточки". А если нам нужно изменить, скажем, адрес какой-нибудь гиперссылки, мы просто меняем его на одной-единственной страничке вместо того, чтобы просматривать все страницы сайта, на которых мы поместили данную гиперссылку.

Отлично! Теперь давайте назовем вещи своими именами. "Форточки", иначе говоря, части, на которые делится окно Web-обозревателя, — это и есть *фреймы* (от англ. *frame* — кадр). А набор таких частей назовем *набором фреймов* (по-английски — *frameset*). Способ же разработки Web-сайтов с использованием фреймов называется *фреймовым дизайном.*

Как же формируется набор фреймов? Что для этого необходимо?

Для этого нужна специально написанная Web-страница, в которой, собственно, и задаются параметры набора фреймов и самих фреймов. Там же

указываются размеры фреймов и интернет-адреса страниц, которые будут в них загружены. Эта Web-страница практически всегда является главной, т. е. загружается при наборе адреса сайта в строке адреса Web-обозревателя.

Запомните, что такая страница не должна содержать никаких полезных данных, никакого текста — только определение набора фреймов. Но даже если бы она и содержала какой-то текст, Web-обозреватель все равно не отобразил бы его, т. к. он не знает, где его отобразить — все свободное пространство его окна будет задано набором фреймов.

Фреймы — достаточно давнее дополнение, введенное в язык HTML. Однако высокоученые головы из комитета W3C только недавно включили их в HTML-стандарт. В результате такой политики неприятия весьма полезного нововведения фреймы на сегодняшний день стали, наверное, самым "нестандартным" из элементов HTML. Старые программы Web-обозревателей поддерживают их по-разному, хотя более новые их версии и пытаются соблюдать стандарты.

А Web-дизайнеры используют фреймы уже достаточно давно. И применяют их настолько часто, что среди снобов от Web-дизайна они уже стали считаться банальностью. И зря!

Схемы наборов фреймов

Теперь поговорим о различных способах разделения окна Web-обозревателя на отдельные фреймы и об информации, помещаемой в эти самые фреймы, иначе говоря, о разных *схемах* фреймового дизайна. Вообще, сколько существует Web-дизайнеров, столько существует и схем фреймового дизайна, так что перечислять их все бесполезно. Однако за довольно долгую историю WWW было разработано несколько устоявшихся схем наборов фреймов, прошедших проверку временем и использующихся в подавляющем большинстве сайтов, построенных по принципам фреймового дизайна. Эти устоявшиеся схемы мы и рассмотрим.

Итак, что должен содержать типичный набор фреймов? Как мы уже выяснили, это набор гиперссылок для перемещения по страницам сайта, заголовок сайта, основное (полезное) содержимое и сведения об авторских правах. Как они размещаются в наборе фреймов, было показано на рис. 7.1.

Давайте рассмотрим эти фреймы подробнее.

Насчет полезного содержимого сайта все ясно. Ради него и создается сайт, поэтому оно обязательно должно присутствовать. Под него выделяют самый большой фрейм набора. Иногда, впрочем, полезное содержимое разделяется на части и "разбрасывается" по нескольким более мелким фреймам, но мы не будем рассматривать здесь такие специфические случаи. Запомните только, что под фрейм с основным содержимым сайта выделяется максимум места в окне Web-обозревателя, и это логично.

Набор гиперссылок — второй по важности фрейм в схеме. Так как Web-сайт отличается от простого набора Web-страниц своей взаимосвязанностью, без гиперссылок, связывающих разнородные страницы в единое целое, не обойтись. Почти всегда набор гиперссылок помещается в вертикальном фрейме, расположенном слева от основного содержимого, вдоль левой границы окна Web-обозревателя. Иногда, правда, набор гиперссылок переносят вправо, как показано на рис. 7.2. А порой он "вытягивается" по горизонтали сверху или снизу (рис. 7.3) основного содержимого.

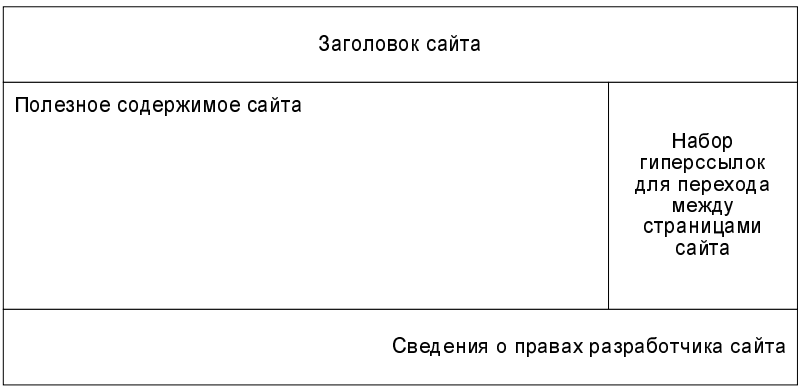


Рис. 7.2. Набор фреймов с набором гиперссылок, расположенным справа

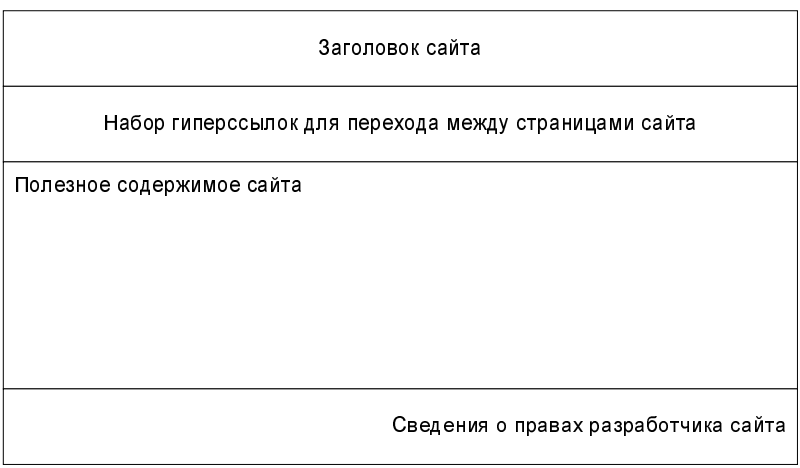


Рис. 7.3. Набор фреймов с набором гиперссылок, расположенным сверху

Автору также встречались наборы фреймов с двумя разными наборами гиперссылок, расположенными сверху и снизу (рис. 7.4) от основного содержимого. Верхний набор гиперссылок указывал на страницы с основным со-

держимым сайта, а нижний — на страницы с дополнительным, неглавным содержимым (сведения о контакте, карта сайта и пр.).

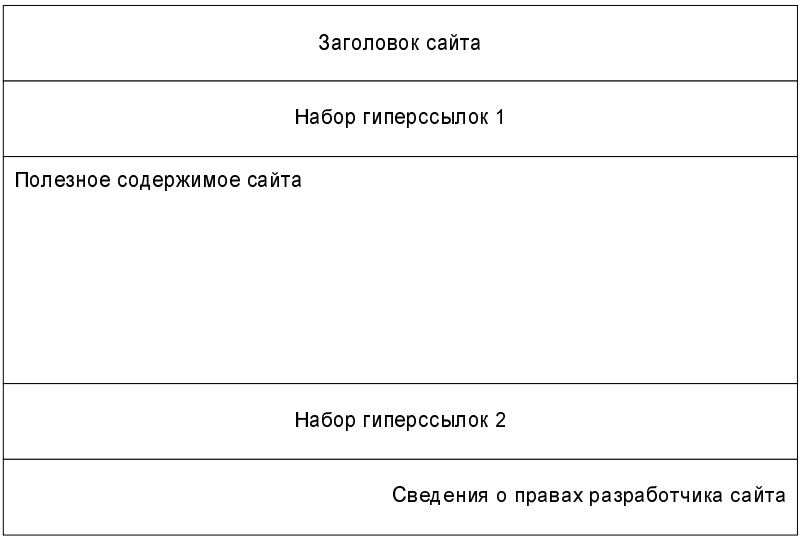


Рис. 7.4. Набор фреймов с двумя наборами гиперссылок, расположенными сверху и снизу

Заголовок сайта — тоже достаточно важная вещь. Можно сказать, это лицо сайта. Поэтому его часто помещают в горизонтальный фрейм, расположенный сверху, вдоль верхней стороны окна Web-обозревателя. Таким образом он хорошо виден, но и не бросается в глаза, если специально на него не смотреть. Иногда его разделяют на два фрейма, в одном из которых находится логотип сайта, а в другом — текст названия (рис. 7.5).

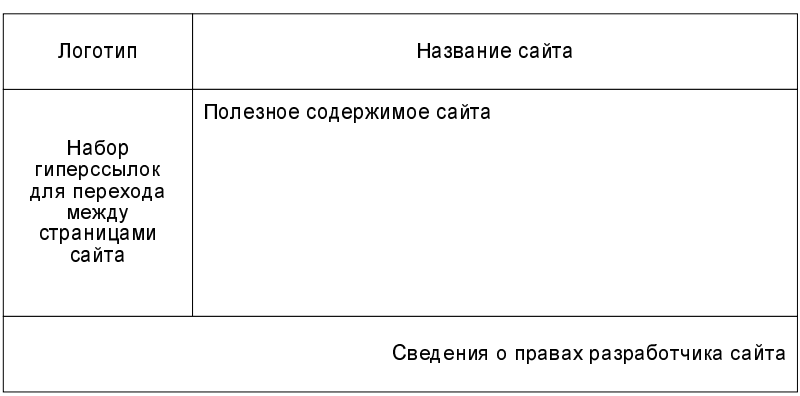


Рис. 7.5. Набор фреймов с заголовком, состоящим из двух частей

Впрочем, часто заголовок помещают в текст начальной страницы, отображаемой в качестве основного содержимого. В этом случае он не занимает место на экране, оставляя его для основного содержимого сайта. Данный подход также правилен. Однако следует помнить, что тогда посетитель сайта может забыть, где он, собственно, находится в данный момент, иначе говоря, "заблудится". Поэтому сведения о сайте необходимо все-таки дублировать на страницах основного содержимого. Компромиссным вариантом будет объединение заголовка сайта и набора гиперссылок в одной странице и, как следствие, в одном фрейме, как показано на рис. 7.6.

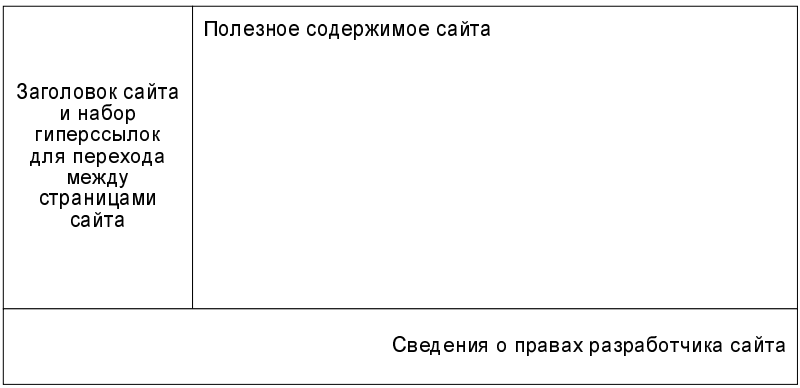


Рис. 7.6. Набор фреймов с заголовком, объединенным с набором гиперссылок

Что касается сведений об авторских правах, то нужда в том, чтобы они постоянно присутствовали на экране, не так велика. Пожалуй, их помещают в отдельный фрейм только те, кто озабочен своими правами на сайт и опубликованные на нем материалы. В этом случае лучше расположить их ненавязчиво — в нижнем горизонтальном фрейме, как показано на рис. 7.1.

В большинстве случаев сведения об авторских правах располагают либо в тексте начальной страницы, что отображается по умолчанию в основном фрейме, либо помещают в один фрейм с набором гиперссылок, в самом его низу. Также можно поместить их в заголовок сайта, если он расположен в отдельном фрейме.

Кроме того, часто в отдельных фреймах помещают рекламу в виде баннеров. Таким образом реклама отделяется от действительно полезной информации. Обычно фреймы с рекламой располагаются горизонтально сверху и (или) снизу окна Web-обозревателя и не очень велики по размерам — как раз на один или два баннера (рис. 7.7).

Особенно часто рекламу во фреймах помещают Web-серверы, бесплатно "раздающие" место под Web-сайты пользователей. В качестве платы за бесплатность они либо заставляют вас поместить на ваши страницы рекламные

баннеры (что не наш случай), либо "втискивают" ваши странички во фрейм, а в соседних фреймах набора "развешивают" свои баннеры. Иногда это бесит, но, в конце концов, надо же владельцам этих Web-серверов на что-то жить. А реклама — иной раз единственный источник дохода в Интернете...

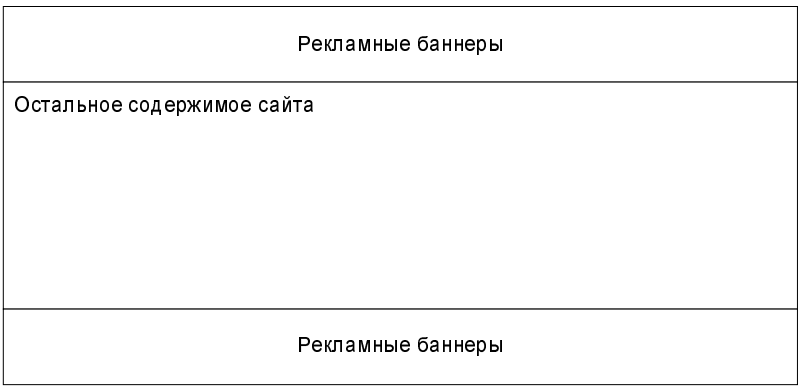


Рис. 7.7. Набор фреймов с двумя рекламными блоками

Здесь следует сказать еще о так называемых *нулевых* или *невидимых фреймах*. Так называются фреймы, чья ширина (для вертикальных фреймов) или высота (для горизонтальных) равна нулю. Фактически такие фреймы не видимы в окне Web-обозревателя. Используются они, как правило, для всяческих неблаговидных целей: размещения рекламных баннеров, которые оплачиваются за факт показа (такой баннер как бы показывается, но на самом деле не видим), скрытия какой-либо важной информации и пр. Часто такие "фокусы" можно увидеть на хакерских и жульнических сайтах.

Здесь перечислены далеко не все схемы фреймовых наборов, имеющих хождение в Сети. Кроме общеупотребительных схем, существуют различные специфические случаи, которых также довольно много. Описываться в этой книге они не будут, т. к. уж очень специфичны и применимы далеко не всегда.

Сложные наборы фреймов

А теперь нужно рассказать об одной особенности фреймов, которую вам обязательно надо знать.

Как вы уже знаете, фреймы являются составными частями набора фреймов. Тег, задающий набор фреймов, определяет размеры и форму фреймов (будут ли они располагаться горизонтально или вертикально), а тег, задающий фрейм, определяет параметры самого фрейма (в частности, интернет-адрес загружаемой в него Web-страницы). При этом теги, задающие фреймы, вкладываются в тег, задающий набор фреймов.

Но проблема в том, что фреймы, входящие в один набор, могут располагаться либо только по горизонтали, либо только по вертикали. Но, скажете вы, как же сделать наборы фреймов, приведенные на рис. 7.1—7.7? Ведь их же кто-то делает, а если не делает, так зачем их было здесь приводить?

Да, вы правы. Но для этого используется одна особенность наборов фреймов HTML — они могут вкладываться друг в друга, порождая *вложенные* наборы фреймов. Такая совокупность наборов фреймов обычно считается одним целым, одним *сложным* набором, состоящим из нескольких *простых*.

Взгляните на рис. 7.1. Давайте подумаем, как приведенный на нем набор фреймов можно реализовать на практике. Здесь мы имеем три фрейма, расположенных горизонтально и занимающих всю ширину набора, и один фрейм, занимающий часть среднего фрейма. Очевидно, что нам понадобятся два набора фреймов: один, *внешний*, делящий окно Web-обозревателя на три части по горизонтали, и второй, *внутренний*, делящий средний фрейм надвое по вертикали. Рис. 7.8 иллюстрирует вышесказанное.

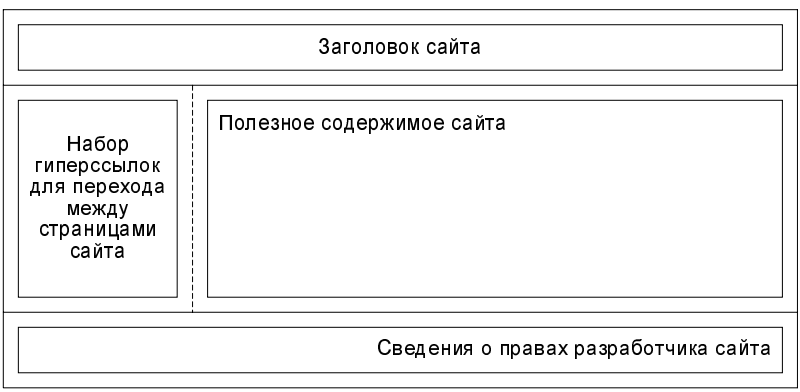


Рис. 7.8. Сложный набор фреймов, состоящий из двух простых. Внешний набор фреймов обозначен сплошными линиями, внутренний — штриховыми

Пользуясь вложенностью, вы можете создавать сколь угодно сложные наборы фреймов. Это очень просто, если понять принцип. Хотя, чтобы создать совсем уж замысловатый набор фреймов, вам придется поломать голову.

Работа с фреймами

Но хватит пустых разговоров! Давайте займемся делом и на конкретном примере рассмотрим принципы фреймового дизайна. Мы создадим новую версию нашего сайта, посвященного гипотетическому Web-дизайнеру Ивану Ивановичу Иванову, версию, основанную на фреймах. И, не мудрствуя лукаво, выберем для него схему, показанную на рис. 7.1.

Проверьте, создали ли вы новую папку Sample2. Также проверьте, не удалили ли вы старую версию сайта, находящуюся в папке Sample1. Ведь одна из важнейших заповедей Web-дизайнера: не разбрасываться старыми материалами — они могут пригодиться для нового сайта. Проверили? Все на месте? Тогда вперед!

Прежде всего, введем в Dreamweaver информацию о новом сайте. Вы можете просто взять сайт Sample site 1 и заменить его имя, корневую папку и имя главной страницы, а можете и создать новый сайт "с нуля". Как это делается, было рассказано в *главе 6*. На вкладке **Remote Info** диалогового окна **Site Definition** в раскрывающемся списке **Access** выберите пункт **None** (см. рис. 6.3) — мы с вами нигде не будем публиковать этот сайт. И имя нашего сайта будет Sample site 2.

После этого создайте новую Web-страницу. Сейчас мы рассмотрим процесс создания фреймов...

Создание фреймов

Да, мы рассмотрим, как создаются фреймы в Dreamweaver.

Но сначала сделайте следующее. Включите в подменю **Visual Aids** меню **View** пункт-выключатель **Frame Borders**. Это нужно, чтобы Dreamweaver показал нам *границы* наших будущих фреймов. Дело в том, что границы между фреймами могут быть невидимыми, что может создать нам немало проблем при написании текста страниц, отображаемых в этих фреймах. Если же вы выберете названный пункт, Dreamweaver будет показывать схематичные линии в тех местах, где проходят границы фреймов.

Внимание!

При открытии страницы, определяющей набор фреймов, вам придется каждый раз включать пункт-выключатель **Frame Borders** подменю **Visual Aids** меню **View**. К сожалению, Dreamweaver не запоминает этой установки.

Проще всего набор фреймов можно создать, воспользовавшись вкладкой **Frames** панели объектов. На рис. 7.9 показаны те ее кнопки, которые нам сейчас нужны. Голубым (на рисунке — светло-серым) цветом закрашен *текущий* фрейм, т. е. тот, в котором в настоящий момент стоит текстовый курсор. Также можно воспользоваться пунктами подменю **Frames** меню **Insert**. В табл. 7.1 приведены описания всех кнопок вкладки **Frames** панели объектов и соответствующих им пунктов подменю **Frames**.

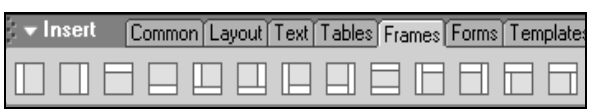


Рис. 7.9. Вкладка **Frames** панели объектов (показана частично)

Таблица 7.1. *Кнопки вкладки* ***Frames*** *панели объектов (в порядке слева направо)*

Название кнопки	Пункт подменю Frames меню Insert	Описание
Left Frame	**Left**	Вставляет фрейм слева от текущего (того, в котором находится текстовый курсор) в тот же набор
Right Frame	**Right**	Вставляет фрейм справа от текущего в тот же набор
Top Frame	**Top**	Вставляет фрейм выше текущего в тот же набор
Bottom Frame	**Bottom**	Вставляет фрейм ниже текущего в тот же набор
Bottom and Nested Left Frame	**Bottom Nested Left**	Вставляет фрейм ниже текущего в тот же набор, создает в текущем фрейме вложенный набор с двумя горизонтальными фреймами и делает текущим правый фрейм вложенного набора
Bottom and Nested Right Frame	**Bottom Nested Right**	Вставляет фрейм ниже текущего в тот же набор, создает в текущем фрейме вложенный набор с двумя горизонтальными фреймами и делает текущим левый фрейм вложенного набора
Left and Nested Bottom Frame	**Left Nested Bottom**	Вставляет фрейм левее текущего в тот же набор, создает в текущем фрейме вложенный набор с двумя вертикальными фреймами и делает текущим верхний фрейм вложенного набора
Right and Nested Bottom Frame	**Right Nested Bottom**	Вставляет фрейм правее текущего в тот же набор, создает в текущем фрейме вложенный набор с двумя вертикальными фреймами и делает текущим верхний фрейм вложенного набора
Top and Bottom Frames	**Top and Bottom**	Вставляет два фрейма сверху и снизу от текущего
Left and Nested Top Frame	**Left Nested Top**	Вставляет фрейм левее текущего в тот же набор, создает в текущем фрейме вложенный набор с двумя вертикальными фреймами и делает текущим нижний фрейм вложенного набора

Таблица 7.1 *(окончание)*

Название кнопки	Пункт подменю Frames меню Insert	Описание
Right and Nested Top Frame	**Right Nested Top**	Вставляет фрейм правее текущего в тот же набор, создает в текущем фрейме вложенный набор с двумя вертикальными фреймами и делает текущим нижний фрейм вложенного набора
Top and Nested Left Frame	**Top Nested Left**	Вставляет фрейм выше текущего в тот же набор, создает в текущем фрейме вложенный набор с двумя горизонтальными фреймами и делает текущим правый фрейм вложенного набора
Top and Nested Right Frame	**Top Nested Right**	Вставляет фрейм выше текущего в тот же набор, создает в текущем фрейме вложенный набор с двумя горизонтальными фреймами и делает текущим левый фрейм вложенного набора

Итак, давайте приступим к созданию нашего набора фреймов.

Прежде всего, создадим верхний фрейм, в котором у нас будет помещаться заголовок сайта. Нажмем кнопку **Top Frame**. В результате получатся два фрейма, расположенные горизонтально, друг над другом (рис. 7.10).

Теперь убедимся, что текстовый курсор стоит в нижнем фрейме, и нажмем кнопку **Bottom Frame**. В результате получим еще один фрейм, но расположенный на этот раз ниже текущего (рис. 7.11).

Теперь поместите курсор мыши на границу между верхним и средним фреймами (заголовок и основное содержимое, соответственно). Вы увидите, что курсор примет вид двунаправленной стрелки, что во всех Windows-программах означает только одно: вы можете перемещать что-либо вверх-вниз. В данном случае вы можете перемещать границу между фреймами, изменяя их размеры. То же самое вы можете делать с границей между средним и нижним фреймами (основное содержимое и сведения об авторских правах).

Измените размеры фреймов так, чтобы средний фрейм занимал максимум места на экране. (Помните, что мы говорили об основном содержимом сайта?) Верхний и нижний фреймы сделайте совсем узкими. А если впоследствии содержимое не будет в них помещаться, мы их увеличим.

Осталось создать еще один фрейм — левый, где будет помещаться набор гиперссылок. Его мы создадим немного по-другому, для чего воспользуемся подменю **Frameset** меню **Modify**.

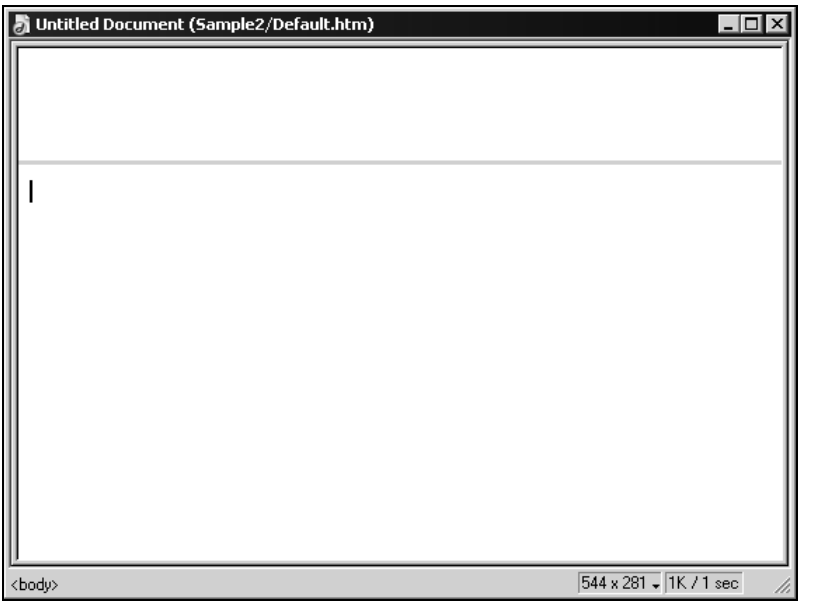


Рис. 7.10. Два горизонтальных фрейма

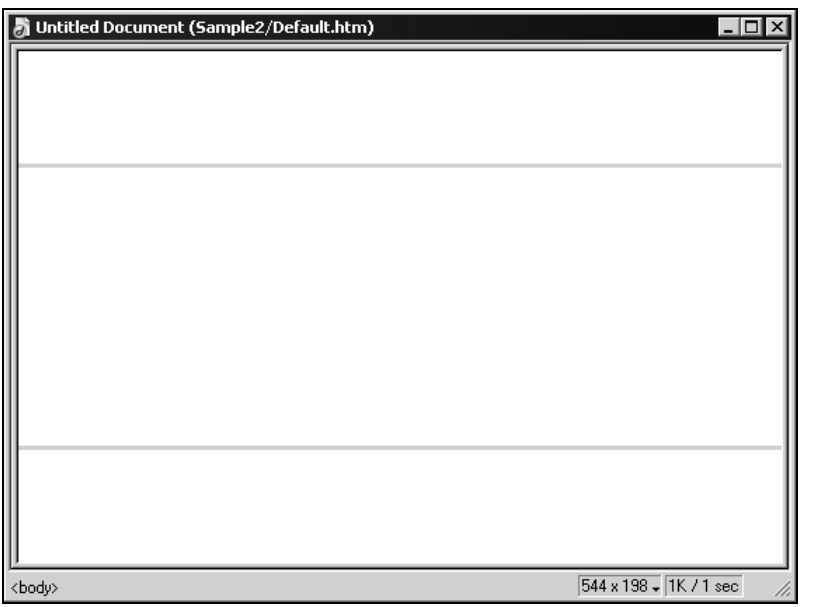


Рис. 7.11. Три горизонтальных фрейма

Там находятся следующие пункты:

- **Split Frame Left** — создание фрейма слева от текущего;
- **Split Frame Right** — создание фрейма справа от текущего;
- **Split Frame Up** — создание фрейма сверху от текущего;
- **Split Frame Down** — создание фрейма снизу от текущего.

Итак, сделайте текущим средний фрейм и выберите в подменю **Frameset** меню **Modify** пункт... правильно, **Split Frame Left**. Вы получите последний фрейм, который как раз и был нам нужен. Уменьшите его ширину — это делается так же, как и в случае с горизонтальными фреймами. У вас должно получиться нечто, похожее на то, что изображено на рис. 7.12

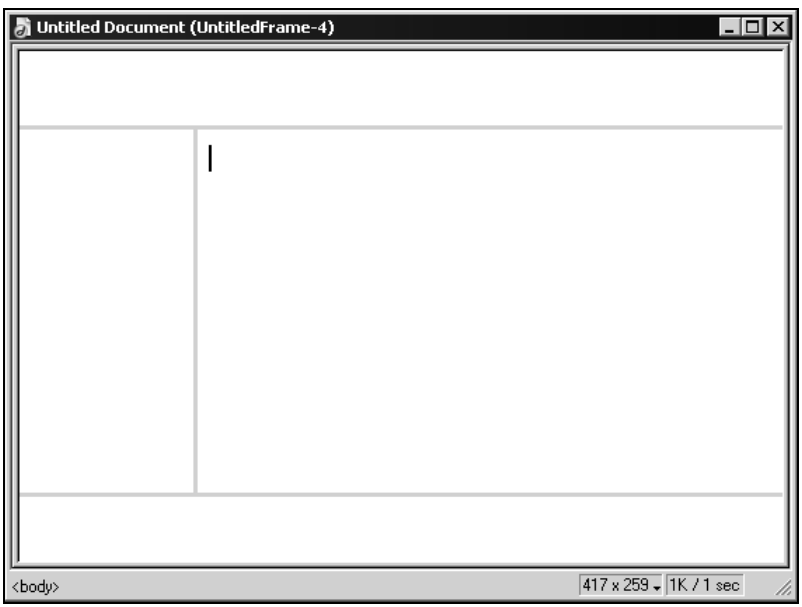


Рис. 7.12. Готовый набор фреймов

Теперь дайте готовой странице с набором фреймов название "Иван И. Иванов" и сохраните ее, выбрав пункт **Save All** меню **File** — это позволит сохранить также содержимое всех фреймов. На экране появится стандартное диалоговое окно сохранения файла Windows, предлагающее вам сохранить сам набор фреймов под именем UntitledFrameset-1.htm. Сохраните его в корневой папке нового сайта под именем default.htm.

Теперь Dreamweaver будет поочередно предлагать вам сохранить содержимое каждого фрейма набора (не забывайте, что содержимое фрейма задает отдельная Web-страница). При этом фрейм, содержимое которого сохраняется в данный момент, выделяется толстой штриховой линией (рис. 7.13),

благодаря чему вы будете знать, что сохраняет Dreamweaver, и какое имя дать той или иной странице.

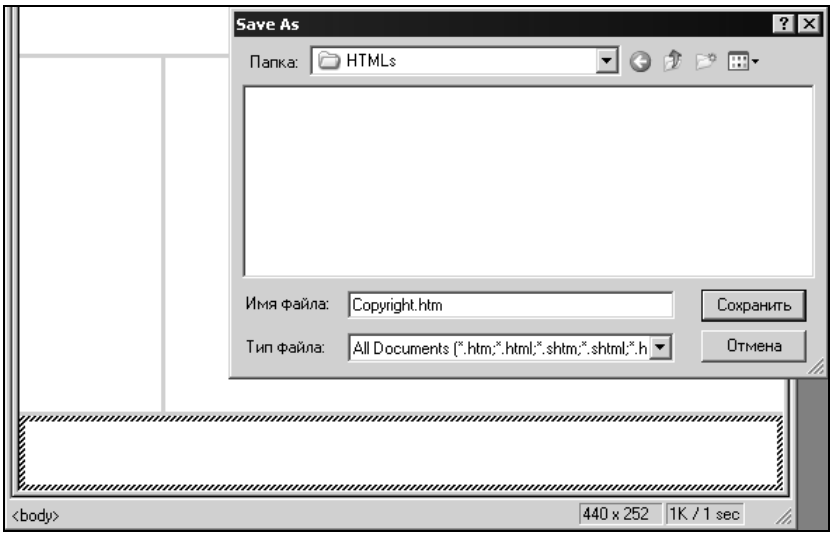


Рис. 7.13. Толстая штриховая линия выделяет фрейм, чье содержимое сохраняется в данный момент

Все остальные страницы сайта сохраните в папке HTMLs. Странице со сведениями об авторских правах дайте имя Copyright.htm. Страницу, по умолчанию отображаемую во фрейме с основным содержимым, назовите Main.htm, страницу со списком гиперссылок — Nav.htm, а страницу с заголовком сайта — Header.htm.

Внимание!

Если вы преобразуете уже готовую Web-страницу в набор фреймов, Dreamweaver поместит содержимое этой страницы в текущий фрейм. Если это главная страница вашего сайта (default.htm), то вам придется перед созданием набора фреймов переименовать ее, иначе не удастся сохранить страницу с набором фреймов под именем default.htm.

Существует еще один способ создания фреймов. Сейчас мы с вами его рассмотрим.

Взгляните на рис. 7.12. Обратите внимание на толстую серую рамку, окружающую весь наш набор фреймов. Вы можете создавать новые фреймы набора, не пользуясь ни панелью объектов, ни меню, а просто захватив эту рамку мышью и перетащив на то место, где должна находиться граница вновь создаваемых фреймов. В частности, вы можете таким образом разбивать один фрейм на два.

Теперь можно сделать небольшой перерыв, после чего мы узнаем еще кое-что о фреймах и особенностях их поддержки Dreamweaver.

Примечание

Наборы фреймов описываются с помощью парного тега <FRAMESET>, который должен содержать один из атрибутов: ROWS или COLS. Названные атрибуты задают список значений высоты (для тега <ROWS>) или ширины (для <COLS>) всех фреймов данного набора. Сами фреймы задаются одинарным тегом <FRAME>. И <FRAMESET>, и <FRAME> поддерживают множество атрибутов, задающих различные свойства набора фреймов и самих фреймов; эти атрибуты мы рассмотрим позднее.

В частности, код, определяющий набор из двух горизонтально расположенных фреймов, верхний из которых занимает 20% пространства окна, а другой — 80%, выглядит так:

```
<FRAMESET ROWS="20%,80%" ...>
  <FRAME ...>
  <FRAME ...>
</FRAMESET>
```

Соответственно, два вложенных набора фреймов будут выглядеть так:

```
<FRAMESET ROWS="20%,80%" ...>
  <FRAME ...>
  <FRAMESET COLS="100,*" ...>
    <FRAME ...>
    <FRAME ...>
  </FRAMESET>
</FRAMESET>
```

Внимание!

Web-страница, описывающая набор фреймов, не должна включать в себя тег <BODY> и его содержимое. Она должна содержать только описание набора фреймов.

Работа с фреймами и наборами фреймов

А теперь выясним, что Dreamweaver позволит нам сделать с наборами фреймов и отдельными фреймами.

Свойства наборов фреймов

Для того чтобы задать свойства наборов фреймов и отдельных фреймов, нужно сначала выбрать один из них. Лучше всего это проделывать в панели **Frames**. Изначально она отсутствует в доке, поэтому выберите пункт **Frames**

в подменю **Others** меню **Window** или нажмите комбинацию клавиш <Shift>+<F2>. Сама панель **Frames** показана на рис. 7.14.

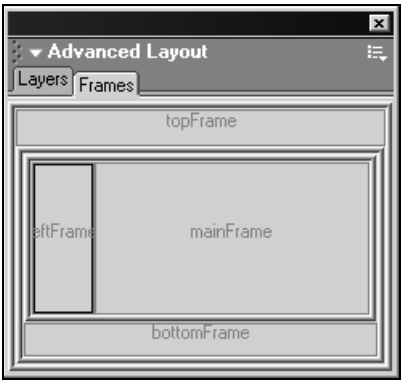


Рис. 7.14. Панель **Frames**

Посмотрите внимательно на эту панель. В ней показаны все фреймы и вся структура наборов фреймов. Фреймы изображены в виде серых прямоугольников, внутри которых написаны их имена (об именах фреймов мы поговорим ниже). Надпись "(no name)" говорит о том, что фрейм не имеет имени. Наборы фреймов представлены в виде толстых темно-серых границ, окружающих пары фреймов. На самом деле, это весьма удобно — все сразу видно. Единственная сложность: поначалу трудно попасть мышью в кажущуюся такой толстой границу фреймового набора.

Да, вы можете щелкать мышью по границам наборов фреймов, чтобы выбрать тот или иной набор. Также вы можете щелкать мышью и по самим фреймам, выбирая их. При этом редактор свойств отобразит параметры выбранного фрейма или набора фреймов.

А теперь самое время рассказать, что Dreamweaver позволяет нам сделать с наборами фреймов. Выберите в панели **Frames** какой-либо набор фреймов, например, самый внешний. Вид редактора свойств в этом случае показан на рис. 7.15.

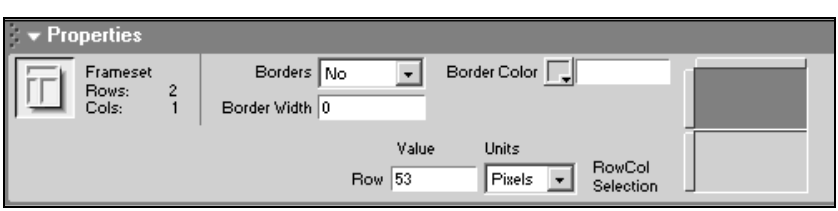


Рис. 7.15. Вид редактора свойств при выделенном наборе фреймов

Элементы управления, расположенные в верхней части редактора свойств, позволяют установить параметры самого набора фреймов. Сейчас мы их перечислим.

Раскрывающийся список **Borders** позволяет задать наличие или отсутствие границ между фреймами набора. Здесь доступны три пункта: **Yes** — границы есть, **No** — границ нет и **Default** — значение по умолчанию, зависящее от Web-обозревателя (как правило, границы есть).

Если вы выставили параметр **Borders** в **Yes** или **Default**, проверьте установки параметра **Border Width**. Поле ввода **Border Width** позволяет задать толщину границы в пикселах; если она равна нулю, то граница невидима. И наоборот, если вы выставили **Borders** в **No**, лучше установите **Border Width** в ноль, иначе могут быть проблемы с отображением в некоторых программах Web-обозревателей. Значение по умолчанию зависит от Web-обозревателя и операционной системы.

Внимание!

Границы малой толщины (меньше двух пикселов) могут отображаться неправильно или не отображаться вообще. Подробнее об особенностях границ фреймов см. в электронном руководстве по HTML (атрибут `BORDER` тега `<FRAMESET>`).

Селектор цвета **Border Color** позволяет задать цвет границы. Разумеется, чтобы этот параметр имел действие, граница между фреймами набора должна быть видимой. Значение по умолчанию зависит от Web-обозревателя (как правило, серая граница с трехмерным эффектом).

Попробуйте поэкспериментировать с этими параметрами и посмотрите, что получится. Для чистоты эксперимента просмотрите полученную страницу в разных программах Web-обозревателей.

Вообще, в настоящее время фреймы с видимыми границами считаются дурным тоном. Сейчас в моде фреймы с невидимыми или очень тонкими границами. Мы тоже сделаем такие.

Теперь обратимся к нижней части редактора свойств. Находящиеся там элементы управления позволят вам задать размеры отдельных фреймов выбранного набора.

В правой части редактора свойств вы видите схематичное изображение набора и содержащихся в нем фреймов. В общем, это похоже на уже знакомую вам панель **Frames**. Вы можете выбрать любой фрейм простым щелчком мыши; при этом выбранный фрейм будет закрашен темно-серым цветом. Однако запомните, что это все-таки не панель **Frames**: вы не можете сделать выбранный фрейм текущим и задать все его свойства.

В поле ввода **Value** вводится значение высоты или ширины фрейма. Это значение может быть задано в пикселах или процентах. Также вы можете

ввести туда знак звездочки ("*"), обозначающий все остальное доступное пространство, оставшееся от других фреймов. Dreamweaver корректно обработает его.

Раскрывающийся список **Units** задает единицу измерения высоты или ширины фрейма. Доступны три пункта: **Pixels** — пикселы, **Percent** — проценты и **Relative** — все остальное пространство (аналогично вводу звездочки в поле **Value**).

Надо сказать, что уже при создании набора фреймов Dreamweaver достаточно "интеллектуально" подставляет значения их параметров. Поэтому вполне возможно, что вам вовсе не придется их изменять. Ну, так, может быть, чуть-чуть подправить...

Загрузите созданную ранее страницу default.htm (см. рис. 7.12), если она еще не загружена. Выберите поочередно все наборы фреймов, проверьте значения их параметров и, если нужно, измените их. Значение **Borders** должно быть установлено в **No**, толщина границ **Border Width** — в ноль, а поле цвета границ **Border Color** должно оставаться пустым. Установите высоты верхнего (заголовок) и нижнего (сведения об авторских правах) фреймов в 30 пикселов, а ширину левого фрейма — в 100 пикселов. Если потом эти фреймы окажутся малы, мы их увеличим.

Свойства фреймов

С наборами фреймов мы разобрались. А что же с самими фреймами?

Выберите в панели **Frames** любой фрейм, скажем, самый верхний. Редактор свойств примет вид, показанный на рис. 7.16.

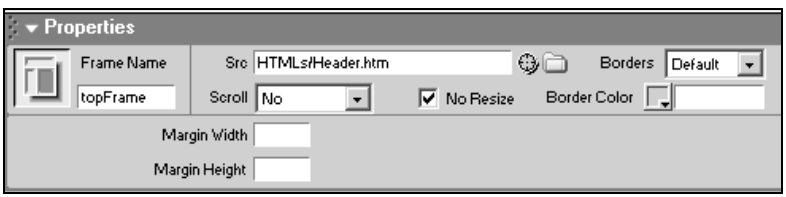


Рис. 7.16. Вид редактора свойств при выделенном фрейме

В поле ввода **Frame Name** вводится *имя фрейма*. Dreamweaver по умолчанию подставляет туда автоматически сформированное имя, и часто оно оказывается весьма удачным. Но иногда его следует изменить. Например, в нашем случае лучше всего будет дать ему имя `Header`, обозначив таким образом отображаемое в нем содержимое.

Зачем фрейму необходимо имя, мы выясним позднее. Сейчас скажем только, что мы можем сослаться на нужный фрейм по его имени. Вообще-то, если на фрейм не надо ссылаться, можно и не давать ему имени.

В поле ввода **Src** вводится имя файла Web-страницы, отображаемой во фрейме. Dreamweaver сам заполняет это поле при создании и сохранении страницы набора фреймов. Данное поле можно и не заполнять; в таком случае во фрейме ничего не будет отображаться.

Раскрывающийся список **Scroll** задает, будет ли фрейм содержать полосы прокрутки. Доступны четыре пункта:

- **Yes** — полосы прокрутки есть всегда;
- **No** — полос прокрутки нет даже тогда, когда содержимое фрейма не помещается в нем;
- **Auto** — полосы прокрутки появляются только тогда, когда в них появляется необходимость (содержимое фрейма не помещается в нем);
- **Default** — значение по умолчанию, зависящее от Web-обозревателя (как правило, аналогично **Auto**).

Флажок **No Resize** позволяет запретить пользователю изменять размеры фреймов перетаскиванием их границ. Обычно пользователь может перетаскивать границы фреймов, изменяя их размеры. Для фреймов, отображающих специальную информацию, таких как набор гиперссылок или заголовок сайта, лучше запретить это, иначе подобные фреймы будут выглядеть неаккуратно. Впрочем, Dreamweaver об этом за нас уже позаботился — включил данный флажок.

Раскрывающийся список **Borders** уже вам знаком. Однако его действие распространяется только на данный фрейм. Таким образом, вы можете задать наличие или отсутствие границ и у отдельного фрейма. Пункт **Default** этого меню позволит вам вернуться к параметрам набора фреймов.

В селекторе цвета **Border Color** вы можете установить цвет границы фрейма.

Поля ввода **Margin Width** и **Margin Height** позволяют задать, соответственно, горизонтальное и вертикальное расстояния между границами фрейма и его содержимым. Значения по умолчанию — 14 пикселов.

Выберите поочередно все четыре фрейма и установите их параметры, как указано в табл. 7.2. Остальные параметры оставьте в положении по умолчанию.

Таблица 7.2. *Параметры фреймов нашего набора*

Фрейм	Frame Name	Scroll	No Resize
Верхний (заголовок сайта)	`Header`	**No**	*
Левый (набор гиперссылок)	`Nav`	**No**	*
Основной (основное содержимое сайта)	`Main`	**Auto**	*
Нижний (сведения об авторских правах)	`Copyright`	**No**	*

Установив параметры наборов фреймов и самих фреймов, сохраните страницу, выбрав пункт **Save Frameset** в меню **File** или нажав комбинацию клавиш <Ctrl>+<S>.

Примечание

Имя фрейма задается атрибутом NAME тега <FRAME>. Адрес Web-страницы, которая будет в нем отображаться, задается атрибутом SRC этого же тега. Остальные атрибуты перечислены в электронном руководстве по HTML, поставляемом в составе Dreamweaver.

Замещение и работа с ним

Мы уже говорили, что фреймы, хоть и стандартизированы совсем недавно, используются в Web-дизайне и поддерживаются Web-обозревателями уже довольно давно. Но все когда-нибудь делается в первый раз... И старейшие программы Web-обозревателей — Microsoft Internet Explorer и Netscape Navigator — поддерживают фреймы далеко не с первой версии. Кроме того, существует много программ, вообще ничего не знающих о фреймах. Они их просто не поддерживают, а все потому, что эти программы слишком старые.

Что случится, если в таком Web-обозревателе открыть Web-страницу, представляющую собой набор фреймов? Скорее всего, он отобразит пустую страницу без всяких следов содержимого. Как вы помните, согласно стандартам HTML, Web-обозреватель обязан игнорировать неизвестные ему теги. Он их и проигнорирует. А поскольку опять по тем же стандартам страница с набором фреймов не должна иметь содержимого (тега <BODY>), то в результате мы получим пустое окно.

Каков же выход из этого положения?

Для таких программ можно сформировать так называемое *замещение* — особое содержимое, которое будет игнорироваться современными программами, "знакомыми" с фреймами, но будет выводиться старыми. Это замещение помещается в той же странице, где определяется набор фреймов. В результате программа, не поддерживающая фреймы, игнорирует теги, определяющие набор фреймов, и выводит на экран текст замещения. А программы, поддерживающие фреймы, игнорируют замещение, т. к. "знают" о его существовании.

Как же можно сформировать замещение в Dreamweaver? Как всегда, очень просто. Включите в подменю **Frameset** меню **Modify** пункт-выключатель **Edit NoFrames Content**. Окно документа очистится; в верхней его части появится серая полоса с надписью **NoFrames Content**. Введите в окно документа текст замещения.

Как должен выглядеть текст замещения? Исходите из того, что пользователи устаревших программ, просматривая ваш сайт, не должны испытывать

никаких неудобств. Возьмите за основу главную страницу нашего предыдущего сайта default.htm. Поместите здесь набор гиперссылок и сведения об авторских правах.

Вы можете делать с текстом замещения все, что угодно. Вы можете помещать сюда внедренные элементы и таблицы, можете форматировать текст любыми доступными в HTML способами, можете задавать параметры текста замещения, как будто это обычная Web-страница, выбрав пункт **Page Properties** меню **Modify**. В частности, обязательно измените параметры текста замещения так, чтобы они совпадали с параметрами остальных страниц нового сайта (черный фон, белый текст и т. д.). (Помните: единообразие оформления прежде всего!) Вы только не можете создавать здесь фреймы. Понятно, почему.

Посмотрите, например, что сделал автор (рис. 7.17). Он скопировал сюда текст страницы default.htm нашего старого сайта.

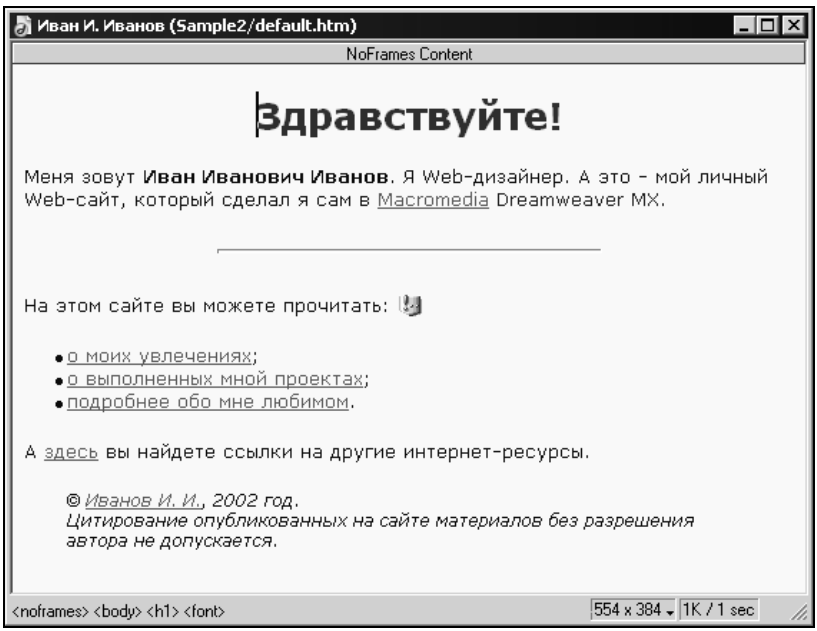


Рис. 7.17. Текст замещения

Чтобы вернуться от текста замещения к набору фреймов, отключите в подменю **Frameset** меню **Modify** пункт-выключатель **Edit NoFrames Content**.

В каких случаях стоит задавать замещение? Только тогда, когда вы точно уверены, что среди посетителей вашего сайта будут обладатели старых программ Web-обозревателей. Причем, таких посетителей должно быть достаточно много, чтобы ради них выполнять дополнительную работу. Но учтите,

что во всем мире таких ретроградов очень и очень мало; абсолютное большинство интернетчиков пользуются программами, поддерживающими фреймы.

Мы даже не знаем, когда у вас может возникнуть нужда в замещении. Подавляющее большинство сайтов в Сети, использующих фреймы, его не имеют. А некоторые, хоть и вспоминают, что не все Web-обозреватели поддерживают их любимые фреймы, но ограничиваются одной-единственной фразой типа "Ваша программа не поддерживает фреймы, обновите ее".

Примечание

Замещение формируется с помощью парного тега <NOFRAMES>. Внутри этого тега располагается тело Web-страницы (с тегом <BODY>), содержащей текст замещения.

Создание содержимого фреймов

Создав набор фреймов и установив все параметры, можно приступать к наполнению фреймов содержимым.

Давайте используем для нашего сайта стильный дизайн, который сейчас в большой моде. Это белый или светло-желтый текст на черном или темно-сером фоне. Гиперссылки при этом выделяются интенсивно-желтым или слабо-красным цветом. Это выглядит довольно эффектно на фоне классических цветовых решений: темный текст на светлом фоне. Да, человечество за всю свою историю не придумало ничего лучше черных чернил и белой бумаги, но до чего же это красиво — белое на черном!

И не забывайте, что содержимое фрейма — на самом деле обычная Web-страница, сохраненная в отдельном файле. Поэтому работа с содержимым фрейма осуществляется таким же образом, как с любой страницей, т. е. вам необходимо задать ее параметры (цвета фона и текста и т. д.), набрать текст, отформатировать его, разместить нетекстовые элементы (горизонтальные линии, таблицы и т. п.) и графические изображения. Ну а это вам уже знакомо.

Если вам неудобно работать со страницей в маленьком фрейме, вы всегда можете открыть эту страницу в отдельном окне документа Dreamweaver.

Заполнение фреймов

Сначала займемся верхним фреймом, содержащим заголовок сайта. Мы поместим в него имя нашего героя, выровненное по центру фрейма. Для этого используем таблицу из одной ячейки, занимающую все свободное пространство фрейма. Но сначала уберем промежутки между границей фрейма и его содержимым, устанавливаемые по умолчанию. Для чего выделите

верхний фрейм в панели **Frames** и установите параметры **Margin Width** и **Margin Height** равными нулю. Это позволит нам отвести нашей таблице действительно все свободное пространство.

Теперь задайте цвета текста и фона страницы (гиперссылок здесь не будет). Поместите во фрейм текстовый курсор и вызовите диалоговое окно **Page Properties** (выберите пункт **Page Properties** в меню **Modify** или контекстном меню или нажмите комбинацию клавиш <Ctrl>+<J>). Задайте черный цвет фона (селектор цвета **Background**) и желтый цвет текста (селектор цвета **Text**). Также задайте нулевое значение для параметров **Left Margin**, **Top Margin**, **Margin Width** и **Margin Height** (расстояния от границы окна до границы содержимого страницы) и нажмите кнопку **OK**. Название страницы можно не задавать — все равно в любом случае Web-обозреватель выведет в заголовке своего окна название страницы набора фреймов.

Поместите во фрейм таблицу из одной-единственной ячейки, занимающей всю его ширину и высоту. Задайте для содержимого ее ячейки выравнивание по центру, по горизонтали и вертикали. Установите флажок **No Wrap** — это запретит Web-обозревателю перенос текста в ячейке таблицы по строкам.

Введите в ячейку текст "Иван Иванович Иванов". Установите подходящий размер шрифта. Поэкспериментируйте с разными шрифтами. У вас получилось следующее (рис. 7.18).

Рис. 7.18. Готовый заголовок сайта

Теперь приступим к сведениям об авторских правах. Поставьте в нижний фрейм текстовый курсор и вызовите диалоговое окно **Page Properties**. Задайте опять черный цвет фона, желтый цвет текста, ярко-синий цвет гиперссылок (селектор цветов **Links**; цвет `#00FFFF`), тускло-синий цвет посещенных гиперссылок (селектор цветов **Visited Links**; `#00CCFF`) и светло-зеленый цвет активной гиперссылки (селектор цветов **Active Links**; `#99FF00`). Опять задайте нулевое значение для параметров **Left Margin**, **Top Margin**, **Margin Width** и **Margin Height**. После этого нажмите кнопку **OK**.

И наберите соответствующий текст. Но лучше сделайте так. Найдите страницу default.htm старого сайта и откройте ее в Web-обозревателе. Выделите текст сведений об авторских правах, расположенный внизу этой страницы, скопируйте его в буфер обмена Windows и вставьте в нижний фрейм.

Значения по умолчанию параметров **Margin Width** и **Margin Height** фрейма слишком велики (14 пикселов). Выберите нижний фрейм в панели **Frames** и установите значения равными 2 пикселам. Так будет лучше.

После этого вам останется только выделить его курсивом или отформатировать как-то особо, чтобы отделить от обычного текста. (Впрочем, он и так у нас отделен от всего остального.) И, возможно, изменить размер фрейма, чтобы содержащийся в нем текст отобразился полностью. У вас получится нечто, похожее на рис. 7.19.

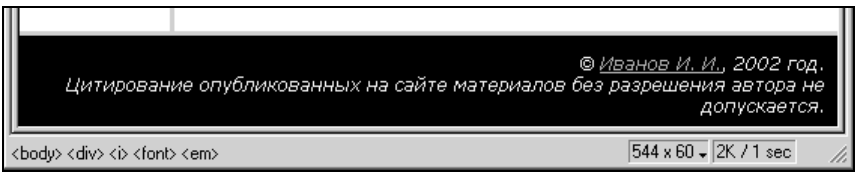


Рис. 7.19. Готовые сведения об авторских правах

Сохраните набор фреймов и все содержимое фреймов, выбрав пункт **Save All Frames** меню **File**. Вообще-то "сохраняться" нужно как можно чаще — запомните это.

И напоследок займемся основным содержимым. По умолчанию в этом фрейме будет отображаться страничка с приветствием и текстом, описывающим сайт. Мы возьмем этот текст опять же со страницы default.htm

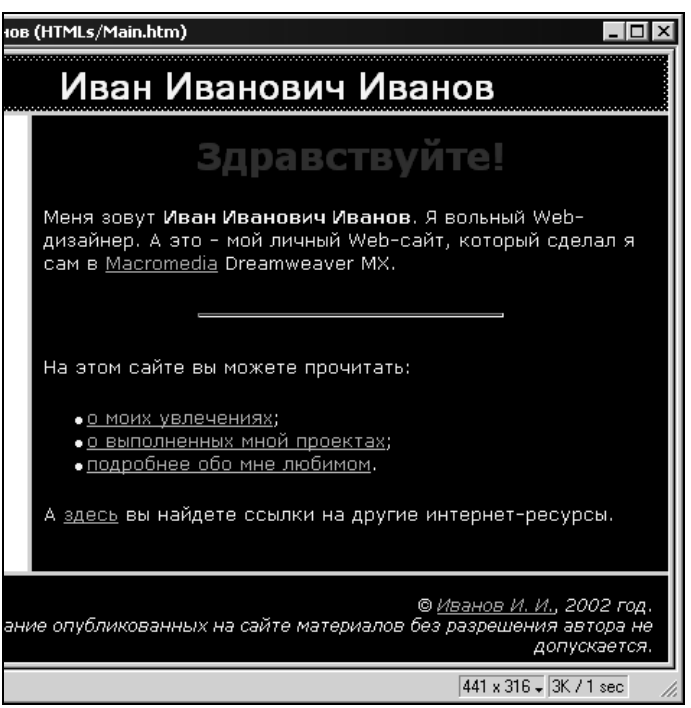


Рис. 7.20. Готовое основное содержимое сайта, отображаемое по умолчанию

старого сайта. Откройте ее в Web-обозревателе, если еще не открыли или уже закрыли.

В свойствах страницы основного содержимого укажите те же параметры, что и для страницы со сведениями об авторских правах. Значения параметров **Margin Width** и **Margin Height** для фрейма не меняйте. Далее переключитесь в окно Web-обозревателя, где открыта страница default.htm старого сайта, выделите весь текст вплоть до сведений об авторских правах, скопируйте его в буфер обмена Windows и вставьте во фрейм основного содержимого сайта. Немного подредактируйте текст, в частности уберите цветовое выделение слов "Macromedia Dreamweaver MX", преобразуйте библиотечный элемент в обычный текст и удалите наконец давно не нужный комментарий.

В результате у вас должно получиться то, что показано на рис. 7.20.

А как же набор гиперссылок? Потерпите, мы создадим и его. И для этого используем замечательные способности Dreamweaver по созданию полосы навигации.

Но сначала создадим остальные страницы нашего сайта.

Создание остальных страниц сайта

Мы не будем подробно рассказывать, как это делается, всецело полагаясь на вашу фантазию. Можем только посоветовать не слишком усердствовать с дизайном страниц и обязательно соблюдать единство их оформления.

Когда мы делали наш предыдущий сайт, мы натворили столько пестрых, разнообразных по оформлению страниц, что у посетителей сайта, который мы уже опубликовали в Интернете, сейчас рябит в глазах. Хотя эту пестроту можно объяснить: мы тогда учились делать различные вещи и экспериментировали на страницах нашего сайта. Но сейчас-то нам не нужно учиться элементарным вещам. Так что давайте больше не будем гнаться за пестротой.

Возьмите за образец страничку основного содержимого сайта Main.htm и делайте остальные страницы, держа ее перед глазами. Вы даже можете просто открыть страницу Main.htm в отдельном окне Dreamweaver, удалить весь текст и ввести новый. Только не сохраняйте страницу! А сделайте так: выберите пункт **Save As** в меню **File** или нажмите комбинацию клавиш <Ctrl>+<Shift>+<S>. На экране появится диалоговое окно сохранения файла Windows; введите новое имя файла и нажмите кнопку **OK**. Все — теперь вы сохранили новую страницу под новым именем.

После этого вам останется только "позаимствовать" текст уже созданных вами страниц старого сайта. Откройте необходимую страницу в окне Web-обозревателя (или Dreamweaver), выделите нужный текст, скопируйте его

в буфер обмена и вставьте в новую страницу. Подредактируйте его, если нужно. И сохраните готовую страницу.

Имена файлов новых страниц сделайте такими же, как у файлов страниц старого сайта. А именно:

- странице сведений об авторе дайте имя About.htm;
- странице ссылок — Links.htm;
- странице со списком увлечений — Passions.htm;
- странице со списком проектов — Projects.htm.

Останется только открыть страницу Main.htm и исправить гиперссылки.

Ну что, закончили? Теперь откройте страницу default.htm нового сайта в окне Web-обозревателя и щелкните по любой гиперссылке. Что произойдет? Нечто ужасное и совершенно не то, что нам нужно. Страница, на которую ссылается гиперссылка, загрузится и займет все окно вместо того, чтобы "втиснуться" в отведенный ей фрейм. Что делать?

Цель гиперссылки

Для того чтобы гиперссылки открывали соответствующую Web-страницу в нужном нам фрейме, надо правильно выставить некоторые их свойства.

Откройте страницу default.htm. Установите текстовый курсор на любой гиперссылке, находящейся во фрейме с основным содержимым. Пусть это будет гиперссылка, ведущая на страницу сведений об авторе About.htm. И обратите внимание на редактор свойств. А именно на раскрывающийся список **Target**, позволяющий задать цель гиперссылки. В развернутом виде он показан на рис. 7.21.

Рис. 7.21. Раскрывающийся список **Target**, содержащий имена фреймов

Вы уже знаете, что этот список позволяет задать, где будет отображаться страница, на которую указывает гиперссылка. В частности, пункт **_blank** этого списка позволит загрузить ее в новое окно Web-обозревателя, а пункт **_self** — в то же самое окно (по умолчанию). Но что мы видим на рис. 7.21? Там перечислены все созданные нами фреймы! А что если выбрать пункт **Main** (фрейм основного содержимого сайта)? Давайте попробуем: выберем и

снова загрузим страницу в окне Web-обозревателя. Ура, гиперссылка работает как надо!

Теперь поочередно выберите каждую гиперссылку в странице и установите параметр **Target** в **Main**, за исключением гиперссылки, ведущей на сайт фирмы Macromedia, и почтовой гиперссылки: для первой установите значение **_blank** (пусть их сайт открывается в новом окне), а для второй оставьте все, как было — все равно для нее параметр **Target** роли не играет.

Теперь давайте подытожим все, что мы узнали о значениях параметра **Target**:

- **_blank** загружает страницу в новое окно Web-обозревателя;
- **_parent** загружает страницу во фрейм набора верхнего уровня, в котором находится текущий фрейм;
- **_top** загружает страницу в текущее окно обозревателя, т. е. эта страница после загрузки заменит собой весь набор фреймов;
- **_self** загружает страницу в текущий фрейм (в котором находится гиперссылка);
- <***имя фрейма***> загружает страницу в заданный фрейм.

Здесь нужны некоторые пояснения, В самом деле, чем отличаются значения **_parent** и **_top** параметра **Target**? А вот чем.

Предположим, мы создали сложный набор фреймов, состоящий из вложенных друг в друга простых наборов, которые назовем *внешним* и *внутренним*. Теперь давайте попытаемся загрузить какую-либо Web-страницу в один из фреймов внутреннего набора. И загружать мы ее будем с разными значениями параметра **Target**. Итак...

- Если мы загрузим страницу со значением **_parent** параметра **Target**, то она будет помещена в тот фрейм внешнего набора, в котором находится внутренний набор. Фактически она заменит собой весь внутренний набор фреймов.
- Если же мы используем значение **_top** параметра **Target**, то страница заменит собой весь наш сложный набор фреймов, т. е. займет окно Web-обозревателя целиком.

Остальные значения параметра **Target** в комментариях не нуждаются.

Теперь, зная, как загрузить Web-страницу в нужный фрейм, можно заняться полосой навигации.

Полоса навигации

Полоса навигации — это обычный набор гиперссылок, организованный в виде вертикальной или горизонтальной полосы и расположенной вдоль

края окна Web-обозревателя. Как правило, полоса навигации располагается в специально выделенном фрейме. И очень часто формируется с использованием таблицы: либо в виде набора текстовых ссылок, либо в виде составного изображения. Во втором случае очень часто гиперссылки полосы навигации делаются "живыми", т. е. реагирующими на наведение курсора мыши и щелчок по ним. Это делается, как вы уже поняли, с помощью активных изображений.

Сделать полосу навигации с текстовыми гиперссылками проще простого. Достаточно вставить в левый фрейм таблицу из пяти строк и одного столбца, занимающую всю его ширину и высоту, вписать в ячейки нужные слова и превратить их в гиперссылки. Конечно, вы можете помудрить над таблицей: "приделать" ей границу, как было описано в *главе 5*, сделать графический фон и т. п. Но, согласитесь, это делается намного проще, чем графическая полоса навигации. По крайней мере не нужно заготавливать уйму графических изображений для каждой гиперссылки.

Вы, наверное, уже задали себе вопрос: зачем нужна пятая ссылка? Для страницы Main.htm. В старом сайте мы ни на одной странице не предусмотрели ссылки для возврата на первую страницу. И зря.

А что же графическая полоса навигации? Она делается с помощью той же таблицы. В ячейках таблицы размещаются графические изображения, выполняющие роль гиперссылок (просматривается аналогия с составным изображением). Вы можете использовать различные визуальные эффекты, например установить большие промежутки между изображениями, манипулируя параметрами **Cell Padding** и **Cell Spacing** каждой ячейки таблицы, или разместить их вплотную друг к другу. Опять же, вы можете задать для таблицы отдельный цвет фона или графический фон. В общем, все в ваших руках.

Вы можете использовать набор обычных статичных изображений для гиперссылок. Но такие статичные ссылки давно вышли из моды; вспомните, что важнейшим признаком современной Всемирной паутины являются динамичность и интерактивность. Чтобы "оживить" ваши гиперссылки, вы можете использовать надписи или кнопки Flash (прекрасная идея!) или активные изображения. Второй подход более популярен, вероятно, потому, что не требует модуля проигрывателя Flash. Так или иначе Dreamweaver позволит вам сделать и то, и другое исключительно просто.

Более того, Dreamweaver имеет встроенные средства создания полосы навигации из набора активных изображений. Вам нужно будет только заготовить соответствующее количество графических изображений, представляющих каждую гиперссылку. Это самая трудоемкая часть работы — все остальное возьмет на себя Dreamweaver.

Давайте же сделаем для нашего сайта такую полосу навигации.

Сначала заготовим набор графических изображений. На каждую гиперссылку нам понадобятся четыре изображения:

- отображаемое в обычном состоянии;
- отображаемое, когда пользователь помещает над ним курсор мыши;
- отображаемое в "нажатом" состоянии (когда страница, на которую указывает эта гиперссылка, загружена);
- отображаемое в "нажатом" состоянии, когда пользователь помещает над ним курсор мыши.

Создадим набор небольших графических изображений в любом графическом редакторе, поддерживающем формат GIF. Он будет включать:

- белую надпись на черном фоне, отображаемую в обычном состоянии;
- красную надпись на черном фоне, отображаемую, когда пользователь помещает над ним курсор мыши;
- черную надпись на белом фоне, отображаемую в "нажатом" состоянии;
- красную надпись на белом фоне, отображаемую в "нажатом" состоянии, когда пользователь помещает над ним курсор мыши.

Создадим в папке Pics подпапку Navbar. Сохраним все полученные файлы в ней, дав им имена в соответствии со следующим соглашением:

- *<имя страницы>*_up.gif — для изображений, отображаемых в обычном состоянии;
- *<имя страницы>*_over.gif — для изображений, отображаемых, когда пользователь помещает над ними курсор мыши;
- *<имя страницы>*_down.gif — для изображений, отображаемых в "нажатом" состоянии;
- *<имя страницы>*_overdown.gif — для изображений, отображаемых в "нажатом" состоянии, когда пользователь помещает над ним курсор мыши.

Всего у вас должно получиться 20 файлов изображений. Что ж, вам придется здорово потрудиться... (Вообще-то для таких дел лучше использовать специализированные пакеты Web-графики, например Macromedia Fireworks. Они автоматически формируют все необходимые изображения для полос навигации и создают соответствующий HTML-код и JavaScript-код, "оживляющий" их. Но описание этих пакетов выходит за рамки данной книги.)

Подготовив изображения, задайте параметры Web-страницы. Как и в других страницах, отображаемых в остальных фреймах, цвет фона будет черным, цвет текста — белым... Короче говоря, подставьте те же значения, что и у страницы основного содержимого. Помните: все страницы сайта должны выглядеть идентично.

Чтобы создать полосу навигации, поставьте текстовый курсор в левый фрейм и нажмите кнопку **Navigation Bar** (рис. 7.22) страницы **Common** пане-

ли объектов. Вы также можете выбрать пункт **Navigation Bar** подменю **Interactive Images** меню **Insert**. На экране появится диалоговое окно **Insert Navigation Bar**, показанное на рис. 7.23.

Рис. 7.22. Кнопка **Navigation Bar** панели объектов

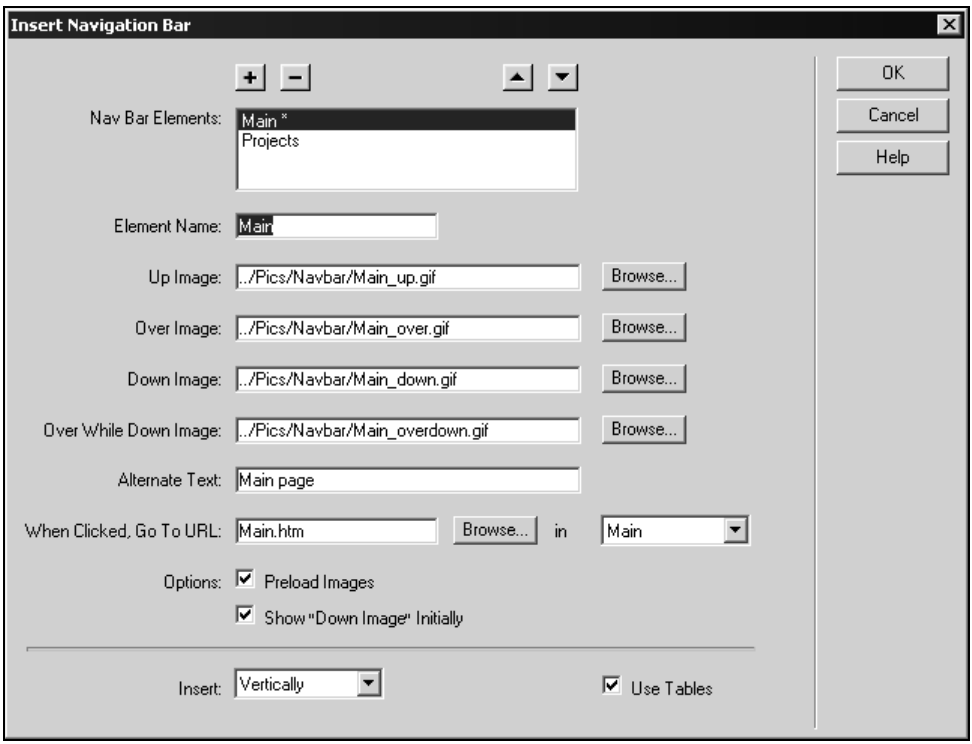


Рис. 7.23. Диалоговое окно **Insert Navigation Bar**

В списке **Nav Bar Elements** перечислены все элементы полосы навигации, которые уже имеются. (Изначально там находится один элемент, созданный для нас Dreamweaver.) Вы можете выбрать любой из элементов и посмотреть либо изменить его параметры.

В поле ввода **Element Name** вводится имя элемента. Советуем давать элементам "говорящие" имена, например `Projects` или `Links`.

В поле ввода **Up Image** вводится имя файла изображения, отображаемого в обычном случае. Вы также можете щелкнуть кнопку **Browse**, расположенную справа от поля ввода, и выбрать нужный файл в диалоговом окне от-

крытия файла Dreamweaver. Точно так же в поле ввода **Over Image** вводится имя файла изображения, отображаемого, когда пользователь помещает над гиперссылкой курсор мыши, в поле **Down Image** — имя файла изображения "нажатой" гиперссылки, а в поле **Over While Down Image** — имя файла изображения "нажатой" гиперссылки, над которой пользователь поместил курсор мыши. Справа от каждого поля ввода находится спасительная кнопка **Browse**.

В поле ввода **Alternate Text** вводится альтернативный текст. К сожалению, Dreamweaver неправильно обрабатывает русские буквы, поэтому вам или придется править сам код HTML, или вводить альтернативный текст на английском, как это сделали мы.

В поле ввода **When Clicked, Go To URL** вводится имя файла Web-страницы, на которую осуществляется переход при щелчке на гиперссылке. В раскрывающемся списке **in**, расположенном справа от поля ввода, выбирается фрейм, в котором будет отображена страница. Пункт **Main Window** этого списка позволяет открыть страницу во всем окне.

Если вы хотите, чтобы данный элемент полосы навигации изначально отображался "нажатым", включите флажок **Show "Down Image" Initially**, находящийся в группе **Options**. В частности, его нужно включить для элемента, обозначающего начальную страницу (в нашем случае это страница Main.htm).

А вот флажок **Preload Images**, находящийся в этой же группе, лучше всегда держать включенным. Он указывает Dreamweaver создать код, заставляющий Web-обозреватель загружать заранее все изображения, задействованные в полосе навигации, и сохранять их на жестком диске в своем кэше. Благодаря чему подстановка нужных изображений будет происходить мгновенно, не ожидая, пока они загрузятся с сайта. В противном случае Web-обозреватель будет вынужден загружать каждое изображение непосредственно перед его отображением. С одной стороны, это позволит ускорить загрузку страницы (не нужно будет загружать все изображения полосы навигации), а с другой — замедлит реакцию полосы навигации на действия пользователя, т. к. Web-обозревателю придется каждый раз загружать с сайта нужное изображение, а не брать его с кэша.

Раскрывающийся список **Insert** позволит задать расположение полосы навигации: горизонтальное (пункт **Horizontally**) или вертикальное (пункт **Vertically**). Флажок **Use Tables** заставит Dreamweaver создать полосу навигации на основе таблиц. Этот флажок по умолчанию включен, и отключать его не стоит.

Осталось рассказать о кнопках, расположенных над списком **Nav Bar Elements**. Кнопка со знаком "плюс" добавляет новый элемент в полосу навигации, а кнопка со знаком "минус" удаляет выбранный в списке элемент. Кнопка ↑ перемещает выбранный в списке элемент на строку выше, а кнопка ↓ — на строку ниже.

Закончив формирование полосы навигации, нажмите кнопку **OK**. Если вы передумали вставлять полосу навигации, нажмите кнопку **Cancel**.

Введите в окно вставки полосы навигации данные о пяти элементах полосы навигации согласно табл. 7.3. В поля ввода **Up Image**, **Over Image**, **Down Image** и **Over While Down Image** введите соответствующие имена файлов; если вы дали им такие имена, какие автор советовал, у вас не будет проблем с поиском необходимого файла. Проследите при этом, чтобы в раскрывающемся списке **Relative To** диалогового окна **Select File** был выбран пункт **Document** (отсчет интернет-адресов относительно текущей страницы) — задание адресов от корневой папки сайта работает только под программой Web-сервера. Элементы должны быть введены в таком порядке, в каком они перечислены в таблице. (Это общепринятый порядок перечисления элементов полосы навигации в подобных сайтах.)

Таблица 7.3. *Данные полосы навигации*

Страница	Element Name	When Clicked, Go To URL
Главная	`Main`	`Main.htm`
Проекты	`Projects`	`Projects.htm`
Увлечения	`Passions`	`Passions.htm`
Ссылки	`Links`	`Links.htm`
Об авторе	`About`	`About.htm`

В раскрывающемся списке **in** должен быть выбран пункт **Main** (имя фрейма основного содержимого сайта). Для элемента `Main` (страница основного содержимого сайта, отображаемая по умолчанию) включите флажок **Show "Down Image" Initially**, т. к. он должен быть по умолчанию "нажатым". В раскрывающемся списке **Insert** выберите пункт **Vertical**, т. к. полоса навигации в нашем случае должна быть вертикальной. Остальные элементы управления не трогайте. После этого нажмите кнопку **OK**.

Осталось немного подредактировать фрейм и его содержимое вручную. Установите значения параметров **Margin Width** и **Margin Height** для фрейма в ноль, чтобы не было этих отвратительных отступов слева и сверху. Потом выделите таблицу и установите параметр **Width** в 100%, чтобы она заняла всю ширину фрейма. Осталось выделить все ячейки таблицы и выставить параметр **Horz** в **Center**, а **Vert** — в **Middle**. Вы помните, что обозначают эти параметры?

Посмотрите на рис. 7.24. Как видите, Dreamweaver здóрово постарался, сделав за нас всю сложную работу по созданию "живой" полосы навигации. Если вы просмотрите сгенерированный им HTML-код, то увидите, как

много работы пришлось ему проделать: сформировать таблицу, разместить в ней все указанные вами графические изображения, превратить их в гиперссылки и — главное — создать сценарии, которые будут ими управлять. Согласитесь, вручную такое не сразу сделаешь.

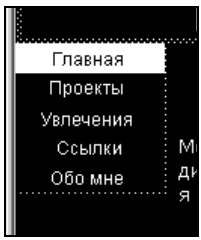


Рис. 7.24. Готовая полоса навигации

Теперь сохраните все фреймы, выбрав пункт **Save All Frames** в меню **File**. И откройте страницу в окне Web-обозревателя. Попробуйте поместить курсор мыши над каким-нибудь элементом полосы навигации и посмотрите, что получится. Пощелкайте по нему.

Как видите, даже очень сложные вещи делаются в Dreamweaver исключительно просто. Например, полоса навигации с активными изображениями для начинающего Web-дизайнера — весьма сложная задача: нужно хорошо знать не только язык описания Web-страниц HTML, но и язык написания сценариев JavaScript. И если с HTML, как правило, проблем нет, то с JavaScript — есть...

Хорошо! Мы сделали полосу навигации и поместили ее во фрейме. Но сайт наш, как и все на свете, изменяется. Что если нам вдруг понадобится добавить новый элемент или изменить графические изображения? Для этого просто выделите полосу навигации (любой ее элемент) и выберите в меню **Modify** пункт **Navigation Bar**. На экране появится диалоговое окно **Modify Navigation Bar**, аналогичное уже знакомому вам окну **Insert Navigation Bar** (см. рис. 7.23), в котором вы сможете изменить все, что нужно.

Оптимизация фреймов

Создав наш первый работающий набор фреймов, можно подумать о его оптимизации.

Уменьшение объема и сложности HTML-кода

Если вы создаете достаточно сложные наборы фреймов, то можете столкнуться с тем, что Dreamweaver не очень оптимально формирует соответствующий HTML-код. В частности (да вы и сами это видели), Dreamweaver не

может формировать наборы более чем из двух фреймов. Судя по всему, разработчики программы облегчили себе задачу за счет увеличения объема и сложности получаемого HTML-кода. Конечно, это можно стерпеть, если учесть, как хорошо Dreamweaver справляется со своими задачами в других случаях, но все же проблема остается.

Взгляните на код, сформированный Dreamweaver для нашего набора фреймов (автор слегка сократил его, убрав не относящиеся к делу атрибуты тегов):

```
<FRAMESET ROWS="*,54">
  <FRAMESET ROWS="30,*">
    <FRAME NAME="Header" SRC="/HTMLs/Header.htm">
    <FRAMESET COLS="100,399">
      <FRAME NAME="Nav" SRC="/HTMLs/Nav.htm">
      <FRAME NAME="Main" SRC="/HTMLs/Main.htm">
    </FRAMESET>
  </FRAMESET>
  <FRAME NAME="Copyright" SRC="/HTMLs/Copyright.htm">
</FRAMESET>
```

Как видите, Dreamweaver формирует только простые наборы из двух фреймов. А если фреймов в одном простом наборе должно быть больше, используются сложные наборы с многократным вложением.

В частности, нам был нужен набор из трех горизонтальных фреймов. Поскольку Dreamweaver формирует только двухфреймовые простые наборы, он создал два вложенных набора, по два фрейма в каждом. (Соответствующий код выделен полужирным шрифтом.) Давайте немного упростим этот фрагмент кода. У нас получится вот что:

```
<FRAMESET ROWS="30,*,54">
  <FRAME NAME="Header" SRC="/HTMLs/Header.htm">
  <FRAMESET COLS="100,399">
    <FRAME NAME="Nav" SRC="/HTMLs/Nav.htm">
    <FRAME NAME="Main" SRC="/HTMLs/Main.htm">
  </FRAMESET>
  <FRAME NAME="Copyright" SRC="/HTMLs/Copyright.htm">
</FRAMESET>
```

Если раньше у нас была трехкратная вложенность фреймов, то теперь мы обошлись двукратной, совместив два двухфреймовых простых набора в один трехфреймовый. (Этот фрагмент кода выделен полужирным шрифтом.) Как видите, HTML-код набора фреймов стал проще и компактнее.

Откройте страницу default.htm в окне документа, переключитесь в режим кода и внесите эти изменения, после чего снова переключитесь в режим

страницы. Что мы видим? В окне документа ничего не изменилось. Как видите, Dreamweaver правильно обрабатывает простые наборы из любого количества фреймов. Вы можете щелкать по ним мышью в окне фреймов, устанавливать значения параметров в редакторе свойств, заполнять их текстом в окне документа. Более того, если вы выберете набор из трех фреймов, Dreamweaver правильно отобразит его структуру в редакторе свойств (рис. 7.25). Получается, что Dreamweaver все-таки поддерживает простые наборы с количеством фреймов более двух.

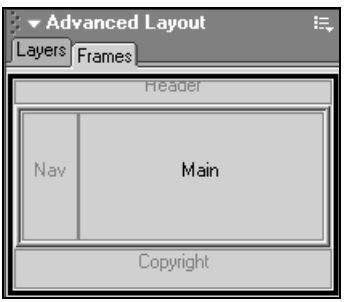


Рис. 7.25. Структура набора из трех фреймов

Ускорение обработки фреймов

Приведем несколько советов, которые помогут Web-обозревателю быстрее обработать и отобразить наборы фреймов.

1. При указании размеров фреймов (высоты или ширины) по возможности используйте абсолютные (в пикселах), а не относительные (в процентах) единицы измерения. Помните, что для того, чтобы отобразить фрейм, Web-обозревателю нужно сначала получить его абсолютный размер. А если вы указали размер фрейма в относительных единицах, Web-обозреватель должен будет сначала вычислить соответствующий размер своего окна, что займет дополнительное время.
2. Внимательно проверяйте корректность интернет-адресов в атрибутах `SRC` тегов `<FRAME>`. Мало того, что обращение по несуществующему интернет-адресу занимает много времени, так один фрейм вашего набора останется "пустым". А нет ничего отвратительнее "пустого" фрейма.
3. Задавайте замещение `<NOFRAMES>` только в тех случаях, если оно действительно нужно. Помните, что любой лишний HTML-код замедляет загрузку Web-страницы, увеличивая ее в размерах. Уже говорилось по поводу замещений и повторять это нет смысла.
4. Оптимизируйте HTML-код, порождаемый Dreamweaver.

Решение проблем с фреймами

Как вы знаете, ничего идеального на свете не бывает. Это утверждение относится и к фреймам.

Недостатки фреймов и их преодоление

Вроде бы, фреймы всем хороши. Они позволяют разделить окно Web-обозревателя на несколько независимых "форточек", в каждой из которых может отображаться своя Web-страница. Таким образом, одна большая Web-страница делится на несколько маленьких, содержащих какую-либо часть большой страницы: заголовок, полосу навигации, сведения об авторских правах и основное содержимое. При этом когда пользователь щелкает по гиперссылке, происходит обновление только фрейма с основным содержимым; содержимое всех остальных фреймов остается неизменным.

Достоинства такого подхода очевидны. Во-первых, не нужно дублировать заголовок, полосу навигации и сведения об авторских правах на всех страницах сайта — достаточно поместить их один раз в соответствующие фреймы, где они и останутся. Во-вторых, сами страницы уменьшатся в размерах и станут быстрее загружаться.

На этом достоинства фреймов кончаются, и начинаются недостатки.

Недостаток первый: невозможность обновления содержимого сразу двух или нескольких фреймов. Давайте рассмотрим гипотетический сайт — каталог программ, структура фреймов которого показана на рис. 7.26.

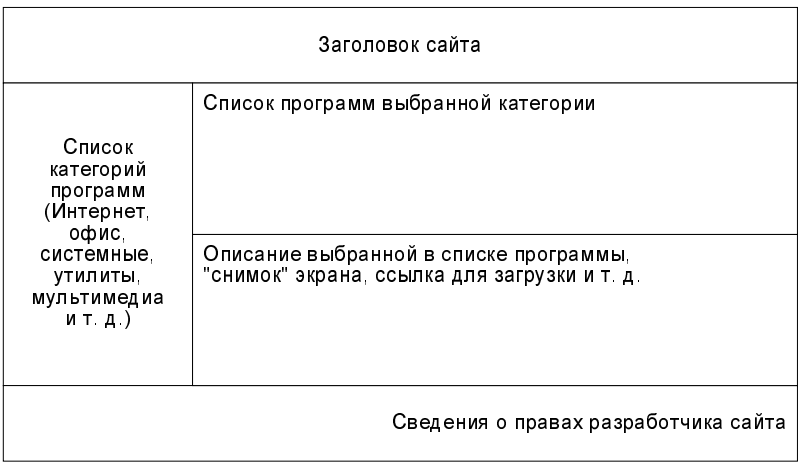


Рис. 7.26. Структура фреймов сайта — каталога программ

Предположим, пользователь щелкает одну из позиций списка категорий, расположенного в левом фрейме. Вам необходимо обновить содержимое

фрейма списка программ, относящихся к выбранной категории, и вы это сделаете. Но, кроме того, нужно будет обновить содержимое фрейма описания выбранной программы, скажем, вывести описание первой или самой популярной программы в списке. Ведь если оставить в нем описание программы, относящейся совсем к другой категории, это будет, мягко говоря, некрасиво.

К несчастью, средствами HTML это сделать невозможно. Обычная гиперссылка HTML, описываемая парным тегом <A>, может указывать только на один файл. Если вы хотите по щелчку загружать сразу две страницы (или более — в данном случае это несущественно) в разные фреймы, вам придется писать сценарий на языке JavaScript. А это уже довольно сложная задача для начинающего Web-дизайнера. На первый раз можем посоветовать только постараться избежать использования таких наборов фреймов или же отказаться от фреймов вообще.

Недостаток второй: невозможность задания текста, который Web-обозреватель помещает в заголовок своего окна. Как вы знаете, он помещает туда название отображаемой в окне Web-страницы. Но поскольку страница набора фреймов никогда не меняется (меняется только содержимое одного из фреймов), текст в заголовке окна Web-обозревателя остается неизменным. В принципе, это не очень страшно — мало кто из посетителей сайтов обращает внимание на заголовок окна Web-обозревателя. Но если вы собираетесь выводить в заголовке окна какую-либо важную информацию, имейте это в виду.

Недостаток третий: "несовместимость". Путь фреймов к признанию комитетом WWWC был так долог и тернист, что за это время успели появиться несколько программ Web-обозревателей, их поддерживающих. Но поскольку единого стандарта на фреймы не было, поддерживали они их по-своему. В результате фреймы стали одним из самых "несовместимых" элементов HTML.

В данном случае можно посоветовать только тщательно тестировать страницы наборов фреймов в разных программах Web-обозревателей. Конечно, Dreamweaver старается генерировать максимально совместимый код, но все-таки не помешает лишний раз проверить, что он сгенерировал.

Недостаток четвертый. О нем мы поговорим в *главе 14*, когда будем рассматривать поисковые системы и "раскрутку" вашего сайта с их помощью.

Проблема с фреймами в старых версиях Navigator

А вот это уже не "функциональная особенность" некогда известной программы, а элементарная ошибка разработчиков. Старые версии Navigator неправильно отображают страницы наборов фреймов после того, как поль-

зователь изменит размеры окна. И разработчики Dreamweaver, чтобы компенсировать невнимательность разработчиков фирмы Netscape, ввели в свой Web-редактор специальную функцию исправления данной ошибки.

Откройте в окне документа Dreamweaver страницу набора фреймов и выберите в меню **Commands** пункт **Add/Remove Netscape Resize Fix**. На экране появится небольшое окно-предупреждение, кратко описывающее проблему с фреймами в Navigator и то, что Dreamweaver может сделать для ее решения (рис. 7.27).

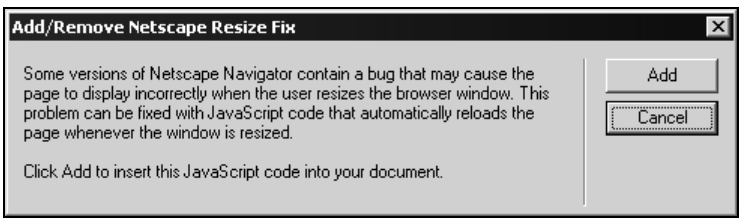


Рис. 7.27. Окно, предупреждающее о проблеме с фреймами в Navigator и предлагающее путь ее решения

Если вы нажмете кнопку **Add**, Dreamweaver добавит в код страницы небольшую программу-сценарий на языке JavaScript, перезагружающую страницу набора фреймов после изменения размеров окна Navigator. Если вы нажмете кнопку **Cancel**, Dreamweaver ничего не сделает.

Может случиться так, что вы захотите удалить эту программу-сценарий, для чего вам необходимо просто еще раз выбрать пункт **Add/Remove Netscape Resize Fix** в меню **Commands**. На экране появится другое окно-предупреждение, показанное на рис. 7.28. Нажмите кнопку **Remove** для удаления данного кода или кнопку **Cancel** — для отказа от этого.

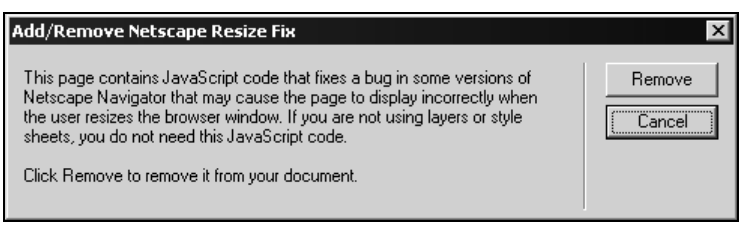


Рис. 7.28. Окно, предлагающее удалить сценарий перезагрузки страницы

Следует ли этим пользоваться? Вероятно, да. Поскольку старые версии Netscape Navigator все еще используются некоторой частью интернетчиков, нужно предусмотреть все, чтобы они могли без проблем просматривать ваши страницы. Тем более что это не отнимет у вас никаких усилий — за вас все сделает Dreamweaver. А помещаемый им сценарий очень мал и не может существенно увеличить размер страницы набора фреймов.

Что дальше?

Вот мы и познакомились с фреймами и фреймовым дизайном. И даже создали новый вариант сайта с их использованием.

Но, как уже говорилось, фреймы потихоньку устаревают. Сейчас в моде дизайн с использованием таблиц. Табличный дизайн позволяет создавать Web-страницы, больше напоминающие печатные документы, например газеты. Они включают в себя несколько колонок с текстом, множество графических иллюстраций, линеек, т. е. всего того, к чему привыкли дизайнеры-полиграфисты. Таблицы позволяют сделать все это с легкостью. А уж как Dreamweaver обращается с таблицами!

Одним словом, вы должны узнать это.

ГЛАВА 8

Табличный дизайн

Сейчас мы будем изучать принципы *табличного дизайна* — одного из способов создания сложных Web-страниц с использованием таблиц HTML. И, естественно, рассмотрим все возможности, предлагаемые Dreamweaver Web-дизайнерам, связавшим свою профессиональную судьбу с таблицами.

Сначала, опять же, создадим новый сайт, чтобы начать дело с пустого места. Поместим его в новой папке Sample3. После этого, как обычно, создадим в папке Sample3 подпапки HTMLs и Pics. И, наконец, зарегистрируем новый сайт в Dreamweaver, использовав знания, полученные в *главе 6*. Назовем этот сайт Sample site 3.

Основы табличного дизайна

Прежде чем приступать к практической работе, изучим теорию. Вспомним, зачем нужны таблицы и как их можно использовать. После этого перейдем непосредственно к табличному дизайну, его основным приемам и схемам табличного дизайна, наиболее часто используемым в Интернете.

Зачем нужны таблицы

В самом деле, зачем нужны таблицы? Казалось бы, у них достаточно узкое назначение: представлять большие объемы числовой и текстовой информации на ограниченном пространстве. Зачем еще строить на их основе целые Web-страницы? Что это нам даст?

Свободу.

Когда мы говорили о рисунках *(см. главу 4)* и фреймах *(см. главу 7)*, то упомянули о "потоке" текста. Давайте поговорим о нем подробнее.

Посмотрите на эту страницу. Полистайте книгу с начала до конца и наоборот. Что вы видите? Сплошной текст, "льющийся" по страницам и расска-

зывающий о принципах работы с Macromedia Dreamweaver. Абзацы текста, рисунки, таблицы находятся в этом "потоке" последовательно, один за другим. Они не могут "вырваться" из этого "потока". Невозможно поместить на одном и том же месте одновременно и рисунок, и таблицу — элементы страницы могут быть расположены только последовательно, один за другим.

"Поток" текста хорош для художественных произведений и руководств, последовательно описывающих какие-либо действия, необходимые для достижения результата. В этом случае текст читается с начала и до конца без резких "скачков" туда-сюда. "Поток" текста очень плох для технических статей, нуждающихся в многочисленных пояснениях, дополнениях и иллюстрациях; в этом случае нужно иметь перед глазами и первое, и второе, и третье, и четвертое. Также "поток" текста абсолютно не подходит для газет: читатель не хочет листать газету взад-вперед в поисках нужной статьи, а хочет видеть все главные новости на ее первой странице.

Вот поэтому дизайнеры постоянно ищут пути "вырваться" из жестких рамок "потока" текста. Ищут с разным успехом.

Дизайнерам-полиграфистам хорошо: они абсолютно свободны в своих действиях. Программы настольных издательств позволяют им практически все: размещать на страницах текстовые блоки, позиционировать их друг относительно друга, создавать "поток" текста, если он действительно нужен, даже создавать множество "потоков" текста, "текущих" параллельно и занимающих сколько угодно страниц. В общем, полиграфисты, что называется, неплохо устроились. И продукты их труда радуют глаз.

Если вы откроете какой-либо из современных журналов, то увидите так называемые *врезки* — небольшие фрагменты текста, помещенные в основной "поток" текста, но не привязанные к нему. Как правило, врезки оформляются отлично от основного текста, чтобы читатель сразу видел, что есть что. Кроме того, полиграфистами давно освоены различные сноски, примечания и т. п., также выбивающиеся из "потока" текста. Да и "потоков" этих в иных печатных изданиях может быть превеликое множество.

Web-дизайнерам, т. е. нам с вами, повезло значительно меньше. Язык HTML, создававшийся изначально для публикации в Сети научных текстов, поначалу даже не имел возможности размещать на страницах графику. Какие уж там сноски и врезки!.. Сейчас, конечно, HTML уже не тот, он "научился" обращаться с графикой и таблицами, но так и не "освоил" сноски и врезки. А ведь Интернет непрерывно развивается: из сети для ученых он давно превратился в сеть для всех. И этих "всех", посещающих Сеть, уже давно не удовлетворяет унылый одномерный дизайн Web-страниц.

Выходит, нет ни одного способа вырваться из "потока" текста?

Есть.

Существуют фреймы. Они позволяют разбить "поток" на несколько независимых "ручейков": основное содержание, полоса навигации, заголовок сайта

и сведения об авторских правах помещаются в отдельные Web-страницы и отображаются в отдельных "форточках" окна Web-обозревателя. Причем, эти "ручейки" настолько независимы друг от друга, что теряют связь с общим "истоком". Фактически они связаны друг с другом только общим содержимым, но, с точки зрения самого Web-обозревателя, не связаны друг с другом никак. Web-обозревателю глубоко безразлично, что отображать в соседних фреймах: страницы с одного и того же сайта или с разных, расположенных в различных концах земного шара.

Фреймы слишком уж независимы друг от друга. И примечания, врезки, сноски и прочие типографские "вкусности" с помощью фреймов не создашь. Нужно что-то другое, "отвязывающее" их от "потока" текста, но не полностью. Требования крайне противоречивые, но ничего не поделаешь — они таковы.

И тогда на помощь приходят таблицы.

В *главе 5*, когда говорилось об особых применениях таблиц, мы всячески экспериментировали с текстом: делали отступы слева и справа, линейки, фоновый узор и т. п. На самом деле, таблицы могут гораздо больше. Например, с их помощью мы можем разместить несколько фрагментов текста так, как нам нужно. И это вполне могут быть основной текст и комментарии к нему.

Взгляните на рис. 8.1. Что вы на это скажете?

Абзац 1	Примечание 1
Абзац 2	Примечание 2
Абзац 3	Примечание 3

Рис. 8.1. Создание примечаний к тексту с помощью таблиц

Как видите, используя таблицу, мы можем разделить текст на несколько фрагментов и поместить рядом соответствующие примечания. Мы можем сделать линии между фрагментами невидимыми, так что посетитель нашего

сайта и не заметит, что для форматирования этого текста использовались таблицы. (Опытный интернетчик, знакомый с HTML, конечно, догадается, ну и бог с ним.) Также мы можем сделать невидимыми линии между фрагментом текста и примечанием, выровнять текст примечания вправо, выделить его курсивом и сделать еще что-нибудь, дабы логически отделить его от основного текста. Но это уже детали. Главное: мы получили то, что хотели.

Значит, есть выход из положения!

Приведем еще один пример. Предположим, мы делаем интернет-магазин. Нам нужно поместить на страницу название, описание, цену и фотографию товара. С таблицами это проще простого — см. рис. 8.2.

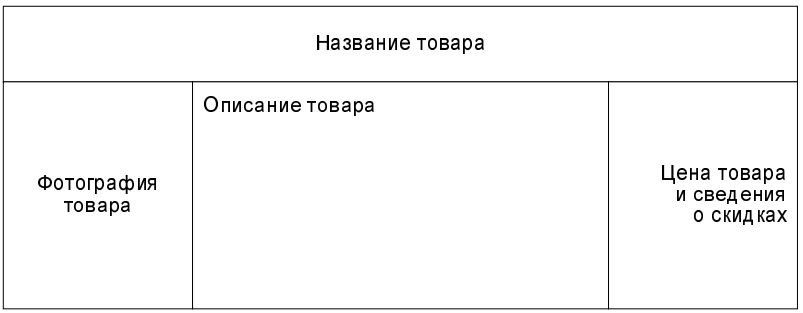


Рис. 8.2. Описание товара в интернет-магазине, сделанное с помощью таблицы

Видите, как компактно и изящно у нас получилось! С помощью специально отформатированной таблицы мы разместили несколько фрагментов текста и графическое изображение так, как нам нужно. А если нам что-то не понравится, мы можем слегка переделать эту таблицу и получить совершенно другой результат.

Web-дизайнеры давно освоили таблицы. И так же давно носилась в воздухе простая, как все гениальное, идея. А что если поместить ВЕСЬ текст Web-страницы в большую, сложно отформатированную таблицу, "растянув" ее на все окно Web-обозревателя? Ведь тогда мы получим практически неограниченные возможности, почти такие же, как у наших коллег-полиграфистов. Мы сможем создавать и примечания, и сноски, и врезки, и множественные "потоки" текста, которые будут начинаться и прерываться там, где нам нужно. Да это произведет революцию в Web-дизайне!

Вот мы и подошли вплотную к табличному дизайну, т. е. способу построения Web-страниц с использованием таблиц.

Основной принцип табличного дизайна уже был приведен. Весь текст и вся графика помещаются в ячейки таблицы, что позволяет делать с ними все, что угодно. Как правило, такие таблицы (назовем их *таблицами разметки*) имеют невидимые границы, а линейки создаются с помощью очень тонких

ячеек с фоновым заполнением. Таблицы разметки очень сложны, используют различное форматирование и многократное объединение ячеек и практически всегда — вложенные таблицы.

Создание сложных таблиц разметки вручную — высший пилотаж Web-дизайна. Не всякий даже опытный Web-дизайнер возьмется делать сложные страницы на основе таблиц. И все это из-за невероятной сложности получающегося HTML-кода. Поэтому очень часто в коде Web-страниц, построенных на основе таблиц, встречаются ошибки, из-за чего их не всегда может отобразить Web-обозреватель. В самом деле, во всех этих многочисленных объединенных ячейках и в сложнейшем форматировании можно элементарно запутаться.

И в то же время любой может значительно улучшить вид своих страничек, использовав относительно простую табличку. Проиллюстрируем это примером.

Откроем страницу default.htm нашего второго сайта, построенного на основе фреймов. Посмотрим на нее. И переделаем с использованием таблиц. Конечно, не будем сразу же делать это в Dreamweaver, а пока что нарисуем на бумаге. У нас получится нечто, похожее на рис. 8.3. Это, кстати, классическая схема табличного дизайна.

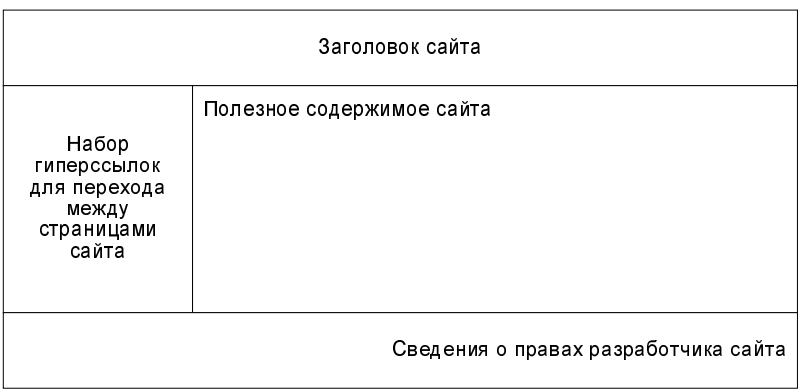


Рис. 8.3. Классический табличный дизайн

Как видите, это совсем простая таблица с двумя объединенными ячейками. Dreamweaver создаст ее для вас в два счета. А каков результат!

А ведь Dreamweaver может много больше. Он может построить для вас сколь угодно сложную таблицу и при этом ни разу не ошибется. Господа Web-дизайнеры, запутавшиеся в собственном HTML-коде, вы слышите?

А пока рассмотрим наиболее общеупотребительные схемы табличного дизайна, встречающиеся в Сети.

Схемы табличного дизайна

Классическая схема табличного дизайна, выдержавшая проверку временем, показана на рис. 8.3. Ничего лишнего — только то, что действительно нужно. Иногда, правда, полосу навигации помещают справа — это выглядит необычно и довольно удобно (рис. 8.4). Автор сам как-то делал такой сайт.

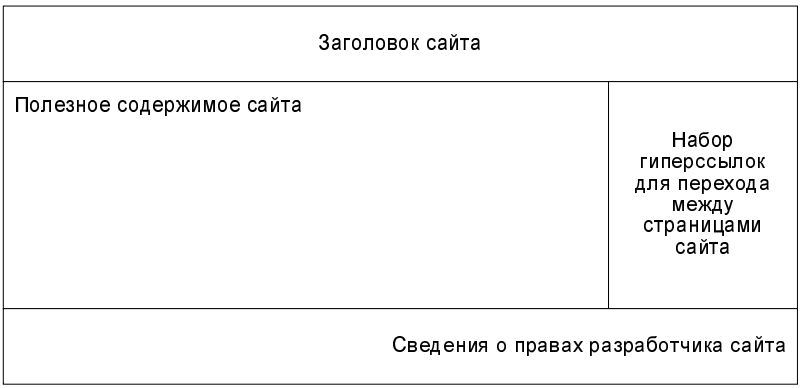


Рис. 8.4. Классический дизайн с полосой навигации, расположенной справа

Часто на первую страницу в отдельной колонке помещают еще и новости сайта (рис. 8.5). Но это стоит делать только тогда, когда ваш сайт часто обновляется, и посетителям нужно знать об этих обновлениях. Впрочем, сейчас все более-менее значимые сайты имеют колонку или раздел новостей.

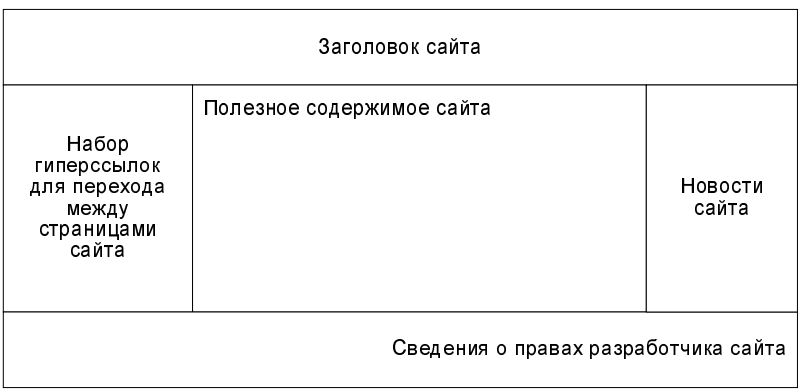


Рис. 8.5. Классический табличный дизайн с колонкой новостей сайта

Как правило, на главной странице помещаются только "свежие" новости (за определенный период), а "старости" располагают на специальной странице архива новостей. Ссылка на эту страницу размещается внизу или, реже, на-

верху колонки новостей. А сами новости и в колонке на главной странице, и на странице архива перечисляются в обратном порядке: от самых "свежих" до самых "старых".

Сайты-каталоги программ, которых очень много в Интернете, имеют весьма похожий дизайн — см. рис. 8.6.

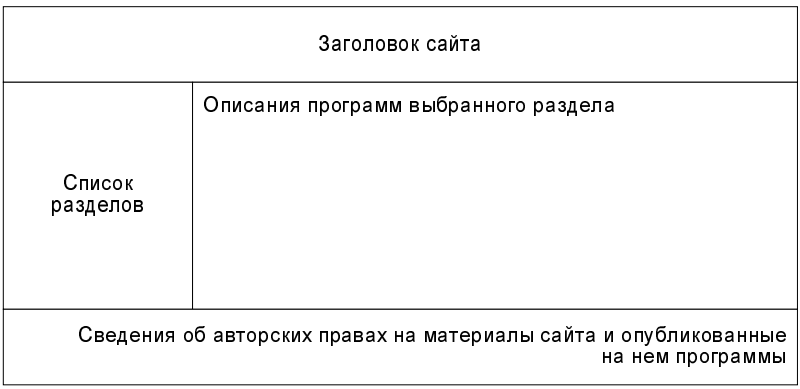


Рис. 8.6. Дизайн сайта-каталога программ

Новостные сайты имеют самый сложный дизайн (рис. 8.7). Состоящие из множества пестрых колонок Web-страницы сразу привлекают внимание посетителей.

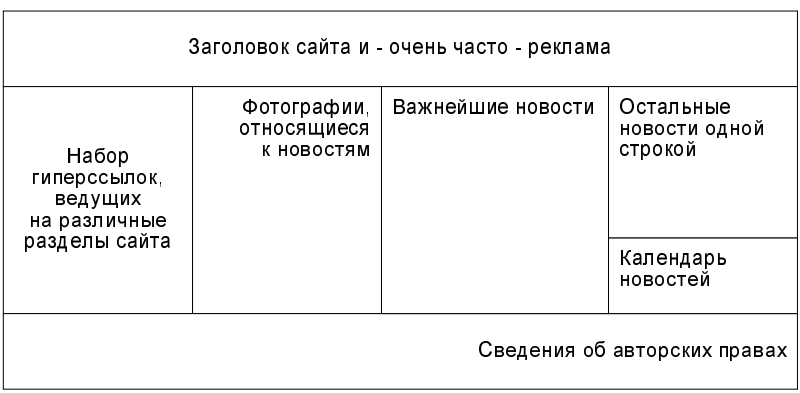


Рис. 8.7. Один из вариантов дизайна новостного сайта

Не менее, а иной раз и более пестрый дизайн имеют музыкальные сайты (рис. 8.8). Порядок следования колонок может меняться, но, в целом, основные элементы главной страницы перечислены полностью.

Надо сказать, что описанные выше схемы применяются, как правило, к главной странице сайта. В самом деле, главная страница современных

сайтов часто настолько перегружена разнородной информацией, что без сложной таблицы разметки не обойтись. Остальные страницы, поскольку они всегда значительно проще начальной, строятся по простейшим, проверенным временем схемам, представленным на рис. 8.3 и 8.4.

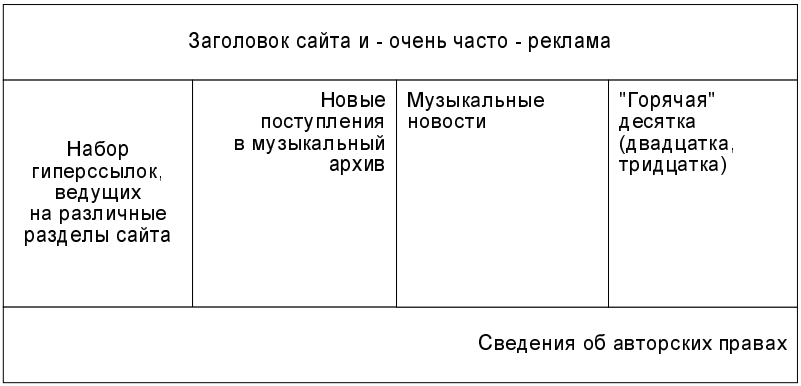


Рис. 8.8. Один из вариантов дизайна музыкального сайта

На многих главных страницах, построенных с применением табличного дизайна, помещаются составные изображения. Они играют роль либо красивого графического заголовка сайта, либо полосы навигации, либо и того и другого. Нередко организуется отдельная колонка или ячейка таблицы разметки, предназначенная для рекламных баннеров, аналогично специальным "рекламным" фреймам.

Очень часто в табличном дизайне используются пустые ячейки фиксированной ширины или высоты с заданным цветом фона либо вообще без фона (т. е. цвет фона такой же, как и у страницы). С помощью первых создаются линейки, а с помощью вторых — *просветы*, визуально отделяющие один фрагмент от другого. Как правило, линейки или просветы помещаются между близко находящимися фрагментами текста, между текстом и полосой навигации, текстом и заголовком. Иногда помимо использования линеек или просветов Web-дизайнер по-разному форматирует соседние фрагменты текста, например, задает им выравнивание в противоположные стороны или меняет цвет фона ячеек, в которых они находятся.

Порой случается, что Web-страница не содержит ничего, кроме текста, и этот текст помещен в таблицу. Это делается либо с целью как-то особым образом его оформить (сделать отступы, линейки, создать графический фон), либо с целью дальнейшего "развития". Вообще, это плохой тон в Web-дизайне. Вспомните, что мы говорили о недостатках таблиц: во-первых, они очень громоздки сами по себе, а во-вторых, пока Web-обозреватель не загрузит весь код таблицы, он зачастую не сможет ее отобразить. Так что если текст, помещенный в таблицу, достаточно велик, посетителю придется

ждать очень долго. В качестве выхода можно предложить либо разбить большую страницу на несколько меньших, либо вообще отказаться от таблиц.

Как правило, таблица разметки занимает всю ширину окна Web-обозревателя, т. е. Web-дизайнер занимает под содержимое всю доступную ему полезную площадь. Иногда, правда, содержимое подобных таблиц трудно правильно отформатировать; при изменении размеров окна содержимое "плывет", отдельные его части изменяют местоположение. В этих случаях используют таблицы с жестко заданной шириной, рассчитанные на определенное разрешение экрана и на то, что пользователь будет просматривать такие страницы в максимизированном (занимающем весь экран) окне. Иногда делают несколько вариантов страниц для разных разрешений экрана, но это очень трудоемкий процесс, поэтому так поступают очень редко.

Опять же повторим, что перечислить все схемы таблиц разметки просто невозможно — их в Сети столько же, сколько и сайтов. (А сайтов в Сети много...) Здесь приведены только самые общие рекомендации. Если вы хотите изучить Web-дизайн, вам потребуются другие, специальные книги. И еще: смотрите, как ваши коллеги делают свои сайты, решайте, нравятся ли они вам, удачны ли, на ваш взгляд. И заимствуйте удачные решения. Но осторожно, не нарушая ничьи авторские права.

Ну, все. С теорией покончено. Запускайте ваш любимый Dreamweaver — будем делать новый сайт.

Построение таблиц разметки вручную

Сейчас мы рассмотрим самый очевидный способ построения таблиц разметки. (Самый очевидный, но не самый простой.) Это создание "скелета" нашей страницы вручную, т. е. помещение на Web-странице обычной таблицы HTML и форматирование ее специальным образом. Такой способ идеален для небольших таблиц разметки и, следовательно, для простых сайтов.

Проверьте, создали ли вы новую папку Sample3 и ввели ли в Dreamweaver сведения о новом сайте Sample site 3. Папки Sample1 и Sample2 пока не удаляйте — мы позаимствуем из их содержимого информацию для нового сайта.

И давайте решим, какой дизайн мы выберем для сайта.

Конечно, можно сделать такой же стильный "белый-на-черном" сайт, как и предыдущий. Но, хоть он и красив, от просмотра таких "негативных" страниц быстро устают глаза. Отсюда следует еще один полезный совет: не используйте такую цветовую гамму для больших страниц с обилием мелкого текста — посетителю будет очень неудобно его читать. Для обширных тек-

стовых документов лучше использовать классическую цветовую гамму "черный-на-белом".

На такой цветовой гамме мы и остановимся. Конечно, мы подберем подходящие цвета для гиперссылок, чтобы они сразу бросались в глаза. Также мы особым образом отформатируем вспомогательный текст: заголовок и сведения об авторских правах, чтобы визуально отделить его от основного содержания. Но сделаем это по ходу дела.

Что касается схемы табличного дизайна, то лучше всего подходит схема, изображенная на рис. 8.3.

А теперь — за работу!

Переключитесь на новый сайт, создайте новый файл default.htm и откройте его в окне документа Dreamweaver. Откройте диалоговое окно **Page Properties**. Задайте для новой страницы следующие параметры:

- цвет фона — белый (`#FFFFFF`);
- цвет текста — черный (`#000000`);
- цвет гиперссылок — темно-красный (`#990000`);
- цвет посещенных гиперссылок — темно-серый (`#666666`);
- цвет активной гиперссылки — ярко-красный (`#FF0000`);
- название страницы — "Иван И. Иванов".

А теперь поместите на странице таблицу. Таблица эта должна содержать три строки и два столбца (см. рис. 8.3) и не должна иметь границы. Ширина этой таблицы должна быть равна 100%, т. е. таблица должна занимать все окно Web-обозревателя по ширине; высоту же не задавайте. Все остальные параметры оставьте по умолчанию.

Теперь откройте в новом окне документа Dreamweaver страницу default.htm нашего второго сайта (Sample site 2). В окне фреймов выберите внешний набор фреймов, в редакторе свойств выберите верхний фрейм (заголовок сайта) и прочитайте и запомните его высоту. После этого переключитесь в окно со страницей default.htm нового сайта, выберите верхнюю строку таблицы и задайте ее высоту равной высоте верхнего фрейма. Повторите то же самое с нижним фреймом (сведения об авторских правах). Высоту средней строки не задавайте.

Теперь переключитесь назад в окно, где открыта страница default.htm второго сайта, и выберите в окне фреймов внутренний набор. В редакторе свойств выберите левый фрейм (полоса навигации) и прочитайте его ширину. Переключитесь в окно новой страницы, выделите левый столбец и установите его ширину равной ширине левого фрейма. Ширину правого столбца не задавайте.

После этого выделите обе ячейки верхней строки и слейте их в одну. Точно так же слейте в одну обе ячейки нижней строки.

В результате у вас должно получиться нечто, напоминающее набор фреймов страницы default.htm второго сайта, но сделанное с помощью таблицы. Сохраните полученную страницу. И приступайте к форматированию ячеек. Страницу default.htm второго сайта пока не закрывайте — мы позаимствуем оттуда содержимое нашей новой страницы.

Поставьте текстовый курсор в верхнюю объединенную ячейку. Задайте такие значения ее параметров:

- горизонтальное выравнивание (раскрывающийся список **Horz**) — **Center**;
- вертикальное выравнивание (**Vert**) — **Middle** (или **Default**);
- не переносить текст (флажок **No Wrap**) — включено.

Остальные параметры оставьте без изменений.

Переключитесь в окно, где открыта страница default.htm второго сайта, скопируйте текст верхнего фрейма, переключитесь обратно в окно с новой страницей и вставьте этот текст в верхнюю ячейку. Измените цвет текста с черного, скажем, на синий, чтобы сделать заголовок приметным. Вы можете также изменить цвет фона ячейки таблицы, но не переусердствуйте, иначе заголовок станет слишком ярким и "забьет" остальное содержимое страницы.

Теперь займемся нижней объединенной ячейкой, где будут находиться сведения об авторских правах. Установите в ней текстовый курсор и задайте следующие значения параметров:

- горизонтальное выравнивание — **Right**;
- вертикальное выравнивание — **Middle** (или **Default**).

Скопируйте в эту ячейку текст из нижнего фрейма страницы default.htm второго сайта. Если хотите, измените форматирование текста и (или) ячейки.

Сделать полосу навигации можно двумя способами. Во-первых, можно просто создать ее заново, повторив все шаги, описанные в предыдущей главе. Во-вторых, ее можно скопировать со страницы default.htm второго нашего сайта, как мы скопировали заголовок и сведения об авторских правах. Согласитесь — это намного проще. Так мы и поступим.

Но, прежде всего, сделайте следующее. Откройте окно Проводника Windows и найдите папку нашего второго сайта (ее имя — Sample2, или, как там вы ее назвали). В ней найдите подпапку Pics, а в подпапке Pics — подпапку Navbar, где сохранены файлы графических изображений, использовавшихся в полосе навигации второго нашего сайта. Скопируйте эту папку со всем ее содержимым в подпапку Pics папки нашего нового сайта (Sample3). Дело в том, что при копировании полосы навигации в новую страницу нового сайта вместе с ней не копируются файлы изображений, и нам придется позаботиться об этом самим.

Теперь поставьте текстовый курсор в левую ячейку таблицы и установите следующие значения параметров:

- горизонтальное выравнивание — **Center**;
- вертикальное выравнивание — **Top**.

Теперь скопируйте полосу навигации со старой страницы и вставьте в эту ячейку. После этого Dreamweaver выдаст ряд предупреждений, что он не может скопировать сценарии, связанные с этой полосой навигации. Чтобы он вновь создал все сценарии, выделите либо саму таблицу, где помещается полоса навигации, либо одну из ее ячеек, либо одно из графических изображений и выберите пункт **Navigation Bar** меню **Modify**. На экране появится уже знакомое вам диалоговое окно **Modify Navigation Bar**.

Но не спешите нажимать кнопку **OK** — нам еще нужно изменить пути доступа к графическим файлам и файлам Web-страниц. Это можно сделать и вручную. Но проще всего щелкнуть кнопку **Browse** одного из полей ввода, выбрать нужный файл в появившемся на экране диалоговом окне **Select File**, посмотреть, как изменился путь к этому файлу, и изменить так же остальные пути. После этого нажмите кнопку **OK**.

Однако работа над полосой навигации не закончена. К сожалению, Dreamweaver не может корректно изменить цели гиперссылок полосы навигации. Даже если вы зададите цели с помощью раскрывающегося списка **in** окна **Modify Navigation Bar**, он этого не сделает. Вам придется выбрать каждый элемент и вручную изменить значение параметра **Target**. Для этого выберите в раскрывающемся списке **Target** редактора свойств пункт **_self** (открытие страницы в том же окне). Только будьте внимательны и проделайте это со всеми гиперссылками полосы навигации.

Осталось поместить в правую ячейку основное содержимое. Поставьте в нее текстовый курсор и задайте такие параметры:

- горизонтальное выравнивание — **Left** (или **Default**);
- вертикальное выравнивание — **Top**.

Осталось сделать совсем немного. Скопируйте сюда содержимое со старой страницы. Подредактируйте его, если хотите. Измените цели всех находящихся здесь гиперссылок, выбрав в раскрывающемся списке **Target** пункт **_self**. И не забудьте сохранить законченную страницу.

Все! У вас должно получиться что-то, похожее на рис. 8.9. Теперь можете открыть эту страницу в окне Web-обозревателя и протестировать ее, пощелкав по гиперссылкам. Ничего, правда, от этого не ожидайте — ведь мы так и не создали остальные страницы этого сайта. Если хотите, можете закончить его. Но именно если хотите — у нас другие планы, которые не стоит откладывать в долгий ящик.

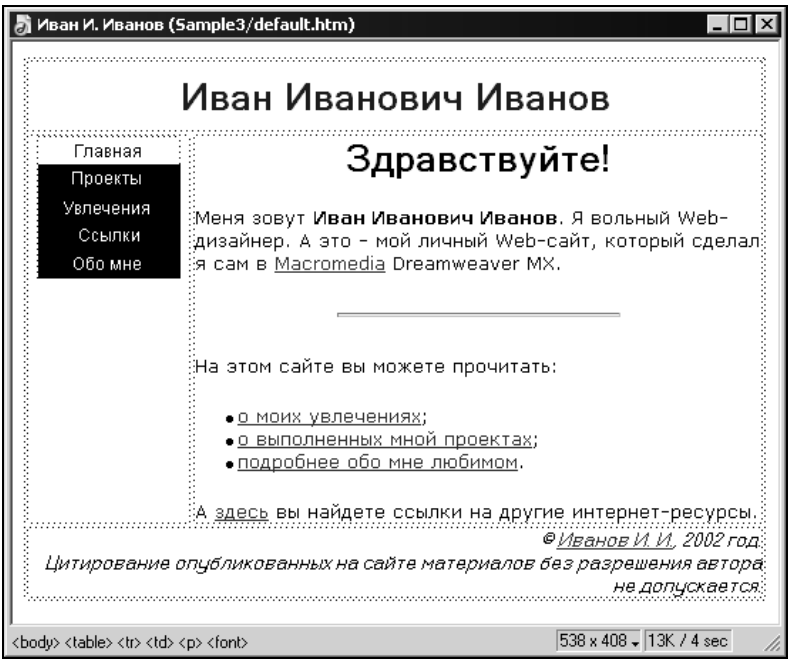


Рис. 8.9. Начальная страница нового сайта, созданная вручную с использованием таблицы разметки

Построение таблиц разметки автоматически

Мы только что создали новую Web-страницу, где для размещения содержимого использовалась таблица разметки. И мы создали эту таблицу вручную. Для этого мы сначала нарисовали схему дизайна сайта на бумаге, а потом, при создании таблицы, сверялись с нарисованной схемой. И это оказалось совсем не сложно.

Недостатки способа создания таблиц разметки вручную

Нам помогло то, что схема дизайна нашего сайта очень проста. (Конечно, это не значит, что она плоха, примитивна, убога; очень и очень многие сайты в Сети построены по этой схеме.) В самом деле, в ней нет ничего сложного: всего четыре ячейки, минимальное количество слияний и полное отсутствие вложенных таблиц. Так что схема наша совершенно очевидна, "прозрачна", как еще говорят.

Но представьте себе, что вам нужно сделать что-то сложное. Это может быть главная страница новостного или музыкального сайта с множеством колонок, сложным заголовком, врезками и сносками. Также это может быть главная страница вполне тривиального сайта с необычным дизайном — такие частенько встречаются в Сети и удивляют привыкшего к простеньким таблицам разметки посетителя. Как сделать такое вручную?

Можно, конечно, во всех деталях расписать схему такого дизайна на бумаге, подробно нарисовать все вложенные таблицы и слитые ячейки. Но сколько времени это займет! Но даже если вы и создадите подобную таблицу разметки, ваши проблемы не исчезнут. А вдруг вам понадобится что-то сюда добавить? Неужели все перерисовывать заново?!

Вы видели, как работает дизайнер-полиграфист? Он не рисует на бумаге никаких — ни сверхпростых, ни сверхсложных — таблиц. В своей любимой программе настольного издательства он просто рисует на странице "рамку", куда будет помещен тот или иной фрагмент ее содержимого, после чего вставляет туда текст из буфера обмена Windows, текстового файла или набирает его вручную. Если фрагмент слишком велик и не вмещает содержимого, он изменяет размеры рамки и, если нужно, переносит ее на следующую страницу. А если ему необходимо что-то добавить или удалить, ему надо всего лишь сделать несколько движений мышью.

Завидно... Не верится, что мы, Web-дизайнеры, сможем когда-нибудь проделывать то же самое со своими Web-страницами. Таблицы разметки поймали нас в свои ячейки, как в капканы. И мы не можем их разрушить...

Что? Неужели можем?..

Режим разметки страницы Dreamweaver

Да, можем!

Dreamweaver предоставляет нам замечательную возможность. Мы можем точно так же, как наши коллеги-полиграфисты, просто рисовать на странице "рамки" и помещать в них фрагменты текста и графические изображения. При этом Dreamweaver сделает за нас всю остальную работу: создаст таблицу разметки, если ее еще нет, и поместит в ней новую ячейку, произведя все необходимые слияния, если без них не обойтись. Мы можем с легкостью перемещать такие "рамки" с места на место и изменять их размеры, а Dreamweaver "на лету" изменит результирующий HTML-код и пересоздаст результирующую таблицу, чтобы отразить ваши изменения.

Таким образом можно создавать Web-страницы, не отличающиеся по сложности от печатных документов. Для этого нужно только переключиться в так называемый *режим разметки страницы* (фактически это четвертый режим отображения Web-страницы; три других были описаны в *главе 2*). В этом режиме Dreamweaver отображает и обрабатывает таблицы особым образом. При этом, находясь в режиме разметки, вы можете переключиться

из режима отображения страницы в режим HTML-кода и взглянуть, что натворил Dreamweaver за вашей спиной.

Но как все-таки перейти в режим разметки? И как им пользоваться?

Очень просто.

Давайте создадим новый вариант начальной страницы нашего нового сайта. Но, прежде всего, переименуйте файл default.htm сайта Sample site 3 (созданная нами ранее страница), скажем, в default_old.htm. Потом откройте эту страницу в окне документа. Мы будем часто копировать с нее фрагменты содержимого в новую страницу, поэтому лучше держать ее открытой.

Далее создайте в панели **Site** новую страницу default.htm, проверьте, является ли она главной страницей, и откройте ее в окне документа. Это и будет новая главная страница нашего сайта.

Чтобы переключиться в режим разметки страницы, нажмите кнопку **Layout View**, расположенную на вкладке **Layout** панели объектов. Слева от нее находится кнопка **Standard View**, позволяющая переключиться в обычный режим отображения страницы. Обе эти кнопки показаны на рис. 8.10. Вы также можете выбрать в подменю **Table View** меню **View** пункт **Layout View** или нажать комбинацию клавиш <Ctrl>+<F6>. Для обратного переключения выберите пункт **Standard View** в том же подменю или нажмите комбинацию клавиш<Ctrl>+<Shift>+<F6>.

Рис. 8.10. Кнопки **Standard View** и **Layout View** панели объектов

В любом случае после этого Dreamweaver выдаст объемистое предупреждение, описывающее, что вы можете делать в режиме разметки. Прочитайте его, если хотите, и закройте, нажав кнопку **Done**. Если вы не хотите, чтобы Dreamweaver выдавал это предупреждение в дальнейшем, перед закрытием его включите флажок **Don't show me this message again**. Автор рекомендует вам так и сделать.

Теперь самое время включить отображение *измерительных линеек*. Это настоящие линейки, проградуированные в одной из единиц измерения и отображаемые вдоль границ окна документа. Они позволят вам в дальнейшем более точно позиционировать фрагменты содержимого страницы. Чтобы увидеть их, включите в подменю **Rulers** меню **View** пункт-выключатель **Show** или нажмите комбинацию клавиш <Ctrl>+<Alt>+<R>. Чтобы убрать линейки, отключите этот пункт. Пункты-переключатели **Pixels** (пикселы), **Inches** (дюймы) и **Centimeters** (сантиметры) того же подменю **Rulers** меню **View** позволят вам задать нужную единицу измерения.

Теперь обратите внимание на сами линейки. В месте их пересечения в левом верхнем углу находится *точка начала отсчета*, "ноль" системы координ-

нат, иначе говоря, точка с координатами (0, 0) (см. рис. 8.11). По умолчанию она находится именно там — в левом верхнем углу. Но вы можете переместить ее в любое другое место; для этого достаточно "захватить" ее мышью и перетащить куда надо. Как только вы отпустите кнопку мыши, шкалы на линейках изменятся и станут отсчитывать все координаты от вновь заданного "нуля". Иногда это бывает полезно. А чтобы вернуть начало координат на свое законное место — в левый верхний угол окна, выберите пункт **Reset Origin** подменю **Rulers** меню **View**.

Рис. 8.11. Точка начала отсчета
в месте пересечения измерительных линеек

Еще одно средство помочь вам разместить фрагменты содержимого в нужном месте — *координатная сетка*. Чтобы отобразить ее, включите пункт-выключатель **Show Grid** подменю **Grid** меню **View** или нажмите комбинацию клавиш <Ctrl>+<Alt>+<G>. Чтобы убрать сетку, отключите этот пункт.

Можно сделать так, чтобы фрагменты содержимого, рисуемые мышью, "прилипали" к линиям сетки. Это поможет вам более точно позиционировать их по делениям шкал. Чтобы включить "прилипание", включите пункт **Snap To Grid** подменю **Grid** меню **View** или нажмите комбинацию клавиш <Ctrl>+<Alt>+<Shift>+<G>.

Если вас не устраивает шаг координатной сетки, вы можете его изменить. Для этого выберите пункт **Grid Settings** подменю **Grid** меню **View**. На экране появится диалоговое окно **Grid Settings**, показанное на рис. 8.12.

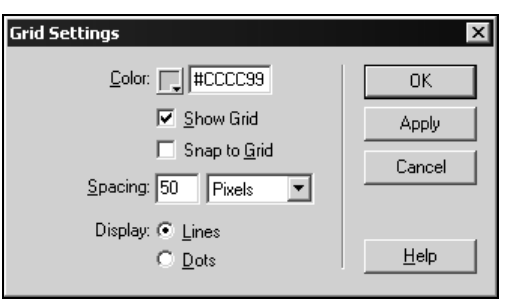


Рис. 8.12. Диалоговое окно **Grid Settings**

С помощью селектора цвета **Color** задается цвет линий сетки.

Флажок **Show Grid** включает или отключает показ сетки. Он аналогичен пункту **Show Grid** подменю **Grid** меню **View**.

Флажок **Snap to Grid** включает или отключает "прилипание" фрагментов содержания к линиям сетки. Он аналогичен пункту **Snap to Grid** подменю **Grid** меню **View**.

Поле ввода **Spacing** задает шаг линий сетки, то, что нам больше всего нужно. Раскрывающийся список, расположенный справа от него, задает единицу измерения: **Pixels** (пикселы), **Inches** (дюймы) или **Centimeters** (сантиметры). Задайте шаг сетки равным одному сантиметру — привычную для нас единицу измерения. Хотя в Web-дизайне чаще используются пикселы, сантиметры будут вам привычнее на первых порах.

Группа переключателей **Display** задает вид линий сетки. Переключатель **Lines** делает их такими, какие вы видите сейчас — сплошными линиями. Это поведение Dreamweaver по умолчанию. Переключатель **Dots** включает отображение точечных линий; в таком виде они меньше отвлекают внимание, однако хуже видны на экране.

Закончив настройку сетки, нажмите кнопку **OK** для сохранения заданных установок или **Cancel** — для отказа от них. Вы можете также нажать кнопку **Apply**, чтобы применить заданные установки без закрытия диалогового окна **Grid Settings**.

Внешний вид окна документа в режиме разметки с включенными линейками и координатной сеткой с шагом в один сантиметр показан на рис. 8.13.

Задайте параметры этой страницы как у предыдущей (default_old.htm). Теперь можно приступать к работе над разметкой.

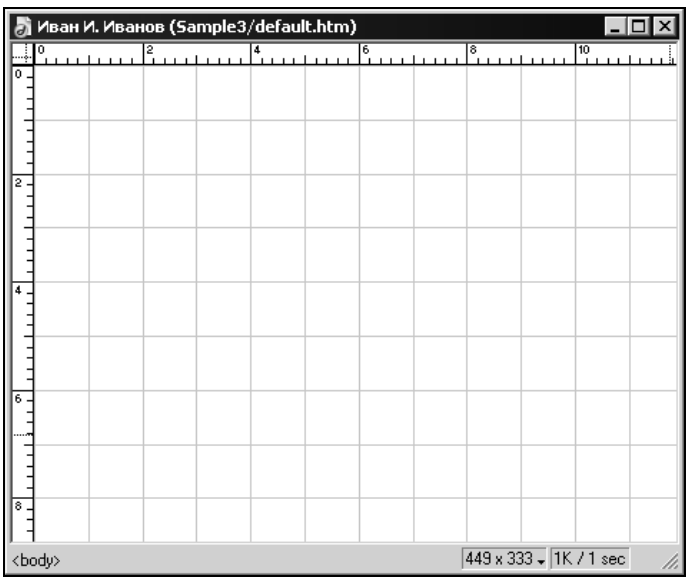


Рис. 8.13. Окно документа в режиме разметки с включенными линейками и сеткой

Таблицы и ячейки разметки

Ну, вот и все! Все подготовительные действия закончились. Можно приступать к собственно разметке.

Как уже говорилось, в режиме разметки вы можете просто мышью на странице рисовать "рамки", в которых будут размещаться фрагменты содержимого вашей страницы. Dreamweaver превратит эти "рамки" в ячейки таблиц, которые сам же сформирует. Вы можете размещать такие "рамки" где угодно на странице; единственное условие — они не должны перекрывать друг друга. Ячейки, где размещаются фрагменты содержимого страницы, называются *ячейками разметки* по аналогии с таблицами разметки.

Dreamweaver предоставляет вам возможность также размещать на странице сами таблицы разметки. Вы просто рисуете прямоугольник, который становится таблицей и помещается на странице или внутри другой таблицы разметки. Это может понадобиться, например, если вы создаете разметку, состоящую из нескольких таблиц.

Сейчас мы узнаем, как ячейки и таблицы разметки размещаются на странице. Рассмотрим это на примере новой начальной страницы нашего сайта. В общем ее схема не будет отличаться от схемы предыдущей страницы. Но, чтобы усложнить работу Dreamweaver, давайте добавим колонку новостей. Она будет располагаться справа от основного содержимого (см. рис. 8.5). В верхней части колонки новостей мы поместим ее заголовок, а в нижней — ссылку на страницу архива новостей.

Но с чего начать? С создания самой таблицы разметки? Или сразу же начинать делать ячейки разметки? В принципе, вы можете сразу же начинать с ячеек — Dreamweaver сам создаст необходимую таблицу разметки. А вы уже потом зададите ее параметры.

Посмотрите на вкладку **Layout** панели объектов. Там расположены две кнопки: кнопка **Draw Layout Cell** позволит вам поместить на страницу ячейку разметки, а кнопка **Draw Layout Table** — таблицу разметки. Эти кнопки показаны на рис. 8.14.

Рис. 8.14. Кнопки **Draw Layout Table** (слева) и **Draw Layout Cell** (справа) панели объектов

Нажмите кнопку **Draw Layout Cell** и переместите курсор мыши в окно документа. Он примет форму крестика. Это значит, что теперь вы можете нарисовать на странице новую ячейку разметки. Поставьте курсор мыши в то место, где будет угол вашей ячейки, нажмите левую кнопку и, не отпуская ее, протащите мышь в точку, где будет другой, противоположный угол

ячейки, после чего отпустите кнопку. Dreamweaver сформирует таблицу разметки и находящуюся в ней ячейку.

Создайте ячейку разметки. Она должна выглядеть так, как изображено на рис. 8.15. Это будет ячейка основного содержания.

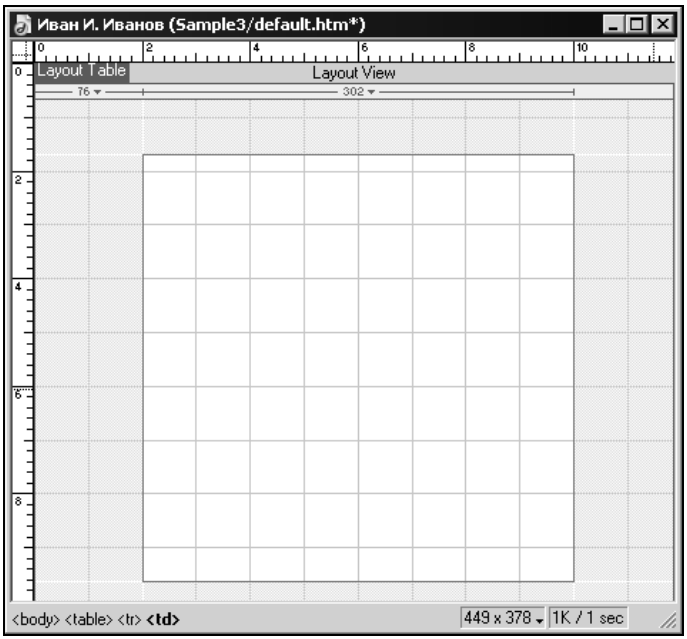


Рис. 8.15. Готовая ячейка основного содержания и таблица разметки

Получившаяся таблица будет отображена в виде большого серого прямоугольника с зеленой рамкой, а ячейка — в виде меньшего белого прямоугольника с синей рамкой, расположенного внутри серого. Кроме того, Dreamweaver ненавязчиво покажет нам структуру результирующей таблицы HTML — если вы присмотритесь, то заметите тонкие белые линии — ячейки результирующей таблицы. Но ярко изображены будут только созданная вами ячейка разметки и таблица, в которой она находится.

Поместите курсор мыши где-либо внутри этой ячейки и щелкните. Граница этой ячейки будет подсвечена, а внутри ее появится текстовый курсор; это значит, что ячейка теперь выделена, и вы можете набирать в ней текст, размещать графические изображения, таблицы и все остальное, чем богат HTML.

Теперь поместите курсор мыши на границе этой ячейки так, чтобы она (граница) стала красной, и опять щелкните. Ячейка будет выделена, и на ее границе появятся девять маркеров изменения размера (рис. 8.16). Пользуясь

этими маркерами, вы можете изменять размеры ячейки, как вам захочется. Чтобы переместить ячейку в другое место, просто "захватите" ее мышью за границу и перетащите, куда нужно. Если же ячейку по каким-либо причинам нельзя перенести на это место или изменить ее размеры, Dreamweaver очень красиво вернет все на свои места. А если вы будете перетаскивать ячейку разметки, удерживая нажатой клавишу <Alt>, ячейка будет скопирована на новое место.

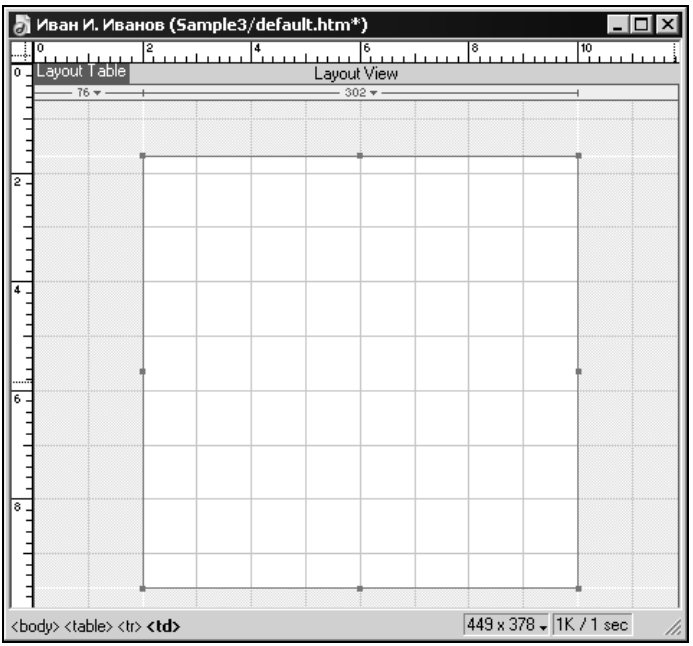


Рис. 8.16. Выделенная ячейка разметки

Если ячейка, размеры которой вы изменяете, содержит в себе что-то, и это "что-то" занимает ее целиком, вы не сможете уменьшить ее размеры. Либо уменьшите размеры содержимого ячейки (подредактируйте текст или уменьшите размеры изображений), либо "перетасуйте" другие ячейки. Также можете попробовать увеличить саму таблицу и раздвинуть ячейки, чтобы освободить пространство. (О том, как работать с таблицами разметки, *см. ниже*.)

Заметьте, что при перетаскивании или изменении размеров ячейки ее граница будет "прилипать" к линиям сетки, если, конечно, вы включили режим "прилипания". Если вы хотите временно отключить этот режим, не забираясь в меню, просто во время перетаскивания или изменения размеров ячейки нажмите и удерживайте клавишу <Alt>.

Чтобы удалить ненужную ячейку, выделите ее и нажмите клавишу <Del>.

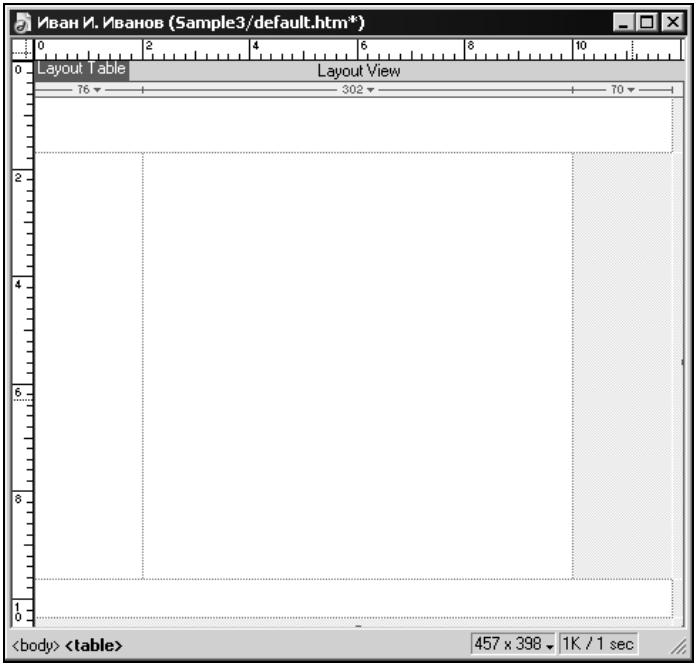


Рис. 8.17. Четыре ячейки разметки (сетка временно отключена)

Нарисуйте теперь еще три ячейки: заголовка, полосы навигации и сведений об авторских правах. Измените размеры уже созданной ячейки, если вам не будет хватать места внутри таблицы разметки. У вас должно получиться что-то, похожее на рис. 8.17.

У нас осталось свободное пространство в правой части получившейся таблицы. Давайте поместим туда новую таблицу разметки. Внутри этой таблицы мы поместим колонку новостей с заголовком и ссылкой на архив.

Создание таблицы разметки почти ничем не отличается от создания ячейки разметки за тем исключением, что щелкать нужно кнопку **Draw Layout Table**. Рисуется таблица так же, как ячейка. Существуют, правда, несколько правил, которым нужно следовать при размещении на странице новой таблицы разметки. Ниже они все перечислены.

1. Если нарисованная таблица находится внутри другой, то она становится вложенной; Dreamweaver сам сформирует необходимую ячейку во внешней таблице.
2. Если же вы рисуете таблицу вокруг уже существующих ячеек разметки, они становятся частью новой таблицы.
3. Поместить таблицу разметки внутри ячейки разметки невозможно — это можно сделать только на свободном пространстве внутри внешней таблицы или вообще вне ее.

4. Таблица разметки обязательно должна содержать ячейки, в которых будет находиться содержимое страницы. Поместить содержимое прямо в таблице невозможно — это запрещают стандарты HTML; создайте хотя бы одну ячейку, занимающую всю таблицу целиком.

Примечание

Вообще, сам Dreamweaver подскажет вам, можно ли на данном месте поместить ячейку или таблицу разметки. Если курсор мыши имеет вид крестика, то поместить на этом месте элемент разметки можно, если же он примет вид перечеркнутого круга — нельзя.

В левом верхнем углу таблицы разметки находится ее *заголовок*, имеющий вид небольшого ярлычка (рис. 8.18). Чтобы выделить таблицу, щелкните по нему. Вы можете также щелкнуть по границе таблицы, но этот способ срабатывает только тогда, когда граница таблицы не совпадает с границей ячейки или другой таблицы. Так что щелчок по заголовку — лучший способ выделить таблицу.

Рис. 8.18. Заголовок таблицы

Вы можете отключить показ заголовков таблиц, если они вам мешают. Для этого достаточно отключить пункт-выключатель **Show Layout Table Tabs** подменю **Table View** меню **View**.

Вдоль верхнего края таблицы располагается *строка ширин* (рис. 8.19). В ней обозначена ширина соответствующих столбцов таблицы. При разработке разметки страницы часто нужно знать ширину того или иного столбца.

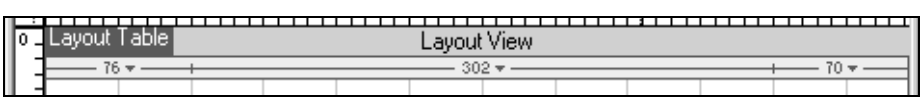


Рис. 8.19. Строка ширин

Может получиться так, что в строке ширин для какой-то ячейки будут отображаться два значения ширины: одно в кавычках и другое в скобках. Обычно это случается, когда в слишком узкую ячейку помещают слишком широкое содержимое, и Dreamweaver бывает вынужден расширить эту ячейку. Первое значение обозначает реальную ширину ячейки, а второе — ширину, заданную для ячейки в HTML-коде. В этом случае лучше всего будет изменить ширину ячейки так, чтобы оба эти значения совпали. Как это проще всего сделать, мы рассмотрим ниже.

Вы можете делать с таблицами разметки все то же, что и с ячейками: изменять размеры, переносить на другое место, удалять. Но имейте в виду, что переносить таблицу можно только "удерживая" ее за "тело", окрашенное серым, а не за границу. А изменять размеры можно только у той таблицы, внутри и вне которой с нужной стороны есть свободное пространство, не занятое ячейками.

Нарисуйте на свободном месте таблицу разметки. Внутри нее создайте три ячейки:

- верхняя — узкая, под заголовок колонки новостей;
- центральная — самая большая, под собственно новости;
- нижняя — также узкая, под гиперссылку на архив новостей ("старостей").

Результат, который должен получиться у вас, изображен на рис. 8.20.

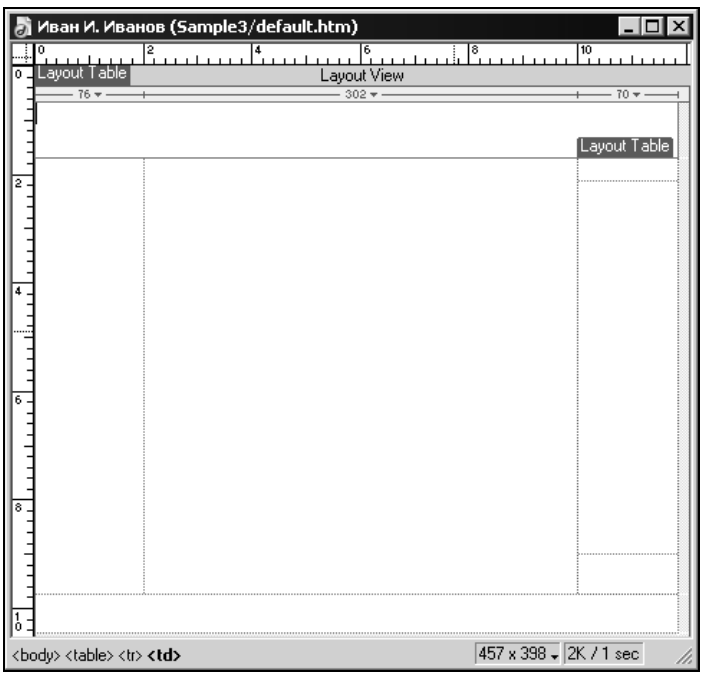


Рис. 8.20. Готовая разметка страницы (сетка временно отключена)

Примечание

Dreamweaver добавляет к тегам `<TABLE>`, определяющим таблицы разметки, специальные атрибуты, в частности атрибут `MM:LAYOUTGROUP`. Как видите, эти атрибуты можно опознать по приставке "`MM:`" (MacroMedia). Web-обозреватель обрабатывать их не будет — они нужны только самому Dreamweaver.

Форматирование таблиц и ячеек разметки

Dreamweaver позволяет задать довольно много параметров таблиц и ячеек разметки. Давайте выясним, какие именно.

Параметры ячеек разметки

Сначала поговорим, какие параметры ячеек разметки позволяет нам изменить Dreamweaver. Вид редактора свойств при выделенной ячейке разметки показан на рис. 8.21.

Рис. 8.21. Вид редактора свойств при выделенной ячейке разметки

Группа переключателей **Width** позволяет вам установить ширину ячейки. Переключатель **Fixed** и расположенное справа от нее поле ввода **Width** задают *фиксированную ширину* ячейки. Прежде чем задавать ширину ячейки, не забудьте включить вышеупомянутый переключатель. Другой переключатель в этой группе — **Autostretch** — задает режим *"саморастягивания"* ячейки, т. е. ячейка будет занимать все пространство таблицы, оставшееся от других ячеек. В таблице может быть только одна такая ячейка.

В поле ввода **Height** задается высота ячейки.

Селектор цвета **Bg** задает цвет фона ячейки.

Раскрывающийся список **Horz** позволяет задать горизонтальное выравнивание содержимого ячейки. Здесь доступны четыре пункта:

- **Default** — выравнивание по умолчанию, обычно, по левому краю. В ячейках строки заголовка текст по умолчанию выравнивается по центру (о строке заголовка *см. ниже*);
- **Left** — выравнивание по левому краю;
- **Center** — по центру;
- **Right** — по правому краю.

Аналогично, раскрывающийся список **Vert** задает вертикальное выравнивание текста в ячейке. Здесь доступно пять пунктов:

- **Default** — выравнивание по умолчанию, обычно посередине;
- **Top** — выравнивание по верху;

- **Middle** — посередине;
- **Bottom** — по низу;
- **Baseline** — по базовой линии.

Флажок **No Wrap** запрещает перенос строк внутри ячейки.

С этими элементами управления и задаваемыми ими параметрами все понятно. За исключением переключателей группы **Width** и поля ввода **Width**.

Вообще, задание ширины ячейки — тема отдельного разговора. Конечно, вы уже знаете, что ячейки могут иметь фиксированную ширину или "растягиваться" автоматически, заполняя все свободное место в таблице. Но на самом деле здесь есть очень много нюансов, вызванных особенностями работы программ Web-обозревателей и взаимовлияниями ячеек таблиц и самих таблиц друг на друга. Давайте поговорим об этом подробнее.

Задание ширины ячеек

Интерактивная справка Dreamweaver рекомендует задавать ширину ячеек только после заполнения их содержимым. По крайней мере, выставлять окончательные значения ширины лучше именно тогда: вы сможете сами увидеть, как страница будет выглядеть. Но предварительно ширину можно выставить и до заполнения ячеек, что мы сейчас и сделаем.

Итак, ячейки могут иметь фиксированную или неограниченную ширину ("саморастягивающиеся" по горизонтали ячейки). (К сожалению, не предоставляются такие же средства для управления высотой ячеек.) Только одна ячейка в таблице может быть "саморастягивающейся"; значения ширины остальных ячеек должны быть фиксированы. И это логично.

Изменить значение ширины ячейки можно простым перетаскиванием маркеров размеров на ее границе, когда ячейка выделена. Также можно ввести нужное значение прямо в поле ввода **Width** редактора свойств. Это справедливо для всех ячеек с фиксированной шириной.

Давайте превратим ячейку с основным содержанием страницы в "саморастягивающуюся". Но как это сделать?

Просто включите переключатель **Autostretch** в редакторе свойств. При этом Dreamweaver сам добавит в остальные (с фиксированной шириной) ячейки *изображения-"распорки"*, представляющие собой уже известный вам "однопиксельный GIF". Это нужно для того, чтобы выдерживать точные размеры ячеек, не заполненных или не полностью заполненных содержимым. А выбранная вами ячейка станет "саморастягивающейся".

Однако сначала Dreamweaver задаст вам один вопрос. На экране появится диалоговое окно **Choose Spacer Image**, показанное на рис. 8.22. В нем вы должны выбрать, откуда будет браться изображение-"распорка".

Рис. 8.22. Диалоговое окно **Choose Spacer Image**

Вы можете выбрать один из трех переключателей:

- **Create a spacer image file** — создать файл изображения-"распорки". Этот переключатель следует выбирать, если у вас еще нет изображения-"распорки";
- **Use an existing spacer image file** — использовать уже имеющийся файл изображения-"распорки";
- **Don't use spacer images for autostretch tables** — не использовать изображения-"распорки" вообще. Этот переключатель следует выбирать, если вы сами собираетесь принять меры против самовольного изменения ширины ячеек Web-обозревателем, например, поместить туда содержимое, заполняющее ячейки целиком. Выбирайте этот переключатель и в том случае, если вы сами собираетесь поместить в ячейки с фиксированной шириной изображение-"распорку". Если же вы не предпримете никаких мер, таблица может быть искажена Web-обозревателем до неузнаваемости, о чем Dreamweaver вас и предупредит.

Сделав свой выбор, нажмите кнопку **OK** для создания "саморастягивающейся" ячейки или **Cancel** — для отказа от этого.

Поскольку у нас нет файла изображения-"распорки", давайте прикажем Dreamweaver создать его. Выберите первый переключатель и нажмите кнопку **OK**. После этого на экране появится диалоговое окно **Save Spacer Image File As**, напоминающее стандартное диалоговое окно сохранения файла Windows. Проверьте, выбран ли в раскрывающемся списке **Relative To** пункт **Document** (задание пути и имени файла относительно страницы, а не сайта), и сохраните файл изображения-"распорки" в папке Pics. По умолчанию он будет называться spacer.gif; если хотите, измените его имя, но можно этого и не делать.

После того как вы задали файл изображения-"распорки" (создали, выбрали уже существующий или отказались от него), Dreamweaver станет его использовать, не спрашивая вас. Если вы хотите изменить имя файла изобра-

жения-"распорки" или создать его заново, закройте Dreamweaver, выбрав пункт **Exit** в меню **File**, запустите его снова и повторите все шаги по созданию "саморастягивающейся" ячейки.

Теперь обратите внимание на строку ширин в верхней части внешней таблицы разметки. Ее части, относящиеся к крайним ячейкам (фиксированные значения ширины), имеют такой вид, как на рис. 8.23; это значит, что внутрь такой ячейки помещено изображение-"распорка". А часть, относящаяся к "саморастягивающейся" ячейке, имеет вид, показанный на рис. 8.24.

Рис. 8.23. Обозначение ячейки с фиксированной шириной и помещенной в нее "распоркой" в строке ширин

Рис. 8.24. Обозначение "саморастягивающейся" ячейки в строке ширин

Чтобы вернуть "саморастягивающейся" ячейке таблицы фиксированную ширину, включите переключатель **Fixed** редактора свойств и, если нужно, задайте ширину в поле ввода **Width**.

Есть еще один способ преобразования ячейки из фиксированной в "саморастягивающуюся" и наоборот — воспользоваться *меню строки ширин*. Получить доступ к этому меню можно, щелкнув мышью точно по значению ширины ячейки (рис. 8.25). Чтобы преобразовать фиксированную ячейку в "саморастягивающуюся", выберите пункт **Make Column Autostretch**. Чтобы вернуть "саморастягивающейся" ячейке фиксированную ширину, выберите пункт **Make Column Fixed Width**.

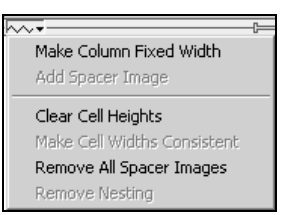


Рис. 8.25. Меню строки ширин

Меню строки ширин также содержит ряд пунктов, позволяющих вам произвести различные манипуляции с изображениями-"распорками". Это пункты:

- **Add Spacer Image** — добавление изображения-"распорки" к данной ячейке;
- **Remove All Spacer Images** — удаление всех изображений-"распорок" во всех ячейках данной таблицы разметки. Это может понадобиться, если

вы вручную помещаете в ячейку содержимое, полностью ее занимающее; таким образом, нужда в искусственной "распорке" отпадает. Например, мы со временем поместим в левую ячейку полосу навигации, и "распорка" станет ненужной;

Нужно ли добавлять изображения-"распорки" в ячейки вручную? На этот вопрос есть только один ответ. Посмотрите, что сделают с вашей таблицей разметки Netscape Navigator и Microsoft Internet Explorer, и решите сами. Старые версии Navigator славятся своим умением уродовать таблицы, так что уделите ему особое внимание. Если таблица выглядит хорошо и без "распорок", лучше без них обойтись. В противном случае, добавьте их. Однако имейте в виду: проверять в Web-обозревателе страницу нужно только после заполнения ее содержимым.

Сделайте центральную ячейку внешней таблицы разметки, где будет отображаться основное содержимое, "саморастягивающейся". Dreamweaver добавит в остальные ячейки изображения-"распорки".

Параметры таблицы разметки

Dreamweaver также позволяет нам настроить кое-какие параметры таблицы разметки. Выберите таблицу и обратите внимание на редактор свойств — см. рис. 8.26.

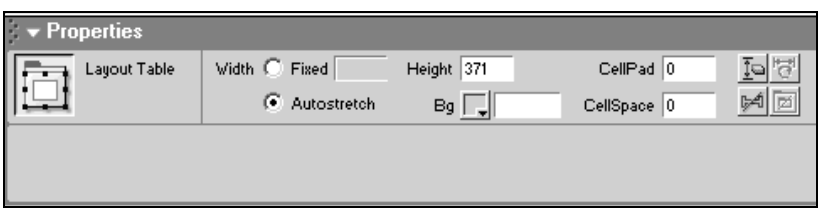


Рис. 8.26. Вид редактора свойств при выделенной таблице разметки

Здесь все вам уже знакомо. Группа переключателей **Width**, поля ввода **Width** и **Height** выполняют точно такие же функции, что их "коллеги", относящиеся к ячейкам разметки. Как видите, таблицы разметки тоже могут быть "саморастягивающимися" (по умолчанию) или иметь фиксированную ширину. Учтите, что таблицы с фиксированной шириной не могут иметь "саморастягивающиеся" ячейки.

Селектор цвета **Bg** задает цвет фона таблицы.

Поля ввода **CellPad** и **CellSpace** задают расстояния соответственно между границей ячейки и ее содержимым и между границами соседних ячеек. По умолчанию они равны нулю.

В правой части редактора свойств расположены четыре кнопки, позволяющие выполнить некоторые манипуляции с таблицей разметки.

1. Левая верхняя кнопка — **Clear Row Heights** — позволяет удалить атрибуты и величины высот строк таблицы `HEIGHT`. (Вы также можете выбрать пункт **Clear Cell Heights** меню строки ширин.) После этого каждая строка примет такую высоту, чтобы полностью вместить содержимое всех ячеек; таким образом, таблица разметки будет иметь в окне Web-обозревателя минимальную высоту. Однако это подходит не для всех страниц, а только для тех, что созданы на основе простейших схем табличных разметок. Если же ваша разметка включает в себя строки фиксированной высоты, лучше не трогайте эту кнопку.
2. Правая верхняя кнопка — **Make Widths Consistent** — задает такие значения ширины ячеек таблицы, чтобы они вместили все свое содержимое. (Вы также можете выбрать пункт **Make Cell Widths Consistent** меню строки ширин.) Скажем, если вы жестко задали величины ширины ячеек, но Dreamweaver был вынужден увеличить или уменьшить их из-за того, что содержимое не помещалось в ячейки, в строке ширин для этой ячейки будут отображаться два значения ширины: одно в кавычках, другое в скобках. Значение в кавычках обозначает реальную ширину ячейки, а значение в скобках — заданное вами. В этом случае вы можете скорректировать ширину нажатием этой кнопки.
3. Левая нижняя кнопка — **Remove All Spacers** — удаляет все изображения-"распорки" во всех ячейках таблицы. Вы также можете выбрать пункт **Remove All Spacer Images** меню строки ширин.
4. Правая нижняя кнопка — **Remove Nesting** — доступна только во вложенных таблицах. (Вы также можете выбрать пункт **Remove Nesting** меню строки ширин.) Она позволяет преобразовать вложенную таблицу в набор ячеек внешней таблицы; при этом содержимое ячеек не теряется. Это позволит уменьшить сложность таблицы разметки и, следовательно, HTML-кода, который ее формирует.

Заполнение начальной страницы

Теперь давайте заполним нашу новую начальную страницу содержимым. Для этого нам понадобится старая начальная страница, файлу которой мы дали имя default_old.htm. Откройте ее в новом окне документа Dreamweaver, если вы ее еще не открыли. Обратите внимание, что она тоже отобразится в режиме разметки.

Итак, сейчас мы перенесем в нашу новую страницу заголовок сайта, панель навигации, основное содержимое и сведения об авторских правах. Колонку новостей пока оставим незаполненной.

Переключитесь в окно, где открыта страница default_old.htm, выделите текст заголовка, скопируйте его в буфер обмена Windows, переключитесь

назад в окно с новой страницей default.htm и вставьте этот текст в верхнюю ячейку. Измените ее размеры, чтобы заголовок не занимал слишком много места. Задайте горизонтальное и вертикальное выравнивание содержимого ячейки по центру. Также можете изменить цвет фона.

Чтобы скопировать в новую страницу полосу навигации, временно переключитесь в обычный режим, щелкнув кнопку **Standard View** панели объектов. Иначе вы не сможете скопировать таблицу, содержащую полосу навигации: если включен режим разметки, Dreamweaver любую таблицу считает таблицей разметки и не позволяет ее скопировать. После того как вы скопируете полосу навигации, Dreamweaver выдаст ряд предупреждений, что он не может скопировать все связанные с ней сценарии. Вам нужно будет открыть диалоговое окно **Modify Navigation Bar**, выбрав пункт **Navigation Bar** меню **Modify**, и нажать в нем кнопку **OK**.

Сведения об авторских правах и основное содержание страницы вы перенесете в default.htm без нашей помощи. Единственное: вам придется очень часто изменять размеры и местоположение отдельных ячеек разметки. Dreamweaver делает это не всегда хорошо.

После этого выберите внешнюю таблицу разметки и задайте параметр **Cell-Pad** (соответствующее поле ввода в редакторе свойств) равным двум пикселам. Это добавит немного свободного пространства между содержимым соседних ячеек. Позднее мы добавим еще и линейки, разделяющие содержимое соседних ячеек таблицы.

У вас должно получиться нечто, похожее на рис. 8.27.

Теперь давайте заполним колонку новостей.

Обратим внимание на вложенную таблицу разметки. Она содержит три ячейки, в которых будут располагаться в порядке сверху вниз: заголовок, собственно новости и ссылка на страницу архива новостей. Каждая из этих ячеек будет иметь свой цвет фона, чтобы их можно было сразу отличить друг от друга. Текст — содержимое этих ячеек — также будет написан разными шрифтами. И — внимание! — текст будет выровнен вправо; таким образом, мы визуально отделим его от текста основного содержимого страницы.

Выберите верхнюю ячейку, задайте выравнивание посередине (вертикальное) и по правому краю (горизонтальное), цвет фона — темно-серый. Напишите строку "Новости", выделите ее и задайте жирный шрифт белого цвета и достаточно большого размера, чтобы выделить заголовок на фоне основного текста новостей. Все, с заголовком покончено.

Следующая на очереди — средняя ячейка, собственно новости. Придумайте что-нибудь и впишите туда. Отформатируйте новости, как хотите; единственное условие — текст должен быть выровнен вправо, как мы договаривались. И сделайте шрифт, которым набраны новости, поменьше, чтобы они

не так бросались в глаза. Также можете как-то выделить даты: жирным шрифтом или цветом.

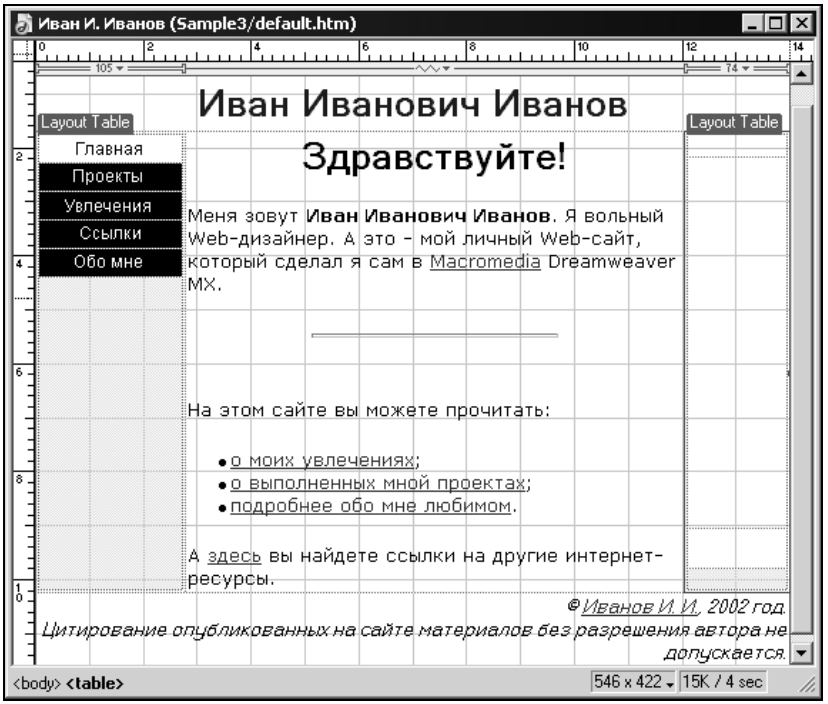


Рис. 8.27. Новая начальная страница сайта (колонка новостей пока не заполнена)

Последнюю ячейку — ссылку на архив новостей — сделайте самой узкой, только-только чтобы поместилась надпись "Архив". Задайте, как и у верхней ячейки, выравнивание посередине (вертикальное) и по правому краю (горизонтальное). Шрифт надписи пусть будет жирным курсивом.

Готовая колонка новостей показана на рис. 8.28. Нам осталось только сделать гиперссылку из надписи "Архив", но это уже совсем несложно. Эта гиперссылка должна ссылаться на страницу Archive.htm, которую мы создадим впоследствии. Запомните или запишите это имя.

Теперь посмотрим на нашу страничку еще раз. Что нам не нравится? Ага, полоса навигации находится слишком близко к основному содержанию. Надо бы поместить между ними свободное пространство, иначе говоря, просвет. Если бы мы создавали нашу разметку вручную, нам бы пришлось вставить в таблицу новую ячейку, для чего пришлось бы "перепахать" весь HTML-код. Но, поскольку мы работаем в Dreamweaver, все сильно упрощается.

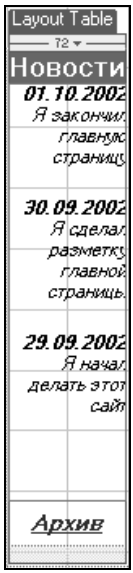


Рис. 8.28. Готовая колонка новостей

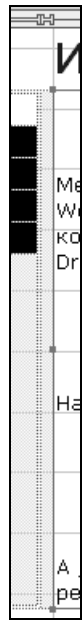


Рис. 8.29. Вертикальный просвет

Выберите ячейку основного содержания. "Ухватите" мышью за ее левый маркер размера и оттащите левую границу немного вправо. Таким образом, мы освободили немного свободного пространства между полосой навигации и основным содержанием. И все! Можете сохранить страницу и просмотреть ее в Web-обозревателе. Только проверьте, что цвет фона таблицы разметки — белый.

Как видите, Dreamweaver сам вставил ячейку в то место, где мы сделали свободное пространство. Более того, он поместил в нее "распорку", так что даже капризный Navigator не сможет уменьшить или увеличить ее ширину. Да посмотрите сами на рис. 8.29. Единственное: ячейка слишком узкая, так что Dreamweaver не может отобразить ее ширину в строке ширин. Чтобы увидеть значение ширины, вам нужно будет подвести к строке ширин курсор мыши.

Загрузите страницу в Web-обозревателе. Попробуйте изменить размеры окна и посмотрите на результат.

Теперь добавьте еще один вертикальный просвет между основным содержимым и колонкой новостей. Вы уже знаете, как это делать, и справитесь без подсказки.

Точно таким же образом вы можете создать горизонтальный просвет. Правда, для этого вам придется выполнить значительно больше работы. Вам придется сузить ячейки основного содержимого, полосы навигации и ко-

лонки новостей по вертикали, освобождая место между ними и ячейкой заголовка. Если же вы не можете сделать этого потому, что содержимое одной из ячеек заполняет ее целиком, вам придется растянуть по вертикали саму таблицу и переместить все эти ячейки ниже. Поэкспериментируйте с размерами ячеек и таблицы, чтобы получить удовлетворительный результат. Этот самый результат показан на рис. 8.30.

Рис. 8.30. Горизонтальный просвет

И напоследок удалите "распорку" из левой ячейки (полоса навигации), для чего выберите в меню строки ширин пункт **Remove Spacer Image**. Поскольку содержимое занимает ее целиком, дополнительные меры по фиксации ее ширины излишни.

Тонкая настройка и оптимизация таблицы разметки

Сколько уже раз мы повторяли, что за Dreamweaver, несмотря на его "интеллектуальность", нужен глаз да глаз! А если серьезно, то Dreamweaver, хоть и пытается изо всех сил создать то, что нам нужно, но это не всегда получается. Если уж люди не всегда понимают друг друга, то программы уж и подавно не понимают людей (да и другие программы тоже). И результаты его работы частенько приходится править вручную.

Но что там править? Сейчас увидим.

Активизируйте окно документа, где открыта наша новая начальная страница default.htm. Переключитесь в обычный режим, нажав кнопку **Standard View** панели объектов. Вы увидите нечто, похожее на рис. 8.31.

Что же нам может понадобиться исправить вручную? Довольно много...

Прежде всего, это размеры и местоположения ячеек. В режиме разметки очень трудно позиционировать одну ячейку относительно другой и правильно выставить ее размеры. Почему-то Dreamweaver так и норовит поместить между ячейками лишнее свободное пространство; а при попытке изменить ширину ячейки другие "расползаются" в разные стороны. Возиться с ними можно до бесконечности и так и не получить нужного результата. Поэтому зачастую проще будет подогнать их размеры и местоположение, удалить лишние ячейки и строки, переключившись в обычный режим.

Также вы можете захотеть изменить вручную значения параметров таблиц, строк и ячеек, которые нельзя установить в режиме разметки. Это цвет рамки таблицы и ячейки, ее видимость или невидимость. Dreamweaver делает

эти рамки невидимыми, что оправдано для таблиц и ячеек разметки. Но у вас могут быть другие планы.

Рис. 8.31. Готовая главная страница в обычном режиме отображения

Ну и, наконец, вы можете выполнить точную подгонку самой таблицы. Например, ради компактности HTML-кода можно объединить некоторые ячейки, образующие пустую строку (просвет). Конечно, в режиме разметки Dreamweaver почти всегда генерирует оптимальную результирующую таблицу. Почти, но не всегда.

Теперь вернемся к нашей странице. Внимательно осмотрите ее на предмет всевозможных огрехов. Автор нашел таковых целых три, хоть Dreamweaver и старался ему угодить.

1. Вложенная таблица колонки новостей занимает отведенную ей ячейку внешней таблицы не целиком.
2. Строку, образующую горизонтальный просвет между заголовком и нижними ячейками, можно объединить в одну ячейку. HTML-код станет несколько компактнее.
3. Dreamweaver влепил-таки в низ таблицы разметки лишнюю строку!

После того как мы выявили все огрехи, можно (и нужно) их исправить. Подстройте высоту вложенной таблицы, объедините ячейки строки просве-

та в одну и удалите лишние строки (если они есть). Переключитесь в режим отображения HTML-кода и посмотрите, все ли там гладко, — скорее всего, Dreamweaver создал еще несколько ненужных строк и ячеек; удалите их. Результат показан на рис. 8.32.

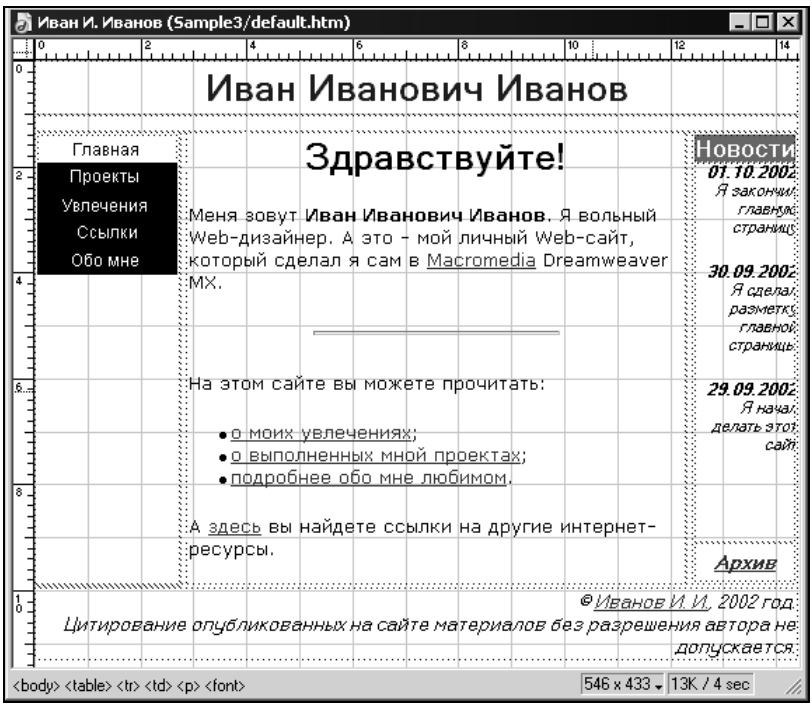


Рис. 8.32. Готовая страница после внесения всех исправлений

Внимание!

При правке таблиц разметки в обычном режиме специальные атрибуты, добавляемые к тегам `<TABLE>` Dreamweaver, могут потеряться. Это значит, что в дальнейшем с такими таблицами возможны проблемы, если вы снова захотите редактировать их в режиме разметки страниц.

Как видите, даже "интеллектуальный" Dreamweaver припасет для вас немного ручной работы. Правда, это, в основном, тонкая оптимизация, "вылизывание", без которой, в крайнем случае, можно обойтись. Но лучше не обходиться, ведь маленькие "шероховатости" HTML-кода могут вырасти в дальнейшем в большие проблемы.

А для того, чтобы устранить эти "шероховатости", лучше переключиться в обычный режим. На наш взгляд, режим разметки подходит только для первоначальной и довольно грубой разработки дизайна. Поскольку вы уже

работали в режиме разметки, то должны были заметить, как "норовист" Dreamweaver. Всю окончательную его доводку лучше производить в обычном режиме отображения страницы, когда он ведет себя более предсказуемо.

Ну вот, мы устранили последние огрехи. Казалось бы, теперь уж все действительно готово. Можно приступать к созданию остальных страниц сайта, взяв за основу главную страницу. Как это сделать?

1. Сохраните готовую главную страницу. Если вы хотите сделать небольшую передышку, то самое время.
2. Откройте главную страницу, если уже ее закрыли.
3. Сохраните ее под другим именем, например Projects.htm. Для этого выберите пункт **Save As** в меню **File** или нажмите комбинацию клавиш <Ctrl>+<Shift>+<S>.
4. Удалите все то, что не должно присутствовать на данной странице (в нашем примере это начальное содержимое — мы подставим вместо него список проектов). Создайте содержимое страницы проектов.
5. Закройте готовую страницу. Не забудьте сохранить ее при этом.
6. Повторите пункты 2—5 этого списка для всех страниц вашего сайта.

Неплохо придумано! Мы заимствуем общие элементы с другой страницы и создаем заново только ее содержимое. Это может сэкономить немало времени и убережет нас от множества ошибок, которые мы неизбежно сделаем, если начнем создавать каждую страницу сайта "с нуля". Это один из способов облегчить себе работу, за которые мы должны благодарить ленивых Web-дизайнеров.

Недостатки табличного дизайна

Осталось разбавить дегтем оставшийся мед.

Здесь мы поговорим о недостатках табличного дизайна. В основном, они происходят от недостатков самих таблиц; а недостатки таблиц мы перечислили в *главе 5*. И все-таки перечислим их еще раз. Но, кроме них, есть и недостатки самого способа построения Web-страниц с использованием таблиц, и мы их также рассмотрим.

Начнем с таблиц как основы табличного дизайна.

Недостатки таблиц

Как уже говорилось, недостатки таблиц были описаны в *главе 5*. Если вы их забыли, давайте вспомним: очень медленная загрузка и отображение и громоздкий, запутанный HTML-код. Рассмотрим их еще раз и уже в приложении к табличному дизайну. (О специфических недостатках, связанных с осо-

бенностями обработки HTML-кода отдельными программами Web-обозревателей, мы говорить не будем. Вы можете прочитать о них все в той же *главе 5*.)

Очень медленная загрузка — недостаток номер один для Web-страниц, созданных на основе таблиц разметки. В самом деле, если какая-то часть страницы долго не отображается, посетитель сайта может подождать. Но если заставляет себя ждать вся страница, это действует на нервы. Это справедливо для всех страниц, созданных на основе таблиц, и если страница достаточно сложна, а соединение с Интернетом медленное, посетитель может даже уйти с сайта, не дождавшись загрузки страницы. Это действительно серьезная проблема, и ее решению следует уделить внимание.

Какие же пути решения этой проблемы существуют?

В основном, это, конечно же, грамотная разработка и тщательная оптимизация таблицы разметки. Таблица разметки должна быть как можно более простой, по возможности не содержать ячеек и строк с относительными значениями ширины и высоты и минимальным количеством вложенных таблиц. Все эти рекомендации описаны в *главе 5*, и автор настоятельно вам советует прочитать их еще раз.

Насчет простоты. Когда таблица занимает всю страницу и вмещает в себя множество разнородных данных, которые должны быть расположены в разных ее ячейках, говорить о простоте неуместно. Таблица в любом случае получится очень сложной. Слегка упростить вы ее сможете за счет тонкой оптимизации HTML-кода, но именно слегка. Это один из недостатков табличного дизайна, который нельзя преодолеть.

Насчет относительных значений ширины и высоты. Таблица разметки, как правило, занимает всю ширину Web-страницы, а значит, без относительного значения этой самой ширины не обойтись. Конечно, иногда таблицы разметки имеют фиксированную ширину, но в этом случае используется не вся полезная площадь окна Web-обозревателя, и могут возникнуть проблемы, если ширина окна меньше ширины таблицы. То же самое можно сказать об относительных значениях ширины и высоты ячеек таблиц: одна из них, как правило, с основным содержимым страницы, должна использовать оставшееся от ее "соседей" свободное пространство по максимуму. А значит, без относительных значений размеров опять же не обойтись.

Совет включать в таблицу минимум данных в случае табличного дизайна, конечно, неприменим. В самом деле, если в таблице помещается все содержимое Web-страницы, и содержимое немаленькое, вы ничего не сможете сделать. Однако здесь можно дать другой совет. Даже два.

Во-первых, по возможности, разбивайте одну большую таблицу на несколько меньших. Вы можете, например, отделить от главной, большой таблицы заголовок сайта и поместить его в отдельной таблице. Также вы можете поступить со сведениями об авторских правах. При этом страница будет за-

гружаться визуально быстрее, и посетитель, видя, что "процесс идет", возможно, дождется его окончания.

Взгляните на рис. 8.33. Здесь дизайн страницы построен на основе не одной большой, а трех небольших таблиц разметки, загружающихся и отображающихся последовательно, одна за другой. Таблицы расположены "впритык" друг к другу. Посетитель сайта и не заметит, что ваша страница на самом деле состоит из отдельных частей. Надо только точно подогнать эти части, чтобы между ними не оставалось пустых мест.

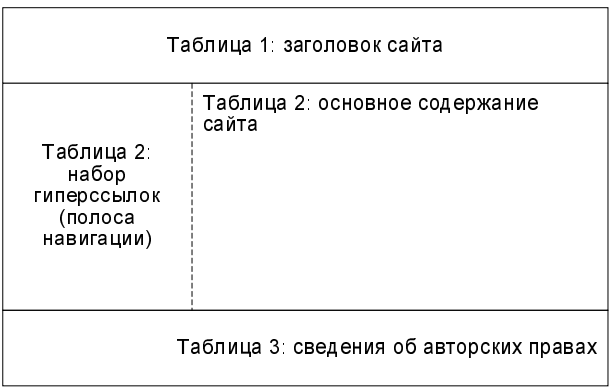


Рис. 8.33. Многотабличный дизайн начальной страницы

И так часто делают! Берите пример!

Во-вторых, некоторые части страницы вы можете вынести вообще за пределы таблицы разметки. Чаще всего так поступают со сведениями об авторских правах, реже — с заголовком. Пример такого подхода показан на рис. 8.34.

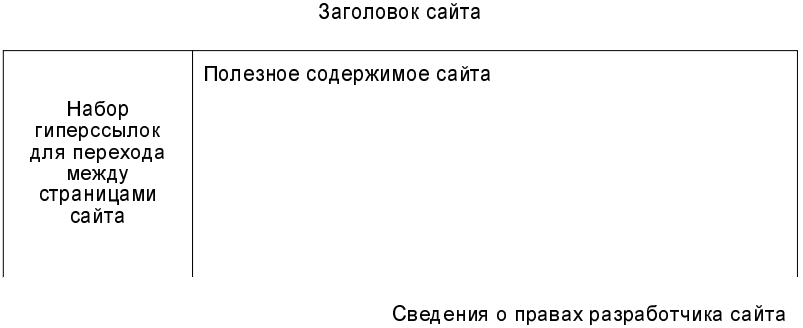


Рис. 8.34. Дизайн главной страницы с заголовком и сведениями об авторских правах, вынесенными за пределы таблицы разметки

Давайте поступим так с нашим сайтом: вынесем за пределы таблицы заголовок сайта и сведения об авторских правах. Конечно, поскольку страницы нашего сайта очень малы, это не даст большого прироста скорости загрузки, но все-таки сделаем это.

Откройте файл default.htm и переключитесь в обычный режим отображения страницы. В этом режиме будет проще выполнить нужные манипуляции.

Сначала нам необходимо вставить перед таблицей разметки пустой абзац. Поместите курсор мыши левее таблицы так, чтобы он принял вид текстового курсора, и щелкните левой кнопкой мыши. Слева от таблицы появится большой мигающий текстовый курсор. Нажмите клавишу <Enter>, и Dreamweaver поместит перед таблицей пустой абзац.

Теперь выделите текст заголовка и перетащите его в этот пустой абзац. Отформатируйте его, чтобы он походил на заголовок сайта. После этого удалите верхнюю строку таблицы, где раньше был заголовок, и следующую за ней узкую строку — горизонтальный просвет.

Далее поместите курсор мыши правее таблицы так, чтобы он принял вид текстового курсора, и щелкните левой кнопкой мыши. Большой мигающий текстовый курсор появится теперь справа от таблицы. Нажмите клавишу <Enter>, и Dreamweaver поместит после таблицы пустой абзац. Переместите туда текст сведений об авторских правах. Останется только уменьшить высоту ячейки колонки новостей, сделав ее равной высоте ячейки основного содержимого, и удалить ненужную строку внизу таблицы. Возможно также, что вам придется заново выставить значения ширины ячеек.

Сохраните теперь получившуюся страницу. У вас должно получиться нечто, похожее на рис. 8.35.

Вы заметили, как быстро мы отредактировали нашу страницу? В режиме разметки нам пришлось бы до бесконечности перемещать с места на место ячейки таблицы, подгоняя их размеры и местоположение, устраняя постоянно появляющиеся просветы и лишние ячейки. В обычном режиме все это заняло минуту.

И все же создавать "с нуля" таблицу разметки лучше всего в режиме разметки...

Второй недостаток таблиц — громоздкий HTML-код — к сожалению, непреодолим. Можно только посоветовать, опять же, не создавать слишком уж заковыристых таблиц, а если и создавать, то с помощью Dreamweaver. Он, в отличие от дизайнера-человека, не делает ошибок. Ну, почти не делает...

И еще один совет. Если вам нужно опубликовать на сайте большой текст, а сайт сделан на основе табличного дизайна, поступите следующим образом. Вместо того чтобы помещать весь текст на одну страницу, создавая настоящего монстра, разбейте его на несколько частей и поместите на не-

скольких страницах. При этом в конце каждой страницы, содержащей часть такого текста, поместите ссылку на страницу со следующей частью.

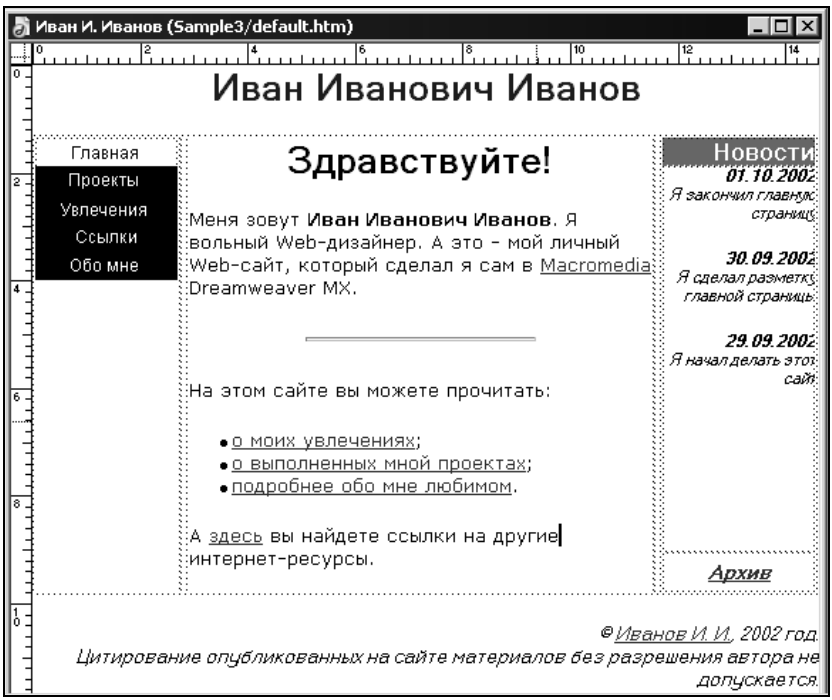


Рис. 8.35. Главная страница с заголовком и сведениями об авторских правах, вынесенными за пределы таблицы разметки

Недостатки табличного дизайна

Всем хорош табличный дизайн. Мы можем точно контролировать, как один фрагмент содержимого страницы расположен относительно другого, а Dreamweaver максимально облегчает нам эту задачу. Мы можем делать наши Web-страницы максимально совместимыми с разными программами Web-обозревателей, что не всегда проходит в случае с фреймами. Мы можем многое. И нам не хочется верить, что у табличного дизайна есть недостатки.

Недостатков, пожалуй, всего два. Но каких!!!

Первый недостаток: большое количество повторяющихся элементов, присутствующих на всех страницах сайта. Посмотрите сами — на каждой странице нашего нового сайта Sample site 3 есть совершенно одинаковые заголовок, полоса навигации и сведения об авторских правах. Именно из-за этого файлы Web-страниц, созданных на основе табличного дизайна, так

велики. И когда посетитель щелкает по гиперссылке, все это хозяйство каждый раз загружается заново.

Когда мы строили сайт на основе фреймов, мы отделили эти повторяющиеся элементы от уникального основного содержимого и поместили их на других страницах, отображаемых в других фреймах. Когда посетитель щелкает по какой-либо гиперссылке, то заново загружается только страница с основным содержимым. Повторяющиеся же элементы как были, так и остаются неизменными в своих фреймах.

Конечно, мы можем уменьшить объем HTML-кода уменьшением объема содержимого страницы, "выносом" некоторых его фрагментов за пределы таблицы (это все мы рассмотрели выше), тщательной оптимизацией, наконец. Но радикального уменьшения объема кода это не даст; страница так и останется достаточно большой. А все из-за этих повторяющихся элементов, от которых никуда не деться.

И здесь мы подходим ко второму недостатку табличного дизайна: большой трудоемкости внесения изменений в повторяющиеся элементы. В самом деле, если вы вдруг захотите добавить в полосу навигации новую гиперссылку, вам придется делать это во всех страницах вашего сайта. Хорошо, если их всего пять. А если пятьдесят?

Первый недостаток табличного дизайна непреодолим. Можно только посоветовать не создавать слишком больших и сложных повторяющихся элементов и тщательно оптимизировать HTML-код. И — смириться с неизбежным...

А вот второй недостаток вполне преодолим. И преодолеть его нам поможет все тот же великолепный Dreamweaver. Для этого он предоставляет замечательное средство, называемое шаблонами и описываемое в *главе 9*.

Что дальше?

Вот мы и познакомились с таблицами разметки и табличным дизайном. Мы создали третий по счету сайт, основанный на таблицах. Наконец, мы научились работать в режиме разметки страницы.

Собственно, вы уже поняли, какая тема будет следующей. Это шаблоны. И вы уже знаете, для решения каких проблем они созданы. Так не будем же терять времени! За дело!

ГЛАВА 9

Использование шаблонов

Вот мы и закончили наш третий Web-сайт. Все работает, все страницы нормально загружаются и отображаются в Web-обозревателе. Казалось бы, нет поводов для беспокойства.

Нет, но только до поры до времени.

Представим себе такую ситуацию. Мы вдруг решили добавить в наш сайт новую страницу. Чтобы это сделать, нам нужно изменить полосу навигации во всех страницах, а для этого придется открыть каждую из них в окне документа, добавить ссылку и сохранить страницу. Наш сайт невелик по размеру, и мы выполним эту работу довольно быстро. А если бы он был велик?

Конечно, мы можем воспользоваться встроенными средствами Dreamweaver. Например, вызвать диалоговое окно **Find and Replace** — средство исключительно мощное. (О возможностях по поиску и замене подстрок *см. в главе 2.*) Или запустить замену гиперссылок, вызвав пункт **Change Link Sitewide** меню **Site** панели **Site** *(см. главу 6)*. Во многих случаях это поможет нам быстро заменить целые фрагменты содержимого страницы или ее HTML-кода. Большую помощь нам могут дать активы и библиотека элементов, также описанные в *главе 6.*

Но что делать, если мы хотим произвести какие-нибудь значительные изменения, например, радикально поменять структуру таблицы разметки? Поиск и замена нам вряд ли помогут, а уж автоматическая замена гиперссылок — тем более. Даже активы с библиотекой нам в этом случае не помогут. Неужели придется переделывать заново все страницы?!

Если бы вы работали не в среде Dreamweaver, вероятно, так бы и пришлось делать. Но нам повезло. Dreamweaver поддерживает такое мощное средство, как шаблоны. Они уже упоминались в *главе 3.* Сейчас мы их рассмотрим подробнее.

Введение в шаблоны Dreamweaver

Шаблон — это своеобразный образец, "скелет" Web-страницы, содержащий общие для всех страниц элементы. Когда вы создаете новую страницу на основе шаблона, вам остается только вписать в нужные места *уникальное содержимое* этой самой страницы и сохранить ее. Впоследствии вы можете изменить шаблон — и Dreamweaver сам обновит все созданные на его основе страницы.

В этом смысле шаблоны схожи с элементами библиотеки *(см. главу 6)*. Отличие в том, что шаблон — это заготовка для целой страницы, а библиотека хранит только ее отдельные элементы. Однако разработчики Dreamweaver считают, что шаблоны ближе всего по "родству" активам *(см. главу 6)*, и поместили список шаблонов в панели **Assets**.

Изначально шаблон изменить нельзя, т. е. когда вы создаете на его основе страницу, вы можете помещать содержимое только в специально отведенных для этого местах — *изменяемых областях*. Элементы самого шаблона вы редактировать не можете, т. к. они являются *неизменяемыми областями*. Если же вы хотите что-то исправить, то вам придется открыть в окне документа сам шаблон. Таким образом, Dreamweaver спасает вас от ошибочного изменения шаблона и, соответственно, от искажения созданных на его основе Web-страниц.

Можно сказать, что шаблоны — это обычные Web-страницы. При редактировании вы можете обращаться с шаблонами, как с обычными Web-страницами, и использовать те же инструменты. Также вы можете задавать параметры страницы, которая будет создана на основе этого шаблона (название, цвет фона, текста и гиперссылок). Однако при создании HTML-кода шаблонов Dreamweaver использует множество собственных тегов и атрибутов, поэтому говорить о том, что шаблон — обычная Web-страница, не совсем корректно.

Шаблоны сохраняются в файлах с расширением dwt в папке Templates, находящейся в корневой папке локальной копии сайта. Из этого следует, что шаблоны — неотъемлемая часть вашего сайта, как и библиотека. Чтобы использовать какой-либо шаблон в другом сайте, вы будете должны скопировать его в тот сайт, использовав стандартные средства Dreamweaver *(см. главу 6)*. В одном сайте могут применяться несколько шаблонов.

Шаблоны более всего пригодятся вам, если вы создаете страницы на основе табличного дизайна. Такие страницы практически всегда содержат множество повторяющихся элементов, обновлять которые вручную крайне трудоемко. Но вы можете заготовить шаблоны и для "обычных" страниц. Это может понадобиться, например, если вы создаете страницы с одинаковым оформлением для какого-то большого фирменного сайта.

Вместе с Dreamweaver поставляется довольно много шаблонов, созданных профессиональными Web-дизайнерами. Вы можете использовать эти шаб-

лоны для создания своих страниц; как это делается, было рассмотрено в *главе 3*. Попробуйте — возможно, вы найдете что-нибудь, для вас подходящее.

Работа с шаблонами

С теорией мы закончили. Пора переходить к практике.

Создание шаблона

Есть два способа создать шаблон Dreamweaver. Во-первых, его можно создать "с нуля", а потом заполнить содержимым, как обычную Web-страницу. Во-вторых, существующую страницу можно сохранить как шаблон, а потом отредактировать, удалив полезное содержимое и оставив только общие элементы. И то, и другое сделать одинаково легко.

Прежде всего, активизируйте панель **Assets** и переключитесь на список шаблонов, щелкнув вторую снизу кнопку в списке категорий. Список шаблонов ничем не отличается от списков других видов активов, знакомых вам по *главе 6*. Поэтому мы не будем приводить его рисунок.

Чтобы создать новый шаблон "с нуля", либо нажмите кнопку **New Template** в правом нижнем углу панели **Assets** (рис. 9.1), либо выберите пункт **New Template** в контекстном или дополнительном меню этой панели. В списке появится новый пункт, вместо названия которого будет находиться поле ввода с текстом `Untitled`. Введите в него имя вновь создаваемого шаблона и нажмите клавишу <Enter>. Имейте при этом в виду, что имя шаблона есть на самом деле имя файла, в котором сохраняется шаблон, поэтому оно должно удовлетворять соглашениям по именованию файлов. (Автор, например, назвал новый шаблон Sample.) Все, вы только что создали новый пустой шаблон.

Рис. 9.1. Кнопка **New Template**

Поскольку ваш шаблон в данный момент ничего не содержит, Dreamweaver предупредит вас об этом, выведя соответствующий текст на панели предварительного просмотра (рис. 9.2). Вам нужно будет поместить в этот шаблон содержимое. Но об этом будет рассказано позже.

Создать новый шаблон на основе существующей Web-страницы еще проще. Для этого откройте нужную Web-страницу, например default.htm, и выберите в меню **File** пункт **Save as Template**. Также вы можете нажать кнопку **Make Template** (рис. 9.3) вкладки **Templates** панели объектов. На экране появится диалоговое окно **Save As Template**, показанное на рис. 9.4.

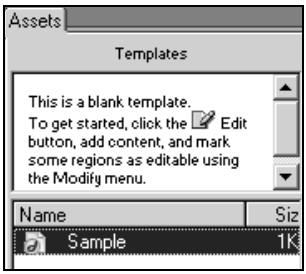


Рис. 9.2. Новый пустой шаблон в списке шаблонов

Рис. 9.3. Кнопка **Make Template** панели объектов

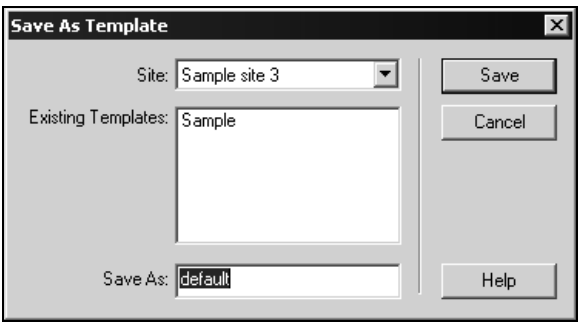


Рис. 9.4. Диалоговое окно **Save As Template**

В раскрывающемся списке **Site** выбирается сайт, в котором сохраняется шаблон. (Шаблоны являются неотъемлемой собственностью сайта, помните?) По умолчанию там выбран текущий сайт.

В списке **Existing Templates** перечислены уже сохраненные в этом сайте шаблоны. В нашем случае там есть только один шаблон — созданный нами ранее пустой шаблон Sample. Вы можете выбрать в этом списке любой шаблон и перезаписать его; в этом случае Dreamweaver переспросит вас, действительно ли вы хотите перезаписать существующий шаблон.

Само имя шаблона вводится в поле ввода **Save As**. Давайте назовем наш новый шаблон Main ("главный"), поскольку это наш главный шаблон, на основе которого мы построим наш сайт.

Введя все нужные данные, нажмите кнопку **Save** для сохранения шаблона или **Cancel** — для отказа от этого. Поскольку мы нуждаемся в шаблоне, нам нужно нажать кнопку **Save**.

После того как мы создадим новый шаблон, последний появится в списке шаблонов. А поскольку мы создали этот новый шаблон на основе существующей Web-страницы, т. е. он имеет содержимое, мы можем просмотреть его на панели предварительного просмотра панели **Assets**. Правда, эта панель очень мала, и, чтобы получить представление о содержимом шаблона, нам придется открыть его в окне документа.

Редактирование шаблона

Пустой шаблон необходимо наполнить содержимым. Шаблон, созданный на основе Web-страницы, нужно отредактировать: убрать уникальное для этой страницы содержимое, оставив только общие для всех страниц сайта элементы. Для этого вам надо будет открыть шаблон для редактирования в окне документа, как обычную Web-страницу.

Сделать это можно тремя способами:

- щелкнуть по нужной позиции списка шаблонов правой кнопкой мыши и выбрать пункт **Edit** контекстного или дополнительного меню;
- выбрать нужный шаблон в списке и нажать кнопку **Edit** в правом нижнем углу панели **Assets** (рис. 9.5);
- дважды щелкнуть по нужному шаблону в списке.

Рис. 9.5. Кнопка **Edit** панели **Assets**

В любом случае на экране появится окно документа, в котором будет открыт выбранный нами шаблон (рис. 9.6). Как видите, сейчас он ничем отличается от Web-страницы default.htm, на основе которой был создан.

Что вы можете сделать с шаблоном? Все что угодно. Можете считать, что это обычная Web-страница с некоторыми особенностями. (Эти особенности будут описаны ниже.) Вы можете набирать текст, форматировать его, размещать изображения, фильмы, таблицы, гиперссылки, переключаться в режим разметки и создавать таблицы и ячейки разметки, создавать наборы фреймов, пользоваться активами и библиотекой, "чистить" HTML-код и т. д. В общем, делать все то, что вы уже знаете.

Но все-таки шаблон — не Web-страница и имеет некоторые особенности. Вспомните, ведь, помимо всего прочего, вам необходимо разместить на нем изменяемые области, в которых впоследствии будет размещено содержимое страниц. Более того, вам обязательно нужно это сделать, иначе шаблон вам, фактически, так и не пригодится. Сейчас вы узнаете, как это сделать.

Откройте шаблон Main, который мы создали на основе страницы default.htm (если вы его еще не открыли). Уберите текст основного содержимого и

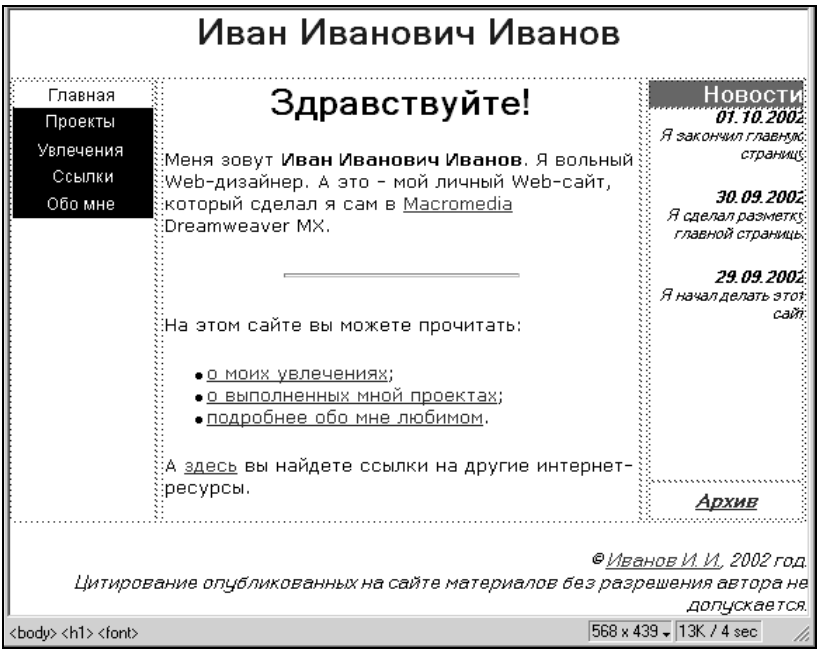


Рис. 9.6. Открытый в окне документа шаблон Main

Рис. 9.7. Отредактированный шаблон Main (включен режим разметки)

колонку новостей, оставьте только заголовок, полосу навигации и сведения об авторских правах. Проще всего это сделать, переключившись в режим разметки. Проверьте, является ли ячейка основного содержимого "саморастягивающейся". В результате у вас должно получиться нечто, похожее на рис. 9.7.

Сохраните шаблон. При этом Dreamweaver предупредит вас, что шаблон не содержит ни одной изменяемой области, выведя небольшое окно-предупреждение. Нажмите кнопку **OK**, чтобы все-таки сохранить шаблон, или **Cancel** — для отказа от этого. Если вы не хотите, чтобы это окно появлялось в дальнейшем, перед тем, как нажать кнопки **OK** или **Cancel**, включите флажок **Don't warn me again**.

Теперь нам нужно разместить на шаблоне изменяемые области. В нашем случае изменяемая область будет всего одна — основное содержимое страницы. И располагаться она будет в самой большой ячейке нашей таблицы разметки.

Создание изменяемых областей

Как и в случае с самим шаблоном, вы можете создать изменяемые области двумя способами. Во-первых, вы можете создать пустую изменяемую область на пустом месте страницы. Во-вторых, вы можете преобразовать фрагмент текущего содержания страницы в изменяемую область. Первый способ лучше всего подходит, если создали шаблон "с нуля", а второй — если вы преобразовали в шаблон существующую страницу.

Поместить в шаблон пустую изменяемую область очень просто. Для этого сначала поставьте текстовый курсор в то место, где вы хотите создать изменяемую область. Если же вы хотите преобразовать в изменяемую область фрагмент содержимого страницы, просто выделите этот фрагмент (в нашем случае — основное содержимое страницы).

После этого выберите в подменю **Template Objects** меню **Modify** окна документа пункт **New Editable Region** или нажмите комбинацию клавиш <Ctrl>+ +<Alt>+<V>. Вы можете также выбрать пункт **New Editable Region** подменю **Template** контекстного меню. Если вы предпочитаете пользоваться панелью объектов, то можете нажать кнопку **Editable Region** на вкладке **Templates** (рис. 9.8). На экране появится диалоговое окно **New Editable Region**, показанное на рис. 9.9.

Рис. 9.8. Кнопка **Editable Region** панели объектов

В единственном поле ввода **Name**, находящемся в этом окне, вводится уникальное имя вновь создаваемой изменяемой области. Каждая созданная ва-

ми в шаблоне изменяемая область должна иметь уникальное имя, иначе Dreamweaver не сможет ее обработать. Это имя может содержать любые символы, кроме букв русского алфавита, кавычек, апострофа и знаков "<", ">" и "&". После этого нажмите кнопку **OK**, чтобы создать изменяемую область, или **Cancel** — для отказа от этого.

Внимание!

Вы не можете дать изменяемой области имя `doctitle`. Почему, будет рассказано позже.

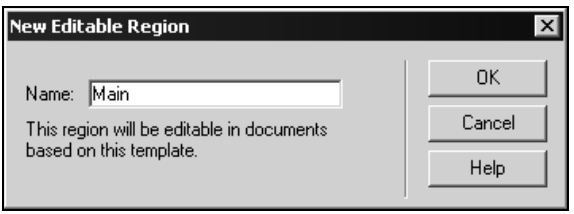


Рис. 9.9. Диалоговое окно **New Editable Region**

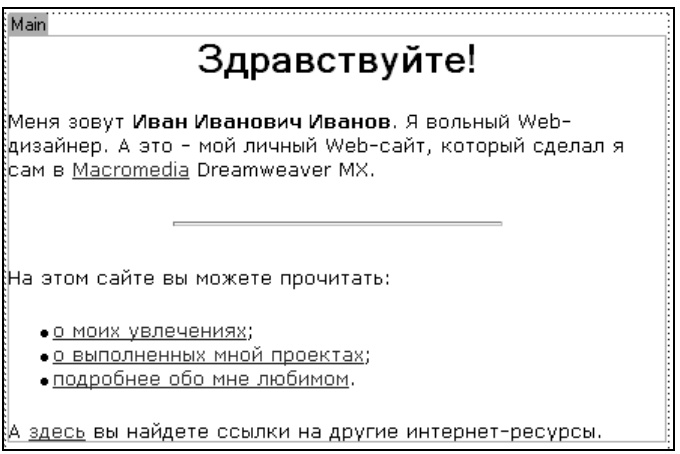


Рис. 9.10. Вновь созданная изменяемая область

Когда вы нажмете кнопку **OK**, вы увидите следующее — см. рис. 9.10. Так выглядит в окне документа Dreamweaver изменяемая область, в которую мы превратили основное содержимое нашей страницы. Вверху вы видите небольшой голубой ярлычок — *заголовок*, содержащий имя изменяемой области; он похож на заголовок таблиц разметки. Щелкнув его, вы выделите изменяемую область. Ниже, в синей рамке, находится содержимое изменяемой области, в нашем случае — текст основного содержимого. (Если вы создали

пустую изменяемую область, она будет содержать только текст ее имени.) При выборе изменяемой области содержимое области будет выделено.

Чтобы поместить в изменяемую область новый текст, сначала выделите все то, что содержится в синей рамке, но не удаляйте, иначе будет удалена сама изменяемая область. После этого вы можете набирать там другой текст — он заменит собой старое содержимое изменяемой области. Также вы можете разместить внутри изменяемой области все, что предлагают вам HTML и Dreamweaver: таблицы, гиперссылки, изображения и т. п. При вводе текста или иного содержимого в изменяемую область окружающая его синяя рамка будет растягиваться, чтобы вместить его полностью.

Запомните, что при редактировании шаблона вы можете изменять все, что угодно. В том числе, вы можете поместить в редактируемые области любое содержимое, а потом изменить его в создаваемых на базе данного шаблона Web-страницах. Но когда вы создаете сами страницы, то изменять можете только содержимое изменяемых областей.

Чтобы выбрать нужную изменяемую область, вы можете щелкнуть по ее заголовку. Но если шаблон достаточно велик и полностью не помещается в окне документа, вам придется постоянно прокручивать его содержимое в поисках нужной изменяемой области. Для такого случая Dreamweaver припас вам приятный сюрприз. Вызовите на экран контекстное меню, откройте подменю **Templates**, а в нем — подменю **Editable Regions**. В нем будут перечислены все созданные в этом шаблоне изменяемые области. Вам останется только выбрать нужный пункт этого подменю, и Dreamweaver покажет вам соответствующую изменяемую область.

Кроме подменю **Editable Regions**, список всех созданных в шаблоне изменяемых областей находится также внизу подменю **Templates** меню **Modify**.

Если вы поместили изменяемую область не в то место, куда хотели, это можно исправить. Выделите нужную изменяемую область, щелкнув мышью по ее заголовку. После этого "ухватите" ее мышью за содержимое и перетащите на нужное место. Вот и все.

Однако, если вы ошиблись в наборе имени, вам придется править HTML-код шаблона. Как уже говорилось, чтобы отделить неизменяемое содержимое шаблона от содержимого изменяемой области, Dreamweaver вставляет в HTML-код специальные комментарии. Они имеют такой вид:

```
<!-- TemplateBeginEditable name="Main" -->
{ Содержимое изменяемой области }
<!-- TemplateEndEditable -->
```

Как видите, здесь всего два комментария: первый исполняет функцию открывающего тега, второй — закрывающего. Имя изменяемой области находится в первом комментарии; оно выделено полужирным шрифтом. Измените его на нужное.

Внимание!

После того, как на основе шаблона были созданы какие-либо Web-страницы, вы не сможете переименовать ни одну из имеющихся в нем изменяемых областей.

И, наконец, может случиться так, что вы захотите удалить изменяемую область. Сделать это можно двумя способами: простым и очень простым. Очень простой способ заключается в том, чтобы выбрать изменяемую область щелчком по заголовку и нажать клавишу <Del>. Простой способ — поставить текстовый курсор куда-либо внутрь содержимого изменяемой области и выбрать пункт **Remove Editable Markup** подменю **Templates** меню **Modify** или контекстного меню. Учтите, что после удаления изменяемой области ее содержимое остается в шаблоне. Так что, если вы хотите удалить всю изменяемую область, вам также придется удалить ее содержимое.

А теперь — небольшой сюрприз. Дело в том, что Dreamweaver при создании любого шаблона автоматически создает небольшую изменяемую область `doctitle`. (Вот поэтому вы и не можете дать изменяемой области имя `doctitle` — такая изменяемая область уже существует!) Эта изменяемая область включает в себя содержимое тега `<TITLE>`, иначе говоря, название Web-страницы. Это *невидимая изменяемая область* — она не отображается ни в подменю **Editable Regions** подменю **Templates** контекстного меню, ни внизу подменю **Templates** меню **Modify**.

Внимание!

Если вы удалите скрытую изменяемую область `doctitle`, то потеряете возможность менять названия Web-страниц, основанных на этом шаблоне. В данном случае все эти страницы будут иметь одно название, совпадающее с названием шаблона.

Создание Web-страниц на основе шаблонов

Создать Web-страницу на основе шаблона можно тремя способами.

Первый способ заключается в том, чтобы использовать уже знакомый вам пункт **New** меню **File** окна документа Dreamweaver. При этом на экране появится диалоговое окно **New Document**, показанное на рис. 3.1. Переключитесь на вкладку **Templates** — и вы увидите то, что показано на рис. 9.11.

В списке **Templates For** выбирается сайт, из которого будет взят шаблон. Это значит, что вы можете создать страницу на основе шаблона, который принадлежит другому сайту.

В списке **Site *<имя сайта>*** выбирается нужный шаблон. Выберите шаблон Main. После этого в расположенной справа панели предварительного просмотра появится изображение выбранного шаблона.

Если флажок **Update Pages when Template Changes** включен (а он включен по умолчанию), при изменении шаблона, на основе которого создается Web-страница, она будет соответственно изменена. Если же этот флажок отключить, то в новую страницу будет просто скопировано содержимое выбранного шаблона. В этом случае она не будет содержать ни изменяемых, ни неизменяемых областей и при изменении шаблона изменяться не будет. Фактически это будет независимая страница, просто содержащая все содержимое шаблона.

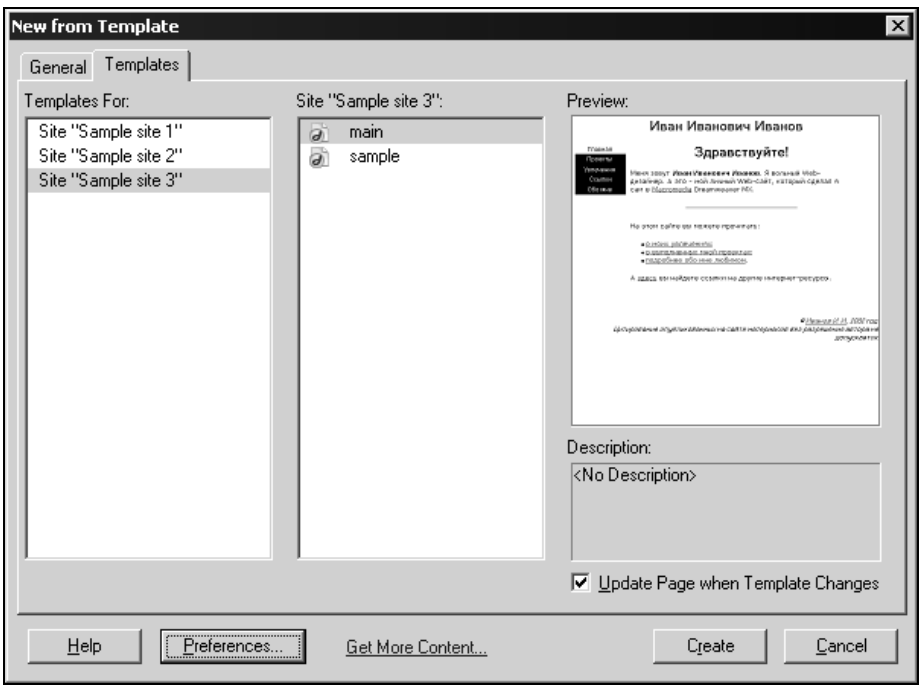


Рис. 9.11. Диалоговое окно **New Document** (вкладка **Templates**)

Если вам нужно создать страницу, чье содержимое будет значительно отличаться от шаблона, отключите этот флажок. После этого вы сможете изменять ее как угодно.

Чтобы создать новую страницу на основе выбранного шаблона, нажмите кнопку **Create**. Чтобы отказаться от создания страницы, нажмите кнопку **Cancel**.

Второй способ создания новой страницы на основе шаблона намного проще. Откройте панель **Assets**, переключитесь на список шаблонов, выберите в списке нужный шаблон и в контекстном или дополнительном меню — пункт **New from Template**.

Как бы то ни было, на экране появится окно документа, содержащее в себе новую Web-страницу (рис. 9.12). Созданная нами изменяемая область выделена синей рамкой. Кроме ее содержимого, ни один другой элемент страницы не может быть изменен; при наведении на него курсора мыши последний меняет форму на перечеркнутый круг. Мы даже не можем их выделить. Также невозможно изменить параметры таблицы или ее элементов. Шаблон надежно защищен от редактирования.

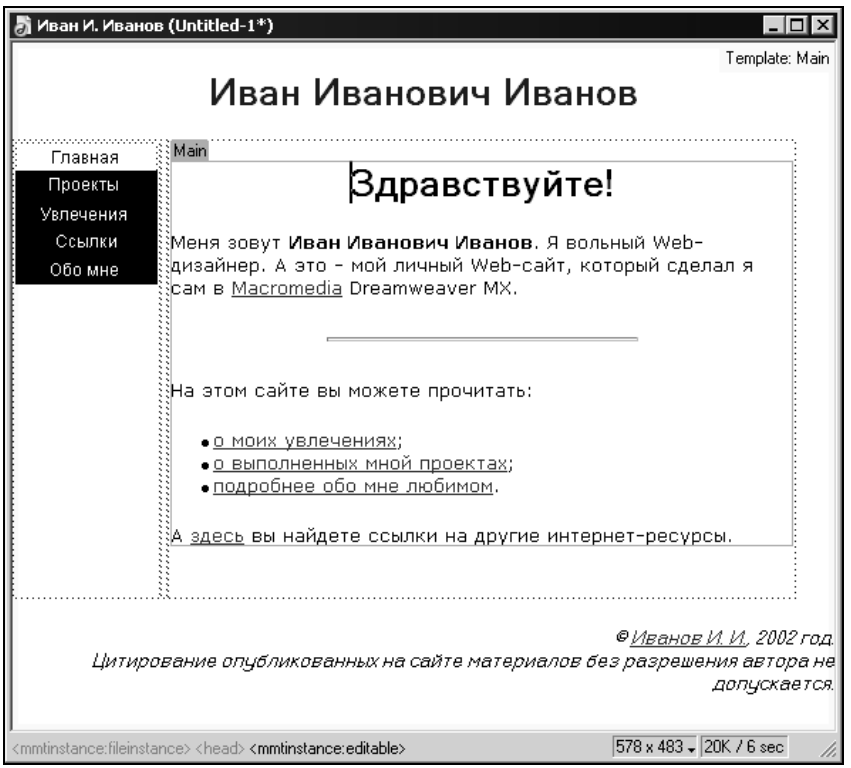


Рис. 9.12. Только что созданная Web-страница, основанная на шаблоне Main

В правом верхнем углу находится небольшой желтый ярлычок, показывающий имя шаблона, на котором базируется создаваемая Web-страница. Таким образом, вы не забудете его имя.

Для поиска изменяемых областей на странице вы также можете пользоваться уже знакомым вам подменю **Editable Regions**, находящемся в подменю **Templates** контекстного меню. Там перечислен список всех имеющихся в шаблоне изменяемых областей; вам нужно будет только выбрать соответствующий пункт. Помните также, что список изменяемых областей приведен внизу подменю **Templates** меню **Modify**.

Если вы переключитесь в режим отображения HTML-кода, вы также не сможете изменять код нередактируемой области вручную. Хотя сможете его просмотреть; HTML-код, принадлежащий шаблону (нередактируемый), выделен в окне документа тускло-серым цветом.

Но хватит созерцать разноцветный HTML-код. Давайте создадим Web-страницу архива новостей. Введите в единственную изменяемую область какой-либо текст. Придумайте какие-нибудь старые новости или просто напишите, что архива пока нет, поскольку нет новостей. Сохраните страницу под именем Archive.htm во вложенной папке HTMLs. В свое время на начальной странице мы поместили гиперссылку, ссылающуюся на эту страницу.

Взгляните на рис. 9.13. Там показано, что у нас должно получиться в результате. Как видите, изменяемые области также выделяются голубым цветом, как при редактировании шаблона.

Рис. 9.13. Изменяемая область на создаваемой странице Archive.htm

Итак, новая страница, базирующаяся на шаблоне, создана! Что можно сделать с ней еще?

Если вам вдруг понадобилось изменить шаблон (например, вы обнаружили в нем ошибку), то вы сможете легко вызвать его на редактирование. Для

этого выберите пункт **Open Attached Template** подменю **Templates** меню **Modify** или аналогичный пункт контекстного меню. На экране тотчас появится окно документа, где будет открыт этот шаблон.

Иногда возникает необходимость создать страницу, чей дизайн сильно отличается от предоставляемого шаблоном. В этом случае вы можете создать на основе данного шаблона страницу, а потом "отвязать" ее от этого шаблона. Для этого выберите пункт **Detach from Template** подменю **Templates** меню **Modify**. Страница будет "откреплена" от шаблона; и желтый ярлычок, мозолящий вам глаза в правом верхнем углу окна документа, пропадет.

Применение шаблонов к уже созданным Web-страницам

Хорошо! Создавать страницы на основе шаблонов "с нуля" мы научились. Но в Web-дизайне, как и во многих других областях человеческой жизни, далеко не всегда приходится создавать что-то с чистого листа. Возникает необходимость использовать чьи-то наработки, часто дорогостоящие, а иной раз и уникальные, неповторимые и невосстановимые. Например, очень часто приходится переделывать дизайн целых сайтов, а это как раз такой случай.

В частности, у нас уже есть набор страниц нашего второго сайта Sample site 2, сделанного на основе фреймов. Нам нужно быстро преобразовать их в новый вид для третьего сайта, который мы создаем сейчас. Как это сделать?

Очень просто!

Давайте преобразуем в новый вид страницу Projects.htm сайта Sample site 2. (Остальные страницы данного сайта преобразуются аналогично.) Для этого найдите соответствующий файл во вложенной папке HTMLs корневой папки второго сайта и скопируйте его во вложенную папку HTMLs нового сайта.

Далее в панели **Site** откройте папку HTMLs сайта Sample site 3 и найдите страницу Projects.htm. Откройте ее в окне документа.

Последующие наши действия будут зависеть от того, активизирована ли у вас панель **Assets** со списком шаблонов. Если она активизирована, применение шаблона к странице выполняется очень просто. Для этого выберите в списке шаблонов нужный и нажмите кнопку **Apply** в левом нижнем углу панели, либо вызовите контекстное или дополнительное меню и выберите пункт **Apply**, либо просто перетащите нужный шаблон из списка в окно документа. После этого на экране появится диалоговое окно **Inconsistent Region Names**, показанное на рис. 9.14.

Большую часть этого диалогового окна занимает список, состоящий из двух колонок: в левой отображаются имена изменяемых областей, имеющихся в открытой странице, а в правой — имена изменяемых областей шаблона, куда будет перемещено их содержимое. В нашем случае страница Projects.htm

старого сайта не имеет изменяемых областей, поэтому в списке будет присутствовать только один значимый пункт — **Document body** (тело страницы). В правой колонке списка против этого пункта будет стоять надпись **<Not Resolved>**, обозначающая, что Dreamweaver не знает, куда поместить содержимое страницы.

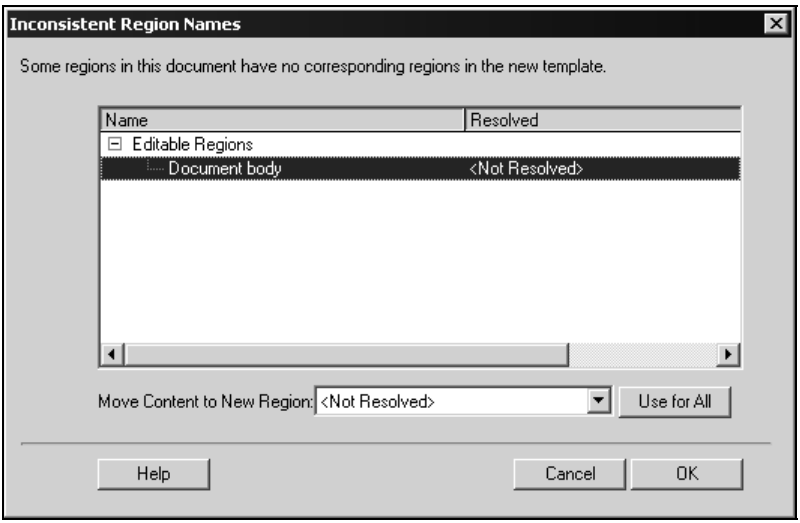


Рис. 9.14. Диалоговое окно **Inconsistent Region Names**

Под списком находится раскрывающийся список **Move Content to New Region**, в котором задается изменяемая область шаблона, куда будет перемещено содержимое страницы. Среди пунктов этого списка будут перечислены все изменяемые области шаблона. Если же вы выберете пункт **Nowhere**, содержимое страницы будет потеряно. Чтобы переместить содержимое всех изменяемых областей открытой страницы в какую-то одну, чье имя выбрано в списке **Move Content to New Region**, нажмите кнопку **Use for All**.

Выделите единственный пункт списка и выберите в раскрывающемся списке **Move Content to New Region** нужную изменяемую область шаблона, в нашем случае Main. После чего нажмите кнопку **OK** для перемещения в нее содержимого страницы или **Cancel** — для отказа от этого.

Полученный после нажатия кнопки **OK** результат показан на рис. 9.15. Как видите, Dreamweaver прекрасно справился со своей задачей. Теперь вы можете, если хотите, подредактировать содержимое страницы и сохранить ее.

Теперь преобразуем в новый вид страницу Links.htm сайта Sample site 2. Найдите ее в папке HTMLs корневой папки второго сайта, скопируйте его в папку HTMLs нового сайта и откройте в окне документа. Предположим теперь, что панель **Assets** у вас закрыта.

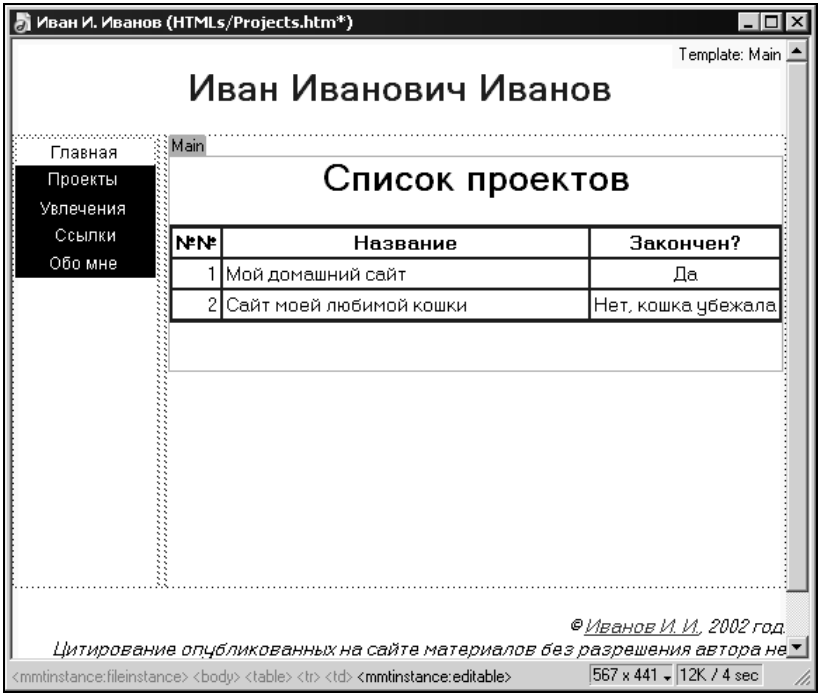


Рис. 9.15. Готовая страница списка проектов Projects.htm

Выберите пункт **Apply Template to Page** подменю **Templates** меню **Modify**. На экране появится диалоговое окно **Select Template**, похожее на окно **Save as Template** (см. рис. 9.4). Выберите нужный шаблон в списке **Templates** и нажмите кнопку **OK**. После этого опять появится диалоговое окно **Inconsistent Region Names** (см. рис. 9.14), где вы будете должны выбрать изменяемую область, куда будет помещено содержимое страницы, и еще раз нажать кнопку **OK**.

Заметьте, что все параметры страниц Projects.htm и Links.htm (цвет фона, текста и гиперссылок) пропадут после применения к странице шаблона. Точнее, они не пропадут, а станут такими же, как у шаблона, поскольку параметры страниц задаются шаблоном и в отдельных страницах не могут быть изменены.

Как видите, можно очень просто подогнать под один шаблон все существующие у вас Web-страницы. Скопируйте остальные страницы предыдущего (второго) сайта в третий (напомним, что все они содержатся во вложенной папке HTMLs корневой папки сайта) и переделайте их так же, как и страницы Projects.htm и Links.htm. После этого можно считать, что наш новый сайт почти готов.

Но что, если страница, которую вы хотите "подогнать" под шаблон, уже основана на каком-то шаблоне? Да ничего сложного! Dreamweaver выполнит

за вас львиную часть работы — вам же останется только указать ему, что куда распихивать.

Когда вы применяете шаблон к странице, которая базируется на другом шаблоне, Dreamweaver сравнивает имена изменяемых областей этих двух шаблонов. Если у каких-то изменяемых областей, находящихся в разных шаблонах, эти имена равны, то содержимое одной области переносится в другую. Это может вам сильно помочь в том случае, если вы не слишком радикально меняете дизайн своего сайта. Для этого просто создайте в новом шаблоне изменяемые области с теми же именами, что и в старом, а остальное — как говорится, дело техники.

Если же в старом шаблоне встретится изменяемая область, для которой Dreamweaver не найдет "достойной" пары в новом шаблоне, понадобится ваша помощь. В этом случае Dreamweaver выведет на экран диалоговое окно **Inconsistent Region Names** (см. рис. 9.14). Здесь вы должны будете выбрать изменяемую область нового шаблона, куда будет перемещено содержимое изменяемой области старого шаблона, не имеющей пары. После нажатия кнопки **OK** все будет сделано в лучшем виде.

Если вы случайно применили к странице не тот шаблон или что-то напутали с изменяемыми областями и их содержимым, Dreamweaver предоставляет вам возможность "отката". Для этого воспользуйтесь "волшебной палочкой" всех изучающих новые программы методом "научного тыка" — комбинацией клавиш <Ctrl>+<Z> или пунктом **Undo** меню **Edit**.

А теперь рассмотрим еще один очень важный вопрос. Он касается гиперссылок, используемых в шаблонах, и поддержания их корректности.

Гиперссылки в шаблонах

Чтобы вы лучше поняли, о чем вам хочет поведать автор, ниже приведена физическая структура нашего сайта. Как и раньше, имена папок не имеют расширения, а имена файлов — имеют. Эта структура приведена не полностью, чтобы не загромождать текст книги и не запутывать вас, а ровно настолько, чтобы вы поняли.

```
<Корневая папка сайта>
  default.htm
  HTMLs
    archive.htm
    . . .
  Pics
    NAVBAR
      . . .
```

```
    spacer.gif
    . . .
  Templates
    main.dwt
```

Возьмем страницу default.htm. Все гиперссылки, находящиеся на ней, даны относительно файла default.htm. Например, гиперссылка, ссылающаяся на страницу списка проектов, имеет следующий адрес:

```
HTMLs/Projects.htm
```

Проверьте по вышеприведенной схеме — он правилен.

Теперь мы создали на основе этой страницы шаблон Main.dwt. Как вы уже знаете, все шаблоны сохраняются в специальной вложенной папке Templates корневой папки сайта, поэтому гиперссылки, содержащиеся в этом шаблоне, по идее, должны "осиротеть". Однако Dreamweaver автоматически скорректирует все ссылки, чтобы они указывали на те же файлы. И ссылка, указывающая на страницу списка проектов, будет иметь такой адрес:

```
../HTMLs/Projects.htm
```

При создании страницы на основе этого шаблона (пусть это будет страница Archive.htm) Dreamweaver опять же корректирует интернет-адреса всех гиперссылок. Таким образом, наша многострадальная гиперссылка будет иметь вид:

```
Projects.htm
```

Проверьте по вышеприведенной структуре сайта — она корректна.

Все это справедливо не только для гиперссылок. Имена файлов графических изображений, фильмов и прочих внедренных объектов корректируются точно так же. Dreamweaver следит за тем, чтобы ни одна ссылка не стала "сиротой".

Конечно, мы рассматривали только те гиперссылки, адреса которых задавались относительно файла страницы. Гиперссылки с адресами, заданными относительно корневой папки сайта, всегда останутся корректными, несмотря на любые перемещения между папками.

Из этого следуют два важнейших правила, которые вам всегда следует соблюдать при использовании шаблонов и интернет-адресов, заданных относительно файлов страницы.

1. Если вы создаете гиперссылку в обычной Web-странице, задавайте ее адрес относительно файла этой страницы.
2. Если вы создаете гиперссылку в шаблоне, задавайте ее адрес относительно файла шаблона. Dreamweaver автоматически выполнит преобразование адресов при создании страниц на основе этого шаблона.

Экспорт Web-страниц, основанных на шаблонах

Иногда бывает так, что вам нужно преобразовать страницы сайта, основанные на шаблонах, в обычные. Иначе говоря, вам нужно "очистить" HTML-код таких Web-страниц от всех служебных тегов, атрибутов и комментариев, которые поместил туда Dreamweaver. В терминологии последнего такое действие называется *экспортом*.

К сожалению, Dreamweaver позволяет экспортировать только сайт целиком, но никак не отдельную страницу. Хотя, это может быть оправдано. В самом деле, экспортировать стоит лишь уже законченные сайты, когда вносить изменения в страницы уже не нужно. (Вспомните, что шаблоны и были созданы как раз для облегчения создания страниц и внесения в них изменений.) Если же страница не закончена, то ее необходимо прежде всего доделать, а уже потом экспортировать.

Чтобы экспортировать сайт, сначала откройте одну из его страниц в окне документа. Далее выберите пункт **Export without Markup** в подменю **Templates** меню **Modify**. После этого на экране появится диалоговое окно **Export Site Without Template Markup** (рис. 9.16).

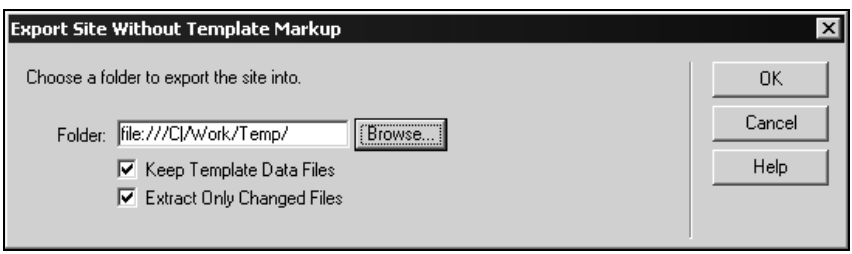


Рис. 9.16. Диалоговое окно **Export Site Without Template Markup**

В поле ввода **Folder** введите имя и путь папки, куда будет экспортирован ваш сайт. (Эта папка станет корневой для нового сайта.) Также вы можете нажать кнопку **Browse** и выбрать нужную папку в появившемся на экране диалоговом окне выбора папки.

Если вы хотите экспортировать файлы шаблонов, включите флажок **Keep Template Data Files** (впрочем, он включен по умолчанию). Если вы перед этим уже экспортировали в выбранную папку сайт, и теперь хотите только обновить его файлы, включите флажок **Extract Only Changed Files** (он также включен по умолчанию).

Задав нужные данные, нажмите кнопку **OK**. Если вы хотите отказаться от экспорта сайта, нажмите кнопку **Cancel**.

Обновление страниц, созданных на основе шаблонов

Теперь представим себе такую ситуацию. Вам понадобилось что-то изменить в созданном ранее шаблоне. Может быть, это простая орфографическая ошибка или опечатка, а может, неверная гиперссылка или неудачная расцветка текста в неизменяемой области. Это не важно. Важно то, что вы открыли этот шаблон (вы знаете, как это сделать) и соответственно изменили его. Как перенести эти изменения во все созданные на его основе Web-страницы?

Как правило, вам не нужно об этом заботиться. Dreamweaver берет на себя контроль за актуальностью страниц, созданных на основе измененного шаблона. И едва вы сохраните этот шаблон, он сразу переносит все сделанные изменения в соответствующие страницы. При этом на экране появляется диалоговое окно **Update Files**, показанное на рис. 6.14. Вам останется только нажать кнопку **Update** (рекомендуется) или **Don't Update** (может иногда пригодиться). Далее на экране появится диалоговое окно со статистикой (сколько страниц было обновлено); закройте его щелчком по кнопке **Close**.

Но иногда может понадобиться обновить страницы сайта вручную.

Чтобы обновить страницу, открытую в активном окне документа, просто выберите пункт **Update Current Page** подменю **Templates** меню **Modify**. Страница будет тотчас обновлена. Если у вас активна панель **Assets**, вы также можете выбрать пункт **Update Current Page** контекстного меню.

Если вы хотите обновить все страницы сайта или все страницы, основанные на каком-либо шаблоне, выберите пункт **Update Pages** подменю **Templates** меню **Modify** или пункт **Update Site** контекстного или дополнительного меню панели **Assets**. На экране появится диалоговое окно **Update Pages**, показанное на рис. 6.38. Использование этого диалогового окна описано в *главе 5*, поэтому здесь ограничимся лишь минимальными рекомендациями. Включите флажок **Templates**, установите другие параметры и нажмите кнопку **Start**. После того как Dreamweaver закончит обновление страниц, закройте это окно, нажав кнопку **Close**.

Вот и все об обновлении страниц.

Управление шаблонами в окне шаблонов

Вы можете управлять шаблонами так же легко, как управляете содержимым библиотеки *(см. главу 5)*.

Вы можете переименовать любой шаблон. Для этого выберите его в списке и либо щелкните мышью по его имени, либо выберите пункт **Rename** в контекстном или дополнительном меню. Вместо имени данного шаблона появится поле ввода, где вы сможете ввести новое имя. После этого нажмите

клавишу <Enter> для сохранения введенного имени или <Esc> — для отказа от него и возврата старого имени.

Чтобы удалить ненужный шаблон, выберите его в списке и либо нажмите клавишу <Del>, либо выберите пункт **Delete** контекстного или дополнительного меню, либо нажмите кнопку **Delete** (рис. 9.17), расположенную в правом нижнем углу панели. Dreamweaver спросит вас, действительно ли вы хотите удалить этот шаблон; нажмите кнопку **Да** (**Yes**) или **Нет** (**No**). Если вы нажмете кнопку **Да**, шаблон будет безвозвратно удален.

Рис. 9.17. Кнопка **Delete** панели **Assets**

Рис. 9.18. Кнопка **Refresh Site List** панели **Assets**

Хорошо подумайте, прежде чем удалять шаблон. Ведь если вы его удалите, основанные на нем страницы потеряют с ним связь, а значит, вы лишитесь всех преимуществ, предоставляемых шаблонами.

Как уже неоднократно говорилось, шаблоны — неотъемлемая собственность сайта. Конечно, вы можете создать страницу на основе шаблона, находящегося в другом сайте, но этого лучше не делать. Ведь после того, как вы закончите работу над сайтом, откуда был взят шаблон, вы его (сайт) удалите, чтобы освободить место на жестком диске. И страница, созданная на базе этого шаблона, потеряет с ним связь со всеми вытекающими отсюда последствиями. Так что лучше следовать всем известному принципу "все свое несу с собой", т. е. скопировать нужный шаблон в текущий сайт.

Скопировать нужный шаблон можно в окне Проводника Windows или аналогичной программе управления файлами (например, FAR). Но проще сделать это в самом Dreamweaver. Для этого, прежде всего, в панели **Site** активизируйте сайт, где находится нужный шаблон. Далее откройте панель **Assets**, если она не открыта, переключитесь на список шаблонов, выберите необходимый шаблон в списке и щелкните по нему правой кнопкой мыши, чтобы открыть контекстное меню. В подменю **Copy to Site** будут перечислены все введенные в Dreamweaver сайты; просто выберите нужный.

Также Dreamweaver может показать вам, в каком файле сохранен тот или иной шаблон. Для этого выберите пункт **Locate in Site** контекстного или дополнительного меню. После этого будет активизирована панель **Site**, в списке файлов которого будет выбран найденный файл шаблона.

Иногда после манипуляций в панели **Site** Dreamweaver или окне Проводника Windows список шаблонов отображает неверную информацию. В частности, в нем могут быть перечислены уже удаленные шаблоны. Чтобы обновить содержимое списка шаблонов, выберите пункт **Refresh Site List** контекстного или дополнительного меню или нажмите кнопку **Refresh Site List**, расположенную в правом нижнем углу панели **Assets** (рис. 9.18).

Новые возможности шаблонов

Здесь мы рассмотрим некоторые весьма полезные возможности, появившиеся в Dreamweaver MX. Конечно же, эти возможности касаются поддержки шаблонов.

Изменяемые атрибуты

Dreamweaver MX позволяет вам сделать некоторые атрибуты любых тегов HTML-кода шаблона *изменяемыми*. Вы можете изменять значения этих атрибутов в страницах, созданных на основе шаблона, содержащего этот тег. Например, вы можете менять цвет фона тех или иных страниц или цвет какой-либо надписи.

Чтобы сделать какой-то атрибут тега изменяемым, сначала откройте в окне документа нужный шаблон. После этого выберите элемент страницы, атрибут тега которого вы хотите сделать изменяемым. Пусть, например, это будет заголовок нашего сайта (текст "Иван Иванович Иванов" в верхней части страницы). Выделите весь заголовок или просто щелкните по тегу `<FONT>` в секции тегов. Теперь все готово.

Выберите пункт **Make Attribute Editable** подменю **Templates** меню **Modify**. На экране появится диалоговое окно **Editable Tag Attributes** (рис. 9.19).

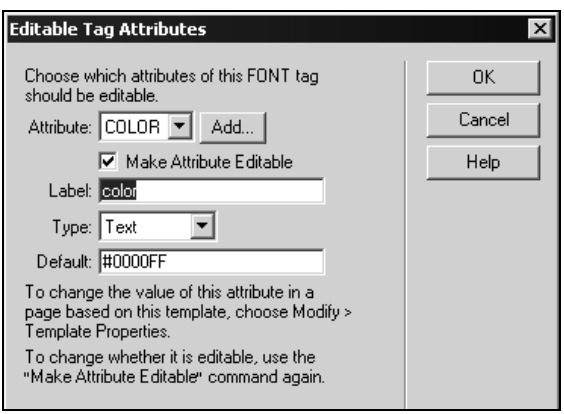


Рис. 9.19. Диалоговое окно **Editable Tag Attributes**

В раскрывающемся списке **Attribute** выберите нужный атрибут (впрочем, там он один — `COLOR`). Если же вы его там не найдете, нажмите кнопку **Add**, введите в единственном поле ввода диалогового окна добавления нового атрибута (рис. 9.20) имя нужного атрибута и нажмите кнопку **OK**.

Чтобы сделать выбранный атрибут изменяемым, включите флажок **Make Attribute Editable**. После этого станет доступной группа элементов управления, находящаяся ниже.

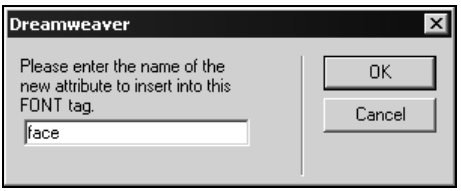


Рис. 9.20. Диалоговое окно добавления нового атрибута

В поле ввода **Label** введите имя созданного изменяемого атрибута. Вообще-то, Dreamweaver создаст его сам, и нам в большинстве случаев не придется вводить его самим.

Раскрывающийся список **Type** задает тип значения атрибута. В нем доступны пять пунктов:

- **Text** — текстовое значение, строка;
- **URL** — интернет-адрес;
- **Color** — значение цвета;
- **True/False** — значение типа "да-нет";
- **Number** — числовое значение.

Выберите пункт **Color**, т. к. атрибут `COLOR` тега `<FONT>` задает именно цвет шрифта.

В поле ввода **Default** введите значение изменяемого атрибута по умолчанию. Dreamweaver сам подставит туда текущее значение атрибута выбранного вами тега, но вы, конечно, можете его изменить.

Закончив ввод данных, нажмите кнопку **OK**. Если вы передумали делать атрибут изменяемым, нажмите кнопку **Cancel**.

Теперь сохраните шаблон и закройте его. Dreamweaver выведет диалоговое окно **Update Files**; нажмите кнопку **Update**, чтобы обновить все созданные на основе этого шаблона Web-страницы.

Теперь давайте откроем какую-нибудь Web-страницу, основанную на этом шаблоне, и изменим значение только что созданного атрибута. Пусть это будет страница Links.htm.

Чтобы задать значение изменяемого атрибута, выберите пункт **Template Properties** в меню **Modify**. На экране появится диалоговое окно **Template Properties** (рис. 9.21).

Большую часть данного окна занимает список изменяемых атрибутов, заданных в шаблоне, на основе которого создана эта страница. В левой колонке списка отображается имя атрибута, а в правой — его текущее значение. Если вы выберете какой-либо атрибут в списке, под ним появятся элементы управления, с помощью которых вы сможете задать его новое значение. В нашем случае это будет селектор цвета **color**.

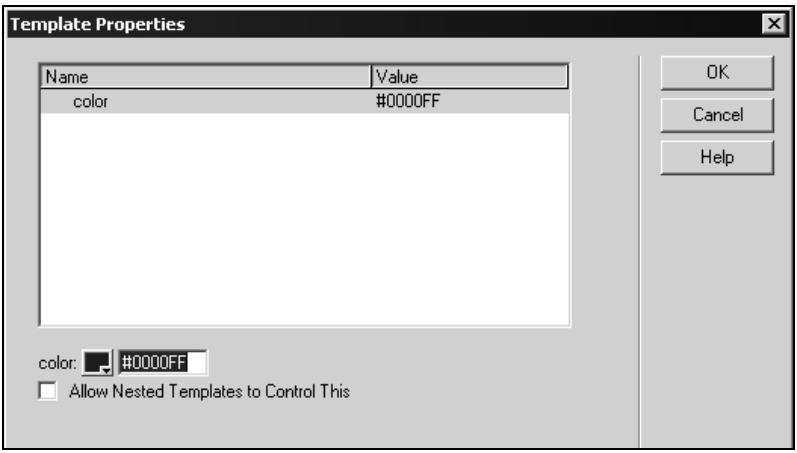


Рис. 9.21. Диалоговое окно **Template Properties**

Задайте для атрибута `color` темно-красный цвет шрифта. После этого нажмите кнопку **OK**. И посмотрите на результат. Нет смысла приводить здесь рисунок, т. к. черно-белая иллюстрация не передает цвет. Но, если вы все сделали правильно, шрифт заголовка должен стать темно-красным.

Необязательные области

Эх, ну до чего же хорош Dreamweaver MX! Что он позволяет сделать! Другие программы Web-дизайнеров, да и старые версии Dreamweaver ему и в подметки не годятся!..

Вы спросите: что это автор так расхваливает этот Dreamweaver, который и так уже у вас в печенках сидит? Ну нравится он автору, и ничего не поделаешь.

Давайте еще раз посмотрим на страницы нашего нового сайта Sample site 3. Вроде, все неплохо. Почти все страницы созданы на основе шаблона Main. Это хорошо: нам не нужно вносить исправления во все повторяющиеся элементы на всех страницах сайта — это сделает за нас Dreamweaver. Уже одно это — повод для восхищения этой замечательной программой.

Но одна страница нашего сайта не основана на шаблоне. Да-да, она самая — главная страница default.htm. Она сильно отличается от остальных тем, что содержит колонку новостей, которой нет в шаблоне. Вносить эту колонку в шаблон смысла нет — на остальных страницах она отсутствует. А даже если мы это сделаем, на страницах сайта появится дополнительный HTML-код, увеличивающий размер файлов и — абсолютно ненужный.

Вот если бы имелась возможность создавать необязательные области, которые, в зависимости от некоего условия, присутствовали бы или не присутствовали бы на странице...

Но такая возможность есть! Она появилась именно в Dreamweaver MX. Он может создавать *необязательные области* (они так и называются), которые могут либо быть, либо не быть. И ответ на риторический вопрос "быть или не быть" даете вы.

Давайте создадим необязательную область, которая будет содержать колонку новостей. Для этого откроем шаблон Main в окне документа. Изменим размеры ячейки основного содержимого, чтобы справа осталось место под колонку новостей. Поместим в нее новую таблицу разметки с тремя ячейками, расположенными вертикально; они будут содержать соответственно заголовок, собственно новости и ссылку на страницу архива новостей. Как это сделать, вы уже знаете, а если забыли, посмотрите *главу 8*. Далее поместим в ячейки заголовка и ссылки на страницу архива новостей соответствующий текст, а ячейку новостей оставим пустой. То, что у нас должно получиться, показано на рис. 9.22.

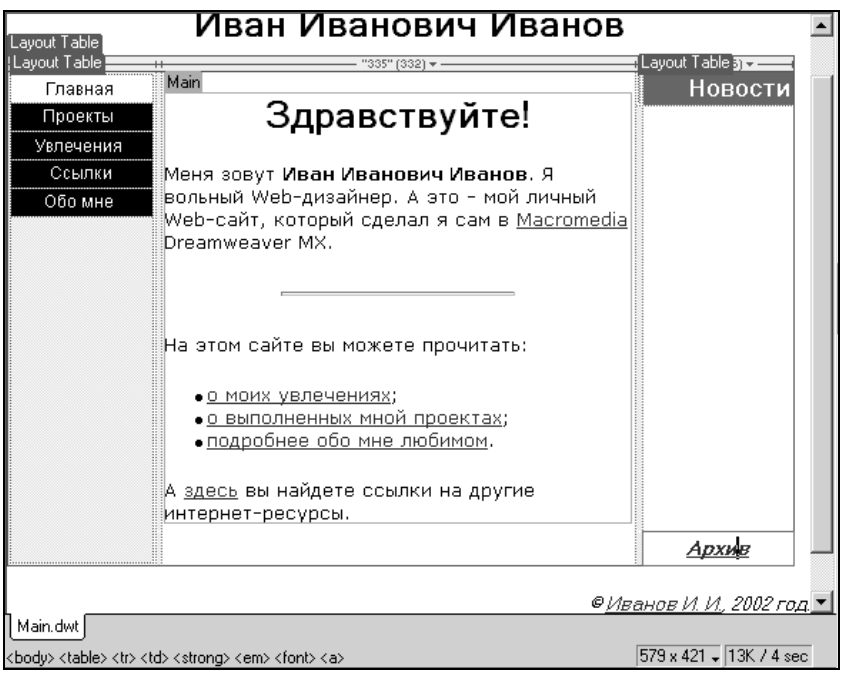


Рис. 9.22. Шаблон Main с колонкой новостей

Теперь поместим всю таблицу разметки, в которой находится колонка новостей, в необязательную область. Для этого сначала выделим эту таблицу. Далее нажмем кнопку **Optional Region** (рис. 9.23) на вкладке **Templates** панели объектов. Также можно выбрать пункт **Optional Region** подменю **Template Objects** меню **Modify** или одноименный пункт подменю **Templates** контекст-

ного меню. На экране появится диалоговое окно **New Optional Region**, показанное на рис. 9.24.

Рис. 9.23. Кнопка **Optional Region** панели объектов

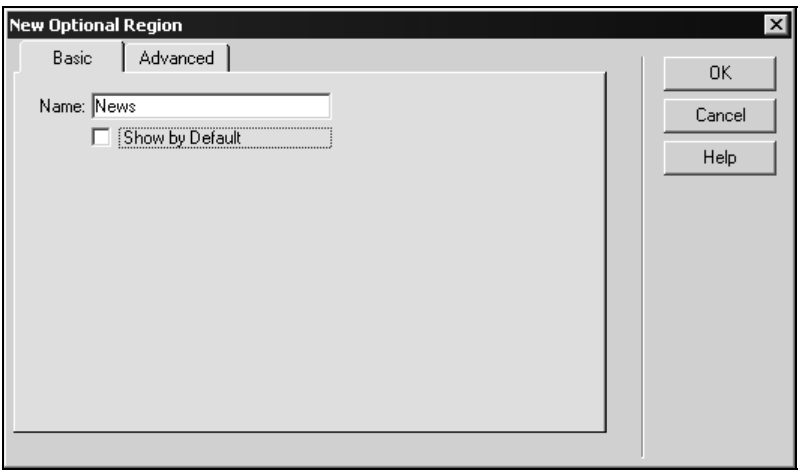


Рис. 9.24. Диалоговое окно **New Optional Region**

В поле ввода **Name** введем имя создаваемой необязательной области — `News`. Отключим флажок **Show by Default**, т. к. мы не хотим, чтобы она отображалась по умолчанию. И нажмем кнопку **OK**. То, что у нас должно получиться, показано на рис. 9.25.

Скорее всего, вам придется подредактировать HTML-код. Проверьте — он должен выглядеть так (служебные комментарии, созданные Dreamweaver для выделения необязательной области, выделены полужирным шрифтом):

```
<!-- TemplateBeginIf cond="_document['News']" -->
  <TD WIDTH="5"> </TD>
    Это ячейка внешней таблицы, задающая просвет между основным
    содержимым и колонкой новостей
  <TD WIDTH="98" VALIGN="TOP">
    <TABLE WIDTH="100%" BORDER="0" CELLPADDING="0" CELLSPACING="0">
      Содержимое внутренней таблицы разметки
    </TABLE>
  </TD>
<!-- TemplateEndIf -->
```

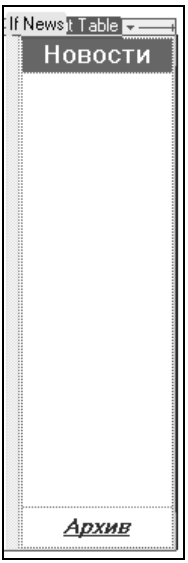


Рис. 9.25. Необязательная область, включающая в себя внутреннюю таблицу разметки страницы default.htm

Внутри необязательной области должны оказаться ячейка просвета и ячейка, в которой находится колонка новостей. Если это не так, отредактируйте HTML-код.

Вот, собственно, и все. Мы создали необязательную область.

Теперь, если вы создаете страницу на основе шаблона Main, то можете задать присутствие или неприсутствие в ней колонки новостей. Сделать это можно с помощью уже знакомого вам диалогового окна **Template Properties** (см. рис. 9.21). Выберите пункт **Template Properties** в меню **Modify** — и вы увидите, что в списке этого окна присутствует пункт **News**, т. е. параметр, задающий присутствие созданной нами необязательной области. А в нижней части диалогового окна находится флажок **Show News**, включив который вы зададите присутствие области на странице.

На этом все? Увы, нет.

Дело в том, что необязательная область Dreamweaver по своей природе не является изменяемой. Нам нужно поместить в нее изменяемую область, чтобы мы смогли задать содержимое колонки новостей.

Но мы сделаем лучше. И Dreamweaver нам поможет.

Повторяющиеся области

Кроме необязательных, Dreamweaver MX позволяет также создавать *повторяющиеся области*. Повторяющаяся область может содержать сколько угодно содержимых *(пунктов)*. В качестве примера такой области можно привести

описание новости в колонке на странице default.htm — их может быть много, но в шаблоне для него создана только одна область.

Давайте создадим такую область.

Откройте шаблон Main и поместите текстовый курсор в ячейку новостей внутренней таблицы разметки. После этого нажмем кнопку **Repeating Region** (рис. 9.26) на вкладке **Templates** панели объектов. Также можно выбрать пункт **Repeating Region** подменю **Template Objects** меню **Modify** или одноименный пункт подменю **Templates** контекстного меню. На экране появится диалоговое окно **New Repeating Region**, показанное на рис. 9.27.

Рис. 9.26. Кнопка **Repeating Region** панели объектов

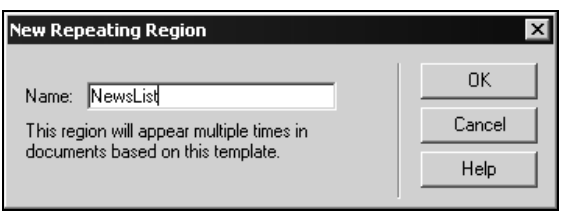


Рис. 9.27. Диалоговое окно **New Repeating Region**

Введите имя создаваемой повторяющейся области в единственном поле ввода этого окна. Давайте назовем ее `NewsList`. После этого нажмите кнопку **OK**. Результат показан на рис. 9.28.

Рис. 9.28. Повторяющаяся область

Теперь отформатируйте содержимое этой области как обычный текстовый абзац, выбрав в раскрывающемся списке **Format** редактора свойств пункт **Paragraph**. Если хотите, уменьшите шрифт текста на одну ступень. На этом создание повторяющейся области можно считать законченным.

Но проблема в том, что повторяющаяся область Dreamweaver также не является изменяемой. Поэтому нам нужно поместить внутрь нее еще и изменяемую область. Таким образом, получится троекратная вложенность "не-

обязательная область — повторяющаяся область — изменяемая область". Сложно, конечно, зато очень гибко.

Поставьте текстовый курсор внутри содержимого повторяющейся области. И создайте изменяемую область. Вы уже знаете, как это делается. Назовите ее `NewsItem`. После этого удалите остатки содержимого повторяющейся области, не входящего в изменяемую область. И поправьте HTML-код — он должен выглядеть так:

```
<P><EM><FONT SIZE="-1">
  <!-- TemplateBeginEditable name="NewsItem" -->
    Содержимое изменяемой области NewsItem
  <!-- TemplateEndEditable -->
</FONT></EM></P>
```

Внутри изменяемой области должен находиться только текст описания новости, без всех форматирующих тегов. (Вставленные Dreamweaver служебные комментарии помечены полужирным шрифтом.) Результат показан на рис. 9.29.

Рис. 9.29. Окончательный вид колонки новостей в шаблоне Main

Но не торопитесь теперь применять шаблон Main к странице default.htm. Если вы это сделаете, Dreamweaver запихает в изменяемую область Main все содержимое этой страницы, и вам придется ее долго редактировать. Лучше удалите страницу default.htm и создайте ее заново на основе шаблона Main. Поскольку текст приветствия уже имеется в шаблоне, вам даже не нужно будет ничего вводить с клавиатуры. Не забудьте только сохранить новую главную страницу сайта в файле default.htm.

Но как же колонка новостей? Сейчас мы ей займемся.

Откройте новую страницу default.htm, если вы ее уже закрыли. Выберите пункт **Template Properties** в меню **Modify** и в появившемся на экране диалоговом окне **Template Properties** задайте для параметра `News` значение `true`, для чего просто включите флажок **Show News**. После этого нажмите кнопку **OK**.

Теперь вы видите, что правее основного содержимого страницы появилась колонка новостей. В ней находятся уже знакомые вам повторяющаяся и

изменяемая области, вложенные друг в друга. Выделите содержимое изменяемой области `NewsItem`, удалите его и введите описание какой-либо новости. У вас должно получиться нечто, похожее на рис. 9.30.

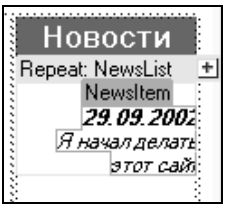


Рис. 9.30. Текст описания первой новости, введенный в колонку

Так, одна новость у нас есть. Но как добавить остальные? Очень просто!

Для работы с пунктами повторяющихся областей Dreamweaver предоставляет набор особых кнопок, появляющихся в заголовке повторяющейся области, и набор пунктов меню. Эти пункты меню находятся в подменю **Repeated Entries**, вложенного в подменю **Templates** меню **Modify**, а также в подменю **Templates** контекстного меню. Все они перечислены в табл. 9.1.

Таблица 9.1. *Кнопки и пункты меню, предназначенные для работы с пунктами повторяющихся областей*

Пункт меню	Кнопка	Назначение
New Entry After Selection	Знак "+"	Добавляет новый пункт ниже (правее) выделенного
New Entry Before Selection		Добавляет новый пункт выше (левее) выделенного
New Entry at End		Добавляет новый пункт в самый конец
New Entry at Beginning		Добавляет новый пункт в самое начало
Delete Repeating Entry	Знак "–"	Удаляет выделенный пункт
Move Entry Up	Знак "↑"	Перемещает выделенный пункт выше
Move Entry Down	Знак "↓"	Перемещает выделенный пункт ниже
Move Entry to Beginning		Перемещает выделенный пункт в самый верх
Move Entry to End		Перемещает выделенный пункт в самый низ
Cut Repeating Entry		Вырезает выделенный пункт в буфер обмена
Copy Repeating Entry		Копирует выделенный пункт в буфер обмена

Как вы уже заметили, в табл. 9.1 не указан пункт **Paste Repeating Entry**, который осуществлял бы вставку пункта из буфера обмена. Это выполняется, как обычно, вызовом пункта **Paste** меню **Edit** или нажатием комбинации клавиш <Ctrl>+<V>.

С помощью этих пунктов вы можете добавлять, удалять и перемещать пункты повторяющейся области. Создайте еще два или три пункта и впишите в них текст новостей. Результат вы можете увидеть на рис. 9.31.

Dreamweaver MX — настоящий король шаблонов!

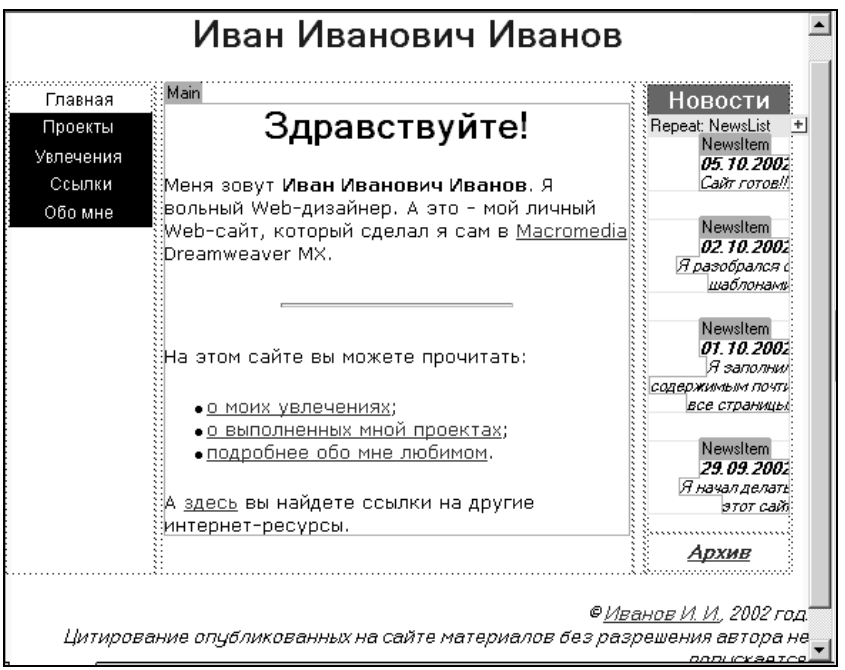


Рис. 9.31. Готовая главная страница сайта Sample site 3 с готовой колонкой новостей

Необязательные изменяемые области

Далеко не всегда приходится создавать такие сложные структуры областей, наподобие той, что мы только что сделали. Часто нужно просто создать в шаблоне необязательную область и поместить в нее изменяемую область. Для таких случаев Dreamweaver предоставляет возможность сделать все это за один раз, т. е. создать *необязательную изменяемую область*.

Откройте созданный нами в начале этой главы пустой шаблон Sample. С ним-то мы и будем экспериментировать.

Чтобы создать необязательную изменяемую область, нажмите кнопку **Editable Optional Region** (рис. 9.32) на вкладке **Templates** панели объектов. Также вы можете выбрать пункт **Editable Optional Region** подменю **Template Objects** меню **Insert**.

Рис. 9.32. Кнопка **Editable Optional Region** панели объектов

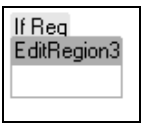


Рис. 9.33. Необязательная изменяемая область, представляющая собой комбинацию из необязательной и изменяемой областей

На экране появится уже знакомое вам диалоговое окно **New Optional Region**. Введите имя создаваемой области в поле ввода **Name**, если хотите, отключите флажок **Show by Default** и нажмите кнопку **OK**. После этого Dreamweaver создаст необязательную изменяемую область (рис. 9.33).

Как видите, Dreamweaver просто вкладывает изменяемую область внутрь необязательной. Вдобавок он еще и называет изменяемую область по-своему, не спрашивая нас. Но все равно это весьма удобно.

Табличная повторяющаяся область

Еще одна весьма удобная возможность Dreamweaver — создание *табличных повторяющихся областей*. Это обычная повторяющаяся область, пункты которой организованы в виде таблицы. При этом каждая ячейка такой таблицы содержит изменяемую область, т. е. вы можете вводить данные в каждую ее ячейку.

Как и в случае с необязательной изменяемой областью, табличную повторяющуюся область можно создать только с помощью кнопки панели объектов. Это кнопка **Repeating Table** (рис. 9.34) вкладки **Templates**. Также вы можете выбрать пункт **Repeating Table** подменю **Template Objects** меню **Insert**. После этого на экране появится диалоговое окно **Insert Repeating Table** (рис. 9.35).

Рис. 9.34. Кнопка **Repeating Table** панели объектов

Большинство элементов управления этого окна знакомо вам по диалоговому окну **Insert Table**. Поэтому здесь они рассматриваться не будут. Представим только элементы управления, находящиеся в группе **Repeat row of table**, которые нам пока еще незнакомы.

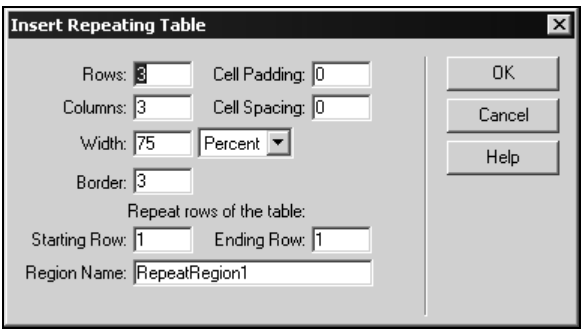


Рис. 9.35. Диалоговое окно **Insert Repeating Table**

Поля ввода **Starting Row** и **Ending Row** позволяют задать номера соответственно первой и последней строк таблицы, которые будут помещены в создаваемую повторяющуюся область. (Это строки, которые станут пунктами повторяющейся области.) Если вы не хотите вносить в повторяющуюся область первую и последнюю строки, в которых часто помещают заголовок и "поддон" таблицы, задайте соответствующие номера.

В поле ввода **Region Name** вводится имя создаваемой повторяющейся области.

Табличная повторяющаяся область будет создана после нажатия кнопки **OK**. Эту область вы можете увидеть на рис. 9.36.

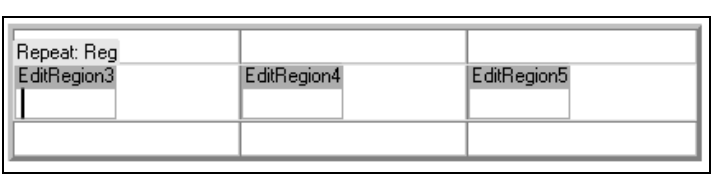


Рис. 9.36. Табличная повторяющаяся область — комбинация таблицы, повторяющейся области и изменяемых областей

Как видите, Dreamweaver и в этом случае особо не мудрствует: он просто создает таблицу, помещает указанные вами строки в повторяющуюся область и создает в каждой ячейке этих строк изменяемые области. Это, кстати, очень удобно; мы могли бы создать такую область для нашей колонки новостей. Попробуйте, если хотите, — это будет вашей самостоятельной работой.

Вложенные шаблоны

Последняя возможность работы с шаблонами, предоставляемая Dreamweaver MX, которую мы рассмотрим, — это *вложенные шаблоны*. Вложенный шаблон — это шаблон, созданный на основе другого шаблона (так называемого *базового шаблона*). При создании вложенного шаблона используются изменяемые области базового шаблона: в них вносится новое неизменяемое содержимое и создаются новые изменяемые области.

Давайте создадим вложенный шаблон, на основе которого будет сделана страница архива новостей Archive.htm. Мы создадим повторяющуюся область, в которой и будет помещаться текст новостей. За основу возьмем колонку новостей нашего основного шаблона Main.

Сначала удалим старую страницу Archive.htm. После этого создадим на основе шаблона Main новую страницу. Сохраним ее как шаблон, выбрав пункт **Save as Template** меню **File**. (Также можно нажать кнопку **Make Nested Template** (рис. 9.37) вкладки **Templates** панели объектов.) В поле ввода **Save As** диалогового окна **Save As Template** (см. рис. 9.4) введем имя нового шаблона — `NewsArchive`. После этого нажмем кнопку **Save**. Готовый шаблон NewsArchive будет в точности похож на шаблон Main. Пока...

Рис. 9.37. Кнопка **Make Nested Template** панели объектов

Теперь удалим все содержимое изменяемой области Main. Сейчас мы создадим для нее новое содержимое в виде повторяющейся и изменяемой областей.

Прежде всего, введите текст "Архив новостей" и отформатируйте его как заголовок. После этого поместите текстовый курсор ниже этого текста и создайте там повторяющуюся область. Назовите ее `News Archive`.

Вы, наверно, уже заметили, что изменяемая область Main базового шаблона изменила свой вид — ее граница стала оранжевой. Так Dreamweaver показывает, что вы начали создание вложенного шаблона.

Теперь поместите внутрь повторяющейся области изменяемую область. Назовите ее `NewsArchiveItem`. На этом работу можно считать законченной. Сохраните готовый шаблон.

Создайте на основе шаблона NewsArchive новую Web-страницу. Введите текст архивных новостей, точнее, "старостей". Сохраните ее под именем Archive.htm.

Теперь, если вы измените что-либо в шаблоне NewsArchive, будет изменена единственная страница, созданная на его основе, — Archive.htm. Если же вы измените какой-либо элемент шаблона Main, будут изменены как все стра-

ницы, созданные на его основе, так и шаблон NewsArchive. Как видите, Dreamweaver MX следит за вложенными шаблонами как за обычными Web-страницами.

Недостатки шаблонов и их преодоление

С преимуществами шаблонов мы уже знакомы. Перечислим их еще раз.

Используя шаблоны, вы не должны создавать всякий раз на каждой странице общие элементы: полосу навигации, заголовок, сведения об авторских правах и т. п. Вам даже не нужно создавать разметку для любой страницы. Все это вы можете сделать один-единственный раз, когда создаете шаблон, и потом Dreamweaver сам будет копировать эти общие элементы на каждую создаваемую на основе этого шаблона страницу. Таким образом, вы экономите время, повторно используя общие элементы ваших страниц.

Если вы что-то изменили в шаблоне, все сделанные вами изменения автоматически или по специальной команде переносятся во все созданные на основе измененного шаблона страницы. Dreamweaver сам определяет, какие страницы и какой именно код в этих страницах надо обновить, т. е. Dreamweaver берет заботы по синхронизации кода шаблона и страниц на себя.

А уж если добавить ко всему этому новые возможности, предлагаемые Dreamweaver MX, то ваши возможности по созданию шаблонов становятся практически безграничными. Необязательные и повторяющиеся области и вложенные шаблоны дают вам в руки невиданную гибкость.

Теперь сдобрим эту огромную бочку меда капелькой дегтя: упомянем о недостатках шаблонов.

Недостатков у них всего два, но не очень серьезных. Поэтому наша капля дегтя не так уж и велика.

Первый недостаток — собственно, даже не недостаток, а, как иногда говорят, "продолжение достоинств" шаблонов. Шаблон — штука очень статичная, он жестко навязывает вам определенный дизайн страниц, загоняет вас в рамки, выйти из которых очень трудно, если вообще возможно. Мы не можем произвольно менять содержимое страниц, основанных на шаблонах, т. к. шаблон, как вы помните, представляет собой сплошную нередактируемую область. Мы даже вынуждены специально указывать места, где мы хотим что-то вставить в дальнейшем.

Второй недостаток — дополнительный код HTML, добавляемый Dreamweaver для собственного удобства. Правда, этот код не слишком велик, но для "пуритан" HTML, "вылизывающих" свой код до байта, может оказаться чрезмерно большим.

Так использовать шаблоны или продолжать все делать вручную?

Вы сами должны решить этот вопрос для себя.

Когда мы начинаем делать страницы на основе шаблона, мы теряем часть своей свободы. Готовы ли вы расстаться даже с этой частью? Не окажется ли эта часть для вас слишком велика? Задумайтесь над этими вопросами перед тем, как начать работу над сайтом.

Автор рекомендует вам все же использовать шаблоны везде, где это возможно. В конце концов, Dreamweaver MX продвинулся в деле "шаблонизации" труда Web-дизайнера куда дальше предыдущих версий, и рамки, навязываемые вам шаблонами, стали не такими жесткими. К тому же, не забывайте, что не кто иной, как вы сами задаете эти рамки. И, если они станут для вас слишком тесными, вы всегда можете изменить шаблоны, добавить необязательные и повторяющиеся области, создать изменяемые атрибуты, одним словом, сделать все, чтобы вам было удобнее.

Что касается "раздувшегося" HTML-кода... Вы помните, что Dreamweaver MX поддерживает экспорт сайтов с удалением всей специальной разметки? Так что вы можете создать свой сайт на основе шаблонов, а опубликовать на Web-сервере его "очищенную" копию, не содержащую вставленный Dreamweaver HTML-код. Даже пуритане HTML должны быть этим довольны, тем более что шаблоны действительно здорово облегчают работу.

Одним словом, шаблоны — это замечательно!

Что дальше?

На этом рассказ о шаблонах заканчивается. Автору нужно еще написать хвалебную оду, посвященную Dreamweaver MX вообще и шаблонам в частности.

Итак, процесс создания Web-страниц и внесения в них изменений мы рассмотрели. А что дальше?

Представим такую ситуацию. Вы разместили на своих Web-страницах массу цитат (использовав парный тег `<CITE>`), выделили их красным цветом и отформатировали курсивом. После этого мы сохранили все страницы и опубликовали сайт на Web-сервере. И тут нам (или начальству) пришло в голову, что цитаты просто жизненно необходимо выделить еще и уменьшенным шрифтом. Как нам быть? Просматривать все страницы на предмет цитат и вручную изменять размер шрифта?

Можно ли автоматизировать и этот процесс? Нет ли в составе Dreamweaver шаблонов не для целых страниц, а для отдельных фрагментов текста? Чтобы изменить шрифт одной цитаты, а Dreamweaver переформатировал бы остальные.

Есть! И это не возможности Dreamweaver, а стандарт, принятый комитетом WWWC и поддерживаемый практически всеми современными Web-обозревателями. Это каскадные таблицы стилей CSS (Cascading Style Sheets), описанные в следующей главе.

ГЛАВА 10

Каскадные таблицы стилей

Каскадные таблицы стилей CSS — это довольно позднее нововведение. Если сам HTML появился в 1989 году, то таблицы стилей — только в 1997. Мало того, таблицы стилей не считаются частью HTML, а "гуляют сами по себе", как кошка Киплинга. И если вы зайдете на сайт WWWC (**http://www.w3c.org**), то увидите, что на описание HTML ведет одна гиперссылка, а на описание CSS — другая.

Связано это с тем, что WWW была создана учеными как средство для обмена текстовыми документами, а HTML был языком, с помощью которого создавали эти документы. Для ученых главным было содержимое документа, а не его оформление. Поэтому первые версии HTML не включали даже средств для размещения на Web-страницах графических изображений, не то что для сложного форматирования текста. Но время шло, и в Интернет пришел обыватель, тотчас потребовавший от Web-дизайнеров "сделать ему красиво". А Web-дизайнеры, в свою очередь, потребовали от разработчиков стандарта HTML средств, облегчающих им работу.

Так и возникли каскадные таблицы стилей. В настоящее время приняты спецификации CSS1 и CSS2, идет работа над CSS3. Однако современные Web-обозреватели, да и то самые последние их версии, полностью поддерживают только CSS1 и, частично, CSS2.

Но даже стандарт CSS1 таблиц стилей предлагает такое, что вы ахнете. Куда там HTML с его примитивным форматированием!

Введение в каскадные таблицы стилей

Что же представляют собой каскадные таблицы стилей? Как они могут помочь нам? Как они создаются, в конце концов? И нужно ли их использовать?

Зачем они нужны

В конце предыдущей главы был приведен пример, когда таблицы стилей могут нам здорово помочь. Вспомним его еще раз.

Вы создали сложный Web-сайт с большим количеством объемистых страниц. Текст этих страниц очень сложно отформатирован: обильно используются различные шрифты разных цветов и размеров, жирное и курсивное выделения и т. п. В частности, все цитаты выделены красным курсивным шрифтом. Сайт создан, опубликован и достаточно давно работает; посетители довольны, никаких неполадок нет, все ошибки давно исправлены, и вы о своем сайте уже, честно говоря, забыли.

Вдруг, откуда ни возьмись, — распоряжение начальства: срочно выделить цитаты на сайте еще и уменьшенным шрифтом. (Этому начальству всегда нечего делать, как давать идиотские указания...) А цитат у вас хватает на всех страницах, вдобавок они разбросаны по всему тексту так, что их нужно еще и поискать. Вспомним также, что сайт очень большой, и страницы его также велики.

Ваши действия?

Чтобы уменьшить шрифт цитаты, его необходимо выделить и выбрать в раскрывающемся списке размеров нужный пункт. Сами понимаете, это крайне трудоемко. Можно также попытаться воспользоваться мощнейшими возможностями, предлагаемыми диалоговым окном **Find and Replace**. Но, чтобы это сделать, надо правильно ввести подстроки для поиска и замены, для чего нужно долго экспериментировать. Наконец можно воспользоваться регулярными выражениями, но это вообще занятие не для слабонервных. Выходит, так и придется форматировать цитаты вручную?

Почему мы вынуждены так поступать? Чтобы объяснить это, надо немного рассказать о принципах написания и хранения программ и используемых ими данных.

Возьмем любую программу, например Microsoft Word. Данные этой программы (документ) находятся в одном файле (с расширением doc), а сама программа — в другом (исполняемый файл exe). Если нужно изменить программу (скажем, вы получили новую версию), вы просто переписываете файл программы, не трогая данные. Если вам надо изменить данные, то вы просто откроете их в программе и измените, а сама программа останется неизмененной.

В этом случае говорят, что данные и их *представление* (т. е. описание правил отображения этих данных или программа, отображающая эти данные) хранятся отдельно. Преимущество такого подхода в том, что мы можем заменить представление, не трогая сами данные, или отредактировать данные, не трогая представления. Все современные программные продукты строятся

на таком принципе; благодаря этому мы можем установить на свой компьютер новую версию программы, сохранив созданные в ней данные.

Это обычные программы. А что же HTML?

Беда HTML в том, что данные и их представление хранятся в одном файле. Более того, они представляют собой единое целое. Откройте любой HTML-файл в текстовом редакторе (например, Блокноте) и посмотрите на него. Текст перемешан с тегами форматирования самым причудливым образом. Найти какую-то нужную информацию в "сыром" HTML-коде — проблема. И еще большая проблема — изменить способ ее представления: для этого надо "перелопачивать" весь файл с риском повредить сами данные. Это очень неудобно, но что ж поделаешь — таков уж этот HTML...

И все же было бы очень неплохо создать некий набор правил форматирования различных фрагментов текста в зависимости от их назначения. Этот набор правил хранился бы в отдельном файле, независимо от содержимого Web-страницы. А в HTML-файл записать только сам текст, разбитый на логические фрагменты тегами `<P>` и `<Hn>`. Таким образом, и данные хранятся отдельно от представления, и форматированы они гораздо "прозрачнее", и нужную информацию искать значительно проще.

У такого подхода есть еще одно преимущество: становится намного легче изменять представление страниц. Скажем, если нам нужно изменить форматирование какого-либо типа фрагментов текста, мы просто корректируем соответствующие ему правила в файле правил. После этого Web-обозреватель при следующей загрузке этой страницы покажет все так, как нам надо. Таким образом, мы можем наконец-то изменить вид наших многострадальных цитат редактированием всего одного файла вместо того, чтобы править все страницы.

Хорошо? А как вам понравится, если окажется, что все это существует уже достаточно давно, с 1997 года? Да-да, существует и называется каскадными таблицами стилей.

Каскадные таблицы стилей или просто *таблицы стилей* (*CSS* — Cascading Style Sheets) — это набор правил, описывающих форматирование разных фрагментов HTML-кода и хранящихся отдельно от него. Одно такое правило, отображающее форматирование какого-то одного фрагмента или однотипной группы фрагментов кода, называется *стилем*. Таблицы стилей описываются на особом языке CSS и хранятся в особых файлах с расширением css, хотя могут быть внедрены в саму Web-страницу.

Таблицы стилей, как и многое другое, лучше представить на примере. Давайте рассмотрим небольшой фрагмент классического, "чистого" HTML-кода, созданного без использования таблиц стилей.

```
<P><FONT COLOR="#FF0000"><I>Это цитата.</I></FONT></P>
```

Как видите, это одна из тех цитат, которые по приказу нашего гипотетического начальства должны были "измельчать". Давайте же все-таки доставим начальству это удовольствие.

```
<P><FONT COLOR="#FF0000" SIZE="-1"><I>Это цитата.</I></FONT></P>
```

С одной цитатой мы справились без труда. А если их сотня? Нет, без таблицы стилей тут не обойтись. Перепишем наш пример с использованием стилей.

Сначала напишем саму таблицу стилей.

```
.cit { font-style: italic;
       color: #FF0000 }
```

Эта таблица стилей содержит определение всего одного стиля — `cit`. Такой стиль, имеющий уникальное имя (чтобы Web-обозреватель мог однозначно его идентифицировать), называется *стилевым классом*.

Этот стиль содержит определение параметров для двух *атрибутов стиля*. Первый атрибут — `font-style` — задает "стиль" текста; в нашем случае значение `italic` делает текст курсивом, словно бы его поместили внутрь тега `<I>`. Второй атрибут — `color` — задает цвет текста. Как видите, два атрибута стиля `cit` заменили оба тега: и `<FONT>`, и `<I>`.

Собственно, красивое форматирование текста таблицы стилей, которое вы видите здесь, совсем не обязательно. Ваша таблица стилей может выглядеть и следующим образом:

```
.cit {font-style:italic;color: #FF0000}
```

Так она получается даже компактнее — весьма немаловажное достоинство во времена господства медленных каналов связи. Но, согласитесь, красиво отформатированный вариант легче читается.

Сохраним нашу таблицу стилей в файле styles.css. И давайте перепишем наш фрагмент HTML-кода так, чтобы он использовал эту таблицу стилей.

```
<P CLASS="cit">Это цитата.</P>
```

Как видите, достаточно просто добавить в тег `<P>` атрибут `CLASS` и в качестве значения присвоить ему имя определенного нами стилевого класса, в нашем случае `cit`. Теперь этот фрагмент текста будет форматироваться курсивным красным шрифтом уменьшенного размера.

Но, кроме того, нам еще нужно добавить в секцию HTML-заголовка (тег `<HEAD>`) нашей страницы ссылку на таблицу стиля, чтобы Web-обозреватель смог ее найти. Эта ссылка будет иметь следующий вид:

```
<LINK REL="stylesheet" HREF="styles.css" TYPE="text/css">
```

Вот теперь точно все!

Вы уже, наверно, догадались, что таблицу стилей, сохраненную в отдельном файле, можно использовать во многих Web-страницах. Более того, она может находиться вообще на другом сайте. В частности, фирма Microsoft, насколько известно автору, предлагает бесплатно воспользоваться библиотекой таблиц стилей, помещенной на ее сайте. (Внимание: информация непроверенная!)

Кроме того, Web-страница может ссылаться одновременно на несколько таблиц стилей. Например, так:

```
<LINK REL="stylesheet" HREF="styles1.css">
<LINK REL="stylesheet" HREF="styles2.css">
```

В этом случае она будет использовать стили, определенные в обеих этих таблицах. Если же какой-то стиль определен и в одной, и в другой таблице, в действие вступает правило каскадности и приоритета, о котором будет рассказано ниже.

Но как же в этом случае уменьшить шрифт всех цитат? Для этого достаточно добавить в определение стилевого класса еще один атрибут и присвоить ему соответствующее значение. В этом случае таблица стилей будет иметь такой вид (вновь вставленный код выделен полужирным шрифтом):

```
.cit { color: #FF0000;
       font-style: italic;
       font-size: smaller }
```

Здесь мы поместили в определение стиля помещен новый атрибут `font-size`, задающий размер шрифта. И присвоили ему значение `smaller`, задающее шрифт на одну ступень меньший, чем у родительского элемента.

И это все! Никаких изменений в коде Web-страниц, ссылающихся на эту таблицу стилей, нам делать не нужно. Достаточно было только изменить файл таблицы стилей.

Удобно? Еще бы!

А еще с помощью таблицы стилей можно изменить внешний вид любого тега HTML. Для этого нужно просто переопределить его в таблице стилей следующим образом:

```
I { color: #FF0000;
    font-size: smaller }
```

После этого все фрагменты текста, заключенные внутрь тега `<I>` (курсив), будут отображаться уменьшенным шрифтом красного цвета. Такой стиль называется *стилем переопределения тега*.

А если вы создадите такой стиль:

```
H1 I { color: #FF0000;
       font-size: smaller }
```

то уменьшенным шрифтом красного цвета будет отображаться только текст, заключенный внутрь тега <I>, который, в свою очередь, находится внутри тега <H1>. Вот такой:

```
<H1><I>Курсивный</I> заголовок</H1>
```

А следующий текст:

```
<I>Обычный курсив</I>
<H2>Экспериментируем с <I>курсивом</I></H2>
```

будет отображаться как обычно.

А такой стиль:

```
I.cit { color: #FF0000;
        font-size: smaller }
```

будет применяться только к тексту, помещенному внутрь тега <I> с атрибутом CLASS, установленном в cit (своеобразный гибрид стилевого класса и переопределения тега; этот стиль так и называется — *гибридный*), т. е. к такому тексту:

```
<I CLASS="cit">Маленький зеленый курсивчик</I>
```

Как видите, тег <I> также поддерживает атрибут CLASS. Вообще, его поддерживают практически все теги, "отвечающие" за видимые элементы страницы. Так что проблем с форматированием страниц у вас возникнуть не должно.

Есть еще один способ применения стиля к элементу страницы: воспользоваться атрибутом ID, который также поддерживается почти всеми тегами.

```
<P ID="cit">Это цитата.</P>
```

Тогда определение стиля в таблице должно выглядеть следующим образом:

```
#cit { font-size: smaller;
       font-style: italic }
```

Такой стиль называется *стилем-селектором*.

Конечно, в одной таблице стилей вы можете определить одновременно несколько стилей:

```
.cit { font-size: smaller;
       font-style: italic }
I    { color: #00FF00 }
H1 I { color: #FF0000;
       font-size: larger }
```

Используя стили, можно задать для текста даже такие параметры, о которых и не мечтал "чистый" HTML. Например, можно создать рамку вокруг фраг-

мента текста или сделать так, что при наведении курсора мыши на текст он (курсор) будет менять форму. Обо всем этом будет рассказано ниже.

С помощью таблиц стилей можно форматировать не только текст. Любому элементу страницы — изображению, таблице, горизонтальной линии — может быть присвоен стилевой класс.

Например, вы можете переопределить поведение тега <BODY> таким образом:

```
BODY { background-color: #000000 }
```

Здесь мы задали черный цвет фона страницы с помощью нового атрибута background-color.

К сожалению, это справедливо в полной мере только для последних версий Web-обозревателей. В частности, в старых версиях Navigator поддержка таблиц стилей очень ограничена и относится, в основном, к тексту. Но поскольку старыми версиями этой программы вряд ли пользуется много народу, вы можете использовать таблицы стилей в своих страницах без опасения, что их "не поймут".

В составе Dreamweaver вместе со справочным руководством по HTML поставляется справочник по CSS. Чтобы вызвать его на экран, сначала вызовите панель **Reference**, выбрав пункт **Reference** в меню **Window** или нажав комбинацию клавиш <Shift>+<F1>. После этого в раскрывающемся списке

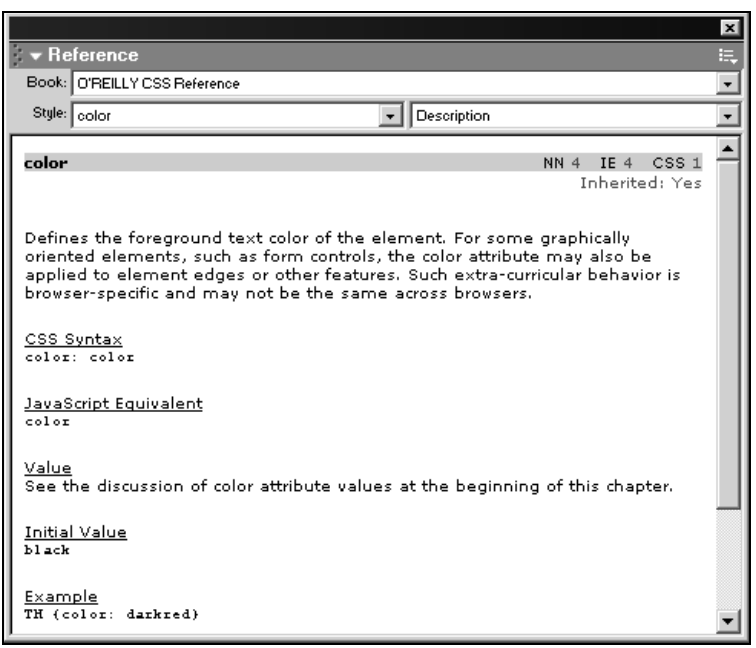


Рис. 10.1. Справка по атрибуту color в панели **Reference**

Book этой панели выберите пункт **O'REILLY CSS Reference**. Чтобы просмотреть справочную информацию по какому-либо атрибуту, достаточно выбрать его имя в раскрывающемся списке **Style**. Например, на рис. 10.1 представлена справочная информация по уже знакомому нам атрибуту `color`.

Мы познакомились с одной из двух разновидностей таблиц стилей — *внешней* или *привязанной*. О второй их разновидности, а также об альтернативном способе задания стиля для элемента страницы мы сейчас узнаем.

Три способа задания стиля

Всего стандарт CSS определяет три способа задания стиля для элемента страницы. Давайте перечислим их и дадим краткое описание каждому способу.

1. *Внешняя* (или *привязанная*) таблица стилей. Стили сохраняются в отдельном файле с расширением css и привязываются к Web-странице с помощью особого тега `<LINK>`. Вы уже познакомились с внешними таблицами стиля, так что не будем сейчас подробно на них останавливаться.
2. *Внутренняя* (или *внедренная*) таблица стилей. Таблица стилей в этом случае имеет тот же самый формат, что и внешняя, но располагается в секции заголовка той же Web-страницы и помещается внутрь специального тега `<STYLE>`.
3. *Внутренние* (также *встроенные* или *внедренные*) стили. Определение стиля помещается прямо в нужный тег, для чего используется специальный атрибут `STYLE`.

С внешними таблицами стилей вы уже познакомились. Поэтому перейдем сразу к внутренним.

Внутренняя таблица стилей по своему устройству аналогична внешней. Разница между ними в том, что внутренняя таблица стилей помещается прямо в заголовке HTML-файла. И, соответственно, может быть использована только в нем.

Давайте превратим внешнюю таблицу стилей, созданную выше, во внутреннюю. Это проще простого.

```
<STYLE>
  .cit { font-size: smaller;
         font-style: italic }
  I    { color: #00FF00 }
  H1 I { color: #FF0000;
         font-size: larger }
</STYLE>
```

Как видите, ничего радикально не изменилось. Единственное, добавился парный тег <STYLE>, определяющий таблицу стилей. Вся эта конструкция помещается в секции заголовка Web-страницы, т. е. внутри тега <HEAD>. Само собой, внутренняя таблица стилей может быть только одна на страницу.

Обращаться к внутренней таблице стилей можно так же, как и к внешней:

```
<P CLASS="cit">Это цитата.</P>
```

Вот, собственно, и все о внутренних таблицах стилей.

Внутренние стили вообще помещаются внутри тегов, определяющих тот или иной элемент страницы. Делается это с помощью атрибута STYLE, поддерживаемого, как и CLASS, практически всеми тегами.

```
<P STYLE="font-size: smaller; font-style: italic">Это цитата.</P>
```

Таким образом, пользуясь внутренними стилями, вы можете переопределить внешний вид любого элемента страницы, вообще не создавая таблицы стилей, ни внешней, ни внутренней. Внутренние стили незаменимы для создания уникальных элементов, не встречающихся больше нигде на странице.

В каких случаях следует применять тот или иной способ задания стилей? На этот случай есть простое правило: выясните, где и как часто будет использован тот или иной стиль.

Если стиль необходим во многих Web-страницах, вынесите его во внешнюю таблицу стилей. Например, если вы создали стиль основного текста страницы, который должен быть использован на всех страницах сайта, определите его в *глобальную* таблицу стилей, одну на весь сайт. Если какой-то стиль применяется не во всех страницах, а в их небольшом подмножестве, определите его во второстепенной таблице стилей, которая будет использована только данным подмножеством страниц. В этом случае вы можете определить весь набор необходимых стилей в одной таблице, а можете "разбросать" его по нескольким; во втором случае не забудьте создать ссылки на все эти таблицы.

Если стиль должен быть использован в нескольких местах одной-единственной Web-страницы, смело переносите его во внутреннюю таблицу стилей. В результате внешние таблицы стилей не будут заполняться стилями, применяемыми только в одной странице, и их файлы сохранят компактность. Можно, конечно, вынести все такие стили во внешнюю таблицу, которая будет использоваться единственной страницей, но такой подход неоптимален.

Если же стиль должен быть использован в одном-единственном месте одной-единственной Web-страницы, внедрите его прямо в HTML-тег. Так он будет "ближе к месту действия" и не станет загромождать таблицы стилей, как внутреннюю, так и внешние.

Но почему таблицы стилей называются еще и каскадными? Дело в том, что у них есть одна интересная и полезная особенность, о которой вам обязательно нужно будет узнать.

Почему "каскадные"

В одной и той же Web-странице могут использоваться и внутренние стили, и таблицы стилей обоих видов: внешние и внутренняя. При этом "в работе" будут все определенные в них стили.

Но что делать, если какой-то стиль будет определен одновременно два или три раза? Одним словом, что делать, если произойдет *конфликт стилей*?

В этом случае каскадные таблицы стилей оправдывают свое название. Вступает в действие правило каскадности, разрешающее конфликт стилей. И разрешается он столь изящно, что нельзя не восторгаться. Эх, если бы так решались все конфликты на свете!..

Давайте рассмотрим пример. Предположим, у нас есть внешняя таблица стилей.

```
P          { font-size: 9pt }
H1         { font-size: 24pt;
             text-align: center }
.copyright { font-size: 8pt;
             font-style: italic;
             text-align: right }
```

Здесь мы переопределили внешний вид текста, отформатированного тегами `<P>` и `<H1>`, и задали новый стиль `copyright`. Атрибут `text-align` задает выравнивание текста параграфа; значение `center` задает выравнивание по центру, а `right` — по правому краю.

Сохраним эту таблицу стилей в файле 10.1.css. И создадим небольшую Web-страницу.

```
<HTML>
  <HEAD>
    <TITLE>Стили</TITLE>
    <LINK REL="stylesheet" HREF="10.1.css">
    <STYLE>
      H1 { font-size: 16pt;
           color: #00FF00 }
      I  { font-size: larger }
    </STYLE>
  </HEAD>
  <BODY>
    <H1>Заголовок</H1>
```

```
    <P>Параграф 1.</P>
    <P>Параграф <I>2</I>.</P>
    <HR>
    <P CLASS="copyright">
      Авторские <SPAN STYLE="font-style: normal">права</SPAN>.
    </P>
  </BODY>
</HTML>
```

Сохраним эту страничку в файле 10.1.htm. И откроем в Web-обозревателе. То, что мы увидим в его окне, показано на рис. 10.2.

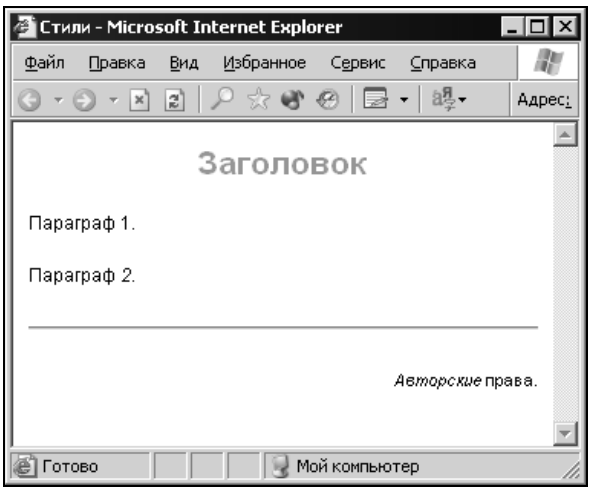


Рис. 10.2. Web-страница, использующая стили

Так что же мы получим в результате?

Во внешней таблице стилей для тега `<H1>` задаются шрифт размером 24 пункта и выравнивание текста параграфа по центру. Однако во внутренней таблице стилей для того же тега задается размер шрифта 16 пунктов и зеленый цвет текста. Какому определению стилю верить?

Web-обозреватель "верит" обоим. Точнее, делает следующее. К определению стиля, сделанному во внешней таблице, добавляет определение, сделанное во внутренней таблице. А если определение затрагивает один и тот же атрибут (в нашем случае — размер шрифта), берется определение, сделанное во внутренней таблице. И результирующий стиль тега `<H1>` будет иметь такой вид:

```
H1 { font-size: 16pt;
     text-align: center;
     color: #00FF00 }
```

Здесь автор просто перевел результирующее определение стиля, хранящееся в памяти Web-обозревателя, на понятный вам язык. Оно-то и будет применено к тексту, отформатированному с использованием тега <H1>.

Говорят, что стиль, определенный во внутренней таблице, имеет *приоритет* перед стилем из внешней таблицы. Аналогично, внутренний стиль, помещенный внутрь тега, имеет приоритет перед и внутренней, и внешней таблицами стилей. Это и называется *каскадностью*.

Давайте рассмотрим стиль copyright.

Во внешней таблице для него были заданы курсивный шрифт размером 8 пунктов и выравнивание по правому краю. Когда мы присвоили его элементу, отформатированному с помощью тега <P>, стиль copyright полностью заменил определение стиля для тега <P>, сделанное в той же внешней таблице. Из этого следует, что стили с именем (т. е. стилевые классы и гибридные стили) имеют приоритет перед переопределениями тегов.

Теперь рассмотрим подробнее последний параграф нашей странички. Вот он:

```
<P CLASS="copyright">
  Авторские <SPAN STYLE="font-style: normal">права</SPAN>.
</P>
```

Здесь присутствует новый, не знакомый нам тег <SPAN>. Этот тег позволяет выделить фрагмент какого-либо текста и произвести над ним некоторые действия, например применить стиль. Что мы и сделали — применили к фрагменту текста новый стиль, встроенный в этот тег. Мы убрали курсивное выделение текста, и наши установки перекрыли определение стиля copyright, сделанное во внешней таблице стилей.

Каскадность — огромное преимущество. Благодаря ему мы можем сократить размеры своих таблиц стилей до минимума. Нам достаточно будет просто переопределить необходимые атрибуты во внутренней таблице стилей или во внутреннем стиле, чтобы изменить Web-страницу до неузнаваемости. Web-обозреватель однозначно разрешает все конфликты стилей, пользуясь правилом каскадности и приоритета. А правило это звучит просто: своя рубашка, т. е. таблица стилей, ближе к телу... простите, тегу.

Однако для неопытного Web-дизайнера каскадность — бич божий. Иной раз бывает трудно сразу выяснить, что перед чем имеет приоритет. И с таким трудом созданные Web-страницы выглядят как кошмар умалишенного. В таком случае помогают только тщательные проверки и перепроверки различных вариантов определения стилей.

Псевдостили гиперссылок

А теперь мы узнаем еще об одной интересной возможности, предлагаемой Web-дизайнеру таблицами стилей. Это *псевдостили гиперссылок*, позволяю-

щие в достаточно широких пределах менять внешний вид гиперссылок на ваших страницах.

Вы уже знаете, что для каждой Web-страницы можно задать цвет фона, текста и кодировку текста. Кроме того, HTML позволяет вам также сменить цвета гиперссылок, отдельно — для непосещенных, отдельно — для посещенных и отдельно — для активной гиперссылки. Это позволяет немного оживить Web-сайт, сделать его не похожим на другие. Но возможности, предлагаемые псевдостилями, несравнимо шире. Вы можете менять шрифт, начертание и даже убирать надоевшее подчеркивание у активной гиперссылки.

Но как же это делается? Обычным переопределением стилей во внешней или внутренней таблице.

Но сначала давайте перечислим все доступные в данный момент псевдостили. Как видно из табл. 10.1, всего их четыре.

Таблица 10.1. *Псевдостили гиперссылок*

Псевдостиль	Описание
A: link	Применяется ко всем гиперссылкам документа
A: active	Применяется ко всем активным гиперссылкам документа
A: visited	Применяется ко всем посещенным гиперссылкам документа
A: hover	Применяется к гиперссылке, на которую указывает курсор мыши

Теперь давайте создадим небольшой пример Web-странички, использующей псевдостили.

```
<HTML>
  <HEAD>
    <TITLE>Псевдостили</TITLE>
    <STYLE>
      A: link    { color: #CC0000;
                   background-color: #FFFFFF;
                   text-decoration: none }
      A: active  { color: #FFFFFF;
                   background-color: #CC0000;
                   text-decoration: none }
      A: visited { color: #CC0000;
                   background-color: #FFFFFF;
                   text-decoration: line-through }
      A: hover   { color: #FFFFFF;
                   background-color: #CC0000;
                   text-decoration: none }
```

```
    </STYLE>
  </HEAD>
  <BODY>
    <P><A HREF="">Гиперссылка 1</A></P>
    <P><A HREF="">Гиперссылка 2</A></P>
    <P><A HREF="">Гиперссылка 3</A></P>
  </BODY>
</HTML>
```

Сохраните эту страницу в файле 10.2.htm и откройте в Web-обозревателе. Попробуйте провести курсором мыши над любой гиперссылкой и посмотрите, что из этого получится.

Автор не будет подробно разбирать таблицу стилей и ее атрибуты. Попробуйте разобраться в ней сами. Учтите только, что атрибут `background-color` задает цвет фона текста, а атрибут `text-decoration` — подчеркивание или зачеркивание текста. Значение `line-through` последнего задает зачеркивание текста, а `none` — отсутствие подобного "украшения".

Вообще, псевдостили лучше всего задавать во внешней таблице стилей, поскольку все таблицы сайта должны выглядеть одинаково. Тем более это касается гиперссылок. Представьте себе удивление и гнев посетителя, если на различных страницах гиперссылки отображаются разным цветом.

Работа с таблицами стилей в Dreamweaver

Вы уже много узнали о таблицах стилей. И, наверно, вас уже давно мучает вопрос: а что же Dreamweaver? Поддерживает ли он работу с таблицами стилей? И если поддерживает, то насколько? И почему автор не упомянул об этом ни слова, а все заваливал нас примерами на этом ужасном и ни капельки не понятном HTML? Неужели нам все-таки придется создавать стили вручную?!!

Успокойтесь. Все в порядке. Dreamweaver прекрасно поддерживает и внешние, и внутренние таблицы стилей и позволяет вам создавать стили, "понятные" самым последним на момент его выхода в свет Web-обозревателям. А применять созданные стили к элементам страницы вообще проще простого.

Но зачем же понадобилось столько исходного кода? Для того чтобы вы поняли, что такое таблицы стилей и как они работают. А разъяснить это проще всего на примерах, написанных на языках HTML и CSS. К тому же, знать языки HTML и CSS Web-дизайнеру совсем не вредно, наоборот, полезно. Лишние знания — багаж необременительный и зачастую оказывающийся совсем не лишним.

В этом есть, правда, еще один резон. Дело в том, что Dreamweaver не поддерживает удобное создание и присвоение элементам страницы встроенных стилей. Единственный способ создавать их — вводить вручную в минередакторе HTML или пользоваться диалоговым окном **Tag Editor**, описанным в *главе 2*.

Создание стилей

Перед тем как начать экспериментировать со стилями, откройте Web-страницу 1.1.htm, созданную нами в самом начале изучения Dreamweaver. Над ее содержимым мы и будем издеваться. Только пересохраните ее под именем 10.2.htm, выбрав пункт **Save As** в меню **File** или нажав комбинацию клавиш <Ctrl>+<Shift>+<S>.

Вся работа со стилями протекает в панели **CSS Styles** Dreamweaver, показанной на рис. 10.3. Чтобы вызвать ее на экран, либо выберите в меню **Window** пункт **CSS Styles**, либо нажмите комбинацию клавиш <Shift>+<F11>.

Рис. 10.3. Панель **CSS Styles**

Чтобы создать новый стиль, выберите пункт **New CSS Style** контекстного или дополнительного меню либо нажмите одноименную кнопку (рис. 10.4), расположенную в правом нижнем углу панели **CSS Styles**. Также вы можете активизировать окно документа и выбрать пункт **New style** подменю **CSS Styles** контекстного меню или одноименный пункт одноименного подменю, находящегося в меню **Text**. В результате любого из этих действий на экране появится диалоговое окно **New CSS Style**, показанное на рис. 10.5.

Рис. 10.4. Кнопка **New CSS Style**

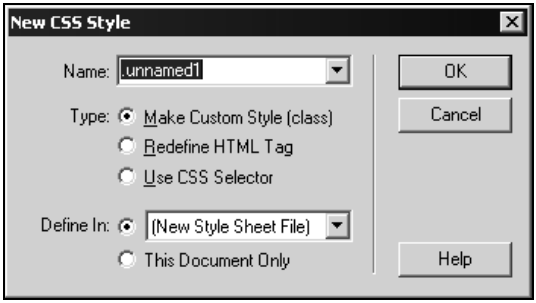


Рис. 10.5. Диалоговое окно **New CSS Style**

Группа переключателей **Type** задает элементы страницы, к которым будет применен вновь создаваемый стиль:

- переключатель **Make Custom Style (class)** позволяет создать новый стилевой класс. Если вы его включите, то в комбинированном списке **Name** сможете ввести имя создаваемого стилевого класса, например `.cit`;
- переключатель **Redefine HTML Tag** позволяет создать стиль переопределения тега. Если выбран этот переключатель, в раскрывающемся списке **Tag** вы сможете выбрать нужный тег;
- переключатель **Use CSS Selector** позволяет создать стиль-селектор. Он же позволит вам создать псевдостиль гиперссылок или комбинированный стиль наподобие `H1 I` или `I.cit`. Если включен этот переключатель, в комбинированном списке **Selector** может быть введено имя создаваемого комбинированного стиля или селектора. Развернув этот список, вы сможете выбрать нужный псевдостиль.

Вторая группа переключателей **Define In** задает, в какой таблице стилей будет находиться вновь создаваемый стиль:

- если выбран верхний переключатель, в расположенном правее раскрывающемся списке вы сможете выбрать имя файла внешней таблицы стилей, привязанной к этой странице. Также вы можете выбрать пункт **(New Style Sheet File)**, чтобы создать новую таблицу стилей. В последнем случае после нажатия кнопки **OK** на экране появится диалоговое окно сохранения файла, где вы сможете ввести имя файла новой таблицы стилей;
- если выбран переключатель **This Document Only**, стиль будет помещен во внутреннюю таблицу стилей.

Введя нужные данные, нажмите кнопку **OK** для создания стиля или **Cancel** — для отказа от этого.

Давайте предположим, что мы хотим переопределить стиль тега `<P>` и поместить определение этого стиля в новую таблицу стилей, которую затем привяжем к новой Web-странице 10.2.htm. Для этого сначала выберите пе-

реключатель **Redefine HTML Tag**, затем — пункт **p** в списке **Tag**, далее — верхний переключатель в группе **Define In** и нажмите кнопку **OK**. В диалоговом окне сохранения файла введите имя файла таблицы стилей 10.2.css и нажмите кнопку сохранения. Все, сам файл таблицы стилей нами создан.

После создания таблицы стилей на экране появится диалоговое окно **CSS Style Definition**. Это диалоговое окно очень велико и состоит из множества вкладок, которые мы рассмотрим последовательно, одна за другой. Эти вкладки выбираются в списке **Category**. Вкладка **Type** этого окна, где задаются параметры шрифта текста, представлена на рис. 10.6.

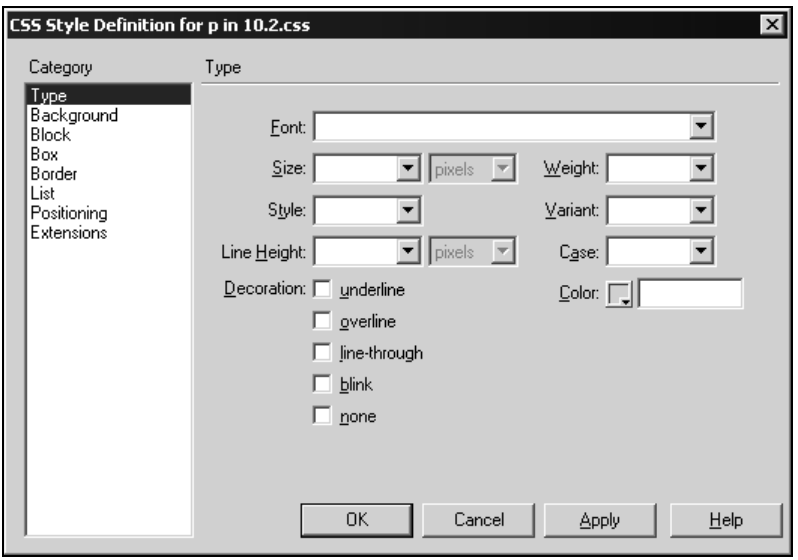


Рис. 10.6. Вкладка **Type** диалогового окна **CSS Style Definition**

Определение стиля

А теперь мы подробно рассмотрим, что же предлагает нам Dreamweaver для создания нового стиля.

Однако прежде чем мы начнем изучать возможности Dreamweaver по созданию стилей, нужно сказать следующее. При создании стилей вам совсем не обязательно задавать все их атрибуты. Задайте только те, что реально вам необходимы. Незаданные же атрибуты автоматически принимают значение по умолчанию, как правило, `normal` или `none`. Таким образом, чтобы вернуть атрибуту значение по умолчанию, просто удалите введенное значение.

И еще. Впервые с начала книги автор нарушит свою традицию помещать в примечаниях описание тегов HTML и атрибутов CSS, с помощью которых

достигается тот или иной эффект. Дело в том, что атрибутов стилей, которые здесь будут описаны, довольно много, и примечания получатся слишком большими. Если вам понадобится выяснить синтаксис какого-то атрибута CSS, обратитесь к электронному руководству, поставляемому в составе Dreamweaver. Как его вызвать, вы уже знаете.

Внимание!

Далеко не все программы Web-обозревателей поддерживают все атрибуты CSS. Поэтому перед тем, как применить тот или иной атрибут, справьтесь в руководстве по CSS, поставляемым с Dreamweaver MX.

Внимание!

Dreamweaver не всегда корректно отображает элементы страницы, к которым было применено специальное CSS-форматирование. Чтобы увидеть правильные результаты использования того или иного атрибута стилей, воспользуйтесь Web-обозревателем.

Параметры шрифта

Взгляните еще раз на рис. 10.6. На нем, как вы помните, показана вкладка **Type** диалогового окна **CSS Style Definition**. На этой вкладке задаются параметры шрифта текста.

Сам шрифт выбирается в комбинированном списке **Font**. Вы можете выбрать один из шрифтов либо ввести его имя вручную. Правда, при этом вы должны быть уверены, что этот шрифт гарантированно установлен на компьютерах пользователей, или специально позаботиться об этом. (Например, распространить файл шрифта среди всех потенциальных посетителей вашего Web-сайта.)

В комбинированном списке **Size** задается размер шрифта. На рис. 10.7 этот список показан в раскрытом виде. Вы можете либо ввести числовое значение в одной из поддерживаемых CSS величин, либо выбрать в списке предопределенное, абсолютное или относительное.

В общем, в этом комбинированном списке можно найти три вида пунктов:

- пункты — числовые значения. Задают размер шрифта в одной из поддерживаемых CSS единиц измерения;
- пункты **xx-small**, **x-small**, **small**, **medium**, **large**, **x-large** и **xx-large**, задающие одно из предопределенных значений размера шрифта;
- пункты **larger** и **smaller**, задающие относительный размер шрифта. Пункт **larger** увеличивает шрифт на одну позицию в списке предопределенных значений, а пункт **smaller** — уменьшает.

Как уже говорилось, вы можете также вводить нужные числовые значения прямо в этом списке.

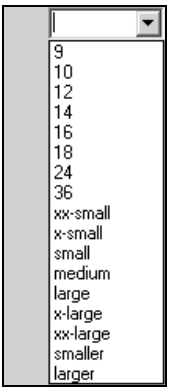


Рис. 10.7. Комбинированный список **Size**, задающий размер шрифта

В раскрывающемся списке, расположенном правее **Size**, выбирается единица измерения размера шрифта. Этот список доступен только тогда, когда в **Size** выбрано или введено числовое значение. Всего здесь доступно девять пунктов, представляющих девять *единиц измерения*, поддерживаемых стандартом CSS. Все они перечислены и описаны в табл. 10.2.

Таблица 10.2. *Единицы измерения, поддерживаемые стандартом CSS*

Единица измерения	Описание
pixels	Пикселы
points	Пункты
in	Дюймы
cm	Сантиметры
mm	Миллиметры
picas	Пики
ems	Размер буквы "m" текущего шрифта
exs	Размер буквы "x" текущего шрифта
%	Проценты от размера шрифта родительского элемента

В комбинированном списке **Weight** задается величина "жирности" шрифта. Вы можете ввести числовое значение "жирности" от 100 до 900, либо выбрать одно из предопределенных абсолютных или относительных значений. Как и в списке **Size**, здесь доступны пункты трех типов:

- пункты — числовые значения. Задают "жирность" шрифта в числах от 100 до 900. Обычный шрифт имеет значение 400, жирный — 700;

- пункты **normal** и **bold**, задающие соответственно обычный и жирный шрифт;
- пункты **bolder** и **lighter**, задающие относительную "жирность" шрифта. Пункт **bolder** увеличивает "жирность" на одну позицию в списке предопределенных значений, а пункт **lighter** — уменьшает.

В раскрывающемся списке **Style** вы можете выбрать "стиль" шрифта, а именно, будет ли он являться курсивом или нет. Доступны три пункта: **normal** — обычный шрифт, **italic** — курсив и **oblique** — зависит от конкретной программы Web-обозревателя, но обычно тоже курсив.

В раскрывающемся списке **Variant** задается вид малых букв шрифта. Пункт **normal** задает нормальный их вид, а **small-caps** делает их похожими на уменьшенные заглавные буквы.

В комбинированном списке **Line Height** задается вертикальный размер строки текста. Значение **normal** задает размер по умолчанию, вычисляемый самим Web-обозревателем. Чтобы задать свой размер, введите его вручную и выберите единицу измерения. Раскрывающийся список, задающий единицу измерения, находится правее. Если вы его откроете, то увидите, что он имеет новый пункт **multiple**; он задает умножение размера по умолчанию на введенное вами число.

В раскрывающемся списке **Case** задается вид текста: будет ли он отображаться только большими или только маленькими буквами. Здесь доступны четыре пункта:

- **capitalize** — каждое слово текста отображается с большой буквы;
- **uppercase** — текст отображается только большими буквами;
- **lowercase** — только маленькими буквами;
- **none** — текст отображается как он набран, без всяческих преобразований (поведение по умолчанию).

Группа флажков **Decoration** задает дополнительное "украшение" текста: подчеркивание, надчеркивание или зачеркивание. Здесь доступно пять флажков:

- **underline** — подчеркивает текст (поведение по умолчанию для гиперссылок);
- **overline** — надчеркивает текст;
- **line-through** — зачеркивает текст;
- **blink** — заставляет текст мерцать (поддерживается только старыми версиями Navigator);
- **none** — убирает все эти "украшения" (поведение по умолчанию для остального текста).

Селектор цветов **Color** задает цвет текста.

Параметры фона

Параметры фона элемента страницы задаются на вкладке **Background** диалогового окна **CSS Style Definition**. Эта вкладка показана на рис. 10.8.

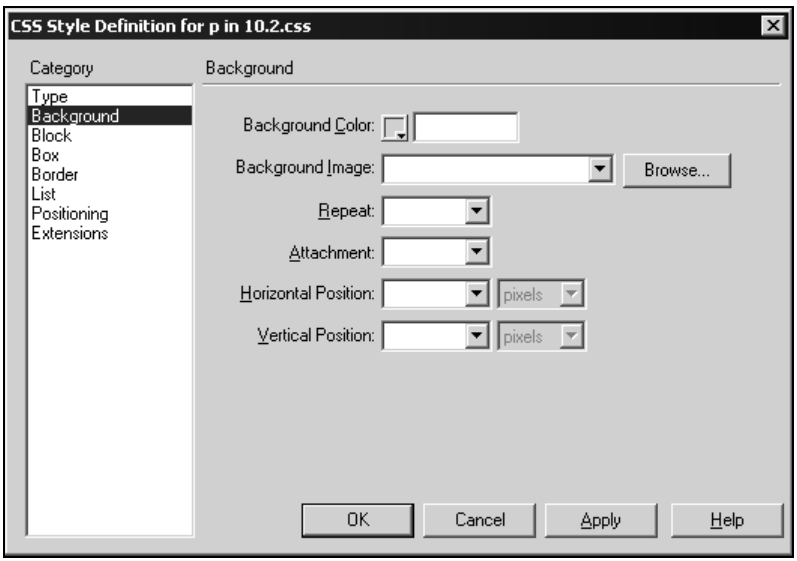


Рис. 10.8. Вкладка **Background** диалогового окна **CSS Style Definition**

Селектор цветов **Background Color** задает цвет фона. Значение по умолчанию зависит от программы Web-обозревателя.

В комбинированном списке **Background Image** задается графическое изображение, которое будет использоваться в качестве графического фона страницы. Вы можете ввести имя нужного файла или выбрать пункт **none**, убирающий графический фон. Если вы не хотите вводить имя файла вручную, щелкните кнопку **Browse** и выберите нужный файл в диалоговом окне **Select File**.

Обычно графическое изображение, использующееся в качестве фона, значительно меньше в размерах, чем сама страница. В этом случае Web-обозреватель многократно повторяет это изображение, выкладывая своего рода "мозаику", занимающую всю страницу. Управлять тем, как будет повторяться это изображение, можно с помощью раскрывающегося списка **Repeat**. Здесь доступны четыре пункта:

- **no-repeat** — отключает повторение. Изображение помещается на страницу всего один раз, в левом верхнем углу элемента;
- **repeat** — включает повторение изображения по горизонтали и вертикали (поведение по умолчанию);

- **repeat-x** — включает повторение изображения только по горизонтали;
- **repeat-y** — устанавливает повторение изображения только по вертикали.

Когда вы прокручиваете содержимое окна Web-обозревателя, то вместе с содержимым Web-страницы прокручивается и графический фон, если он есть. Некоторые Web-обозреватели поддерживают одну забавную возможность: можно запретить прокрутку графического фона, зафиксировав его на месте. Это может придать странице некоторую оригинальность, однако при прокрутке отнимет больше ресурсов компьютера. И сделать это можно с помощью раскрывающегося списка **Attachment**. Пункт **scroll** этого списка заставляет графический фон прокручиваться (поведение по умолчанию), а **fixed** — фиксирует его на месте.

Также некоторые Web-обозреватели позволяют вам поместить графический фон в нужное место страницы. Для этого предназначены раскрывающиеся списки **Horizontal Position** и **Vertical Position**. Вы можете ввести в них числовые значения соответственно горизонтальной и вертикальной координат графического фона. В раскрывающихся списках, расположенных правее, выбирается одна из единиц измерения, поддерживаемых CSS.

Кроме того, в раскрывающемся списке **Horizontal Position** доступны три пункта:

- **left** — выравнивание графического фона по левому краю элемента страницы (поведение по умолчанию);
- **center** — выравнивание по центру;
- **right** — выравнивание по правому краю.

В раскрывающемся списке **Vertical Position** доступны также три пункта:

- **top** — выравнивание графического фона по верху элемента страницы (поведение по умолчанию);
- **center** — выравнивание по центру;
- **bottom** — выравнивание по низу.

Очевидно, что задавать эти параметры нужно, предварительно задав соответствующие параметры других атрибутов стиля, перечисленных на вкладке **Background**. А для достижения необходимого результата наверняка придется поэкспериментировать.

Параметры абзаца

Параметры текстового абзаца задаются на вкладке **Block** диалогового окна **CSS Style Definition**. Здесь же задаются параметры, применяемые к нетекстовым элементам страницы, таким как графические изображения и таблицы. Эта вкладка показана на рис. 10.9.

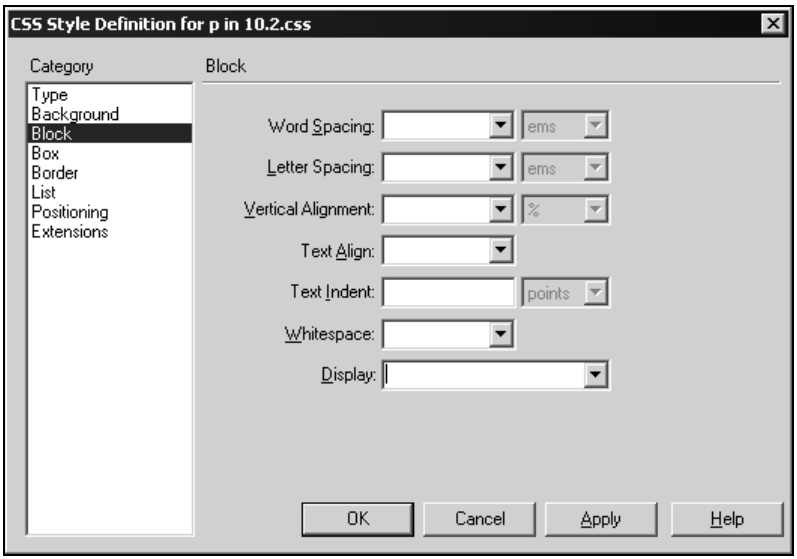


Рис. 10.9. Вкладка **Block** диалогового окна **CSS Style Definition**

В комбинированном списке **Word Spacing** задается величина дополнительного пространства, помещаемого между словами. Введите нужную величину и выберите в расположенном правее раскрывающемся списке единицу измерения. Здесь доступны все единицы измерения, кроме процентов. Пункт **normal** комбинированного списка **Word Spacing** убирает дополнительное расстояние между словами текста.

В комбинированном списке **Letter Spacing** задается величина дополнительного пространства, помещаемого между символами. Введите нужную величину и выберите в расположенном правее раскрывающемся списке единицу измерения. Здесь доступны все единицы измерения, кроме процентов. Пункт **normal** комбинированного списка убирает дополнительное расстояние между символами текста.

Комбинированный список **Vertical Alignment** задает вертикальное выравнивание элемента страницы относительно родителя. Здесь доступны следующие пункты:

- **baseline** — выравнивание базовой линии элемента страницы по базовой линии текста родителя (поведение по умолчанию);
- **sub** — помещает текст ниже остального текста, имитируя нижний индекс;
- **super** — помещает текст выше остального текста, имитируя верхний индекс;
- **top** — выравнивание элемента страницы по верхнему краю родительского элемента;

- **text-top** — выравнивает верхний край элемента страницы по верхнему краю текста родителя;
- **middle** — помещает середину элемента страницы на половину высоты шрифта выше базовой линии текста родителя;
- **bottom** — выравнивание элемента страницы по нижнему краю родительского элемента:
- **text-bottom** — выравнивает нижний край элемента страницы по нижнему краю текста родителя.

Кроме того, вы можете ввести в этот комбинированный список процентное значение, задающее, насколько выше или ниже базовой линии родителя находится базовая линия элемента страницы.

Как бы то ни было, для того чтобы получить в окне Web-обозревателя необходимый результат, вам придется поэкспериментировать с разными значениями этого атрибута. Рассказать, как будет выглядеть тот или иной элемент страницы при том или ином значении, невозможно — его нужно видеть своими глазами. (Это, впрочем, справедливо и для многих других параметров элементов страниц, задаваемых в Dreamweaver.)

Раскрывающийся список **Text Align** задает горизонтальное выравнивание текста в абзаце. Здесь доступны четыре пункта:

- **left** — выравнивание по левому краю (поведение по умолчанию);
- **center** — по центру;
- **right** — по правому краю;
- **justify** — по ширине.

В поле ввода **Text Indent** задается величина отступа красной строки. Эта величина может быть как положительной, так и отрицательной (выступ). В раскрывающемся списке, расположенном правее, выбирается единица измерения.

В раскрывающемся списке **Whitespace** задается поведение пробелов внутри текста абзаца. Здесь доступны три пункта:

- **normal** — задает обычное отображение текста, т. е. Web-обозреватель может переносить строки по пробелам, если они (строки) не будут помещаться в родительском элементе;
- **pre** — превращает текст абзаца в фиксированно отформатированный. Как вы помните из *главы 5*, такой текст отображается так, как он введен в исходном HTML-коде, с учетом пробелов, возвратов каретки и т. п. Одним словом, текст ведет себя так, словно заключен внутрь тега `<PRE>`;
- **nowrap** — запрещает перенос текста по пробелам. Вам будет необходимо поставить в нужные места строк теги разрыва текста `<BR>`, если вы хоти-

те, чтобы они все-таки переносились. Это может быть полезно, если вы захотите контролировать, как текст будет переноситься на другую строку.

Раскрывающийся список **Display** задает *поведение* элемента страницы. В нем доступно очень много пунктов:

- **none** — элемент вообще не будет присутствовать на странице, словно он и не задан в ее коде;
- **inline** — элемент страницы ведет себя как отдельный символ текста *(встроенный элемент)*;
- **block** — элемент страницы ведет себя как абзац текста *(блочный элемент)*;
- **list-item** — элемент страницы ведет себя как пункт списка;
- **run-in** — *встраивающийся* элемент. Если за таким элементом идет блочный элемент, он становится первым символом блочного элемента, в противном случае он сам становится блочным элементом;
- **compact** — *компактный* элемент. Если за таким элементом идет блочный элемент, он форматируется как однострочный встроенный элемент и помещается на левой границе блочного элемента (если эта граница задана). В противном случае он сам форматируется как блочный элемент;
- **marker** — маркер списка;
- **table** — таблица;
- **inline-table** — таблица, отформатированная как встроенный элемент;
- **table-row-group** — строки секции тела таблицы;
- **table-header-group** — строки секции заголовка таблицы;
- **table-footer-group** — строки секции "поддона" таблицы;
- **table-row** — строка таблицы;
- **table-column-group** — группа колонок таблицы;
- **table-column** — колонка таблицы;
- **table-cell** — ячейка таблицы;
- **table-caption** — заголовок таблицы.

Параметры размеров и размещения

Параметры, задающие размеры и размещение элемента страницы, задаются на вкладке **Box** диалогового окна **CSS Style Definition**. Эта вкладка показана на рис. 10.10.

Ширину и высоту элемента страницы вы можете задать в комбинированных списках **Width** и **Height**. Эти размеры могут быть заданы как абсолютные величины или процентное значение от ширины или высоты родительского элемента (например, самой страницы). Единица измерения, как обычно,

выбирается в раскрывающемся списке, расположенном справа от соответствующего комбинированного списка. Чтобы вернуть Web-обозревателю управление размерами элемента страницы, выберите значение `auto` или вообще очистите соответствующий комбинированный список.

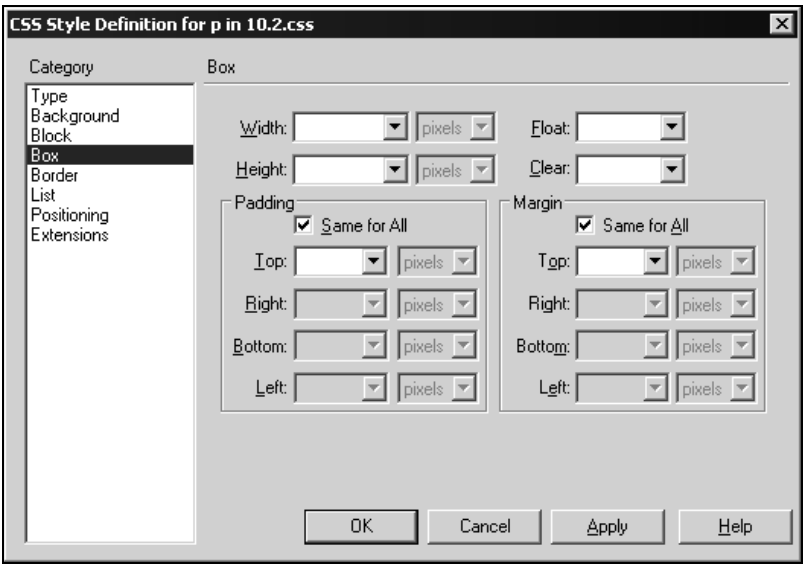


Рис. 10.10. Вкладка **Box** диалогового окна **CSS Style Definition**

Воспользовавшись раскрывающимся списком **Float**, вы до какой-то степени сможете "освободить" элемент страницы, вынеся его за пределы пресловутого "потока" текста. В этом списке доступны три пункта:

- **left** — "прижимает" элемент к левому краю страницы. Все остальное содержимое страницы "обтекает" "освобожденный" элемент;
- **right** — "прижимает" элемент к правому краю страницы;
- **none** — оставляет элемент в "потоке" текста (поведение по умолчанию).

На рис. 10.11 показано графическое изображение, вынесенное за пределы "потока" текста и выровненное по правому краю (атрибут **Float** был установлен в **right**).

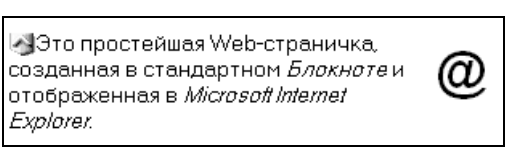


Рис. 10.11. Графическое изображение, выровненное по правому краю

Вы также можете задать, будет ли текущий элемент страницы отображаться с той же стороны и в той же горизонтальной позиции, что и соседствующий с ним "освобожденный" элемент. Задается это с помощью раскрывающегося списка **Clear**, причем значение этого атрибута должно быть одинаковым у обоих элементов.

В списке **Clear** доступны четыре пункта:

- **left** — если "освобожденный" и текущий элементы должны находиться слева, то текущий элемент перемещается ниже "освобожденного";
- **right** — то же самое, только справа;
- **both** — то же самое, с обеих сторон;
- **none** — разрешает "освобожденному" и текущему элементам занять одну и ту же горизонтальную позицию (поведение по умолчанию).

На рис. 10.12 показано уже знакомое нам графическое изображение, выровненное по правому краю. Внизу вы видите текстовый абзац, находящийся в другой горизонтальной позиции. Значения атрибута **Clear** и у изображения, и у абзаца должны быть установлены в **right**, иначе ничего не получится.

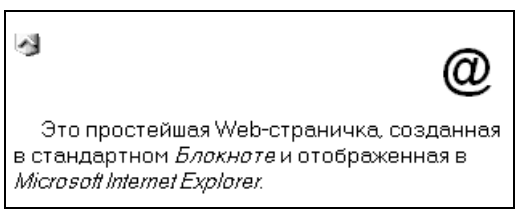


Рис. 10.12. Графическое изображение и текстовый абзац, размещенные в разных горизонтальных позициях

Вам придется поэкспериментировать с этими двумя атрибутами стиля, чтобы получить нужный результат. Слишком много тут всяких нюансов, которые полностью никогда не опишешь и не всегда учтешь.

А теперь мы с вами подходим к самому интересному.

Воспользовавшись особыми атрибутами стиля, вы можете задать *отступы* от границы элемента до содержащегося в нем текста. Это позволит вам обойтись без громоздких таблиц. Посмотрите, например, на рис. 10.13 — там для текста в абзаце заданы сантиметровые отступы слева и справа.

Величины отступов задаются в группе элементов управления **Padding**. В нее входят раскрывающиеся списки **Top** (задает отступ сверху), **Right** (справа), **Bottom** (снизу) и **Left** (слева). Единицы измерения выбираются, как обычно, в раскрывающихся списках, расположенных справа. Учтите, что доступны только абсолютные единицы — процентов нет.

Группа элементов управления **Margin** аналогична группе **Padding** хотя бы тем, что в нее также входят раскрывающиеся списки **Top** (задает величину

отступа сверху), **Right** (справа), **Bottom** (снизу) и **Left** (слева). Однако, в отличие от **Padding**, списки этой группы задают расстояние между границей элемента и другими элементами страницы. Единицы измерения выбираются также в раскрывающихся списках справа. В отличие от атрибутов группы **Padding**, здесь доступны относительные единицы измерения — проценты, т. е. вы можете задать отступ как процент от соответствующего размера родителя.

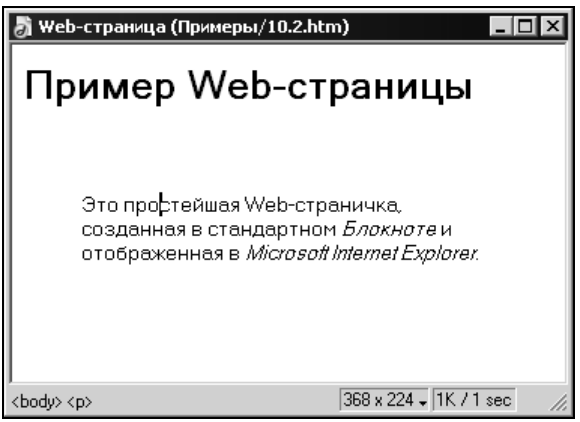


Рис. 10.13. Текст с отступами, заданными атрибутами стилей

Обе этих группы создают на странице свободное пространство. Но, если свободное пространство, полученное с помощью элементов группы **Padding**, принадлежит текущему элементу страницы, то созданное с помощью группы **Margin** — не принадлежит ему, а точнее, принадлежит самой странице. Это можно использовать для получения различного рода эффектов, наподобие тех, что изображены на рис. 10.14 и 10.15.

На рис. 10.14 изображен текстовый абзац, расположенный на странице с отступами слева и справа, равными 1 см. Эти отступы были созданы с помощью элементов управления группы **Margin**. Сравните его с рис. 10.15, где эти же отступы были созданы с помощью группы **Padding**. (Для вашего удобства для этого абзаца был задан светло-серый фон.) Как видите, в первом случае создаются отступы от границ абзаца до границ родителя (в нашем случае — страницы), т. е. создаваемое свободное пространство не является частью абзаца. Во втором же случае создаваемые отступы находятся внутри абзаца, между его границами и его же содержимым; и свободное пространство, как видите, принадлежит самому абзацу.

Вы можете комбинировать эти два способа создания свободного пространства. Взгляните, например, на рис. 10.16. И попробуйте догадаться, какие атрибуты стиля были заданы в этом случае.

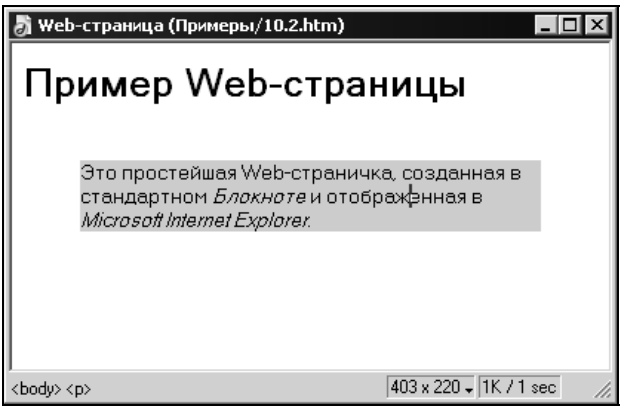


Рис. 10.14. Текстовый абзац с отступами, созданными с помощью группы атрибутов **Margin**

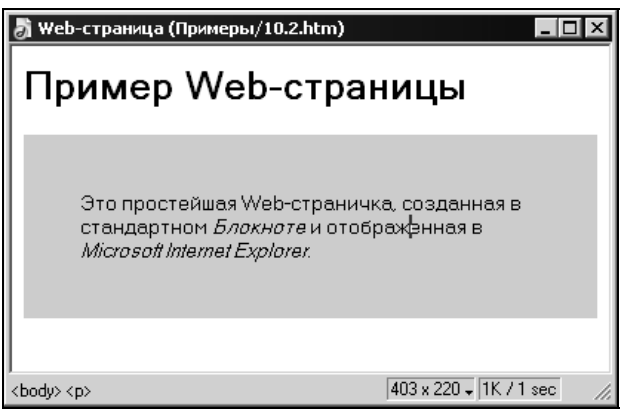


Рис. 10.15. Текстовый абзац с отступами, созданными с помощью группы атрибутов **Padding**

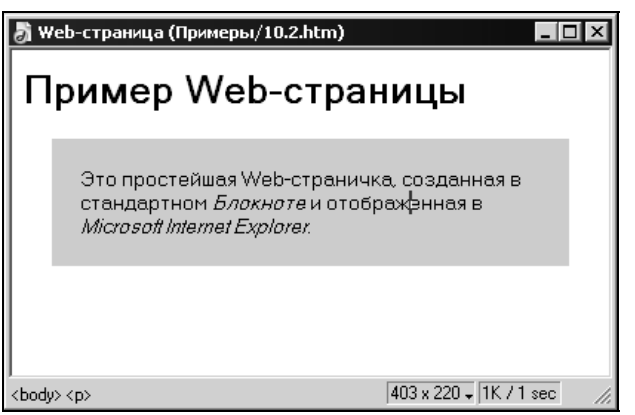


Рис. 10.16. Текстовый абзац с отступами, созданными с помощью групп атрибутов **Padding** и **Margin**

Если вы хотите задать одинаковые отступы со всех сторон, то можете включить флажок **Same for All**, находящийся в соответствующей группе элементов управления. После этого введите нужную величину отступа в комбинированный список **Top** (остальные списки станут недоступными). Флажок **Same for All** включен по умолчанию, т. к. отступы для элемента страницы, как правило, задаются одинаковыми со всех сторон.

Пользуясь атрибутами стиля, задаваемыми на вкладке **Box** диалогового окна **CSS Style Definition** (см. рис. 10.10), вы можете в некоторой степени обойтись без таблиц. В частности, если вы используете таблицы для задания отступа или отступов вокруг текста (иначе говоря, просветов), то теперь можете от них отказаться. В самом деле, атрибуты стилей **Margin** и **Padding** предпочтительнее, так они как не обладают недостатками, в изобилии присущими таблицам.

Параметры рамки

Параметры, определяющие цвет и толщину *рамки* вокруг элемента страницы, задаются на вкладке **Border** диалогового окна **CSS Style Definition**. Эта вкладка показана на рис. 10.17.

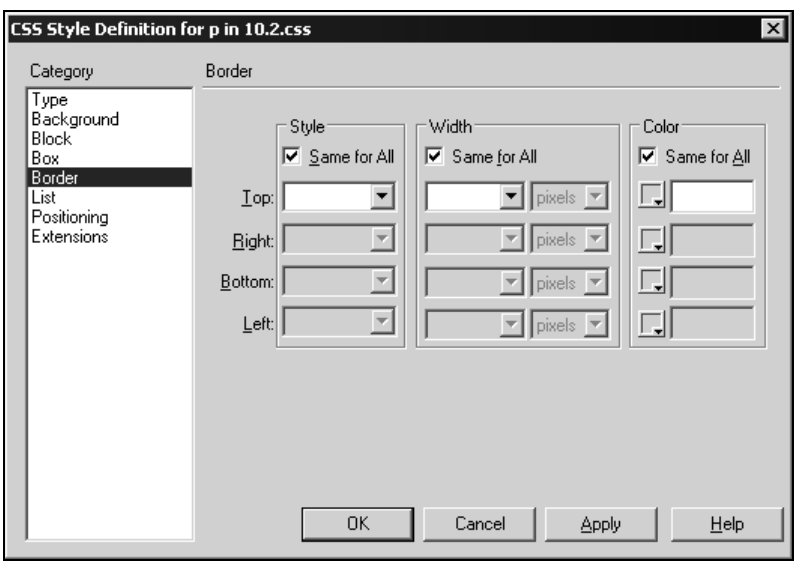


Рис. 10.17. Вкладка **Border** диалогового окна **CSS Style Definition**

Рамка рисуется по воображаемой границе элемента страницы (рис. 10.18). Значит, вы можете задать расстояние от границы до содержимого элемента, используя атрибуты группы **Padding**. А расстояние от границы до "соседей" этого элемента задается атрибутами **Margin**. Эти атрибуты задаются на вкладке **Box** диалогового окна **CSS Style Definition** (подробнее *см. выше*).

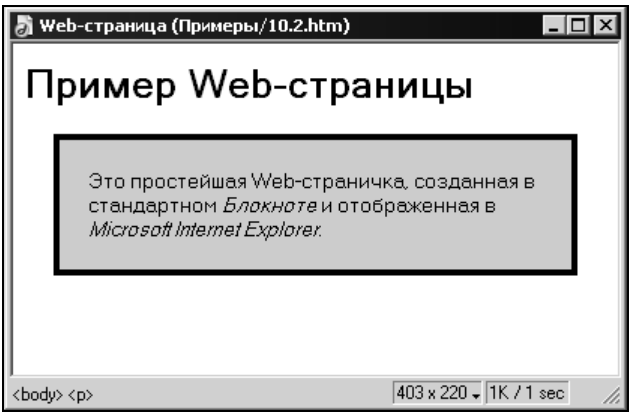


Рис. 10.18. Рамка вокруг абзаца (фон абзаца для удобства сделан светло-серым)

Группа комбинированных списков **Style** задает стиль линии рамки: будет ли она сплошная, точечная или вообще невидимая. Всего в каждом из этих списков доступны девять пунктов:

- **none** — рамка отсутствует (поведение по умолчанию);
- **dotted** — точечная линия;
- **dashed** — пунктирная линия;
- **solid** — сплошная линия;
- **double** — двойная линия;
- **groove** — "вдавленная" трехмерная линия;
- **ridge** — "выпуклая" трехмерная линия;
- **inset** — элемент страницы напоминает выпуклый трехмерный прямоугольник (рис. 10.19);
- **outset** — элемент страницы напоминает вдавленный трехмерный прямоугольник (рис. 10.20).

Как видите, стандарт CSS предоставляет вам довольно много различных стилей для рамок. Пользуясь ими, вы можете создавать довольно интересные вещи, например, подобие стандартных кнопок Windows. (Забегая вперед, можно отметить, что и вести себя они будут как обычные кнопки, если вы кое-что измените в их коде.)

Толщина рамки задается группой комбинированных списков **Width**. В эту группу входят списки **Top** (задает толщину верхней стороны рамки), **Right** (правой), **Bottom** (нижней) и **Left** (левой). Вы можете ввести в любой из этих списков числовое значение и выбрать в расположенном правее раскрывающемся списке единицу измерения, абсолютную или относительную.

Кроме того, в этих комбинированных списках доступны три пункта, задающие предопределенные значения толщины рамки:

- **thin** — тонкая рамка;
- **medium** — средняя;
- **thick** — толстая.

Описанным выше способом мы можем создавать линейки, не пользуясь, как раньше, для этой цели таблицами. Как видите, стили — самый радикальный способ преодолеть недостатки таблиц.

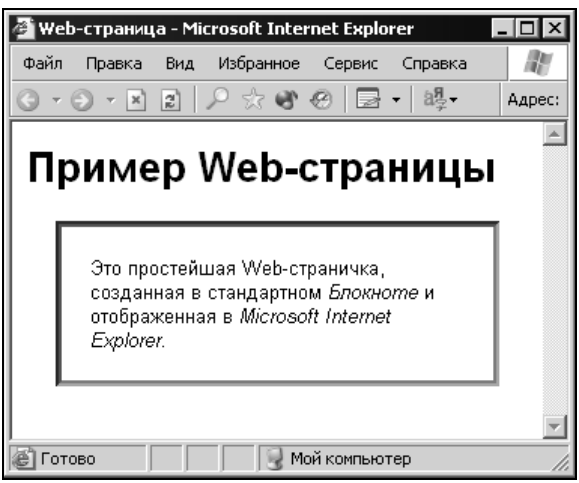


Рис. 10.19. Элемент страницы, созданный с использованием стиля рамки `inset`

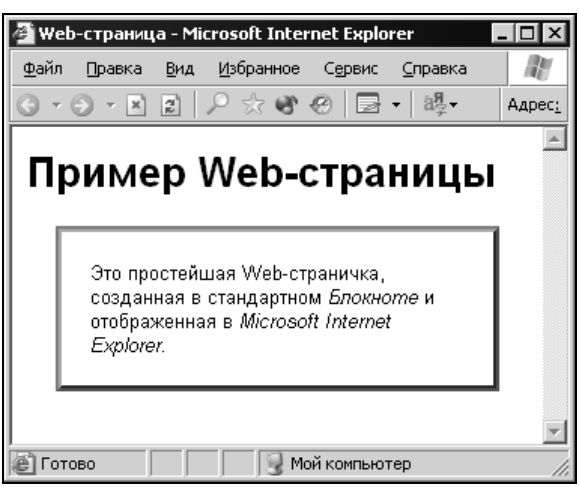


Рис. 10.20. Элемент страницы, созданный с использованием стиля рамки `outset`

С помощью группы селекторов цветов **Color** задаются цвета каждой из сторон рамки. Если цвет не задан, рамка отображается тем цветом, который задан как цвет текста текущего элемента.

Опять же, если вы хотите задать одинаковые параметры для всех сторон рамки, включите флажок **Same for All**, находящийся в соответствующей группе элементов управления. После этого задайте нужные параметры в списке или селекторе цвета, которые станут после этого доступными.

Да, рамка обычно делается одинаковой со всех сторон. Но Dreamweaver и таблицы стилей позволяют вам задать различные стили, значения ширины и цвета для каждой стороны рамки отдельно. Вы даже можете убрать какие-то из сторон рамки или вообще оставить только одну сторону. Такой пример показан на рис. 10.21. Здесь мы задали для левой стороны значение `thick`, а для остальных сторон — 0. Кроме того, мы задали величину отступа **Padding** слева равной 0,5 см, а с других трех сторон убрали его совсем. Попробуйте сделать так же!

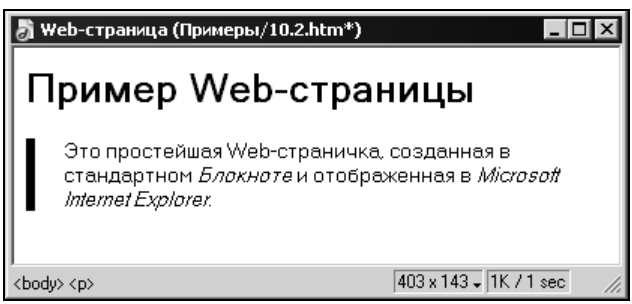


Рис. 10.21. Вертикальная линия слева от текста абзаца

Параметры маркеров списка

Параметры, задающие вид маркеров списка, задаются на вкладке **List** диалогового окна **CSS Style Definition**. Эта вкладка показана на рис. 10.22.

Как вы помните, HTML предоставляет возможность создания списков, состоящих из множества позиций, помеченных специальными значками — маркерами или цифрами (буквами). В первом случае список называется *маркированным*, а во втором — *нумерованным*. Пример маркированного списка вы можете увидеть выше, в описании стилей рамки. Ну а пример нумерованного списка без труда можно найти в этой книге — здесь они используются довольно часто.

HTML предоставляет вам некоторые способы изменить маркер или нумерацию позиций списка. CSS предлагает практически те же возможности.

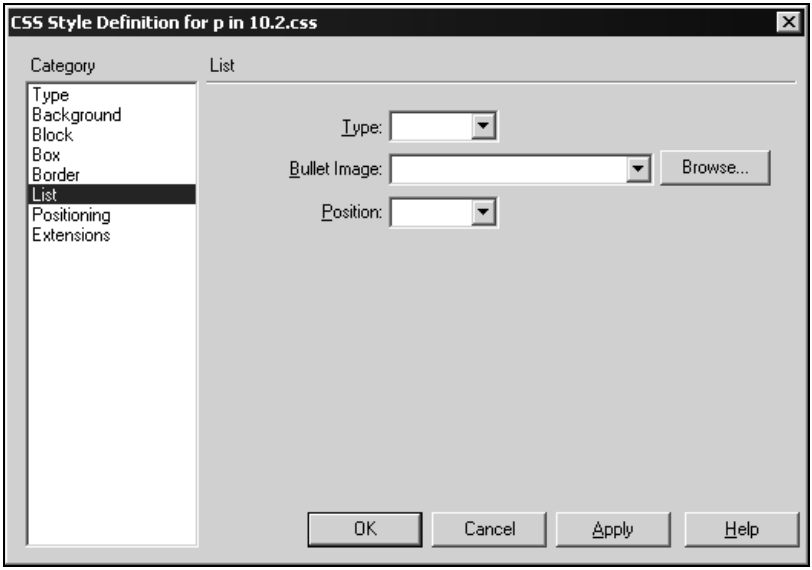


Рис. 10.22. Вкладка **List** диалогового окна **CSS Style Definition**

В раскрывающемся списке **Type** задается вид маркера или нумерации позиций списка. Здесь доступны девять пунктов:

- **disc** — маркер в виде кружка с заполнением (поведение по умолчанию для маркированных списков);
- **circle** — маркер в виде окружности без заполнения;
- **square** — маркер в виде маленького квадратика, с заполнением (на компьютерах Apple Macintosh отображается без заполнения);
- **decimal** — нумерация арабскими цифрами (поведение по умолчанию для нумерованных списков);
- **lower-roman** — нумерация малыми римскими цифрами;
- **upper-roman** — нумерация большими римскими цифрами;
- **lower-alpha** — нумерация малыми латинскими буквами;
- **upper-alpha** — нумерация большими латинскими буквами;
- **none** — нет ни маркера, ни нумерации (поведение по умолчанию для обычных текстовых абзацев).

В поле ввода **Bullet Image** вводится имя файла графического изображения, которое будет использоваться в качестве маркера. Вы можете ввести его вручную или щелкнуть кнопку **Browse** и выбрать нужный файл в диалоговом окне **Select File**. Заметьте, что если задан графический маркер, установки атрибута **Type** теряют силу.

Раскрывающийся список **Position** задает местонахождение маркера или нумерации. Здесь доступны два пункта. Если выбран пункт **inside**, то маркер (нумерация) располагается внутри границ текстового абзаца, т. е. принадлежит ему. Если же выбран пункт **outside**, то маркер (нумерация) размещается за пределами границ абзаца. Таким образом, если задано значение **inside**, позиция списка выглядит компактнее, а если **outside** — текст позиции будет лучше читаться. Значение по умолчанию — **outside**.

Параметры местонахождения

Вкладку **Positioning** диалогового окна **CSS Style Definition** мы рассмотрим позже, когда будем изучать свободно позиционируемые элементы. Сейчас атрибуты, устанавливаемые с ее помощью, нам не пригодятся.

Дополнительные параметры

Стандарт CSS предоставляет целый набор дополнительных атрибутов, задающих особые параметры элементов страницы. Эти параметры задаются на вкладке **Extensions** диалогового окна **CSS Style Definition**, которая показана на рис. 10.23.

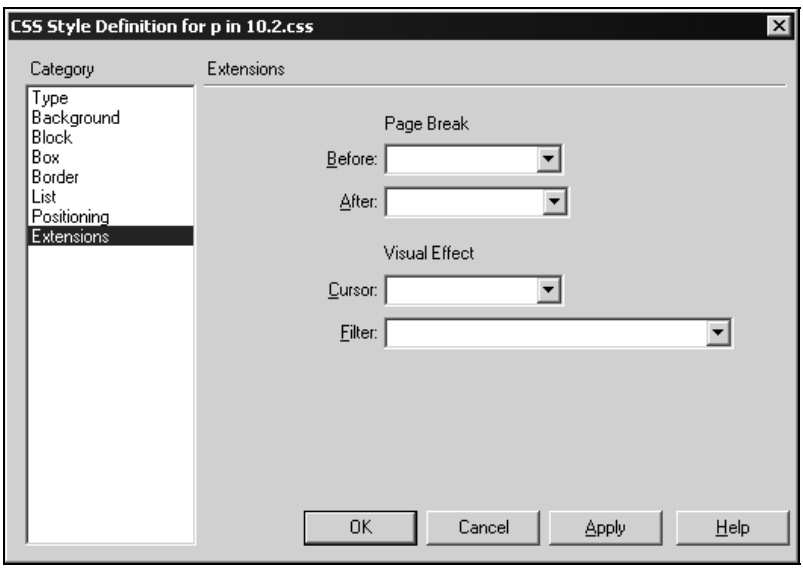


Рис. 10.23. Вкладка **Extensions** диалогового окна **CSS Style Definition**

Группа раскрывающихся списков **Page Break** пригодится, если вы планируете дать возможность пользователю распечатать вашу Web-страницу на принтере. С помощью списков этой группы вы можете дать команду драйверу принтера выполнить прогон листа до или после печати какого-либо элемен-

та страницы и продолжить печать с нового листа. Это может быть полезно, если вы хотите предотвратить появление "висячих" строк или если вы разработали хитроумный печатный дизайн, где одни элементы помещаются на левых, а другие — на правых листах (если используется двусторонняя печать). В этом случае воспользуйтесь раскрывающимися списками **Before** и **After**, задающими прогон листа соответственно перед и после текущего элемента страницы.

В обоих этих списках доступно по четыре пункта. Поскольку они абсолютно одинаковы и вызывают практически идентичные действия, мы рассмотрим их вместе.

Пункт **auto** передает управление размещением информации на бумажных листах Web-обозревателю, т. е. сам Web-обозреватель будет размещать элементы страницы на листах, как ему заблагорассудится. Это поведение по умолчанию.

Если вы хотите, чтобы перед текущим элементом страницы или после него принтер начинал печать с нового листа, выберите в соответствующем раскрывающемся списке пункт **always**. В частности, вы можете установить атрибут **Before** в **always** для всех заголовков, т. к. "повисшие" заголовки — грубейшая ошибка в полиграфии.

Также вы можете захотеть, чтобы какой-то элемент страницы напечатался только на левой или правой странице, если пользователь использует двустороннюю печать. В этом случае вы должны будете выбрать соответственно пункт **left** или **right** раскрывающегося списка. Но учтите при этом, что какая-то страница может остаться вообще пустой.

И еще учтите, что атрибуты управления прогоном листа поддерживаются только достаточно новыми версиями Web-обозревателей.

Группа раскрывающихся списков **Visual Effect** управляет визуальным представлением текущего элемента страницы. В частности, с ее помощью вы можете применить к элементу такое замечательное нововведение, появившееся в Internet Explorer 4.0, как фильтры и преобразования.

Но сначала обратимся к раскрывающемуся списку **Cursor**. С его помощью вы можете задать вид курсора мыши, который он примет при наведении на текущий элемент страницы. Это может быть очень забавно.

Все доступные формы курсора мыши перечислены в табл. 10.3.

Таблица 10.3. *Формы курсора мыши, доступные для задания в раскрывающемся списке* ***Cursor***

Пункт раскрывающегося списка Cursor	Вид курсора мыши
hand	"Указующий перст", появляющийся при наведении курсора мыши на гиперссылки

Таблица 10.3 *(окончание)*

Пункт раскрывающегося списка Cursor	Вид курсора мыши
crosshair	Перекрестье, "прицел"
text	Текстовый курсор
wait	Песочные часы, обозначающие, что Windows "думу думает"
default	Обычная стрелка
help	Обычная стрелка с вопросительным знаком справа
e-resize	Стрелка вправо, "на восток"
ne-resize	Стрелка вверх и вправо, "на северо-восток"
n-resize	Стрелка вверх, "на север"
nw-resize	Стрелка вверх и влево, "на северо-запад"
w-resize	Стрелка влево, "на запад"
sw-resize	Стрелка вниз и влево, "на юго-запад"
s-resize	Стрелка вниз, "на юг"
se-resize	Стрелка вниз и вправо, "на юго-восток"
auto	Управление формой курсора мыши передается Web-обозревателю. Поведение по умолчанию

Имейте, однако, в виду, что перед тем, как задавать форму курсора мыши для каких-то элементов вашей страницы, подумайте, нужно ли это. Скажем, если при наведении курсора мыши на текст он будет принимать форму песочных часов, пользователь вас, мягко говоря, не поймет. Поэтому сначала уясните, что обозначает та или иная форма курсора. Не от нечего же делать Windows ее меняет!

А вот раскрывающийся список **Filter** даст в ваши руки неограниченные возможности по созданию графических спецэффектов. Например, вы можете сделать графическое изображение или текст заголовка размытым или заставить его плавно появляться и так же плавно исчезать. Поверьте — это выглядит потрясающе! И задать эти спецэффекты вы сможете с помощью раскрывающегося списка **Filter**, но... Описание фильтров может растянуться на целую главу — такой это мощный инструмент, поэтому мы не будем описывать их здесь. Если вы заинтересовались фильтрами или преобразованиями, найдите хорошую книжку по CSS или, что еще лучше, обратитесь на сайт Microsoft за нужной документацией.

Применение стилей

После того как вы зададите атрибуты стиля, нажмите кнопку **OK** диалогового окна **CSS Style Definition**. Вы также можете нажать кнопку **Apply**, чтобы сохранить стилевые установки без закрытия этого диалогового окна. Чтобы отказаться от создания нового стиля, нажмите кнопку **Cancel** (хотя таблица стилей при этом все-таки будет создана).

После того как стиль сохранен в таблице (внешней или внутренней — неважно), его нужно применить к выделенному фрагменту текста или целому абзацу. Конечно, если это стиль переопределения тега, его не надо никак применять — Web-обозреватель использует заданные в нем установки при обработке соответствующего тега. Другое дело стилевой класс — нам нужно будет самим указать, что данный текст форматируется с его использованием.

Предположим, что вы задали для тега `<P>` шрифт Verdana размером 9 пунктов. Теперь задайте для тега `<BODY>` светло-желтый цвет фона, но сохраните эту установку не во внешней таблице стиля, а в самой Web-странице 10.2.htm. Для этого в диалоговом окне **New CSS Style** (см. рис. 10.5) включите переключатель **This Document Only**. Это пригодится нам в дальнейшем, когда мы будем изучать работу со стилями.

Создайте еще два стиля для наших экспериментов. Первый из них будет называться `centered`, а второй — `red`. (Как вы уже догадались, это стилевые классы, и поэтому для их создания нужно выбрать переключатель **Make Custom Style (class)** и ввести их имена в меню **Name**.) Для первого стиля задайте выравнивание текста посередине, а для второго — красный цвет текста. Сохраните их также во внешней таблице стилей 10.2.css.

После всех этих манипуляций наша страница примет вид, показанный на рис. 10.24, а панель **CSS Styles** — на рис. 10.25.

Как видите, в списке CSS-стилей отображаются три пункта: **No CSS Style**, **centered** и **red**. Первый пункт обозначает отсутствие любого стиля, т. е. элемент страницы будет форматироваться по умолчанию. Два других задают

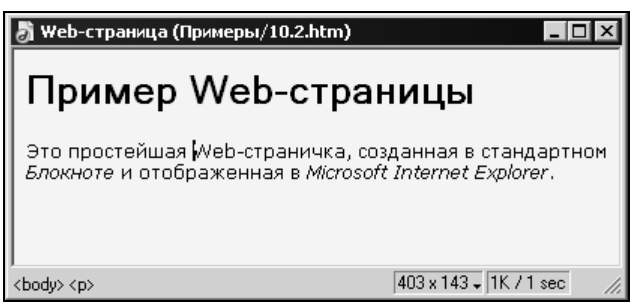


Рис. 10.24. Web-страница 10.2.htm после задания всех стилей

Рис. 10.25. Панель **CSS Styles** после задания всех стилей

созданные нами стилевые классы. Заметьте, что стили, переопределяющие теги, здесь не отображаются, и это правильно.

Имейте в виду также следующее. Если вы задали стиль вида, скажем, `H1.right` (гибрид стилевого класса и переопределения тега), Dreamweaver отобразит его как `right` в списке стилей. При этом он позволит вам присвоить его любому элементу страницы, что будет неправильно, т. к. этот стиль будет работать только в теге `<H1>`. Очень досадно, но если вы используете в своих страницах подобные гибридные стили, вам придется самим помнить о них и выполнять только корректные присвоения.

Так как же применить стилевой класс к элементу страницы? Если у вас открыта панель **CSS Styles**, вам достаточно только выбрать нужный стиль в списке. Естественно, перед этим вы будете должны выделить в окне документа элемент страницы, к которому хотите применить стилевой класс.

Давайте поставим текстовый курсор на заголовок нашей страницы и щелкнем по стилевому классу `centered`. После этого выделим слова "Web-страничка" и щелкнем по стилевому классу `red`. Что получится? (Ответ см. на рис. 10.26.)

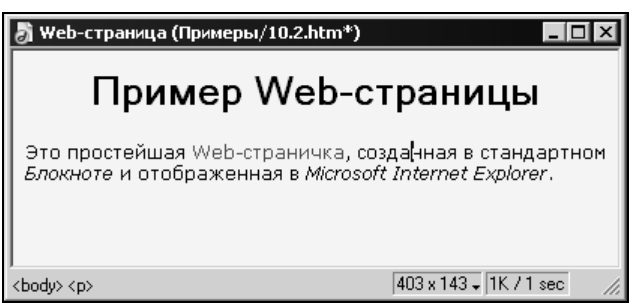


Рис. 10.26. Страница 10.2.htm после применения стилевых классов `centered` и `red`

Теперь поставьте текстовый курсор куда-нибудь на текст заголовка и взгляните на окно CSS-стилей. Dreamweaver подсветит в списке стилевой класс `centered`. То же самое произойдет с классом `red`, если вы поместите тексто-

вый курсор на выделенную красным надпись. Таким образом, вы всегда будете знать, с помощью какого стилевого класса отформатирован тот или иной элемент страницы.

Но что делать, если панель **CSS Styles** у вас закрыта? Специально для такого случая в меню **Text** и контекстном меню предусмотрено подменю **CSS Styles**. В верхней части этого подменю перечислены все созданные вами к этому моменту стилевые классы. Выбранный класс будет выделен галочкой. Само собой, также присутствует пункт **None**, позволяющий убрать форматирование с использованием стилевого класса.

Вообще, Dreamweaver (как и многие популярные Windows-приложения, например, Microsoft Word) ценен тем, что почти любое действие в нем можно выполнить несколькими различными способами. Вам остается только выбрать тот способ, который придется вам больше по душе.

Управление стилями

Осталось рассмотреть, как в Dreamweaver осуществляется управление стилями, в частности, изменение и удаление (как создавать новые стили, вы уже знаете).

В верхней части панели **CSS Styles** вы уже заметили два переключателя: **Apply Styles** и **Edit Styles**. Мы не рассматривали их, т. к. они пока что нам были не нужны. Но сейчас они пригодятся.

Как видите, по умолчанию включен переключатель **Apply Styles**, и список панели **CSS Styles** имеет вид, показанный на рис. 10.3. Говорят, что панель **CSS Styles** работает в *режиме применения стилей*. При этом вы можете щелкнуть по любому пункту этого списка, и Dreamweaver тотчас применит соответствующий стиль к выделенному вами элементу страницы. Но, к сожалению, править стили при этом неудобно: вы не можете выбрать стиль без того, чтобы он не был применен. Что же делать?

Включите переключатель **Edit Styles**. Панель в этом случае примет такой вид, как на рис. 10.27. Она переключится в *режим правки стилей*, в котором очень удобно именно управлять стилями, не опасаясь, что услужливый Dreamweaver применит выбранный стиль к выделенному вами элементу страницы.

Как видите, список стилей в этом режиме имеет несколько другой вид — он стал иерархическим. В качестве ветвей списка выступают таблицы стилей и Web-страницы, в которых сохранен тот или иной стиль. Более того, в этом случае Dreamweaver правильно показывает ВСЕ заданные вами стили, даже переопределения тегов! Это же просто замечательно: теперь нам видно все, что мы нагородили! Но Dreamweaver идет дальше: он показывает даже сам код определения стилей — в правой колонке списка. (Правда, если этот код

достаточно велик, вам придется прокручивать список по горизонтали, но это не вина Dreamweaver.)

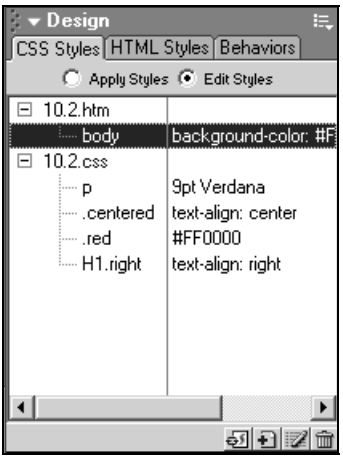


Рис. 10.27. Панель **CSS Styles** в режиме правки стилей

Чтобы изменить необходимый стиль, выделите его и выберите пункт **Edit** контекстного или дополнительного меню. Также вы можете дважды щелкнуть по нужному пункту списка стилей. После этого на экране появится диалоговое окно **CSS Style Definition**, в котором вы сможете выполнить требуемые изменения. После нажатия кнопки **OK** все сделанные изменения сохранятся и будут тотчас применены.

Удалить стиль вы можете, воспользовавшись пунктом **Delete** контекстного или дополнительного меню. Но проще и нагляднее нажать кнопку **Delete Style** (рис. 10.28) в правом нижнем углу панели **CSS Styles**. Естественно, перед этим вы должны будете выбрать нужный стиль в списке.

Рис. 10.28. Кнопка **Delete Style** панели **CSS Styles**

Стиль удаляется сразу же, без всякого предупреждения. Форматирование всех элементов страницы, к которым был применен удаленный стиль, приводится к виду по умолчанию. Однако атрибуты `CLASS` и их значения в тегах этих элементов все же сохраняются, и если вы в дальнейшем создадите стилевой класс с таким же названием, он будет тотчас к ним применен.

Также Dreamweaver предоставляет вам возможность создать копию выбранного стиля. Это может быть полезно, если вы хотите создать новый стиль, слегка переделав уже существующий. Для этого выберите нужный стиль и воспользуйтесь пунктом **Duplicate** контекстного или дополнительного меню.

На экране появится уже надоевшее вам диалоговое окно **CSS Style Definition**. Ну а с ним-то вы знаете, что делать.

Управление таблицами стилей

Кроме манипуляций над отдельными стилями, Dreamweaver предоставляет возможность управлять целыми таблицами стилей. Вы можете редактировать их, создавать и удалять, а также присоединять к документу внешние таблицы и экспортировать внутренние таблицы во внешний файл.

Как правило, нужда редактировать саму таблицу стилей возникает нечасто. Dreamweaver предоставляет достаточно мощные средства правки стилей по отдельности, вне зависимости от того, где они сохранены. Но если все же вам понадобится отредактировать таблицу стилей, выполните одно из нижеперечисленных действий:

- нажмите кнопку **Edit Style Sheet** (рис. 10.29) в правом нижнем углу панели **CSS Styles**;
- дважды щелкните по свободному пространству списка стилей (он должен находиться в режиме правки стилей, т. е. должен быть включен переключатель **Apply Styles**);
- выберите пункт **Edit Style Sheet** в контекстном или дополнительном меню панели **CSS Styles**;
- выберите пункт **Edit Style Sheet** в подменю **CSS Styles** контекстного меню окна документа или меню **Text**.

Рис. 10.29. Кнопка **Edit Style Sheet** панели **CSS Styles**

После этого на экране появится диалоговое окно **Edit Style Sheet**, показанное на рис. 10.30.

Большую часть этого диалогового окна занимает список стилей, определенных во внутренней таблице (если таковая есть), и внешних таблиц стилей, привязанных к этой странице. Вы видите, что этот список в нашем случае содержит два пункта. Нижний пункт обозначает стиль, переопределяющий тег `<BODY>`. (Помните, мы задали для тела документа светло-желтый цвет?) Верхний же пункт обозначает внешнюю таблицу стилей 10.2.css, привязанную к нашей странице. Заметьте, что этот пункт помечен специальным значком.

Под списком находится небольшое текстовое поле, где выводится код определения выбранного в списке стиля. Так что вы сразу можете видеть, что за стиль попал вам под руку... т. е. под мышку.

Рис. 10.30. Диалоговое окно **Edit Style Sheet**

Левее списка находятся пять кнопок. Перечислим их по порядку.

Кнопка **Link** позволяет привязать к странице внешнюю таблицу стилей. Если вы на нее нажмете, на экране появится диалоговое окно **Link External Style Sheet**, показанное на рис. 10.31. В поле ввода **File/URL** введите имя файла внешней таблицы стилей, которую вы хотите привязать к своей странице. Если вы не хотите вводить ее вручную, щелкните кнопку **Browse** и выберите нужный файл в диалоговом окне **Select File**. После этого нажмите кнопку **OK** для привязки таблицы стилей или **Cancel** — для отказа от этого.

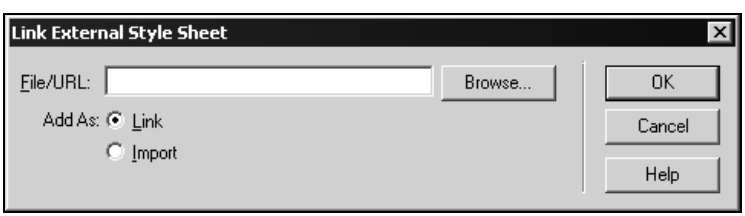


Рис. 10.31. Диалоговое окно **Link External Style Sheet**

В диалоговом окне **Link External Style Sheet** имеется также группа переключателей **Add As**. Переключатель **Link** (включен по умолчанию) выполняет обычную привязку таблицы стилей с помощью тега `<LINK>`; это, собственно, делается почти всегда. А переключатель **Import** позволяет вам *импортировать* внешнюю таблицу стилей, т. е. перенести все содержимое внешней таблицы стилей во внутреннюю. Однако такой возможностью пользоваться не рекомендуется, поскольку команда CSS `@import`, с помощью которой выполняется импортирование, поддерживается далеко не всеми программами

Web-обозревателей. А, кроме того, при импортировании возможен *неразрешимый конфликт стилей*, когда во внутренней и внешней таблицах оказывается два стиля с одинаковым именем. (При импортировании механизм "каскадности" не работает.)

Однако вернемся к диалоговому окну **Edit Style Sheet**.

Кнопка **New** позволит вам создать новый стиль. После ее нажатия на экране появится уже диалоговое окно **CSS Style Definition**. В нем вы сможете создать новый стиль, а, возможно, и новую внешнюю таблицу стилей.

Кнопка **Edit** позволит вам отредактировать существующий стиль. Не будем рассказывать, как это делается, — вы уже это знаете. Чтобы вызвать нужный стиль для редактирования, также можно дважды щелкнуть по нужному пункту списка.

Кнопка **Duplicate** позволит вам создать стиль, являющийся точной копией уже существующего стиля, выбранного в списке. Это полезно, если вы хотите создать новый стиль, взяв за основу уже существующий и слегка его подредактировав.

Кнопка **Remove** удаляет выбранный стиль.

И, наконец, кнопка **Done** закрывает диалоговое окно **Edit Style Sheet**. Вы также можете нажать кнопку закрытия окна, находящуюся в его заголовке.

Ну, вот и все. С кнопками мы разобрались. Но остается вопрос: как все-таки добраться до внешней таблицы стилей? Как ее отредактировать? Как, в конце концов, удалить ссылку на нее?

Очень просто.

Чтобы отредактировать внешнюю таблицу стилей, выберите в списке стилей соответствующий ей пункт и нажмите кнопку **Edit** или просто дважды щелкните по этому пункту. На экране появится еще одно диалоговое окно **Edit Style Sheet**, но в нем будет отображаться содержимое внешней таблицы стилей. В этом-то окне вы и сможете отредактировать ее содержимое. Не верите? Посмотрите на рис. 10.32.

Чтобы удалить ссылку на внешнюю таблицу стилей, воспользуйтесь все той же кнопкой **Remove**.

Выше мы рассматривали, как можно привязать к Web-странице еще одну таблицу стилей. Это можно было сделать в диалоговом окне **Edit Style Sheet**. Однако это совсем не обязательно — то же самое вы можете выполнить, нажав кнопку **Attach Style Sheet** (рис. 10.33) в правом нижнем углу панели **CSS Styles** либо выбрав пункт **Attach Style Sheet**, расположенный в контекстном и дополнительном меню этой же панели. Также этот пункт находится в подменю **CSS Styles** меню **Text** и контекстном меню окна документа. При выборе этого пункта на экране появится диалоговое окно **Select File**, где вы

сможете выбрать нужный файл. После нажатия кнопки **Select** выбранная таблица стилей будет тотчас привязана к странице.

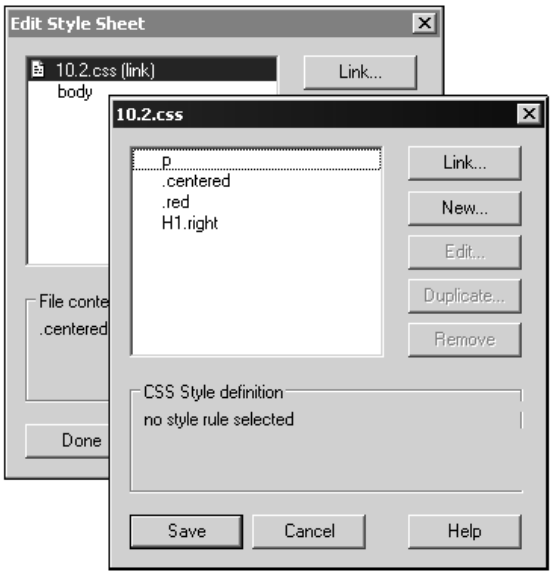


Рис. 10.32. Содержимое внешней таблицы стилей 10.2.css в диалоговом окне **Edit Style Sheet**

Рис. 10.33. Кнопка **Attach Style Sheet** панели **CSS Styles**

Если вы считаете, что сделали удачную внутреннюю таблицу стилей, и хотите сохранить ее в отдельном файле как внешнюю (*экспортировать* таблицу стилей), то Dreamweaver поможет вам и в этом случае. Для этого выберите пункт **Export Style Sheet**, расположенный в контекстном и дополнительном меню панели **CSS Styles**. Также этот пункт располагается в подменю **CSS Styles**, находящемся в меню **Text** и контекстном меню окна документа. После выбора этого пункта на экране появится стандартное диалоговое окно сохранения файла Windows. Введите имя файла и нажмите кнопку сохранения.

Поддержка внутренних стилей

Вот мы и выяснили все о том, как Dreamweaver поддерживает внутренние и внешние таблицы стилей. А что же внутренние стили? А с внутренними стилями ситуация сложнее.

Дело в том, что Dreamweaver поддерживает внутренние стили не полностью. Он отображает на экране результаты их применения, но не предлагает никаких удобных инструментов для создания внутренних стилей и управления ими. По мнению автора, это самый большой и досадный его недостаток. Так что, если вы хотите привязать к какому-либо элементу страницы внутренний стиль, то вам придется править HTML-код вручную либо пользоваться диалоговым окном **Tag Editor**.

Как это сделать?

Давайте рассмотрим работу с внутренними стилями на примере. Откроем все ту же нашу многострадальную страницу 10.2.htm, раскрасим текст заголовка в зеленый цвет, а текст, выделенный курсивом, увеличим в размерах на одну ступень. И сделаем мы это разными способами.

Сначала займемся курсивным текстом. Выделите его, щелкнув по нужной кнопке секции тегов, и выберите в контекстном меню пункт **Edit Tag <*название тега*>**. После этого выберите в списке, расположенном в левой части диалогового окна **Tag Editor**, пункт **Style Sheet/Accessibility**. Окно примет вид, показанный на рис. 10.34.

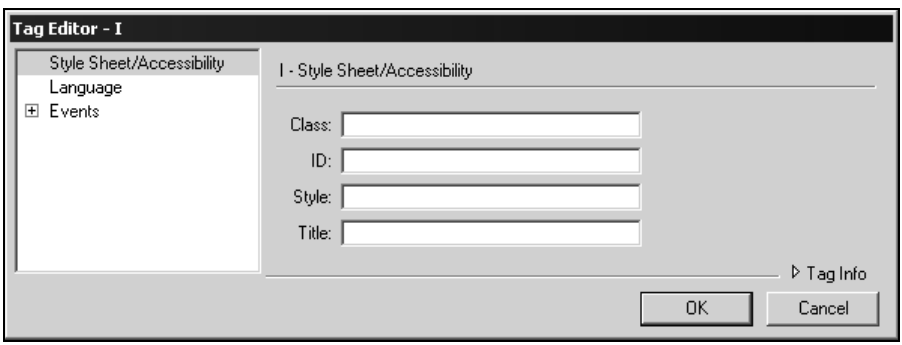


Рис. 10.34. Диалоговое окно **Tag Editor** (выбрана вкладка **Style Sheet/Accessibility**)

Собственно определение тега вводится в поле ввода **Style**. Введите в него текст `font-size: larger`. И нажмите кнопку **OK**. После этого шрифт курсивного текста увеличится в размерах — наш внутренний стиль работает!

Теперь примемся за заголовок. И используем мини-редактор HTML, описанный еще в *главе 2* этой книги. Конечно, можно работать в режиме отображения кода окна документа. Но мини-редактор HTML несколько удобнее, т. к. вы можете видеть одновременно и саму страницу, и нужный фрагмент кода.

Поместите текстовый курсор в текст заголовка и выберите в контекстном меню пункт **Edit Tag <H1>**... Результат этого действия показан на рис. 10.35.

Вставьте в тег `<H1>` атрибут `STYLE` и присвойте ему значение `color: #00FF00;`. Должно получиться так, как показано на рис. 10.36.

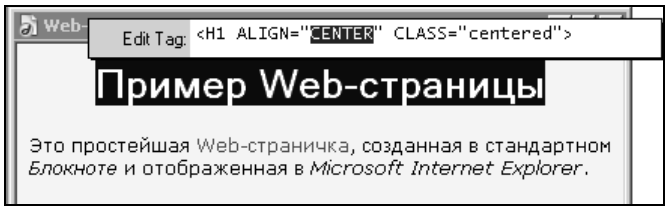


Рис. 10.35. Мини-редактор HTML, отображающий HTML-код тега `<H1>`

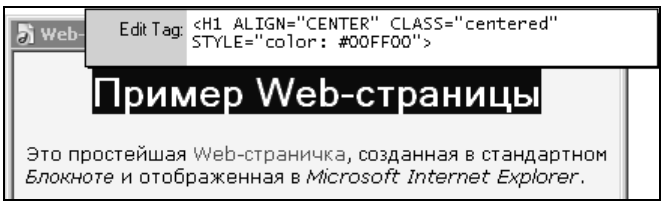


Рис. 10.36. Измененный HTML-код в мини-редакторе HTML

Теперь закройте мини-редактор, нажав клавишу <Enter>. (Нажатие клавиши <Esc> позволит вам отменить сделанные изменения.) И посмотрите на заголовок. Он позеленел, но не от злости, а от того, что мы применили к нему встроенный стиль.

Так что вы можете создавать в Dreamweaver и встроенные стили. Однако имейте в виду, что в этом случае вам придется полагаться только на свое знание HTML и CSS — Dreamweaver почти не автоматизирует ваш труд. А узнать побольше о HTML и CSS вам помогут интерактивные руководства, поставляемые вместе с этой замечательной программой.

Таблицы стилей и шаблоны

А теперь представим такую ситуацию. Вы прочитали о замечательных возможностях таблиц стилей и решили переделать свой сайт Sample site 3 с использованием таблиц стилей. Вы задали таблицу стилей (внешнюю или внутреннюю) для шаблона, сохранили его и обновили все основанные на нем Web-страницы. Все у вас получилось, и, кажется, ничто не предвещает беды...

Но вот вы открыли одну из основанных на шаблоне страниц и решили создать для нее внутреннюю таблицу стилей, дополняющую ту, что находится в шаблоне. Но как это сделать? Ведь внутренняя таблица стилей записывается в HTML-заголовок страницы, куда, как было сказано ранее, хода нет!

Не думайте об этом! Используйте те же, уже знакомые вам, инструменты для создания новых стилей. Единственное: вы не можете переопределить

атрибуты стилей, заданные в таблице, что находится в шаблоне, иначе это вызовет неразрешимый конфликт. За исключением этого, вы полностью свободны!

Дело в том, что при создании шаблона Dreamweaver MX создает еще одну изменяемую область по имени `head`. Эта изначально пустая область находится в HTML-заголовке страницы, внутри тега `<HEAD>`. Когда вы создаете в Web-странице внутреннюю или привязываете к ней внешнюю таблицу стилей, необходимый код помещается как раз в эту область. Вы можете посмотреть HTML-код страницы, основанной на шаблоне, и убедиться в этом сами.

Временные таблицы стилей

Еще одна из новых возможностей Dreamweaver MX — это поддержка так называемых *временных таблиц стилей*. Временные таблицы стилей действуют не все время, а только тогда, когда вы редактируете свою Web-страницу в среде Dreamweaver *(таблицы времени редактирования)*, но не действуют в Web-обозревателе. Другие таблицы могут, наоборот, быть отключены, когда вы редактируете страницу в Dreamweaver *(таблицы времени просмотра)*, а действовать только в Web-обозревателе. Иногда такие таблицы стилей бывают полезны.

Как же работать с временными таблицами стилей? Очень просто!

Выберите пункт **Design Time Style Sheets** в контекстном или дополнительном меню панели **CSS Styles**. Этот пункт также доступен в подменю **CSS Styles** контекстного меню окна документа и меню **Text**. На экране появится диалоговое окно **Design Time Style Sheets** (рис. 10.37).

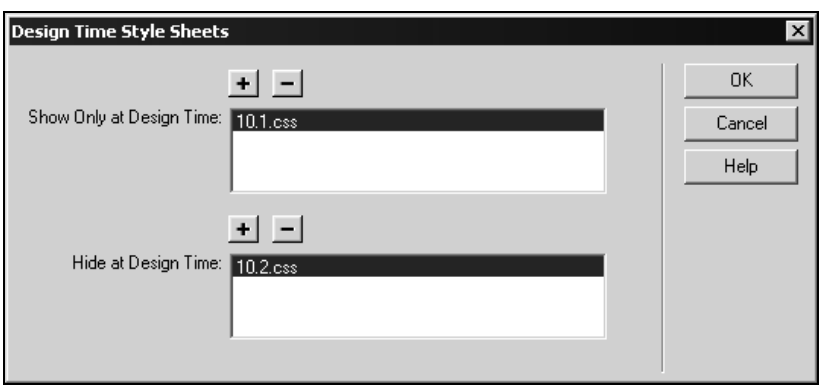


Рис. 10.37. Диалоговое окно **Design Time Style Sheets**

В этом окне находятся два списка. В списке **Show Only at Design Time** перечислены таблицы стилей времени редактирования. В списке **Hide at Design**

Time перечислены таблицы стилей времени просмотра. Точнее, не сами таблицы стилей, а имена файлов, в которых они сохранены.

Чтобы добавить файл в один из списков, нажмите кнопку со знаком "плюс", находящуюся над ним. После этого вам останется выбрать нужный файл в диалоговом окне **Select File** и нажать кнопку открытия.

Чтобы удалить файл из списка, выберите его и нажмите кнопку со знаком "минус", находящуюся над списком.

После нажатия кнопки **OK** Dreamweaver примет ваши указания к сведению. Причем, немедленно.

Для примера вы можете открыть страницу 10.2.htm и задать в качестве таблицы времени редактирования 10.1.htm, а в качестве таблицы времени просмотра — 10.2.htm. Получившийся результат показан на рис. 10.38.

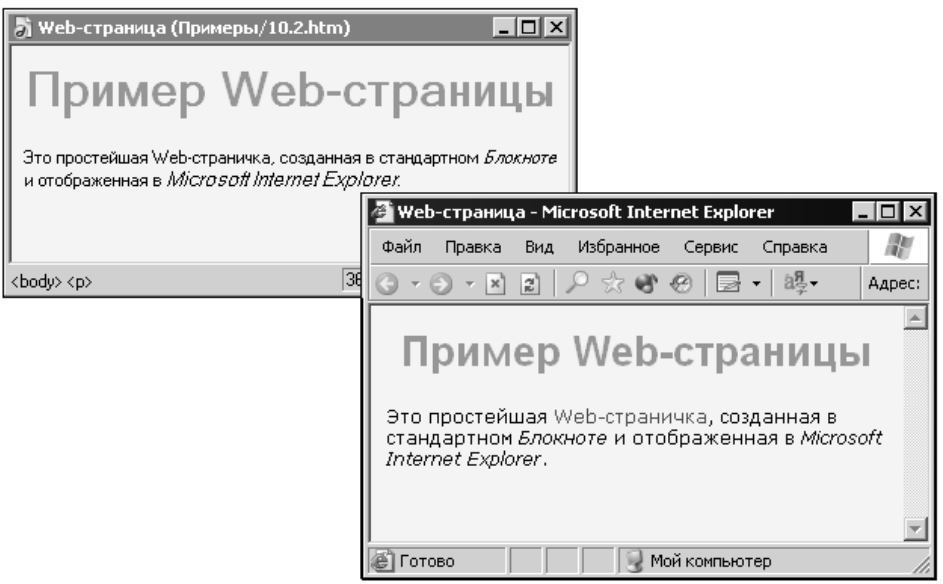


Рис. 10.38. Страница 10.2.htm, для которой были заданы временные таблицы, в окне документа Dreamweaver и в окне Web-обозревателя

Недостатки таблиц стилей и их преодоление

Настала пора поговорить о недостатках таблиц стилей. Всего их два:

- сравнительная "молодость" стандарта CSS, из-за которой его поддерживают только сравнительно новые программы Web-обозревателей, выпущенные после 1997 года;

- крайняя несовместимость между реализациями поддержки таблиц стилей в разных Web-обозревателях, в основном в старых версиях Internet Explorer и Navigator.

Теперь рассмотрим эти недостатки подробнее.

Как мы уже говорили, CSS — довольно молодой стандарт. Он был принят в 1997 году, уже после того, как многие разработчики выпустили на рынок свои программы Web-обозревателей. Конечно, наиболее популярные из них в то время — Internet Explorer и Navigator — вскорости обзавелись поддержкой недавно принятого стандарта. Но очень и очень многие программы не знали о его существовании. И такие "неграмотные" программы используются до сих пор, в частности, в США и Европе, на множестве старых компьютеров, которые там все еще в ходу.

Так какие же Web-обозреватели поддерживают таблицы стилей?

Во-первых, это, конечно, популярнейший Microsoft Internet Explorer. Полная поддержка стандарта CSS появилась в нем, начиная с версии 4.0, хотя некоторые элементы, как говорят, поддерживались еще в приснопамятной 3.0. (Помнится, эта версия принимала файлы таблиц стилей с расширением css за видеофильмы. С чего бы это?) Версии 5.0 и 5.5 поддерживают расширенный набор атрибутов стилей, как стандартных, так и фирменных, не определенных в стандарте. Полное описание атрибутов стилей, поддерживаемых Internet Explorer, вы можете найти на сайте Microsoft для разработчиков: **http://msdn.microsoft.com**. Поверьте — их очень много, и среди них есть совсем уж экзотические. И Dreamweaver поддерживает далеко не все из них.

Во-вторых, следующий по популярности Netscape Navigator. Таблицы стилей он также стал поддерживать, начиная с версии 4.0. Набор поддерживаемых им атрибутов стилей невелик и практически не включает нестандартные, фирменные. Хотя последние версии этой программы — 6.0 и 7.0 — полностью поддерживают стандарт CSS1. Полный список поддерживаемых Navigator атрибутов стилей вы можете найти на сайте Netscape: **http://developer.netscape.com**.

В третьих, новейшие Web-обозреватели: Opera и Mozilla. У них, как говорят разработчики, все с поддержкой стилей в порядке. На самом же деле, у каждой программы свои причуды...

Эти программы составляют подавляющее большинство современного парка Web-обозревателей. Так что вы можете использовать стили в своих страницах без боязни, что кто-то, обладающий "древней" программой, не сможет их прочитать. Абсолютное большинство современных Web-дизайнеров так и поступает. И никаких акций протеста наподобие "Пользователи старого программного обеспечения против таблиц стилей" автор что-то не припомнит.

Но если ваша аудитория будет включать достаточно большой процент пользователей таких "несовместимых" программ, вам придется принимать особые меры, чтобы сделать ваши страницы доступными для них. Какие это меры, объяснять особо не нужно: вы должны полностью отказаться от стилей, где можно, заменить их тегами и атрибутами HTML, а где нельзя — отказаться от всех "наворотов", предлагаемых CSS. Конечно, ваши страницы от этого во многом проиграют, но их, по крайней мере, смогут просмотреть посетители.

Dreamweaver идет вам навстречу в этом нелегком труде. Чтобы сделать вашу страницу совместимой со старыми программами Web-обозревателей, просто откройте ее в окне документа и выберите пункт **3.0 Browser Compatible** подменю **Convert** меню **File**. На экране появится диалоговое окно **Convert to 3.0 Browser Compatible**, показанное на рис. 10.39.

Рис. 10.39. Диалоговое окно **Convert to 3.0 Browser Compatible**

Чтобы преобразовать страницу, использующую стили, в совместимую со старыми программами, включите один из переключателей в группе **Convert**: либо **CSS Styles to HTML Markup**, либо **Both**. (О переключателе **Layers to Table** мы поговорим потом, а сейчас вы пока можете не обращать на него внимания — стилей он все равно не затрагивает.) После этого нажмите кнопку **OK** для запуска преобразования, либо **Cancel** — для отказа от него.

По окончании преобразования Dreamweaver выведет на экран новое окно документа, в котором будет находиться новая Web-страница, являющаяся результатом преобразования старой, "несовместимой" страницы. Это очень удобно: вы можете оставить у себя старую (использующую стили) страницу для каких-то нужд. На взгляд автора, лучше всего делать так: вносить все изменения в старые (использующие стили) версии страниц, а потом преобразовывать их в "совместимые" и именно их публиковать в Сети. Таким образом, вам не нужно будет вносить одни и те же изменения по два раза в разные версии страниц.

Попробуйте преобразовать в "совместимый" вид страничку 10.2.htm. Посмотрите на рис. 10.40 — даже встроенный стиль преобразовался! Хотя фон страницы, заданный в стиле, переопределяющем тег `<BODY>`, Dreamweaver все же "потерял". Ну и растяпа!..

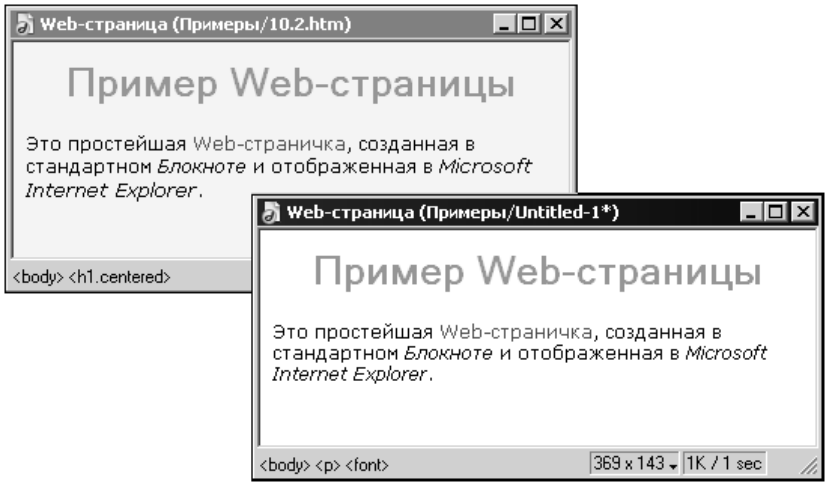


Рис. 10.40. Результат преобразования страницы 10.2.htm (на переднем плане)

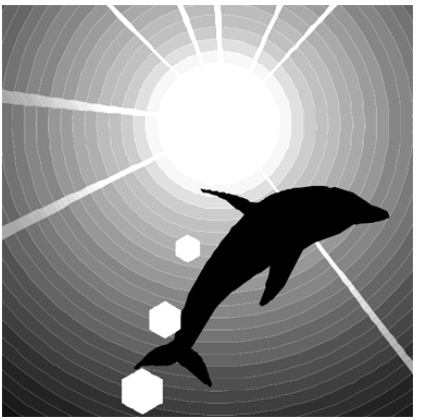
Что дальше?

Вот мы и познакомились с таблицами стилей CSS. В качестве домашнего задания автор советует вам сделать что-нибудь конкретное, чтобы узнать их еще ближе, так сказать, "подержать в руках". Вы можете, скажем, переделать сайт Sample site 3 с использованием таблиц стилей. Или поэкспериментировать с другими Web-страницами.

О стилях рассказывать больше нечего (ну, почти нечего). Существует фирменная документация, имеется описание стандарта CSS на сайте WWWC (**http://www.w3c.org**), так что дополнительную информацию по этому вопросу вы найдете. И, наконец, ничто не может дать вам так много, как личный опыт.

А теперь давайте еще раз взглянем на наши страницы. Да-а-а... Не очень-то и много возможностей дает нам HTML, чтобы расположить их элементы так, как нам хочется. Фреймы, таблицы, стили — что только мы не использовали, и все равно остаются какие-то ограничения. А нам хочется просто задать координаты и размеры, скажем, фрагмента текста, чтобы он точно "лег" на это место без всяких там фреймов и таблиц. Как это делают дизайнеры-полиграфисты.

Значит, хочется... Что ж, читайте дальше!



ЧАСТЬ III

Используем новейшие технологии

Глава 11. Свободно позиционируемые элементы

Глава 12. Анимация элементов Web-страниц

Глава 13. Использование сценариев

Глава 14. Метатеги и серверные директивы

ГЛАВА 11

Свободно позиционируемые элементы

Свободно позиционируемые элементы — совсем недавнее нововведение в HTML. Фактически они появились в 1997 году, одновременно с таблицами стилей. И это не случайно: для создания свободно позиционируемых элементов используются особые атрибуты стилей, задающие координаты и размеры таких элементов.

Было сказано, что свободно позиционируемые элементы появились совсем недавно, но в мире компьютерных вообще и интернет-технологий в частности время летит очень быстро, и пять лет истории какого-либо нововведения — это целая эпоха. Однако до сих пор свободно позиционируемые элементы не снискали особой популярности у Web-дизайнеров. И автор не знает, в чем причина: консерватизм Web-дизайнеров или неприспособленность этих элементов к применению в Web-дизайне.

Однако потенциал свободно позиционируемых элементов очень велик. В самом деле, они позволяют полностью освободиться от "потока" текста, расположить фрагменты содержимого страницы так, как нужно Web-дизайнеру, без оглядки на ограничения таблиц и фреймов. При этом фрагменты могут располагаться друг относительно друга как угодно и даже перекрывать друг друга, чего никак не удастся сделать, используя таблицы. И все это великолепие достигается столь компактным HTML-кодом, что свободно позиционируемые элементы оказываются вне конкуренции.

У свободно позиционируемых элементов есть еще одно неоспоримое преимущество: они предоставляют Web-дизайнеру полнейший контроль над Web-страницей. Используя специально написанные сценарии, разработчик может заставить отдельные элементы страниц двигаться, создавая впечатляющие анимационные эффекты. (Вы, наверно, часто встречали на некоторых страницах анимированные курсоры мыши. Это как раз свободно позиционируемые элементы, "наученные" двигаться за мышью.) Идя дальше, программист (именно так, не Web-дизайнер, а Web-программист!) может

создать на странице строку меню с выпадающими подменю и, в конце концов, превратить Web-страничку в подобие окна Windows-приложения. Один знакомый автора написал, таким образом, целую информационную систему — поверьте, это впечатляет!

Но хватит пустых восторгов! Давайте трезво посмотрим на свободно позиционируемые элементы и выясним, что они собой представляют.

Введение в свободно позиционируемые элементы

Прежде всего нам необходимо выяснить, как создаются свободно позиционируемые элементы страниц. И рассмотрим мы это на небольшом примере, который сделаем вручную, не прибегая к помощи Dreamweaver. Это позволит нам лучше понять, что такое свободно позиционированные элементы, "подержать" их в руках.

Но сначала — немного теории. Куда же без нее...

Что такое свободно позиционируемый элемент

Давайте возьмем какой-нибудь элемент страницы и рассмотрим его. Неважно, будет ли он фрагментом текста, изображением, фильмом или элементом управления ActiveX. С точки зрения Web-обозревателя, все они одинаковы (до известного предела, разумеется). Неважно, будет ли рассматриваемый нами элемент страницы помещен в основной "поток" текста, вынесен во фрейм или "втиснут" в ячейку таблицы. И первый, и второй, и третий способы никак не позволяют реально "освободить" элемент страницы, а только создают видимость этой свободы.

Откройте какую-нибудь страницу, сделанную ранее, и выберите любой ее элемент. Что он такое? Часть содержимого этой страницы. Его размерами управляет Web-обозреватель, основываясь на размерах его родителя, иначе говоря, на размерах свободного пространства, которое можно под него выделить. Точно так же дело обстоит и с позиционированием элемента страницы — оно зависит от размеров и месторасположения его "соседей" и его родителя. А они, в свою очередь, зависят от размера окна Web-обозревателя. Как видите, способа точно контролировать размеры и координаты элемента страницы не существует — все эти параметры зависят от множества факторов, исключить которые обычным способом невозможно.

Конечно, Web-дизайнеры пытаются как-то ограничить действие этих факторов. Они форматируют страницы с использованием таблиц разметки, в результате чего получают более полный контроль над параметрами отдельных элементов страниц. Однако это не более чем выдумки, на которые,

как говорится, голь хитра. Каждый элемент страницы в любом случае зависит от своих "соседей".

Выход: убрать этих "соседей" подальше! "Переселить" элемент страницы из "коммунальной" в "отдельную" квартиру!

Представьте себе, что, помещая на Web-страницу фрагмент текста или изображение, вы задаете его координаты и размеры, руководствуясь только своими нуждами, не принимая во внимание "соседей". А если ваш элемент и "наползет" на тот или иной соседний элемент, ничего страшного не произойдет — он его просто перекроет. (Вы даже можете специально перекрывать одним элементом страницы другие.) Такой "освобожденный" элемент страницы так и называется — *свободно позиционируемый* или просто *свободный*. Он не только вынесен за пределы "потока" текста, но и не зависит от "соседей".

Какими же параметрами свободно позиционируемого элемента можно управлять? А вот какими:

- координатами его левого верхнего угла;
- геометрическими размерами;
- слоем (уровнем), или z-индексом;
- видимостью или невидимостью;
- поведением в случае, если содержимое этого элемента выйдет за его размеры;
- некоторыми другими, о которых будет рассказано ниже.

О *z-индексе* нужно рассказать подробнее. Предположим, что все созданные нами на странице свободно позиционируемые элементы "сложены" в своеобразную "стопку". И не просто "сложены", а еще и пронумерованы в порядке "снизу" "вверх". При этом, как вы поняли, элементы с большими номерами перекрывают элементы с меньшими номерами. Так вот этот номер в воображаемой "стопке" и есть z-индекс.

Внимание!

Свободно позиционируемый элемент в любом случае перекрывает обычное содержимое страницы, лежащее в "потоке" текста.

Как видите, вы можете задать для свободного элемента довольно много параметров. Вы даже можете сделать его видимым или невидимым, что часто используется в анимации и при создании различного рода спецэффектов. Кроме того, это пригодится, если вы будете программировать для своей страницы интерфейс, аналогичный интерфейсу Windows-приложения.

Отдельно хотелось бы остановиться на одной особенности свободных элементов. Как вы поняли, они могут содержать внутри себя некое содержи-

мое, т. е. могут являться родителями для других элементов. Так вот: можно задать поведение такого свободного элемента в случае, если его содержимое перестанет в нем помещаться. Вы можете задать, появится ли в элементе-родителе полоса прокрутки или не помещающееся в нем содержимое будет "отрезано". Таким образом, вы можете создавать своеобразные "документы в документе" без использования фреймов.

И еще. Свободно позиционируемый элемент может содержать внутри себя другие свободно позиционируемые элементы. Во многих случаях это может быть полезным.

Как создается свободно позиционируемый элемент

Итак, мы выяснили, что такое свободно позиционируемый элемент. Осталось разобраться, как он создается.

Рассмотрим небольшую Web-страничку, содержащую фрагмент текста. Ее HTML-код приведен ниже.

```
<HTML>
  <HEAD>
    <TITLE>Пример Web-страницы</TITLE>
  </HEAD>
  <BODY>
    <P>Это текст, который будет показан в окне Web-обозревателя. Это
      текст, который будет показан в окне Web-обозревателя. Это текст,
      который будет показан в окне Web-обозревателя. Это текст, который
      будет показан в окне Web-обозревателя. Это текст, который будет
      показан в окне Web-обозревателя. Это текст, который будет показан в
      окне Web-обозревателя.</P>
  </BODY>
</HTML>
```

Как видите, ничего сложного в нем нет. Эта страничка специально была максимально упрощена, чтобы никакие "навороты" не отвлекали вас от главного.

Что же мы увидим, если откроем данную страничку в Web-обозревателе? Ничего особенного: просто абзац с повторяющимся текстом. Вы можете проверить, если не верите автору на слово.

Этот текст — типичный пример *фиксированного* элемента страницы, находящегося в "потоке" текста и зависящего от "соседей". В данном случае, правда, никаких "соседей" нет. Однако Web-обозреватель сам решает, где и как расположить этот абзац, основываясь на свободном пространстве, которое он может ему отвести. Мы не можем поместить этот абзац там, где хочется, и дать ему нужные размеры.

Теперь преобразуем его в свободно позиционируемый элемент. Для этого просто допишем кое-какой код (в листинге он выделен полужирным шрифтом).

```
<HTML>
  <HEAD>
    <TITLE>Пример Web-страницы</TITLE>
    <STYLE>
      #para {position: absolute;
             left: 50;
             top: 50;
             width: 200;
             height: 100;
             background-color: #00FF00;}
    </STYLE>
  </HEAD>
  <BODY>
    <DIV ID="para">Это текст, который будет показан в окне
      Web-обозревателя. Это текст, который будет показан в окне
      Web-обозревателя. Это текст, который будет показан в окне
      Web-обозревателя. Это текст, который будет показан в окне
      Web-обозревателя. Это текст, который будет показан в окне
      Web-обозревателя.</DIV>
  </BODY>
</HTML>
```

Сохраните этот код в файле под именем 11.1.htm и откройте его в Web-обозревателе. Результат показан на рис. 11.1.

Вот мы и сделали первый свободно позиционируемый элемент. И для этого нам потребовалось внести в исходный HTML-код совсем небольшие изменения. Давайте рассмотрим их подробнее.

Прежде всего мы для создания абзаца использовали тег `<DIV>` вместо тега `<P>`. Тег `<DIV>` применяется для создания любого элемента страницы, фиксированного или свободно позиционируемого, содержащего внутри себя любое, простое или сложное, содержимое. В частности, его можно использовать для создания обычного текстового абзаца, что только что и было сделано. А элемент страницы, созданный с помощью тега `<P>`, т. е. обычный текстовый абзац, не может быть позиционирован свободно.

Затем мы дали нашему элементу уникальное имя. Назвали его `para` и сделали это с помощью атрибута `ID`, поддерживаемого практически всеми "видимыми" тегами. С помощью данного имени мы в дальнейшем зададим для этого абзаца стиль.

Такой принцип создания свободно позиционируемых элементов использовался старыми версиями Dreamweaver. Две последние версии — 4.0 и MX — применяют для этого внутренние стили, что, на взгляд автора, порождает более компактный HTML-код, но, возможно, менее наглядно.

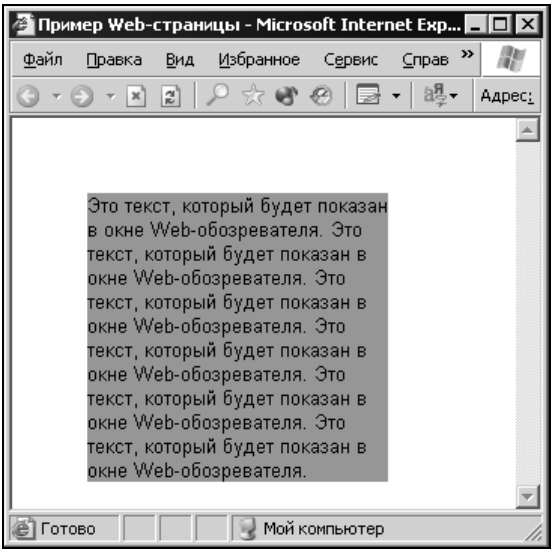


Рис. 11.1. Свободно позиционируемый абзац текста на странице 11.1.htm

Теперь рассмотрим таблицу стилей. В ней мы использовали множество новых атрибутов, которые сейчас и рассмотрим.

Этот атрибут делает элемент страницы свободно позиционируемым:

```
position: absolute;
```

Запомните его — он обязательно должен присутствовать в определении стиля любого свободного элемента. Если у вас что-то не работает, прежде всего, проверьте, задали ли вы для данного атрибута соответствующее значение.

У обычного фиксированного элемента названный атрибут установлен в значение `static`. Это же его значение по умолчанию.

Эти атрибуты задают, соответственно, горизонтальную и вертикальную координаты левого верхнего угла свободного элемента:

```
left: 50;
top: 50;
```

В данный момент они заданы в пикселах, хотя вы можете использовать любую из поддерживаемых CSS единиц измерений. Имейте в виду, что данные

атрибуты имеют силу только для свободно позиционируемых элементов, т. е. тех, у которых атрибут `position` установлен в значение `absolute`.

Примечание

Запомните, что координаты свободно позиционируемого элемента отсчитываются относительно родителя, а не относительно окна Web-обозревателя. Поскольку в нашем случае родителем является сама страница, это несущественно, но вообще об этом забывать не следует.

А эти атрибуты задают, соответственно, ширину и высоту свободно позиционируемого элемента:

```
width: 200;
height: 100;
```

Они также даны в пикселах, хотя могут быть заданы в любой другой из поддерживаемых CSS единиц измерения. И также имеют силу только у свободно позиционируемых элементов.

А теперь взгляните еще раз на рис. 11.1. Вы видите, что высота нашего свободного элемента заметно больше ста пикселов (сравните ее с шириной). Дело в том, что по умолчанию свободный элемент растягивается по вертикали, если его содержимое в нем не помещается. А у нас как раз такой случай. (Как уже говорилось, вы можете задать другое поведение, но об этом позже.)

```
background-color: #00FF00;
```

Здесь специально задан зеленый фон для свободно позиционируемого элемента, чтобы он был заметнее.

А теперь держитесь крепче! Добавлением всего одной строки в таблицу стилей мы превратим свободно позиционируемый элемент в небольшую "страничку в странице" (добавленный текст выделен полужирным шрифтом).

```
#para {position: absolute;
       left: 50;
       top: 50;
       width: 200;
       height: 100;
       background-color: #00FF00;
       overflow: scroll}
```

Сохраните новый файл под именем 11.2.htm и откройте его в Web-обозревателе. И как оно вам (рис. 11.2)?

Давайте еще раз взглянем на добавленную нами строку:

```
overflow: scroll
```

`overflow` — атрибут, как раз и задающий поведение свободно позиционируемого элемента, когда его содержимое в нем не помещается. Значение по умолчанию — `auto` — заставляет элемент растягиваться по вертикали, что мы и наблюдали в предыдущем случае. А значение `scroll` заставляет элемент отобразить полосы прокрутки.

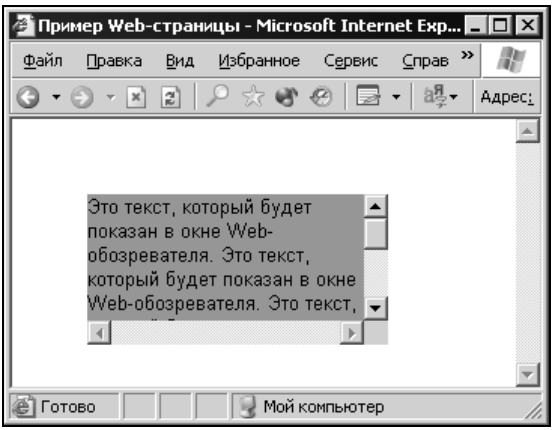


Рис. 11.2. Свободно позиционируемый абзац текста с возможностью прокрутки содержимого

Теперь посмотрим, как это выглядит в Dreamweaver MX, т. е. самой последней (да и в предыдущей версии — 4.0 — этот код будет выглядеть так же).

```
<HTML>
  <HEAD>
    <TITLE>Пример Web-страницы</TITLE>
    <SCRIPT>
      <!-- Сценарий, необходимый для нормальной работы этой
      Web-страницы в старых версиях Navigator -->
    </SCRIPT>
  </HEAD>
  <BODY>
    <DIV ID="para" STYLE="position: absolute; left: 50; top: 50;
      width: 200; height: 100; background-color: #00FF00;
      overflow: scroll">Это текст, который будет показан в окне
      Web-обозревателя. Это текст, который будет показан в окне
      Web-обозревателя. Это текст, который будет показан в окне
      Web-обозревателя. Это текст, который будет показан в окне
      Web-обозревателя. Это текст, который будет показан в окне
      Web-обозревателя. Это текст, который будет показан в окне
      Web-обозревателя.</DIV>
```

```
  </BODY>
</HTML>
```

Этот код дает тот же самый результат, что и предыдущий. Просто выглядит более компактным, особенно если выбросить код сценария, "отвечающего" за совместимость с Navigator.

Вот и все. Как видите, превратить фиксированный элемент страницы в свободный не составляет особого труда. Нужно всего лишь задать соответствующие стили и заменить теги <P> на <DIV>. А в большинстве случаев вам даже не нужно будет более-менее серьезно переделывать дизайн ваших страниц, если, конечно, они не основаны на фреймах или таблицах.

Вопрос в другом. Так ли уж нужны нам эти свободно позиционируемые элементы?

Зачем нужны свободно позиционируемые элементы

Действительно, зачем?

Подавляющее большинство Web-дизайнеров и Web-сайтов до сих пор прекрасно обходятся традиционными способами дизайна. Применять на своих страницах свободные элементы никто особо не хочет. Даже анимированные элементы, казалось бы, замечательное средство привлечь внимание посетителя, — большая редкость на просторах Сети.

Вполне возможно, это обычный консерватизм, уже погубивший множество перспективных технологий. Конечно, не стоит сбрасывать со счетов и то, что свободно позиционируемые элементы поддерживаются только новыми версиями программ Web-обозревателей (фактически теми из них, которые поддерживают таблицы стилей). Но это не довод: те же таблицы стилей применяются несравнимо чаще. Видимо, все-таки консерватизм...

Но если подавляющее большинство Web-дизайнеров как-то обходятся без свободно позиционируемых элементов, так ли уж они нужны? Стоит ли их применять? Может, лучше остаться при своих таблицах и фреймах?

Это решать вам.

Нужны ли вам свободно позиционируемые элементы? Нужны ли они вашему сайту? Сможете ли вы без них обойтись? Вот вопросы, ответы на которые вам стоит поискать.

Конечно, свободные элементы имеют множество преимуществ. Они позволяют располагать фрагменты содержимого Web-страниц так, как вам нужно, и при этом обходиться минимальным количеством кода (сравните с громоздкими таблицами разметки). Однако, если вы надумаете переделывать ваш старый сайт в новый, использующий свободные элементы, переделка

будет весьма значительной. Особенно если ваш сайт достаточно велик и основан на таблицах. Так что, если сайт работает нормально и без свободно позиционируемых элементов, лучше его не трогать.

Правда, иногда бывает, что без свободно позиционируемых элементов не обойтись. Давайте рассмотрим каждую из подобных ситуаций.

- Как гласил рекламный лозунг фильма "Годзилла", размер имеет значение. В самом деле, если ваши Web-страницы сделаны на основе таблиц разметки и достаточно сложны, HTML-код догоняет в габаритах вышеупомянутое земноводное. А чем больше страница, тем дольше она будет загружаться... Свободно позиционируемые элементы позволят радикально сократить размер HTML-кода страниц и при этом добавят им такие возможности, которых у таблиц просто нет и не будет.
- Вы задумали такой дизайн для своих Web-страниц, который никак не создать без использования свободно позиционируемых элементов. Подобный дизайн может включать "окошки в окошке" для показа фрагментов текста, перекрывающиеся элементы и т. п.
- Вы собираетесь анимировать некоторые элементы страницы. Но прежде чем их анимировать, нужно сделать их свободно позиционируемыми.

Вот, вроде бы, и все случаи, когда вам не обойтись без свободных элементов. В остальных случаях они не обязательны.

Работа со свободно позиционируемыми элементами

Теперь рассмотрим, как в Dreamweaver создаются свободно позиционируемые элементы.

Создание свободно позиционируемых элементов

Создайте в Dreamweaver новую Web-страницу. С ней мы и будем экспериментировать.

Однако прежде, чем начинать свои эксперименты, давайте выполним некоторые предварительные операции. Сначала включим измерительные линейки, для чего включим пункт-выключатель **Show** в подменю **Rulers** меню **Modify** или нажмем комбинацию клавиш <Ctrl>+<Alt>+<R>. После этого включим координатную сетку, включив пункт-выключатель **Show Grid** подменю **Grid** меню **View** или нажав комбинацию клавиш <Ctrl>+<Alt>+<G>. И напоследок включим "прилипание", включив пункт-выключатель **Snap To Grid** подменю **Grid** меню **View** или нажав комбинацию клавиш <Ctrl>+ +<Alt>+<Shift>+<G>. Подробнее обо всех этих манипуляциях *см. главу 8.*

Примечание

Имейте, однако, в виду, что это всего лишь рекомендации. Измерительные линейки, сетка и "прилипание", по идее, должны помочь вам, но если вы привыкли работать без таких визуальных "подсказок", не включайте их.

Чтобы создать свободно позиционируемый элемент, проще всего нажать кнопку **Draw Layer** (рис. 11.3), расположенную на вкладке **Common** панели объектов.

Рис. 11.3. Кнопка **Draw Layer** панели объектов

Однако после нажатия этой кнопки новый свободный элемент на странице не появится. Его нужно будет нарисовать примерно так же, как мы рисовали ячейки и таблицы разметки в *главе 8*. Поместите курсор мыши в окно документа — вы видите, что он принял вид крестика. Это значит, что теперь вы можете нарисовать свободный элемент. Щелкните мышью туда, где должен быть его верхний левый угол, и проведите мышью, не отпуская ее левой кнопки. Вы увидите, что между курсором мыши и точкой, где вы щелкнули, тянется прямоугольник, обозначающий границы вновь создаваемого элемента. После того как он примет желаемые размеры, отпустите левую кнопку мыши. Все, свободно позиционируемый элемент готов (рис. 11.4).

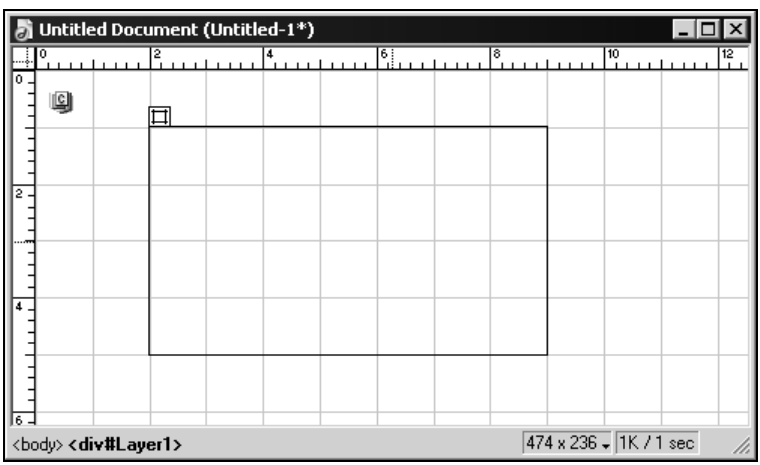


Рис. 11.4. Свободно позиционируемый элемент в окне документа Dreamweaver

Второй способ создания свободного элемента — выбор пункта **Layer** меню **Insert**. В этом случае вам не придется рисовать в окне элемент — он появится сразу же. Вам останется только изменить его размеры и местоположение.

Посмотрите на рис. 11.4. Изображенный на нем свободный элемент в настоящий момент выбран. Текстовый курсор находится внутри него, поэтому вы сразу же можете создать какое-либо содержимое. Чтобы выйти из свободного элемента, щелкните мышью где-либо вне его. А чтобы вернуться в свободный элемент, щелкните мышью по нему (но не по его границе).

Заметьте, что граница невыбранного свободного элемента отображается тускло-серым цветом, чтобы не отвлекать вас. Если же она вам все-таки мешает, вы можете отключить показ границ невыбранных свободных элементов. Для этого отключите пункт-выключатель **Layer Borders** подменю **Visual Aids** меню **Views**. После этого границы невыбранных свободных элементов пропадут, однако граница выбранного элемента все еще будет видна.

Теперь щелкните где-нибудь на границе свободного элемента. Вы также можете щелкнуть внутри свободного элемента, удерживая нажатой клавишу <Shift>. Результат показан на рис. 11.5.

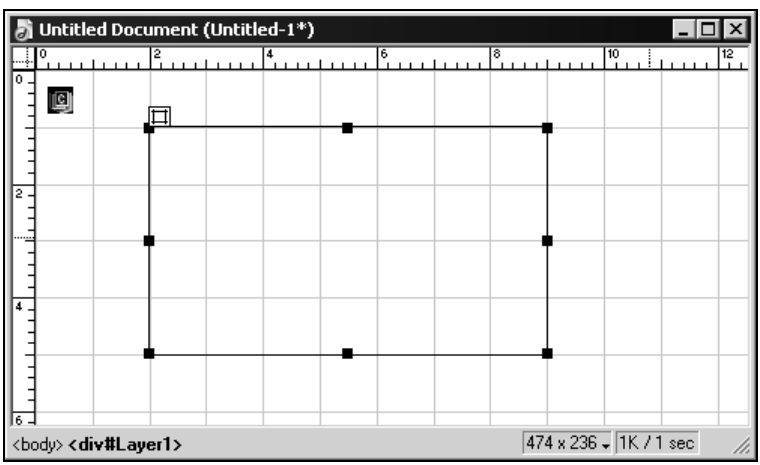


Рис. 11.5. Свободно позиционируемый элемент с выделенной границей

Как видите, в данном случае свободно позиционируемый элемент предлагает вам набор маркеров изменения размера. Кроме того, в верхнем левом углу вы видите небольшой прямоугольник, называемый *"захватом"* (рис. 11.6). "Ухватившись" за него мышью, вы можете двигать свободный элемент. Точно такого же результата можно достичь, "ухватившись" мышью за саму границу элемента, но специальный "захват", согласитесь, удобнее. Щелкнув по нему, вы также сможете выделить границу свободного элемента.

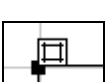


Рис. 11.6. "Захват" свободно позиционируемого элемента

Кроме всего прочего, в окне документа заметен некий значок, представленный на рис. 11.7. Это *значок свободного элемента*, показывающий, где он должен был бы находиться, если бы не был свободно позиционирован. Вы можете щелкнуть по этому значку для того, чтобы выделить свободный элемент.

Рис. 11.7. Значок свободно позиционируемого элемента

Чтобы удалить свободный элемент, выделите его границу и нажмите клавишу <Del>.

Вы также можете создавать свободно позиционируемые элементы, вложенные в другие свободно позиционируемые элементы. Для этого достаточно просто нарисовать или вставить новый элемент внутрь уже существующего. Пример вложенных друг в друга свободных элементов показан на рис. 11.8.

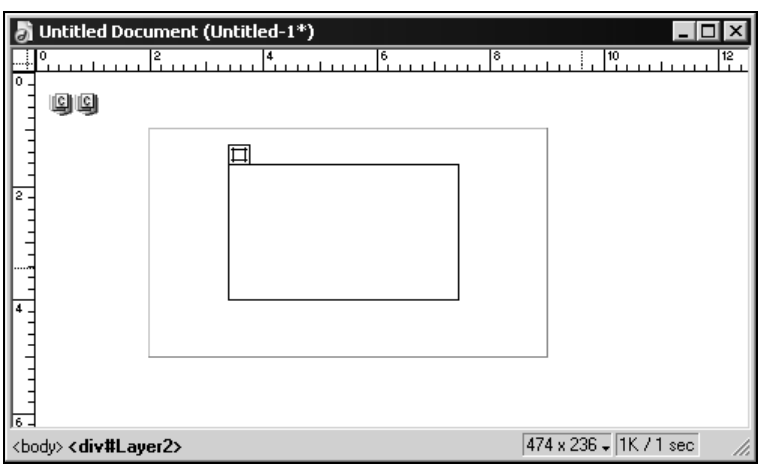


Рис. 11.8. Свободно позиционируемые элементы, вложенные друг в друга (сетка временно отключена)

Вы уже знаете, что свободно позиционируемые элементы могут перекрывать друг друга. Обычно элементы, созданные позже, перекрывают элементы, созданные раньше, но вы можете изменить порядок перекрытия. Для этого выберите нужный свободный элемент и в подменю **Arrange** меню **Modify** выберите пункт **Bring To Front**, чтобы увеличить z-индекс элемента, или пункт **Send To Back**, чтобы его уменьшить. Однако эти пункты работают очень странно, и для задания z-индекса вам лучше воспользоваться другими средствами, описанными ниже.

В том же самом подменю **Arrange** меню **Modify** находится пункт-выключатель **Prevent Layer Overlaps**. Если он включен, Dreamweaver не позволит

вам "наложить" один свободно позиционируемый элемент на другой, т. е. вы не сможете ни переместить его, ни изменить его размеры так, чтобы он перекрыл другие свободные элементы. Однако те элементы, которые уже "лежат" на других, останутся на своих местах.

Сохраните готовую Web-страницу под именем 11.3.htm.

Параметры свободно позиционируемых элементов

Набор параметров свободных элементов, которые мы можем задать в Dreamweaver, весьма обширен. Все они будут представлены в редакторе свойств, когда вы выделите границу свободного элемента (рис. 11.9).

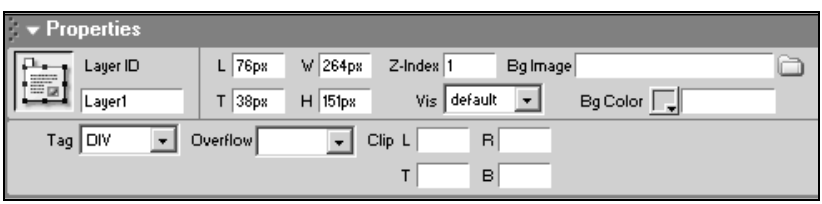


Рис. 11.9. Вид панели редактора свойств при выделенном свободном элементе (тег <DIV> или <SPAN>)

В поле ввода **Layer ID** вводится уникальное имя свободного элемента. Dreamweaver по умолчанию подставляет туда автоматически сгенерированное имя вида `Layer<Порядковый номер>`. Если хотите, можете ввести более вразумительное имя. Вы можете также выбрать пункт **ID** контекстного меню свободного элемента и ввести новое имя в диалоговом окне **Change Attribute**.

В полях ввода **L** и **T** вводятся, соответственно, горизонтальная и вертикальная координаты верхнего левого угла свободно позиционируемого элемента. Заметьте, что эти величины могут быть заданы в любой из единиц измерений, поддерживаемых CSS.

В полях ввода **W** и **H** вводятся, соответственно, ширина и высота свободно позиционируемого элемента. Эти величины также могут быть заданы в любой из единиц измерений, поддерживаемых CSS.

В поле ввода **Z-Index** задается уже известный вам z-индекс, иначе говоря, порядковый номер данного свободного элемента в "стопке" других. Эта величина может быть как положительной, так и отрицательной. Свободный элемент перекрывает все элементы с меньшим значением z-индекса и перекрывается элементами с большим его значением. Кроме того, свободный элемент перекрывает обычное содержимое страницы, лежащее в "потоке" текста.

С помощью раскрывающегося списка **Vis** вы можете задать, будет ли свободный элемент видим на странице. Здесь доступны четыре пункта:

- **inherit** — заставляет элемент "наследовать" видимость от родителя, т. е. если родитель видим, видим и сам элемент, и наоборот;
- **visible** — делает элемент видимым;
- **hidden** — делает элемент невидимым (скрытым);
- **default** — как правило, аналогично **inherit**.

Похожую функцию выполняет подменю **Visibility** контекстного меню свободного элемента. Оно содержит те же четыре пункта, что и раскрывающийся список **Vis**.

В поле ввода **Bg Image** вводится имя файла графического изображения, которое будет использовано в качестве фона. Вы также можете щелкнуть значок папки, расположенный справа от поля ввода, и выбрать нужный файл в появившемся на экране диалоговом окне **Select File**.

С помощью селектора цветов **Bg Color** задается цвет фона свободного элемента.

В раскрывающемся списке **Tag** выбирается тег, с помощью которого создается свободно позиционируемый элемент. Этот список содержит только два пункта: **SPAN** и **DIV**. Их назначение должно быть вам понятно. По умолчанию свободный элемент создается с использованием тега `<DIV>`.

Аналогичную функцию выполняет подменю **Tag** контекстного меню свободного элемента. Оно содержит те же четыре пункта, что и раскрывающийся список **Tag**.

С помощью раскрывающегося списка **Overflow** задается поведение свободного элемента в случае, если его содержимое в нем не помещается. Здесь доступны четыре пункта:

- **visible** — заставляет свободный элемент растянуться по вертикали, чтобы вместить в себя все содержимое;
- **hidden** — просто скрывает ("обрезает") ту часть содержимого свободного элемента, которая превышает его размеры;
- **scroll** — заставляет свободный элемент отобразить полосы прокрутки независимо от того, помещается ли в нем все содержимое или нет;
- **auto** — заставляет свободный элемент отобразить полосы прокрутки только тогда, когда его содержимое в нем не помещается;
- пункт **<пустая строка>** — аналогичен **visible** (значение по умолчанию).

Пункт **scroll** предусмотрен для таких случаев, когда содержимое свободно позиционируемого элемента точно выверено, и появление и исчезновение полос прокрутки может его изменить. В других случаях используйте пункт **auto** или другие пункты.

Группа полей ввода **Clip** позволяет задать координаты *видимой области* свободного элемента. Каждый свободный элемент может иметь такую видимую область, иначе говоря, фрагмент содержимого, который виден в окне Web-обозревателя. Содержимое, не входящее в эту область, скрывается. Задание видимой области может понадобиться для создания каких-либо спецэффектов на Web-странице.

В группу полей ввода **Clip** входят четыре поля:

- **L** — горизонтальная координата верхнего левого угла;
- **T** — вертикальная координата верхнего левого угла;
- **R** — горизонтальная координата нижнего правого угла;
- **B** — вертикальная координата нижнего правого угла.

Работа с группой свободно позиционируемых элементов

Dreamweaver, кроме всего прочего, позволяет выделить одновременно несколько свободно позиционируемых элементов и произвести над ними некоторые манипуляции. В этом случае все выделенные элементы ведут себя как одна группа.

Чтобы выделить сразу несколько свободных элементов, поступите следующим образом. Сначала выделите один из них, щелкнув на нем мышью (выделен сам элемент или его граница — значения не имеет). После этого щелкните на границах всех остальных элементов, удерживая нажатой клавишу <Shift>. Все элементы, по границам которых вы щелкнули, будут выделены. Заметьте, что маркеры изменения размеров всех этих элементов, кроме того, на котором вы щелкнули в последнюю очередь, будут иметь вид белых, а не черных квадратов (рис. 11.10).

Прежде всего, вы можете перемещать выделенные таким образом свободные элементы. При этом будет перемещаться сразу вся группа элементов.

Также вы можете изменить размеры выделенных элементов. Точнее, уравнять их ширину или высоту. Для этого выберите в подменю **Align** меню **Modify**, соответственно, пункт **Make Same Width** (или нажмите комбинацию клавиш <Ctrl>+<Shift>+<7>) или **Make Same Height** (комбинация клавиш <Ctrl>+<Shift>+<9>).

Для группы свободно позиционируемых элементов также доступна возможность выравнивания. Свободные элементы выравниваются по одной из сторон одного из элементов группы, расположенного с соответствующего края. Например, если задано выравнивание по верху, все элементы выравниваются по верхней стороне самого верхнего элемента.

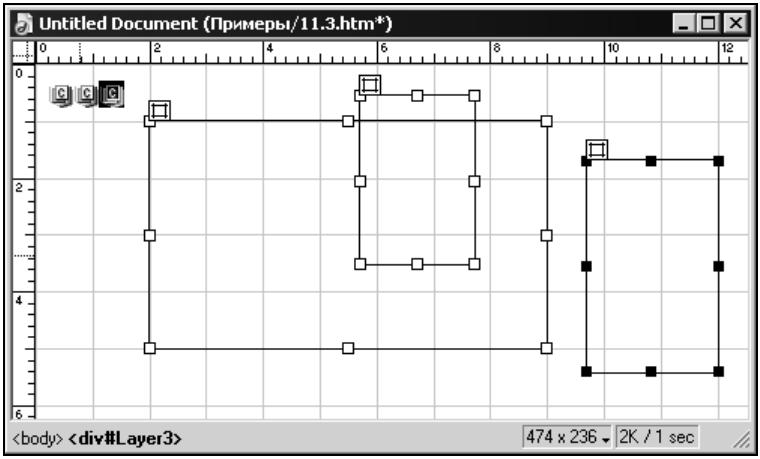


Рис. 11.10. Группа выделенных свободно позиционируемых элементов

Для задания выравнивания используется четыре других пункта подменю **Align** меню **Modify**:

- **Left** (комбинация клавиш <Ctrl>+<Shift>+<1>) — выравнивание по левому краю;
- **Right** (комбинация клавиш <Ctrl>+<Shift>+<3>) — по правой стороне;
- **Top** (комбинация клавиш <Ctrl>+<Shift>+<4>) — по верхней стороне;
- **Bottom** (комбинация клавиш <Ctrl>+<Shift>+<6>) — по нижней стороне.

Имейте в виду, что у свободных элементов нет никаких параметров, задающих выравнивание. Реально выравнивание производится заданием соответствующих значений параметров местоположения.

И наконец, вы можете задавать некоторые параметры сразу для нескольких свободных элементов. К таким параметрам относятся: местоположение и размеры (поля ввода **L**, **T**, **W** и **H** редактора свойств), видимость (раскрывающийся список **Vis**), тег (раскрывающийся список **Tag**), фоновое изображение (поле ввода **Bg Image**) и цвет фона (поле ввода **Bg Color**).

Работа со свободно позиционируемыми элементами

Если на вашей странице создано несколько свободно позиционируемых элементов, управлять ими удобнее всего в специальной панели. Панель **Layers** появляется на экране после включения пункта-выключателя **Layers** подменю **Others** меню **Window** или нажатия клавиши <F2>. Вы можете также щелкнуть по любому свободному элементу правой кнопкой мыши и выбрать в контекстном меню пункт **Layers Panel**. Сама эта панель показана на рис. 11.11.

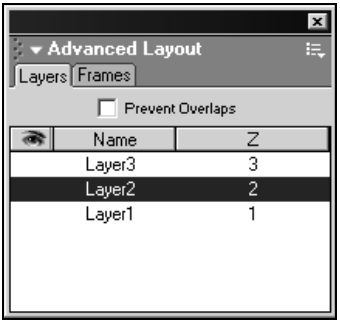


Рис. 11.11. Панель **Layers**

Как видите, большую часть этой панели занимает список свободно позиционируемых элементов, имеющихся на странице. Данный список представляет собой таблицу, состоящую из трех колонок: изображение глаза (видимость), **Name** (имя свободного элемента) и **Z** (z-индекс). Вы можете менять размеры двух последних колонок, перетаскивая мышью границу между ними. И, разумеется, любой из элементов, перечисленных в списке, можно выбрать и произвести над ним предусмотренные в Dreamweaver манипуляции.

В верхней части панели находится флажок **Prevent Overlaps**. Если он включен, Dreamweaver следит, чтобы свободные элементы не перекрывали друг друга. Этот флажок аналогичен пункту **Prevent Layer Overlaps**, находящемуся в подменю **Arrange** меню **Modify**. Так что вы можете пользоваться либо пунктом меню, либо флажком.

Пункты списка свободных элементов выбираются так же, как пункты любого другого списка. Но, кроме того, вы можете выбрать сразу несколько элементов. Для этого выберите первый элемент обычным щелчком мыши, после чего щелкните на остальных элементах, удерживая нажатой клавишу <Shift>.

Как только вы выбираете какие-либо из элементов, они тотчас же отображаются в окне документа как выделенные. Это очень удобно — вы сразу же видите, какие именно свободные элементы вы выбрали. И, наоборот, когда вы выделяете один или несколько элементов в окне документа, они отображаются выбранными в панели **Layers**.

Вы можете изменить имя любого из свободных элементов, перечисленных в списке. Для этого используется уже знакомая вам методика: вы выбираете соответствующий пункт списка и щелкаете по нему снова в колонке **Name** (имя). (Возможно, щелкнуть придется дважды.) После этого вместо имени свободного элемента появится поле ввода, в котором вы сможете ввести новое имя. Не забудьте нажать клавишу <Enter> для сохранения нового имени или клавишу <Esc> — для отказа от него и возврата к старому имени.

Теперь обратите внимание на колонку **Z** списка элементов. Дело в том, что свободные элементы отображаются в списке в том порядке, в котором они перекрывают друг друга. Иначе говоря, они отсортированы по z-индексу. И Dreamweaver предоставляет вам несколько возможностей по изменению данного порядка.

Самый простой способ изменить порядок перекрытия — это перетащить нужный элемент на новое место в списке. При этом Dreamweaver сам изменит значения z-индексов всех свободных элементов, расположенных на странице. Увы, иногда он делает это не очень корректно; в таком случае не обойтись без ручной коррекции, о которой мы сейчас и узнаем.

Если вы хотите точно управлять перекрытием одними свободно позиционируемыми элементами других, задайте значения z-индекса вручную. Для этого также выберите нужный элемент в списке и снова щелкните по нему в колонке **Z**. Вместо значения z-индекса появится поле ввода, в котором вы сможете ввести новое значение, после чего нажмите клавишу <Enter> для его сохранения или клавишу <Esc> — для его отмены. Останется только проверить и, если нужно, изменить значения z-индекса у соседних свободных элементов в списке, чтобы они не совпадали с уже введенным.

Также панель **Layers** предлагает вам самый простой способ изменить видимость свободного элемента. Для этого используется колонка с изображением глаза, самая левая в списке. Выберите нужный элемент в списке и последовательно щелкайте по нему в колонке со значком глаза. При этом содержимое колонки будет последовательно меняться:

- изображение закрытого глаза — элемент невидим;
- изображение открытого глаза — элемент видим;
- пустая колонка — элемент "наследует" видимость у родителя.

В качестве примера см. рис. 11.12 — на нем изображены все три значения параметра видимости, доступных для свободно позиционируемого элемента.

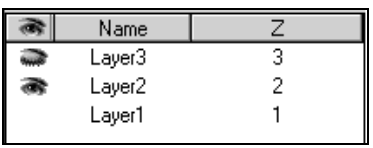


Рис. 11.12. Три свободно позиционируемых элемента (сверху вниз): невидимый, видимый, "наследующий" видимость у родителя

Самая интересная возможность, предлагаемая панелью **Layers**, — возможность "переноса" одних элементов в другие. Иначе говоря, создание вложенных элементов. Для этого также используется перетаскивание свободных элементов с помощью мыши.

Выберите в списке свободный элемент, который вы хотите "вложить" в другой. После этого найдите в списке элемент, в который вы хотите "вложить" выбранный, но не выбирайте его. А перетащите на него выбранный элемент, удерживая нажатой клавишу <Ctrl>; при этом элемент, на который вы "тащите" выбранный, будет выделен тонкой синей рамкой. Осталось только "отпустить" перетаскиваемый элемент; на рис. 11.13 изображен результат этого действия. Также, возможно, придется подкорректировать z-индекс.

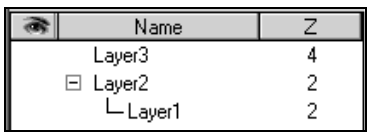


Рис. 11.13. Вложенные свободно позиционируемые элементы в списке панели **Layers**

Как видите, в случае наличия вложенных свободных элементов в списке создается "дерево", "растущее" из родителя. Вы можете упорядочивать вложенные элементы, изменяя их z-индекс, независимо от невложенных элементов. Также вы можете создавать целые структуры свободных элементов, многократно вложенных друг в друга. Пример такой структуры показан на рис. 11.14. Это вполне возможно сделать, правда, трудно вообразить себе такую ситуацию, когда бы такие структуры понадобились.

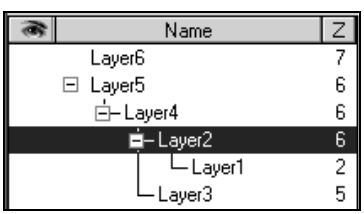


Рис. 11.14. Структуры свободно позиционируемых элементов многократной вложенности

Если вы хотите "вынести" вложенные элементы из родителя, просто перетащите их в нужную позицию в списке. После этого в большинстве случаев придется изменить z-индекс.

Пример использования свободно позиционируемых элементов

Напоследок рассмотрим небольшую Web-страницу, использующую свободно позиционируемые элементы для размещения содержимого. Пусть это будет новая начальная страница для нашего сайта. Разместим на ней заго-

ловок, портрет гипотетического создателя, полосу навигации, начальный текст и сведения об авторских правах. Предположим, что мы уже разработали умопомрачительный дизайн, и перейдем сразу к следующему этапу.

Откройте панель **Site** Dreamweaver, если вы ее уже закрыли. Переключитесь на первый наш сайт Sample site 1. Создайте в этом сайте новую страницу по имени default2.htm. Это и будет наша новая главная страница. Откройте ее.

Автор уже не будет предлагать вам задать ее параметры, т. е. название, цвета фона, текста и гиперссылок. Это само собой разумеется. И не будет предлагать вам значения этих параметров. Примените фантазию. Единственное условие: ваше творение должно хорошо выглядеть. Хотя, вероятно, наилучший результат — следовать оформлению старой главной страницы этого сайта (помните о принципе единства оформления). Мы так и сделали.

Теперь давайте откроем старую главную страницу. Разместите окна со старой и новой главными страницами на экране так, чтобы они не перекрывали или не сильно перекрывали друг друга. Таким образом, вы будете видеть обе страницы и сможете без проблем копировать содержимое из одной страницы в другую.

Как любой театр начинается с вешалки, так и любая Web-страница начинается с заголовка. Для него мы создадим свободно позиционируемый элемент, расположенный в верхней части страницы, и назовем его `Header`. Просто создадим его, не заботясь о реальных размерах и местоположении. Все эти параметры мы скорректируем потом, когда будет готово все содержимое страницы.

Скопируйте со старой страницы текст заголовка в созданный нами свободный элемент. Измените тег с `<P>` (присваиваемый по умолчанию) на `<H1>` (заголовок первого уровня), выбрав пункт **Heading 1** в раскрывающемся списке **Format** редактора свойств. Задайте для заголовка выравнивание по центру. (Как видите, текст теперь будет выровнен по центру свободного элемента.) Измените размеры свободного элемента так, чтобы он занимал минимальное пространство страницы. Результат показан на рис. 11.15.

Продолжаем работу над страницей. Давайте создадим тень для нашего свободно позиционируемого заголовка. Для этого поместим под него еще один свободно позиционируемый элемент точно таких же размеров, но не имеющий содержимого и "залитый" черным цветом. Это обычная практика в такого рода случаях.

Рис. 11.15. Заголовок страницы default2.htm

Создадим данный элемент. Назовем его `Headershadow` и зададим ему черный цвет фона. После этого переключимся на элемент `Header` и зададим для него такой же цвет, как у фона самой Web-страницы, иначе расположенный ниже элемент-тень будет "просвечивать" сквозь него. Заодно установим параметр **Overflow** в **hidden**, чтобы размеры свободного элемента-заголовка случайно не изменились (Internet Explorer имеет такой грешок). Перепишем (или запомним) значения размеров, местоположения и z-индекса элемента-заголовка. Они потом нам очень пригодятся.

Теперь вернемся к нашему элементу-тени. Зададим для него такую же ширину (поле ввода **W**) и высоту (**H**), как и для элемента-заголовка, — это важно. А для горизонтальной и вертикальной координат (поля ввода, соответственно, **L** и **T**) зададим небольшое смещение (пиксела на четыре в большую сторону), ведь тень всегда немного сдвинута относительно "хозяина". Z-индекс (поле ввода **Z**) зададим меньшим, чем у элемента-заголовка. И посмотрим на результат — рис. 11.16.

Рис. 11.16. Готовый заголовок с тенью

Красиво, не так ли? Единственный недостаток: если вы хотите изменить размеры и (или) местоположение такого заголовка, вам также придется, соответственно, изменить размеры и (или) местоположение его тени.

Теперь давайте поместим на страницу начальный текст, содержащий приветствие. Это совсем простая задача. Создадим новый свободный элемент, назовем его `Content` и поместим под заголовком. После этого остается только скопировать в него со старой страницы сам текст приветствия. Результат см. на рис. 11.17.

Точно так же создадим еще один свободный элемент и поместим в него портрет нашего героя. Вновь созданный элемент назовем `Photo` и поместим его правее текста приветствия. Для портрета зададим достаточно маленькие размеры, чтобы он поместился на небольшом пространстве нашей странички. То, что у вас должно получиться в результате, показано на рис. 11.18.

Следующий элемент — полоса навигации. Но мы с вами не хотим создавать унылую, выровненную по линеечке, словно взвод солдат, замерших по стойке "смирно", полосу навигации. Давайте применим свою фантазию, прибавим к ней изумительные свойства свободно позиционируемых элементов и создадим что-нибудь пооригинальнее. А именно набор гиперссылок, "рассыпанных" по всей странице.

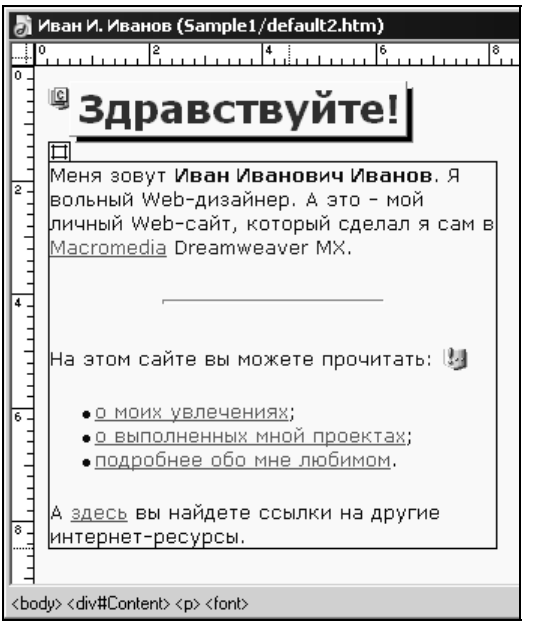


Рис. 11.17. Заголовок и текст приветствия

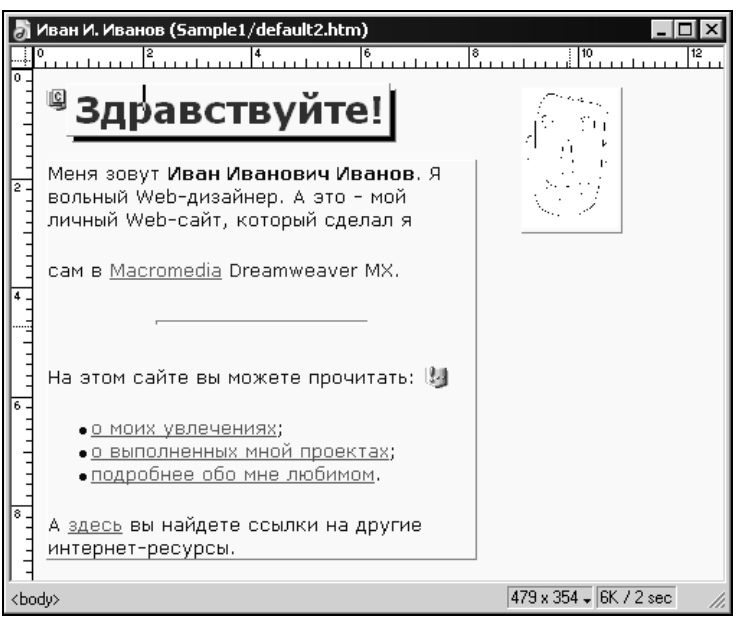


Рис. 11.18. Заголовок, текст приветствия и портрет

Первая такая гиперссылка будет вести на страницу списка увлечений нашего незабвенного Ивана Ивановича. А именно на страницу Passions.htm. Создадим небольшой новый элемент, назовем его `Passionslink` и зададим для него белый фон. После этого введем в него слово `Увлечения`, выровняем его по центру и превратим в гиперссылку. Результат показан на рис. 11.19.

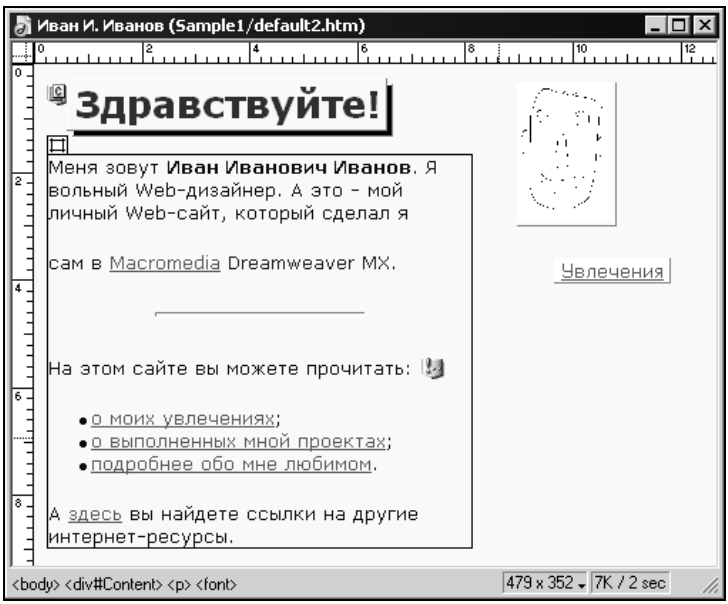


Рис. 11.19. Заголовок, текст приветствия, портрет и гиперссылка

Остальные гиперссылки делаются по образу и подобию первой. Это проще всего сделать так. Выделите границу первого свободного элемента гиперссылки, созданного нами, и скопируйте его в буфер обмена. После чего щелкните мышью где угодно на странице (не на свободно позиционируемых элементах), чтобы поставить на нее текстовый курсор, и выполните вставку из буфера обмена. Новый свободный элемент будет помещен прямо поверх старого, который мы и копировали; вам останется только изменить его местоположение, имя и, разумеется, содержимое.

Создайте таким образом все гиперссылки набора и разместите их на странице, назвав, соответственно, `Projectslink` (список завершенных проектов), `Linkslink` (набор ссылок на "дружественные" ресурсы) и `Aboutlink` (сведения об авторе сайта). В результате у вас может получиться что-то похожее на рис. 11.20.

Теперь настала очередь сведений об авторских правах. Поместим их внизу, прямо под текстом приветствия, а свободный элемент назовем `Copyrights`.

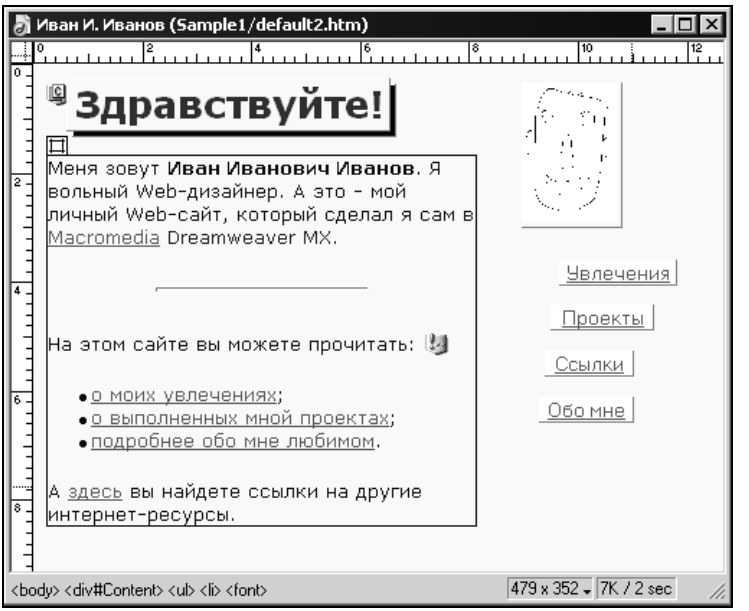


Рис. 11.20. Заголовок, текст приветствия, портрет и готовый набор гиперссылок

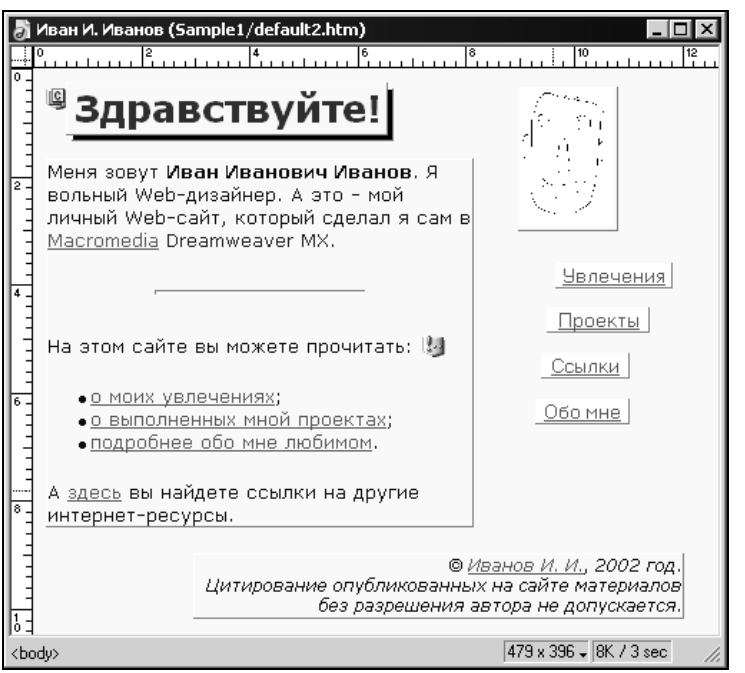


Рис. 11.21. Заголовок, текст приветствия, портрет, набор гиперссылок и сведения об авторских правах

Не забываем выровнять текст по правой стороне. Получится у нас вот что (рис. 11.21).

Вот и вся главная страница. Откройте ее в Web-обозревателе и посмотрите на окончательный результат.

При желании вы можете переделать весь сайт на основе свободно позиционируемых элементов. Назовите его Sample site 4 или еще как-нибудь. Таким образом, у вас будет уже четвертый сайт, созданный вашими собственными руками.

В этом случае вам сильно могут помочь шаблоны, описанные в *главе 9*. Создайте шаблон, содержащий заголовок, набор гиперссылок и сведения об авторских правах, поместите на нем пустой свободный элемент, где будет находиться содержимое страниц, и преобразуйте его в изменяемую область. После этого вам только останется создать на основе этого шаблона новую страницу и вписать в эту изменяемую область нужное содержимое.

Конечно, никто не заставляет вас использовать шаблоны. Вы можете сделать каждую страницу сайта непохожей на другие. Благо свободно позиционируемые элементы дают вам такую возможность.

Недостатки свободно позиционируемых элементов и их преодоление

Напоследок, по уже устоявшейся традиции, поговорим о недостатках свободно позиционируемых элементов. Итак, чем же они нам не угодили?

Прежде всего, свободно позиционируемые элементы предполагают статичный дизайн Web-страниц, не меняющийся при изменении размеров окна Web-обозревателя. Поясним это на примере. Когда вы просматриваете в Web-обозревателе обычную страницу, при изменении размеров окна, в котором она отображается, содержимое страницы также меняет размеры. Загрузите в Web-обозревателе страницу default.htm нашего первого сайта Sample site 1 и измените размеры его окна. Вы увидите, что ее содержимое "расползается" или "сужается" для того, чтобы максимально полно занять окно по ширине. А теперь загрузите только что законченную страницу default2.htm, попробуйте изменить размеры окна — и что же? Свободные элементы в любом случае остаются на своих местах, не делая попытки "подстроиться" под новые размеры. Они статичны, как гвозди, вколоченные в стену.

"Классические" Web-страницы почти всегда разрабатываются с определенным запасом "гибкости", позволяющей им подстраиваться под любой размер окна Web-обозревателя. Исключений из этого правила всего два: страницы, созданные на основе таблицы разметки фиксированной ширины, и... на основе свободно позиционируемых элементов.

В самом деле, свободные элементы всегда находятся на тех местах страницы, куда вы их поместили. Сдвинуть их с места средствами HTML абсолютно невозможно. Язык HTML не предназначен для того, чтобы описывать поведение элементов страницы при наступлении какого-либо события, например изменения размеров окна Web-обозревателя. HTML описывает внешний вид страницы — и только. Если же вы хотите, чтобы при изменении размеров окна менялся также и дизайн страницы, пишите сценарии на языке программирования JavaScript, помещайте их в HTML-код и привязывайте к соответствующим событиям. Но для этого нужно быть не только знатоком HTML и JavaScript, но и виртуозным программистом, да и в этом случае у вас может не все получиться.

Бесспорно, что в некоторых случаях статичность свободных элементов может быть даже полезна. Если вы разрабатываете стильный "авторский" дизайн, свободные элементы помогут вам сделать то, что надо, и сохранить все на своих местах. Также свободные элементы могут пригодиться для создания дизайна, напоминающего интерфейс Windows-приложений. И, разумеется, без свободных элементов не обойтись, если вы собираетесь оживить свои страницы анимацией. В других случаях, вероятно, лучше обойтись "классическими" Web-страницами.

Так, со статичностью разобрались. Какие же сюрпризы еще приготовили нам свободные элементы?

Несовместимость.

Как вы помните, свободно позиционируемые элементы возникли одновременно с каскадными таблицами стилей CSS. И программами Web-обозревателей они стали также поддерживаться одновременно: в 1997 году, начиная с четвертых версий Internet Explorer и Navigator. С более старыми версиями они не совместимы.

Что значит "несовместимы"? Не поддерживают. Содержимое свободно позиционируемых элементов они отобразят, конечно, правильно, только находиться оно будет в общем "потоке" текста. И уж, естественно, сложнейший дизайн, над которым столько дней и ночей корпел автор, исчезнет без следа. И ничего тут не сделаешь.

Конечно, в настоящее время подавляющее большинство программ Web-обозревателей, эксплуатируемых интернетчиками, составляют более-менее "свежие". Процент же старых программ настолько мал, что им можно пренебречь. Но вполне возможна такая ситуация, что вам придется делать страницы, которые обязательно должны поддерживаться всеми версиями этих программ (за исключением, может быть, самых уж древних и экзотических). Что делать в таком случае?

Самый очевидный путь — преобразовать дизайн Web-страниц, построенный на свободных элементах, в графическое изображение, простое или состав-

ное, состоящее из множества простых. Это графическое изображение может представлять собой либо набор изображений-гиперссылок (для составных изображений), либо карту-изображение (для простых). Однако этот путь подходит только для небольшого процента страниц, в основном тех, где свободные элементы содержат в себе только элементы дизайна (изображения, гиперссылки, редко меняющийся текст и т. п.). Если на вашей странице в свободных элементах помещаются тексты новостей или полезных советов, отбираемых из базы данных по случайному признаку, — согласитесь, этот путь не для вас.

Второй путь — преобразовать свободные элементы в таблицы разметки, т. е. использовать табличный дизайн. Это решает многие проблемы. (Если, конечно, свободные элементы не являются жизненно необходимыми, например, для создания анимации.) Благо Dreamweaver предоставляет для такого решения стандартные инструменты преобразования свободных элементов в таблицы и, наоборот, таблиц в свободные элементы.

Один из таких инструментов мы уже использовали в *главе 10*, когда преобразовывали стили CSS в HTML-разметку. Этот инструмент доступен после выбора пункта **3.0 Browser Compatible** подменю **Convert** меню **File**. На рис. 10.39 показано диалоговое окно **Convert to 3.0 Browser Compatible**, открывающееся после выбора названного пункта. В группе переключателей **Convert** мы рассмотрели только переключатель **CSS Styles to HTML Markup**, выполняющий упомянутое выше преобразование. А для преобразования свободных элементов в таблицы предназначен переключатель **Layers to Table**. Переключатель **Both** преобразует и то и другое.

Однако существует и другой инструмент, позволяющий выполнить более точное преобразование. Он появляется после выбора пункта **Layers to Table** в подменю **Convert** меню **Modify**. Точнее, не инструмент, а появляющееся при этом диалоговое окно **Convert Layers to Table**, показанное на рис. 11.22.

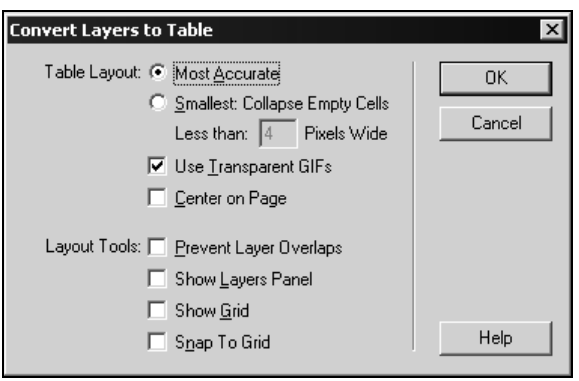


Рис. 11.22. Диалоговое окно **Convert Layers to Table**

Элементы управления группы **Table Layout** задают параметры таблицы разметки, которая будет сгенерирована Dreamweaver на основе исходного набора свободных элементов.

Переключатель **Most Accurate** включает режим максимально точного преобразования. При этом будут соблюдены все размеры и расстояния. В свою очередь, переключатель **Smallest: Collapse Empty Cells** предписывает Dreamweaver убрать из получившейся таблицы ячейки и строки, размеры которых меньше заданной в поле ввода **Less than Pixels Wide** величины. (Данная величина задается в пикселах.) При этом получившаяся таблица разметки может немного отличаться от того, что планировалось, но будет компактнее.

Включенный флажок **Use Transparent GIFs** предписывает Dreamweaver использовать в получившейся таблице разметки "распорки" из прозрачных однопиксельных GIF-изображений. В этом случае таблица разметки будет правильно отображаться любым браузером. Рекомендуется всегда оставлять этот флажок включенным.

Включенный флажок **Center on Page** заставляет Dreamweaver помещать получившуюся таблицу разметки в центре страницы.

Флажки группы **Layout Tools** задают вспомогательные параметры:

- **Prevent Layer Overlaps** — запрещает или разрешает свободным элементам перекрывать друг друга;
- **Show Layers Panel** — выводит на экран панель **Layers** или убирает ее;
- **Show Grid** — выводит на экран координатную сетку или скрывает ее;
- **Snap to Grid** — включает или отключает режим "прилипания" к линиям координатной сетки.

Преобразование выполняется после нажатия кнопки **OK**. Кнопка **Cancel** позволяет вам отказаться от преобразования.

Чтобы выполнить обратное преобразование — таблицу разметки в набор свободно позиционируемых элементов — используйте пункт **Tables to Layers** в подменю **Convert** меню **Modify**. После его выбора на экране появляется диалоговое окно **Convert Tables to Layers**, показанное на рис. 11.23.

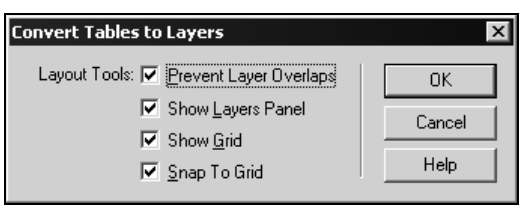


Рис. 11.23. Диалоговое окно **Convert Tables to Layers**

В этом окне находится только группа флажков **Layout Tools**, уже нам знакомая. Поэтому не будем описывать ее снова, а скажем только, что кнопка **OK** запускает преобразование, а кнопка **Cancel** позволяет от него отказаться.

Что дальше?

Вот мы и познакомились со свободно позиционируемыми элементами. И даже сделали одну Web-страничку на их основе.

И что с того? Все это можно было сделать традиционными путями, использовав таблицы разметки, а то и обычным текстом. Так какой смысл в применении свободных элементов?

А такой, что свободные элементы можно запросто перемещать по странице с помощью специально написанных программ-сценариев, т. е. создавать анимированные шоу, этакие мультики на Web-страницах.

Как это делается? Читайте дальше!

ГЛАВА 12

Анимация элементов Web-страниц

Одним из основных преимуществ свободно позиционируемых элементов является возможность управлять их местоположением, размером и видимостью с помощью специально написанных сценариев. Вы можете создавать простую анимацию на Web-страницах, заставляя свободно позиционируемые элементы перемещаться по заданным траекториям. Причем никакой модификации HTML-кода, кроме добавления соответствующих сценариев, не требуется (конечно, если соответствующие элементы на странице уже созданы).

Такая анимация может служить двум задачам. В простейшем случае она позволит оживить Web-страницы, сделать их привлекательнее, завлечь посетителя на сайт. (Количество посетителей — это единственная мера качества того или иного интернет-ресурса.) Также она позволит создать учебные пособия, которые можно выложить в Интернет или распространять на компакт-дисках. (Преподаватели школ и прочих учебных заведений — это для вас!) К несчастью, Web-дизайнеры мало используют такую интересную возможность.

Было сказано, что, кроме написания сценариев, собственно реализующих анимацию, и привязки их к определенным событиям, никакой особой переделки HTML-кода не требуется. Однако написание сценариев — задача, довольно сложная сама по себе. Мало того, что для этого нужно знать сам язык программирования JavaScript, следует хорошо представлять себе, как работают внутренние механизмы программы Web-обозревателя и как получить к ним доступ, чтобы использовать потом в своих нуждах. Кроме того, необходимо знать, как реализуется анимация, за счет чего элемент страницы будет казаться движущимся. И, конечно же, нужно иметь художественный вкус и — обязательно! — чувство меры.

Хорошо! С художественным вкусом и чувством меры проблем нет. Но есть проблемы с JavaScript и внутренними механизмами Web-обозревателя. Точ-

нее сказать, мы не знаем ни того, ни другого. И все же очень хотим создать на своих страницах какую-нибудь, хотя бы простенькую, анимацию. Что делать?

Воспользоваться Dreamweaver. Он предоставляет очень удобный способ создания анимированных элементов без углубления в JavaScript и темные недра Web-обозревателя. Вы просто указываете траекторию движения того или иного элемента, после чего задаете некоторые параметры — и сразу же получаете результат. Dreamweaver сам создает необходимые JavaScript-сценарии и помещает их в HTML-код страницы. Удобно, правда?

Давайте же выясним, как все это делается. Но не сейчас. В настоящее время нам нужно узнать кое-что еще. А именно о программах-сценариях, их назначении и о том, как создается анимация. Не пугайтесь: мы с вами не будем писать сценарии вручную (пусть это делает Dreamweaver), а просто поговорим об основных принципах, которые знать в любом случае невредно.

Итак...

Введение в Web-сценарии

Зачем нужны сценарии? Чтобы преодолеть ограничения HTML. Не все, конечно, — только одно.

Ограничения HTML

Так в чем же ограничивает нас HTML? Зачем понадобилось придумывать еще один язык? Нельзя ли обойтись без него?

В общем, можно. Обходились же мы на протяжении целых одиннадцати глав обычным HTML, и ничего! Но дело в том, что странички-то мы делали *неинтерактивные*, т. е. не взаимодействующие с пользователем, не меняющие свое содержимое в ответ на его действия. Для таких страниц хватит и "чистого" HTML. А как только вы захотите "научить" страницу взаимодействовать с пользователем, т. е. сделать *интерактивной*, вы сразу упретесь в непреодолимую стену.

А все потому, что HTML не позволяет задать поведение страницы в ответ, скажем, на щелчок мыши по одному из ее элементов. С его помощью задается только ее внешний вид. И этим он в корне отличается от *"классических"* языков программирования, описывающих как раз поведение.

Собственно, HTML — и не язык программирования, а язык описания Web-страниц — и не более. Он описывает, как должна выглядеть информация в окне Web-обозревателя, только и всего. Причем, этот внешний вид жестко стандартизирован; Web-дизайнер, как вы уже поняли, значительно сильнее ограничен в своих возможностях, чем дизайнер-полиграфист. Особых воль-

ностей на Web-страницах он не сделает. (Фреймы, таблицы разметки и каскадные таблицы стилей CSS снимают многие ограничения, но не все.)

Конечно, такие ограничения обусловлены благой целью — обеспечением совместимости и единообразия внешнего вида страниц на любой компьютерной платформе. Но кому нужна совместимость такой ценой!

Вы можете сказать, что любая Web-страница и так является интерактивной, т. к. реагирует на щелчки по гиперссылкам. Но это не интерактивность, а стандартное поведение, заданное HTML. Интерактивность — это когда в ответ на перемещение курсора мыши за ним "бежит" картинка, когда при щелчке на фотографии внизу появляется ее описание, когда прямо на страничке "тикают" цифровые часы. Вот это настоящая интерактивность: в первом и втором случаях страница реагирует на действия пользователя, а в третьем — на события внешней среды. (Здесь под внешней средой подразумевается операционная система; работающий в ней системный таймер заставляет идти нарисованные часы.) И обычный HTML не может ее обеспечить.

Чтобы преодолеть этот недостаток "чистого" HTML, были придуманы два пути.

Первый путь — это специальные, как правило, нестандартные, "фирменные", теги, расширяющие возможности языка. В подавляющем большинстве случаев они добавляют Web-страницам интерактивные возможности. Например, нестандартный парный тег `<MARQUEE>...</MARQUEE>`, поддерживаемый Internet Explorer, позволяет создать на странице "прокручивающийся" текст. Но, согласитесь, эти расширения ограничивают Web-дизайнера теми возможностями, что вложили в них разработчики Web-обозревателя. И вы не сможете заставить текст в элементе `<MARQUEE>` не прокручиваться, а последовательно менять цвета — разработчики Internet Explorer этого не предусмотрели.

Мы хотим сами задавать поведение элементов страниц в ответ на действия пользователя. Мы хотим сами контролировать их поведение, держать все нити управления в своих руках. HTML их нам предоставить не может; ограничения, обусловленные самой его природой, — стена, преодолеть которую невозможно. Однако, если стену нельзя перепрыгнуть, ее можно обойти. Так мы и сделаем.

Web-сценарии

Второй путь — внедрение в HTML-код особых программ, написанных на "классических" языках программирования, т. е. описывающих поведение элементов. Такие программы называются *сценариями* (или *скриптами* от англ. *script* — сценарий). (Почему они так называются? Можете считать, что эти программы описывают сценарии поведения какого-нибудь элемента при

наступлении того или иного события.) Эти программы помещаются в HTML-код с помощью особого, специально для такого случая стандартизированного комитетом WWWC парного тега `<SCRIPT>...</SCRIPT>`. Web-обозреватель, "расшифровывая" такой HTML-код, читает сценарии и либо выполняет их сразу же, либо "откладывает в долгий ящик", чтобы выполнить после наступления какого-либо события.

Именно после "открытия" второго пути в интерактивность Всемирная паутина стала "живой", а многие Web-дизайнеры стали по совместительству еще и Web-программистами.

Для того чтобы писать сценарии, были специально созданы несколько языков программирования, называемых *языками написания сценариев* (или *скриптовыми языками*). Наибольшую популярность из них получили всего два: JavaScript и VBScript.

Язык *JavaScript* был разработан в фирме Netscape на основе известного "классического" языка программирования *Java*. Поэтому впервые поддержка языка JavaScript появилась именно в Web-обозревателе Netscape Navigator 2.0. Microsoft Internet Explorer начал поддерживать JavaScript с версии 3.0. Кроме того, в фирме Microsoft был разработан свой диалект JavaScript под названием *JScript*, отличающийся от "прародителя" только наличием нескольких новых команд. Поэтому эти два языка очень часто не различают, называя их одинаково — JavaScript.

Язык *VBScript* был разработан в Microsoft в качестве "адекватного ответа" на JavaScript. Этот язык поддерживается только Internet Explorer, начиная с версии 4.0. Из-за своей "несовместимости" VBScript не получил большого распространения в Web, но завоевал прочные позиции в серверном программировании, а именно, написании активных серверных страниц Microsoft ASP *(см. главу 15)*. В настоящее время для написания Web-сценариев он практически не применяется.

Кроме JavaScript и VBScript, были попытки использования для написания Web-сценариев языков Perl, Python и некоторых других. Однако сколько-нибудь широкого распространения эти языки не получили.

В настоящее время практически везде для написания сценариев применяется язык JavaScript. Он достаточно прост для изучения и предоставляет разработчику весьма обширные возможности. Кроме того, написанные на нем программы невелики по объему, что актуально в эпоху безраздельного господства медленных каналов связи.

Для поддержки языка сценариев используется так называемый *интерпретатор* (или *виртуальная машина*) языка. Web-обозреватель выделяет сценарии из HTML-кода страницы и передает его на выполнение интерпретатору. Последний анализирует код сценариев, расшифровывает их и выполняет, передавая результат выполнения обратно Web-обозревателю.

Для каждого языка написания сценариев, поддерживаемого Web-обозревателем, нужен свой отдельный интерпретатор. Благодаря такому подходу Web-обозреватель можно "научить" поддерживать сценарии, написанные практически на любом языке программирования, если для этого языка существует интерпретатор, написанный определенным образом.

Языки сценариев называются *интерпретируемыми*, т. к. написанные на них программы расшифровываются при каждом выполнении. Этим они коренным образом отличаются от *компилируемых* языков (C++, Pascal, Java и многих других). Написанные на компилируемых языках программы расшифровываются однократно и преобразуются в машинный код, который выполняется самим процессором компьютера, без помощи интерпретатора. Для такого преобразования используются специальные программы, называемые *компиляторами*. Конечно, откомпилированный код выполняется значительно быстрее, но специфика Web-программирования такова, что в нем могут применяться только интерпретируемые языки.

Ну, вот и все о языках программирования, на которых пишутся Web-сценарии. Теперь поговорим о том, как эти Web-сценарии выполняются.

События

Событие в системе происходит при наступлении некого условия, внешнего или внутреннего. В зависимости от того, внутри или вне системы наступило это условие, различают *внешние* и *внутренние* события. К внешним событиям относятся щелчок мышью на странице или ее элементе, перемещение курсора мыши, нажатие клавиши на клавиатуре, изменение размеров окна Web-обозревателя и т. п. Внутренние события — это завершение загрузки Web-страницы, очередной "тик" системного таймера и т. п. Эта классификация событий сделана только для удобства обучения начинающих программистов, ведь с точки зрения языка сценариев оба вида событий между собой не различаются и обрабатываются одинаково.

Если брать во внимание только события, происходящие в Web-странице, то нужно иметь в виду, что события могут происходить как в самой странице, так и в отдельных ее элементах. В самом деле, если пользователь щелкает мышью на части страницы, не занятой какими-либо элементами, говорят, что событие происходит в самой странице. Если же пользователь щелкнет на графическом изображении или текстовом абзаце, событие происходит в этом графическом изображении или текстовом абзаце, т. е. сообщение о наступлении события "щелчок левой кнопкой мыши" в первом случае получит сама страница, а во втором — один из ее элементов.

Что же нам делать с этими событиями, спросите вы? Обрабатывать.

Обработка события заключается в реакции на них. А для реакции на них используются особым образом написанные сценарии, называемые *обработчиками событий*. Обработчики *привязываются* к тем или иным событиям,

происходящим в каком-либо элементе страницы или самой странице. И делается это с помощью особого расширения HTML, специально предназначенного для поддержки Web-сценариев.

В частности, если требуется, чтобы при наведении курсора мыши на изображение внизу появлялось его краткое описание, нужно сделать следующее. Под изображением помещается свободно позиционируемый элемент, содержащий текст описания, и этот элемент делается невидимым. Далее пишутся два сценария, привязанные к событиям, возникающим при наведении курсора мыши на графическое изображение и при "уводе" его прочь. Первый из этих сценариев делает элемент с описанием видимым, второй — невидимым. После этого написанные сценарии привязываются к соответствующим событиям, происходящим в элементе графического изображения. Вот и все.

Не будем рассматривать конкретные детали реализации этого эффекта. Во-первых, они слишком зависят от выбора языка программирования, а первый (и, кстати, последний в этой книге) сценарий JavaScript мы напишем и изучим в *главе 13*. Во-вторых, в данный момент требуется анимировать элемент страницы, т. е. заставить его двигаться.

В большинстве интерактивных Web-страниц большая часть сценариев представляет собой как раз обработчики событий. Именно они реализуют большую часть логики поведения страницы при взаимодействии с пользователем. (Хотя существует определенный процент сценариев, выполняемых сразу же после расшифровки; в основном, это код, производящий всяческие предустановки.) Такой подход, когда почти вся программа представляет собой обработчики событий, называется *событийно-управляемым программированием*.

Но мы отвлеклись. Давайте все же вернемся к нашей анимации. А разговор о программировании продолжим в *главе 13*, посвященной как раз Web-сценариям и работе с ними в среде Dreamweaver.

Основные принципы анимации

Наконец-то дошла очередь и до анимации. Рассмотрим некоторые базовые понятия, без которых в дальнейшем нам не обойтись.

Не раз уже говорилось, что анимировать можно только свободно позиционируемые элементы. Не забывайте это. Только свободные элементы могут быть помещены в любое место страницы, и лишь они могут управляться из сценариев. Запомните это.

Простейшая анимация

Итак, за счет чего же достигается эффект анимации? Почему элемент страницы кажется движущимся по ней?

Дело в том, что координаты элемента периодически меняются. И меняются весьма быстро, не реже 12 раз в секунду, за счет чего мы и наблюдаем непрерывное движение. Такая частота выбрана оттого, что именно на ней человеческий глаз теряет способность различать отдельные приращения, "скачки" прерывистого движения. Короче говоря, за этим пределом прерывистое движение становится для человека визуально непрерывным. (На этом же принципе, кстати, работают кинематограф и телевидение.) Компьютеры у нас сейчас достаточно мощные, так что обеспечить такую частоту смены координат вполне реально.

Но мало всего лишь изменить координаты элемента. Нужно изменить их по особому закону, описывающему траекторию движения элемента. Если элемент движется по прямолинейной траектории, этот закон очень прост. В случае криволинейного движения он сильно усложняется. Поэтому не стоит создавать на своих страницах слишком сложные анимации: маломощный компьютер просто не "потянет" чересчур замысловатые траектории.

Закон, по которому изменяются координаты анимированного элемента, иначе говоря, его траектория, описывается особой функцией (назовем ее *функцией траектории*). Такая функция принимает некоторые параметры и возвращает координаты элемента. Она имеет вид:

```
{x, y, z} = f(Q, q, dq)
```

С возвращаемыми этой функцией результатами все просто. `x`, `y` и `z` — координаты анимированного элемента, соответственно, горизонтальная, вертикальная и порядок перекрытия (уже знакомый вам z-индекс). Конечно, функция, возвращающая сразу три координаты, — это общий случай. Обычно изменяется одна или две координаты элемента — `x` и `y`.

А вот с параметрами, принимаемыми функцией, все несколько сложнее. Всего, как вы видите, их три, и назначение их неочевидно. Давайте же рассмотрим все эти параметры по порядку.

Самый первый параметр — `Q`. Это *длина траектории* движения анимированного элемента. Она может измеряться в пикселах, миллиметрах, градусах или каких-либо относительных единицах, например процентах. Вообще, единица измерения зависит от самой траектории: для прямолинейной больше подойдут пикселы или миллиметры, а для круговой — градусы или радианы. Важно одно: параметр `Q` должен каким-то образом обозначать полную длину траектории, по которой будет двигаться наш элемент.

Второй параметр (`q`) обозначает *позицию на траектории*, занимаемую в данный момент времени нашим элементом. Иными словами, это расстояние в единицах измерения траектории `Q`, которое он уже "пробежал". Получив его, наша функция должна сказать: "элемент здесь" и, что называется, "ткнуть пальцем" в точку на его траектории.

Последний, третий параметр — `dq`. Он задает *приращение*, на которое будет меняться дистанция `q` при каждом "скачке" анимированного элемента. Ве-

личина этого параметра задает скорость движения анимированного элемента.

Итак, что же должна делать функция траектории. Ниже перечислены все ее задачи в порядке выполнения.

1. Принять начальные параметры и выполнить предварительные установки (прежде всего, установить анимированный элемент в *начальную точку* траектории движения).
2. Вычислить значения координат анимированного элемента на основании значения `q` и переместить этот элемент в точку с вычисленными координатами.
3. Проверить, дошел ли элемент до *конечной точки* траектории. Для этого `q` сравнивается с `Q`, и, если они равны либо `q` больше `Q`, анимация останавливается.
4. Увеличить значение `q` на величину `dq`.
5. Перейти к шагу 2.

Это общий алгоритм, которому должны следовать все функции траектории, даже самые сложные. Фактически основная сложность заключается в реализации шага 2, т. е. вычислении координат на основе значения `q`. Остальные шаги реализуются очень просто и не составят трудности даже для начинающего программиста.

В разговоре о простейшей анимации подразумевалось, что существует одна-единственная функция, реализующая ее. Реально для создания анимации может использоваться несколько программ-сценариев. Автор объединил их в одну функцию, чтобы вам было понятнее.

Предполагалось, что когда анимированный элемент достигает конца траектории (`q` становится равной или больше `Q`), анимация останавливается. На самом деле, функция траектории может продолжить работу. Она может, например, просто снова выполнить начальные установки, поместив анимированный элемент в начало траектории, и запустить анимацию снова. Но лучше всего инвертировать значение `dq` (изменить положительное значение на отрицательное и наоборот) и пустить анимацию "задом наперед". Такая анимация, проигрывающаяся несколько или бесконечное количество раз, называется *зацикленной*.

Конечно, зацикленная анимация выглядит эффектнее. Но злоупотреблять ей не стоит, особенно бесконечной. Почему — вы сейчас узнаете.

Анимация реального времени

Мы только что рассмотрели способ создания на Web-странице простейшей анимации. Но дело в том, что в реальной жизни такая анимация применя-

ется крайне редко. Более того, автор категорически не рекомендует вам даже пытаться ее сделать.

Почему? Дело в том, что вышеописанный способ создания анимации имеет только одно более чем сомнительное достоинство и два огромнейших недостатка.

Достоинство — простота, более того, очевидность реализации. Любой программист, даже малоквалифицированный, может сесть и в пять минут накропать сценарий, реализующий движение элемента по прямолинейной траектории. Да, он это сделает, даже если до этого ни разу не занимался Web-программированием.

Но пусть он только попробует выложить свое творение в Сеть!

Давайте представим двоих интернетчиков, решивших посетить его страничку с анимацией. У одного компьютер десятилетней давности, старенький, заслуженный, едва-едва обрабатывающий современные сложные Web-странички, даже без мультимедийных "наворотов". У другого — мощнейшая наисовременнейшая машина, только что сошедшая со сборочных столов какого-нибудь известного "брэнда". Вот они заходят-таки на страничку, загружают ее и принимаются наблюдать за анимацией.

Компьютер первого посетителя с натугой загрузит ее и, треща жестким диском, начнет ни шатко ни валко выполнять сценарии, реализующие анимацию. Анимированный элемент худо-бедно движется по странице, посетитель доволен — работает. Теперь оставим его и взглянем на посетителя номер два. Его компьютер мгновенно загрузит страничку и мгновенно же выполнит все сценарии. Анимированный элемент пролетит по своей траектории так быстро, что человеческий глаз его движения даже не заметит.

Спрашивается, кому нужна такая анимация?

Но это полбеды. Беда настанет, если наш незадачливый программист решит сделать анимацию зацикленной, бесконечной. Теперь смотрите, что получится. Страница загружается в Web-обозревателе, интерпретатор начинает выполнять сценарии... и выполняет... выполняет... выполняет... А пока он их выполняет, пользователь ничего не сможет сделать со страничкой: ни щелкнуть по гиперссылке, ни даже прокрутить ее в окне. Единственный способ прервать затянувшееся "кино" — закрыть сам Web-обозреватель.

Но что делать? Ведь Web-программисты как-то реализуют анимацию на своих страницах, и она работает нормально, в смысле, не мешает пользователю прокручивать страницу и щелкать по ссылкам. Может, они знают какой-нибудь секрет?

Да, знают. И заключается этот секрет в том, что они используют немного другую функцию траектории:

```
{x, y, z} = f(Q, q, dq, t)
```

От уже знакомой нам функции она отличается тем, что принимает еще один параметр — t. Этот параметр — время. Иначе говоря, при расчете координат функция траектории учитывает текущее время. И анимация оказывается жестко привязанной к времени, которое везде течет одинаково, в отличие от тактовой частоты процессоров, которые различаются у разных компьютеров.

Как это реализуется? Дело в том, что новая функция выполняется не все время, пока работает анимация, а вызывается время от времени, тогда, когда нужно произвести очередной "скачок" анимации, и после этого прекращает свою работу, дожидаясь очередного вызова. Эта функция реализуется в виде сценария-обработчика внутреннего события — "тика" *системного таймера*.

Таким образом, анимация оказывается четко привязанной к времени. И на стареньком компьютере посетителя номер один, и на суперсовременной машине номера два анимированный элемент будет двигаться с одной и той же скоростью. (Конечно, на более мощном компьютере анимация, возможно, будет выполняться плавнее, но с той же скоростью, что и на более мощном.)

Далее. Так как функция траектории вызывается только время от времени, а не работает постоянно, пользователь может нормально взаимодействовать с Web-страницей. Web-обозревателю между "тиками" таймера остается предостаточно времени, чтобы обработать пользовательские запросы.

Давайте приведем список задач новой функции траектории в порядке выполнения.

1. Принять начальные параметры и выполнить предварительные установки (установить анимированный элемент в начальную точку траектории движения, запустить системный таймер и привязать к его событиям сценарий-обработчик).
2. Вычислить значения координат анимированного элемента на основании значения q и переместить этот элемент в точку с вычисленными координатами.
3. Проверить, дошел ли элемент до конца траектории. Для этого q сравнивается с Q, и, если они равны либо q больше Q, выполняется переход к шагу 5.
4. Увеличить значение q на величину dq.
5. Остановить системный таймер и "отвязать" от его событий обработчик.

Для реализации этой функции используются два сценария. Один из них выполняется при загрузке страницы и реализует задачу шага 1. Второй — собственно обработчик событий системного таймера — реализует задачи шагов 2—5.

Такая анимация, привязанная к системному таймеру, а не к процессору, называется *анимацией реального времени*. Именно она и применяется для создания движущихся элементов на Web-страницах. Простейшая анимация, описанная нами ранее, практически нигде не используется. Вы уже поняли, почему.

Анимация — подход Dreamweaver

Описанный выше способ создания анимации с помощью сценариев, реализующих функцию траектории, обладает множеством достоинств. Такие сценарии имеют очень малый размер и очень быстро выполняются, т. к. программист имеет возможность написать функцию траектории наиболее оптимальным способом. Однако у них есть один огромный недостаток — негибкость. Фактически для изменения траектории движения анимированного элемента нужно писать новый сценарий, реализующий новую функцию траектории.

Но негибкость — не единственный недостаток этого способа создания анимации. Если анимированный элемент должен совершать достаточно сложное движение, реализующий эту траекторию сценарий получается очень большим и медленным. И чем сложнее траектория, тем больше и медленнее получается сценарий.

Третий недостаток — сложность реализации сложных траекторий. Извините за тавтологию, но это так. Малоопытные программисты, особенно не имеющие серьезной математической подготовки, обычно ограничиваются простенькими траекториями, как правило, прямолинейными. Максимум, на что они отваживаются, — это простой эллипс. А ведь для некоторых задач, которые будут рассмотрены ниже, нужны как раз довольно сложные траектории движения. И обойтись простыми траекториями весьма проблематично.

Однако и из этого положения есть выход. На помощь неопытным программистам, не владеющим высшей математикой, придет новый вид функции траектории, принимающей всего два параметра: *массив ключевых точек траектории* и, естественно, текущее время:

```
{x, y, z} = f([p1, t1, p2, t2..., t])
```

Насчет времени все понятно, но что такое массив ключевых точек? Ничего сложного: это набор точек, с помощью которого задается траектория движения нашего анимированного элемента. Поясним это более подробно.

Предположим, что нам нужно создать очень сложную анимацию, когда элемент страницы движется по весьма причудливой траектории. При этом высшей математикой мы не владеем, поэтому вывести формулу этой траектории не сможем. Однако мы придумали вот что.

Можно сделать немного по-другому. Прежде всего, пометим и пронумеруем ключевые точки прямо на нарисованной траектории. Далее отдельно прове-

дем по линейке прямую и проставим на ней координатную шкалу времени, проградуировав ее, скажем, в секундах. Отметим вдоль этой шкалы напротив соответствующих делений точки начала и конца анимации, проведем между ними линию потолще и назовем ее *дорожкой анимации*. После этого останется только пронумеровать ключевые точки и проставить их на дорожке напротив соответствующих отметок времени. В результате мы получим набор точек, координаты и время прохождения которых можно легко вычислить (соответственно, на нарисованной схеме и временной шкале). Согласитесь, так много нагляднее.

Мы начертили траекторию движения нашего элемента на миллиметровой бумаге, после чего выделили на данной траектории нужное количество *ключевых точек*, которые ее, собственно, и создают (точки перегиба, начало и конец траектории и т. п.), и выписали их координаты на отдельную бумажку. Далее, зная время, за которое анимированный элемент пройдет всю траекторию, мы можем выяснить, в какой момент времени он появится в той или иной ключевой точке. Осталось только выписать эти значения времени на ту же бумажку, проставив их напротив координат соответствующих точек. Вот мы и получили массив ключевых точек, который можно передать новой функции траектории.

Этот подход и используется Dreamweaver.

Для примера мы привели анимацию, включающую один-единственный анимированный элемент. Но Dreamweaver позволяет помещать на временную шкалу сразу нужное количество элементов, создавая несколько анимационных дорожек, что может быть использовано для создания очень сложных анимаций, в которых все элементы движутся согласованно. Более того, Dreamweaver позволяет создавать на одной странице несколько работающих независимо друг от друга анимаций, каждая из которых может включать в себя любое количество анимированных элементов. Так что даже самый привередливый аниматор будет доволен.

Функция траектории, принимающая в качестве параметра массив ключевых точек, очень сложна, но нам и не нужно ее реализовывать. Многие мощные Web-редакторы предоставляют пользователям возможность создания анимации и используют для этого как раз такой подход. Набор Web-сценариев, реализующих функцию траектории, в этом случае вставляется в HTML-код страницы автоматически, пользователь даже не догадывается об этом.

Конечно, Dreamweaver не исключение. Он тоже предоставляет пользователям такую возможность. И также делает всю черновую работу сам.

Недостатком такого способа создания анимации являются большой размер и не очень высокое быстродействие получившегося кода. Поэтому для простейших анимаций, наверное, будет оправдан все-таки первый подход: специально написанные сценарии, непосредственно реализующие функцию траектории.

Ну вот, с техническими вопросами мы разобрались. Осталось выяснить, в каких случаях можно применять анимацию, а в каких — лучше воздержаться.

Зачем нужна анимация

Как правило, анимация на Web-страницах преследует три цели:

- оживить страницы;
- привлечь к чему-либо внимание;
- показать что-либо в учебных целях.

Поэтому можно выделить три цели применения анимации:

- развлечение;
- реклама;
- образование.

Все остальные цели, на взгляд автора, можно свести к трем вышеприведенным. А о трех вышеперечисленных мы поговорим подробнее.

Развлечения в Интернете — достаточно молодая отрасль Web-строительства. Изначально Интернет был создан как сеть для ученых, которым нужно было обмениваться текстовыми документами (сначала даже без графики) и связывать их в некое подобие структуры. Потом в Сеть пришел обыватель, и Web-дизайнеры ринулись угождать его вкусам. (Не будем спорить, насколько они возвышенны или низменны. Личное мнение автора: обыватель слишком разнолик, чтобы свести его к одному-единственному ярлыку.) В Интернете появились аудио и видео, на Web-страницы пришли сложная графика и анимация. Сейчас все эти "навороты" используются так часто, что ими уже мало кого можно удивить.

Умелое и умеренное использование анимации значительно оживит ваши страницы. Важно только понять, какого эффекта вы хотите достичь, и сделать все для того, чтобы он был достигнут. Не переусердствуйте — применяйте анимацию только там, где она действительно нужна. И уж, не дай бог, ваша страница будет рябить, как экран телевизора с отключенной антенной, — в этом нет ничего хорошего. Никогда не забывайте принцип "содержимое превыше всего, все остальное — украшения", не позволяйте украшениям заслонить содержимое.

Теперь реклама. Она появилась в Сети вместе с развлечениями, а значит, вместе с обывателем. Она уже здорово надоела, эта интернет-реклама, едва ли не больше, чем реклама телевизионная. Но отдадим должное рекламе (и интернетовской, и телевизионной) — благодаря ей получили возможность существовать очень многие популярные некоммерческие проекты. Если реклама вдруг исчезнет, эти проекты пропадут сразу же вслед за ней.

Традиционно для рекламных целей в Сети используются так называемые *баннеры* — небольшие графические изображения жестко стандартизированных форматов. Почти все баннеры создаются в формате "анимированный GIF", т. е. уже используют возможности анимации по привлечению внимания потенциального клиента. Рекламную анимацию, основанную на Web-сценариях, похоже, никто еще не применял, а если и использовал, то очень мало. Так что перед вами обширное непаханое поле деятельности, которое вполне может принести неплохие плоды.

Анимация, основанная на свободно позиционируемых элементах и Web-сценариях, значительно компактнее любых видеофайлов, будь то анимированный GIF, видеофайлы форматов AVI или Apple QuickTime. Однако анимированные GIF-файлы поддерживаются абсолютно всеми Web-обозревателями, даже самыми старыми, поэтому и применяются так широко. Web-сценарии же будут работать только на достаточно новых Web-обозревателях (которыми, надо сказать, сейчас пользуется подавляющее большинство интернетчиков).

Так или иначе, но попробовать Web-сценарии в рекламе сто́ит. И, кажется, кое-кто уже пробует.

От рекламы плавно перейдем к образованию. Программы — учебные пособия, часто используемые в образовании, как правило, пишутся на компилируемых языках программирования и представляют собой обычные MS-DOS- или Windows-приложения. Учебные пособия, сделанные на основе "живых" Web-страниц, встречаются пока еще довольно редко, хотя это направление весьма перспективно. Такие учебные пособия можно очень быстро создавать и модифицировать; а по сравнению с обычными программами они исключительно компактны (ну сколько места могут занимать несложная Web-страница и пара изображений?). Вдобавок такие пособия прямо-таки просятся в Интернет, на Web-сайты, а значит, можно без особых проблем организовать модное ныне дистанционное обучение.

Правда, данные учебные пособия имеют один огромный недостаток. Любой достаточно опытный интернетчик без труда сможет просмотреть исходный код самой страницы и ее Web-сценариев и, при желании, позаимствовать оттуда парочку ваших находок. Поэтому, если вы собираетесь применять в учебных пособиях какие-либо ноу-хау, лучше реализуйте их в обычной, откомпилированной программе, взломать которую гораздо труднее.

Вот три случая, когда анимация будет очень полезна. А в ряде случаев и необходима. Если же ваш случай к ним не относится, лучше обойтись обычными Web-страничками, больше внимания уделив их содержимому.

Создание анимации в Dreamweaver

Вот и пришла пора нашей любимой программы — Dreamweaver. Сейчас будет рассказано, как она поможет в создании анимированного элемента. А в качестве элемента, который мы будем "оживлять", возьмем заголовок страницы default2.htm, созданной в *главе 11*.

Но, прежде всего, условимся о терминологии, используемой в Dreamweaver. Знание ее поможет нам в дальнейшей работе. Итак:

- набор анимированных элементов, чье движение синхронизировано относительно одной и той же временной шкалы, назовем *анимацией*. Это нужно нам, потому что одна Web-страница может содержать несколько независимых анимаций, т. е. наборов анимированных элементов, синхронизированных относительно независимых временных шкал;
- полоска, проходящая из точки начала анимации одного из элементов в точку ее конца, пусть называется *дорожкой анимации*. Этот термин мы уже ввели и теперь его закрепим. Дорожка показывает, когда анимированный элемент начнет двигаться и когда он остановится;
- свободное пространство на временной шкале анимации, которое может быть занято дорожкой, будет называться *каналом*. Количество доступных каналов определяет максимальное количество дорожек в одной анимации, а значит, максимальное количество входящих в нее анимированных элементов.

Вот и все термины. Теперь откроем страницу default2.htm в окне документа и начнем...

Создание простейшей анимации

Для работы нам сразу же понадобится *панель Timelines*, в которой отображаются все анимации, созданные на Web-странице. Чтобы вывести эту панель на экран, включите пункт-выключатель **Timelines** в подменю **Others** меню **Window** или нажмите комбинацию клавиш <Alt>+<F9>. Сама панель **Timelines** показана на рис. 12.1.

Как видите, панель **Timelines** находится в еще одном доке, занимающем нижнюю часть главного окна Dreamweaver. Это значит, что мы можем скрыть панель **Timelines** (и любые другие панели, которые вы поместите в док), если она нам не нужна, щелкнув по кнопке скрытия дока, а потом вернуть ее на экран.

Большую часть этой панели занимает временная шкала, но проградуированная не в секундах, а в кадрах анимации — так удобнее. Значения времени (в кадрах) написаны на серой *временной шкале*, расположенной выше. Под этой шкалой отображаются все доступные каналы анимации, и заня-

тые, и незанятые. В верхней же части панели находятся несколько элементов управления, предназначенных для задания некоторых параметров дорожек и самой анимации.

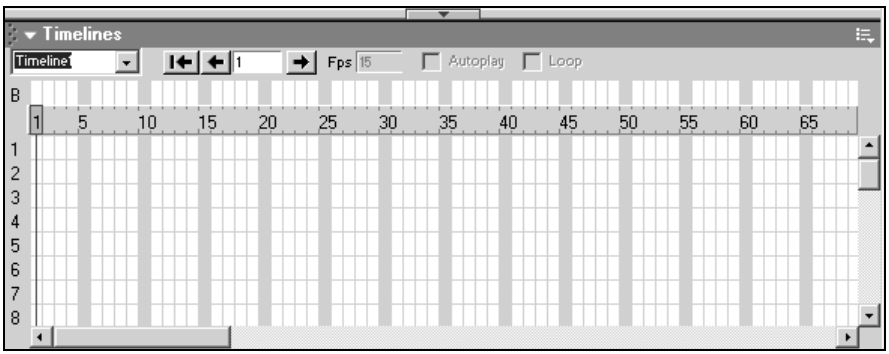


Рис. 12.1. Панель **Timelines**

"Но ведь Dreamweaver позволяет работать одновременно с несколькими временными шкалами, — скажете вы. — Где же остальные шкалы?" Чтобы увидеть их, нужно выбрать соответствующий пункт комбинированного списка анимаций, показанного на рис. 12.2.

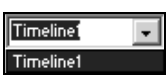


Рис. 12.2. Комбинированный список анимаций

Работой с несколькими разными анимациями мы займемся потом. А сейчас давайте создадим одну из них, самую простую.

Чтобы анимировать какой-либо из доступных свободно позиционируемых элементов, нужно поместить его на шкалу времени, создав дорожку анимации. Для этого выделим требуемый свободный элемент, щелкнем по нему правой кнопкой мыши и выберем в контекстном меню пункт **Add to Timeline**. Вы также можете выбрать пункт **Add Object** контекстного меню временной линии или пункт **Add Object to Timeline** подменю **Timeline** меню **Modify**. И, наконец, вы можете просто нажать комбинацию клавиш <Ctrl>+<Alt>+<Shift>+<T>.

После добавления в анимацию свободного элемента Dreamweaver выводит предупреждение, показанное на рис. 12.3. Этим он хочет сказать, что может управлять в процессе анимации следующими параметрами свободного элемента:

- горизонтальной и вертикальной координатами левого верхнего угла (атрибуты `LEFT` и `TOP`);

- шириной и высотой (атрибуты `WIDTH` и `HEIGHT`), причем работать это будет только в Internet Explorer;
- порядком перекрытия, иначе говоря, z-индексом (атрибут `Z-INDEX`);
- видимостью (атрибут `VISIBILITY`).

Рис. 12.3. Предупреждение, выводимое Dreamweaver после добавления свободного элемента в анимацию

Это значит, что вы можете управлять местоположением, размерами и видимостью свободного элемента. Как видите, возможностей, предлагаемых Dreamweaver для анимации, даже больше, чем мы думали.

Теперь закройте данное предупреждение нажатием кнопки **OK**. Если вы не желаете больше его видеть, перед закрытием включите флажок **Don't show me this message again**.

Наконец, после всех треволнений, мы увидим в списке панели **Timelines** новую дорожку (рис. 12.4). Она отображается в виде светло-синей полосы, на которой написано имя свободного элемента, которому она принадлежит. По обеим сторонам данной полосы можно заметить светлые кружки. Это ключевые точки; их пока всего две: начало и конец траектории. Первая ключевая точка — начало — находится на первом кадре; это значит, что анимация для данного элемента начнется с первого кадра. Вторая ключевая точка — конец траектории — находится на пятнадцатом кадре; там наш анимированный элемент перестанет двигаться.

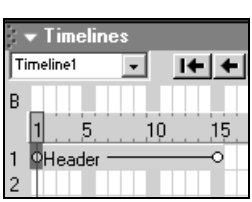


Рис. 12.4. Новая дорожка анимации

Вы можете выбирать как ключевые точки (в данном случае они выделяются темно-синим цветом), так и саму дорожку (в этом случае она вся становит-

ся темно-синей), просто щелкая по ним мышью. Более того, вы можете щелкнуть по любому месту на дорожке анимации, выделив таким образом какой-либо ее кадр. В этом случае на временной шкале появится *маркер выделенного кадра*, показанный на рис. 12.5.

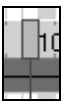


Рис. 12.5. Маркер выделенного кадра

Если вас не устраивает местоположение дорожки анимации, вы можете легко его изменить. Для этого "ухватите" мышью дорожку (не ключевые точки!) и перетащите ее вдоль шкалы, пока она не займет нужную позицию. Например, вы можете заставить анимацию начинаться с десятого кадра, а заканчиваться — двадцать пятым. Также вы можете изменять длину дорожки, а значит, продолжительность анимации, перетаскивая на необходимую позицию теперь уже ключевые точки. И, разумеется, вы можете удалить ненужную дорожку, выделив ее и нажав клавишу <Del> или выбрав пункт **Remove Object** контекстного меню или подменю **Timeline** меню **Modify**.

Ну что ж, давайте проверим свежесозданную анимацию. Нажимаем клавишу <F12>, чтобы загрузить страницу default2.htm в Web-обозревателе, и... наблюдаем совершенно неподвижное изображение. В чем же дело?

А дело в том, что начало и конец траектории движения нашего анимированного заголовка совпадают, т. е. никакой траектории нет, заголовок все время стоит в одной точке! Мы забыли задать траекторию!!!

Ну, ничего. Это не страшно. Сейчас мы все исправим.

Траектория в Dreamweaver задается следующим образом. Каждая ключевая точка представляет собой как бы "снимок" того состояния, в котором будет пребывать анимированный элемент. Выбрав ключевую точку, вы задаете нужные параметры свободно позиционируемого элемента (координаты, размеры и видимость), а Dreamweaver их запоминает, создавая тем самым данный "снимок". Исходя из этих снимков, он сам вычисляет все промежуточные состояния анимированного элемента, в которых он будет находиться между ключевыми точками. Поэтому, чтобы создать анимацию любой сложности, вам достаточно будет задать необходимые параметры нужных элементов страницы только в ключевых точках анимации. Dreamweaver сам сделает все остальное.

Предположим, наш заголовок будет "выплывать" из нижнего правого угла страницы и перемещаться прямо на свое законное место. Поскольку траектория его движения очень проста (прямая линия), для ее задания мы нуждаемся всего в двух ключевых точках. В первой ключевой точке заголовок находится в нижнем правом углу страницы, а во второй — в ее верхней час-

ти, там, где он и должен быть. В этих точках мы и должны задать параметры нашего заголовка, точнее, всего два параметра — горизонтальную и вертикальную координаты. (Наш заголовок во время движения не меняет ни размеры, ни видимость.)

Выделим первую ключевую точку, находящуюся в начале дорожки, щелкнув по ней мышью. Далее "захватим" заголовок (свободный элемент `Header`) мышью и переместим его в правый нижний угол страницы, в начало его траектории. Теперь выделим вторую ключевую точку. Здесь нам менять ничего не нужно, т. к. заголовок уже стоит на своем месте. Вот, собственно, и все.

На рис. 12.6 показано то, что у нас получится после всех этих манипуляций. Хорошо видна тонкая серая *линия траектории*, отображаемая в окне документа, если в окне документа выделен анимированный элемент `Header`. Причем анимированный элемент будет находиться в том месте траектории, которое мы выделим на дорожке анимации. Так, если выделить вторую ключевую точку, он перескочит на свое законное место в верхней части страницы. А если выделить какую-либо промежуточную точку между первой и второй ключевыми точками, он займет соответствующее этой точке место.

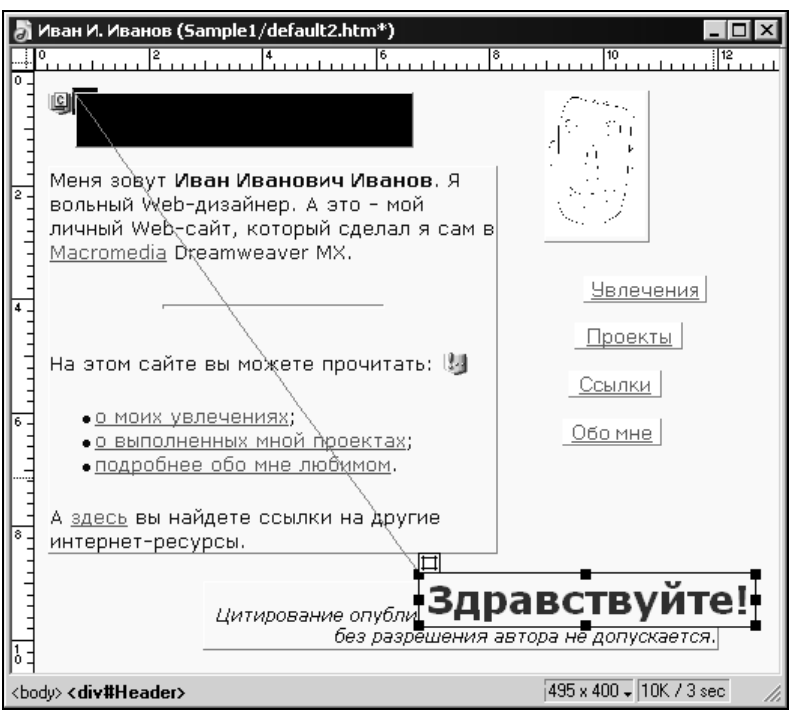


Рис. 12.6. Готовая траектория анимированного элемента (выделена первая ключевая точка)

Вот теперь можно и проверить наше творение. Загрузите получившуюся страницу в Web-обозревателе и посмотрите, что получится. А получится... все то же самое — опять-таки неподвижный заголовок. Что же мы на этот раз забыли?

Чтобы анимация запустилась сразу после загрузки страницы Web-обозревателем, Dreamweaver должен создать небольшой сценарий, который, собственно, ее и запускает. Но, по умолчанию, он этого не делает. Он предполагает, что вы хотите, чтобы анимация проигрывалась в ответ на действие пользователя, например щелчок по изображению. Но нам-то нужно, чтобы заголовок начинал свой путь сразу же, как только страница будет загружена. Для этого нам придется сделать соответствующие установки.

Сделать их очень просто. Вернемся в панель **Timelines** и включим флажок **Autoplay**, расположенный в верхней части этой панели. Dreamweaver, по своему обыкновению, выдаст очередное предупреждение, что сейчас в код страницы будет добавлен соответствующий сценарий. Избавьтесь от этого предупреждения, нажав кнопку **OK**; если не хотите больше его видеть, можете включить перед этим флажок **Don't show me this message again**. Вот теперь все на самом деле готово. Загрузите страницу в Web-обозреватель и убедитесь в этом.

Ура! Заработало!

Анимированный заголовок бодро пробежит заданную нами траекторию и замрет прямо над своей тенью. Конечно, плохо, что тень существует независимо от него, но мы это вскоре исправим. Не совсем хорошо также и то, что заголовок перемещается под остальными свободно позиционируемыми элементами, но вы можете исправить это прямо сейчас, просто изменив его z-индекс. Сохраним полученную страницу. И продолжим...

Более сложная анимация

Все-таки нехорошо, что заголовок и его тень (свободный элемент `Headershadow`) существуют отдельно. Давайте сделаем так, чтобы тень, пока заголовок движется к ней, была невидима на экране и появлялась лишь в самый последний момент. Для этого нам необходимо будет добавить свободный элемент, создающий эту тень, в анимацию, создав еще одну анимационную дорожку.

Выберите свободный элемент `Headershadow` и добавьте его в анимацию. Как это сделать, вы уже знаете. Получившийся результат можно увидеть на рис. 12.7.

Проследите, чтобы обе дорожки находились точно друг под другом. Если вторая дорожка получилась смещенной относительно первой, измените ее местоположение и (или) размер.

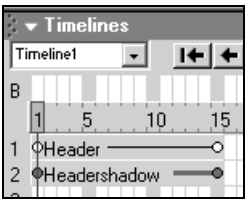


Рис. 12.7. Две дорожки анимации

Для тени мы не будем создавать траекторию движения, т. к. она не движется, а стоит на месте. Она должна просто появляться, когда заголовок "дойдет до точки". Поэтому никуда ее не перемещайте. А просто измените параметр видимости.

Итак, в начале траектории движения заголовка (первая ключевая точка) тень должна быть невидимой. Поэтому выделите первую ключевую точку второй дорожки и задайте в редакторе свойств значение **hidden** для параметра **Vis**. Теперь тень останется невидимой до тех пор, пока мы не изменим значение параметра видимости во второй ключевой точке (конец траектории движения заголовка). Так как по умолчанию параметр **Vis** имеет значение **inherit** ("наследование" видимости у родителя), а родитель (сама Web-страница) у нас видимый, специально задавать это значение во второй ключевой точке нам не нужно. На всякий случай проверьте, что на конце временной линии тень видима (параметр **Vis** должен иметь значение **inherit** или **visible**).

Сохраните результат вашей работы и откройте его в Web-обозревателе. Теперь тень будет скромно "прятаться", пока не "подъедет" ее "хозяин".

Все! Надоели нам прямолинейные, как трамвайные рельсы, траектории движения наших элементов. Хотим чего-нибудь извилистого и зигзагообразного, как пресловутый путь к истине. Пусть наш заголовок в своем пути "наверх" обходит всю Web-страницу. Это будет выглядеть эффектнее, да и мы заодно проверим, так ли хорош этот Dreamweaver, как его расхваливают.

Что ж, Dreamweaver достойно выдержит испытание извилистостью. И давайте в этом убедимся. Но сначала слегка удлиним траекторию, скажем, до 50 кадров. "Захватим" мышью вторую ключевую точку дорожки заголовка и переместим ее до соответствующего деления временной шкалы. И не забудем сделать то же самое с дорожкой элемента-тени!

Вот теперь можно и попетлять.

Как вы помните, ключевые точки, собственно, и задают траекторию движения анимированного элемента. Добавим также, что траектория его движения должна задаваться минимальным количеством ключевых точек. Для прямой, как мы уже убедились, вполне хватает двух точек, для изломанной линии — трех, а для окружности их понадобится, как минимум, десятка два, чтобы движение анимированного элемента выглядело более-менее

плавным. В общем, Dreamweaver, приняв от вас набор ключевых точек, должен построить такую траекторию движения анимированного элемента, которая вам необходима. И количество таких точек, как указывалось, должно быть минимальным.

Исходя из этого правила, построим новую траекторию движения нашего анимированного заголовка. Если вы хотите, чтобы он двигался по криволинейной траектории, проставьте на дорожке анимации достаточно большое количество ключевых точек. Проще всего сначала нарисовать нужную траекторию на бумаге, расставить ключевые точки, а уже потом реализовывать это в Dreamweaver.

Но, предположим, что все это мы уже сделали. Теперь нам необходимо реализовать траекторию, что называется, в коде, поместив несколько ключевых точек на дорожку анимации. Как это сделать?

Прежде всего установите маркер выделенного кадра (см. рис. 12.5) на то деление временной шкалы, где будет находиться первая из вновь создаваемых ключевых точек. Для этого щелкните мышью по дорожке напротив нужного деления. Далее щелкните на этом месте правой кнопкой мыши и выберите в появившемся контекстном меню пункт **Add Keyframe**. Вы также можете выбрать одноименный пункт в подменю **Timeline** меню **Modify** или просто нажать клавишу <F6>. На дорожке в этом месте появится новая ключевая точка (рис. 12.8).

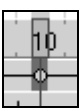


Рис. 12.8. Ключевая точка, помещенная на дорожке анимации (в данный момент выделена)

Теперь проверим, выделена ли наша новая ключевая точка, и, если нет, выделим ее. После этого активизируем окно документа и переместим заголовок в то место на траектории, где он должен находиться в настоящий момент. Тонкая серая линия, обозначающая траекторию движения заголовка, тотчас изогнется, следуя за заголовком (рис. 12.9).

Остальные ключевые точки траектории помещаются на дорожку анимации точно таким же образом. Поместите их и задайте для них новое положение анимированного заголовка. Изобретите траекторию посложнее, чтобы заставить Dreamweaver работать по-настоящему. Если вы поставили ключевую точку не в том месте, в котором хотели, выделите ее и выберите пункт **Remove Keyframe** контекстного меню. (Также вы можете выбрать одноименный пункт в подменю **Timeline** меню **Modify**.) После этого протестируйте страницу с новой анимацией в Web-обозревателе. Вы увидите, что заголовок следует точь-в-точь по траектории, которую вы для него задали.

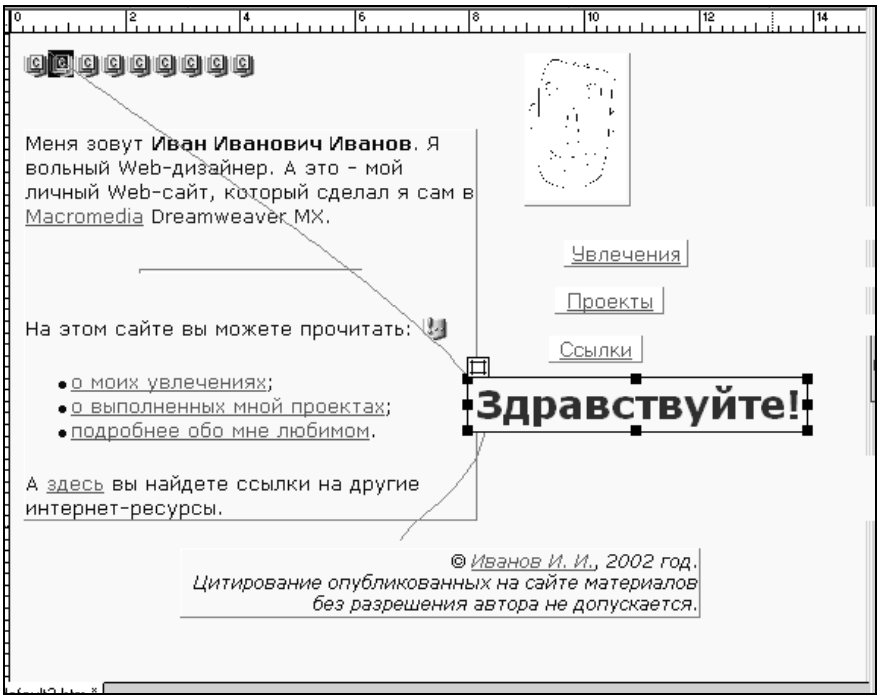


Рис. 12.9. Положение анимированного заголовка в новой ключевой точке

Здесь есть один интересный момент. Вы можете удивиться, что мы зачем-то создавали на дорожке анимации новые ключевые точки вместо того, чтобы поставить маркер на нужный кадр и просто переместить на необходимое место анимированный заголовок. Однако в этом случае мы переместили бы на другое место всю траекторию движения нашего заголовка, вместе с начальной, конечной и всеми промежуточными ключевыми точками. Попробуйте так сделать — и вы сами все поймете.

Создание траектории вручную, путем расстановки на дорожке анимации ключевых точек и задания новых параметров анимированного элемента в этих ключевых точках годится только для относительно простых случаев. В самом деле, создайте-ка таким образом что-либо более сложное, чем простая дуга! К тому же, даже опытный Web-дизайнер далеко не с первого раза сделает именно ту траекторию, какая ему нужна. Как раз для таких случаев Dreamweaver предоставляет замечательную возможность просто нарисовать нужную траекторию в окне документа. При этом он сам сформирует дорожку анимации и расставит на ней требуемое количество ключевых точек.

Давайте испытаем эту возможность. Только сначала выполним несколько подготовительных действий. Прежде всего, выделим последнюю ключевую точку (конец) траектории и запомним или запишем конечные координаты

заголовка. Это нужно для того, чтобы впоследствии, когда мы создадим новую анимацию, точно его позиционировать. После этого удалим дорожку анимации заголовка, но оставим дорожку анимации его тени. Для этого выделим необходимую дорожку и выберем пункт **Remove Object** в контекстном меню. Теперь осталось только поставить заголовок в начало его предполагаемой траектории — и можно приниматься за дело.

Выделим заголовок и щелкнем по нему правой кнопкой мыши. В появившемся контекстном меню выберем пункт **Record Path**. Вы также можете выбрать пункт **Record Path of Layer** контекстного меню панели **Timelines**; этот пункт становится доступным, если в окне документа выделен какой-нибудь свободный элемент. Кроме того, в подменю **Timeline** меню **Modify** также доступен пункт **Record Path of Layer**. Как видите, Dreamweaver предоставляет вам несколько возможностей сделать свой выбор.

Что произойдет после этого? Ничего. Кроме того, что при перемещении заголовка за ним будет тянуться тонкая серая линия — Dreamweaver начнет отслеживать и записывать его траекторию. А как только вы отпустите кнопку мыши, устанавливая заголовок в конец его траектории, он создаст новую дорожку анимации. После этого вам останется только выставить координаты заголовка в его конечной точке, воспользовавшись значениями, записанными (запомненными) ранее, и "растянуть" либо "сузить" дорожку

Рис. 12.10. Сложная траектория

анимации тени, чтобы она была равна дорожке своего "хозяина", иначе тень будет появляться раньше или позже, чем заголовок закончит свой извилистый путь.

Полученный результат вы можете увидеть на рис. 12.10. Сохраните страницу default2.htm, откройте ее в Web-обозревателе и посмотрите на результат ваших трудов.

Ну, вот и все. Мы познакомились с основными возможностями, предлагаемыми Dreamweaver для создания анимации. Более сложные анимации, включающие в себя множество элементов, согласованно движущихся по разным траекториям, создаются аналогичным образом. Только тогда вам предстоит несколько больше работы по заданию траекторий, выставлению параметров в ключевых точках анимации и согласованию движения анимированных элементов.

А сейчас мы познакомимся еще с некоторыми возможностями, предоставляемыми Dreamweaver Web-аниматорам. Они помогут вам сделать свою работу быстрее и проще.

Управление анимациями

Вы уже знаете, что Dreamweaver позволяет создавать несколько одновременно работающих и независимых друг от друга анимаций. Они могут работать как синхронно, так и асинхронно, создавая на Web-странице настоящие хороводы анимированных элементов. Конечно, если анимированных элементов на странице немного, вполне можно обойтись и одной анимацией. Но если их количество превышает десяток, значительно удобнее работать с несколькими независимыми анимациями, каждая из которых охватывает небольшую группу элементов, чем пытаться "запихать" их в одну анимацию. Впрочем, это дело вкуса...

Сейчас давайте рассмотрим средства управления этими независимыми анимациями.

Вы уже знаете, что в панели **Timelines** в данный момент отображается только одна анимация. Чтобы переключиться на другую, выберите соответствующий пункт уже знакомого вам комбинированного списка анимаций, показанного на рис. 12.2.

Чтобы создать новую анимацию, выберите пункт **Add Timeline** контекстного меню панели **Timelines**. Также вы можете выбрать одноименный пункт подменю **Timeline** меню **Modify**. Если вы теперь откроете комбинированный список анимаций, вы увидите, что в нем появился новый пункт.

По умолчанию Dreamweaver присваивает вновь создаваемым анимациям имена вида `Timeline<`*номер*`>`. Если же вы хотите дать какой-либо анимации более вразумительное имя, введите его прямо в комбинированный список анимаций. Другой способ — выберите пункт **Rename Timeline**, находящийся

в контекстном меню панели **Timelines** и в подменю **Timeline** меню **Modify**. После этого на экране появится диалоговое окно **Rename Timeline** (рис. 12.11). Введите новое имя анимации в единственное поле ввода этого диалогового окна и нажмите кнопку **OK**.

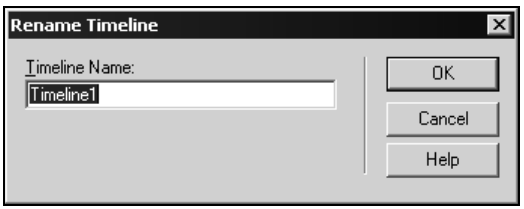


Рис. 12.11. Диалоговое окно **Rename Timeline**

Чтобы удалить ненужную анимацию, переключитесь на нее, использовав список анимаций, и выберите пункт **Remove Timeline**, находящийся в контекстном меню панели **Timelines** и в подменю **Timeline** меню **Modify**. Имейте, однако, в виду, что самую первую анимацию, создаваемую Dreamweaver при первом открытии панели **Timelines**, удалить нельзя — вместо удаления Dreamweaver ее просто очистит.

В верхней части панели **Timelines** находятся несколько еще не знакомых нам элементов управления. Рассмотрим некоторые из них.

Поле ввода **Fps** служит для задания *частоты кадров* анимации, определяющей скорость ее проигрывания. Эта величина измеряется в кадрах в секунду (по-английски — frames per second, или fps). Значение по умолчанию — 15.

Флажок **Autoplay** вам уже знаком. Он сигнализирует Dreamweaver, что в код страницы нужно добавить сценарий, запускающий анимацию сразу же после того, как страница будет загружена в окне Web-обозревателя. По умолчанию этот флажок отключен, и включить его начинающие пользователи Dreamweaver часто забывают. Поэтому, если ваша анимация почему-то не работает, прежде всего, проверьте, включили ли вы этот флажок.

Флажок **Loop** зацикливает анимацию. Если он включен, анимация будет проигрываться бесконечное число раз. Если его включить, Dreamweaver выведет еще одно из несметного множества своих сообщений; пока что просто закройте его. По умолчанию этот флажок выключен.

И, наконец, группа из трех кнопок и поля ввода, расположенная правее комбинированного списка анимаций, служит для перемещения между отдельными кадрами. Давайте рассмотрим его подробнее (рис. 12.12).

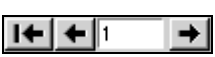


Рис. 12.12. Набор элементов управления, служащий для перемещения между кадрами анимации

Кнопки, на которых изображены направленные в разные стороны стрелки, позволят вам перемещаться по отдельным кадрам анимации. Предположим, вы выделили первую ключевую точку анимации (начало траектории), находящуюся на первом кадре. Если вы щелкнете кнопку со стрелкой вправо, то переместитесь на второй кадр. При этом на втором кадре анимации в панели **Timelines** будет установлен маркер выделенного кадра, а анимированный заголовок в окне документов переместится на один "скачок". Щелкая дальше на кнопке со стрелкой вправо, вы будете перемещаться все дальше к концу траектории. Если же вы щелкнете на кнопке со стрелкой влево, то переместитесь на предыдущий кадр.

Кнопку со стрелкой вправо вы можете использовать для предварительного просмотра созданной анимации прямо в окне документа, не открывая страницу в Web-обозревателе. Для этого поставьте на нее курсор мыши, нажмите левую кнопку и не отпускайте. При этом вы будете перемещаться с кадра на кадр вперед по траектории; в панели **Timelines** по дорожке будет перемещаться маркер, а в окне документа — анимированный элемент. Отпустите кнопку, когда "прокрутите" анимацию до конца. К сожалению, более удобного способа предварительного просмотра анимации Dreamweaver не предлагает.

Кнопка, на которой нарисована стрелка влево, упирающаяся в препятствие, служит для быстрого перемещения на самый первый кадр.

Если вам нужно переместиться прямо на какой-то кадр анимации, известный по номеру, вы можете просто ввести этот номер в поле ввода, расположенное между кнопками-стрелками, и нажать клавишу <Enter>. Нужный кадр будет тотчас найден и выделен.

С анимациями мы разобрались. Теперь давайте посмотрим, какие Dreamweaver предоставляет возможности для управления отдельными дорожками и кадрами.

Вы можете вырезать и копировать дорожки в буфер обмена Windows, а также вставлять их в другие каналы текущей анимации или вообще в другую анимацию на любой Web-странице. Чтобы вырезать выделенную дорожку, выберите пункт **Cut** контекстного меню или меню **Edit** либо нажмите комбинацию клавиш <Ctrl>+<X>. Чтобы скопировать выделенную дорожку, выберите пункт **Copy** в этих же меню либо нажмите комбинацию клавиш <Ctrl>+<C>. Ну, а чтобы вставить находящуюся в буфере обмена дорожку в один из уже занятых каналов, добавив к уже существующей дорожке, выберите пункт **Paste** или нажмите комбинацию клавиш <Ctrl>+<V>. К сожалению, вставить дорожку в свободный канал вы не сможете.

Внимание!

Dreamweaver помещает в буфер обмена не только дорожку анимации, но и свободно позиционируемый элемент, для которого она была создана, со всеми

его параметрами и содержимым. И если вы вставляете эту дорожку в другую страницу, которая уже содержит свободно позиционируемый элемент с таким же именем, Dreamweaver присваивает эту дорожку ему. Если же такого элемента на странице нет, он будет создан по образу и подобию скопированного.

После того как вы вставили новую дорожку из буфера обмена, вы можете захотеть присвоить ее другому свободному элементу. Dreamweaver предоставляет вам и такую возможность. Выберите пункт **Change Object** в контекстном меню или подменю **Timeline** меню **Modify**. После этого на экране появится диалоговое окно **Change Object**, показанное на рис. 12.13. Выберите нужный элемент в раскрывающемся списке **Object to Animate** и нажмите кнопку **OK**.

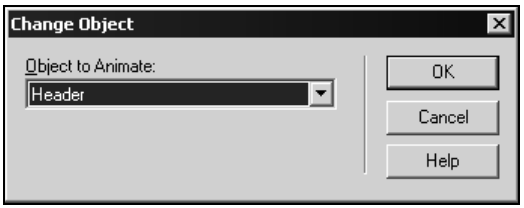


Рис. 12.13. Диалоговое окно **Change Object**

Иногда бывает необходимо растянуть какой-либо участок дорожки на несколько кадров или, наоборот, сузить. Для этого служат пункты **Add Frame** и **Remove Frame**, находящиеся в контекстном меню и подменю **Timeline** меню **Modify**. Первый пункт вставляет кадр в то место на дорожке, где находится маркер выделенного кадра, а второй — удаляет оттуда кадр.

Вот и все об анимации свободно позиционируемых элементов.

Анимация графических изображений

Выше мы с вами рассмотрели анимацию свободно позиционируемых элементов. По идее только их и можно анимировать, ведь только они могут позиционироваться где угодно и иметь какие угодно размеры. А значит, только их можно двигать по странице, изменять их размеры и делать невидимыми. С обычными, фиксированными элементами такого не сделать.

Правда, у этого правила есть одно исключение. Можно анимировать графические изображения, и Dreamweaver предоставляет такую возможность. Но изменяться в процессе анимации может только имя отображаемого файла (атрибут `SRC` тега `<IMG>`).

Однако даже этого часто бывает достаточно. В самом деле, изменяя имя графического файла, можно создавать анимации с быстро сменяющимися кадрами. (Фактически, таким образом можно проигрывать на Web-странице

настоящий фильм, не прибегая к созданию видеофайла, даже в формате "анимированный GIF".) Нужно будет только запастись необходимым количеством графических файлов, содержащих разные фазы анимации, но это уже проблема Web-художника.

Как же анимировать графическое изображение? Очень просто. Давайте создадим новую страничку и назовем ее 12.1.htm. Она не будет иметь отношения к сайту Sample site 1, но позаимствует из него графические файлы.

Поместите на эту страницу графическое изображение. В качестве отображаемого файла задайте Email.gif, находящийся в папке Pics, вложенной в папку Sample1 (в ней расположены файлы сайта Sample site 1). Это совсем просто, поэтому нет смысла пускаться в длительные объяснения.

Теперь создадим новую дорожку анимации для вновь созданного изображения. Выделим его и выберем пункт **Add Object** в контекстном меню панели **Timelines** (в контекстном меню графического изображения такого пункта нет). После этого Dreamweaver выведет очередное сообщение, предупреждающее, что он может управлять только параметром имени отображаемого файла (рис. 12.14). Закройте его, нажав кнопку **OK**.

Рис. 12.14. Предупреждение, выводимое Dreamweaver после добавления графического изображения в анимацию

Внешне созданная для графического изображения дорожка анимации не отличается от таковой для свободно позиционируемого элемента. Поэтому автор советует задавать для каждого анимируемого элемента страницы уникальное имя. Имя графического изображения можно задать в поле ввода **Image** редактора свойств.

Установим продолжительность анимации в пять секунд или 75 кадров (5 секунд умножить на 15 кадров в секунду). Для этого перетащим влево правую границу (вторую ключевую точку) дорожки до отметки "75" на шкале времени. И зададим новый графический файл, который отобразится во время достижения второй ключевой точки. Для этого проверим, выделена ли вторая точка, и изменим содержимое поля ввода **Src** редактора свойств так, чтобы оно указывало на файл Email2.gif, также находящийся в папке Pics, вложенной в папку Sample1. Данный файл будет отображен при достижении конца... нет, не траектории, а дорожки (ведь изображение не движется).

Осталось включить флажок **Autoplay** и загрузить страницу 12.1.htm в Web-обозревателе. Через пять секунд после окончания загрузки страницы изображение, отображаемое в ней, изменится, — наша анимация работает.

Если вам нужно последовательно сменить несколько изображений (фаз анимации), создайте соответствующее количество ключевых точек на дорожке. В каждой из этих ключевых точек установите нужное значение поля ввода **Src**.

Недостатки анимации, основанной на Web-сценариях, и их преодоление

Напоследок поговорим о недостатках опробованного нами способа создания анимаций на Web-страницах. И выясним, когда стоит его применять, когда можно ограничиться обычными видеофайлами, а когда можно обойтись вообще без всякой анимации.

Самый главный недостаток анимаций, основанных на Web-сценариях, — это... нет, отнюдь не несовместимость со старыми программами Web-обозревателей. (Хотя, несовместимость тоже следует принимать в расчет.) Это "тяжесть" необходимого для их обработки программного обеспечения, его ресурсоемкость и не очень высокое быстродействие. Это может быть критично, если среди посетителей вашего сайта будет много обладателей старых, маломощных компьютеров, на которых оно будет работать очень медленно, из-за чего анимация станет воспроизводиться рывками. Если вообще будет...

Как вы помните, Web-сценарии выполняются не самим Web-обозревателем, а специальной программой-интерпретатором, являющимся по отношению к Web-обозревателю внешней программой. Интерпретатор, расшифровывающий код сценария и исполняющий его, довольно требователен к ресурсам. Мало того, что нужно где-то хранить код сценария, исходный и расшифрованный, и свои данные, так еще нужно обрабатывать и сохранять данные самого сценария. И чем больше и сложнее сценарии, тем больше интерпретатор требует системных ресурсов.

В качестве решения этой проблемы можно только посоветовать не создавать без нужды сложных анимаций. Однако если уж без них не обойтись, стоит подумать о том, чтобы создать специальную версию сайта вообще без использования анимации и вообще сценариев. Тогда обладатели старых программ и (или) маломощных компьютеров смогут посещать ее, а счастливые владельцы современной техники получат удовольствие от "полновесной" версии сайта. Такой подход используется очень часто; почти все крупные и популярные интернет-проекты имеют "бессценарную" версию как раз на такой случай.

Есть и другой путь. Если ваша анимация представляет собой набор последовательно сменяющихся картинок или элемент, движущийся по небольшой площади страницы, попробуйте реализовать ее в виде видеофильма. Практически все программы Web-обозревателей, за исключением совсем уже старых, поддерживают формат "анимированный GIF-файл". Этот формат используется в Web-графике так долго, что даже подмял под себя некоторые стандарты, например стандарт на рекламные баннеры. Существует огромное количество программ для создания анимации в формате GIF. А для помещения такой анимации на Web-страницу вам нужен только хорошо знакомый тег `<IMG>` — и никаких сценариев!

Точно так же можно обойти проблему несовместимости со старыми программами.

Но есть еще одна проблема, решение которой весьма затруднительно. Это проблема авторских прав на программный код. Точнее, проблема охраны этих самых авторских прав.

Давайте рассмотрим обычные программы, которыми вы пользуетесь. Это могут быть популярнейший текстовый редактор Microsoft Word, проигрыватель мультимедийных файлов Nullsoft WinAmp, игра Quake 3 или сама операционная система Windows. Все эти программы были созданы с использованием компилируемых языков программирования, т. е. программа, написанная на каком-либо языке программирования (C++, Pascal или Assembler), была откомпилирована в набор инструкций центрального процессора и сохранена в файле с расширением exe. Впоследствии, если пользователь запустит этот файл, он будет исполняться непосредственно центральным процессором, без участия каких-либо программ-интерпретаторов.

Предположим, вы написали программу на компилируемом языке и хотите ее продавать. Причем продавать за большие деньги, т. к. в программе используется несколько принадлежащих вам ноу-хау. Если теперь хакер-злоумышленник захочет посмотреть, каков же принцип работы вашей программы, он потерпит неудачу. Дело в том, что читать машинный код процессора — занятие не для слабонервных. Далеко не всегда возможно просто понять, что он делает. А уж восстановить по нему исходный код программы абсолютно невозможно. (Хотя, если очень хочется, можете попытаться. Однако положительный результат этих попыток опять же не гарантирован.)

Программы Web-сценариев пишутся на интерпретируемых языках программирования. Это значит, что в Web-странице такие программы находятся прямо в виде исходного кода. Любой JavaScript-программист может без труда получить доступ к этому коду и выяснить, как работает программа. А это значит, что никаких секретов в JavaScript-коде не утаишь.

Выход из этой ситуации один: не реализовывать в качестве сценария сложные авторские алгоритмы. Интернет — среда, открытая по определению, и

распространять по ней закрытую, секретную информацию весьма проблематично. С другой стороны, на то они и секреты, чтобы хранить их от любопытных глаз, которых во Всемирной паутине более чем достаточно.

Есть, вообще-то, еще один недостаток, но он свойствен не самой анимации, основанной на Web-сценариях, а подходу, предлагаемому Dreamweaver. Дело в том, что необходимый для создания анимации набор Web-сценариев помещается в секцию HTML-заголовка страницы (тег <HEAD>). Из этого следует, что если вы используете для создания страниц шаблоны, вы можете создать анимированные (и вообще любые свободно позиционируемые) элементы только в самом шаблоне, но никак не в созданных на его основе страницах. Dreamweaver просто не даст вам этого сделать. В самом деле, секция заголовка — это неизменяемая область, а поместить какой-либо код в неизменяемую область Dreamweaver вам не позволит. Имейте это в виду. Хотя, конечно, вы можете отказаться от анимации или "открепить" страницу от шаблона, а потом уже делать с ней все, что вам заблагорассудится.

Что дальше?

Кино закончилось. Да здравствует кино!

Как видите, создание анимации на Web-странице — задача довольно простая, если, конечно, вы работаете в среде Dreamweaver. Вы только задаете траекторию движения нужного элемента, а Dreamweaver делает всю остальную работу по "вдыханию" в него "жизни". Если хотите посмотреть, что же он делает с вашей страницей в этом случае, посмотрите исходный HTML-код в режиме отображения кода. Вы будете удивлены, сколько всего сделал Dreamweaver только для того, чтобы заголовок вашей страницы сдвинулся с места, какие огромные Web-сценарии он для этого создал.

Web-сценарии... Сколько мы уже говорили о них! А могут ли они что-нибудь еще, кроме того, что двигать элементы страниц взад-вперед? Конечно! Как раз следующая глава будет посвящена только Web-сценариям...

ГЛАВА 13

Использование сценариев

В *главе 12*, когда говорилось об анимации на Web-страницах и способах ее создания, очень часто упоминались Web-сценарии. В этой главе мы их рассмотрим подробнее.

Как вы уже знаете, Web-сценарии позволяют преодолеть ограничение HTML, обусловленное его природой языка разметки страницы. Это ограничение заключается в том, что с помощью HTML можно описать только внешний вид данных, отображаемых на Web-странице, но никак не их поведение при наступлении того или иного события (о событиях и их обработке *см. в главе 12*). Говорят, что HTML — язык представления данных, но не описания способов их обработки.

А если вы хотите анимировать какой-либо элемент страницы, вам придется описать поведение этого элемента в виде *алгоритма* — последовательности операций, выполняемых компьютером. Этот алгоритм будет выполняться при наступлении особого внутреннего события — "тика" системного таймера. При каждом таком "тике" элемент будет сдвигаться на небольшое расстояние, а поскольку такие "тики" повторяются очень часто, человеческий глаз воспринимает скачкообразное движение как непрерывное. Вот вкратце основной принцип создания анимированных элементов.

Для описания алгоритмов, по которым будет двигаться анимированный элемент, используются Web-сценарии. Именно они реализуют вычисление координат анимированного элемента и его перемещение.

Конечно, в *главе 12* для создания анимаций мы использовали Dreamweaver, который позволяет делать это, просто рисуя в окне документа траекторию движения нужного элемента. Однако основные принципы все те же. Web-сценарии, написанные особым образом, вызываются при каждом "тике" таймера и заставляют анимированный элемент "скакнуть" на очередную позицию.

Web-сценарии практически всегда пишутся на языке программирования JavaScript. Конечно, существуют и другие языки, но именно JavaScript завоевал широкую популярность. Этот язык начал полноценно поддерживаться Internet Explorer 4.0 и Navigator 3.0. Также ограниченная поддержка JavaScript была реализована в Internet Explorer 3.0. Ну, а Opera и Mozilla поддерживают его с самого "рождения".

Для чего еще могут быть использованы Web-сценарии? О-о-о, много для чего. Давайте поговорим об этом, прежде чем приступим к изучению языка JavaScript.

Подробнее о Web-сценариях

Здесь будет рассказано о задачах, успешно решаемых с использованием Web-сценариев. Далее будет дан краткий урок основ JavaScript, после чего мы с вами перейдем к рассмотрению стандартных сценариев, поддерживаемых Dreamweaver. Автор не ставит цели сделать из вас знатоков JavaScript — в конце концов, эта книга совсем о другом. Если же вы хотите побольше узнать об этом языке, найдите другие книги, благо сейчас их издано очень много.

Итак, пора дать ответ на вопрос...

Зачем нужны Web-сценарии

Web-сценарии могут быть применены везде, где не может быть использован "чистый" HTML, т. е. там, где элементы страницы должны вести себя так, как стандартом HTML не предусмотрено.

Возьмем ту же самую анимацию. За очень небольшим исключением (например, нестандартный тег `<MARQUEE>`, поддерживаемый Internet Explorer), элементы страницы не могут по ней двигаться. Если вам все же нужно, чтобы они двигались, не обойтись без применения Web-сценариев. Зная язык JavaScript, вы с легкостью реализуете нужное поведение элементов, заставите их летать с требуемой скоростью и по желаемой траектории. HTML такое не может.

Очень часто Web-сценарии используются для изменения содержимого Web-страницы или даже для создания нового содержимого в ответ на действия пользователя. В частности, при наведении курсора мыши на гиперссылку рядом с ней может появляться краткое описание, либо изображение-гиперссылка может менять свой вид. (Кстати, такие "горячие" изображения, меняющиеся при наведении на них курсора мыши, мы уже создавали в *главе 3* при помощи Dreamweaver. Такие "фокусы" реализуются также с помощью Web-сценариев.) Иногда содержимое Web-страницы корректируется сразу же в процессе ее загрузки Web-обозревателем, что достигается помещением в HTML-код сценариев, исполняющихся при загрузке.

Вообще, фантазия Web-дизайнера, решившего создать интерактивную Web-страницу с использованием Web-сценариев, почти ничем не ограничена. (Ограничения, конечно, существуют, но их очень мало, и касаются они, в основном, доступу к содержимому дисков клиентского компьютера.) В настоящее время существуют даже Web-страницы, содержимое которых может изменяться самим пользователем прямо в окне Web-обозревателя. (Другой вопрос: зачем это нужно?..) На страницах может отображаться, например, текущее время, причем, "часики" действительно будут "тикать". Также Web-сценарии часто используются для правильного размещения на странице свободно позиционируемых элементов. В частности, можно создать свободный элемент, который при любых изменениях размеров окна Web-обозревателя всегда будет находиться в его центре.

С помощью Web-сценариев можно создать принципиально новый интерфейс пользователя для своей страницы. На многих страницах применяется иерархический список, в котором перечислены все разделы и подразделы сайта. Такие страницы напоминают окно Проводника Windows. Часто с помощью свободно позиционируемых элементов и сложных Web-сценариев создают принципиально новые элементы управления для ввода данных, не снившихся даже самой великой и ужасной Windows. Но это уже — удел достаточно опытных программистов.

Ну и, конечно, Web-сценарии применяются для написания настоящих программ, использующих в качестве интерфейса Web-страницы. Известны, например, игры "15", "Лото", различные головоломки и викторины, созданные на основе Web-сценариев и размещенные в Интернете. Часто таким же образом создаются и более серьезные программы, например, утилиты подбора цветов для Web-страниц или даже целые Web-редакторы, конечно, не очень сложные.

Вообще, создание таких вот Web-программ — достаточно новая область деятельности, еще даже толком не развившаяся, но довольно перспективная. В самом деле, такие программы очень просто распространять и сопровождать. Они не требуют создания разработчиком дистрибутивного пакета, иной раз весьма "увесистого", и распространения его каким-либо образом, а потенциальным пользователям не придется его подолгу загружать. Они не требуют установки на компьютере пользователя, а становятся доступны сразу после набора в окне Web-обозревателя нужного адреса. А при создании новой версии такой программы разработчику достаточно будет только обновить на своем сайте соответствующую Web-страничку.

Web-программы имеют только два недостатка. Во-первых, они доступны только после подключения к Интернету. (Хотя, конечно, их можно распространять на дискетах, CD, по электронной почте и другими путями.) Во-вторых, из-за ограничений интерпретатора JavaScript с помощью таких программ невозможно сохранение документов на дисках клиентского компьютера.

Язык JavaScript

Выше автор предупреждал, что не будет давать вам полный курс JavaScript. Эта книга не о JavaScript, а о Macromedia Dreamweaver MX, а все остальное вторично. Если вы хотите побольше узнать о JavaScript, принципах написания программ на этом языке и, вообще, о "классическом" программировании, найдите хорошие книги по этой теме. А мы вернемся к Dreamweaver.

Но, для того чтобы вы поняли, что есть на самом деле Web-сценарии, необходимо провести хотя бы краткий ликбез. И сейчас он будет проведен.

Давайте вернемся к уже знакомому нам HTML. Что он собой представляет? Набор тегов, описывающих внешний вид и (отчасти) структуру представления данных, которые будут отображаться в Web-обозревателе. HTML, как вы помните, язык описания данных.

JavaScript же — язык описания алгоритмов обработки этих самых данных. Поэтому он содержит набор инструкций, предписывающих выполнение тех или иных действий над данными. Чтобы облегчить работу программиста, способ написания этих инструкций максимально приближен к обычной математической нотации, слегка "разбавленной" словами обычного английского языка. (Вот еще один повод взяться за английский.)

Такие инструкции, описывающие действия над данными, называются *выражениями*. Выражения состоят из *операторов* и *операндов*; операторы описывают сами действия, а операнды — данные, над которыми будут производиться эти действия. Причем, операндами могут быть как *константы* (числа, строки, даты, одним словом, неизменяемые значения), так и *переменные* (специально отведенные ячейки для хранения промежуточных результатов вычислений).

Давайте рассмотрим одно из таких выражений:

```
b = z / t;
```

Это выражение делит значение переменной `z` на значение переменной `t` и результат помещает в переменную `b`. Как вы уже поняли, знак "`/`" обозначает оператор деления, а знак "`=`" — оператор *присваивания* значения какой-либо переменной. Знак "`;`", помещенный в конце выражения, обозначает, собственно, его конец; встретив этот знак, интерпретатор JavaScript считает выражение законченным.

Вот еще два выражения:

```
sum = a1 + a2 + a3 – 20;
square = x * y;
```

Первое выражение складывает друг с другом значения переменных `a1`, `a2` и `a3`, после чего вычитает из результата `20` и помещает его в переменную `sum`.

Второе выражение перемножает значения переменных x и y и помещает результат в переменную square.

Чтобы управлять порядком вычисления выражения, вы можете использовать *скобки*:

```
s = (a + b) / (c – d);
```

В этом выражении сначала будет вычислена сумма a и b, потом — разность c и d, после чего сумма будет поделена на разность. Если бы мы не использовали скобки, то выражение имело бы вид:

```
s = a + b / c – d;
```

При этом сначала было бы вычислено частное от деления b на c, после чего к нему была бы прибавлена a, а затем из этой суммы вычлось бы d. Попробуйте подставить вместо a, b, c и d реальные числовые значения и вычислить результаты этих выражений. Вы увидите, что они получатся разными.

В JavaScript можно написать и так:

```
a = a + 1;
```

и это выражение будет правильным. Оно предписывает интерпретатору извлечь значение из переменной a, увеличить его на единицу и поместить в ту же переменную. Это выражение можно записать и короче:

```
a += 1;
```

или совсем сжато:

```
a++;
```

Кстати, любая переменная перед тем, как будет использована, должна быть объявлена. *Объявление* переменных выполняется с помощью специального оператора объявления var:

```
var a1, a2, a3, x, y
```

Объявляемые переменные просто перечисляются после этого оператора, после чего их можно использовать в сценарии. Такие переменные называются *глобальными*, т. к. могут быть использованы где угодно в программе.

Говорилось, что переменные могут содержать какие угодно данные. В самом деле, посмотрите сюда:

```
a = 10;
s = "Строка символов";
```

Здесь первой переменной присваивается числовое значение, а второй — строка (строковое значение). (Вообще, Web-сценарии, как правило, чаще обрабатывают текст, а не числа, так что в этом нет ничего необычного.)

В общем, одна переменная всегда содержит одно значение. Если переменной присвоить новое значение, то старое значение пропадет. Но существует другой вид переменных, которые могут содержать несколько значений. Такие переменные называются *массивами*. Отдельные значения, содержащиеся в массиве, называются его *элементами*; доступ к элементам массива осуществляется с помощью так называемого *индекса*.

```
var arr;
arr[0] = 1;
arr[1] = 2;
arr[5] = arr[0] + arr[1];
```

Как видите, индекс элемента массива указывается после его имени в квадратных скобках; этим массив и отличается от обычной переменной *(скаляра)*. Сам массив объявляется точно так же, как обычная переменная, и со значениями его элементов можно делать то же самое, что и со значениями обычных переменных. Заметьте, что специально объявлять его как массив не нужно — как только вы при обращении к переменной укажете индекс элемента, интерпретатор JavaScript тут же преобразует переменную в массив.

Для примера был создан *одномерный* массив. Но бывают и массивы *двумерные*:

```
var arr2;
arr2[0][0] = 0;
arr2[0][1] = 1;
arr2[1][0] = 2;
arr2[1][1] = 3;
```

Ключевыми словами называются специально зарезервированные слова, обозначающие некоторые операторы языка программирования. Так, *условный* оператор `if-else` состоит из двух ключевых слов. Он позволяет выполнить какое-либо выражение только в том случае, если становится верным какое-либо *условие*. Если условие верно, выполняется выражение, стоящее после ключевого слова `if`, а если неверно, — стоящее после ключевого слова `else`. Говорят, что оператор `if-else` организует *ветвление*.

Ниже приведен пример использования этого оператора.

```
if (x == 1)
  f = 2
else
  f = 12;
```

Здесь, если значение переменной `x` равно единице, переменной `f` присваивается `2`, в противном случае — `12`.

Если в какой-либо из частей оператора `if-else` нужно выполнить несколько выражений, используется так называемое *составное выражение*:

```
if (x == 1) {
  f = 2;
  h = 3; }
else {
  f = 12;
  h = 14; }
```

Как видите, составное выражение состоит из нескольких простых выражений, помещенных внутрь фигурных скобок. Интерпретатор JavaScript считает их одним выражением.

Существует также "урезанная" форма оператора `if-else`, без части `else`. В этом случае, если условие неверно, никакого кода не выполняется.

```
if (x == 1) {
  f = 2;
  h = 3; }
```

Существует также *оператор-переключатель* `switch-case`. Этот оператор заменяет множество операторов `if-else`.

```
switch (a) {
  case 1 :
    out = "Единица";
    break;
  case 2 :
    out = "Двойка";
    break;
  case 3 :
    out = "Тройка";
    break;
  default :
    out = "Другое число";
}
```

Если значение переменной `a` равно 1, переменной `out` будет присвоено значение `Единица` (часть `case 1`). Если значение переменной `a` равно 2, то переменная `out` получит значение `Двойка` (часть `case 2`), и т. д. Если же переменная `a` содержит значение, не перечисленное в списке `case`, выполняется часть `default`, и `out` принимает значение `Другое число`.

Как видите, оператор-переключатель просто выполняет разные фрагменты кода в зависимости от значения переменной, находящейся в скобках после ключевого слова `switch`. Поэтому он и называется переключателем.

Оператор *цикла* `for` позволит вам выполнить какое-либо выражение (простое или составное) нужное количество раз, т. е. организовать *цикл со счетчиком*. При этом на каждом проходе цикла содержимое переменной-*счетчика* будет сравниваться с *конечным значением*, и если оно его превышает, цикл немедленно завершается. В противном случае содержимое счетчика увеличивается или уменьшается на единицу, и выполняется новый *проход* цикла.

Рассмотрим применение оператора `for` на примере.

```
for (i = 1; i < 101; i++) {
  a = a * i;
  b = b + i;
}
```

Первая строка сообщает интерпретатору JavaScript следующее:

- переменная `i` будет использоваться в качестве счетчика цикла, т. е. в ней будет находиться значение количества уже выполненных проходов цикла;
- *начальное значение* счетчика — 1 (`i = 1`);
- конечное значение счетчика — 100, т. е. повторять цикл нужно, пока содержимое счетчика `i` остается меньше 101 (`i < 101`);
- при каждом проходе цикла нужно будет увеличивать значение счетчика на единицу (`i++`; оператор `++` указывает увеличить значение какой-либо переменной на единицу, другими словами, *инкрементировать*, и поместить его в ту же переменную). (Оператор `++` называется оператором *инкремента*.)

В результате выполнения приведенного выше выражения в переменной `a` будет находиться факториал от 100, а в переменной `b` — сумма от 1 до 100.

Кроме цикла со счетчиком, в JavaScript можно организовать и циклы без счетчика, так называемые *циклы с условием*. Они выполняются до тех пор, пока остается истинным какое-либо условие. Это циклы `do-while` и `while`.

Сначала рассмотрим цикл `do-while`.

```
do {
  a = a * i + 2;
  i = ++i;
} while (a < 100);
```

Этот цикл будет выполняться, пока значение переменной `a` остается меньше 100. Заметьте, что условие проверяется после прохода цикла, поэтому этот цикл выполнится хотя бы один раз, даже если условие будет изначально ложно.

В цикле `while` условие проверяется перед проходом цикла, поэтому, если условие изначально ложно, цикл не выполнится ни разу.

```
while (a < 100) {
  a = a * i + 2;
  i = ++i;
  }
```

Оператор *завершения* `break` немедленно *завершает* цикл, а оператор *продолжения* `continue` — *продолжает* цикл, т. е. прекращает выполнение текущего прохода и начинает выполнение следующего (конечно, если условие завершения цикла это допускает).

```
while (a < 100) {
  i = ++i;
  if (i > 50) break;
  if (i < 5) continue;
  a = a * i + 2;
  }
```

Попробуйте сами разобраться, как работает вышеприведенный пример.

Если вам нужно выполнять в нескольких местах программы один и тот же фрагмент кода с разными числовыми значениями, вы можете создать на его основе функцию. *Функция* — это фрагмент кода, оформленный особым образом, который может быть вызван из различных мест программы и из других функций. Функция может принимать любое количество *параметров* и возвращать один *результат*, который может быть использован в дальнейших вычислениях.

Давайте рассмотрим небольшой пример функции, увеличивающей переданное ей значение на два и возвращающей результат.

```
function valuePlus2(v) {
  var c;
  c = v + 2;
  return c;
}
```

Автор назвал нашу функцию `valuePlus2` и передал ей единственный параметр `v` (см. в скобках после имени функции). Код, находящийся внутри функции, сначала объявляет *локальную* переменную `c`, "видимую" только внутри этой функции (вызвавшая функцию программа не сможет к ней обратиться). Далее этой переменной присваивается сумма значения переданного параметра `v` и 2. Последний оператор возвращает результат вызвавшей функцию программе (`return` — оператор *возврата значения*).

Как можно использовать полученную функцию? Например, так:

```
h = d + valuePlus2(r);
```

Здесь мы передаем функции `valuePlus2` значение `r` и используем возвращенный результат в выражении. В частности, мы складываем его с `d` и присваиваем результат `h`.

Кстати, нашу функцию можно было написать значительно короче и оптимальнее. (Да, и здесь оптимизация!)

```
function valuePlus2(v) { return v + 2; }
```

Опытные программисты так и пишут. Во-первых, код становится проще и "прозрачнее". Во-вторых, что еще важнее, код становится меньше и быстрее.

Функция может и не принимать параметров:

```
function someFunc1() { return 2 + 2; }
```

Функции могут вызывать друг друга:

```
function valuePlus3(v) { return valuePlus2(v) + 1; }
```

На этом краткий курс основ языка JavaScript подошел к концу. Нам осталось рассмотреть только объекты.

Объекты

Выше были рассмотрены два вида переменных: обычные переменные, иначе говоря, скаляры и массивы. Скаляры могут содержать только одно значение, массивы же — множество пронумерованных значений, доступ к которым можно получить по их номеру — индексу. И скаляры, и массивы находят свое применение в программах на JavaScript; нет смысла рассказывать, в каких случаях что применять, — это и так очевидно. Давайте лучше поговорим еще об одном виде переменных, которого мы до сих пор не касались.

Это объекты.

Объектом называется сложный тип данных, содержащий не какое-то одно значение, а целую сущность. Эта сущность может иметь набор свойств и методов, с помощью которых программа может ей управлять; *свойство* — это своего рода переменная, принадлежащая объекту, а *метод* — функция, также принадлежащая объекту и выполняющая над ним какие-либо действия. Объект (и сущность, содержащаяся в нем) представляет собой "вещь в себе"; ее внутренняя структура и принцип действия неизвестны использующему объект программисту.

В качестве примера объекта можно рассмотреть прекрасно знакомый вам Web-обозреватель. У него есть свойство "адрес" и метод "открыть Web-страницу, которой принадлежит этот адрес". Вы можете присвоить этому свойству нужный адрес, а также можете его оттуда считать и присвоить какой-либо переменной или использовать иным способом:

```
currentWebBrowser.address = "http://www.w3c.org";
currAddr = currentWebBrowser.address + "/css/";
```

Именно такой синтаксис используется для доступа к свойствам: <имя объекта>.<имя свойства> (не забудьте поставить знак точки). В первом случае мы поместили в свойство address объекта currentWebBrowser строку с интернет-адресом комитета WWWC. Во втором случае мы извлекли из этого свойства находящийся в нем адрес, прибавили к нему справа текст "/css/" и присвоили переменной currAddr. (Как видите, для *слияния* (или *конкатенации*) строк использовался оператор +.)

Для вызова метода используется аналогичный синтаксис: <имя объекта>.<имя метода>(). Сейчас мы вызовем метод go вышеупомянутого объекта:

```
currentWebBrowser.go();
```

Метод go — не что иное, как обычная функция. В данный момент возвращенное ей значение игнорируется, но в других случаях оно может использоваться. Также метод, как и любая функция, может принимать параметры:

```
currentWebBrowser.goTo("http://www.w3c.org");
```

Таким образом мы можем управлять Web-обозревателем. И — заметьте — ничего не зная о его внутреннем устройстве. Более того, мы не обязаны о нем знать. Использование объектов как раз и направлено на то, чтобы позволить программистам пользоваться различными инструментами и дополнительными компонентами, не зная их внутреннего устройства, а также создавать такие инструменты и компоненты для своих коллег.

Говорят, что JavaScript — *объектно-ориентированный* язык программирования, т. к. использует объекты. Этим он отличается от обычных, *процедурных* языков, которые используют только функции.

Каждый объект перед тем, как с ним можно будет работать, должен быть создан. Выполняется это с помощью оператора *создания объекта* new.

```
var currentWebBrowser;
currentWebBrowser = new WebBrowser();
```

Здесь объявляется переменная currentWebBrowser и ей присваивается вновь созданный с помощью оператора new объект. Этот объект создан на основе *класса* WebBrowser, который является как бы шаблоном для создаваемых объектов. А currentWebBrowser также иногда называется *экземпляром* класса WebBrowser.

Удалить ненужный объект можно с помощью метода delete.

```
currentWebBrowser.delete();
```

Имейте в виду, что ненужные объекты, созданные вами, всегда надо удалять, чтобы освободить ресурсы системы. Исключение составляют только

системные объекты, создаваемые самим интерпретатором или предоставляемые интерпретатору Web-обозревателем. Если вы попытаетесь удалить системный объект, интерпретатор выдаст сообщение об ошибке. О системных объектах мы поговорим ниже.

Но каким же образом создаются классы объектов? Исключительно просто. Более того, если бы остальные языки объектно-ориентированного программирования увидели, как легко создаются классы в JavaScript, они бы умерли от зависти.

Взглянем еще раз на выражение, с помощью которого создаются объекты:

```
currentWebBrowser = new WebBrowser();
```

Согласитесь, это сильно напоминает вызов функции. Но дело в том, что `WebBrowser` — и есть функция, только специальным образом написанная. Она называется *конструктором* класса.

Давайте напишем конструктор для класса `WebBrowser`.

```
function WebBrowser() {
  this.address = "http://www.server.ru/my_home_page.htm";
}
```

Этот код создает в классе `WebBrowser` одно-единственное свойство `address`. Обратите внимание на синтаксис, с помощью которого это делается. В качестве имени класса используется ключевое слово `this`, обозначающее текущий класс. Интерпретатор JavaScript создает свойства класса сразу же при первом обращении к ним; вот и в нашем случае он создал свойство `address` и дал ему значение, присвоенное нами.

Конструктор может принимать параметры:

```
function WebBrowser(homePage) {
  this.address = homePage;
}
```

Таким образом мы можем передать конструктору класса адрес домашней страницы, с которой начнется путешествие по Интернету.

```
var currentWebBrowser;
currentWebBrowser = new WebBrowser("http://www.server.ru/home_page.htm");
```

"Но, позвольте! — скажете вы. — А как же создаются методы класса?" Тоже очень просто.

Прежде всего, нам нужно написать функцию, реализующую этот метод.

```
function fGoTo(pageAddress) {
  this.address = pageAddress;
}
```

Мы назвали эту функцию fGoTo. Это имя состоит из буквы f (от англ. *function* — функция) и имени метода goTo, который она реализует.

Теперь перепишем функцию-конструктор таким образом, чтобы создать новый метод goTo.

```
function WebBrowser(homePage) {
  this.address = homePage;
  this.goTo = fGoTo;
}
```

Как видите, мы "присваиваем" написанную нами функцию fGoTo свойству goTo класса WebBrowser. После этого можно вызывать метод goTo этого класса.

```
var currentWebBrowser;
currentWebBrowser = new WebBrowser("http://www.server.ru/home_page.htm");
currentWebBrowser.goTo("http://www.w3c.org");
```

Только что вы научились создавать простейшие классы. Но на самом деле вам в очень редких случаях придется делать это. Куда чаще вы будете пользоваться уже готовыми *системными классами*, предоставляемыми самим интерпретатором JavaScript и другими программами, например Web-обозревателем. Поэтому давайте поговорим о системных классах.

Системных классов, предоставляемых программисту интерпретатором, довольно много. Среди них есть, например, класс Date, предназначенный для работы со значениями даты и времени.

```
var d;
d = new Date();
```

Только что мы создали объект класса Date. Так как мы не передали параметр в функцию конструктора этого класса, интерпретатор поместил в этот объект значение текущей даты. После этого можно, например, узнать номер дня недели:

```
var dayNumber;
dayNumber = d.getDay();
```

Или выяснить год:

```
var year;
year = d.getFullYear();
```

Некоторые методы класса Date перечислены в табл. 13.1.

Как видите, класс Date предусматривает полный набор методов для работы со значениями даты и времени. Другие системные классы также имеют достаточно большие наборы свойств и методов, которые мы не будем здесь рассматривать.

Таблица 13.1. *Некоторые методы класса* `Date`

Метод	Описание
`getDate()`	Возвращает число
`getDay()`	Возвращает цифру, обозначающую день недели (0 — воскресенье, 1 — понедельник, 2 — вторник и т. д.)
`getFullYear()`	Возвращает год
`getHours()`	Возвращает час
`getMilliseconds()`	Возвращает миллисекунды
`getMinutes()`	Возвращает минуты
`getMonth()`	Возвращает цифру, обозначающую месяц (от 0 до 11)
`getSeconds()`	Возвращает секунды
`getTime()`	Возвращает время в виде количества миллисекунд, прошедших с полуночи 1 января 1970 года

Системный класс `Math` — другого рода. Он включает набор свойств и методов, реализующих различные математические и тригонометрические функции. Единственный его объект создается самим интерпретатором JavaScript при запуске и уничтожается при завершении работы. Так что вам не нужно самим создавать объекты этого класса. Это как раз типичный случай системного объекта.

```
var f;
f = Math.sin(Math.PI);
```

В результате вычисления вышеприведенного выражения в переменной `f` окажется значение $\sin(\pi)$.

Существуют также системные классы `Number` и `String`, служащие для хранения данных соответственно числового и строкового форматов, и несколько других системных классов, используемых значительно реже.

Язык JavaScript имеет одну любопытную особенность. Дело в том, что любая переменная обычного типа данных может быть представлена как объект какого-либо класса. Давайте, например, рассмотрим следующий код:

```
var s, l;
s = "JavaScript";
l = s.length;
```

Здесь мы сначала помещаем в переменную `s` строку "JavaScript", а потом вызываем свойство `length` этой переменной. В таком случае интерпретатор

считает переменную `s` объектом класса `String` и беспрепятственно "пускает" нас к свойству `length` этого класса, возвращающему длину строки текста, которую мы и помещаем в переменную `l`.

Аналогичный "финт" можно проделать и с числовой величиной:

```
var a, s;
a = 16765247;
s = a.toString();
```

Здесь вызываем метод `toString` класса `Number`, возвращающий строковое представление числа.

Раньше говорилось, что, помимо интерпретатора JavaScript, системные классы могут представляться также и другими программами. К числу этих самых "других" программ относится Web-обозреватель. Его классы и системные объекты — это что-то особенное...

Объектная модель документа (DOM)

А теперь настало время поговорить о классах и объектах Web-обозревателя подробнее.

Давайте посмотрим на какую-нибудь Web-страницу. Что она собой представляет? Фактически, иерархию элементов. Мельчайшие элементы страницы, например текстовые абзацы, являются потомками более крупных и сложных элементов, например, свободно позиционируемых элементов. Свободные элементы, в свою очередь, могут находиться в других свободных элементах или непосредственно в самой странице. Получается весьма сложная многоуровневая структура, в которой одни элементы зависят от других.

Описать такую структуру можно с помощью объектов. Точнее, сложной иерархии объектов, вложенных друг в друга и зависящих друг от друга. Такая структура называется *объектной моделью документа* (Document Object Model, *DOM*). Все современные программы Web-обозревателей представляют Web-страницу как иерархический набор объектов.

Какое преимущество дает пользователям такой подход? Никакого. Все это рассчитано только на программистов, разрабатывающих Web-сценарии.

Каждый из объектов, из которых состоит Web-страница, имеет набор свойств, предоставляющих доступ к значениям различных атрибутов соответствующего тега, методов, с помощью которых этим объектом можно манипулировать, и событий, которые могут в этом объекте происходить и которые можно обрабатывать. Это позволяет управлять практически любым элементом страницы, самой страницей и даже самим Web-обозревателем, используя специально написанные Web-сценарии.

Например, именно таким образом на Web-странице создаются анимированные элементы *(см. главу 12)*. Также с помощью сценариев можно изменять

цвета, параметры шрифта и даже само содержимое элементов страницы. Более того, так можно управлять и самим Web-обозревателем: открывать и закрывать вспомогательные окна, перемещаться взад-вперед по списку "истории" и даже принудительно загружать нужные Web-страницы без участия пользователя.

Неужели вы думали, что Web-программисты пройдут мимо такой возможности! Если вы думаете, что пройдут, — вы плохо их знаете. Вспомните, например, что говорилось в предыдущей главе об анимации. Едва появилась возможность немного подвигать по странице тем или иным рисунком, как Web-программисты эту возможность реализовали. А сейчас анимацию поддерживают даже Web-редакторы, и Dreamweaver тому пример.

Но хватит говорить на отвлеченные темы! Давайте все-таки займемся объектной моделью документа.

Сначала выясним, каким образом можно получить доступ к нужному элементу. Для этого ему необходимо дать уникальное *имя*. Делается это с помощью атрибута `ID` или `NAME`. Атрибут `ID` поддерживается практически всеми тегами HTML, атрибут `NAME` — только некоторыми. К тегам, поддерживающим этот атрибут, относятся формы, элементы управления, фреймы, гиперссылки и некоторые другие.

Но почему же нельзя пользоваться только атрибутом `ID`? Все дело в проклятой несовместимости Navigator 4.*x* и интернет-стандартов. Старые версии Navigator используют атрибут `ID` только для присвоения элементам стилей, а для задания имен признают только атрибут `NAME`. Видите, какая морока!..

Дав с помощью атрибутов `ID` или `NAME` элементу страницы уникальное имя, вы можете обращаться к нему из сценария, вызывая его методы и свойства. Интерпретатор сам найдет нужный элемент по его имени. Для доступа к элементу страницы в Internet Explorer используется синтаксис:

```
<Имя элемента, заданное в атрибутах ID или NAME>.<Свойство или метод>
```

Если же вы пишете сценарий для Navigator 4.*x*, делайте так:

```
document["<Имя элемента, заданное в атрибуте NAME>"].<Свойство или метод>
```

Давайте приведем небольшой пример, поясняющий вышесказанное.

```
<P ID="para" STYLE="color: #0000FF">Некий текст.</P>
```

Сначала мы создали текстовый абзац, назвали его `para` (обратите внимание на значение атрибута `ID`) и присвоили ему встроенный стиль, задающий цвет текста. Сейчас мы этот цвет текста изменим, воспользовавшись Web-сценарием.

```
para.style.color = "#FF0000";
```

Здесь нужны дополнительные пояснения. Дело в том, что каждый элемент страницы в объектной модели документа имеет внутренний объект `style`, дающий доступ к его встроенному стилю. Обратиться к этому объекту можно через одноименное свойство, что мы и сделали. А уж объект `style` предоставляет доступ ко всем атрибутам встроенного стиля через одноименные свойства. В данном примере для получения доступа к значению атрибута стиля `color` использовалось свойство `color`.

Вы можете изменить выравнивание текста этого абзаца, обратившись к свойству `align`:

```
para.align = "center";
```

Свойство `align` предоставляет доступ к значению атрибута `ALIGN` тега `<P>`.

К несчастью, вышеприведенный пример будет работать только в самых последних программах Web-обозревателей. Пресловутый Navigator 4.*x* не позволяет изменять внешний вид и содержимое элементов страницы после ее загрузки. Исключение составляют только графические изображения, фреймы и слои.

А вот этот код будет работать везде:

```
<IMG NAME="someImage" SRC="image1.gif">
. . .
document["someImage"].src = "image2.gif";
```

Он меняет файл, содержимое которого отображается в элементе графического изображения, на другой. Для этого он присваивает иное значение свойству `src` объекта `someImage`, предоставляющее доступ к атрибуту `SRC` тега `<IMG>`.

Чтобы получить доступ к самой Web-странице, воспользуйтесь системным объектом `document`. В частности, вы можете задать цвет гиперссылок, воспользовавшись свойством `linkColor`, предоставляющим доступ к значению атрибута `LINK` тега `<BODY>`:

```
document.linkColor = "#FF0000";
```

Оба рассмотренных нами объекта представляют собой соответственно видимый элемент страницы и саму страницу. Оба этих объекта были созданы с помощью тегов HTML. Теперь же мы начнем рассмотрение объектов, не являющихся элементами страницы. Это объекты Web-обозревателя, не видимые пользователю.

Объект `document` имеет внутренний объект `location`, предоставляющий доступ к интернет-адресу страницы и различным его частям. Доступен он через одноименное свойство. Воспользовавшись этим объектом, мы можем узнать, с какого интернет-адреса была загружена данная страница:

```
address = document.location.href;
```

выяснить имя файла этой страницы:

```
filename = document.location.pathname;
```

либо загрузить другую страницу:

```
document.location.href = "http://www.othersite.ru/otherpage.htm";
```

Объект `window` представляет текущее окно Web-обозревателя либо текущий фрейм, если страница загружена во фрейме. С помощью этого объекта вы можете, например, закрыть это окно:

```
window.close();
```

или заменить текст, отображаемый в его строке статуса:

```
window.status = "Сейчас работает Web-сценарий!";
```

Объект `window` имеет внутренний объект `navigator`, предоставляющий доступ к самой программе Web-обозревателя. Доступен он через одноименное свойство. Воспользовавшись этим объектом, мы можем выяснить, например, версию программы:

```
version = window.navigator.appVersion;
```

или название:

```
programName = window.navigator.appName;
```

Объект `window` имеет внутренний объект `history`, предоставляющий доступ к списку "истории" Web-обозревателя. Он доступен также через одноименное свойство. Воспользовавшись этим объектом, мы можем "путешествовать" по списку "истории" вперед:

```
window.history.forward();
```

и назад:

```
window.history.back();
```

В объектной модели документа существует также еще несколько других объектов, но рассматриваться они не будут. Эти объекты применяются достаточно редко и в весьма специфических случаях. Если же вы все-таки захотите узнать побольше об объектной модели документа, обратитесь к соответствующей литературе.

Как пишутся Web-сценарии

Вот и закончился ликбез по JavaScript и объектной модели документов. Пришло время поговорить о том, как же пишутся Web-сценарии.

Прежде всего, выясним, как JavaScript-код помещается в HTML-код. Для этого служит парный тег `<SCRIPT>...</SCRIPT>`, внутри которого помещается

код сценария. Вне этого тега помещать сценарий можно, но выполняться он не будет.

Вообще, все Web-сценарии можно разделить на два вида: выполняющиеся при загрузке страницы *(загрузочные)* и вызывающиеся в ответ на событие *(обработчики событий)*. В зависимости от того, является сценарий загрузочным или обработчиком, различается способ его реализации.

Сценарии, выполняющиеся при загрузке страницы, представляют собой обычный код на JavaScript, помещенный в тег <SCRIPT>. Он имеет такой вид:

```
<FORM ACTION="http://www.somesite.ru/cgi/program.pl">
  <INPUT TYPE="text" NAME="txtDate">
  <SCRIPT>
    var d;
    d = new Date();
    document.forms[0].txtDate.value = d.toString();
  </SCRIPT>
</FORM>
```

Этот сценарий помещает в поле ввода txtDate значение текущей даты. (О формах и элементах управления *см. главу 16.*) Для этого он вызывает свойство value, отображающее значение, введенное в поле ввода. Заметьте, что для именования поля ввода мы использовали атрибут NAME, поэтому вышеприведенный код будет работать также и в Navigator 4.*x*.

Этот сценарий будет выполняться непосредственно при загрузке страницы. Сначала Web-обозреватель загрузит и распознает тег <FORM>, потом — тег <INPUT>, а сразу за ним выполнит наш сценарий. Такие сценарии очень часто используются для занесения начальных значений в элементы управления и вообще для выполнения различных предустановок.

Мы поместили код сценария сразу же за тегом, задающим поле ввода, к которому мы обращаемся. Если бы мы поместили сценарий перед тегом, Web-обозреватель не смог бы найти объект txtDate, т. к. он еще не был бы создан, и выдал бы сообщение об ошибке. Имейте это в виду, когда будете писать загрузочные сценарии.

Если нужно, чтобы какие-либо предустановки выполнились перед тем, как будет загружен какой-либо "видимый" на странице HTML-код, выполняющий их сценарий помещается в секцию HTML-заголовка страницы (тег <HEAD>). В основном, это касается кода, выполняющего предустановки для других сценариев, чаще всего, обработчиков событий.

Теперь поговорим о сценарии обработчика событий. И рассмотрим пример такого обработчика:

```
<HEAD>
  . . .
  <SCRIPT>
    function para_onClick() {
      para.style.color = "#FF0000";
    }
  </SCRIPT>
</HEAD>
<BODY>
  . . .
  <P ID="para" STYLE="color: #0000FF" onClick="para_onClick();">Некий
    ↳текст.</P>
  . . .
</BODY>
```

Это как раз пример сценария, помещаемого в секцию HTML-заголовка страницы. Он представляет собой функцию `para_onClick`, чей код меняет цвет текста абзаца `para`. Поместив сценарий в секцию HTML-заголовка, мы можем быть уверены, что Web-обозреватель обработает этот код перед тем, как будет загружена страница, и "отложит" определения функций в "долгий ящик", чтобы вызвать их впоследствии, при наступлении соответствующего события.

А теперь обратимся к нашему текстовому абзацу. Что видим? Нечто странное... Откуда-то взялся новый, не знакомый нам атрибут `onClick`. Что он делает?

Это не совсем атрибут. Вернее, совсем не атрибут, хотя и выглядит похоже. Таким способом в HTML к какому-либо событию, происходящему в элементе страницы, привязывается обработчик. Общий синтаксис такой "привязки" следующий:

```
<Имя события>="<Код сценария обработчика>"
```

В данном случае мы привязали вызов функции `para_onClick` в качестве обработчика к событию `onClick`, происходящему, когда пользователь щелкает мышью по этому абзацу.

Вообще, код этого обработчика столь мал, что его можно без всяких последствий поместить прямо в тег `<P>`:

```
<P ID="para" STYLE="color: #0000FF" onClick="para.style.color =
  ↳'#FF0000';">Некий текст.</P>
```

Таким образом, мы значительно уменьшим размер HTML-кода страницы, что нам совсем не помешает.

Простейший Web-сценарий

Давайте создадим простейшую Web-страничку, содержащую работающий Web-сценарий. Этот сценарий будет выводить текущие дату и время.

```
<HTML>
  <HEAD>
    <TITLE>Сегодня</TITLE>
  </HEAD>
  <BODY>
    <P>
      <SCRIPT LANGUAGE="JavaScript">
        var d;
        d=new Date();
        document.write(d.toString());
      </SCRIPT>
    </P>
  </BODY>
</HTML>
```

Сохраните этот код в файле под именем 13.1.htm и откройте в Web-обозревателе. Вы увидите, что на странице будут стоять сегодняшние дата и время (рис. 13.1).

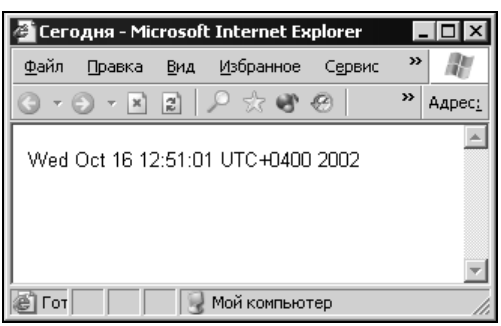


Рис. 13.1. Текущая дата на Web-странице

Давайте рассмотрим наш пример подробнее.

Если мы отбросим весь маловажный код, у нас останется единственный текстовый абзац `<P>`, внутри которого помещен сценарий. Вот он:

```
<P>
  <SCRIPT LANGUAGE="JavaScript">
    var d;
    d=new Date();
```

```
    document.write(d.toString());
  </SCRIPT>
</P>
```

Сам абзац не представляет собой ничего особенного. Сценарий — по большому счету, тоже. Мы уже познакомились с такими сценариями; они выполняются непосредственно при загрузке страницы и обычно производят различные предустановки. В нашем случае такой предустановкой является помещение в текстовый абзац строки, содержащей сегодняшние дату и время.

Как это происходит? С помощью метода `write` объекта `document`. Автор не рассматривал этот метод, когда говорил об объекте `document`. Все объекты, входящие в объектную модель документа, имеют множество свойств и методов, многие из которых используются в сценариях довольно часто. А объем книги ограничен; мы не можем уделять слишком много места описанию объектной модели в ущерб Dreamweaver. Поэтому кое о чем автор умолчал.

Теперь же настало время рассказать о методе `write`. Он помещает строку, переданную ему в качестве параметра, в то место документа, где встретилось выражение, вызывающее этот метод. Только и всего. В нашем случае он проделывает это со строковым значением текущей даты и помещает его внутри текстового абзаца `<P>`.

Остальной код вполне понятен. Мы объявляем переменную, помещаем в него значение текущей даты в виде объекта класса `Date` и преобразуем это значение в строку методом `toString`. Вы также можете для этого использовать метод `toLocaleString`, выполняющий это преобразование с учетом национальных настроек операционной системы, но такой код, возможно, будет работать не во всех Web-обозревателях.

Более сложный Web-сценарий

В прошлой главе говорилось об анимации. Давайте же рассмотрим Web-сценарий, анимирующий какой-либо элемент страницы, а именно, заставляющий его двигаться по горизонтали взад-вперед. Вы увидите, что все это делается очень просто и довольно очевидно.

```
<HTML>
  <HEAD>
    <TITLE>Анимация</TITLE>
    <STYLE>
      DIV { font-size: 72; font-weight: bold }
    </STYLE>
    <SCRIPT>
```

```
      var dx, timer;
      dx = 2;

      function moveImage() {
        livediv.style.pixelLeft += dx;
        if (livediv.style.pixelLeft + livediv.style.pixelWidth >=
          ↳document.body.clientWidth) dx = -dx;
        if (livediv.style.pixelLeft <= 0) dx = -dx;
      }

      function setupAnimation() {
        timer = window.setInterval("moveImage()", 100)
      }
    </SCRIPT>
  </HEAD>
  <BODY onLoad="setupAnimation();">
    <DIV ID="div1" STYLE="top: 50; left: 50; position: absolute;
      ↳z-index: 1">J</DIV>
    <DIV ID="div2" STYLE="top: 50; left: 100; position: absolute;
      ↳z-index: -1">a</DIV>
    <DIV ID="div3" STYLE="top: 50; left: 150; position: absolute;
      ↳z-index: 1">v</DIV>
    <DIV ID="div4" STYLE="top: 50; left: 200; position: absolute;
      ↳z-index: -1">a</DIV>
    <DIV ID="livediv" STYLE="top: 80; left: 0; width: 30;
      ↳position: absolute"><IMG SRC="tips.gif"></DIV>
  </BODY>
</HTML>
```

Сохраните этот код в файле под именем 13.2.htm. Графическое изображение tips.gif, использованное в этом примере, вы можете найти в папке Web, вложенной в папку Windows или WinNT. Поместите файл tips.gif в ту же папку, где находится файл 13.2.htm, после чего откройте последний в Web-обозревателе. К сожалению, рис. 13.2 не может передать движение.

Рассмотрение кода нашего примера начнем с секции HTML-заголовка страницы (тег `<HEAD>`). Здесь у нас находится небольшая таблица стилей с единственным стилем, переопределяющим шрифт тега `<DIV>`. Но, кроме того, здесь располагается код сценария. Он-то нас и интересует.

Этот код включает определение двух функций: `setupAnimation` и `moveImage`. Первая функция выполняет некие предустановки, необходимые для правильной работы анимации, а вторая — осуществляет само движение анимированного элемента. (В дальнейшем вы тоже старайтесь давать вашим функциям "говорящие" имена.) Но, кроме этих функций, здесь есть и другой код, выполняющийся при загрузке страницы:

```
var dx, timer;
dx = 2;
```

В этом коде мы объявляем две переменные: `dx` и `timer`. Первая из них задает значение приращения, "скачка" анимации, и с ней все понятно. Мы присваиваем ей значение 2; если движение анимированного элемента покажется вам слишком медленным, вы можете это значение увеличить. А вот второй переменной займемся поближе.

Рис. 13.2. Анимированный элемент страницы

Уже не раз говорилось, что анимация всегда привязывается к системному таймеру. Это позволяет проигрывать одно и то же "кино" и на самых медленных, и на самых быстрых компьютерах с одинаковой скоростью. Как правило, делается это следующим образом: пишется специальная функция-обработчик события "тика" этого системного таймера, которая и заставляет анимированный элемент двигаться.

Для того чтобы привязать функцию-обработчик к событию "тика" таймера, нужно использовать метод `setInterval` объекта `window`. Этот метод принимает в качестве параметра строку с именем функции-обработчика и интервал между "тиками" в миллисекундах, а возвращает особое значение, называемое *идентификатором интервала*. Впоследствии, когда анимация должна быть закончена, нужно будет вызвать метод `clearInterval` объекта `window`, передав ему этот самый идентификатор. Но, т. к. у нас анимация проигрывается бесконечно, то и метод `clearInterval` не вызывается, и идентификатор, хранимый в переменной `timer`, фактически не нужен.

Функцию-обработчик мы привязываем к "тику" таймера в функции `setupAnimation`. Ее код приведен ниже.

```
function setupAnimation() {
  timer = window.setInterval("moveImage()", 100)
}
```

Эта функция, в свою очередь, является обработчиком события onLoad, возникающего сразу по окончании загрузки Web-страницы. В качестве параметров метода setInterval мы передаем имя функции moveImage и 100 — интервал между "тиками" в миллисекундах. Обратите также внимание на следующий код:

```
<BODY onLoad="setupAnimation();">
```

С его помощью к событию onLoad привязывается функция-обработчик.

Обратимся теперь к функции, выполняющей движение анимированного элемента, — moveImage. Ее код выглядит так:

```
function moveImage() {
  livediv.style.pixelLeft += dx;
  if (livediv.style.pixelLeft + livediv.style.pixelWidth >=
    ↳document.body.clientWidth) dx = -dx;
  if (livediv.style.pixelLeft <= 0) dx = -dx;
}
```

Разберем его по строкам.

Первая строка:

```
livediv.style.pixelLeft += dx;
```

или, как понятнее,

```
livediv.style.pixelLeft = livediv.style.pixelLeft + dx;
```

увеличивает значение свойства pixelLeft объекта Style на значение переменной dx. Свойство pixelLeft представляет значение горизонтальной координаты элемента страницы в пикселах.

Вторая строка:

```
if (livediv.style.pixelLeft + livediv.style.pixelWidth >=
  ↳document.body.clientWidth) dx = -dx;
```

осуществляет проверку, дошел ли анимированный элемент до правого края страницы. Для этого мы сначала складываем значение уже знакомого свойства pixelLeft и нового свойства pixelWidth, представляющего ширину элемента страницы в пикселах. После этого мы сравниваем получившуюся сумму со значением свойства clientWidth объекта body и, если она стала больше этого значения, меняем знак у значения переменной dx. Свойство clientWidth представляет ширину элемента страницы в пикселах, а объект

body — саму Web-страницу, значит, выражение body.clientWidth возвращает значение ширины всей страницы в пикселах.

И последняя, третья, строка:

```
if (livediv.style.pixelLeft <= 0) dx = -dx;
```

осуществляет проверку, дошел ли анимированный элемент до левого края страницы. Для этого мы просто сравниваем значение свойства pixelLeft с нулем и, если оно стало меньше нуля, изменяем знак значения переменной dx.

После того как мы сменим знак значения dx, анимированный элемент "поедет" в противоположную сторону. Таким образом, анимация будет проигрываться бесконечно.

Кстати, рассмотренный нами Web-сценарий не будет работать только в Navigator 4.*x*. Чтобы он работал в этой капризной программе, его нужно немного изменить. Измененный код страницы 13.2.htm по имени 13.3.htm приведен ниже. Подробно описываться он не будет — попробуйте разобраться с этим кодом сами, на досуге.

```
if (livediv.style.pixelLeft <= 0) dx = -dx;
<HTML>
  <HEAD>
    <TITLE>Анимация</TITLE>
    <SCRIPT>
      var dx, timer;
      dx = 2;

      function moveImage() {
        document.livediv.left += dx;
        if (document.livediv.left >= document.width) dx = -dx;
        if (document.livediv.left <= 0) dx = -dx;
      }

      function setupAnimation() {
        timer = window.setInterval("moveImage()", 100)
      }
    </SCRIPT>
  </HEAD>
  <BODY onLoad="setupAnimation();" STYLE="font-size: 72;
    ↳font-weight: bold">
    <DIV ID="div1" STYLE="top: 50; left: 50; position: absolute;
      ↳z-index: 1">J</DIV>
    <DIV ID="div2" STYLE="top: 50; left: 100; position: absolute;
      ↳z-index: -1">a</DIV>
```

```
    <DIV ID="div3" STYLE="top: 50; left: 150; position: absolute;
      ↳z-index: 1">v</DIV>
    <DIV ID="div4" STYLE="top: 50; left: 200; position: absolute;
      ↳z-index: -1">a</DIV>
    <DIV ID="livediv" STYLE="top: 80; left: 0; width: 30;
      ↳position: absolute"><IMG SRC="tips.gif"></DIV>
  </BODY>
</HTML>
```

Правда, исправленный вариант страницы перестанет работать в других Web-обозревателях...

Вот такая анимация...

Web-сценарии — подход Dreamweaver. Поведения

Сейчас, когда все объяснено, вам может показаться, что писать Web-сценарии проще простого. Однако это совсем не так. Web-программирование — штука очень и очень непростая. Хотя бы из-за того, что вам придется держать в голове множество имен объектов, классов, свойств, методов и событий, с помощью которых создается тот или иной эффект. Вдобавок многие вещи делаются отнюдь не так очевидно, как, скажем, анимация. Поверьте, но над первыми своими Web-сценариями автор сидел не один час.

Но разработчики Dreamweaver решили облегчить жизнь начинающих Web-программистов. Для этого они ввели понятие *поведения* (по-английски — behavior) — заранее определенного действия, которое будет происходить в ответ на какое-либо событие, произошедшее внутри или вне системы. Dreamweaver поддерживает достаточный набор таких поведений, который покроет практически все потребности начинающего, да и опытного Web-дизайнера. Работа с поведениями в среде Dreamweaver осуществляется так же просто, как и с анимациями — щелчками мыши и нажатиями клавиш клавиатуры.

Когда вы задаете какое-либо поведение для того или иного элемента страницы, Dreamweaver автоматически создает необходимый для этого набор Web-сценариев и помещает его в HTML-код страницы. Это делается незаметно от пользователя; пользователь просто работает со списком созданных им поведений, отображаемым в специальном списке. Таким образом, Dreamweaver дает возможность прикоснуться к Web-программированию даже тем Web-дизайнерам, которые не знают ни языка JavaScript, ни объектной модели документа.

Все поведения, поддерживаемые Dreamweaver, приведены в табл. 13.2.

Таблица 13.2. *Поведения, поддерживаемые Dreamweaver*

Название в терминологии Dreamweaver	Описание
`Call JavaScript`	Вызов фрагмента JavaScript-кода или функции, написанной на JavaScript
`Change Property`	Изменение значения свойства какого-либо объекта
`Check Browser`	Перенаправление посетителей на разные Web-страницы в зависимости от программы Web-обозревателя
`Check Plugin`	Перенаправление посетителей на разные Web-страницы в зависимости от того, установлен ли у них нужный модуль расширения Web-обозревателя
`Control Shockwave or Flash`	Управление проигрыванием фильма Shockwave или Flash
`Drag Layer`	Позволяет посетителю перетаскивать свободно позиционируемый элемент страницы с места на место
`Go to Timeline Frame`	Перемещение к заданному кадру анимации
`Go to URL`	Перенаправление посетителей на другую Web-страницу
`Hide Pop-Up Menu`	Скрытие меню гиперссылок, выведенного ранее на экран с помощью поведения `Show Pop-Up Menu`
`Jump Menu`	Привязывается самим Dreamweaver при создании раскрывающегося списка гиперссылок. Пользователю обычно не нужно создавать это поведение самому
`Jump Menu Go`	Привязывается самим Dreamweaver при создании раскрывающегося списка гиперссылок к кнопке **Go** (Перейти). Пользователю обычно не нужно создавать это поведение самому
`Open Browser Window`	Открытие какой-либо Web-страницы в новом окне Web-обозревателя
`Play Sound`	Проигрывание аудиофайла
`Play Timeline`	Запуск проигрывания анимации
`Popup Message`	Вывод окна-предупреждения с заданным текстом и кнопкой **OK**
`Preload Images`	Загрузка из файлов графических изображений. Эти изображения могут использоваться затем для вывода на страницу в качестве ответа на событие
`Set Nav Bar Image`	Создается самим Dreamweaver при создании полосы навигации. Пользователю обычно не нужно создавать это поведение самому

Таблица 13.2 *(окончание)*

Название в терминологии Dreamweaver	Описание
`Set Text of Frame`	Помещение какого-либо текста во фрейм
`Set Text of Layer`	Помещение какого-либо текста в свободно позиционируемый элемент
`Set Text of Status Bar`	Помещение какого-либо текста в строку статуса окна Web-обозревателя
`Set Text of Text Field`	Помещение какого-либо текста в поле ввода
`Show Pop-Up Menu`	Вывод на экран меню гиперссылок, например, в ответ на помещение посетителем страницы курсора мыши над каким-либо элементом страницы
`Show-Hide Layers`	Показ или скрытие свободно позиционируемого элемента
`Stop Timeline`	Остановка проигрывания анимации
`Swap Image`	Замена одного графического изображения другим
`Swap Image Restore`	Восстановление изначально присутствовавшего на странице графического изображения после его замены с помощью поведения `Swap Image`
`Validate Form`	Проверка введенных в форму данных на правильность (о формах *см. главу 16*)

Осталось привести список поддерживаемых Dreamweaver событий (табл. 13.3). Имейте, однако, в виду, что это не полный список поддерживаемых событий — некоторые события используются достаточно редко, и поэтому описываться здесь не будут.

Таблица 13.3. *События, поддерживаемые Dreamweaver*

Название	Описание
`onAbort`	Наступает, когда пользователь останавливает загрузку Web-страницы, например, нажав кнопку **Останов**
`onBlur`	Наступает, когда элемент управления или сама страница теряет фокус ввода
`onChange`	Наступает, когда посетитель изменяет значение элемента управления
`onClick`	Наступает при щелчке мышью по элементу страницы или по самой странице

Таблица 13.3 *(окончание)*

Название	Описание
onDblClick	Наступает при двойном щелчке мышью по элементу страницы или по самой странице
onError	Наступает при возникновении ошибки при загрузке страницы, графического изображения или внедренного элемента
onFocus	Наступает, когда элемент управления или сама страница получает фокус ввода
onHelp	Наступает, когда пользователь вызывает интерактивную справку Web-обозревателя
onKeyDown	Наступает при нажатии клавиши клавиатуры
onKeyPress	Наступает при нажатии и отпускании клавиши клавиатуры
onKeyUp	Наступает при отпускании клавиши клавиатуры
onLoad	Наступает по окончании загрузки страницы, графического изображения или внедренного элемента
onMouseDown	Наступает при нажатии кнопки мыши
onMouseMove	Периодически наступает при перемещении курсора мыши над элементом страницы или над самой страницей
onMouseOut	Наступает при "уводе" курсора мыши с элемента страницы
onMouseOver	Наступает, когда курсор мыши "заходит" на элемент страницы
onMouseUp	Наступает при отпускании кнопки мыши
onMove	Наступает при перемещении окна Web-обозревателя
onReset	Наступает, когда посетитель нажимает кнопку **Сброс** формы
onResize	Наступает при изменении размеров окна Web-обозревателя
onScroll	Наступает при прокрутке содержимого страницы или ее элемента посетителем
onSelect	Наступает при выделении текста в поле ввода
onSubmit	Наступает, когда посетитель нажимает кнопку **Отправить** формы, запуская отправку данных
onUnload	Наступает, когда посетитель покидает текущую Web-страницу

На этом закончим наше затянувшееся вступление. Пора переходить к работе над поведениями в Dreamweaver.

Работа с поведениями

Здесь мы изучим приемы работы с поведениями в нашем любимом Dreamweaver. Изучать мы их будем на примере страницы default2.htm, созданной в качестве новой главной страницы первого сайта Sample site 1. Откройте ее в окне документа.

Панель *Behaviors*

Вся работа с поведениями будет протекать в специальной панели **Behaviors**. Чтобы вызвать ее на экран, включите пункт-выключатель **Behaviors** меню **Window** или нажмите комбинацию клавиш <Shift>+<F3>. Панель **Behaviors** показана на рис. 13.3.

Рис. 13.3. Панель **Behaviors**

Большую часть этой панели занимает список уже созданных к этому времени поведений, привязанных к выделенному в окне документов элементу страницы. Наименование тега элемента, к которому привязаны отображаемые в списке поведения (по умолчанию это сама страница), находится выше этого списка. Если вы выделите в окне документа другой элемент, в списке панели тотчас отобразятся привязанные к нему поведения. К сожалению, узнать, к какому элементу страницы привязаны поведения, иным способом нельзя.

Список поведений представляет собой таблицу из двух колонок: **Events** (событие) и **Actions** (действие, происходящее в ответ на событие). Вы можете выбрать любое поведение в списке и произвести над ним какие-либо манипуляции.

Если вы поставите текстовый курсор на саму Web-страницу, выделив ее таким образом, то увидите, что в списке уже присутствует одно поведение —

`Play Animation`, привязанное к событию `onLoad`. Это поведение было добавлено самим Dreamweaver, когда мы включили флажок **Autoplay** в панели **Timelines**. Так что некоторые поведения создаются самим Dreamweaver без вашего ведома.

Вы, наверно, уже заметили, что в столбце **Events** списка, правее отображаемого в нем значения, находится небольшая кнопка со стрелкой, направленной вниз. При нажатии на эту кнопку на экране появляется *меню событий*, в котором вы можете выбрать событие, в ответ на которое будет происходить заданное вами действие. Это меню показано на рис. 13.4. Таким образом, вы можете изменить событие, в ответ на которое выполняется какое-либо действие, например, заставить анимацию проигрываться в ответ на щелчок мышью.

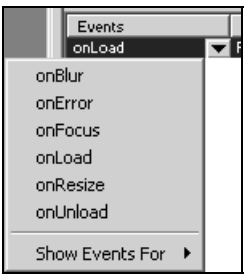


Рис. 13.4. Меню событий панели **Behaviors**

Чтобы отобразить в этом меню только события, поддерживаемые какой-либо программой Web-обозревателя, выберите пункт **Show Events For**. На экране появится небольшое подменю; включите пункт-выключатель, соответствующий нужной программе. Все пункты этого подменю перечислены в табл. 13.4.

Таблица 13.4. *Пункты подменю* ***Show Events For*** *меню событий*

Пункт подменю	Отображаемые события
3.0 and Later Browsers	Internet Explorer и Navigator, версии 3.0 и более поздние
4.0 and Later Browsers	Internet Explorer и Navigator, версии 4.0 и более поздние
IE 3.0	Internet Explorer 3.0
IE 4.0	Internet Explorer 4.0
IE 5.0	Internet Explorer 5.0
IE 5.5	Internet Explorer 5.5
IE 6.0	Internet Explorer 6.0

Таблица 13.4 *(окончание)*

Пункт подменю	Отображаемые события
Netscape 3.0	Navigator 3.0
Netscape 4.0	Navigator 4.0
Netscape 6.0	Navigator 6.0

Dreamweaver также предоставляет вам возможность изменить параметры выбранного в списке поведения. Для этого просто дважды щелкните по нему мышью, после чего на экране появится диалоговое окно задания параметров соответствующего поведения. Например, для поведения `Play Timeline` такое окно выглядит так, как на рис. 13.5. В раскрывающемся списке **Play Timeline** выберите нужную анимацию и не забудьте нажать кнопку **OK** для сохранения сделанных изменений или **Cancel** — для отказа от них.

Рис. 13.5. Диалоговое окно **Play Timeline**

Внимание!

Когда будете изменять событие или параметры поведения, не забудьте выделить в окне документа нужный элемент страницы.

В верхней части панели **Behaviors** находятся четыре кнопки: +, −, ↑ и ↓. Давайте рассмотрим их по очереди.

Кнопка со знаком "плюс" позволит вам создать новое поведение. При ее нажатии на экране появляется довольно большое *меню поведений* (рис. 13.6), в котором можно выбрать нужное поведение. Некоторые из пунктов этого меню могут быть недоступными; это обычно означает, что в данный момент времени на странице нет необходимых элементов. Также некоторые поведения могут "прятаться" в подменю, например поведения, управляющие анимациями, находятся в подменю **Timeline**.

С помощью пунктов уже знакомого вам подменю **Show Events For**, находящегося в этом же меню, можно задать, поддерживаемые каким Web-обозревателем события будут отображаться в меню событий. Пункты этого подменю перечислены в табл. 13.4. Чтобы вывести нужный набор событий, просто выберите соответствующий пункт-выключатель.

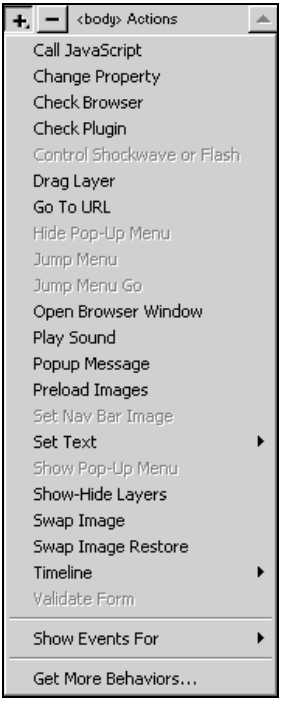


Рис. 13.6. Меню поведений панели **Behaviors**

Итак, чтобы создать новое поведение, сначала выделите нужный элемент страницы. Вы также можете привязать поведение к самой странице, для чего щелкните по пустому месту на ней или выберите тег `<BODY>` в строке тегов. Затем выберите необходимый пункт в меню поведений. После этого на экране появится диалоговое окно задания параметров выбранного вами поведения. Такое окно для поведения `Play Timeline` показано на рис. 13.5. Задайте требуемые параметры и нажмите кнопку **OK**. Диалоговые окна задания параметров для других поведений будут рассмотрены ниже.

Кнопка со знаком "минус" позволит вам удалить выбранное в списке поведение. С таким же успехом вы можете нажать клавишу <Del>. Не забудьте только выделить в окне документа нужный элемент страницы.

Кнопки ↑ и ↓ перемещают выбранное в списке поведение на позицию вверх или вниз. Дело в том, что если несколько поведений привязано к одному и тому же событию, они выполняются в порядке следования в списке поведений. С помощью кнопок ↑ и ↓ вы можете изменить этот порядок.

Создание поведений

Теперь настало время рассмотреть все поведения, предлагаемые нам Dreamweaver, их создание и все их параметры, которые вы можете задать.

Название поведения в терминологии самого Dreamweaver приведено в скобках, так что вы сразу сможете найти его в меню поведений.

Имейте только в виду, что здесь приводятся не все поведения, поддерживаемые Dreamweaver. Поведения, служащие для поддержки форм и элементов управления, рассмотрим в *главе 16*.

Вызов JavaScript-кода *(Call JavaScript)*

Если вы хотите, чтобы в ответ на какое-либо событие, произошедшее в том или ином элементе страницы, выполнялся некий JavaScript-код, создайте для этого элемента поведение `Call JavaScript`. Для этого вызовите одноименный пункт меню поведений. После этого на экране появится диалоговое окно **Call JavaScript**, показанное на рис. 13.7.

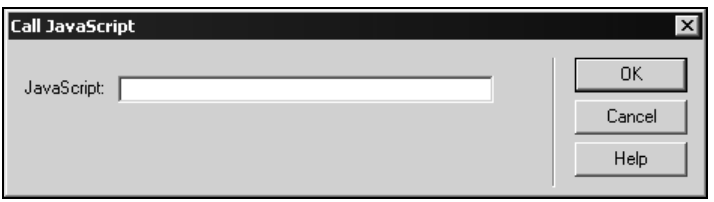


Рис. 13.7. Диалоговое окно **Call JavaScript**

В этом окне находится единственное поле ввода **JavaScript**, в котором вводится нужный программный код. Это может быть, например, вызов какой-либо функции, написанной вами и помещенной в секцию заголовка страницы, вызов метода, принадлежащего какому-либо элементу или объекту, или иной код. Не забудьте только нажать кнопку **OK** после того, как введете этот код.

Изменение значения свойства *(Change Property)*

Чтобы изменить в ответ на происшедшее событие значение какого-либо свойства того или иного объекта, воспользуйтесь поведением `Change Property`. Выберите одноименный пункт меню поведений. На экране появится диалоговое окно **Change Property** (рис. 13.8).

Прежде всего необходимо выбрать тег элемента страницы, свойство которого будет изменено. Это делается с помощью раскрывающегося списка **Type of Object**. После этого выберите сам объект в раскрывающемся списке **Named Object**. Имейте в виду, что в этом списке отображаются имена элементов, чей тег выбран в списке **Type of Object**; для элементов, не имеющих уникального имени, отображается строка **unnamed**. Поэтому автор рекомендует для всех элементов страницы, которые вы собираетесь использовать в сценариях, задавать уникальное имя.

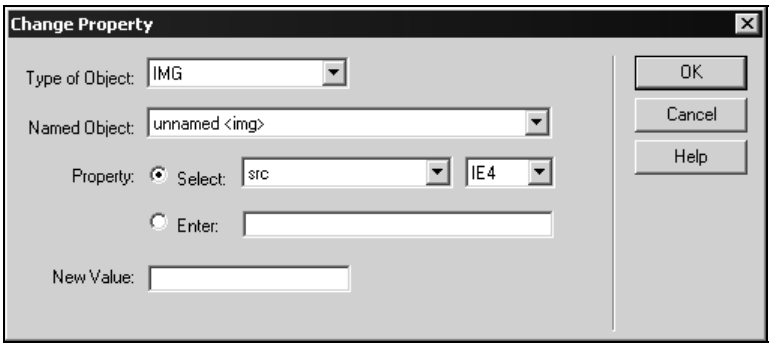


Рис. 13.8. Диалоговое окно **Change Property**

Набор переключателей **Property** позволяет выбрать способ задания нужного свойства.

Если выбран переключатель **Select**, вы сможете выбрать нужное свойство в раскрывающемся списке, находящемся правее этого переключателя. Имейте, однако, в виду, что в этом списке для многих объектов отображаются как свойства самого этого объекта, так и его внутреннего объекта `style`. Таким образом, вы можете изменить также и значения атрибутов его стиля.

Правее раскрывающегося списка свойств находится еще один раскрывающийся список. С его помощью вы можете задать, поддерживаемые каким Web-обозревателем свойства будут отображаться в списке свойств. В этом списке доступны четыре пункта:

- **NS3** — отображаются только свойства, поддерживаемые Navigator 3.0;
- **IE3** — Internet Explorer 3.0;
- **NS4** — Navigator 4.0;
- **IE4** — Internet Explorer 4.0 (выбран по умолчанию).

Если выбран переключатель **Enter**, вы можете ввести имя нужного свойства в поле ввода, расположенном правее. Используйте эту возможность для ввода имен свойств, поддерживаемых новейшими программами Web-обозревателей, но не "известных" Dreamweaver.

Осталось ввести новое значение свойства в поле ввода **New Value** — и можно нажимать кнопку **OK**.

Перенаправление на другую страницу в зависимости от версии Web-обозревателя *(Check Browser)*

Несовместимость различных Web-обозревателей уже давно стала притчей во языцех. В качестве решения этой проблемы Web-дизайнеры очень часто создают различные версии одной и той же Web-страницы, предназначенные для разных Web-обозревателей. Иногда для того, чтобы перенаправить посе-

тителя на соответствующую страницу, используются "говорящие" гиперссылки вида: "Пользователям Navigator 1.0 — сюда. А вы слышали: вышла версия 2.0!". Но чаще всего на главной странице сайта помещается Web-сценарий, определяющий версию программы Web-обозревателя и перенаправляющий посетителя на соответствующую ей страницу.

Практически всегда такие сценарии выполняются при наступлении события `onLoad` объекта страницы (тег `<BODY>`), т. е. когда страница полностью загрузится. При этом посетитель перемещается на другую страницу автоматически, не делая никаких щелчков по гиперссылкам. (Конечно, если его программное обеспечение поддерживает Web-сценарии. В противном случае нужно все-таки будет предусмотреть на главной странице соответствующие гиперссылки, иначе он не попадет, куда нужно.)

Автоматическое перенаправление на другую страницу стало в наше время настолько популярным, что все более-менее мощные Web-редакторы позволяют создавать такие штуковины. Dreamweaver не стал исключением. Если вы выберете пункт **Check Browser** меню поведений, вы в этом убедитесь.

А пока посмотрим на диалоговое окно **Check Browser**, показанное на рис. 13.9. Что же предлагает нам Dreamweaver?

А предлагает он нам три возможности:

- остаться на текущей странице;
- перейти на "основную" страницу ("основная" — в терминологии Dreamweaver);
- перейти на "альтернативную" страницу ("альтернативная" — также в терминологии Dreamweaver).

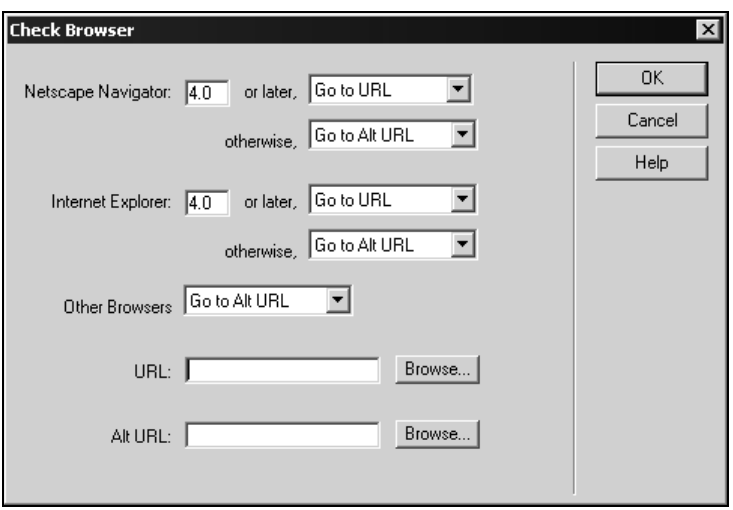


Рис. 13.9. Диалоговое окно **Check Browser**

Интернет-адрес "основной" страницы задается в поле ввода **URL**. Интернет-адрес "альтернативной" страницы — в поле ввода **Alt URL**. Конечно, вы можете нажать кнопку **Browse**, находящуюся правее нужного поля ввода, и выбрать необходимый файл в диалоговом окне **Select File**.

Возможно, выбор всего из трех страниц покажется вам небогатым. Однако, если вы не делаете сверхзаумного сайта, где на каждую версию каждого Web-обозревателя отведена своя версия каждой страницы, вам должно этого хватить.

Группа элементов управления **Netscape Navigator** позволяет задать, на какие страницы будет произведено перенаправление, если посетитель сайта пользуется Navigator. Сама версия Web-обозревателя задается в небольшом поле ввода, называющемся так же — **Netscape Navigator**. Если версия Web-обозревателя окажется такой же, как вы ввели, или более поздней, осуществляется переход на страницу, заданную в раскрывающемся списке **or later**. В противном случае переход будет выполнен на страницу, заданную в раскрывающемся списке **otherwise**.

Раскрывающиеся списки **or later** и **otherwise** имеют по три пункта:

- **Stay on this Page** — оставаться на текущей странице;
- **Go to URL** — перейти на "основную" страницу;
- **Go to Alt URL** — перейти на "альтернативную" страницу.

Как видите, все довольно просто.

Точно так же задаются страницы для пользователей различных версий Internet Explorer. Для этого служит группа элементов управления **Internet Explorer**. Для задания страницы, предназначенной для пользователей других программ Web-обозревателей, используется единственный раскрывающийся список **Other Browsers**.

По умолчанию Dreamweaver предлагает нам следующее:

- пользователей Internet Explorer 4.0 и Navigator 4.0 или более новых версий этих программ — переслать на "основную" страницу;
- пользователей более старых версий Internet Explorer 4.0 и Navigator 4.0, а также других программ Web-обозревателей — переслать на "альтернативную" страницу.

Как правило, этих установок бывает достаточно. Вам остается только задать интернет-адреса "основной" и "альтернативной" Web-страниц. И, разумеется, нажать кнопку **OK**.

Проверка наличия модуля расширения *(Check Plugin)*

В *главе 4*, посвященной графическим изображениям и мультимедийным элементам, помещаемым на Web-страницы, говорилось о модулях расшире-

ния Web-обозревателя — специальных программах, работающих совместно с Web-обозревателем и предназначенных для обработки не поддерживаемых им непосредственно данных. Существуют модули расширения Web-обозревателя для проигрывания фильмов Flash, Shockwave, аудио- и видеоклипов форматов RealMedia, MP3 и т. п. Некоторые из модулей расширения поставляются в составе Web-обозревателей, другие же должны покупаться или загружаться из Интернета отдельно.

Если вы используете в своих страницах мультимедийное или иное содержимое, не поддерживаемое ни самим Web-обозревателем, ни одним из поставляемых с ним модулем расширения, но для которого существуют модули расширения, доступные в Сети, вы должны выполнять проверку, установлен ли этот модуль на компьютере посетителя сайта.

Обычно такая проверка выполняется прямо на Web-странице с данными, отображаемыми с помощью модуля расширения. Если такой модуль установлен, данные будут успешно отображены. В противном случае вы можете перенаправить посетителя на другую страницу с разъяснениями и предложением загрузить соответствующий модуль или теми же данными в более "удобоваримом" для Web-обозревателя формате. Как и в случае проверки версии Web-обозревателя, выполнять проверку установленного модуля расширения лучше всего сразу после загрузки страницы, привязав поведение к событию `onLoad` тега `<BODY>`.

Для выполнения такой проверки вы можете воспользоваться поведением `Check Plugin`, для чего вам достаточно будет выбрать одноименный пункт меню поведений. После этого на экране появится диалоговое окно **Check Plugin**, показанное на рис. 13.10.

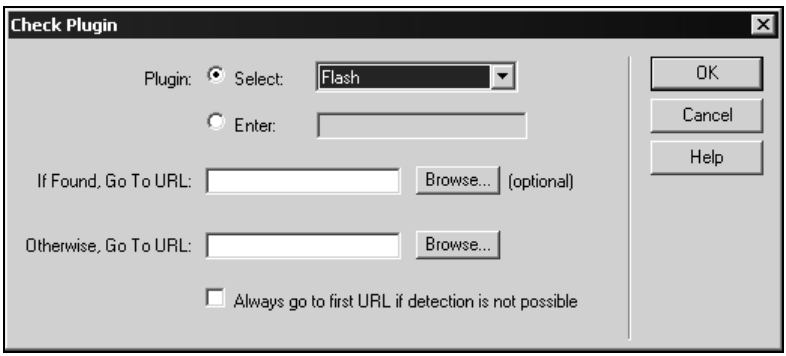


Рис. 13.10. Диалоговое окно **Check Plugin**

Группа переключателей **Plugin** задает способ, которым выбирается модуль расширения. Если вы включите переключатель **Select** (он, кстати, включен по умолчанию), то сможете выбрать нужный модуль расширения в раскры-

вающемся списке, расположенном правее кнопки. В этом списке доступны пять пунктов:

- **Flash** — модуль расширения Macromedia Flash;
- **Shockwave** — Macromedia Shockwave;
- **LiveAudio** — Creative LiveAudio;
- **QuickTime** — Apple QuickTime;
- **Windows Media Player** — стандартный проигрыватель мультимедийных файлов, поставляемый в составе Microsoft Windows.

Как видите, в этом списке перечислены не все модули расширения, которые могут быть реально у вас установлены, а только "известные" Dreamweaver. Если же вам нужно проверить "существование" какого-то другого модуля, выберите переключатель **Enter** и введите имя нужного модуля расширения в поле ввода, расположенное правее этого переключателя.

В поле ввода **If Found, Go To URL** вводится интернет-адрес страницы, на которую будет осуществлен переход в случае, если нужный модуль расширения будет найден на компьютере. Если вы хотите, чтобы в этом случае посетитель оставался на текущей странице, оставьте это поле пустым; часто так и делают.

В поле ввода **Otherwise, Go To URL** вводится интернет-адрес страницы, на которую будет осуществлен переход в случае, если нужного модуля расширения не будет найдено. Если вы хотите, чтобы в этом случае посетитель оставался на текущей странице, оставьте это поле пустым. Обычно, если какого-либо модуля расширения на компьютере клиента нет, выполняется переход на страницу с разъяснениями и предложением установить его, а если такой модуль есть, посетитель остается на той же странице и наблюдает данные, отображаемые с помощью этого модуля.

Бывает так, что Web-сценарий, определяющий присутствие модуля расширения, не работает или работает некорректно. (В частности, так происходит в некоторых версиях Internet Explorer.) В этом случае выполняется автоматический переход на страницу, чей адрес указан в поле ввода **Otherwise, Go To URL**. Если же вы хотите, чтобы в этом случае всегда осуществлялся переход на страницу, чей интернет-адрес указан в поле ввода **If Found, Go To URL** (т. е. как будто проверка прошла удачно, и модуль расширения был бы найден), включите флажок **Always go to first URL if detection is not possible**.

Как обычно, кнопка **OK** сохраняет сделанные вами установки, а кнопка **Cancel** — отменяет их.

Управление фильмом Shockwave или Flash *(Control Shockwave or Flash)*

Видеоролики разной длины и различного качества часто используются на современных Web-страницах. Нередко они управляются Web-сценариями.

Так, фильм Flash может запускаться в ответ на нажатие кнопки или наведение курсора мыши на какой-либо элемент страницы. Существуют сайты, почти целиком реализованные на Flash-фильмах и Web-сценариях, которые ими управляют.

Dreamweaver предоставляет вам возможность управлять фильмом в формате Shockwave или Flash из обработчика того или иного события. Для этого используется поведение `Control Shockwave or Flash`. Для создания такого поведения выберите одноименный пункт меню поведений. Диалоговое окно **Control Shockwave or Flash**, показанное на рис. 13.11, позволит вам задать параметры этого поведения.

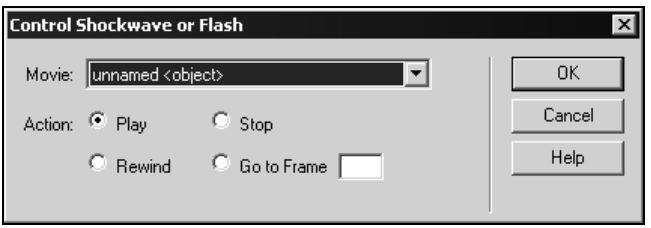


Рис. 13.11. Диалоговое окно **Control Shockwave or Flash**

Прежде всего, выберите соответствующий нужному фильму элемент `<EMBED>` или `<OBJECT>` в раскрывающемся списке **Movie**. Имейте в виду, что в этом меню отображаются имена таких элементов; для тех из них, кто не имеет уникального имени, отображается строка **unnamed**. Поэтому автор рекомендует для всех фильмов, которыми вы собираетесь управлять из сценариев, задавать уникальное имя.

Выбрав фильм, задайте действия, которое хотите над ним совершить. Это действие задается с помощью набора переключателей **Action**. Всего переключателей в этом наборе четыре:

- **Play** — начинает проигрывание фильма;
- **Stop** — останавливает его;
- **Rewind** — перематывает его к началу;
- **Go to Frame** — перематывает его к заданному в расположенном правее поле ввода кадру.

Задав нужные параметры, нажмите кнопку **OK**.

Перетаскивание свободно позиционируемого элемента *(Drag Layer)*

Мы уже говорили об анимированных элементах страницы. Однако вы можете предоставить возможность пользователю самому перетаскивать свободно позиционируемые элементы по странице. Это может быть полезно,

если вы делаете интерактивную игру, наподобие головоломки, учебное пособие или полноценную программу в виде Web-страницы.

Поведение `Drag Layer`, реализующее перетаскивание свободных элементов, привязывается к событию `onLoad` тега `<BODY>`. Выберите одноименный пункт меню поведений; на экране появится диалоговое окно **Drag Layer**, показанное на рис. 13.12.

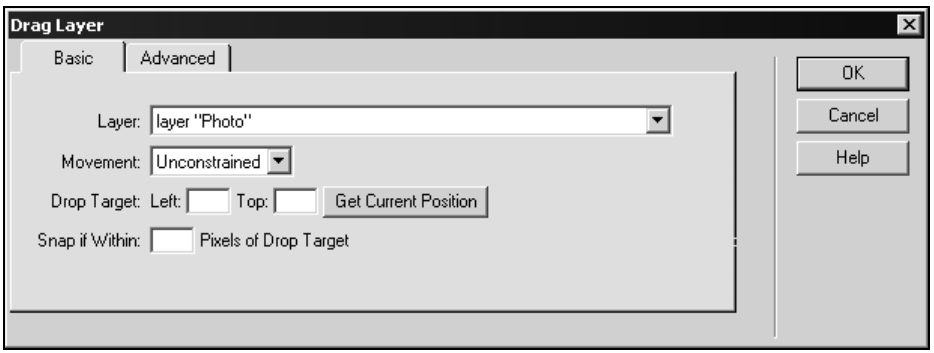


Рис. 13.12. Диалоговое окно **Drag Layer** (вкладка **Basic**)

В раскрывающемся списке **Layer** выбирается свободный элемент, который вы хотите позволить посетителю перемещать.

Вы можете выбрать ограниченное или неограниченное движение выбранного свободного элемента. Это делается с помощью раскрывающегося списка **Movement**. Пункт **Unconstrained** задает неограниченное движение; в этом случае свободный элемент может перемещаться посетителем куда угодно. Пункт **Constrained** списка задает ограниченное движение; в этом случае свободный элемент может двигаться в пределах прямоугольной области, чьи размеры задаются с помощью набора полей ввода, которые в этом случае появятся правее списка **Movement**:

- **Up** — задает вертикальную координату верхней границы области в пикселах;
- **Down** — вертикальную координату нижней границы;
- **Left** — горизонтальную координату левой границы;
- **Right** — горизонтальную координату правой границы.

Вы также можете предусмотреть некую точку на странице, куда будет "стремиться" перемещаемый свободный элемент. Координаты этой точки задаются в пикселах в полях ввода группы **Drop Target**: **Left** (горизонтальная) и **Top** (вертикальная). Нажатие кнопки **Get Current Position** позволит вам поместить в эти поля ввода текущие координаты свободного элемента. В поле ввода **Snap if Within ... Pixels of Drop Target** задается расстояние в пикселах

до вышеуказанной точки, при достижении которого перемещаемый элемент сам "приклеивается" к ней.

Выше мы рассмотрели элементы управления, находящиеся на вкладке **Basic** диалогового окна **Drag Layer**, показанного на рис. 13.12. Если вы делаете простейшую головоломку, их вам будет достаточно. Если же хотите большего, переключитесь на вкладку **Advanced** (рис. 13.13).

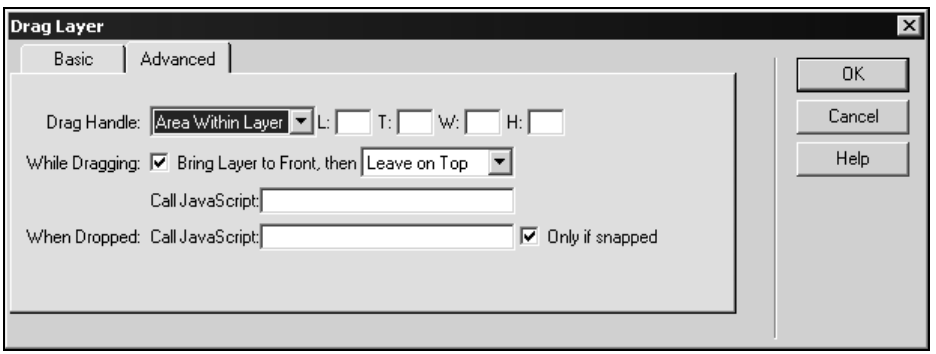


Рис. 13.13. Диалоговое окно **Drag Layer** (вкладка **Advanced**)

По умолчанию, чтобы перетащить свободный элемент на другое место, посетитель должен "ухватиться" мышью за любое его место. С помощью раскрывающегося списка **Drag Handle** и набора полей ввода правее его вы можете задать ограниченную прямоугольную область внутри этого элемента, за которую его можно будет таскать. Для этого выберите в данном списке пункт **Area Within Layer** и введите в поля ввода соответствующие координаты:

- в поле ввода **L** — горизонтальную координату левой границы области в пикселах;
- **T** — вертикальную координату верхней границы;
- **W** — ширину области;
- **H** — высоту области.

Чтобы задать поведение по умолчанию, выберите пункт **Entire Layer** раскрывающегося списка **Drag Handle**.

С помощью флажка **While Dragging** и раскрывающегося списка **Bring Layer to Front, then** вы можете задать поведение свободного элемента при перетаскивании. Если включен флажок **While Dragging**, перетаскиваемый элемент будет находиться над всеми остальными свободными элементами, имеющимися на странице. При этом если в раскрывающемся списке **Bring Layer to Front, then** выбран пункт **Leave on Top**, то этот элемент так и останется "наверху" после отпускания, а если выбран пункт **Restore z-index**, то он будет помещен на ту же позицию в порядке перекрытия, на которой находился до начала перетаскивания.

В поле ввода **Call JavaScript** вы можете ввести строку JavaScript-кода, например вызов написанной ранее функции, которая будет вызываться периодически во время перетаскивания элемента по странице. Этот код может, например, показывать координаты элемента в строке статуса окна Web-обозревателя.

В поле ввода **When Dropped: Call JavaScript** вы можете ввести строку JavaScript-кода, которая будет вызвана после отпускания элемента. При этом если включен флажок **Only if snapped**, этот код будет вызван только тогда, когда перетаскиваемый элемент "приклеится" к конечной точке, чьи координаты были заданы на вкладке **Basic** диалогового окна **Drag Layer**.

Задав нужные параметры, нажмите кнопку **OK**.

Переход на заданный кадр анимации *(Go To Timeline Frame)*

При создании анимации на Web-страницах иногда бывает очень полезно "перескочить" на заданный кадр анимационной дорожки. Например, вы можете создать гиперссылку или кнопку, позволяющую посетителю сайта "перемотать" анимацию в начало. Или ваша анимация может состоять из нескольких фрагментов, и вы хотите дать посетителю возможность просмотреть каждый из этих фрагментов отдельно, для чего также предусмотрели несколько кнопок. Да и мало ли может быть случаев, когда такое может понадобиться!

Для таких случаев Dreamweaver предусматривает поведение `Go To Timeline Frame`. Выберите одноименный пункт в подменю **Timeline** меню поведений. На экране появится диалоговое окно **Go To Timeline Frame** (рис. 13.14).

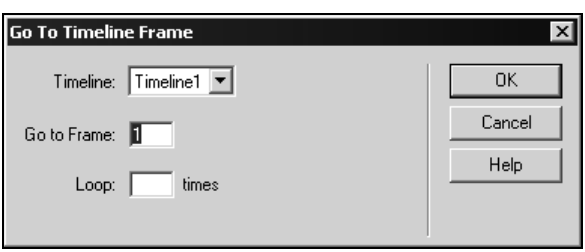


Рис. 13.14. Диалоговое окно **Go To Timeline Frame**

Прежде чем задать кадр анимации, нужно выбрать саму анимацию. Это делается в раскрывающемся списке **Timeline**. Номер же кадра вводится в поле ввода **Go to Frame**. После этого остается только нажать кнопку **OK**.

Вы, наверно, удивились, почему мы не рассмотрели поле ввода **Loop ... times**. Потерпите, о нем еще будет рассказано, когда мы будем рассматривать специальные случаи создания поведений. Это будет ближе к концу главы.

Переход на другую Web-страницу *(Go to URL)*

Иногда бывает нужно загрузить в текущем окне Web-обозревателя другую страницу, не дожидаясь, пока это сделает посетитель. Для таких случаев Dreamweaver предлагает поведение `Go to URL`. Выберите в меню поведений соответствующий пункт; на экране появится диалоговое окно **Go To URL**, показанное на рис. 13.15.

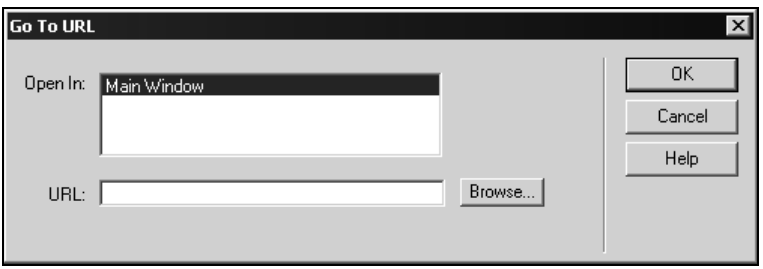


Рис. 13.15. Диалоговое окно **Go To URL**

Сам интернет-адрес задается в поле ввода **URL**. Вы также можете нажать кнопку **Browse** и выбрать нужную страницу в диалоговом окне **Select File**.

В списке **Open In** задается фрейм, в котором будет открыта новая страница. Если ваша страница не представляет собой набор фреймов, в этом списке будет присутствовать единственный пункт **Main Window** (все окно), как в нашем случае.

Задав нужные параметры, нажмите кнопку **OK**.

Скрытие меню гиперссылок *(Hide Pop-Up Menu)*

Это поведение скрывает выведенное ранее на экран меню гиперссылок (о создании меню гиперссылок *см. ниже*). Как правило, Dreamweaver создает его сам, но иногда вам придется делать это самим.

После выбора в меню поведений пункта **Hide Pop-Up Menu** на экране появится небольшое окно-предупреждение. Закройте его, нажав кнопку **OK**. На этом создание поведения `Hide Pop-Up Menu` закончено. Никаких параметров оно не имеет.

Нажатие кнопки **Cancel** вышеупомянутого окна-предупреждения позволит вам отказаться от создания поведения `Hide Pop-Up Menu`.

Открытие нового окна Web-обозревателя *(Open Browser Window)*

Это поведение аналогично поведению `Go to URL` за тем исключением, что оно открывает новую страницу в новом окне Web-обозревателя. При этом

оно позволяет задать различные параметры нового окна: размеры, наличие инструментария и строки статуса и т. п.

Выберите в меню поведений пункт **Open Browser Window**. На экране появится диалоговое окно **Open Browser Window**, показанное на рис. 13.16.

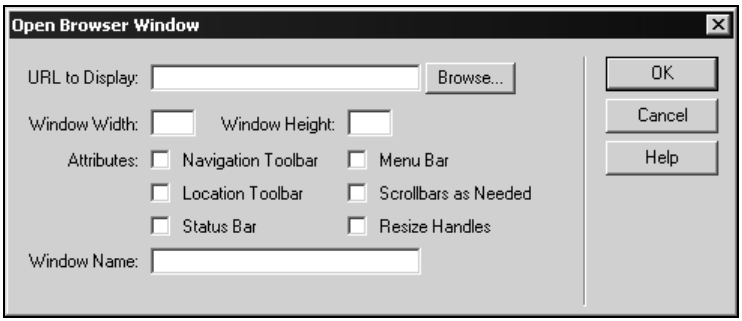


Рис. 13.16. Диалоговое окно **Open Browser Window**

В поле ввода **URL to Display** задается интернет-адрес страницы, которая будет показана в новом окне. Вы можете нажать кнопку **Browse** и выбрать нужную страницу в диалоговом окне **Select File**.

В полях ввода **Window Width** и **Window Height** задаются соответственно ширина и высота нового окна. Заметьте, что координатами окна управлять вы не можете.

С помощью группы флажков **Attributes** задаются дополнительные параметры нового окна:

- **Navigation Toolbar** — включает или отключает наличие у нового окна главного инструментария с кнопками **Вперед** (Forward), **Назад** (Back), **Остановить** (Stop) и **Обновить** (Reload);
- **Location Toolbar** — включает или отключает наличие у нового окна инструментария с полем ввода интернет-адреса;
- **Status Bar** — включает или отключает наличие у нового окна строки статуса;
- **Menu Bar** — включает или отключает наличие у нового окна системного меню;
- **Scrollbars as Needed** — разрешает или запрещает появление у нового окна полос прокрутки, если его содержимое в нем не помещается;
- **Resize Handles** — разрешает или запрещает пользователю изменять размеры нового окна.

Имейте в виду, что если вы не задали размеров окна, то новое окно будет иметь случайные размеры и полный набор параметров, перечисленных выше (т. е. оба инструментария, строку статуса и т. д.). Если же вы зададите

размеры, то новое окно, наоборот, не будет иметь ни одного из этих атрибутов, поэтому вам самим придется задавать нужные параметры, включая соответствующие флажки группы **Attributes**.

В поле ввода **Window Name** задается имя создаваемого окна. Впоследствии вы можете использовать это имя, например, при создании поведения `Go to URL` для вывода в нем новой Web-страницы.

Кнопка **OK** диалогового окна сохраняет сделанные вами установки, а кнопка **Cancel** — отменяет их.

Проигрывание аудиоклипа *(Play Sound)*

В свете последних тенденций к наполнению Интернета мультимедийной информацией Dreamweaver предоставляет вам возможность в ответ на какое-либо событие проиграть аудиоклип с помощью поведения `Play Sound`. Для этого выберите пункт **Play Sound** меню поведений. После этого на экране появится диалоговое окно **Play Sound** (рис. 13.17).

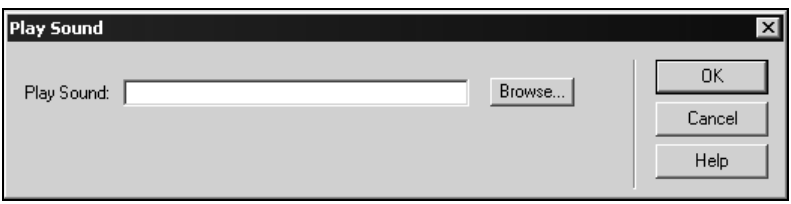


Рис. 13.17. Диалоговое окно **Play Sound**

Это диалоговое окно содержит одно-единственное поле ввода **Play Sound**, предназначенное для ввода имени аудиофайла, который будет проигран. Вы также можете нажать кнопку **Browse** и выбрать нужный файл в диалоговом окне **Select File**. После этого нажмите кнопку **OK**.

Запуск проигрывания анимации *(Play Timeline)*

Конечно же, Dreamweaver предоставляет специальные поведения для управления проигрыванием анимации в ответ на какое-либо событие. Это реализуется с помощью поведения `Play Timeline`. Одноименный пункт для создания этого поведения находится в подменю **Timeline** меню поведений.

Диалоговое окно **Play Timeline** показано на рис. 13.5. Оно содержит один-единственный раскрывающийся список, называющийся **Play Timeline**. В нем выбирается анимация, которую нужно проиграть, после чего нажимается кнопка **OK**. Как видите, все очень просто.

Вывод предупреждения *(Popup Message)*

Очень часто для сообщения пользователю о чем-либо (например, необходимости ввести данные в поле ввода на странице) используются окна-пре-

дупреждения. Они представляют собой обычное стандартное предупреждение Windows: небольшое окно с текстом предупреждения, изображением восклицательного знака и кнопкой **OK**. Для вывода такого предупреждения вы можете воспользоваться поведением `Popup Message`. Выберите одноименный пункт меню поведений, после чего на экране появится диалоговое окно **Popup Message** (рис. 13.18).

Рис. 13.18. Диалоговое окно **Popup Message**

Введите текст предупреждения в область редактирования **Message** и нажмите кнопку **OK**.

Вы можете использовать в тексте предупреждения любой JavaScript-код, заключив его в фигурные скобки. Например, чтобы вывести посетителю сайта текущую дату, вы можете использовать такой код:

```
Сегодня {new Date()}
```

Если же вам понадобится для каких-то целей ввести в текст фигурную скобку, предварите ее обратной косой чертой (`/{` или `/}`).

Предварительная загрузка графических изображений *(Preload Images)*

Как вы помните из материала предыдущей главы, на Web-страницах часто используются анимированные картинки, представляющие собой обычный набор графических изображений, быстро сменяющих друг друга. Такая анимация в среде Dreamweaver создается очень просто, да и при "ручной" работе также не вызывает особой сложности у достаточно опытного Web-программиста. Проблема в другом: при загрузке страницы из Интернета загружается только одно, первое изображение, а остальные Web-обозревателю приходится подгружать в процессе проигрывания анимации, создавая значительные задержки при проигрывании анимации в первый раз.

Решить эту проблему позволяет предварительная загрузка нужного набора изображений. Изображения загружаются с помощью специального сценария, но нигде не отображаются, а просто сохраняются в кэше Web-обозревателя. При проигрывании такой анимации Web-обозреватель берет

нужные графические файлы из своего кэша, вместо того, чтобы загружать их из Сети.

Для выполнения предварительной загрузки графических изображений могут применяться два подхода:

1. *Статический*, самый очевидный. На странице создается соответствующее количество графических изображений (тег `<IMG>`) размером 1×1 пиксел (или даже 0×0, если это сработает). В качестве значения атрибута `SRC` указывается имя одного из файлов, содержащих разные кадры анимации. При загрузке страницы Web-обозреватель тотчас загружает все нужные для анимации файлы и сохраняет их в своем кэше. Этот способ очень прост и надежен, если анимация содержит небольшое количество кадров. Однако, если кадров в анимации достаточно много, Web-обозреватель займет для сохранения этих изображений в памяти слишком много системных ресурсов, что не всегда приемлемо.
2. *Динамический.* Используется Web-сценарий, загружающий все эти файлы. При этом графические изображения не хранятся в памяти и, таким образом, не занимают системные ресурсы, стало быть, анимация может содержать сколько угодно кадров. Однако этот подход несколько сложнее из-за необходимости создания сценария.

И все же для предварительной загрузки изображений рекомендуется использовать второй, динамический подход. И Dreamweaver вам в этом поможет, предоставив поведение `Preload Images`. Очевидно, что это поведение привязывается к событию `onLoad` тега `<BODY>`. Чтобы создать его, выберите одноименный пункт в меню поведений. После этого на экране появится диалоговое окно **Preload Images**, показанное на рис. 13.19.

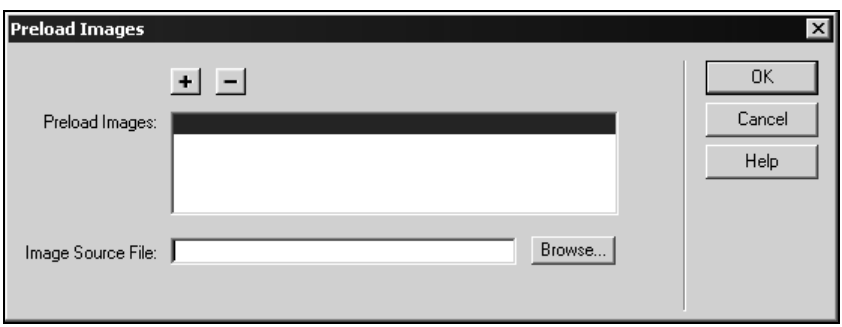


Рис. 13.19. Диалоговое окно **Preload Images**

Файлы, которые должны быть предварительно загружены, перечислены в списке **Preload Images**. Чтобы добавить файл в этот список, введите его имя в поле ввода **Image Source File** и нажмите кнопку со знаком "плюс". Вы также можете нажать кнопку **Browse** и выбрать нужный файл в диалоговом

окне **Select File**. Чтобы изменить какой-либо файл, выберите его в списке, измените его имя в поле ввода **Image Source File** и после этого обязательно переключитесь на другой элемент управления, лучше всего — на сам список **Preload Images**. Чтобы удалить ненужный файл, выберите его в списке и нажмите кнопку со знаком "минус".

Теоретически, с помощью этого поведения можно предварительно загружать не только графические, но и любые другие файлы. В частности, это могут быть аудиофайлы, которые будут впоследствии проигрываться в ответ на действия пользователя, или видеоклипы Flash. Хотя в документации по Dreamweaver этого почему-то не написано.

После задания списка файлов, которые должны быть предварительно загружены, нажмите кнопку **OK**. Кнопка **Cancel** позволит вам отказаться от сделанных изменений.

Изменение изображения-элемента полосы навигации *(Set Nav Bar Image)*

О полосе навигации речь шла в *главе 7*, посвященной фреймам и фреймовому дизайну. Как вы помните, при создании полосы навигации Dreamweaver создает также множество сценариев, обеспечивающих смену изображений каждого элемента в ответ на различные события. Эти сценарии и представляют собой поведение `Set Nav Bar Image`.

Как правило, вам самим не нужно создавать это поведение — это сделает за вас Dreamweaver. Если же это вам понадобится, выберите пункт **Set Nav Bar Image** в меню поведений. На экране появится диалоговое окно **Set Nav Bar Image** (рис. 13.20).

Как видите, содержимое вкладки **Basic** этого окна почти совпадает с содержимым диалогового окна **Insert Navigation Bar**, показанного на рис. 7.23. Элементы управления этой вкладки позволяют настроить выделенный вами элемент полосы навигации: задать для него изображения, "альтернативный" текст, интернет-адрес гиперссылки и др. Все это вам уже знакомо по *главе 7*.

А вот содержимое вкладки **Advanced** данного окна позволяет вам задать кое-какие дополнительные настройки, затрагивающие другие элементы полосы навигации. Эта вкладка показана на рис. 13.21. В частности, с ее помощью вы можете задать изменение изображения другого элемента полосы, если посетитель щелкнет выделенный вами элемент.

В раскрывающемся списке **When element *<название выделенного элемента>* is displaying** выбирается состояние, в котором находится выделенный вами элемент полосы навигации. Если выбран пункт **Over Image or Over While Down Image**, то элемент должен находиться в состоянии, когда посетитель поместил над ним курсор мыши. Если выбран пункт **Down Image**, то посетитель должен щелкнуть по этому элементу ("нажатое" состояние).

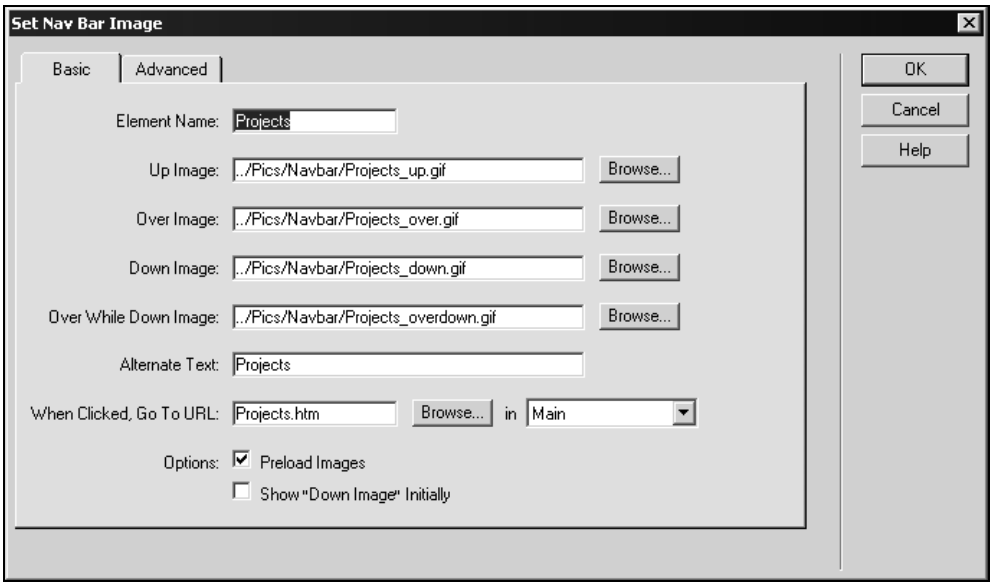


Рис. 13.20. Диалоговое окно **Set Nav Bar Image** (вкладка **Basic**)

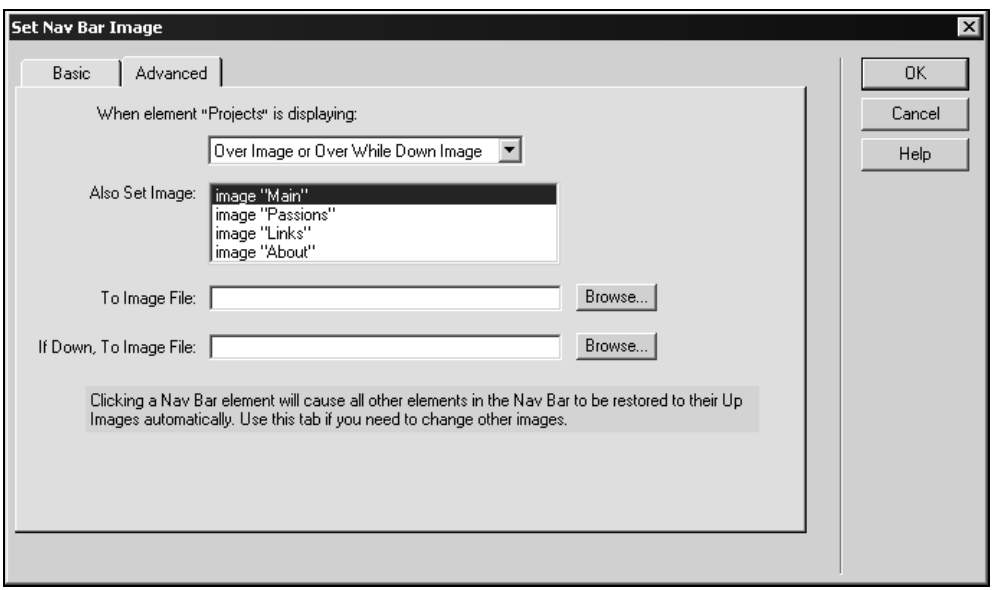


Рис. 13.21. Диалоговое окно **Set Nav Bar Image** (вкладка **Advanced**)

В списке **Also Set Image** выбирается элемент полосы навигации, вид которого вы хотите изменить.

В поле ввода **To Image File** вводится имя файла изображения для выбранного в списке **Also Set Image** элемента. Это изображение будет отображаться, если выбранный в списке элемент находится в "ненажатом" состоянии. Вы также можете щелкнуть кнопку **Browse** и выбрать нужный файл в диалоговом окне **Select File**.

Если вы выбрали пункт **Over Image or Over While Down Image** в раскрывающемся списке **When element <*название выделенного элемента*> is displaying**, то становится доступным также поле ввода **If Down**, **To Image File**. В нем задается имя файла изображения, которое будет отображаться, если выбранный в списке пункт находится в "нажатом" состоянии. Также здесь доступна кнопка **Browse**.

Вы можете задать особое поведение сразу для нескольких элементов полосы навигации. Только не забудьте нажать кнопку **OK**.

Задание нового содержимого фрейма *(Set Text of Frame)*

Dreamweaver позволяет вам поместить новое содержимое в любой из фреймов текущего набора. Для этого выберите пункт **Set Text of Frame** в подменю **Set Text** меню поведений. На экране появится диалоговое окно **Set Text of Frame**, показанное на рис. 13.22.

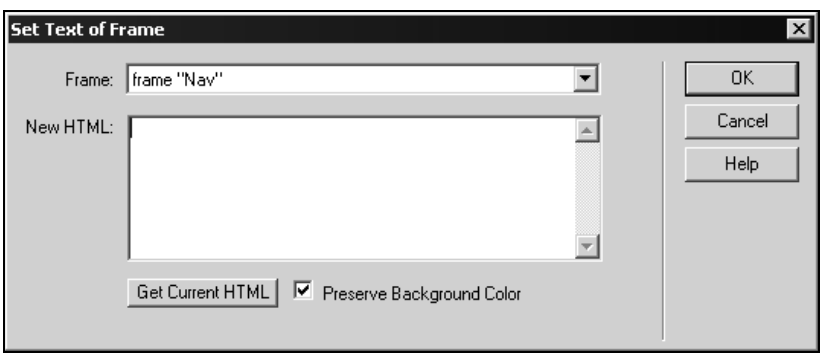


Рис. 13.22. Диалоговое окно **Set Text of Frame**

В раскрывающемся списке **Frame** выбирается фрейм, в который вы хотите поместить новое содержимое. Само содержимое в виде HTML-кода вводится в область редактирования **New HTML**. Вы также можете нажать кнопку **Get Current HTML**, чтобы скопировать в эту область редактирования текущее содержимое фрейма, после чего внести необходимые изменения. Если вы включите флажок **Preserve Background Color**, Web-сценарий, обновляющий содержимое фрейма, сохранит цветовые настройки текста и фона;

в противном случае эти настройки пропадут. Задав нужные настройки, нажмите кнопку **OK**.

Вы можете использовать в HTML-коде нового содержимого фрейма любой JavaScript-код, заключив его в фигурные скобки. Например, чтобы вывести посетителю сайта текущую дату, вы можете использовать такой код:

```
Сегодня {new Date()}
```

Если же вам понадобится ввести в текст фигурную скобку, предварите ее обратной косой чертой (`/{` или `/}`).

Задание нового содержимого свободно позиционируемого элемента *(Set Text of Layer)*

Dreamweaver позволяет вам поместить новое содержимое в любой из свободно позиционируемых элементов, имеющихся на странице. Для этого выберите пункт **Set Text of Layer** в подменю **Set Text** меню поведений. На экране появится диалоговое окно **Set Text of Layer**, показанное на рис. 13.23.

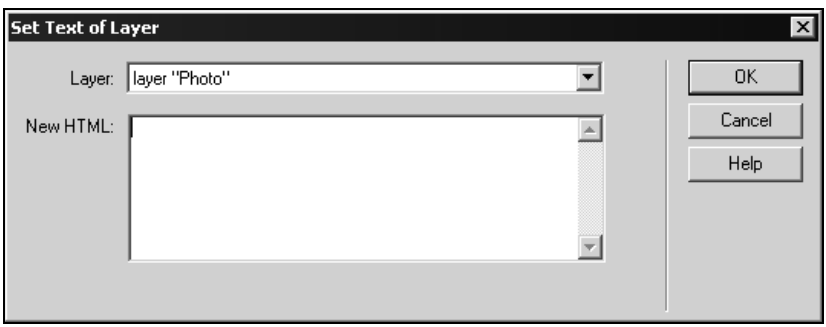


Рис. 13.23. Диалоговое окно **Set Text of Layer**

В раскрывающемся списке **Layer** выбирается свободный элемент, в который вы хотите поместить новое содержимое. Само содержимое в виде HTML-кода вводится в область редактирования **New HTML**. Задав нужные настройки, нажмите кнопку **OK**.

Вы можете использовать в HTML-коде нового содержимого свободного элемента любой JavaScript-код, заключив его в фигурные скобки. Например, чтобы вывести посетителю сайта текущую дату, вы можете использовать такой код:

```
Сегодня {new Date()}
```

Если же вам понадобится ввести в текст фигурную скобку, предварите ее обратной косой чертой (`/{` или `/}`).

Вывод текста в строке статуса окна Web-обозревателя *(Set Text of Status Bar)*

Dreamweaver также позволяет вам вывести любой текст в строке статуса текущего окна Web-обозревателя. Это может быть использовано, например, для показа краткого описания гиперссылки, над которой находится курсор мыши, или просто краткого описания текущей страницы сайта. (Иногда в статусной строке выводится текущее время, "ползущий" справа налево либо "вырастающий" слева направо текст или другие чудеса.) Выберите пункт **Set Text of Status Bar** в подменю **Set Text** меню поведений. На экране появится диалоговое окно **Set Text of Status Bar**, показанное на рис. 13.24.

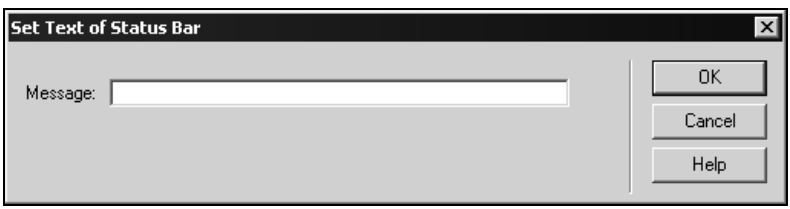


Рис. 13.24. Диалоговое окно **Set Text of Status Bar**

Это диалоговое окно содержит единственное поле ввода **Message**, в котором вводится текст, отображаемый в строке статуса. Введя его, нажмите кнопку **OK**.

Вы можете использовать в тексте, отображаемом в строке статуса, любой JavaScript-код, заключив его в фигурные скобки. Например, чтобы вывести посетителю сайта текущую дату, вы можете использовать такой код:

```
Сегодня {new Date()}
```

Если же вам понадобится ввести в текст фигурную скобку, предварите ее обратной косой чертой (`/{` или `/}`).

Чтобы отобразить в строке статуса текст по умолчанию (он формируется самим Web-обозревателем — обычно надпись "Готово" или адрес гиперссылки), задайте в диалоговом окне следующий текст:

```
{window.defaultStatus}
```

Свойство `defaultStatus` объекта `window` как раз и возвращает этот текст по умолчанию.

Внимание!

Учтите, что помещать посторонний текст в строку статуса окна Web-обозревателя сейчас считается дурным тоном. Поэтому используйте эту возможность только тогда, когда без нее действительно не обойтись.

Вывод на экран меню гиперссылок *(Show Pop-Up Menu)*

В последнее время в моду вошли *меню гиперссылок*, раскрывающиеся после наведения курсора мыши на какой-либо элемент Web-страницы или щелчка по нему. Родоначальником этой моды, вероятно, были Web-дизайнеры, разрабатывавшие сайт фирмы Microsoft, после чего меню "разбежались" по другим сайтам и страницам. Разумеется, разработчики Dreamweaver не могли пройти мимо этого и ввели в свое детище возможность создания такого меню с помощью поведения `Show Pop-Up Menu` и привязки его к любому элементу страницы.

Внимание!

Перед тем как создать меню гиперссылок, дайте элементу страницы, к которому оно привязывается, уникальное имя.

Выберите в меню поведений пункт **Show Pop-Up Menu**. На экране появится диалоговое окно **Show Pop-Up Menu** (рис. 13.25).

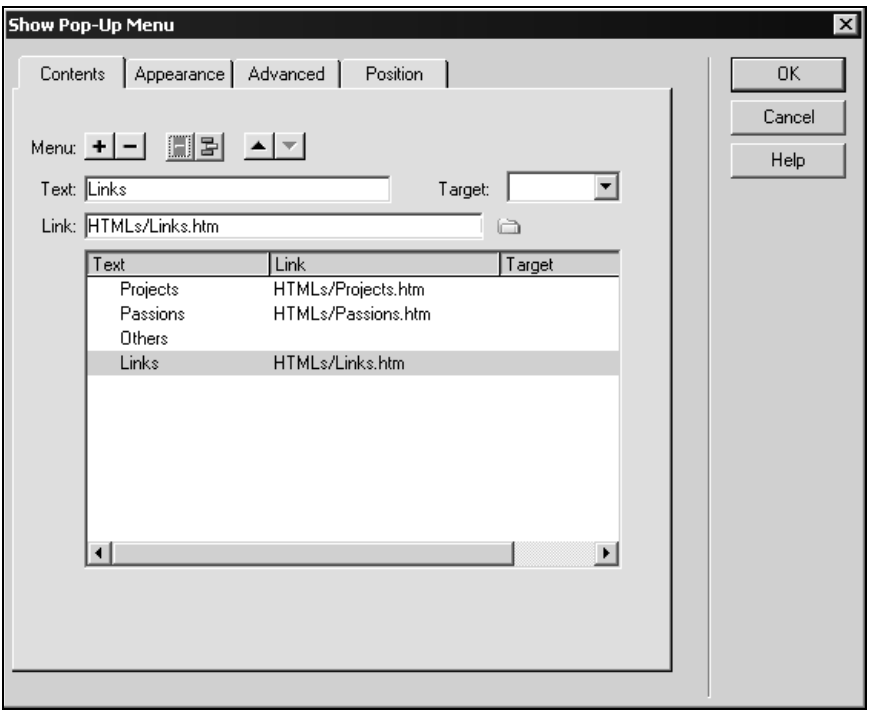


Рис. 13.25. Диалоговое окно **Show Pop-Up Menu** (вкладка **Contents**)

Чтобы создать новый пункт меню гиперссылок, введите его текст в поле ввода **Text**. В поле ввода **Link** вводится интернет-адрес соответствующей

пункту гиперссылки; также вы можете щелкнуть расположенный правее этого поля ввода значок папки и выбрать нужный файл в диалоговом окне **Select File**. А с помощью раскрывающегося списка **Target** — задать цель гиперссылки.

Внимание!

Dreamweaver и в этом случае верен своему кредо!.. Если вы введете русский текст в поле ввода **Text**, он будет искажен до неузнаваемости. Поэтому либо задавайте для пунктов меню гиперссылок только английский текст, либо правьте сам JavaScript-код вручную.

Вы можете не вводить данные в поле ввода **Link**. В этом случае пункт меню не будет гиперссылкой и, соответственно, не будет реагировать на щелчки мышью. Таким образом вы можете создавать, например, разделители.

Все созданные вами к данному времени пункты меню гиперссылок отображаются в списке, занимающем большую часть вкладки **Contents**. Этот список состоит из трех колонок: **Text** (текст пункта меню), **Link** (интернет-адрес гиперссылки) и **Target** (цель гиперссылки). Вы можете выбрать любой пункт в этом списке и выполнить над ним различные манипуляции.

Чтобы создать новый пункт, нажмите кнопку со знаком "плюс", расположенную над списком пунктов, и задайте в полях ввода **Text**, **Link** и раскрывающемся списке **Target** нужные данные.

Чтобы изменить какой-либо пункт меню, выберите его в списке пунктов и измените его данные, пользуясь перечисленными выше элементами управления.

Чтобы удалить ненужный пункт, выберите его в списке пунктов и нажмите кнопку со знаком "минус", расположенную над этим списком.

Кнопки ↑ и ↓ позволят переместить выбранный вами пункт на позицию вверх или вниз соответственно.

Вы также можете создавать вложенные меню. Для этого создайте пункт основного меню, при щелчке на котором будет открываться подменю, причем не вводите ничего в поле ввода **Link**. Далее создайте первый пункт подменю и поместите его точно перед созданным ранее пунктом. Теперь вам останется нажать кнопку **Indent Item** (рис. 13.26).

Рис. 13.26. Кнопка **Indent Item** вкладки **Contents** диалогового окна **Show Pop-Up Menu**

Результат показан на рис. 13.27. В данном случае пункт **Links** станет единственным пунктом подменю, раскрывающемся при щелчке на пункте **Others**.

Как видите, пункт **Links** отображается с отступом; это говорит о том, что он принадлежит подменю.

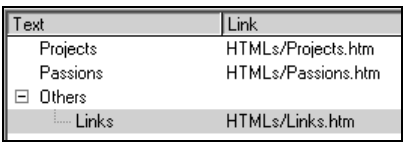


Рис. 13.27. Подменю в меню гиперссылок

Чтобы вернуть пункт из подменю в основное меню или подменю более высокого уровня, нажмите кнопку **Outdent Item** (рис. 13.28).

Рис. 13.28. Кнопка **Outdent Item** вкладки **Contents** диалогового окна **Show Pop-Up Menu**

Вкладка **Appearance** диалогового окна **Show Pop-Up Menu** (рис. 13.29) позволит вам настроить внешний вид меню гиперссылок.

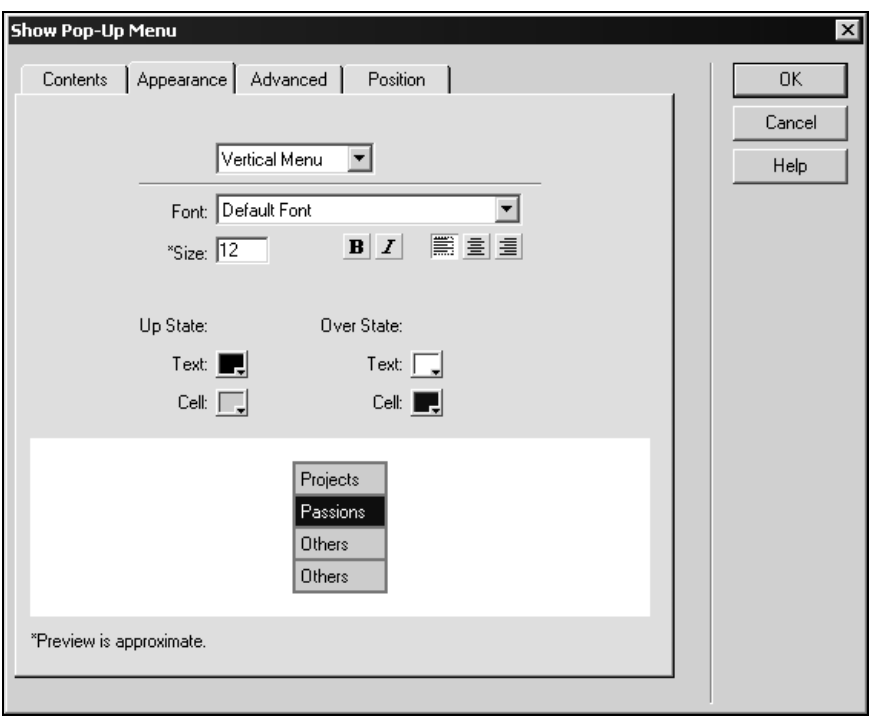


Рис. 13.29. Диалоговое окно **Show Pop-Up Menu** (вкладка **Appearance**)

С помощью раскрывающегося списка, расположенного выше всех остальных элементов управления, вы можете выбрать расположение меню гиперссылок: вертикальное (пункт **Vertical Menu**) или горизонтальное (**Horizontal Menu**).

В раскрывающемся списке **Font** выбирается шрифт, которым набран текст пунктов меню. По умолчанию выбран пункт **Default Font**, обозначающий шрифт, заданный для выбранного вами в окне документа элемента страницы.

В поле ввода **Size** задается размер шрифта, которым набран текст пунктов меню.

С помощью кнопок-выключателей **B** и ***I*** вы можете сделать шрифт пунктов меню соответственно жирным или курсивным.

Правее кнопок **B** и ***I*** находится набор из трех кнопок-переключателей, задающих выравнивание текста пунктов меню. Они задают соответственно левое выравнивание, центрирование и правое выравнивание в порядке слева направо.

Группа селекторов цвета **Up State** задает цвета пунктов меню в "ненажатом" состоянии. Селектор цвета **Text** задает цвет текста, а селектор цвета **Cell** — цвет фона (фактически — цвет фона ячейки таблицы, которая и создает меню гиперссылок). Аналогично, группа селекторов цвета **Down State** задает цвета пункта меню, над которым посетитель поместил курсор мыши.

Результаты применения заданных вами параметров вы можете увидеть в расположенной в нижней части диалогового окна панели предварительного просмотра.

Вкладка **Advanced** диалогового окна **Show Pop-Up Menu** (рис. 13.30) позволит вам задать некоторые дополнительные параметры таблицы, создающей меню гиперссылок.

Поля ввода и раскрывающиеся списки **Cell Width** и **Cell Height** задают соответственно ширину и высоту ячеек этой таблицы. Если вы выберете в раскрывающихся списках пункт **Automatic**, то соответствующий размер будет устанавливаться самим Web-обозревателем. Чтобы самим задать нужный размер, выберите в соответствующем раскрывающемся списке пункт **Pixels** и введите этот размер в пикселах в поле ввода, расположенное левее.

Поля ввода **Cell Padding** и **Cell Spacing** задают соответствующие параметры таблицы.

В поле ввода **Text Indent** задается величина отступа красной строки в пикселах.

В поле ввода **Menu Delay** вводится задержка открытия меню и его подменю в миллисекундах.

Включите флажок **Show Borders**, если хотите, чтобы границы меню и подменю были видимы.

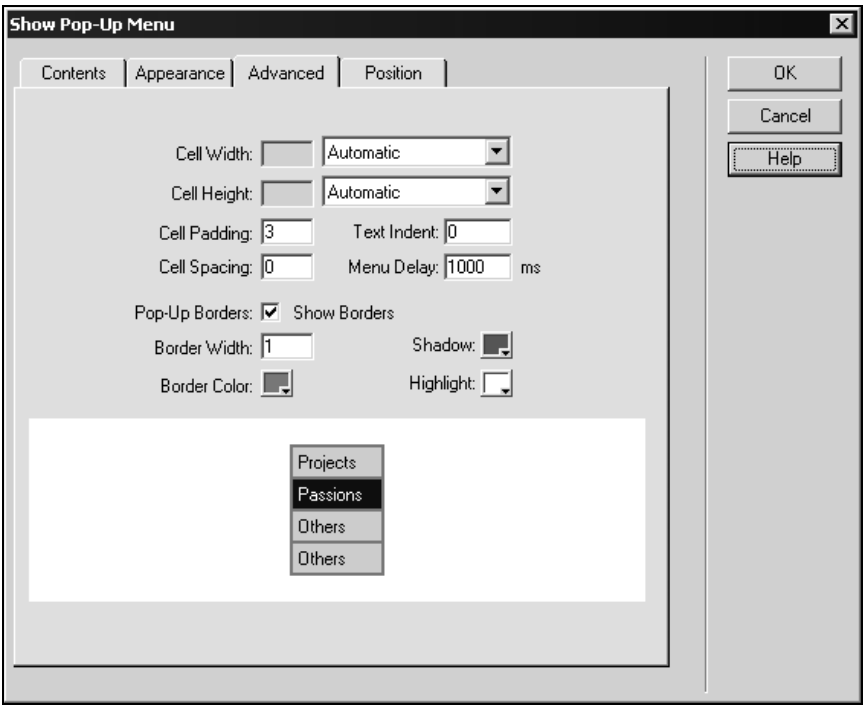


Рис. 13.30. Диалоговое окно **Show Pop-Up Menu** (вкладка **Advanced**)

В поле ввода **Border Width** вводится толщина границы меню и подменю в пикселах.

Селектор цвета **Border Color** задает цвет границы меню и подменю. Также вы можете задать цвет "затемненной" стороны границы в селекторе цвета **Shadow**, а цвет "освещенной" стороны — в селекторе цвета **Highlight**.

Результаты применения заданных вами параметров вы также можете увидеть в расположенной в нижней части диалогового окна панели предварительного просмотра.

Вкладка **Position** диалогового окна **Show Pop-Up Menu** (рис. 13.31) позволит вам задать местонахождение меню гиперссылок относительно элемента страницы, к которому оно привязано.

Набор из четырех больших кнопок-переключателей **Menu Position** позволит вам задать собственно местоположение меню гиперссылок. Эти кнопки заставляют меню появляться соответственно в правом нижнем углу, ниже, выше и в правом верхнем углу элемента, к которому оно привязано. (Порядок перечисления кнопок — слева направо.)

Если вас не устраивает ни одно из четырех предопределенных местоположений меню, вы можете задать свое собственное. Для этого введите в поля

ввода **X** и **Y** соответственно горизонтальную и вертикальную координаты верхнего левого угла меню.

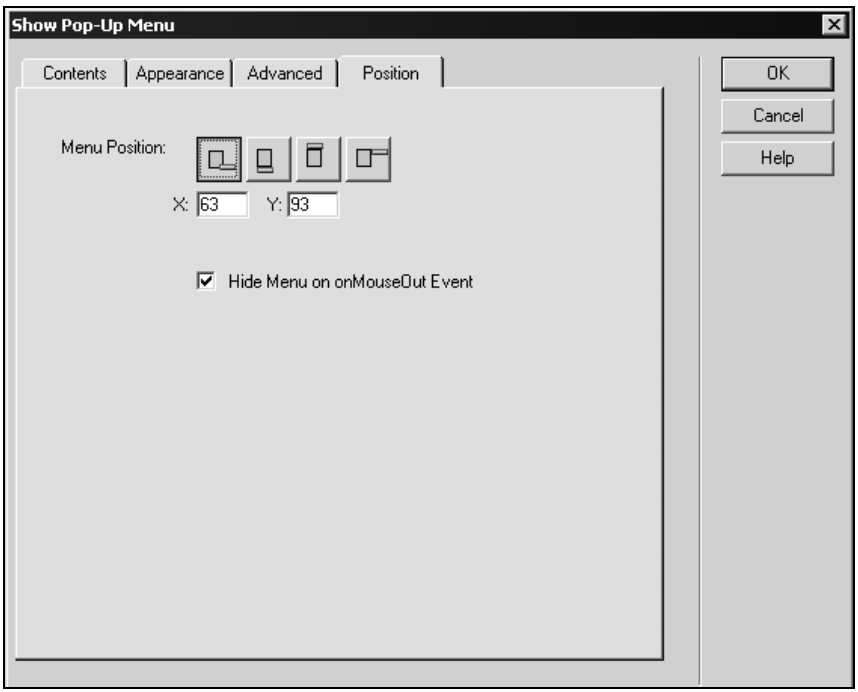


Рис. 13.31. Диалоговое окно **Show Pop-Up Menu** (вкладка **Position**)

По умолчанию меню гиперссылок убирается с экрана сразу же после того, как посетитель уберет курсор мыши с элемента страницы, к которому оно привязано. Если вы хотите, чтобы оно оставалось на экране, отключите флажок **Hide Menu on onMouseOut Event**.

После задания всех параметров меню гиперссылок не забудьте нажать кнопку **OK**.

Внимание!

При создании на странице меню гиперссылок Dreamweaver помещает в корневую папку сайта файл mm_menu.js, содержащий необходимые сценарии, и графическое изображение arrows.gif. Не удаляйте их! И не забудьте поместить их на Web-сервер.

Показ и скрытие свободно позиционируемых элементов *(Show-Hide Layers)*

Показ и скрытие свободно позиционируемых элементов страницы можно использовать в разных целях. Например, показывать различные подсказки

при наведении курсора мыши на гиперссылки или графические изображения. Также можно на их основе создавать начальные заставки, показываемые посетителю сайта на самой первой его странице. (Почти всегда, однако, для этого используются фильмы форматов Flash, RealMedia или GIF. Но эту традицию стоит нарушить.) И, разумеется, появляющиеся и пропадающие в ответ на действия пользователя элементы страницы незаменимы при создании игр, обучающих пособий и программ, созданных на основе Web-страниц.

Вы уже знаете, как создать на странице свободный элемент, как задать его размеры и сделать невидимым (до поры до времени). Поговорим теперь, как заставить его появляться и исчезать в ответ на события. А для этого Dreamweaver предоставляет поведение `Show-Hide Layers`.

Выберите пункт **Show-Hide Layers** в меню поведений. На экране появится диалоговое окно **Show-Hide Layers** (рис. 13.32).

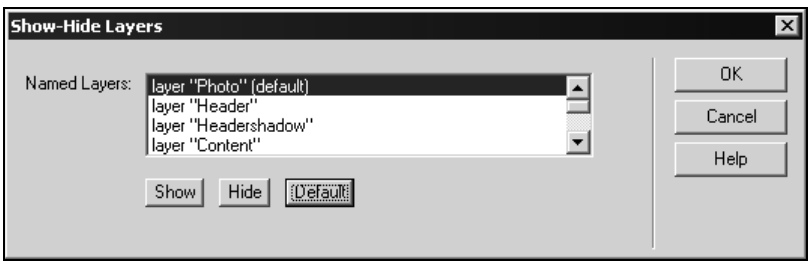


Рис. 13.32. Диалоговое окно **Show-Hide Layers**

Все имеющиеся на странице свободно позиционируемые элементы перечислены в списке **Named Layers**. Вы можете выбрать в этом списке любой из них и задать для него выполняемое действие: показ или скрытие.

Для задания действия же служат три кнопки, расположенные под списком:

- **Show** — показывает свободный элемент (при этом в соответствующей строке списка правее его имени появляется надпись "(show)");
- **Hide** — скрывает свободный элемент (надпись "(hide)");
- **Default** — восстанавливает видимость свободного элемента, заданную при его создании (надпись "(default)").

Выберите необходимый свободный элемент в списке и нажмите нужную кнопку. Чтобы отменить ошибочно выбранное действие, нажмите ту же кнопку еще раз. После этого нажмите кнопку **OK**, чтобы сохранить сделанные установки.

Останов проигрывания анимации *(Stop Timeline)*

Поведение `Stop Timeline` позволяет вам остановить проигрывание анимации, запущенной ранее поведением `Play Timeline`. Для его создания вос-

пользуйтесь пунктом **Stop Timeline** подменю **Timeline** меню поведений. На экране появится диалоговое окно **Stop Timeline** (рис. 13.33).

Рис. 13.33. Диалоговое окно **Stop Timeline**

Это диалоговое окно содержит только раскрывающийся список **Stop Timeline**. В этом списке выбирается анимация, которую нужно остановить. Пункт **** ALL TIMELINES **** позволяет остановить проигрывание всех анимаций, созданных на данной странице. После выбора анимации не забудьте нажать кнопку **OK**.

Изменение графического изображения *(Swap Image)*

Dreamweaver предоставляет вам возможность изменить рисунок, отображающийся в любом имеющемся на странице элементе графического изображения. Для этого используется поведение `Swap Image`; специальный Web-сценарий меняет значение атрибута `SRC` тега `<IMG>`. Это будет полезно во многих случаях.

Выберите пункт **Swap Image** меню поведений. На экране появится диалоговое окно **Swap Image**, показанное на рис. 13.34.

Рис. 13.34. Диалоговое окно **Swap Image**

Все имеющиеся на странице графические изображения перечислены в списке **Images**. Выберите нужное. После этого введите имя файла нового изо-

бражения в поле ввода **Set Source to**; также вы можете нажать кнопку **Browse** и выбрать нужный файл в появившемся на экране диалоговом окне **Select File**.

Если вы хотите изменить сразу несколько изображений, выполните вышеприведенную последовательность действий для всех нужных элементов. Созданный Dreamweaver Web-сценарий выполнит все эти замены одновременно.

Если включен флажок **Preload Images** (а он включен по умолчанию), Dreamweaver создаст еще один сценарий, выполняющий предварительную загрузку нужных файлов изображений. (Фактически для этого он создаст поведение `Preload Images`.) Если вы отключите этот флажок, то сами должны будете выполнить их предзагрузку, либо мириться с задержками при изменении изображений.

Если вы привязываете поведение `Swap Image` к событию `onMouseOver`, то Dreamweaver автоматически создаст поведение `Swap Image Restore`, восстанавливающее все изначальные изображения, и привязывает его к событию `onMouseOut`. Если вы не хотите, чтобы он так делал, отключите флажок **Restore Images onMouseOut**.

Закончив работу, нажмите кнопку **OK**.

Восстановление всех изначальных изображений *(Swap Image Restore)*

Если вы хотите после вызова поведения `Swap Image` восстановить изображения, отображавшиеся на странице изначально, создайте поведение `Swap Image Restore`. Для этого выберите пункт **Swap Image Restore** меню поведений. На экране появится небольшое окно-предупреждение; нажмите кнопку **OK**, чтобы закрыть его. Нажатие кнопки **Cancel** позволит вам отказаться от создания этого поведения.

Внимание!

Создавать поведение `Swap Image Restore` можно только после создания поведения `Swap Image`.

Написание своих Web-сценариев

Как видите, Dreamweaver предлагает достаточно ограниченный набор поведений и, стало быть, Web-сценариев, которые могут быть созданы на страницах. Конечно, они покрывают 95% нужд начинающего и даже опытного Web-дизайнера, и у вас, возможно, очень и очень нескоро появится потребность писать JavaScript-код самим. И все же...

Но вспомните, что Dreamweaver — гибридный Web-редактор! Он предоставляет не только удобный интерфейс для визуального создания и редактиро-

вания Web-страниц, но и такой же удобный доступ к их исходному HTML-коду. Поэтому вы можете без труда "вписать" туда свои сценарии, реализующие все, что вашей душе угодно. Правда, Dreamweaver в этом случае никак облегчить вашу работу не сможет...

Особые случаи создания поведений

Итак, создание поведений и привязку их к элементам страницы и событиям мы рассмотрели. Но все это были весьма тривиальные случаи. Сейчас же поговорим кое о чем особенном...

Привязка поведений к тексту

Мы знаем, как привязать поведение к элементу страницы, например гиперссылке или графическому изображению. Но что делать, если мы хотим привязать его к фрагменту текста в абзаце?

Если вы готовы пожертвовать совместимостью с Navigator, можете использовать такой прием:

```
<P>Это <SPAN onClick="doSomething();">текст, по которому можно
  ↳щелкать</SPAN>.</P>
. . .
function doSomething() {
  Что-то делаем...
}
```

Как видите, нужный фрагмент текста выделен тегом `<SPAN>` и привязан к этому тегу обработчик события `onClick`. Это работать будет, но только в Internet Explorer. Этот Web-обозреватель предоставляет Web-программисту доступ к любому элементу страницы и, соответственно, может обрабатывать события, происходящие в них.

Это можно сделать и в среде Dreamweaver, воспользовавшись диалоговым окном **Tag Chooser**. Выделите нужный фрагмент текста и выберите пункт **Tag** меню **Insert** или нажмите комбинацию клавиш <Ctrl>+<E>. В левом списке окна **Tag Chooser** разверните ветвь **HTML Tags** и выберите пункт **Formatting and Layout**, далее выберите в правом списке пункт **span** и нажмите кнопку **Insert**. После этого выделенный вами фрагмент текста будет "взят" в тег `<SPAN>`. Вы можете поставить в него текстовый курсор и привязать к нему любое поведение.

Кстати, так можно привязать поведение к любому тегу. Смотрите сюда:

```
<FONT SIZE="+2" onClick="doSomething();">Это текст, по которому можно
  ↳щелкать.</FONT>
```

И в этом случае Dreamweaver придет вам на помощь. Вам останется только поставить текстовый курсор на нужный текст и создать поведение.

Но это будет работать только в Internet Explorer.

У Navigator поддержка Web-программирования значительно ограничена. Он дает доступ к ограниченному набору элементов: гиперссылкам, графическим изображениям, внедренным элементам, фреймам, свободно позиционируемым элементам и еще кое-чему. К обычным абзацам текста и уж тем более элементам `<SPAN>` нашему брату Web-программисту путь заказан.

Что делать?

Есть один способ. Если уж Navigator требует гиперссылки, дадим ему гиперссылку.

Взгляните на следующий код:

```
<P>Это <A HREF="javascript:;" onClick="doSomething();">текст, по которому
  ↳можно щелкать</A>.</P>
```

Мы преобразовали нужный нам фрагмент текста в гиперссылку, привязали к ней обработчик события `onClick`. В качестве значения интернет-адреса мы задали `"javascript:;"`, чтобы Web-обозреватель при щелчке по такой гиперссылке никуда не переходил. Можете проверить — такой код будет работать и в Internet Explorer, и в Navigator.

Чтобы созданная нами гиперссылка совсем уж не походила на гиперссылку, вы можете встроить в тег `<A>` стиль. Задайте атрибуту стиля `text-decoration` значение `none`, чтобы отключить подчеркивание, а атрибуту `color` — значение цвета вашего текста. И пусть после этого кто-нибудь скажет, что это гиперссылка!

Конечно, все вышеописанные манипуляции вы можете выполнить и в среде Dreamweaver. Вы можете ввести текст `javascript:;` в поле ввода **Link** редактора свойств, поставить в только что созданную гиперссылку текстовый курсор и воспользоваться панелью **Behaviors**. Автор описал все это в HTML, чтобы вам было понятнее.

Поведения и анимации

Вы познакомились с двумя, можно сказать, "детищами" Dreamweaver: анимацией и поведением. Они называются так потому, что не являются возможностями ни HTML, ни JavaScript, ни объектной модели документа DOM. И анимация, и поведения "придуманы" самим Dreamweaver, чтобы помочь вам создавать сложные Web-сценарии, решающие весьма непростые задачи. Вы работаете с анимациями и поведениями, используя удобный интерфейс Web-редактора, создаете их, изменяете и удаляете, а Dreamweaver за вашей спиной выполняет над кодом довольно сложные действия.

Неудивительно, что анимации и поведения могут взаимодействовать. Вы уже знаете, как проиграть анимацию в ответ на какое-либо действие пользователя. Но, кроме того, вы можете привязать поведение к тому или иному кадру анимации, и когда системный таймер "дотикает" до соответствующей отметки шкалы времени, это поведение сработает. Например, таким образом вы можете заставить звуковой файл проигрываться в нужный момент, либо во время анимации показывать на странице скрытые до поры до времени свободные элементы.

Давайте еще раз откроем панель **Timelines**, выбрав пункт **Timelines** подменю **Others** меню **Window**, или иным способом. И еще раз посмотрим на нее (см. рис. 12.1). Над серой линейкой с отметками времени (в кадрах) вы можете увидеть еще один канал анимации, пока еще нами не использованный. Это *канал поведений*. Он обозначен буквой **B**, от англ. *behavior* — поведение.

Чтобы привязать к какому-либо кадру анимации поведение, сначала выберите необходимый кадр в канале поведений. В результате этот кадр будет выделен черным. Далее выберите в меню поведений нужный пункт, например, **Show-Hide Layers**. Как обычно, на экране появится диалоговое окно задания параметров выбранного поведения, в нашем случае — **Show-Hide Layers**. Введите параметры и нажмите кнопку **OK**. И поведение, привязанное к кадру анимации, будет создано.

Против соответствующего кадра анимации в канале поведений появится небольшой значок (рис. 13.35). А в списке панели **Behaviors** — новое поведение, привязанное к событию `onFrame<номер кадра>` (рис. 13.36). Такого события в объектной модели документа не предусмотрено; Dreamweaver сам создал его так же, как создал анимации и поведения.

Рис. 13.35. Кадр анимации в канале поведений, к которому привязано поведение

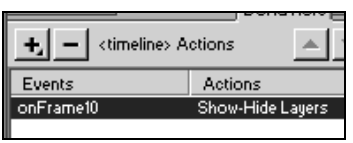


Рис. 13.36. Поведение, привязанное к кадру анимации, в списке панели **Behaviors**

Таким образом, вы можете создать сколько угодно поведений, привязанных к кадрам анимации. Чтобы просмотреть поведения, привязанные к какому-либо кадру, просто выберите его в канале поведений панели **Timelines**. Имейте в виду, что выбирать можно только кадры, помеченные знаком, показанным на рис. 13.35. Также вы можете привязать к одному кадру не-

сколько поведений, просто добавив их в список поведений при выбранном кадре.

Вы можете изменить параметры привязанного к кадру анимации поведения, дважды щелкнув по соответствующей строке в списке поведений. Чтобы удалить ненужное поведение, либо удалите его из списка поведений, либо используйте пункт **Remove Behavior** контекстного меню панели **Timelines** или одноименный пункт подменю **Timeline** меню **Modify**.

Вы можете удивиться, почему автор не использовал для создания поведений пункт **Add Behavior** контекстного меню панели **Timelines**. (Кроме того, в подменю **Timeline** меню **Modify** есть пункт **Add Behavior to Timeline**.) Если вы выберете этот пункт, Dreamweaver выведет на экран предупреждение, предлагающее вам воспользоваться для создания поведения тем самым путем, который мы описали. Зачем разработчикам Dreamweaver это понадобилось, непонятно.

Ну, вот и все о создании поведений и о написании Web-сценариев вообще. Поговорим о следующем, последнем шаге работы над любой программой — об отладке.

Отладка Web-сценариев

Но прежде, чем мы начнем говорить об отладке Web-сценариев, поговорим об ошибках и о том, как они выявляются.

Какие ошибки бывают в программах

Очень часто только что написанная программа содержит ошибки. Более того, если она имеет объем более десяти строк, будьте уверены — хоть одна ошибка там обязательно присутствует. (Уж не говоря о более сложных программах...) Проистекает все это из пресловутой "ненадежности" людей. В самом деле, компьютеры при всех их недостатках никогда не ошибаются сами по себе. Все компьютерные ошибки всегда обусловлены либо неблагоприятными внешними воздействиями (космическое излучение, скачки напряжения в сети, отсутствие дискеты в дисководе и т. п.), либо внутренними (ненадежная электроника, опять же, ошибки в программах). Только люди ошибаются без всяких причин.

Ошибки в программах бывают двух видов:

- *Синтаксические ошибки* — это неточности в написании самого программного кода. В результате синтаксической ошибки интерпретатор или компилятор не может правильно расшифровать выражение. В этом случае он выдает сообщение о синтаксической ошибке. Например, Internet Explorer огорошит нас таким сообщением — см. рис. 13.37.

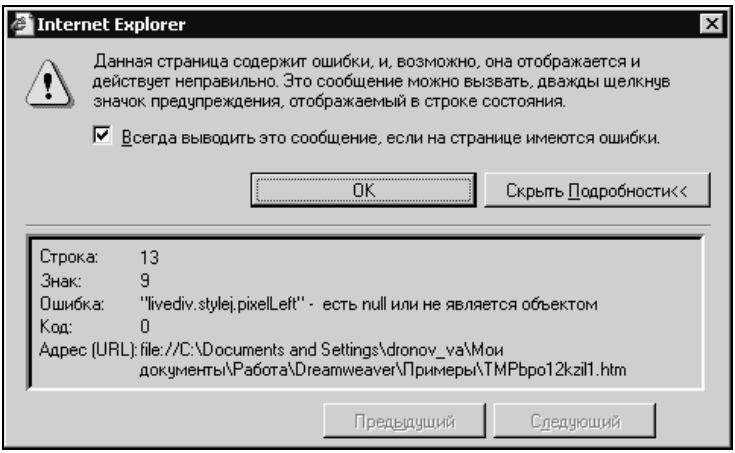


Рис. 13.37. Сообщение о синтаксической ошибке Internet Explorer

- *Логические ошибки* — это ошибки в логике работы программы. Программный код выглядит абсолютно правильным, но работает неправильно. Такие ошибки выявить и исправить намного сложнее.

Обычно программист сразу же после написания программы обязательно ее проверит, "погоняет" сам, чтобы прячущиеся в ней ошибки "вылезли" наружу, проявились. Этот процесс и называется *отладкой*. За отладкой обычно следует исправление найденных ошибок и новая отладка. И так повторяется довольно долго, пока программист не будет уверен, что ошибок в его творении больше нет.

Не всегда, правда, это бывает так. Очень часто уже готовые программы содержат ошибки. Но разговор сейчас не об этом. Давайте поговорим, почему процесс отладки не всегда позволяет выявить все ошибки.

Вообще-то все ошибки в достаточно сложной, многокомпонентной программе выявить невозможно. Дело в том, что программа может устанавливаться на компьютерах самых разных конфигураций, и для того, чтобы быть уверенным, что программа действительно будет на всех них работать, разработчик должен на всех них ее и проверить. Сами понимаете, что это физически невозможно. Другое дело, что программист может пропустить какую-то ошибку, не исправить ее. Здесь можно говорить как о его невнимательности, так и о его ограниченности. С невнимательностью все понятно. А вот с ограниченностью...

Программный код на современных компьютерах выполняется с большой скоростью. Более того, этот код зачастую представляет собой "вещь в себе"; разработчик не может "влезть" внутрь его, посмотреть, что хранится в той или иной переменной, хотя часто без этого не обойтись. В этом его ограниченность.

А человек всегда преодолевает свою ограниченность посредством созданных им машин. Поэтому были созданы специальные программы — *отладчики*, позволяющие "заглянуть" в программный код и увидеть, как он работает. Сейчас все средства для разработки программ имеют в своем составе такие отладчики.

Не стал исключением и Dreamweaver. Он позволяет вам проверить программы, используемые в Web-страницах, а именно Web-сценарии. Если вы пишете сценарии сами, без отладчика вам будет трудно обойтись.

Какие же возможности предлагают нам современные программы-отладчики?

1. Установка так называемых *точек останова*. Это особые метки в коде программы, на которых выполнение приостанавливается, после чего программист может посмотреть на результат работы своей программы. После этого выполнение программы может быть продолжено до ее конца или до следующей точки останова. Точки останова применяются, чтобы выяснить, что происходит в том или ином участке кода программы.
2. *Пошаговое выполнение* кода. Программист может выполнять программный код выражение за выражением и наблюдать за результатами выполнения. Пошаговое выполнение позволит точно узнать, как выполняется программа, и что происходит во время ее выполнения.
3. *Просмотр значений переменных*. Может пригодиться практически всегда.

Так, с отладчиками вообще мы разобрались. Обратимся теперь к конкретному представителю этого славного семейства — встроенному отладчику Dreamweaver.

Встроенный отладчик Dreamweaver

Чтобы вызвать встроенный в Dreamweaver отладчик JavaScript-кода, нажмите комбинацию клавиш <Alt>+<F12> или <Ctrl>+<Alt>+<F12>. Первая комбинация запускает отладку в первичном Web-обозревателе, а вторая — во вторичном. (О первичном и вторичном Web-обозревателях см. *главу 2*.) Также вы можете выбрать нужный пункт в подменю **Debug in Browser** меню **File** или меню кнопки **Preview/Debug in Browser** инструментария документа (см. рис. 2.31). После этого нажмите кнопку **OK** в нескольких появившихся на экране окнах-предупреждениях, чтобы собственно запустить отладчик. И подождите некоторое время, пока все это загрузится и запустится.

Внимание!

Для того чтобы запустить отладчик Dreamweaver, вы должны иметь на своем компьютере установленную виртуальную машину Java фирмы Microsoft. Будьте внимательны: в составе Windows XP эта виртуальная машина не поставляется — вам самим придется загружать ее с сайта этой фирмы.

После загрузки и запуска отладчика Dreamweaver выведет небольшое предупреждение с кнопкой **OK** и флажком **Don't show me this message again**. Это предупреждение предлагает вам установить точки останова и запустить отладку. Закройте это окно, нажав **OK**; вы можете перед этим включить флажок **Don't show me this message again**, если больше не хотите его видеть.

Само окно отладчика показано на рис. 13.38. Оно разделено на две части. В верхней части отображается выполняемый программный код, причем отладчик не разделяет HTML- и JavaScript-код, а отображает все вместе. В нижней части находится список переменных и их текущих значений, изначально пустой. Вы можете изменять относительные размеры этих двух частей, перемещая мышью находящуюся между ними полоску разделителя. В самом верху окна находится инструментарий отладчика, с помощью которой выполняются все доступные действия в этом окне.

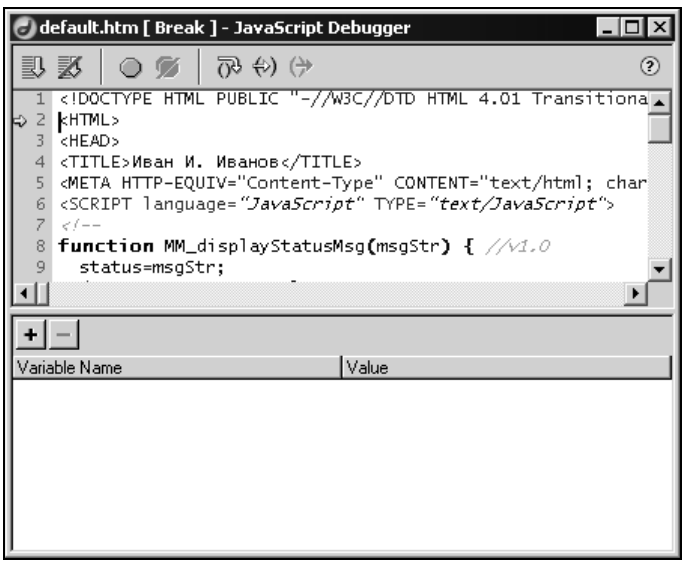


Рис. 13.38. Окно встроенного отладчика Dreamweaver

Программный код, отображаемый в верхней половине окна отладчика, пронумерован по строкам. Номера строк отображаются на серой полосе, тянущейся вдоль левого края всей верхней половины. Строка, которая в данный момент должна быть выполнена, помечена маленькой желтой стрелкой, направленной вправо.

Теперь мы должны, как предлагает Dreamweaver, установить точки останова и запустить отладку. Давайте так и сделаем.

Чтобы установить точку останова, поставьте текстовый курсор на нужной строке программного кода (кода JavaScript, а не HTML) и нажмите клавишу

<F7>. Вы также можете выбрать пункт **Set/Remove Breakpoint** в контекстном меню или нажать одноименную кнопку инструментария отладчика (рис. 13.39, слева). Точка останова обозначается специальным маркером — небольшим красным кружком, стоящим на серой полосе с номерами строк напротив соответствующей строки кода — см. рис. 13.40.

Рис. 13.39. Кнопки **Set/Remove Breakpoint** и **Remove All Breakpoints** инструментария отладчика

Рис. 13.40. Маркер точки останова

Если вы случайно ошиблись и поставили точку останова не там, поставьте на строке, где стоит точка останова, текстовый курсор и снова нажмите клавишу <F7>, кнопку **Set/Remove Breakpoint** или пункт в контекстном меню. Вы также можете убрать одновременно все точки останова в коде, нажав кнопку **Remove All Breakpoints** (рис. 13.39, справа) или выбрав в контекстном меню одноименный пункт.

Расставив точки останова, можно запускать код на выполнение. Для этого нажмите кнопку **Run** инструментария отладчика (рис. 13.41, слева) или клавишу <F8>. Кнопка **Stop Debugging** (рис. 13.41, справа) позволит вам отключить отладчик и продолжить выполнение JavaScript-кода без него.

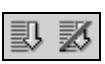


Рис. 13.41. Кнопки **Run** и **Stop Debugging** инструментария отладчика

Когда Web-сценарий выполняется из-под отладчика, его быстродействие невелико, особенно на маломощных компьютерах. Это происходит потому, что отладчик перехватывает и проверяет каждое выражение, выполняемое интерпретатором. Поэтому, чтобы увидеть реальное быстродействие вашего кода, "прогоните" его впоследствии без отладчика.

Дойдя до ближайшей установленной вами точки останова, выполнение кода временно прервется. Маркер точки останова в этом случае примет такой вид — см. рис. 13.42. Чтобы запустить его дальше, нажмите кнопку **Run** инструментария отладчика или клавишу <F8>.

Рис. 13.42. Маркер точки останова, на которой остановилось выполнение кода

Вы также можете использовать кнопки **Step Over**, **Step Into** и **Step Out** инструментария отладчика для пошагового выполнения кода, выражение за выражением. Эти кнопки показаны на рис. 13.43.

Рис. 13.43. Слева направо: кнопки **Step Over**, **Step Into** и **Step Out** инструментария отладчика

Нажатие кнопки **Step Over** или клавиши <F9> вызывает простой переход на следующее выражение.

Кнопка **Step Into** (нажатие клавиши <F10>) аналогична **Step Over** за одним исключением. Если следующее выражение программного кода является вызовом функции, то далее начнет выполняться первое выражение кода функции, т. е. вы как бы "зайдете" в функцию. Имейте в виду, что "зайти" таким образом можно только в функции, созданные самим программистом.

Кнопка **Step Out** (нажатие клавиши <F11>) действует только тогда, когда выполняется код какой-либо функции. При ее нажатии выполняется весь код функции, и очередная остановка происходит на выражении, следующим за вызовом этой функции, т. е. кнопка **Step Out** позволит вам "выйти" из функции.

Теперь обратимся к списку переменных, находящемуся в нижней части окна отладчика. Вы видите, что этот список, как и многие списки Dreamweaver, представляет собой таблицу, состоящую из двух колонок. В первой колонке — **Variable Name** — отображается имя переменной или свойства, а во второй — **Value** — его значение. Вы можете выделить любую строку в списке и выполнить над ней некоторые действия.

Над самим списком находятся две кнопки со знаками "плюс" и "минус". Их назначение уже должно быть вам понятно. Кнопка со знаком "плюс" добавляет новую строку в список, после чего вам останется ввести в добавленную строку имя нужной переменной или нужного свойства и нажать клавишу <Enter> (рис. 13.44). Кнопка со знаком "минус" позволит вам удалить выделенную в списке строку. Как видите, все очень удобно, за исключением того, что, как вы уже заметили, русские буквы опять отображаются некорректно.

Variable Name	Value
window.status	"-B> B5;AB!!!"
document.MM_returnValue	[null]

Рис. 13.44. Две переменные и их значения в списке переменных отладчика

Также вы можете изменить имя переменной (свойства) или его значение. Для этого выберите нужную строку в списке и щелкните мышью по значе-

нию в колонке **Variable Name** или **Value**. После этого введите новое имя или значение в появившееся в строке поле ввода и нажмите клавишу <Enter>.

Чтобы закончить отладку, либо закройте окно отладчика и окно Web-обозревателя, либо нажмите кнопку **Stop Debugging**. В первом случае выполнение всех сценариев будет прервано, а сам Web-обозреватель — выгружен, а во втором — выполнение продолжится без отладчика.

Альтернативные технологии

О недостатках Web-сценариев и их преодолении было написано в *главе 12*. Здесь же можно только кое-что добавить. А именно, опишем две альтернативные технологии, которые можно применять вместо сценариев для расширения функциональности Web-страниц.

Прежде всего, давайте вспомним недостатки Web-сценариев, написанных на интерпретируемых языках. Таких недостатков два, и оба они могут быть критичными во многих случаях, когда Web-дизайнеру понадобится расширить функциональность своих страниц.

1. Невысокое быстродействие интерпретируемого программного кода.
2. "Открытость", т. е. легкочитаемость, интерпретируемого кода.

Оба эти недостатка являются следствием самой природы языка JavaScript и никак не решаются. Единственное: их можно обойти, не создавая слишком сложных сценариев и не реализуя в них ноу-хау. И, если вам все же придется создавать сложные сценарии, содержащие какие-либо уникальные и, не дай бог, патентованные разработки, обратите внимание на альтернативные Web-сценариям технологии.

А таких технологий всего две. И обе они являются близкими родственниками, что бы ни говорили пламенные приверженцы каждой из них. И обе они найдут свои области применения.

Первая технология вам уже знакома — это элементы ActiveX, рассмотренные нами в *главе 4*, посвященной графическим изображениям и вообще внедренным элементам. Напомним, что это такое.

Элементы ActiveX были разработаны как общий стандарт компонентов для использования в различных приложениях, Web-страницах и в качестве расширений самой операционной системы Windows. Эта технология представляет собой дальнейшее развитие технологии *COM* (Component Object Model — объектная модель компонентов), позволяющей встраивать одни программы (так называемые *COM-серверы*) в другие (*COM-клиенты* или *COM-контейнеры*). Элемент ActiveX представляет собой библиотеку DLL, содержащую сервер COM; как всякий COM-сервер, он регистрируется в особом разделе Реестра Windows, чтобы его смогли найти клиенты.

Элемент ActiveX помещается на Web-страницу с помощью парного тега <OBJECT>...</OBJECT>. Внутри этого тега пишутся определения свойств элемента, выполненные с помощью набора одинарных тегов <PARAM>. Подробнее о работе с элементами ActiveX в среде Dreamweaver *см. главу 4*.

Элементы ActiveX создаются на компилируемых языках программирования, поэтому реализованный у них код работает очень быстро и почти не поддается "взлому". Распространяются они по Сети в виде дистрибутивных файлов, представляющих собой файл-архив, содержащий все необходимые для работы элемента ActiveX файлы. Как правило, Web-обозреватель, если нужный ему элемент ActiveX на компьютере не установлен, сам загружает его дистрибутивный файл, руководствуясь значениями атрибутов тега <OBJECT>.

Недостатками элементов ActiveX являются привязанность к операционной системе Windows и невысокая безопасность. Как утверждают знатоки, злоумышленнику ничего не стоит написать вредоносный элемент ActiveX и внедрить его на внешне безобидную Web-страницу. После того как этот элемент будет загружен на компьютер жертвы, он сможет натворить там много бед. Однако о конкретных примерах таких атак на клиентские компьютеры автору пока что ничего не известно. И все же, элементы ActiveX нашли достаточно ограниченное применение, в основном, в корпоративных интрасетях, построенных на платформе Windows.

Вторая технология, о которой обязательно нужно рассказать и которая в наши дни, кажется, завоевала популярность, — это *апплеты* Java. Апплеты (калька с англ. *applet* — буквально, "приложеньице") — это аналоги элементов ActiveX, но написанные на языке программирования Java и не привязанные к какой-либо компьютерной платформе. По большому счету, они похожи на элементы ActiveX за несколькими исключениями, которые сейчас будут перечислены.

Прежде всего, программный код апплетов Java выполняется не самим процессором компьютера, а интерпретатором (так называемой виртуальной машиной) Java. Этот особый программный код, называемый *байт-кодом*, не является командами центрального процессора. Он порождается компилятором Java, который как бы компилирует программу "не до конца". Такие "недокомпилированные" программы выполняются значительно быстрее, чем исходный код JavaScript, но все же несколько медленнее, чем программы, откомпилированные "до конца", в "родные" команды процессора. С другой стороны, программы в байт-коде не привязаны к конкретному процессору, а значит, к конкретной компьютерной платформе, поэтому могут быть выполнены где угодно. Эта платформонезависимость и обусловила бóльшую популярность Java-апплетов по сравнению с элементами ActiveX.

Далее, интерпретатор Java-кода организует для апплетов своего рода "компьютер в компьютере", "внутри" которого они и исполняются. (Поэтому интерпретатор Java и называют виртуальной, "кажущейся" машиной.) "В реаль-

ный компьютер" апплеты "выбраться" оттуда не могут; благодаря этому даже вредоносный Java-апплет, созданный с явно преступными целями, не может серьезно набедокурить — интерпретатор его просто остановит, предупредив об этом пользователя.

Апплеты Java помещаются на страницу с помощью парного тега <APPLET>... </APPLET>. Как и у тега <OBJECT>, внутри него находятся определения свойств апплета, выполненные с помощью набора одинарных тегов <PARAM>.

Чтобы поместить на страницу апплет Java, работая в Dreamweaver, вы можете нажать кнопку **Applet** (рис. 13.45), находящуюся на вкладке **Media** панели объектов. Также вы можете выбрать пункт **Applet** подменю **Media** меню **Insert**. После этого на экране появится диалоговое окно **Select File**, где вы должны будете задать имя файла, содержащего код апплета (так называемый *файл класса апплета* Java). Помещенный на страницу апплет выглядит так — см. рис. 13.46.

Рис. 13.45. Кнопка **Applet** панели объектов

Рис. 13.46. Помещенный на страницу апплет Java

Вид редактора свойств при выбранном апплете Java показан на рис. 13.47. Мы рассмотрим подробно только те параметры, которые характерны именно для апплета. А таких параметров всего два.

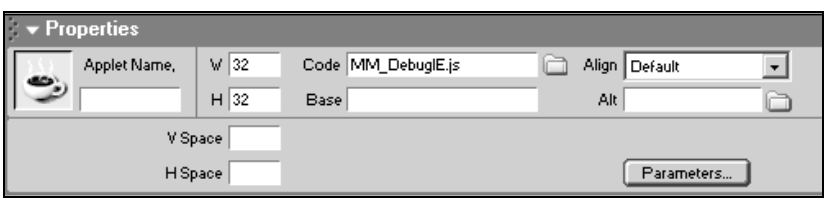


Рис. 13.47. Вид редактора свойств при выбранном апплете Java

В поле ввода **Code** задается имя файла класса, реализующего данный апплет. Вы можете также щелкнуть по значку папки справа от поля ввода и выбрать нужный файл в диалоговом окне **Select File**. Также вы можете выбрать пункт **Code** контекстного меню апплета.

Файл класса Java может содержаться как на вашем компьютере, так и на удаленном Web-сервере. Во втором случае Web-обозреватель скопирует

файл класса и все файлы, необходимые для его работы, на компьютер клиента и поместит их в специальную папку. В дальнейшем, если этот же апплет понадобится снова, он будет загружать его оттуда вместо того, чтобы снова "тянуть" по Сети.

В поле ввода **Base** вводится папка Web-сервера, где находится файл класса, реализующего апплет, и все необходимые для его работы файлы. Если вы пользовались диалоговым окном **Select File** для выбора файла класса, Dreamweaver сам подставит в поле ввода **Base** нужное значение.

Остальные параметры (и соответствующие им элементы управления редактора свойств) сходны с параметрами элемента ActiveX, описанного в *главе 4.*

Где же можно найти нужные элементы ActiveX и апплеты Java? Во-первых, вы можете сами их написать. Но для этого придется изучить либо Java, либо один из компилируемых языков программирования (C++, Pascal и т. п.). Во-вторых, вы можете поискать бесплатные или условно-бесплатные элементы ActiveX или апплеты, выложенные на специальных сайтах-архивах для общего использования. (Кстати, существуют аналогичные архивы и для JavaScript-сценариев.) В-третьих, вы можете их купить или заказать другому программисту.

В любом случае, ответ на вечный вопрос "где достать?" выходит за рамки этой книги.

Что дальше?

Ох, сколько всего можно сотворить на обычной Web-странице! Просто глаза разбегаются...

Кстати, насчет глаз... То, что делалось до этого, в большинстве случаев затрагивает внешний вид наших страничек (форматы текста, таблицы, дизайн, изображения, анимации, сценарии...). А в следующей главе мы затронем нечто такое, что "невооруженным" глазом не видно. Это "нечто" позволит нам нанести завершающие штрихи на наше Web-творение.

Итак, просим любить и жаловать: метатеги и серверные директивы!

ГЛАВА 14

Метатеги и серверные директивы

А сейчас будет рассказано о том, чему неопытные Web-дизайнеры мало уделяют внимания. Это возможности не Dreamweaver, а самого языка HTML и различных программ Web-серверов, используемых во Всемирной паутине. Это так называемые метатеги и серверные директивы.

Метатеги позволят вам поместить в ваши Web-страницы дополнительные указания Web-обозревателю, Web-серверу и другим программам, о которых будет рассказано в дальнейшем. В частности, с помощью метатегов Dreamweaver задает текстовую кодировку, в которой созданы ваши страницы. Также с помощью метатегов вы можете сделать своим страницам неплохую рекламу на просторах Сети... Да-да, и здесь реклама! А куда же без нее...

Серверные директивы — это особые команды по изменению кода HTML, исполняемые Web-сервером. В частности, они позволяют поместить в текст страницы текущую дату, причем это выполнит сам Web-сервер — вам нужно будет только вписать в HTML-код несколько символов. Также имеется возможность вставить в HTML-код содержимое другого файла. Вы еще не догадались, что это позволяет сделать? Тогда слушайте...

Стоп! Не будем торопить события. Скоро вы обо всем узнаете. Прежде чем рассказывать о метатегах и серверных директивах, нам нужно выяснить, зачем они нужны и какую пользу могут принести нам, простым Web-дизайнерам. А для этого слегка отойдем от Web-дизайна и поговорим о некоторых вещах, напрямую не относящихся к теме этой книги. И первой этой темой будет реклама сайта в Сети.

Реклама в Интернете

Вообще, рекламировать сайты в Сети можно по-разному. Очень грубо сетевую рекламу можно разделить на две разновидности: *активную* и *пассивную*

или *"раскрутку"*. Активная реклама — самая очевидная. Она включает в себя рисование рекламных баннеров, "расклеивание" их по чужим страницам в обмен на чужие баннеры, упоминание в сетевых новостях, написание хвалебных статей в сетевые СМИ, открытие почтовой рассылки и т. п., т. е. вы активно призываете сетевой люд посетить ваш сайт.

А вот о пассивной рекламе так коротко не скажешь. Начать придется издалека...

Поисковые машины

Чтобы выяснить, что же такое пассивная реклама сайта, нам придется поговорить о *поисковых машинах* — особых Web-сайтах, предназначенных для поиска информации в Интернете. Если вы давно во Всемирной паутине, то знаете, что такое поисковая машина и как ей пользоваться. Для тех из вас, кто пока не так искушен в сетевых реалиях, предлагается небольшой ликбез.

А для этого придется углубиться в древнюю историю Интернета...

Итак, мысленно перенесемся в 1989 год. После создания языка HTML Сеть, а именно, та, что стали называть Всемирной паутиной WWW, стала расти как на дрожжах. Сначала Сетью пользовались, в основном, ученые, а значит, информация туда выкладывалась большей частью серьезная, небольшими частями, и найти ее было несложно. Некоторые из старых интернетчиков называют это время "золотым веком" Сети. Что ж, может, они и правы... Но только отчасти.

Впоследствии произошло то, чего одни ждали с нетерпением, а другие — с ужасом. Сеть перестала быть пристанищем ТОЛЬКО ученого люда. В нее пришли сначала всяческого рода маргиналы, жаждущие общения, потом — компьютерщики, а в самое последнее время — обыватели, занятые в "некомпьютерных" областях человеческой деятельности. Сеть стала огромной, качество (в смысле, достоверность) выкладываемой в нее информации катастрофически снизилось, а поиск нужной информации среди тонн электронного хлама стал представлять серьезную проблему.

Эту проблему надо было как-то решать. И ее решили, когда Сеть набрала достаточный объем.

В 1995 году несколько американских студентов решили объединить свои коллекции полезных интернет-ссылок в своего рода базу данных, которой мог бы пользоваться каждый. При этом база данных содержала и сами адреса Web-страниц, и их описания, и набор *ключевых слов* — особых слов, однозначно описывающих содержимое каждой страницы (например, для музыкальных сайтов такими словами будут "музыка", "MP3", "аудио" и т. п.). Потом они написали специальную программу, осуществляющую поиск адреса по набранному в поле ввода слову и выдающую результат в виде Web-страницы с набором найденных ссылок. Кроме того, они написали еще од-

ну программу, сканирующую Сеть в поисках новых Web-страниц и заносящую новые адреса в базу данных; сейчас такую программу называют *поисковым агентом*, *поисковым роботом*, или, в шутку, *"пауком"*. Так возникла первая поисковая машина "Yahoo!" (**http://www.yahoo.com**), получившая огромнейшую популярность и существующая до сих пор.

Задумка оказалась настолько удачной, что породила волну подражаний. Поисковые машины плодились, как грибы после дождя. Сейчас их существует столько, что список даже более-менее крупных из них занял бы несколько страниц убористого текста. Ограничимся только несколькими (табл. 14.1).

Таблица 14.1. *Популярнейшие поисковые машины*

Зарубежные/ отечественные	**Название**	**Интернет-адрес**
Зарубежные	Yahoo!	**http://www.yahoo.com**
	AltaVista	**http://www.altavista.com**
	ListBot	**http://www.listbot.com**
	InfoSeek	**http://www.infoseek.com**
	Lycos	**http://www.lycos.com**[1]
	Google	**http://www.google.com**
Отечественные	Апорт	**http://www.aport.ru**
	Яндекс	**http://www.yandex.ru**
	Рамблер	**http://www.rambler.ru**

Поисковые машины делятся на две большие группы, отличающиеся принципом работы — на классические поисковые машины и каталоги ссылок. Сейчас мы их рассмотрим.

Классическая поисковая машина (поисковик) — это, грубо говоря, база данных ссылок плюс программа поиска. Это поисковая машина в чистом виде. Она занимается тем, что хранит адреса Web-страниц в базе данных и выдает пользователям результаты поиска по ключевым словам. При этом она работает полностью в автоматическом режиме: сама находит и заносит в базу данных новые Web-страницы, сама формирует их описания и наборы ключевых слов, сама обновляет адреса уже существующих страниц, если они изменились, и их описания.

Присутствие пресловутого "человеческого фактора" в работе классических поисковиков минимально. (Хотя, разработчик может заложить в программу-

[1] Существует русская версия — **http://www.lycos.ru**.

агента и программу поиска какие-то свои наработки в плане оптимизации поиска, но на этом влияние человека на машину заканчивается.)

Также в базах данных поисковиков отсутствует какая-либо систематизация результатов их бесконечного поиска. Они работают по принципу "вали валом — потом разберем". (За тем исключением, что это "потом" никогда не наступит.) Они просто собирают адреса, автоматически формируют описание и набор ключевых слов и "валят" это в свою базу данных. Они могут лишь искать и регистрировать, потому что они — только программы. Способность же анализировать и систематизировать доступна исключительно человеку.

Каталоги ссылок базируются совсем на другом принципе. Во-первых, они пополняются не роботами, а людьми; люди заносят в них адреса и описания страниц и наборы ключевых слов для поиска. Во-вторых, ссылки в их базах данных жестко систематизированы и разбиты по разделам и подразделам. В-третьих, в каталог допускаются далеко не все страницы, а только те, которые действительно содержат что-то полезное.

Как же выглядит процесс занесения новой ссылки в каталог? Давайте распишем его по шагам.

1. Владелец сайта формирует набор параметров, вносимых в каталог. Это название сайта, его краткое описание, набор ключевых слов, разумеется, интернет-адрес сайта и свой почтовый адрес. Почтовый адрес может понадобиться, например, чтобы уведомить владельца о том, что его сайт внесен в базу данных каталога.

2. Владелец сайта заходит на особую Web-страницу и вводит все эти данные в форму ввода, после чего они сохраняются в базе данных новых поступлений. Заметьте, что эти данные не помещаются сразу в основную базу данных каталога, в которой, собственно, и осуществляется поиск, — это будет выполнено позже.

3. Человек, работающий на администрацию каталога, своего рода цензор или, как говорят опытные интернетчики, *модератор* каталога, просматривает данные вновь внесенного сайта в базе данных новых поступлений. После этого он заходит на этот сайт и проверяет, соответствует ли он введенным владельцем данным. Если это так, данные о новом сайте переносятся в основную базу данных каталога, а владельцу сайта посылается соответствующее уведомление. В противном случае данные о новом сайте удаляются, о чем его владельцу опять же посылается уведомление.

4. Владелец сайта получает посланное ему уведомление. Если его сайт внесен в базу данных каталога, это уведомление содержит номер, под которым сайт находится в базе данных, и пароль для доступа к его данным. В дальнейшем владелец может изменить данные сайта в базе данных каталога. Если же новый сайт почему-то не попал в каталог, модератор

извиняется перед его владельцем и советует наполнить его Web-творение чем-нибудь полезным.

Недостатками каталогов являются медленная работа и неполный охват Сети. Люди-модераторы работают значительно медленнее программы, осуществляющей поиск страниц в Сети, поэтому процесс добавления данных о новом сайте в базу каталога может затянуться надолго. К тому же, люди часто ошибаются. Далее, из-за такой достаточно жесткой цензуры новых поступлений может оказаться, что каких-то сайтов или страниц в базе данных каталога нет, т. к. модератор их почему-то забраковал.

В настоящее время классические поисковые машины и каталоги в чистом виде очень редко встречаются. Абсолютное большинство современных поисковых машин используют оба этих подхода, т. е. содержат и классический поисковик, и каталог ссылок. Таким образом, они комбинируют достоинства и обходят недостатки этих двух разновидностей поисковых машин.

И все же, несмотря на обилие поисковых машин обоих типов поиск в Интернете — задача чрезвычайно сложная. Особенно она осложняется, если вам нужно найти что-то редкое или очень конкретное. Существует даже такая профессия — искатель информации в Интернете, профессия достаточно редкая, требующая, в основном, интуиции и поэтому хорошо оплачиваемая. Это даже не профессия — это призвание, как, например, музыкант или писатель.

Ну ладно. Оставим в покое каталоги (с ними все ясно) и подробнее поговорим о классических поисковиках. Точнее, о программах-агентах, осуществляющих поиск новых Web-страниц. Давайте выясним, как они работают.

Как работают поисковые агенты

Поисковый агент — это программа, периодически сканирующая всю Сеть и проверяющая, есть ли по тому или иному адресу Web-страница. Если страница есть, агент проверяет ее HTML-код, извлекает из него интернет-адреса, ведущие на другие Web-страницы. Также он извлекает ее название и пытается извлечь описание и набор ключевых слов, если они есть, после чего заносит все это в базу данных поисковика (или, как говорят опытные интернетчики, выполняет *индексирование*). Далее он обращается к Web-страницам, находящимся по найденным на текущей странице адресам, и проделывает с ними то же самое.

Поисковые агенты, как правило, пишутся на заказ очень опытными и знающими программистами. Хороший поисковый агент — настоящее произведение программистского искусства. Он должен отсекать несуществующие адреса, правильно опознавать HTML-теги, "интеллектуально" создавать описания и при этом быстро работать. Сами понимаете: Сеть велика, а времени всегда не хватает...

Вообще, проанализировать HTML-код — задача не очень сложная. В самом деле, название страницы помещается внутри тега <TITLE> в секции заголовка <HEAD>, а интернет-адреса других страниц с большой долей вероятности следует искать в тегах <A> и <AREA>. Написать программу, извлекающую такую информацию, для опытного программиста — пара пустяков.

Сложность здесь совсем в другом.

Давайте вспомним, какие данные помещаются в базу данных поисковой машины, и классического поисковика, и каталога. Кроме адреса и названия страницы, это еще и краткое описание и набор ключевых слов. А откуда их взять?

Проанализировать текст, помещенный на Web-странице? А как? Как обычная программа, даже очень сложная и подающая признаки легкого "интеллекта", узнает, какое из многих сотен, а то и тысяч слов текста страницы можно использовать как ключевое? Как она "ужмет" многокилобайтовый текст в краткое описание из 200 символов (такое ограничение на размер текста описания накладывают многие каталоги)? Как при этом выбрать нужную информацию и выжать всю "воду"?

Нет, это задача непроста даже для человека-модератора. Что уж требовать от программы...

Да, но как-то нужно выбирать из текста страниц необходимую информацию!

Как раз для этого и предназначены метатеги HTML. Именно с их помощью можно представить в HTML-коде информацию, которая не будет предназначена человеку, а исключительно программам. Человек ее даже не увидит (если, конечно, не станет специально для этого просматривать HTML-код).

Конечно, поисковые агенты пользуются для добычи информации не только метатегами. Они также проверяют текст страницы, причем, в зависимости от тега, в котором находится этот текст, они придают ему разное значение. Также они сравнивают, какие данные находятся в метатегах и основном тексте страницы, и на основании результатов этого сравнения делают определенные выводы. Уже говорилось, что хороший поисковый агент — это настоящее произведение программистского искусства, несущее в себе частицу его создателя.

А теперь все же обратимся к метатегам.

Метатеги

Метатег — это особый тег HTML, предназначенный для помещения в код Web-страницы информации о ней самой. Эта информация используется, в основном, программами — поисковыми агентами, Web-обозревателями и, возможно, Web-редакторами — и никак не отображается в окне Web-обозревателя.

С помощью метатегов в код Web-страницы можно поместить следующую информацию:

- краткое описание страницы;
- набор ключевых слов;
- интернет-адрес страницы, на которую будет выполнен переход по истечении некоего промежутка времени (задержка перед перенаправлением на другую Web-страницу);
- базовый адрес, от которого будут отсчитываться все относительные адреса (об абсолютных и относительных интернет-адресах *см. главу 6*);
- "родственную связь" между двумя страницами и тип этой "связи";
- текстовую кодировку страницы;
- некоторые другие данные, создаваемые и используемые, в основном, специальными программами, например популярнейшим Web-редактором Microsoft FrontPage.

Как видите, метатеги позволяют сделать весьма много.

Все метатеги помещаются в секции HTML-заголовка (тег `<HEAD>`) Web-страницы. Большинство из них создается с помощью одинарного тега `<META>` и трех его атрибутов. Тип метатега задается атрибутом `NAME` или `HTTP-EQUIV`, а сами данные — атрибутом `CONTENT`. Ниже приведен пример метатега, задающего текстовую кодировку.

```
<META HTTP-EQUIV="Content-Type" CONTENT="text/html;
  ↳charset=windows-1251">
```

Этот фрагмент взят из HTML-кода страницы default.htm нашего первого сайта Sample site 1.

А следующий метатег задает набор ключевых слов, которые могут быть использованы поисковым агентом:

```
<META NAME="keywords" CONTENT="музыка,аудио,MP3">
```

И не только могут быть, но и используются! Поисковые агенты всегда проверяют содержимое метатегов страниц, чтобы найти нужную для занесения в базу данных информацию. И, "подсовывая" им определенное содержимое, можно делать свой сайт более популярным.

Вот это и есть пассивная интернет-реклама.

Начинающие Web-дизайнеры часто пренебрегают метатегами, недооценивают их важность. И зря. Метатеги — мощный инструмент в руках знающего специалиста; пользуясь ими, он может сделать на какое-то время популярным даже совершенно провальный интернет-проект. И наоборот, неудачные метатеги могут "убить" даже очень перспективный сайт. Конечно, это крайние случаи, но...

Давайте же поговорим о пассивной интернет-рекламе и о более активном ее использовании.

Пассивная интернет-реклама

Специалисты по интернет-рекламе уже давно знают, что одни ключевые слова пользуются на поисковых машинах большей, а другие — меньшей популярностью. В самом деле, если вы зайдете на сайт поисковой машины, зачастую на первой же странице будет присутствовать список самых популярных слов, набираемых в поле ввода запроса. На русскоязычных поисковых машинах это, прежде всего, "работа", "музыка", "аудио", "MP3", вездесущие "секс" и "порно" и т. д. и т. п.

Почему так происходит? Вроде бы слов в языке много, а список популярнейших ключевых слов практически не меняется со временем и почти одинаков для любого языка. А все это из-за того, что люди ищут в Сети примерно одно и то же, называя искомое одними и теми же словами. И это правильно: больше шансов найти "музыку", а не "гармоничное сочетание звуков, производимое с использованием особых инструментов". А уж поисками лучшей работы занимаются фактически все, за исключением тех, кто на ней уже работает.

Но шутки в сторону. Выходит, что набор популярнейших ключевых слов совсем невелик. И, поместив все популярнейшие ключевые слова в метатег

```
<META NAME="keywords" CONTENT="музыка,аудио,MP3,работа,секс,анекдот">
```

можно поднять популярность своей Web-странички, даже если на ней нет и намека на музыку в формате MP3 (или, как правильно, MPEG 1, уровень 3) или анекдоты про секс вместо работы. Так это или не так?

Так-то оно так. Да не совсем.

Безусловно, создав удачный набор ключевых слов в метатеге, можно повысить популярность своей страницы или сайта. Но только при том условии, что содержимое этой страницы будет соответствовать ключевым словам.

Когда-то, во времена "глупых" поисковых агентов такие фокусы проходили. Но не сейчас. Поисковые агенты стали не в пример "умнее". Помните, что они, вдобавок к метатегам, проверяют еще и само содержимое страниц, причем учитывают еще и тег, с помощью которого отформатирован тот или иной текст. "Обмануть" такого поискового агента почти так же сложно, как победить агента 007.

Прежде всего, ключевые слова, которые вы перечислили в метатеге, должны также присутствовать в описании страницы. А описание страницы создается с помощью другого метатега:

```
<META NAME="description" CONTENT="Крутая страничка о музыке и аудио
  ↳в формате MP3, работе, сексе с анекдотами обо всем этом">
```

Далее, все перечисленные вами ключевые слова должны присутствовать в названии страницы, задаваемом с помощью тега <TITLE>.

```
<TITLE>Крутая страничка о музыке и аудио в формате MP3, работе, сексе
  ↳с анекдотами обо всем этом</TITLE>
```

Хорошо, если все эти ключевые слова будут встречаться также и в заголовках, созданных с помощью тегов <H1>, ..., <H6>. Причем, чем больше уровень заголовка, тем бóльшим весом будут обладать ключевые слова. Например, заголовок первого уровня:

```
<H1>Музыка и секс</H1>
```

имеет больший вес в глазах поискового агента, чем заголовок шестого уровня:

```
<H6>Семья, работа и прочая досадная ерунда</H6>
```

И наконец, нужно, чтобы ключевые слова присутствовали и в обычном тексте:

```
<P>К вопросу о музыке и сексе в формате MP3. Анекдоты об этом до сих пор
  ↳тревожат международную общественность. Наша работа состоит в том,
  ↳чтобы свести их к минимуму.</P>
```

Теперь нужно дать ответ еще на один важный вопрос. Нужно ли помещать метатеги с описаниями и ключевыми словами на всех страницах сайта или достаточно будет только его главной страницы? Вообще, лучше, если каждая страница сайта будет иметь полный набор метатегов; в этом случае поисковый агент сможет занести в базу данных ВСЕ страницы сайта. И, когда посетитель производит поиск по какому-либо ключевому слову, поисковая машина может выдать ссылку прямо на нужную страницу. Но иногда набором метатегов снабжают только главную страницу сайта, что вполне оправдано для сайтов "для своих", не стремящихся к особой популярности.

В свете вышесказанного, в крайне невыгодном положении оказываются страницы, вообще не содержащие никакого текста. Прежде всего, это наборы фреймов, в которых находится только набор тегов, описывающих различные фреймы и их содержимое. Поэтому в страницах наборов фреймов приходится создавать фиктивное содержимое с помощью парного тега <NOFRAMES>, где и перечислять необходимые ключевые слова. Или, что даже лучше, помещать набор ключевых слов в комментарий.

Как видите, чтобы сделать свою страничку популярной, используя методы пассивной интернет-рекламы, нужно хорошо постараться. Придется не только расставить везде соответствующие метатеги, но и серьезно поработать над содержимым страниц, чтобы ненавязчиво разместить в нем ключевые слова. Иногда для этого даже приходится переделывать весь сайт.

И все же положительный результат не гарантирован. Как сказал кто-то, половина средств, потраченных на рекламу, выброшены на ветер. Знать бы только, какая половина...

К тому же, различные поисковые агенты трактуют информацию, помещенную на Web-страницу, по-разному. Одни тщательно проверяют и метатеги, и содержимое, выискивая соответствия между заявленным и действительным. Другие ограничиваются метатегами, да и то не всеми. Существует даже особая профессия — специалист по поисковым системам. Такие специалисты специально переделывают страницы так, чтобы их "заметил" тот или иной поисковый агент. Таких специалистов во всем мире очень немного (как и профессиональных "искальщиков" информации в Интернете), и ценятся они очень дорого.

Теперь самое время сказать о *"черной рекламе"*, недобросовестных способах "раскрутки" сайтов. Заключаются они в том, что в коде страницы создается набор ключевых слов, совершенно не соответствующих ее содержанию. Кроме того, с помощью специальных приемов на странице также создается особое "скрытое", не видимое посетителям содержимое, призванное "обмануть" поискового агента.

Как можно создать такое содержимое? Например, с помощью тега комментария:

```
<!-- <TITLE>Крутая страничка о музыке и аудио в формате MP3, работе,
  ↳сексе с анекдотами обо всем этом</TITLE> -->
<TITLE>Руководство по разведению суматранских тараканов</TITLE>
```

Как видите, с помощью тега комментария здесь создается второе, невидимое название. Также можно поместить на страницу дополнительный невидимый текст, "согласующийся" с ключевыми словами:

```
<!-- <H1>Музыка и секс</H1> -->
<H1>Ваши тараканы</H1>
<!-- <P>К вопросу о музыке и сексе в формате MP3. Анекдоты об этом до сих
  ↳пор тревожат международную общественность. Наша работа состоит в том,
  ↳чтобы свести их к минимуму.</P> -->
<P>Чтобы ваши тараканы были здоровыми, они должны хорошо питаться. Лучшее
  ↳питание для породистых тараканов — хлебные крошки Pedigree Pal.</P>
```

Конечно, это чистейшей воды жульничество. И, как и любое жульничество, оно рассчитано на получение сиюминутного эффекта. Если вы серьезно работаете над своим интернет-представительством, никогда так не делайте.

Работа с метатегами в среде Dreamweaver

А теперь давайте поговорим о том, как работать с метатегами в среде Dreamweaver. Мы уже знаем, что такое метатеги и как они могут нам помочь (и навредить), осталось выяснить, поддерживает ли их наша любимая программа.

Поддерживает. И замечательно поддерживает!

Однако, чтобы вы смогли с ними нормально работать, вам придется выполнить кое-какие настройки. Вспомните: метатеги представляют информацию, которая никак не отображается на странице. Поэтому нам нужно сделать их видимыми.

Включите пункт-выключатель **Head Content** в меню **View** или нажмите комбинацию клавиш <Ctrl>+<Shift>+<W>. В верхней части окна документа Dreamweaver, ниже инструментариев появится *панель заголовка* Web-страницы (рис. 14.1). На этой панели в виде символических значков представлены все имеющиеся на странице метатеги.

Рис. 14.1. Панель заголовка

Вы можете выбрать любой значок и изменить значения атрибутов метатега, пользуясь редактором свойств. Для помещения на Web-страницу новых метатегов применяется подменю **Head Tags** меню **Insert** или вкладка **Head** панели объектов. Вы можете использовать тот способ, который вам больше нравится. Чтобы удалить ненужный метатег, выберите его и нажмите клавишу <Del>.

А сейчас давайте рассмотрим различные метатеги, которые вы можете поместить на страницу, их параметры и случаи, когда они могут быть полезны.

Описание Web-страницы

Чтобы поместить на страницу метатег, содержащий ее описание, выберите пункт **Description** подменю **Head Tags** меню **Insert**. Также вы можете нажать кнопку **Description** (рис. 14.2), находящуюся на вкладке **Head** панели объектов. На экране появится диалоговое окно **Description**, показанное на рис. 14.3.

Рис. 14.2. Кнопка **Description** панели объектов

Текст описания страницы вводится в область редактирования **Description**. После этого остается нажать кнопку **OK**, чтобы создать метатег описания, или кнопку **Cancel** — для отказа от этого. Готовый метатег описания отображается в панели заголовка значком, показанным на рис. 14.2, т. е. так же, как и кнопка **Description** в панели объектов.

Рис. 14.3. Диалоговое окно **Description**

Не сто́ит создавать слишком большого описания. Будьте кратки. Большие описания очень плохо читаются, если читаются вообще. К тому же, многие поисковые машины ограничивают длину описания двумястами символов.

Dreamweaver и здесь проявляет свою нелюбовь к русским буквам. Поэтому вам придется либо вводить описание страницы латинскими буквами, либо править сам HTML-код вручную.

Если выбрать в панели заголовка метатег описания, редактор свойств примет вид, показанный на рис. 14.4. Как видите, вы можете изменить описание страницы в области редактирования **Description**.

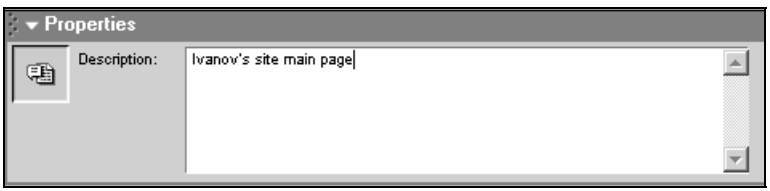


Рис. 14.4. Редактор свойств при выбранном метатеге описания Web-страницы

Ключевые слова

Чтобы поместить на страницу метатег, содержащий набор ключевых слов, выберите пункт **Keywords** подменю **Head Tags** меню **Insert**. Также вы можете нажать кнопку **Keywords** (рис. 14.5), находящуюся на вкладке **Head** панели объектов. На экране появится диалоговое окно **Keywords**, показанное на рис. 14.6.

Рис. 14.5. Кнопка **Keywords** панели объектов

Ключевые слова, разделенные запятыми, вводятся в область редактирования **Keywords**. (Dreamweaver и в этом случае искажает русские буквы, так что

будьте внимательны.) После этого остается нажать кнопку **OK**, чтобы создать метатег ключевых слов, или кнопку **Cancel** — для отказа от этого. Готовый метатег ключевых слов отображается в панели заголовка значком, показанным на рис. 14.5, т. е. так же, как и кнопка **Keywords** панели объектов.

Рис. 14.6. Диалоговое окно **Keywords**

Здесь так же, как и в случае с описанием страницы, не стоит создавать слишком большого набора ключевых слов. Ограничьтесь только теми ключевыми словами, которые максимально полно описывают вашу страницу. Помните, что и на длину набора ключевых слов также существует ограничение в 200 символов. И не гонитесь за дешевой сиюминутной популярностью, помещая в набор все привлекательные ключевые слова, которые можете вспомнить.

Если выбрать в панели заголовка метатег набора ключевых слов, редактор свойств примет вид, показанный на рис. 14.7. Как видите, вы можете изменить набор ключевых слов в области редактирования **Keywords**.

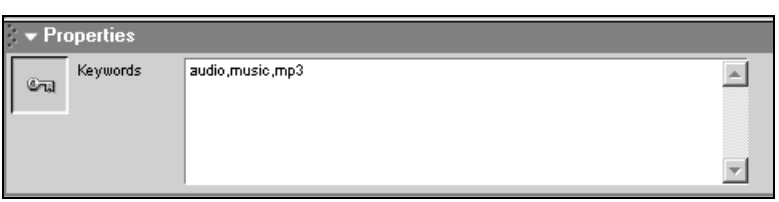


Рис. 14.7. Редактор свойств при выбранном метатеге набора ключевых слов

Базовый интернет-адрес

Метатег, задающий базовый интернет-адрес, может быть полезен, если страницы вашего сайта разбросаны по разным Web-серверам. Вместо того, чтобы в гиперссылках указывать полные интернет-адреса, вы можете указать относительные, а в секции заголовка страницы поместить метатег базового адреса. В этом случае полные интернет-адреса будут вычисляться относительно его.

Кроме того, такой метатег может устранить ошибки загрузки страниц, которые должны отображаться во фреймах, но почему-то выводятся в полном окне Web-обозревателя. Для этого достаточно указать в этом метатеге имя базового фрейма, в котором должна отображаться страница.

Чтобы поместить на страницу метатег базового интернет-адреса, выберите пункт **Base** подменю **Head Tags** меню **Insert**. Также вы можете нажать кнопку **Base** (рис. 14.8), находящуюся на вкладке **Head** панели объектов. На экране появится диалоговое окно **Base**, показанное на рис. 14.9.

Рис. 14.8. Кнопка **Base** панели объектов

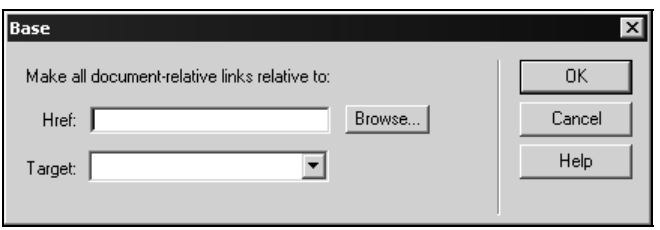


Рис. 14.9. Диалоговое окно **Base**

Базовый интернет-адрес вводится в поле ввода **Href**. Вы также можете нажать кнопку **Browse** и выбрать в диалоговом окне **Select File** какой-либо файл, находящийся в нужной папке. После этого вам останется только удалить имя файла, оставив лишь путь.

Базовый фрейм выбирается с помощью раскрывающегося списка **Target**. Если ваша страница представляет собой набор фреймов, в этом списке будут перечислены все их имена.

Задав базовые адрес и фрейм, нажмите кнопку **OK**, чтобы создать метатег, или кнопку **Cancel** — для отказа от этого. Готовый метатег базового адреса отображается в панели заголовка значком, изображенным на рис. 14.8, т. е. так же, как и кнопка **Base** панели объектов.

Рис. 14.10. Редактор свойств при выбранном метатеге базового интернет-адреса

Если выбрать в панели заголовка метатег базового интернет-адреса, редактор свойств примет вид, показанный на рис. 14.10. В нем вы можете изменить сам базовый адрес в поле ввода **Href** и выбрать новый базовый фрейм в раскрывающемся списке **Target**.

Перезагрузка

Метатег перезагрузки задает период времени, по истечении которого Web-обозреватель перезагрузит текущую страницу. Этот метатег может быть использован на сайтах новостей, причем новостей быстроизменяющихся. Посетитель может открыть страницу с такими "скоропортящимися" новостями и читать их, а Web-обозреватель сам по истечении заданного времени загрузит более "свежие" новости.

Метатег перезагрузки также может применяться для автоматического перенаправления посетителя на другую страницу. Это бывает нужно, когда какой-либо популярный Web-сайт меняет свой адрес. Тогда Web-дизайнер изготавливает простейшую страничку, содержащую текст типа "Сейчас вы окажетесь на нашем сайте..." и гиперссылку для перехода на новый адрес, если автоматическое перенаправление почему-то не сработает. После этого ему остается добавить в секцию HTML-заголовка этой страницы метатег перезагрузки.

Другое применение метатега перезагрузки — создание начальных страничек для сайтов, сделанных на основе фреймового дизайна. Как вы помните, наборы фреймов очень плохо обрабатываются поисковыми агентами. Первый выход из этой ситуации — поместить полный набор ключевых слов в тег комментария или <NOFRAMES>. Второй выход — создать начальную страничку с тем же набором ключевых слов, необходимых метатегов, заставкой и просьбой подождать пару секунд. Web-дизайнеры делают и так и этак, в зависимости от своих личных пристрастий.

Чтобы поместить на страницу метатег перезагрузки, выберите пункт **Refresh** подменю **Head Tags** меню **Insert**. Также вы можете нажать кнопку **Refresh** (рис. 14.11), находящуюся на вкладке **Head** панели объектов. На экране появится диалоговое окно **Refresh**, показанное на рис. 14.12.

Рис. 14.11. Кнопка **Refresh** панели объектов

В поле ввода **Delay** вводится значение задержки перед загрузкой Web-обозревателем другой страницы. Это значение вводится в секундах. Если введен ноль, Web-обозреватель выполняет загрузку немедленно, без всякой задержки.

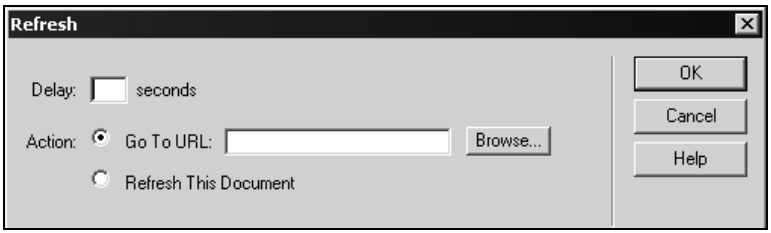


Рис. 14.12. Диалоговое окно **Refresh**

С помощью набора переключателей **Action** задается собственно действие, которое произойдет по истечении заданной задержки. Dreamweaver позволяет выбрать два возможных действия:

- перезагрузка текущей страницы (переключатель **Refresh This Document**);
- перенаправление на другую страницу (переключатель **Go To URL**). Адрес этой страницы вводится в поле ввода **Go To URL**. Вы также можете нажать кнопку **Browse** и выбрать нужную страницу в диалоговом окне **Select File**.

Задав параметры перезагрузки, нажмите кнопку **OK** для создания метатега или кнопку **Cancel** — для отказа от этого. Готовый метатег перезагрузки отображается в панели заголовка значком, показанным на рис. 14.11, т. е. так же, как и кнопка **Refresh** в панели объектов.

Если выбрать в панели заголовка метатег перезагрузки, редактор свойств примет вид, показанный на рис. 14.13. В нем вы можете изменить значение задержки в поле ввода **Delay** и выбрать другое действие с помощью набора переключателей **Action** и поля ввода **URL**.

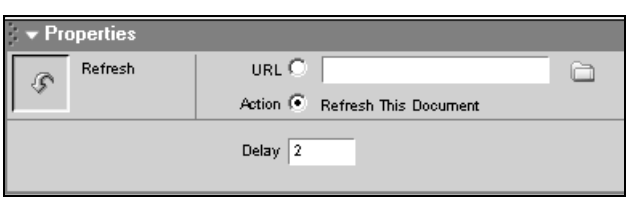


Рис. 14.13. Редактор свойств при выбранном метатеге перезагрузки

Связи между Web-страницами

Метатег, задающий связь между двумя разными Web-страницами или между Web-страницей и другим файлом, используется очень редко и практически всегда — только специальными программами. Обычными Web-обозревателями применяется только одна его форма, которая служит для привязки к странице внешней таблицы стилей. (О таблицах стилей *см. главу 10.*) Другие формы этого метатега не нашли широкого применения.

Однако давайте все же рассмотрим этот метатег. А вы уж сами для себя решите, использовать его или не использовать.

Итак, чтобы поместить на страницу метатег связи, выберите пункт **Link** подменю **Head Tags** меню **Insert**. Также вы можете нажать кнопку **Link** (рис. 14.14), находящуюся на вкладке **Head** панели объектов. На экране появится диалоговое окно **Link**, показанное на рис. 14.15.

Рис. 14.14. Кнопка **Link** панели объектов

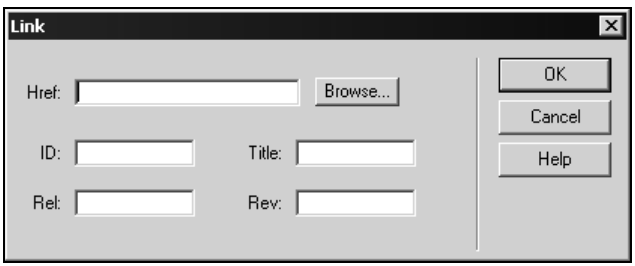


Рис. 14.15. Диалоговое окно **Link**

Адрес Web-страницы (или другого файла, например, таблицы стилей), с которой связывается текущая страница, вводится в поле ввода **Href**. Вы также можете нажать кнопку **Browse** и выбрать нужный файл в диалоговом окне **Select File**.

Поля ввода **ID** и **Title** нужно заполнять только в том случае, если вы собираетесь управлять этим метатегом из Web-сценария. Internet Explorer позволяет такое сделать. Первое из этих полей ввода задает уникальный идентификатор, с помощью которого можно обратиться к метатегу из сценария. Второе задает необязательное название данного метатега.

Внимание!

Каждый тег HTML обязан поддерживать атрибут `TITLE`, с помощью которого задается название этого тега. Это название высвечивается, если пользователь подведет курсор мыши к соответствующему элементу страницы. Однако мало кто заполняет этот атрибут для каждого тега. Да и поддерживается этот атрибут в полном объеме только Internet Explorer.

Вид устанавливаемой связи задается с помощью полей ввода **Rel** и **Rev**. Первое из них задает, чем файл, интернет-адрес которого указан в поле ввода **Href**, является для текущей страницы (например, таблицей стилей). Второе — чем текущая страница является для того файла (например, стра-

ницей-оглавлением). Все возможные значения, которые можно вводить в эти поля ввода, перечислены в табл. 14.2.

Таблица 14.2. *Возможные значения полей ввода* ***Rel*** *и* ***Rev***

Значение	Описание
`Alternate`	Заменяющая страница
`Appendix`	Страница, содержащая приложение к большому многостраничному документу
`Bookmark`	Закладка
`Chapter`	Страница, содержащая отдельную часть большого многостраничного документа
`Contents`	Страница-оглавление
`Copyright`	Страница, содержащая сведения об авторских правах
`Glossary`	Страница со словарем специальных терминов
`Help`	Страница со справочными сведениями
`Index`	Страница-оглавление или список всех страниц сайта (так называемая карта сайта)
`Next`	Следующая страница в последовательности
`Prev`	Предыдущая страница в последовательности
`Section`	Страница, содержащая отдельную часть большого многостраничного документа
`Start`	Первая страница в последовательности
`Stylesheet`	Таблица стилей
`Subsection`	Страница, содержащая отдельную часть большого многостраничного документа, более мелкую, чем `Chapter` или `Section`

Вы также можете ввести в поле ввода **Rel** или **Rev** несколько значений, разделив их пробелами. Введя все данные, нажмите кнопку **OK** для создания метатега или кнопку **Cancel** — для отказа от этого. Готовый метатег связи отображается в панели заголовка значком, изображенным на рис. 14.14, т. е. так же, как и кнопка **Link** панели объектов.

Если выбрать в панели заголовка метатег связи, редактор свойств примет вид, показанный на рис. 14.16. В нем вы можете изменить значение адреса связанного файла в поле ввода **Href**, задать новый тип связи в полях ввода **Rel** и **Rev** и изменить параметры **ID** и **Title**.

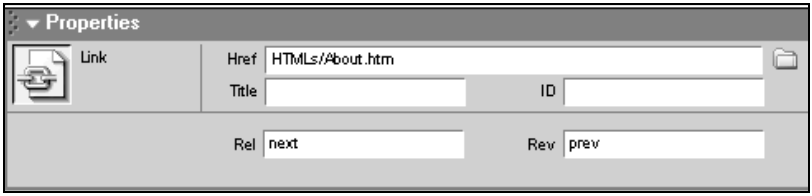


Рис. 14.16. Редактор свойств при выбранном метатеге связи

Как может использоваться описанный выше метатег связи? Например, вы можете задавать связь между отдельными страницами, содержащими части большого документа, и его оглавлением. В этом случае отдельная страница будет помечена как `Chapter` или `Section` (а то и `Subsection`), а оглавление — `Contents` или `Index`. Также первую страницу документа можно пометить как `Start`, последующую — как `Next` и т. д.

Примечание

Метатег задания связи создается с помощью тега `<LINK>`, а не `<META>`.

Специальные метатеги

Метатеги могут содержать любые другие данные. В частности, каждая Web-страница, созданная в Dreamweaver, содержит сведения о текстовой кодировке, помещенные в нее с помощью такого вот специального метатега. Популярный Web-редактор Microsoft FrontPage с помощью специального метатега метит страницы, созданные в нем. Также и сам Microsoft Internet Explorer вставляет в секцию заголовка всех страниц, которые вы сохраняете на диске, особую метку вида "здесь был Internet Explorer". Так что специальные метатеги без работы не остаются.

Чтобы поместить на страницу специальный метатег, выберите пункт **Meta** подменю **Head Tags** меню **Insert**. Также вы можете нажать кнопку **Meta** (рис. 14.17), находящуюся на вкладке **Head** панели объектов. На экране появится диалоговое окно **Meta**, показанное на рис. 14.18.

Рис. 14.17. Кнопка **Meta** панели объектов

С помощью раскрывающегося списка **Attribute** выбирается тип создаваемого метатега. Если он должен содержать описательную информацию о странице, например имя автора или название программы, в которой она была создана, выберите пункт **Name**. Если же эта информация носит системный характер, например задает текстовую кодировку, выберите пункт **HTTP-equivalent**.

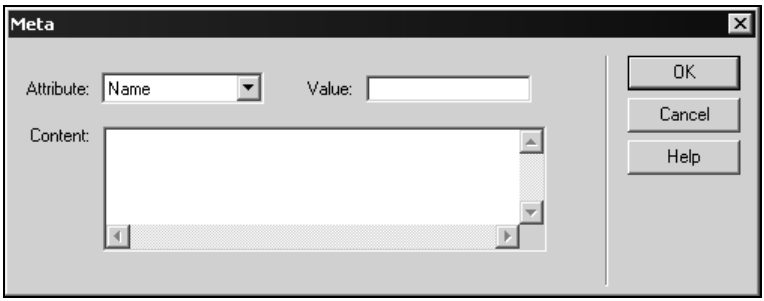


Рис. 14.18. Диалоговое окно **Meta**

Задав тип метатега, можно ввести собственно данные. В поле ввода **Value** вводится наименование информации, например "Creator", если вы хотите ввести название программы, в которой была создана страница. Сами же данные вводятся в область редактирования **Content** (в нашем случае — "Macromedia Dreamweaver").

Введя все данные, нажмите кнопку **OK** для создания метатега или кнопку **Cancel** — для отказа от этого.

Если при создании специального метатега в раскрывающемся списке **Attribute** был выбран пункт **Name**, то этот метатег будет отображаться в панели заголовка значком, показанным на рис. 14.17, т. е. так же, как и кнопка **Meta** в панели объектов. Если же был выбран пункт **HTTP-equivalent**, значок будет таким, как на рис. 14.19.

Рис. 14.19. Значок, которым в панели заголовка отображается метатег типа **HTTP-equivalent**

Если выбрать в панели заголовка специальный метатег, редактор свойств примет вид, показанный на рис. 14.20. В нем вы можете изменить тип метатега с помощью раскрывающегося списка **Attribute**, задать новое наименование информации в поле ввода **Value** и изменить саму информацию в поле ввода **Content**.

Вот и все о метатегах.

Рис. 14.20. Редактор свойств при выбранном специальном метатеге

Серверные директивы

Теперь пришла пора поговорить о серверных директивах.

Введение в серверные директивы

Весь HTML-код, из которого состоит Web-страница, пересылаемая по Сети, обрабатывается клиентом. Клиентом практически всегда является Web-обозреватель (существуют и другие программы, обрабатывающие Web-страницы на стороне клиента, но мы не будем их рассматривать). Именно для Web-обозревателя предназначены все эти теги, что мы создаем вручную или в Web-редакторе. Web-сервер же их совсем не понимает.

Задача Web-сервера совсем другая: принять запрос от клиента, найти нужный файл на своих дисках и переслать его клиенту для обработки. Ну и, разумеется, он должен устойчиво работать и потреблять как можно меньше системных ресурсов. Обработка HTML-кода в его обязанности не входит.

Однако разработчики Web-серверов все же заставили их понимать некоторые команды, с помощью которых можно обработать HTML-код. Как вы уже поняли, эта обработка выполняется на стороне сервера, более того, самим Web-сервером. Конечно, эти команды позволяют выполнить самые простые действия: вставить в содержимое одного файла содержимое другого, поместить в код текущую дату и т. п. Но даже в этом случае они — хорошее подспорье Web-дизайнеру.

Эти команды и называются *серверными директивами*. Несмотря на всю их простоту, они позволяют решить много проблем, неустранимых иными средствами.

Помните, как мы пытались поместить на Web-страницу текущую дату? Для этого нам пришлось в свое время писать Web-сценарий. А если Web-обозреватель не поддерживает Web-сценарии? А если компьютер, на котором он работает, настолько маломощен, что пользователь отключил поддержку сценариев? В таком случае сценарий, помещающий на страницу дату, не выполнится.

Web-сервер же работает всегда, и серверная директива, осуществляющая вставку в HTML-код текущей даты, выполнится в любом случае. Web-обозреватель же, загрузивший страницу, даже не будет знать, что эта страница была предварительно обработана сервером. (Вообще-то, об этом догадаться может если не Web-обозреватель, то опытный пользователь, имеющий представление о Web-серверах и их настройке.)

Так же трудно вставить HTML-код, находящийся в одном файле, в другой файл средствами клиентских Web-сценариев. Web-сервер же с помощью специальной директивы делает это с легкостью. Таким образом, можно создавать Web-страницы, "собираемые" на стороне сервера из множества час-

тей: заголовка, полосы навигации, сведений об авторских правах и, разумеется, основного содержимого. Это идеальное решение многих проблем сайтов, созданных на основе табличного дизайна. Далее об этом будет рассказано подробнее.

К несчастью, как уже было сказано, серверные директивы никем не стандартизировались. Каждый Web-сервер поддерживает свой набор этих директив, а каких — зависит от доброй воли его создателя. Правда, существует некий набор директив, поддерживаемых большинством серверов, этакий неписаный стандарт "де-факто". Применяя в своих страницах директивы из этого набора, вы можете быть уверены, что они с большой долей вероятности будут поддерживаться сервером, на котором вы собираетесь их (страницы) разместить.

А пока что давайте рассмотрим серверные директивы, поддерживаемые подавляющим большинством Web-серверов, за исключением самых уж примитивных. Назовем их *стандартным набором серверных директив*, хотя их фактически никто не стандартизировал. Но все равно перед тем, как применять серверные директивы, обратитесь к документации на Web-сервер или к его администратору, чтобы выяснить, какие директивы ваш сервер поддерживает.

Стандартный набор серверных директив

Прежде всего, давайте рассмотрим, как серверные директивы вставляются в HTML-код. Проблема заключается в том, что их нужно как-то выделить, чтобы Web-сервер сразу их "увидел", а Web-обозреватель, наоборот, "не заметил". Поэтому серверные директивы помещаются в тег комментария и помечаются значком # ("решетка"):

```
<!-- #<Директива> <Атрибут1>=<Значение1> <Атрибут2>=<Значение2> ... -->
```

Однако, кроме этого, надо выполнить еще кое-какие действия. Чтобы Web-сервер "знал", что в какой-либо странице используются серверные директивы, нужно изменить расширение файла, в котором сохранена эта страница. Как вы помните, обычно HTML-файлы имеют расширение htm или html. Так вот: эти расширения нужно изменить соответственно на shtm и shtml.

Подавляющее большинство Web-серверов требуют, чтобы Web-страницы с серверными директивами находились в файлах с расширениями shtm и shtml. Правда, встречаются серверы с другими требованиями (например, малоизвестный Web-сервер со странным названием "π^3" требует, чтобы такие файлы имели расширение ssi), но подобные случаи весьма редки. И все же, невредно было бы прежде, чем применять в своих страницах серверные директивы, прочитать документацию к серверу или проконсультироваться у его администратора.

Как видите, серверная директива может иметь (и практически всегда имеет) несколько атрибутов, которым присвоены значения. В этом смысле серверные директивы чем-то похожи на теги HTML.

include

Директива `include` вставляет в содержимое текущего файла содержимое другого. (Эта директива также называется *серверным включением*, а файл, содержащий включаемый фрагмент кода, — *файлом включения*). Вставка осуществляется в то место файла, где встретилась эта директива. Записывается она так:

```
<!-- #include file="<Имя файла>" -->
```

или так:

```
<!-- #include virtual="<Имя файла>" -->
```

В первом случае `<Имя файла>` представляет собой обычное имя файла, заданное с учетом особенностей файловой системы серверного компьютера. Например, так:

```
<!-- #include file="c:\Inetpub\wwwroot\includes\header.inc" -->
```

В данном случае в текущий файл вставляется содержимое файла header.inc, находящегося в папке Inetpub\wwwroot\includes на диске C.

Во втором случае файл `<Имя файла>` ищется относительно корневой папки сайта. Также он может находиться на другом Web-сайте; в этом случае в качестве значения атрибута `virtual` указывается полный интернет-адрес этого файла.

```
<!-- #include virtual="/htmls/includes/header.inc" -->
```

В данном случае файл header.inc располагается в папке htmls/includes, вложенной в корневую папку сайта.

```
<!-- #include virtual="http://www.othersite.ru/includes/header.inc" -->
```

А в этом случае файл header.inc находится вообще на другом сайте.

Существует негласное правило: все файлы включений должны иметь расширение inc (от англ. *include* — включение). Хотя не возбраняются другие расширения. Также, если у вас имеется достаточно много файлов включений, лучше всего поместить их в особую папку, например includes. В противном случае можно оставить их в той же папке, где находятся все Web-страницы.

Внимание!

Не все Web-серверы поддерживают обе разновидности директивы `include`. Например, Microsoft Internet Information Server поддерживает только форму с атрибутом `file`.

echo

Директива `echo` помещает в HTML-код содержимое одной из *встроенных переменных* Web-сервера. Она имеет такой синтаксис:

```
<!-- #echo var="<Имя переменной>" -->
```

В качестве значения атрибута `var` задается имя нужной переменной. Встроенные переменные, поддерживаемые большинством Web-серверов, перечислены в табл. 14.3.

Таблица 14.3. *Встроенные переменные Web-сервера*

Переменная	Описание
`DOCUMENT_NAME`	Возвращает имя файла, содержащего текущую Web-страницу
`DOCUMENT_URI`	Возвращает путь к файлу, содержащему текущую Web-страницу, относительно корневой папки сайта
`DATE_LOCAL`	Возвращает текущую дату для местной временной зоны
`DATE_GMT`	Возвращает текущую дату по Гринвичу
`LAST_MODIFIED`	Возвращает дату последнего изменения файла текущей Web-страницы для местной временной зоны
`LAST_MODDATE`	Возвращает дату последнего изменения файла текущей Web-страницы по Гринвичу

В частности, именно с помощью данной серверной директивы на страницу помещается текущая дата. Для этого, как вы поняли, нужно использовать директиву:

```
<!-- #echo var="DATE_LOCAL" -->
```

или

```
<!-- #echo var="DATE_GMT" -->
```

если нужно вывести дату по Гринвичу.

Также многие Web-серверы поддерживают другие серверные переменные. Чтобы выяснить их, обратитесь к документации по серверу или к его администратору.

Как использовать серверные директивы

А сейчас давайте поговорим, как можно использовать серверные директивы.

Ну, насчет директивы `echo` все ясно. Используйте ее, если на страницу нужно поместить текущую дату или имя файла страницы. Для этого доста-

точно подставить в качестве значения атрибута `var` имя соответствующей переменной.

А вот как можно использовать директиву `include`?

Для разделения кода Web-страниц на отдельные части.

Зачем это нужно?

Необходимо это во многих случаях. Но нужнее всего, если сайт построен по принципам табличного дизайна, либо странички содержат какие-либо повторяющиеся стандартные элементы. В этом случае умелое разделение кода на фрагменты принесет истинное облегчение Web-дизайнеру. Даже если он использует Dreamweaver.

Давайте вернемся назад и вспомним, чем отличаются Web-страницы, построенные на основе табличного дизайна, от прочих. Как вы помните, все, что находится на такой странице, представляет собой содержимое одной огромной таблицы, занимающей всю эту страницу. А какие недостатки таблиц мы знаем?

Недостаток первый: очень медленная загрузка. Web-обозреватель не может вывести на экран таблицу, пока не загрузит ее целиком. А если таблица довольно велика, а канал связи с Интернетом достаточно нетороплив, загрузка может продолжаться очень долго.

Недостаток второй: большой объем получающегося HTML-кода. В самом деле, вспомните, какие размеры были у страниц, созданных на основе табличного дизайна! Сравните их со страничками сайта, основанного на фреймах! Отчасти это происходит из-за того, что сам код, создающий таблицу, очень велик — это плата за гибкость. Но основная причина этого другая: при использовании табличного дизайна каждая страница включает в себя все повторяющиеся элементы (заголовок, полоса навигации, сведения об авторских правах), которые во втором случае "вынесены" в отдельные фреймы, т. е. размер HTML-кода еще больше увеличивается.

Недостаток третий, свойственный не самим таблицам, а табличному дизайну: каждая страница содержит все стандартные, повторяющиеся элементы оформления. А теперь представьте, что вам нужно слегка изменить стандартное примечание, а страниц в вашем сайте добрая сотня. Что делать? Открывать каждую из них и вручную править текст? А если вы пропустите какую-нибудь из них? А если забудете сохранить при закрытии? Уже не стоит говорить о том, что перелопатить вручную такое количество страниц трудно чисто физически.

Dreamweaver пытается решить эту проблему, предоставляя уже известные вам шаблоны. Но это не решение проблемы, а половина решения. Да, вы можете внести изменения в шаблон, после чего Dreamweaver за вас будет переносить эти изменения во все страницы сайта. Но это не устраняет проблему увеличения размеров страниц. Страницы, созданные на основе таб-

личного дизайна, все же остаются большими, занимая место на дисках сервера, которого всегда не хватает. Более того, размер страниц увеличивается еще за счет разного рода специальных комментариев, вносимых самим Dreamweaver, чтобы выделить HTML-код шаблона.

Так как же решить эту проблему? И есть ли у нее решение?

Есть.

Решение в том, чтобы вынести повторяющиеся элементы страниц в отдельные файлы, оставив в них только основное содержимое, т. е. выделить в HTML-коде страницы фрагменты, создающие ее заголовок, полосу навигации и сведения об авторских правах, вырезать их и сохранить в других файлах. А на их место поместить серверные директивы `include`, указывающие на эти файлы.

Поясним вышесказанное на примере. Давайте для примера создадим страничку, код которой приведен ниже.

```
<HTML>
  <HEAD>
    <TITLE>Страница</TITLE>
  </HEAD>
  <BODY>
    <TABLE BORDER="1">
      <TR>
        <TD><A HREF="page1.htm">Страница 1</A></TD>
        <TD ROWSPAN="4">Это основное содержимое страницы.</TD>
      </TR>
      <TR>
        <TD><A HREF="page2.htm">Страница 2</A></TD>
      </TR>
      <TR>
        <TD><A HREF="page3.htm">Страница 3</A></TD>
      </TR>
      <TR>
        <TD><A HREF="page4.htm">Страница 4</A></TD>
      </TR>
    </TABLE>
  </BODY>
</HTML>
```

Сохраните этот код в файле 14.1.htm и откройте его в Web-обозревателе. Вы увидите нечто, похожее на рис. 14.21.

Эта страница сделана на основе табличного дизайна. Значит, она содержит все повторяющиеся элементы. Давайте вынесем их в другие файлы.

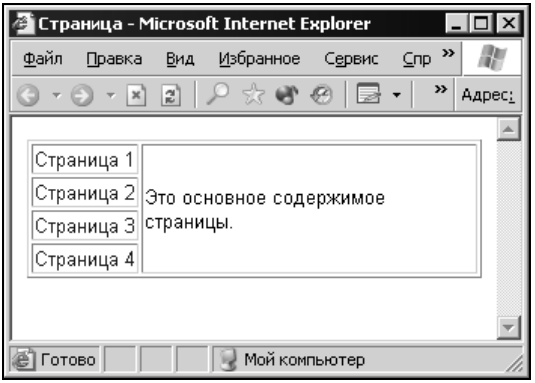


Рис. 14.21. Страница-пример, сделанная без использования серверных директив

Сначала нужно выяснить, где находится основное содержимое страницы. В нашем случае это строка: "Это основное содержимое страницы.". Все остальное — повторяющиеся элементы.

Наш случай — самый простой. Мы вынесем все, что находится от тега <BODY> до приведенной выше строки, в файл 14.1.inc, а все, что расположено после нее и до тега </BODY>, — в файл 14.2.inc. А саму страницу переименуем в 14.2.shtm. После всех этих пертурбаций ее код будет выглядеть так:

```
<HTML>
  <HEAD>
    <TITLE>Страница</TITLE>
  </HEAD>
  <BODY>
    <!-- #include file="14.1.inc" -->
    Это основное содержимое страницы.
    <!-- #include file="14.2.inc" -->
  </BODY>
</HTML>
```

Как видите, от изначального HTML-кода страницы осталось не так уж и много. А теперь вообразите, что мы проделали такую операцию со всеми страницами большого и сложного сайта. Представляете, насколько уменьшатся все его файлы!

Тогда файл 14.1.inc будет выглядеть так:

```
<TABLE BORDER="1">
  <TR>
    <TD><A HREF="page1.htm">Страница 1</A></TD>
    <TD ROWSPAN="4">
```

А файл 14.2.inc — так:

```
    </TD>
  </TR>
  <TR>
    <TD><A HREF="page2.htm">Страница 2</A></TD>
  </TR>
  <TR>
    <TD><A HREF="page3.htm">Страница 3</A></TD>
  </TR>
  <TR>
    <TD><A HREF="page4.htm">Страница 4</A></TD>
  </TR>
</TABLE>
```

К сожалению, просмотреть страницы этого "дивного нового" сайта в Web-обозревателе мы не сможем. Ни один Web-обозреватель не понимает серверных директив, как ни один Web-сервер не понимает HTML-тегов. Это не его обязанности. Чтобы увидеть, как Web-сервер обработает серверные директивы, нам понадобится сам Web-сервер. А его у нас нет.

Поддержка Dreamweaver серверных директив

К сожалению, поддержка серверных включений Dreamweaver MX оставляет желать лучшего.

Разработчики Dreamweaver утверждают, что их программа читает файлы включений, расшифровывает их и выводит в окно документа вместе содержимым страницы, использующей эти включения. Так-то оно так, но не совсем. Dreamweaver действительно выводит содержимое серверных включений в окне документа, но некорректно обрабатывает HTML-разметку: он "проглатывает" весь HTML-код, находящийся в серверных включениях, и выводит только текст. Посмотрите, что он сделал с нашей страницей 14.2.shtm (рис. 14.22).

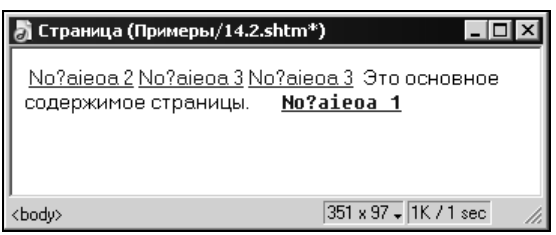


Рис. 14.22. Web-страницы, содержащие серверные включения, Dreamweaver отображает некорректно

Иначе, как грубейшей ошибкой, назвать это нельзя. Ведь автор прекрасно помнит, что третья версия Dreamweaver (3.0) нормально отображала содержимое файлов включений. Досадно, что новая версия, вместе с новыми возможностями, таит в себе новые ошибки.

Однако не все так плохо. Вы можете отключить вывод содержимого серверных включений; в этом случае они будут просто помечаться особым значком (рис. 14.23). Для этого откройте окно **Preferences**, выбрав пункт **Preferences** в меню **Edit**, и переключитесь на вкладку **Invisible Elements**. Отключите флажок **Server-Side Includes** и нажмите кнопку **OK**.

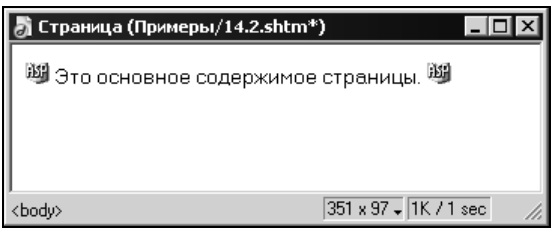


Рис. 14.23. Серверные включения, отображаемые в виде значков

На этом поддержка серверных директив `include` Dreamweaver не заканчивается. Вы можете вставить серверное включение в код страницы, выбрав пункт **Server-Side Include** подменю **Script Objects** меню **Insert** или нажав кнопку **Server-Side Include** (рис. 14.24), находящуюся на вкладке **Script** панели объектов. После этого вам останется только выбрать нужный файл в появившемся на экране диалоговом окне **Select File**.

Рис. 14.24. Кнопка **Server-Side Include** панели объектов

Если вы выберете значок серверной директивы `include`, редактор свойств примет вид, представленный на рис. 14.25. С его помощью вы сможете изменить значения атрибутов этой директивы.

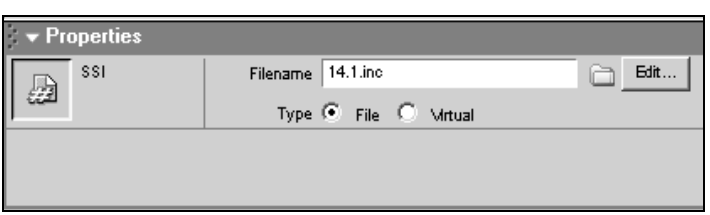


Рис. 14.25. Вид редактора свойств при выбранной серверной директиве `include`

Разновидность директивы `include` — с атрибутом `file` или `virtual` — задается с помощью набора переключателей **Type**. Вам необходимо будет выбрать нужный переключатель: **File** или **Virtual**.

Само имя файла задается в поле ввода **Filename**. Вы также можете нажать значок папки, расположенный справа этого поля ввода, и выбрать нужный файл в появившемся на экране диалоговом окне **Select File**.

Нажав кнопку **Edit**, вы сможете отредактировать файл серверного включения. Dreamweaver отобразит его в окне документа в виде HTML-кода.

Да, Dreamweaver очень странно поддерживает серверные директивы. Будем надеяться, что в следующей его версии этот недостаток будет устранен. А пока что можем только посоветовать изучать язык HTML и править код таких вот разделенных на части страниц вручную.

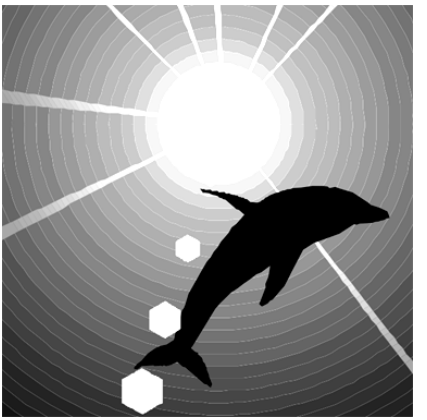
На этом рассказ о серверных директивах окончен.

Что дальше?

На этом мы закончим разговор о Web-дизайне и начнем говорить о Web-программировании. Удивлены? Скажете, что мы вроде бы уже говорили о программировании в Интернете в *главе 13*? Да, говорили.

Те программы, которые мы писали ранее, работали на стороне клиента, под управлением Web-обозревателя и виртуальной машины. Но знаете ли вы, что можно писать программы, которые будут работать на стороне Web-сервера? Эти программы будут принимать от посетителя сайта какие-либо данные, обрабатывать их и выдавать результат в виде сформированной самой программой Web-страницы.

Заинтригованы? Тогда читайте дальше.



ЧАСТЬ IV

Пишем серверные программы

Глава 15. Введение в серверное программирование

Глава 16. Формы

Глава 17. Простейшие серверные приложения

Глава 18. Создание интерактивных сайтов

ГЛАВА 15

Введение в серверное программирование

Что? Какое еще серверное программирование? Что это за беда? И зачем она нам нужна?

Мы, вроде бы, научились создавать Web-страницы в среде Dreamweaver. Мы даже научились создавать с его помощью целые Web-сайты и публиковать их на Web-сервере. Мы изучили две разновидности дизайна страниц: фреймовый, когда сайты строятся на основе наборов фреймов, и табличный, когда содержимое страницы помещается в большую сложную таблицу. Мы познакомились с таблицами стилей, метатегами и серверными директивами. И, наконец, узнали о Web-программировании и Web-сценариях, позволяющих добавить "жизни" нашим статичным страничкам. Что же еще надо для счастья?

Да, изученного ранее нам вполне хватит, чтобы создавать вполне приличные сайты. Многие Web-дизайнеры на этом и останавливаются. Но ведь мы хотим большего, не так ли?

Так давайте же сделаем следующий шаг — перейдем от страниц, хранящихся в файлах на сервере, к страницам, генерируемым специальными программами. Как раз написанием таких программ и занимается серверное программирование.

Но давайте по порядку. И начнем мы с того, что выясним, зачем нужны эти серверные программы.

Что такое серверное программирование

Действительно, что это такое и с чем его едят?

Зачем нужны серверные программы

Вы когда-нибудь посещали интернет-магазин? Например, популярнейший "Озон" (**http://www.ozon.ru**). Помните, как там выполняется заказ товара?

Если не помните или вообще не знаете, что такое интернет-магазин, давайте вспомним (или узнаем).

Вы заходите на Web-страничку, описывающую нужный вам товар. После долгих мук совести вы все-таки решаетесь на покупку и щелкаете на кнопке **Купить**. После этого вы получаете несколько Web-страниц, где можете задать свой адрес, способы оплаты и доставки товара и, наконец, подтвердить покупку. Все это вы делаете, щелкая соответствующие кнопки и вводя данные в соответствующие поля ввода, расположенные прямо на страницах.

Что происходит при этом? Как обрабатываются введенные вами данные? Неужели самим Web-обозревателем?

Отнюдь. Эти данные обрабатываются на Web-сервере.

Интернет-магазин — просто один из примеров, пришедших в голову автору, являющемуся поклонником и постоянным клиентом вышеупомянутого "Озона". Точно так же работают серверы электронной почты, основанной на Web, поисковые машины, электронные доски объявлений, форумы, вообще, любые Web-сайты, принимающие от посетителя какие-то данные и обрабатывающие их. Во всех этих случаях Web-обозреватель принимает от посетителя данные и отправляет их Web-серверу, который обрабатывает их и выдает результат обработки в виде автоматически сформированной Web-страницы.

Как это происходит на деле? Сейчас мы это выясним. И первым делом ответим на вопрос...

Как Web-сервер обрабатывает данные пользователя

Итак, каким же образом программа Web-сервера обрабатывает данные, отправленные ей пользователем?

Да никак. Web-сервер не приспособлен их обрабатывать. Его задача: прием от Web-обозревателя запроса на файлы (Web-страницы, таблицы стилей, графические изображения, фильмы, звуки, архивы, исполняемые файлы и т. п.), поиск этих самых файлов на жестких дисках серверного компьютера и отправка найденных файлов назад Web-обозревателю. Это его основная задача. Конечно, некоторые особо мощные серверы могут выполнять дополнительные действия над отправляемыми файлами перед собственно их отправкой (в частности, выполнять серверные директивы). Есть и программы-"многостаночники", выполняющие функции не только Web-сервера, но и сервера FTP, почты, новостей UseNet и бог знает чего еще. Но основная функция: простая выдача файлов по требованиям клиентов — и не более того.

Секрет в том, что данные посетителя обрабатываются не самим Web-сервером. Для этого применяются специальные программы, работающие

вместе с Web-сервером на том же серверном компьютере. Они называются *серверными программами*, не имеют интерфейса пользователя и "общаются" только с Web-сервером, принимают от него введенные пользователем данные и возвращают ему результат. Этим они коренным образом отличаются от клиентских программ, работающих непосредственно с пользователем. (К клиентским программам относится, в частности, ваш любимый Web-обозреватель.)

Из этого следует, что Web-сервер умеет-таки принять данные от пользователя. Да, это входит в его основную задачу: принять данные и перенаправить их серверной программе. В свою очередь серверная программа их обработает и вернет Web-серверу результат.

Вот тут-то и начинается самое интересное. Дело в том, что результат, возвращаемый серверной программой Web-серверу, — это не что иное, как обычный HTML-код! Фактически серверная программа возвращает готовую Web-страницу, сформированную на основе данных, введенных посетителем. Такая страница называется *динамической*, в отличие от *статических* страниц, написанных Web-дизайнером и сохраненных в файлах на дисках серверного компьютера. А уж эту динамическую страницу Web-сервер и направляет клиенту в качестве ответа на введенные данные.

Серверные программы делятся на следующие четыре вида.

1. Исполняемые программы, работающие через интерфейс *CGI* (Common Gateway Interface — общий интерфейс обмена), так называемые *CGI-программы*. Эта разновидность серверных программ — самая старая, однако отнюдь не устаревшая.
2. *Расширения Web-сервера* (приложения формата ISAPI, NSAPI, модули расширения Apache и т. п.). Новый способ, позволяющий встраивать серверные программы в сам Web-сервер, делая их его составными частями. Впервые предложен фирмой Microsoft для их сервера Microsoft Internet Information Server (интерфейс ISAPI) и разработчиками популярного бесплатного Web-сервера Apache.
3. *Активные серверные страницы* (ASP, JSP и др.). Фактически это обычные статические Web-страницы, сохраненные в файлах, которые, кроме обычного HTML-кода, включают в себя команды, обрабатываемые либо самим Web-сервером, либо его расширением. Также новый способ, впервые предложенный Microsoft для того же Internet Information Server.
4. *Серверные сценарии*, написанные на интерпретируемом языке (Perl, Python, VBScript, JavaScript и др.). Обычные сценарии, работающие через интерфейс CGI или ISAPI на стороне сервера.

Теперь рассмотрим все это разнообразие подробнее.

CGI-программы представляют собой обычные исполняемые файлы, написанные на любом языке программирования и откомпилированные в ма-

шинный код процессора. Они не имеют интерфейса пользователя (как и все серверные программы), а работают с Web-сервером, получают от него входные данные и ему же пересылают результаты своей работы. Запускаются они самим Web-сервером, когда в них возникает нужда (когда необходимо обработать полученные от пользователя данные), и работают под управлением операционной системы серверного компьютера. При этом, если Web-серверу поступает одновременно несколько запросов на обработку данных от пользователей, он запускает соответствующее количество копий CGI-программы.

К достоинствам CGI-программ можно отнести легкость создания (многие среды разработки программ поддерживают создание таких приложений, в частности популярнейший Borland Delphi, начиная с версии 3) и простоту отладки. Также, поскольку CGI-приложения представляют собой независимые программы, они выполняются отдельно от Web-сервера (как говорят программисты и системные администраторы, выполняются в другом адресном пространстве). Это значит, что при сбое в CGI-программе завершается только она — сам Web-сервер остается "на плаву". А недостаток у CGI-программ всего один: большой расход системных ресурсов, поскольку для обработки каждого набора данных запускается отдельная копия серверной программы. И если Web-серверу поступит слишком много запросов на обработку данных, серверный компьютер может и зависнуть.

Расширения Web-сервера — более новая разновидность серверных программ. Они представляют собой обычные библиотеки DLL, в которых реализована вся логика серверной программы. Такие библиотеки как бы встраиваются в программу Web-сервера и работают как ее неотъемлемая часть. Поскольку библиотеки DLL работают только в среде Windows, для того чтобы создавать расширения в иных операционных системах, были придуманы и другие форматы. В частности, модули расширения сервера Apache не являются библиотеками DLL.

Именно в виде библиотек DLL создаются расширения Web-серверов Internet Information Server фирмы Microsoft и Netscape Web Server фирмы Netscape. В первом случае расширения имеют формат *ISAPI* (Internet Server Application Programming Interface — интерфейс программирования приложений интернет-сервера), а во втором — *NSAPI* (Netscape Server Application Programming Interface — интерфейс программирования приложений сервера Netscape). Формат модулей расширения Apache так и называется — модули Apache.

Достоинство у расширений Web-сервера одно: бережный расход системных ресурсов. Дело в том, что для обработки всех наборов данных пользователя запускается всего один экземпляр расширения, который отнимает существенно меньше ресурсов, чем уйма запущенных CGI-программ. Однако расширения труднее создавать и отлаживать, к тому же они не так безопасны,

как CGI-программы. Поскольку они работают как часть Web-сервера, любая ошибка в расширении приведет к зависанию сервера.

Оба описанных выше вида серверных программ обладают одним огромным недостатком. Прежде чем они смогут работать, они должны быть написаны на языке программирования и откомпилированы в машинные коды процессора, что отнимает много времени, особенно при отладке. Конечно, откомпилированные программы работают быстрее интерпретируемых, т. е. тех, где каждая инструкция читается, расшифровывается и обрабатывается специальной программой-интерпретатором. Но у интерпретируемых программ есть и свои преимущества, главными из которых являются простота и быстрота написания. Две следующие разновидности серверных программ, которые будут описаны, как раз будут интерпретируемыми.

Как уже говорилось, активные серверные страницы — это обычные Web-страницы, включающие в себя особые *серверные сценарии*, выполняемые самим Web-сервером или специальной серверной программой (CGI-приложением или расширением Web-сервера). В частности, *ASP* (Active Server Pages — активные серверные страницы), поддерживаемые Microsoft Internet Information Server, и *JSP* (Java Server Pages — серверные страницы, написанные на JavaScript), поддерживаемые рядом других Web-серверов, работают именно таким образом. Серверные страницы ASP пишутся на языках JavaScript и VBScript, а JSP — только на JavaScript.

Достоинства активных серверных страниц вы уже знаете: легкость и быстрота написания и простота отладки. Кроме того, поскольку активные серверные страницы — это обычные Web-страницы с "вкраплениями" программного кода, их написание легко освоят все, кто знаком с HTML. Недостаток: относительная медлительность и повышенные требования к системным ресурсам.

Серверные сценарии подобны активным серверным страницам тем, что являются интерпретируемыми, однако представляют собой "чистый" программный код, без HTML-"примесей". Интерпретатор практически всегда представляет собой CGI-программу, однако ничто не мешает разработать его в виде расширения Web-сервера. Сценарии обычно пишутся на языке программирования Perl, специально предназначенном для обработки текста; также используются языки Python, JavaScript, VBScript и даже (как говорят) язык командных файлов MS-DOS. Фактически писать сценарии можно на любом языке программирования, для которого есть интерпретатор.

Достоинства и недостатки серверных сценариев те же, что у активных серверных страниц. Однако сценарии потребляют исключительно много системных ресурсов, даже больше, чем CGI-приложения. Ведь для обработки каждого набора данных пользователя запускается своя копия интерпретатора, а интерпретатор, в свою очередь, расходует много ресурсов на обработку сценария. И все же, несмотря на это, сценарии — самый популярный способ создания серверных программ.

В табл. 15.1 приведены расширения файлов серверных программ.

Таблица 15.1. *Расширения файлов серверных программ*

Вид серверных программ	Подвид	Расширение
CGI-программы	—	exe
Расширения Web-сервера	ISAPI и NSAPI	dll
	Модули Apache	Нет расширения
Активные серверные страницы	ASP	asp
	JSP	jsp
Серверные сценарии	Язык Perl	pl, cgi
	JavaScript	js, cgi
	VBScript	vbs, cgi
	Другие языки	cgi

Вот мы и рассмотрели, как Web-сервер обрабатывает (точнее, не обрабатывает) данные пользователя. Теперь переместимся в начало цепочки и рассмотрим, как Web-обозреватель отправляет данные пользователя серверной программе.

Как Web-обозреватель отправляет введенные данные

В начале этой главы мы говорили о том, что для сбора данных посетителя используются элементы управления, помещаемые на самих Web-страницах. Это обычные элементы управления, знакомые вам по приложениям Windows: поля ввода, кнопки, списки, флажки и пр. Посетитель сайта вводит в них данные и нажимает особую кнопку, запускающую отправку данных Web-серверу, а значит, и серверной программе.

Элементы управления, предназначенные для ввода данных посетителя, помещаются в форму. *Форма* — это особый элемент страницы, выполняющий собственно кодирование данных и пересылку их Web-серверу. (Можно сказать, что форма является родителем для элементов управления.) Сами же элементы управления только принимают данные от посетителя, но не кодируют и не передают их.

Каждый элемент управления, находящийся в форме, должен иметь уникальное имя. Эти имена используются Web-обозревателем для того, чтобы представить введенные в форму данные в удобочитаемом для серверной

программы виде — в виде набора пар вида "имя элемента управления" = "значение". Например:

```
name1 = Ivan
surname = Ivanovich
name2 = Ivanov
age = 30
```

Здесь мы рассмотрели идеальный случай, когда значения каждого элемента управления содержат только *допустимые* с точки зрения протокола HTTP символы: буквы латинского алфавита, цифры, тире, подчеркивания и некоторые другие знаки. (Как вы помните, HTTP — протокол передачи файлов, используемый Web-сервером.) Если же вы передаете данные, содержащие недопустимые символы, скажем, пробелы или буквы русского алфавита, каждый такой символ будет представлен в виде шестнадцатеричного кода, которому предшествует знак процента, например, так (закодированные символы пробела выделены полужирным шрифтом):

```
fullname=Ivan%20Ivanovich%20Ivanov
age=30
```

Благодаря такому формату данных написание обрабатывающих их серверных программ становится очень простым. В частности, язык Perl имеет встроенные средства расшифровки для этого формата.

Представленные в вышеуказанном виде данные впоследствии кодируются с использованием одного из предопределенных *методов кодирования* и отправляются по Сети серверной программе. Всем этим фактически занимается форма (но не элементы управления).

Сам процесс отправки данных начинается после того, как пользователь нажмет особую кнопку. Эта кнопка носит название **Отправить** (Submit — в англоязычных программах) и обязательно должна присутствовать в форме. Также в форме может присутствовать кнопка **Сброс** (Reset), обнуляющая введенные пользователем данные. Как правило, эти кнопки располагаются в самом низу формы.

Вы можете считать, что форма — это некое подобие обычного диалогового окна Windows-приложения, принимающего данные от пользователя, кодирующего их определенным образом и отсылающего основному окну. (Диалоговые окна Windows-приложений также обязательно имеют две кнопки: **OK** и **Отмена** (Cancel).) Но если в случае обычного Windows-приложения программист должен явно задать, как будут шифроваться и пересылаться данные, в случае с Web-формой этого делать не нужно. Вам необходимо будет только задать три обязательных параметра:

- интернет-адрес серверной программы, которая будет обрабатывать данные формы;

- метод кодирования отправляемых данных;
- один из двух методов отправки данных.

О методах отправки данных мы поговорим чуть позже. А сейчас выясним все о двух других обязательных параметрах формы.

Интернет-адрес серверной программы очень похож на интернет-адрес любого другого файла, например Web-страницы. Взгляните сами — так будет выглядеть адрес CGI-программы:

```
http://www.somesite.ru/bin/program.exe
```

Так — адрес расширения Web-сервера:

```
http://www.somesite.ru/bin/extension.dll
```

Это — адрес активной серверной страницы:

```
http://www.somesite.ru/asps/active_page.asp
```

А это — адрес программы-сценария, написанной на языке Perl:

```
http://www.somesite.ru/scripts/perl_script.pl
```

Как видите, ничего сложного в этом нет. Серверная программа — обычный файл, помещенный на жестких дисках серверного компьютера, и ссылка на него также не представляет ничего особенного.

Для кодирования передаваемых по Сети данных обычно используются три наиболее популярных метода: `application/x-www-form-urlencoded`, `multipart/form-data` и (значительно реже) `text/plain`. Возможно применение и других способов кодирования, но в абсолютном большинстве случаев используются три перечисленных. Вдобавок эти три метода кодирования поддерживаются большинством программ Web-обозревателей.

Примечание

Если вы еще помните, что такое тип данных MIME, то, взглянув на приведенные выше названия методов кодирования данных, сразу увидите, что это как раз типы MIME. Именно с их помощью задаются методы кодирования.

В подавляющем большинстве случаев используется метод кодирования `application/x-www-form-urlencoded`. Кстати, именно он применяется по умолчанию, если метод кодирования не задан. Метод кодирования `multipart/form-data` используется, если вы собираетесь отправить на Web-сервер файлы; он обеспечивает соответствующее такому случаю преобразование двоичных данных. Последний метод — `text/plain` — представляет данные в виде обычного текста, что может быть полезно, если данные формы будут отправляться по электронной почте (иногда применяется и такой способ передачи данных).

Итак, с кодированием данных разобрались. Остается выяснить, как же эти данные передаются по каналам Сети.

Как данные передаются по Сети

Как вы уже знаете, для пересылки данных по Интернету, да и по любой локальной или глобальной компьютерной сети, используется особый набор правил, называемый *протоколом*. Протокол определяет, каким образом данные будут шифроваться и упаковываться для последующей передачи по сети. Естественно, что и передающая, и принимающая программы должны поддерживать один и тот же протокол, чтобы "понять" друг друга. (Иначе возникнет так называемая несовместимость по протоколу передачи данных, штука весьма неприятная.) Собственно, уже говорилось об интернет-протоколах и повторяться сейчас нет смысла.

Также вы знаете, что для пересылки по Сети Web-страниц и связанных с ними файлов (графических изображений, звуков, архивов и т. п.) используется протокол HTTP. Он же применяется и для передачи данных, причем для этого предусмотрены два *метода передачи* данных. Оба способа широко используются в интернет-программировании и имеют свои преимущества и недостатки. Давайте их рассмотрим.

Первый способ носит название *GET* по значению соответствующего параметра формы. При его использовании данные передаются как часть интернет-адреса в HTTP-запросе.

Как вы помните, Web-обозреватель для того, чтобы получить от Web-сервера нужный ему файл, отправляет этому серверу так называемый HTTP-запрос, включающий в себя интернет-адрес необходимого файла. Так вот, данные могут быть переданы как часть этого адреса.

Возьмем, например, такой набор данных, приведенный чуть выше:

```
name1 = Ivan
surname = Ivanovich
name2 = Ivanov
age = 30
```

Теперь подготовим его для пересылки по методу GET (сами данные выделены полужирным шрифтом):

```
http://www.somesite.ru/bin/program.exe?name1=Ivan&surname2=Ivanovich&
↳name2=Ivanov&age=30
```

Как видите, пересылаемые по методу GET данные помещаются в самый конец интернет-адреса и отделяются от него вопросительным знаком. При этом пары "имя" = "значение" отделяются друг от друга знаком "коммерческое и" ("&"). Все очень просто и наглядно.

Такая простота и наглядность представления данных — основное преимущество метода GET. Как говорится, все на виду. Также значительно упрощается отладка Web-страниц: поскольку передаваемый Web-серверу адрес отображается в строке адреса Web-обозревателя, вы всегда сможете увидеть, что именно было передано. (Однако, как вы понимаете, конфиденциальные данные таким методом не передашь — их увидят все, кто стоит за вашей спиной.)

Часто создаются Web-сайты, гиперссылки в которых организуется следующим образом:

```
http://www.mysite.ru/bin/choose.exe?chapter=3
```

Как видите, фактически это ссылки на серверную программу, содержащие один параметр `chapter` и его значение. Это значит, что все остальные страницы такого сайта формируются серверной программой динамически, на основании полученных параметров. По такому принципу очень часто строятся сайты-справочники, сайты-каталоги программ, электронные магазины и другие сайты, содержащие большое количество классифицированной информации.

К несчастью, метод GET обладает огромным недостатком: с его помощью невозможно передавать большие объемы данных. Это происходит из-за ограничения, накладываемого стандартами на длину интернет-адреса: не более 256 символов. Вычтите отсюда длину собственно адреса серверной программы — и вы получите максимально допустимый размер ваших данных. Второй недостаток метода GET — обратная сторона его достоинства. Данные, пересылаемые им, открыты для всеобщего обозрения и могут быть легко прочитаны в строке адреса Web-обозревателя.

Метод GET стоит использовать, если пересылаемые серверной программе данные заведомо невелики и не являются секретными. В частности, он используется для пересылки ключевых слов поисковым машинам, в сайтах, построенных на основе серверной программы *(см. выше)* и т. п. Если же вам нужно пересылать объемистые либо конфиденциальные данные, используйте второй метод передачи, называемый POST.

Метод *POST* передает данные серверной программе все в том же HTTP-запросе, но уже не частью интернет-адреса, а в виде так называемых дополнительных данных. Поскольку размер дополнительных данных не ограничен (по крайней мере, он может быть очень велик), вы можете передавать все, что угодно, в каких угодно количествах. В частности, именно этим способом Web-серверу могут передаваться даже файлы.

Достоинства метода POST: отсутствие ограничения на объем передаваемых данных и "невидимость" их. Недостатки: сложность расшифровки данных и трудность отладки. Методом POST передаются, например, анкетные данные, адреса покупателей в электронных магазинах, литературные произве-

дения на сайты **http://www.stihi.ru** и **http://www.proza.ru** и т. п. В общем, то, что имеет большие объемы.

Как говорят, комитет WWWC намерен вообще со временем отказаться от метода GET и все данные передавать с помощью метода POST. Пока что метод GET просто объявлен не рекомендованным для использования во вновь создаваемых сайтах, реально же он еще поддерживается Web-обозревателями.

Вот мы и выяснили все о серверных программах. Ну, может, не все, но пока нам этого достаточно. Теперь поговорим о том, насколько полно все это поддерживается Dreamweaver MX.

Серверное программирование — подход Dreamweaver

Трудно ли писать серверные программы? Да, трудно. Вероятно, даже труднее, чем Web-страницы.

Однако вспомните, часто ли нам приходилось писать HTML-код вручную? Совсем редко, не правда ли? А все потому, что Dreamweaver заботливо оберегал нас от этого, предоставляя удобный интерфейс для визуального создания страниц. Мы просто писали текст, форматировали его, помещали на страницу изображения, таблицы, применяли к элементам страницы поведения и т. п. Одним словом, чувствовали себя комфортно.

И неужели нам теперь придется писать серверные программы вручную?!

Нет, совсем не обязательно.

Уже упоминалось, что Dreamweaver предоставляет неопытным пользователям и вообще тем, кто не хочет иметь дела с JavaScript-кодом, так называемые поведения. *Поведение* — это уже готовый сценарий, написанный профессиональными программистами и помещаемый в код Web-страницы самим Dreamweaver после того, как пользователь выберет это поведение в меню поведений панели **Behaviors**. Пользователю не надо заботиться о том, что в определенное место кода страницы необходимо поместить сценарий, выполняющий то или иное действие, ему не нужно будет проверять этот сценарий на наличие ошибок и согласовывать его с другими сценариями. Все это за него делает Dreamweaver.

Но Dreamweaver идет дальше. Он предлагает поведения не только для клиентских, но и для серверных сценариев. Так что вы можете создавать с его помощью не только статические, но и динамические серверные страницы.

В самом деле, набор операций, применяемых в серверном программировании, очень невелик. Открыть базу данных, получить из нее какие-то данные, вывести их на странице, записать новые данные — что еще нужно!

Конечно, если вы хотите реализовать какие-то хитроумные действия, вам придется писать серверные страницы "врукопашную". Но, согласитесь, такое бывает нечасто.

Итак, мы выяснили две вещи. Во-первых, Dreamweaver MX поддерживает создание серверных Web-страниц. Во-вторых, для помещения в них сценариев используются хорошо знакомые вам поведения. Dreamweaver предлагает большой набор поведений для серверных сценариев, которых вам хватит на первых порах.

Осталось выяснить, какие именно серверные страницы позволяет создавать Dreamweaver. Точнее, какие технологии создания серверных страниц он позволяет использовать. Таких технологий четыре, и сейчас они будут перечислены.

1. ASP. Эта технология вам уже знакома.
2. ASP.NET. Дальнейшее развитие ASP.
3. PHP. Бесплатная технология, распространяемая с открытыми исходными текстами. Довольно популярна и часто используется с бесплатными же Web-серверами, например Apache.
4. Macromedia ColdFusion. Собственная разработка фирмы Macromedia.

Какую же из них выбрать? Ведь нам надо будет создавать примеры Web-страниц.

Давайте выберем ASP. И вот почему.

Для того чтобы работать с серверными страницами, нам понадобится Web-сервер. Без него серверные страницы просто не будут работать. А, возможно, вы знаете, что со всеми более-менее новыми версиями Windows поставляется небольшой Web-сервер. В системах Windows 95/98/ME это Personal Web Server, а в Windows NT/2000/XP — Internet Information Server. Его возможностей нам вполне хватит для экспериментов с серверным программированием.

Так вот, этот Web-сервер полноценно поддерживает технологию ASP. Вы можете писать активные серверные страницы и выполнять их под ним. И вам не придется загружать никаких дополнительных компонентов — все, что вам нужно, уже включено в состав дистрибутивного комплекта Windows. Вот поэтому и был выбран ASP.

Технологию ASP.NET пока что не поддерживает практически ни один Web-сервер; фирма Microsoft пока что не выпустила ни один продукт с поддержкой этой технологии. Для работы с PHP-страницами вам придется искать и загружать по Сети довольно большой дистрибутивный комплект, содержащий обработчик PHP, а потом долго его настраивать. Что касается технологии ColdFusion, то вряд ли вы в состоянии выложить за нее несколько тысяч долларов. Так что вариантов у нас практически нет.

Введение в базы данных

Последнее, что мы рассмотрим в данной главе, — это базы данных и работу с ними. Поскольку львиная доля серверных программ работает именно с базами данных, нам эти знания очень пригодятся. Конечно, мы не будем рассматривать все технологии баз данных во всех подробностях, а, как и раньше, ограничимся кратким ликбезом. Вы сами можете поискать нужные книги и тексты в Сети, если заинтересуетесь этим.

Что такое *база данных*? Ничего особенного, обычный файл или группа файлов, содержащих данные, организованные особым образом. Если база данных состоит из множества файлов, она все равно остается единым целым. Данные, содержащиеся в базе, обрабатываются при помощи особой программы, называемой *процессором баз данных*. Процессор баз данных может быть как совершенно отдельной программой, работающей на том же компьютере, что и использующая данные программа, так и входить в ее состав.

Базы данных могут быть организованы по-разному. Но подавляющее большинство баз данных, эксплуатируемых в настоящее время, являются *реляционными*. Данные в таких базах организуются в виде таблиц. Каждая такая база данных может включать в себя одну или большее количество таблиц; сложные базы данных, как правило, имеют много таблиц, связанных между собой.

Каждая таблица в свою очередь состоит из набора строк, разделенных на ячейки, причем в каждой ячейке содержатся данные определенного типа: текст, числа, логические величины, даты и т. д. Строки таблиц баз данных называются *записями*, а ячейки, на которые делится каждая запись, — *полями*. Как уже говорилось, каждое поле имеет определенный тип и обязательно снабжено *именем*, по которому программа, работающая с данными, и осуществляет доступ к этому полю.

На рис. 15.1 приведен пример такой таблицы. Как видите, эта таблица имеет три поля:

- `NAME` — название технологии создания серверных страниц, текстовое;
- `PRICE` — цена программы-обработчика, числовое;
- `USING` — поддерживается ли эта технология в настоящее время, логическое ("да-нет").

Кроме того, эта таблица имеет четыре записи, соответствующие технологиям, поддерживаемым Dreamweaver MX: ASP, ASP.NET, PHP и ColdFusion.

Очень часто одно из полей таблицы делается *ключевым*. Значение ключевого поля используется для однозначной идентификации какой-либо записи. Разумеется, ключевые поля всех записей таблицы должны в этом случае содержать уникальные значения. Иногда ключевое поле называется *полем счетчика*.

NAME	PRICE	USING
ASP	0	+
ASP.NET	0	-
PHP	0	+
ColdFusion	2000	+

Рис. 15.1. Пример таблицы, входящей в базу данных

Как получить доступ к отдельному полю, вы уже знаете, — по его имени. А как получить доступ к нужной записи?

Дело в том, что в один момент времени программа, использующая данные базы, может работать только с одной записью, называемой *текущей*. Она может извлекать данные из полей этой записи и, возможно, изменять их. Но чтобы получить данные другой записи, программа должна выполнить команду перемещения. При этом процессор баз данных перемещает особый *указатель текущей записи* на нужную запись, и программа получает возможность работать с ней.

Также программа может использовать *поиск* нужной записи по какому-либо критерию. Очень часто такой поиск выполняется по значению ключевого поля.

Также программа имеет возможность добавлять новые и удалять ненужные записи таблицы. Чтобы добавить новую запись, программа выполняет команду добавления записи и заносит в ее поля необходимые данные. Чтобы удалить запись, программа сначала должна сделать ее текущей, а потом выполнить команду удаления.

В общем случае, последовательность работы с данными, содержащимися в базе, выглядит так:

1. Программа *открывает* базу данных, выполняя операцию открытия. Это необходимая операция, без которой невозможно получить доступ к базе.
2. Программа открывает нужную таблицу базы. После этого процессор возвращает ей так называемый *набор записей* (по-английски — recordset), с которым и работает программа.
3. Программа выполняет собственно работу с данными.
4. Программа *закрывает* таблицу, после чего процессор убирает соответствующий этой программе набор записей из памяти.
5. Программа закрывает базу данных, разрывая все связи с ней.

Учтите, что две последние операции — закрытие таблицы и базы данных — выполнять так же необходимо, как и открытие. Дело в том, что набор записей и прочие структуры данных, формируемые процессором в памяти для программы, работающей с данными, отнимают много системных ресурсов. Поэтому, как только вы закончили работу с таблицей или базой данных, сразу же закройте ее, чтобы освободить ресурсы для других пользователей.

В последнее время большую популярность приобрели *серверы баз данных*. Это обычные процессоры данных, но реализованные в виде серверных программ и работающие на серверных компьютерах. Их преимущества перед обычными процессорами данных:

- они работают на серверном компьютере, который, как правило, мощнее клиентского, поэтому их производительность выше;
- они обеспечивают бóльшую защищенность данных за счет разграничения доступа и некоторых других механизмов;
- они просто мощнее, в смысле, поддерживают больше различных нововведений, появившихся в последнее время.

Чтобы получить доступ к серверу баз данных и самим данным, клиентская программа посылает ему особые команды. Для составления таких команд был разработан язык описания запросов *SQL* (Structured Query Language — язык структурированных запросов). С помощью команд, составленных на этом языке, клиентская программа может открыть нужную таблицу, считать данные, добавить, изменить, удалить запись и, в конечном счете, закрыть базу данных, когда нужда в ней отпадет.

Серверы баз данных применяются сейчас очень часто, а в Web-программировании — почти повсеместно. К наиболее популярным можно отнести Oracle, Microsoft SQL Server, Sybase, мощнейший IBM DB2, Borland InterBase (в России он продается под названием IBase), набирающий популярность PostgressSQL и бесплатный сервер MySQL, распространяющийся с открытыми исходными текстами. Последний сервер, кстати, очень часто используется в связке с Web-сервером Apache и технологией серверных страниц PHP.

На этом рассказ о базах данных и о серверном программировании можно считать законченным.

Что дальше?

Уже говорилось, что отправку данных серверной программе выполняют особые элементы Web-страниц — формы. Именно о формах и о работе с ними в Dreamweaver мы с вами будем говорить в следующей главе. А уже потом перейдем собственно к серверному программированию.

ГЛАВА 16

Формы

Формы (также их называют Web-формами), как вы помните, служат для сбора данных пользователя и отправки их Web-серверу. Они представляют собой набор из элементов управления (полей ввода, флажков, переключателей, списков и обычных кнопок), размещаемых на Web-странице. В этом смысле они аналогичны обычным диалоговым окнам Windows-приложений и работают так же.

Web-формы — такой же часто встречающийся элемент страниц, как, скажем, графические изображения или таблицы. Очень и очень многие сайты содержат анкеты для опроса посетителей. Почтовые системы, работающие через Web-обозреватель ("Web-почта"), спрашивают у вас кодовое имя (или, как говорят опытные интернетчики, *"логин"*, по-английски — login) и пароль прежде, чем допустить к содержимому вашего почтового ящика. Разнообразные сайты-справочники предусматривают возможности поиска, для чего предоставляют специальные формы с полем ввода, где вы должны будете ввести ключевое слово. А электронные магазины требуют у вас почтовый адрес, чтобы отправить вам купленный товар.

Интернет становится все более и более интерактивным, и в дальнейшем эта тенденция будет только нарастать. А для интерактивности ему нужно принимать данные от своих пользователей. Единственный в настоящее время способ сделать это — использовать серверные программы и, разумеется, формы.

О серверных программах подробно говорилось в *главе 15*. Так что не будем повторять все это, а сразу перейдем к работе с формами в среде Dreamweaver.

Работа с формами в Dreamweaver

Создайте в Dreamweaver новую страницу. Мы поместим в нее форму, собирающую анкетные данные пользователя и отсылающую их гипотетической серверной программе program.exe.

Создание формы

Чтобы поместить на Web-страницу форму, проще всего воспользоваться панелью объектов. Для этого переключитесь на вкладку **Forms**, где находятся кнопки, помещающие на Web-страницу форму или один из предусмотренных в HTML элементов управления. Кнопка, помещающая на страницу форму, показана на рис. 16.1 и называется **Form**. Также вы можете выбрать пункт **Form** в меню **Insert**.

Рис. 16.1. Кнопка **Form** панели объектов

Пустая форма показывается в виде узкой полоски, ограниченной красной пунктирной рамкой (рис. 16.2). Когда вы поместите что-нибудь в форму, ее размеры соответственно увеличатся.

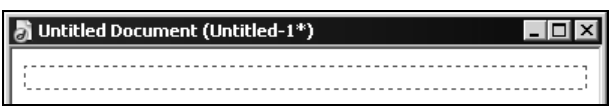


Рис. 16.2. Пустая форма на Web-странице

Вновь созданная форма будет сразу же выделена, чтобы вы смогли установить нужные параметры. К несчастью, это никак не сигнализируется визуально: выделенная форма ничем не отличается от невыделенной. Поэтому, если вы сомневаетесь, выделена ли форма, выделите ее сами. Для этого просто щелкните мышью по ее красной штриховой рамке.

Что может быть бесполезнее пустой формы! Поэтому мы не будем останавливаться на ней, а поскорее перейдем к элементам управления. Опишем только, какие параметры вы можете задать для формы. А параметров этих очень и очень немного (рис. 16.3).

Рис. 16.3. Вид редактора свойств при выделенной форме

Сначала о поле ввода **Form Name**. В нем вводится имя формы. По умолчанию Dreamweaver сам подставляет туда автоматически сгенерированное имя вида `form1`, `form2` и т. д. В подавляющем большинстве случаев автоматически сгенерированного имени будет вполне достаточно; вообще, имя формы особой роли не играет, в отличие от имен элементов управления. Но если

вы любите аккуратность во всем, можете задать для своей формы более вразумительное имя.

Имя формы вы также можете задать, выбрав пункт **Name** в контекстном меню формы (т. е. контекстном меню, появляющемся при щелчке правой кнопкой мыши по форме). После этого на экране появится диалоговое окно **Change Attribute**. Введите новое значение параметра в поле ввода и нажмите кнопку **OK** для его сохранения или **Cancel** — для отмены.

В поле ввода **Action** вводится интернет-адрес серверной программы, которая будет обрабатывать введенные в форму данные. Вы можете также щелкнуть по значку папки справа от этого поля ввода и выбрать нужный файл в диалоговом окне **Select File**. Кроме того, вы можете выбрать пункт **Action** контекстного меню формы, чтобы вызвать на экран все то же диалоговое окно.

В комбинированном списке **Target** задается, как вы помните, цель гиперссылки. Конечно, сейчас у нас нет гиперссылки — в данном случае цель задает, куда будет выводиться Web-страница, сгенерированная серверной программой.

Метод пересылки данных задается с помощью раскрывающегося списка **Method**. В этом списке доступны три пункта: **GET**, **POST** и **Default**. Первые два пункта, как вы поняли, задают метод пересылки, а третий — метод, используемый по умолчанию (как правило, GET). Для аналогичных целей служит подменю **Method** контекстного меню формы, содержащее все те же три пункта.

Кодировка данных формы задается в комбинированом списке **Enctype**. В этом списке есть два пункта: **application/x-www-form-urlencoded** и **multipart/form-data**. Если вы собираетесь использовать другую кодировку, например, `text/plain`, можете ввести ее название прямо в этот список.

Примечание

Форма создается с помощью парного тега `<FORM>...</FORM>`, внутри которого помещается все содержимое формы, т. е. элементы управления. Атрибут `ACTION` задает адрес серверной программы, `ENCTYPE` — кодировку, а `METHOD` — метод отправки данных.

Задайте какие-нибудь данные для нашей формы. И сохраните страницу в файле под именем 16.1.htm.

Вот и все о форме. Теперь пора наполнить ее содержимым.

Элементы управления

Элементы управления помещаются на форму с помощью все той же вкладки **Forms** панели объектов или пунктов подменю **Form Objects** меню **Insert**. Ниже мы подробно рассмотрим каждый элемент управления, предлагаемый нам стандартом HTML.

Элементы управления вставляются в "поток" текста и ведут себя как обычные элементы страницы. Вы можете разносить их по текстовым абзацам или помещать в одну и ту же строку, разделяя пробелами. Также вы можете вставлять их в ячейки таблицы и свободно позиционируемые элементы. Чтобы удалить ненужный элемент управления, просто выделите его и нажмите клавишу <Del>.

Поле ввода

Обычное поле ввода, служащее для ввода текста, самый популярный в формах элемент управления, помещается в форму с помощью кнопки **Text Field** (рис. 16.4) панели объектов или пункта **Text Field** подменю **Form Objects** меню **Insert**.

Рис. 16.4. Кнопка **Text Field** панели объектов

Появившееся на экране вновь созданное поле ввода будет выделено. Об этом сигнализирует тонкая штриховая рамка, окружающая его. Таким образом, вы можете сразу же установить необходимые параметры поля ввода (рис. 16.5). Если же поле ввода почему-то не выделено, щелкните по нему мышью.

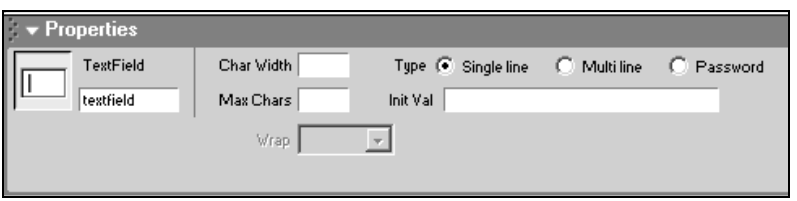


Рис. 16.5. Вид редактора свойств при выделенном поле ввода

В поле ввода **TextField** вводится уникальное имя элемента управления. Dreamweaver автоматически подставит туда сгенерированное им самим имя, но вам сразу же следует его изменить. Вы также можете выбрать пункт **Name** в контекстном меню поля ввода и изменить это имя в появившемся на экране диалоговом окне **Change Attribute**.

В поле ввода **Char Width** задается ширина поля ввода, т. е. его горизонтальный размер, в символах.

В поле ввода **Max Chars** задается максимальное количество символов, которое может быть введено в созданное на Web-странице поле ввода. Это число может быть больше, чем **Char Width**; в этом случае содержимое поля вво-

да будет прокручиваться по горизонтали. Но ни в коем случае не задавайте **Max Chars** меньше, чем **Char Width**.

В поле ввода **Init Val** задается начальное значение, помещаемое в поле ввода при загрузке формы. Вы также можете выбрать пункт **Value** в контекстном меню поля ввода и изменить начальное значение в диалоговом окне **Change Attribute**.

Группа переключателей **Type** задает тип создаваемого поля ввода. Здесь доступны три переключателя:

- **Single line** — обычное поле ввода в одну строку;
- **Multi line** — многострочное поле ввода, иначе говоря, область редактирования;
- **Password** — поле ввода пароля.

Поле ввода пароля ничем не отличается от обычного однострочного поля ввода за тем исключением, что вместо вводимых символов в нем отображаются звездочки. Такие поля ввода широко применяются для ввода паролей или других конфиденциальных данных. Если вы хотите скрыть вводимые данные от любопытных соседей, применяйте поле ввода пароля.

Область редактирования вам уже знакома. От обычного поля ввода она отличается тем, что в нее может быть введен многострочный текст. Для области редактирования Dreamweaver предоставляет дополнительные параметры. Вы увидите их, если выберете переключатель **Multi line** (рис. 16.6).

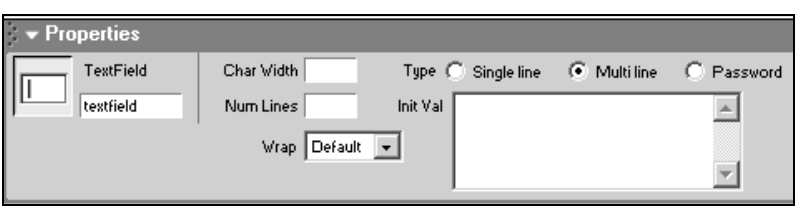


Рис. 16.6. Вид редактора свойств при выделенной области редактирования

Прежде всего, вместо поля ввода **Max Chars** появится поле **Num Lines**, в котором задается вертикальный размер области редактирования в строках. Заметьте, что размер текста, который может быть помещен в область редактирования, не ограничен.

Вместо поля ввода **Init Val** появится одноименная область редактирования. В самом деле, для задания начального значения области редактирования обычного поля ввода может быть мало. Обратите внимание, что в этом случае пункт **Value** из контекстного меню исчезнет.

И — самое важное дополнение. С помощью раскрывающегося списка **Wrap** задается, как область редактирования будет выполнять перенос текста и

в каком виде этот текст будет отправлен серверной программе. В этом списке доступны четыре пункта. Рассмотрим их подробнее.

Пункт **Off** отключает перенос строк. В этом случае, если текст не помещается в область редактирования по горизонтали, в ней появляется горизонтальная полоса прокрутки. (При этом пользователь может вставить в текст *"жесткий" перевод строки*, нажав клавишу <Enter>.)

Если выбран пункт **Virtual**, область редактирования будет выполнять перенос длинных строк, вставляя так называемые *"мягкие" переводы строк*. При отправке содержимого области редактирования на сервер эти переводы удаляются. (Собственно, никаких "мягких" переводов в текст вообще не помещается — это просто такой термин.)

Если же выбран пункт **Physical**, то перед отправкой данных все "мягкие" переводы строк преобразуются в "жесткие". Фактически в нужных местах текста вставляются символы перевода строк.

Последний пункт — **Default** — задает поведение по умолчанию. Как правило, он аналогичен **Off**.

Какой пункт раскрывающегося списка выбрать? Простого ответа на этот вопрос не существует. Точнее, он зависит от того, что вы хотите иметь на Web-странице и что должна будет обрабатывать серверная программа. Перебрав все варианты, автор составил табл. 16.1, которая может вам помочь хотя бы на первых порах работы с формами.

Таблица 16.1. *Выбор пункта раскрывающегося списка* ***Wrap***

Как должны отображаться данные	Как данные должны получаться серверной программой	Пункт списка
Неизменными	Неизменными	**Off**
Измененными (с "мягкими" переносами)	Неизменными	**Virtual**
Измененными (с "мягкими" переносами)	Измененными (с "жесткими" переносами строк)	**Physical**

Если вы в своей форме используете область редактирования, проверьте, задан ли в качестве метода передачи данных POST. Поскольку в область редактирования может быть введен очень большой текст, метод GET не сможет передать его из-за своих ограничений.

Внимание!

Если вы хотите быстро вставить в форму область редактирования, воспользуйтесь кнопкой **Textarea** (рис. 16.7) панели объектов или пунктом **Textarea** подменю **Form Objects** меню **Insert**.

Рис. 16.7. Кнопка **Textarea** панели объектов

Примечание

Обычное поле ввода создается посредством одинарного тега `<INPUT TYPE="text">`. (Обратите внимание на значение атрибута `TYPE`.) Поле ввода пароля создается с помощью одинарного тега `<INPUT TYPE="password">`, область редактирования — с помощью парного тега `<TEXTAREA>...</TEXTAREA>`, внутри которого помещается содержимое области редактирования. Значение, введенное в поле ввода или область редактирования, преобразуется к виду `<Имя поля ввода>=<Значение>` и отправляется в таком виде. Имя элемента управления задается атрибутом `NAME` тегов `<INPUT>` и `<TEXTAREA>`.

Кнопка

Кнопка — второй по популярности элемент управления в Web-формах. Для помещения ее в форму используется кнопка **Button** (рис. 16.8) панели объектов или пункт **Button** подменю **Form Objects** меню **Insert**. Доступные параметры показаны на рис. 16.9.

Рис. 16.8. Кнопка **Button** панели объектов

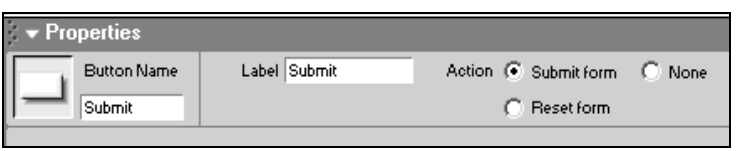


Рис. 16.9. Вид редактора свойств при выделенной кнопке

В поле ввода **Button Name** вводится уникальное имя элемента управления. Dreamweaver автоматически подставит туда сгенерированное им самим имя, имеющее вид `Submit` (для кнопки отправки данных), `Reset` (для кнопки сброса) или `Button` (для обычной кнопки). Как правило, этого имени вполне достаточно, но вы, если хотите, можете его изменить. Вы также можете выбрать пункт **Name** в контекстном меню кнопки и изменить это имя в появившемся на экране диалоговом окне **Change Attribute**.

В поле ввода **Label** вводится текст, который будет отображаться на кнопке, иначе говоря, *надпись на кнопке*. Dreamweaver сам подставит туда текст `Submit` (для кнопки отправки данных), `Reset` (для кнопки сброса) или `Button` (для обычной кнопки). Естественно, его придется менять на аналогичные

русские надписи. Вы также можете выбрать пункт **Label** в контекстном меню кнопки и изменить этот текст в появившемся на экране диалоговом окне **Change Attribute**.

С помощью группы переключателей **Action** задается действие, происходящее при нажатии этой кнопки. Здесь доступны три переключателя:

- **Submit form** — отправка данных, введенных в форму, серверной программе *(кнопка отправки данных)*;
- **Reset form** — сброс данных, введенных в форму (точнее, замена их начальными значениями) *(кнопка сброса формы)*;
- **None** — ничего не происходит *("кнопка-бездельник")*.

С первыми двумя действиями мы уже знакомы. Это стандартные действия, выполняемые самим Web-обозревателем. Как мы знаем, каждая форма, отправляющая данные серверной программе, должна включать в себя кнопку отправки данных. (Кнопка сброса не является обязательной; ее ввели для удобства посетителя, но реально от нее нет особой пользы.)

Но зачем нужны кнопки, при нажатии на которых ничего не происходит? Собственно, в форме они не нужны. Такие кнопки работают вместе со сценариями; при нажатии на такую кнопку запускается сценарий, выполняющий какое-либо действие над Web-страницей. Так что вы можете привязать к такой кнопке какое-либо поведение (о поведениях *см. главу 13*).

Примечание

Кнопка отправки данных создается с помощью одинарного тега `<INPUT TYPE="submit">`, кнопка сброса данных формы — `<INPUT TYPE="reset">`, "кнопка-бездельник" — `<INPUT TYPE="button">`. Каждая кнопка формирует данные в виде пары `<Имя кнопки>=1` и отсылает их серверной программе, но эти данные практически никогда реально не обрабатываются.

Флажок

Флажки используются в формах также довольно часто. Для помещения его в форму используется кнопка **Checkbox** (рис. 16.10) панели объектов или пункт **Check Box** подменю **Form Objects** меню **Insert**. Доступные параметры показаны на рис. 16.11.

Рис. 16.10. Кнопка **Checkbox** панели объектов

В поле ввода **CheckBox** вводится уникальное имя элемента управления. Dreamweaver автоматически подставит туда сгенерированное им самим имя, но вам сразу же следует его изменить. Вы также можете выбрать пункт

Name в контекстном меню флажка и изменить это имя в появившемся на экране диалоговом окне **Change Attribute**.

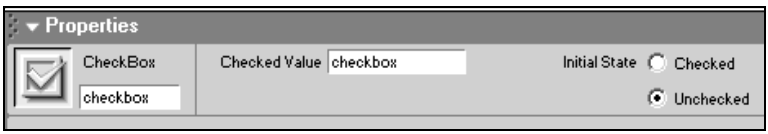


Рис. 16.11. Вид редактора свойств при выделенном флажке

В поле ввода **Checked Value** вводится текст, отправляемый серверной программе, если флажок будет включен посетителем, т. е. если флажок будет включен, он отправит такую пару:

```
<Имя флажка>=<Значение Checked Value>
```

Если же флажок не был включен, отправляемая пара будет, так сказать, "неполной":

```
<Имя флажка>=
```

Вы также можете выбрать пункт **Value** в контекстном меню флажка и изменить отправляемый текст в появившемся на экране диалоговом окне **Change Attribute**.

Группа переключателей **Initial State** задает начальное состояние флажка. Если включен переключатель **Unchecked**, флажок будет отключен (значение по умолчанию), а переключатель **Checked** сделает его изначально включенным.

Примечание

Флажок создается с помощью одинарного тега `<INPUT TYPE="checkbox">`.

Переключатели

Переключатели в Web-формах, как и в обычных Windows-окнах, применяются только группами. В самом деле, в одиночку они абсолютно бесполезны — в таком случае гораздо удобнее использовать флажки.

Для помещения в форму переключателя используется кнопка **Radio Button** (рис. 16.12) панели объектов или пункт **Radio Button** подменю **Form Objects** меню **Insert**. Доступные параметры показаны на рис. 16.13.

Рис. 16.12. Кнопка **Radio Button** панели объектов

Рис. 16.13. Вид редактора свойств при выделенном переключателе

В поле ввода **RadioButton** вводится уникальное имя группы переключателей. Переключатели, входящие в одну группу, должны иметь одно и то же имя. В данном случае принцип "каждому элементу управления — уникальное имя" нарушается. Уникальное имя должна иметь каждая группа переключателей.

Dreamweaver автоматически подставит в поле ввода **RadioButton** сгенерированное им самим имя, но вам сразу же следует его изменить. Вы также можете выбрать пункт **Name** в контекстном меню переключателя и изменить это имя в появившемся на экране диалоговом окне **Change Attribute**.

В поле ввода **Checked Value** вводится текст, отправляемый серверной программе, если переключатель будет включен посетителем, т. е. если он будет включен, серверной программе будут отправлены такие данные:

```
<Имя группы переключателей>=<Значение Checked Value>
```

По значению `Checked Value` серверная программа может определить, какой переключатель группы был включен. Если же ни один переключатель в наборе не был включен, отправляемая пара опять же будет "неполной":

```
<Имя группы переключателей>=
```

Вы также можете выбрать пункт **Value** в контекстном меню переключателя и изменить отправляемый текст в появившемся на экране диалоговом окне **Change Attribute**.

Группа переключателей **Initial State** задает начальное состояние созданного вами переключателя. Если включен переключатель **Unchecked**, наш переключатель будет отключен (значение по умолчанию), а переключатель **Checked** сделает его изначально включенным. Но не забудьте, что соглашения фирмы Microsoft о пользовательском интерфейсе Windows-приложений требуют, чтобы один из переключателей в группе обязательно был включен изначально.

Примечание

Переключатель создается с помощью одинарного тега `<INPUT TYPE="radio">`.

Группа переключателей

Dreamweaver предоставляет удобные средства создания сразу целой группы переключателей. В самом деле, если переключатели целесообразно использовать только группой, неудобно создавать их по одному.

Чтобы создать в форме группу переключателей, нажмите кнопку **Radio Group** (рис. 16.14) панели объектов или выберите пункт **Radio Group** подменю **Form Objects** меню **Insert**. После этого на экране появится диалоговое окно **Radio Group** (рис. 16.15).

Рис. 16.14. Кнопка **Radio Group** панели объектов

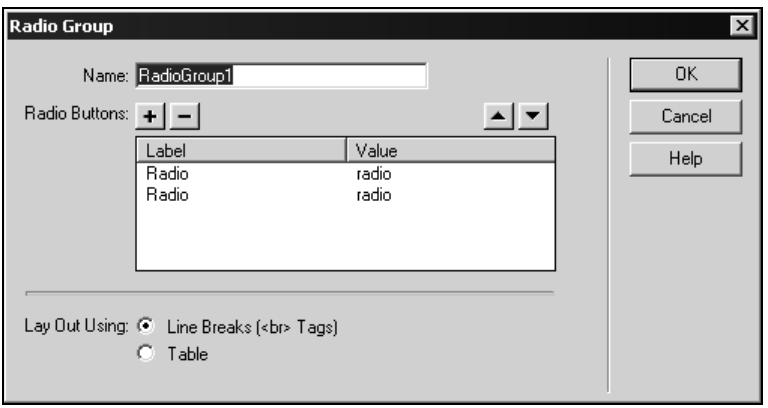


Рис. 16.15. Диалоговое окно **Radio Group**

В поле ввода **Name** этого окна вводится уникальное имя создаваемой группы переключателей. Dreamweaver автоматически подставит туда сгенерированное им самим имя, но вам сразу же следует его изменить.

Все созданные вами переключатели группы отображаются в списке **Radio Buttons**. Этот список состоит из двух колонок: **Label** (надпись возле переключателя) и **Value** (значение, отсылаемое серверной программе, если этот переключатель был включен посетителем). Вы можете выбрать любой пункт этого списка и произвести над ним некоторые манипуляции.

Как видите, Dreamweaver уже создал для вас два переключателя; вам остается только задать для них надпись и значение. Для этого выберите нужный пункт списка и щелкните по значению, находящемуся в соответствующей колонке. Вместо значения появится небольшое поле ввода; введите, что хотите, и нажмите клавишу <Enter>.

Чтобы добавить еще один переключатель в список, нажмите кнопку со знаком "плюс", находящуюся над списком. В списке появится еще один пункт; задайте для него нужные значения.

Чтобы удалить ненужный переключатель, выберите в списке соответствующий ему пункт и нажмите кнопку со знаком "минус".

Кнопки ↑ и ↓ позволят вам переместить выбранный пункт списка на позицию выше и ниже.

Группа переключателей **Lay Out Using** позволяет задать, каким образом переключатели создаваемой группы будут размещаться в форме. Если включен переключатель **Line Breaks (
 Tags)**, переключатели будут размещены в одном текстовом абзаце и отделены друг от друга тегом разрыва строки <BR>. Если же включен переключатель **Table**, Dreamweaver поместит все создаваемые переключатели в таблицу.

Задав нужные данные, нажмите кнопку **OK**. Если вы передумали создавать группу переключателей, нажмите кнопку **Cancel**.

После того как группа переключателей будет создана, вы можете выделить любой из них и изменить его параметры с помощью редактора свойств.

Список

Для помещения в форму обычного или раскрывающегося списка используется кнопка **List/Menu** (рис. 16.16) панели объектов или пункт **List/Menu** подменю **Form Objects** меню **Insert**. Доступные параметры показаны на рис. 16.17.

Рис. 16.16. Кнопка **List/Menu** панели объектов

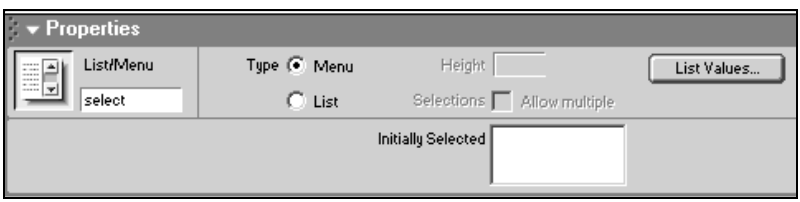


Рис. 16.17. Вид редактора свойств при выделенном списке

В поле ввода **List/Menu** вводится уникальное имя списка. Dreamweaver автоматически подставит туда сгенерированное им самим имя, но вам сразу же следует его изменить. Вы также можете выбрать пункт **Name** в контекстном меню списка и изменить это имя в появившемся на экране диалоговом окне **Change Attribute**.

С помощью группы переключателей **Type** вы можете выбрать между обычным списком (переключатель **Menu**) и раскрывающимся списком (переключатель **List**).

При нажатии на кнопку **List Values** на экране появляется диалоговое окно **List Values**. Это окно показано на рис. 16.18. В нем вы сможете ввести

пункты, которые будут отображаться в создаваемом вами списке. Вызвать это окно можно также, выбрав пункт **List Items** в контекстном меню списка.

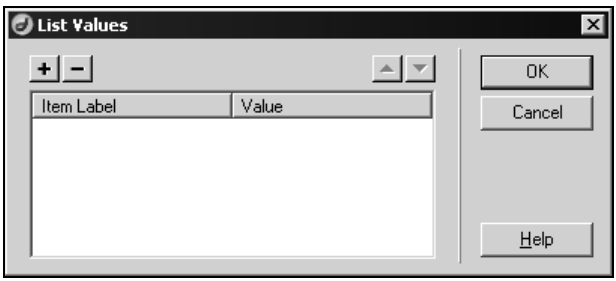


Рис. 16.18. Диалоговое окно **List Values**

Большую часть этого окна занимает список пунктов. Каждая позиция этого списка состоит из двух полей: поля названия пункта (**Item Label**), которое отображается в форме, и поля значения (**Value**), которое будет отправлено серверной программе, если этот пункт будет выбран, т. е. отправлена будет такая пара:

```
<Имя списка>=<Значение пункта>
```

Если ни один из пунктов списка не будет выбран, серверная программа получит уже знакомую вам "неполную" пару:

```
<Имя списка>=
```

При щелчке на строке списка в одной из колонок там появится поле ввода, где вы сможете ввести либо название пункта, либо его значение. Нажимая клавишу <Tab>, вы сможете перемещаться по позициям списка, от первой до последней. Если вы нажмете эту клавишу, находясь в колонке **Value** последней позиции списка, в список будет добавлена новая пустая позиция.

Кнопка со знаком "плюс" позволяет добавить в список новую позицию, кнопка со знаком "минус" удаляет выбранную. Кнопки ↑ и ↓ перемещают выбранную позицию списка выше или ниже.

Введя все пункты создаваемого списка, нажмите кнопку **OK**, после чего список будет заполнен введенными пунктами.

В списке **Initially Selected** вы сможете задать пункт создаваемого на Web-странице списка, который будет выбран в нем изначально.

Если в наборе **Type** выбран переключатель **List** (т. е. создается список), становятся доступными поле ввода **Height** и флажок **Allow Multiple**.

В поле ввода **Height** задается высота списка в пунктах. Вы можете задать высоту либо равной общему количеству пунктов, либо меньше ее. Если вы

зададите слишком большое значение, список будет заполнен не полностью, что, собственно, не страшно, но и не очень красиво.

Флажок **Allow Multiple** позволяет включить особый режим работы списка, при котором пользователь может выбрать в нем несколько пунктов. Для этого он должен будет, щелкая мышью на нужных пунктах, удерживать нажатой клавишу <Ctrl>. Он также может выбрать целую группу пунктов, сначала щелкнув на первом пункте, а потом — на последнем, удерживая клавишу <Shift>. В этом случае серверная программа получит следующую пару:

```
<Имя списка>=<Значение пункта 1>, <Значение пункта 2>, ...
```

Примечание

И обычный, и раскрывающийся списки создаются с помощью одного и того же парного тега `<SELECT>...</SELECT>`, внутри которого помещается набор тегов, определяющих отдельные пункты. Атрибут `SIZE` задает высоту списка; если его значение равно 1, отображается раскрывающийся список, если же оно больше 1 — обычный. Отдельный же пункт списка создается с помощью парного тега `<OPTION>...</OPTION>`. Название пункта помещается внутрь этого тега, а значение задает атрибут `VALUE`. Атрибут `SELECTED` тега `<OPTION>` позволяет сделать пункт изначально выделенным; этот атрибут не имеет значения — достаточно только его присутствия в теге.

Поле ввода имени файла

Поле ввода имени файла позволит вам отправить серверной программе целый файл. Единственное условие: данные должны быть закодированы с использованием кодировки `multipart/form-data` и отправлены с помощью метода POST.

Поле ввода имени файла внешне представляет собой обычное поле ввода. Справа от него расположена кнопка **Обзор** (Browse), открывающая стандартное диалоговое окно открытия файла Windows. И поле ввода, и кнопка представляют собой один элемент управления.

Для помещения в форму поля ввода имени файла используется кнопка **File Field** (рис. 16.19) панели объектов или пункт **File Field** подменю **Form Objects** меню **Insert**. Доступные параметры показаны на рис. 16.20.

Рис. 16.19. Кнопка **File Field** панели объектов

В поле ввода **FileField Name** вводится уникальное имя элемента управления. Dreamweaver автоматически подставит туда сгенерированное им самим имя, но вам сразу же следует его изменить. Вы также можете выбрать пункт

Name в контекстном меню поля ввода и изменить это имя в появившемся на экране диалоговом окне **Change Attribute**.

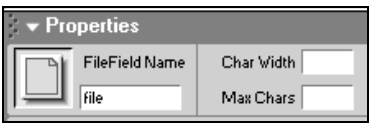


Рис. 16.20. Вид редактора свойств при выделенном поле ввода имени файла

В поле ввода **Char Width** задается ширина поля ввода, т. е. его горизонтальный размер, в символах.

В поле ввода **Max Chars** задается максимальное количество символов, которое может быть введено в данное поле ввода. Это число может быть больше, чем **Char Width**; тогда содержимое поля ввода будет прокручиваться по горизонтали. Но ни в коем случае не задавайте **Max Chars** меньше, чем **Char Width**.

Очень странно, но поля ввода **Init Val**, в котором задается начальное значение имени файла, в этом случае не предусмотрено. Однако вы можете выбрать в контекстном меню поля ввода пункт **Value** и изменить начальное значение в диалоговом окне **Change Attribute**.

Примечание

Поле ввода имени файла создается с помощью одинарного тега `<INPUT TYPE="file">`.

Графическая кнопка

Графическая кнопка — это обычное графическое изображение, "по совместительству" выполняющее функцию кнопки отправки данных, т. е. при щелчке по такому изображению данные, введенные в форму, будут отправлены серверной программе.

Для помещения графической кнопки в форму используется кнопка **Image Field** (рис. 16.21) панели объектов или пункт **Image Field** подменю **Form Objects** меню **Insert**. После этого на экране появится диалоговое окно **Select File**, в котором вы должны будете выбрать файл графического изображения. После нажатия кнопки открытия графическая кнопка будет создана.

Доступные параметры графической кнопки показаны на рис. 16.22.

Рис. 16.21. Кнопка **Image Field** панели объектов

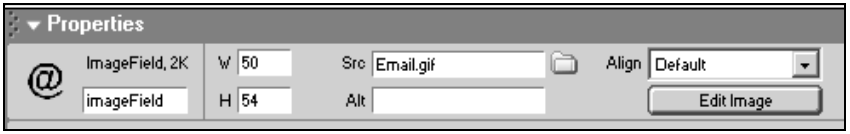


Рис. 16.22. Вид редактора свойств при выделенной графической кнопке

В поле ввода **ImageField** вводится уникальное имя элемента управления. Dreamweaver автоматически подставит туда сгенерированное им самим имя, но вам сразу же следует его изменить. Вы также можете выбрать пункт **Name** в контекстном меню графической кнопки и изменить это имя в появившемся на экране диалоговом окне **Change Attribute**.

Графическая кнопка отправляет серверной программе целых две пары значений вида:

```
<Имя графической кнопки>.x=X
<Имя графической кнопки>.y=Y
```

где `X` и `Y` — координаты точки на графической кнопке, на которой пользователь щелкнул мышью.

В полях ввода **W** и **H** вводятся, соответственно, ширина и высота графической кнопки в пикселах. Dreamweaver сам рассчитывает эти величины и помещает их в названные поля. Если вы хотите уменьшить или увеличить графическую кнопку, измените их вручную. Маркеров изменения размеров Dreamweaver в этом случае почему-то не предусматривает.

В поле ввода **Src** задается имя файла графического изображения. Вы также можете щелкнуть значок папки, расположенный справа от поля ввода, и выбрать нужный файл в диалоговом окне **Select File**. Вызвать это окно вы можете, выбрав пункт **Source File** в контекстном меню графической кнопки.

В поле ввода **Alt** задается "альтернативный" текст, отображаемый вместо графической кнопки, если она еще не загружена, или если пользователь запретил загрузку графических изображений в настройках Web-обозревателя. Вы также можете выбрать в контекстном меню графической кнопки пункт **Alt Text** и задать этот текст в появившемся на экране диалоговом окне **Change Attribute**.

Раскрывающийся список **Align** задает, каким образом графическая кнопка будет выравниваться относительно окружающего ее содержимого страницы. Здесь доступны следующие пункты:

- **Top** — верх кнопки выравнивается по верхнему краю окружающего текста;
- **Middle** — середина кнопки выравнивается посередине окружающего текста;

- **Bottom** — низ кнопки выравнивается по нижнему краю окружающего текста;
- **Left** — кнопка прижимается к левому краю страницы. Все остальное содержимое страницы обтекает кнопку;
- **Right** — кнопка прижимается к правому краю страницы;
- **Browser Default** — выравнивание по умолчанию, как правило, аналогично **Bottom**.

Вообще-то, чтобы получить нужный результат, вам придется поэкспериментировать с различными значениями этого параметра.

Кнопка **Edit Image** позволит вам запустить программу графического редактора для того, чтобы отредактировать изображение. При этом будет запущена программа, зарегистрированная для данного типа графических файлов по умолчанию. Вы также можете выбрать пункт **Edit With <*название программы*>** контекстного меню графической кнопки.

Примечание

Графическая кнопка создается с помощью одинарного тега `<INPUT TYPE="image">`.

Скрытое (невидимое) поле

Скрытое поле — это особый элемент управления, вообще не отображающийся в Web-странице. Значение, изначально присвоенное скрытому полю, не может быть изменено посетителем сайта. Можете считать, что это своего рода "тайные" данные, "спрятанные" в форме и недоступные посетителю.

Скрытое поле обычно используется в формах, генерируемых серверными программами. В частности, если покупатель электронного магазина щелкает по некому товару, серверная программа выдаст ему форму оформления заказа, в которую с помощью скрытого поля "внедрено" имя этого покупателя. После того как другая серверная программа получит данные о заказе, она может сразу же выяснить, кем заказан товар. В статических же страницах полезность скрытого поля весьма сомнительна.

Для помещения в форму скрытого поля используется кнопка **Hidden Field** (рис. 16.23) панели объектов или пункт **Hidden Field** подменю **Form Objects** меню **Insert**. Помещенное в форму скрытое поле показано на рис. 16.24, а очень немногие доступные для него параметры — на рис. 16.25.

Рис. 16.23. Кнопка **Hidden Field** панели объектов

Рис. 16.24. Помещенное на страницу скрытое поле

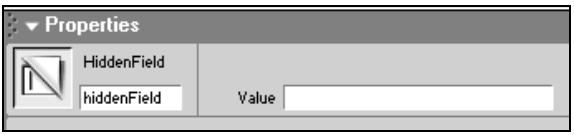


Рис. 16.25. Вид редактора свойств при выделенном скрытом поле

В поле ввода **HiddenField** вводится уникальное имя элемента управления. Dreamweaver автоматически подставит туда сгенерированное им самим имя, но вам сразу же следует его изменить. Вы также можете выбрать пункт **Name** в контекстном меню скрытого поля и изменить это имя в появившемся на экране диалоговом окне **Change Attribute**.

В поле ввода **Value** задается значение, которое будет отправлено скрытым полем серверной программе. Вы также можете выбрать пункт **Value** в контекстном меню скрытого поля и задать значение в появившемся на экране диалоговом окне **Change Attribute**.

Примечание

Скрытое поле создается с помощью одинарного тега `<INPUT TYPE="hidden">`.

Дополнительные элементы управления

Дополнительные элементы управления служат не для ввода данных, а для удобства посетителя страницы. Они поддерживаются только достаточно новыми программами Web-обозревателей, а именно Internet Explorer, начиная с версии 5.0, Navigator, начиная с версии 6.0, и всеми версиями Opera и Mozilla.

Метка

Метка — это просто текстовая подпись, добавляемая к обычному элементу управления, например, полю ввода или переключателю.

Чтобы привязать метку к элементу управления, выделите его и нажмите кнопку **Label** (рис. 16.26) панели объектов или выберите пункт **Label** подменю **Form Objects** меню **Insert**. После этого Dreamweaver переключится в режим отображения содержимого страницы и ее HTML-кода, и в окне документа вы увидите такой код:

```
<LABEL> ...Код элемента управления... </LABEL>
```

Рис. 16.26. Кнопка **Label** панели объектов

Как видите, метка создается с помощью парного тега <LABEL>. HTML-код элемента управления, к которому привязывается метка, находится внутри этого тега. Текст же метки вводится также внутрь этого тега, перед кодом элемента управления или после него. В первом случае текст метки появится перед элементом управления, а во втором — после него.

Например, следующий код поместит на Web-странице метку и — после нее — поле ввода:

```
<LABEL>Поле ввода <INPUT TYPE="text" NAME="textfield"></LABEL>
```

А этот код поместит поле ввода ниже метки:

```
<LABEL>Поле ввода<BR><INPUT TYPE="text" NAME="textfield"></LABEL>
```

Можете проверить — оба этих фрагмента кода действительно работают.

Группа

Группа элементов управления — это простой прямоугольник, объединяющий несколько элементов управления и имеющий текстовую метку.

Чтобы объединить несколько элементов управления в группу, выделите их и нажмите кнопку **Fieldset** (рис. 16.27) панели объектов или выберите пункт **Fieldset** подменю **Form Objects** меню **Insert**. После этого Dreamweaver переключится в режим отображения содержимого страницы и ее HTML-кода и выведет небольшое диалоговое окно **Fieldset**, показанное на рис. 16.28. Введите в единственное поле ввода этого окна текст метки и нажмите кнопку **OK**.

Рис. 16.27. Кнопка **Fieldset** панели объектов

Рис. 16.28. Диалоговое окно **Fieldset**

После этого вы увидите такой код в окне документа:

```
<FIELDSET>
  <LEGEND>Текст метки</LEGEND>
  ...Коды элементов управления...
</FIELDSET>
```

Да, группа создается с помощью парного тега <FIELDSET>, внутри которого помещается HTML-код всех входящих в группу элементов управления. Текст метки помещается внутри особого парного тега <LEGEND>, находящегося внутри тега <FIELDSET>. В зависимости от места, где находится тег <LEGEND>, метка будет помещена вверху или внизу группы.

Внимание!

Элементы управления, помещенные в группу, должны быть окружены тонким прямоугольником. Но Dreamweaver не отображает этот прямоугольник. Чтобы увидеть его, откройте страницу в Web-обозревателе.

Раскрывающийся список гиперссылок

Теперь поговорим еще об одном элементе управления, напрямую не относящемся к формам. Это так называемый *раскрывающийся список гиперссылок* — обычный раскрывающийся список, содержащий список гиперссылок, ведущих на разные страницы текущего или других сайтов. При выборе любой страницы Web-обозреватель сразу же ее загружает и отображает. Фактически это обычный набор гиперссылок, только "свернутый" для достижения компактности.

Раскрывающийся список гиперссылок внешне ничем не отличается от обычного раскрывающегося списка. Вся его функциональность реализуется с помощью набора особых поведений (фактически — сценариев, написанных на языке JavaScript и помещенных в коде страницы). Эти поведения автоматически формируются самим Dreamweaver.

Вряд ли можно заменить раскрывающимся списком гиперссылок обычную полосу навигации на главной странице сайта. Одна из основных задач полосы навигации: представить посетителю список всех разделов сайта, чтобы он сразу выбрал то, что ему нужно. Раскрывающийся список, к сожалению, показывает одновременно только один пункт; чтобы увидеть все пункты списка, его придется развернуть. Поэтому он не может заменить старой доброй полосы навигации.

Однако в качестве дополнительного инструмента навигации по сайту раскрывающийся список может очень даже пригодиться, особенно если полоса навигации находится только на главной странице вашего сайта. В этом случае вы можете поместить такой список в углу каждой страницы вашего сайта — он не будет занимать слишком много места и излишне привлекать внимание. А посетитель вашего сайта сможет быстро перейти на любую его страницу, не возвращаясь на главную страницу, а просто воспользовавшись этим списком.

Как создается раскрывающийся список гиперссылок? Очень просто, как все в Dreamweaver.

Создайте новую Web-страницу. Переключитесь на вкладку **Forms** панели объектов и щелкните кнопку **Jump Menu** (рис. 16.29). Вы также можете выбрать пункт **Jump Menu** подменю **Form Objects** меню **Insert**. После этого на экране появится диалоговое окно **Insert Jump Menu**, показанное на рис. 16.30.

Рис. 16.29. Кнопка **Jump Menu** панели объектов

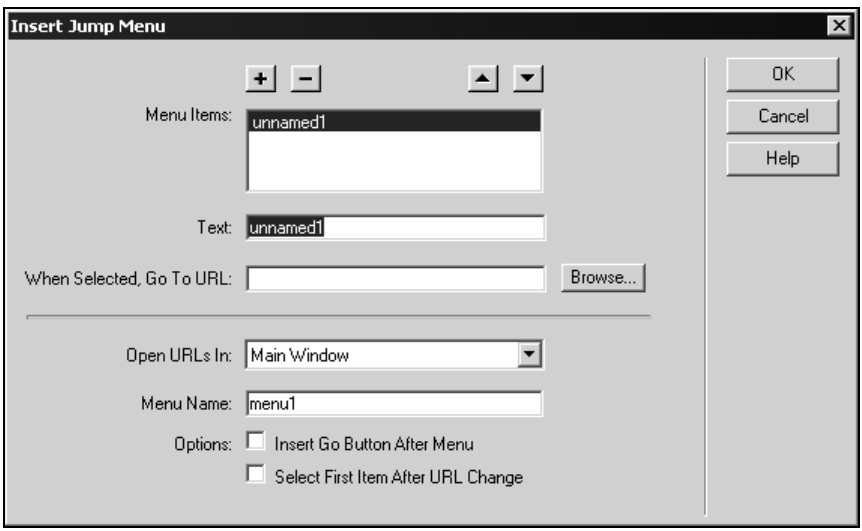


Рис. 16.30. Диалоговое окно **Insert Jump Menu**

В списке **Menu Items** перечислены все пункты раскрывающегося списка, созданные вами. Вы можете добавлять в этот список новые строки, щелкнув кнопку со знаком "плюс", и удалять ненужные, выбрав их и нажав кнопку со знаком "минус". Кнопки ↑ и ↓ позволят переместить выбранную строку вверх или вниз на одну позицию.

В поле ввода **Text** задается название пункта списка, иначе говоря, текст гиперссылки. К несчастью, и в этом случае Dreamweaver демонстрирует свою русофобию, искажая русские имена. Адрес же гиперссылки вводится в поле ввода **When Selected, Go To URL**. Вы также можете щелкнуть кнопку **Browse**, расположенную правее этого поля ввода, и выбрать нужный файл в появившемся на экране диалоговом окне **Select File**.

С помощью раскрывающегося списка **Open URLs In** вы можете задать, где будут отображаться страницы, вызванные с помощью списка гиперссылок, т. е. цель гиперссылки. Пункт **Main Window** заставит список гиперссылок

выводить их в основном окне Web-обозревателя. Если ваш сайт построен с использованием фреймов, в меню **Open URLs In** будут перечислены также названия всех созданных вами фреймов.

В поле ввода **Menu Name** вводится уникальное имя списка гиперссылок. Автор вам советует сразу же ввести в него какое-нибудь "вразумительное" имя, наподобие `JumpMenu`.

Обычно переход по соответствующей гиперссылке будет осуществлен сразу же после того, как посетитель сайта выберет соответствующий пункт списка гиперссылок. Однако вы можете добавить справа от списка кнопку **Go** (Перейти), и для перехода по гиперссылке посетитель после выбора нужной гиперссылки должен будет нажать на эту кнопку. Для этого включите флажок **Insert Go Button After Menu**, расположенный в группе **Options**. Правда, после этого вам придется вручную изменить надпись на этой кнопке на "Перейти".

Зачем это может понадобиться? Представьте себе такую ситуацию. Посетитель сайта выбрал какую-либо гиперссылку и перешел на соответствующую страницу. После этого он нажимает кнопку **Назад (Back)** на панели инструментов Web-обозревателя и возвращается обратно, на страницу со списком гиперссылок. Если теперь он снова захочет попасть на ту же самую страницу, у него ничего не получится: в списке гиперссылок будет выбрана та же гиперссылка. Если выбрать ее снова, список гиперссылок не сработает. Вот именно для этого и может пригодиться кнопка **Перейти**. Посетителю достаточно будет щелкнуть по ней еще раз, чтобы попасть на ту же страницу.

Есть еще один способ решения этой проблемы: заставить каждый раз при попадании на страницу со списком гиперссылок делать первую его строку выбранной автоматически. (Часто в этой первой строке помещают небольшую подсказку вида "Выберите гиперссылку...".) Для этого создайте первую строку соответствующего вида и включите флажок **Select First Item After URL Change**, расположенный все в той же группе **Options**.

Закончив, нажмите кнопку **OK**. Раскрывающийся список гиперссылок, созданный нами, будет помещен на страницу.

Уже было сказано, что список гиперссылок не относится к формам и совсем не обязательно должен находиться в форме. Однако, если вы поместите его не в форме, Dreamweaver все равно создаст форму специально для него. Это служит для того, чтобы список работал в Navigator, нормально воспринимающем элементы управления только в составе форм.

Чтобы изменить пункты раскрывающегося списка гиперссылок, можно воспользоваться уже знакомым вам диалоговым окном **List Values**, показанном на рис. 16.18. (Кстати, если вы хотите задать для пунктов списка русские имена, вы можете сделать это в этом же окне.) Чтобы его вызвать, выделите список и нажмите кнопку **List Items** редактора свойств (см. рис. 16.17). Также вы можете выбрать в контекстном меню списка пункт **List Items**.

Поведения, предназначенные для работы с формами

Dreamweaver предусматривает несколько поведений, предназначенных для работы с формами и элементами управления. (Подробнее о поведениях и работе с ними говорилось в *главе 13*.) Сейчас мы их рассмотрим.

Создание списка гиперссылок *(Jump Menu)*

Это поведение используется при создании раскрывающегося списка гиперссылок. Именно оно обеспечивает переход на другую Web-страницу, когда посетитель выбирает соответствующий пункт в этом списке.

Когда вы создаете список гиперссылок, Dreamweaver сам создает поведение `Jump Menu`. Но если вы хотите преобразовать в список гиперссылок уже существующий список, вам придется создавать его вручную. Для этого выделите нужный список в окне документа и выберите в меню поведений пункт **Jump Menu**. После этого на экране появится уже знакомое вам диалоговое окно **Insert Jump Menu**, в котором вы сможете задать пункты списка гиперссылок.

Создание кнопки перехода для списка гиперссылок *(Jump Menu Go)*

Это поведение тоже используется для создания списка гиперссылок, но уже с кнопкой **Перейти**. Как и в предыдущем случае, вам также обычно не нужно создавать его самим — это сделает за вас Dreamweaver. Другое дело, если вы хотите "приделать" кнопку **Перейти** к уже существующему списку гиперссылок. В этом случае создайте кнопку, выделите ее и выберите в меню поведений пункт **Jump Menu Go**. (Список гиперссылок уже должен быть создан.) На экране появится диалоговое окно **Jump Menu Go**, показанное на рис. 16.31. Все, что вам нужно сделать, — это выбрать в раскрывающемся списке **Choose Jump Menu** нужный список гиперссылок и нажать кнопку **OK**.

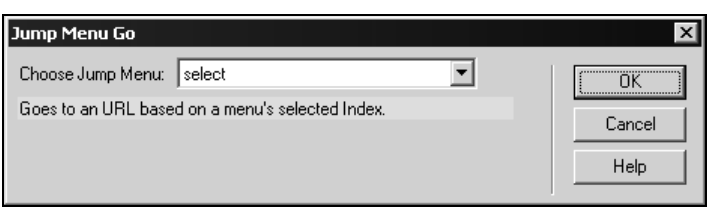


Рис. 16.31. Диалоговое окно **Jump Menu Go**

Задание нового значения поля ввода *(Set Text of Text Field)*

Поведение `Set Text of Text Field` позволяет вам поместить новое значение в любое поле ввода или любую область редактирования из имеющихся на странице. Чтобы создать его, выберите пункт **Set Text of Text Field** в подменю **Set Text** меню поведений. На экране появится диалоговое окно **Set Text of Text Field**, показанное на рис. 16.32.

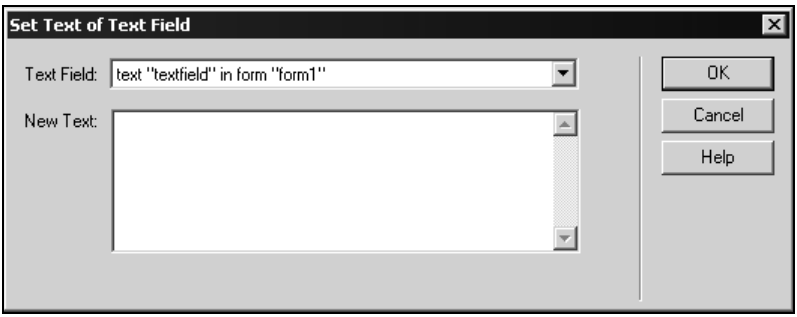


Рис. 16.32. Диалоговое окно **Set Text of Text Field**

В раскрывающемся списке **Text Field** выбирается поле ввода или область редактирования, куда вы хотите поместить новое значение. Само это значение вводится в область редактирования **New Text**. Задав нужные настройки, нажмите кнопку **OK**.

Вы можете использовать в тексте нового значения любой JavaScript-код, заключив его в фигурные скобки. Например, чтобы поместить в поле ввода текущую дату, вы можете использовать такой код:

```
Сегодня {new Date()}
```

Если же вам понадобится ввести в текст фигурную скобку, предварите ее обратной косой чертой (`/{` или `/}`).

Проверка данных, введенных в форму *(Validate Form)*

Очень часто нужно проверять данные, введенные посетителем страницы в форму, на правильность. Сами понимаете, компьютеры ненадежны, но люди еще ненадежнее (один из законов Мерфи).

Для этого Dreamweaver предоставляет поведение `Validate Form`. С его помощью можно проверять введенные данные на правильность: являются ли они адресом электронной почты, числом и введены ли вообще. Это поведение обычно привязывается к событию `onSubmit` тега `<FORM>` и срабатывает,

когда посетитель страницы нажимает кнопку отправки данных. Но оно может также привязываться и к событиям `onBlur` или `onChange` тега `<INPUT>`, происходящим, когда посетитель переключается на другой элемент управления или изменяет данные в поле ввода, соответственно. В первом случае введенные данные проверяются "скопом", во втором — после ввода каждого значения. На практике применяются оба подхода; сами решайте, какой из них использовать. Если вы хотите привязать поведение к форме, выберите элемент формы, если к полю ввода — нужное поле ввода. Выберите одноименный пункт меню поведений. И не забудьте выбрать необходимое событие в меню колонки **Events** списка. После этого можно вводить нужные данные в диалоговое окно **Validate Form** (рис. 16.33).

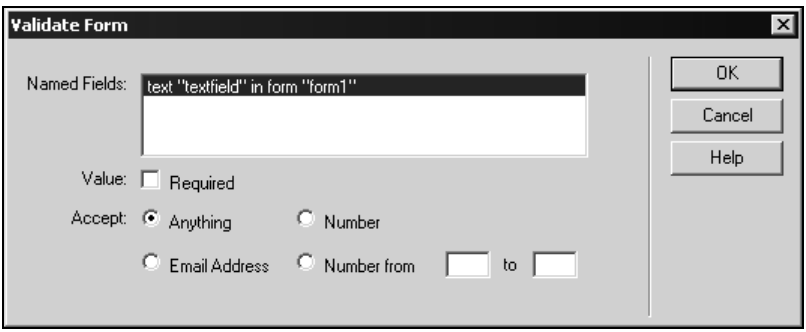


Рис. 16.33. Диалоговое окно **Validate Form**

В списке **Named Fields** этого диалогового окна перечислены все поля ввода, созданные на странице. Вы можете выбрать любое из них и задать для него нужную проверку.

Группа элементов управления **Value** позволяет задать, должен ли пользователь ввести в выбранное поле ввода значение. В эту группу входит единственный элемент управления — флажок **Required**. Если он включен, будет выполняться проверка, введено ли в поле ввода значение. Если же он отключен, такая проверка выполняться не будет.

С помощью группы переключателей **Accept** вы можете задать, какого вида значение должно быть введено в поле ввода. Всего переключателей четыре:

- **Anything** (включен по умолчанию) — любое значение;
- **Email Address** — адрес электронной почты;
- **Number** — любое число;
- **Number from ... to ...** — число в заданном диапазоне. Нижнее значение этого диапазона задается в поле ввода **from**, верхнее — в поле ввода **to**.

Если вы выберете любой переключатель в этой группе, кроме **Anything**, Dreamweaver создаст Web-сценарий, проверяющий введенные в поле ввода

данные на правильность. Если данные такую проверку не пройдут, сценарий выведет окно-предупреждение, предлагающее посетителю страницы исправить ошибку.

Если вы привязываете это поведение к форме, вы можете в этом же диалоговом окне задать проверку для нескольких полей ввода. Для этого просто выберите другое поле ввода в списке **Named Fields** и задайте нужные критерии проверки. Если вы привязываете поведение к полю ввода, вы можете задать проверку только для этого поля ввода.

Введя нужные данные, не забудьте нажать кнопку **OK**.

Простейшая Web-форма

Для примера давайте создадим небольшую форму для опроса посетителей нашего сайта. В нее будут вводиться следующие данные о посетителе:

- имя;
- фамилия;
- пол;
- год рождения;
- стаж работы в Интернете (новичок, опытный пользователь, гуру).

Для пробной формы этого больше чем достаточно. Формы, имеющиеся в настоящее время в Сети, собирают значительно больше информации (в частности, адрес электронной почты, дату и месяц рождения, иногда — домашний адрес и т. п.), но нам в данном случае важнее научиться работать с формами в Dreamweaver.

Конечно, наша форма будет содержать обязательную кнопку **Отправить** и кнопку **Сброс**. В последней, правда, особой нужды нет, но все же давайте ее сделаем.

В качестве серверной программы зададим program.exe. Отправлять данные будем с использованием метода GET. Думается, 245 символов (256 минус длина слова "program.exe") для этого хватит. Конечно, если вы сомневаетесь в этом, можете воспользоваться методом POST.

Какие же элементы управления мы используем в нашей форме? Внимательно посмотрим на список данных, которые мы намереваемся собрать, и хорошенько подумаем. Имя, фамилия и год рождения представляют собой текст, различных вариантов ввода может быть бесконечно много, значит, для них используем поля ввода. Различных полов на этой планете существует всего два, поэтому используем набор из двух переключателей. Что касается стажа работы в Интернете, то здесь также имеет место ограниченный набор вариантов выбора, поэтому можно, в принципе, использовать группу

переключателей. Однако, чтобы не "раздувать" размеры нашей формы, применим раскрывающийся список, который значительно компактнее.

Решив, что мы хотим получить в результате, приступим к работе.

Создайте новую страницу. Если хотите, задайте ее параметры (название, цвета текста и фона). Можете добавить заголовок и какой-нибудь поясняющий текст. И сохраните под именем, например 16.2.htm.

Теперь поместим на свежесозданную страницу форму. Для этого воспользуемся кнопкой **Form** (см. рис. 16.1) вкладки **Forms** панели объектов или пунктом **Form** меню **Insert**. Пустая форма выглядит точно так, как показано на рис. 16.2. Не очень вдохновляет, правда?

Теперь зададим параметры нашей формы. В поле ввода **Form Name** редактора свойств введите имя формы `VisitorData`. В поле ввода **Action** задайте имя серверной программы `program.exe`. В раскрывающемся списке **Method** выберите пункт **GET**, а в раскрывающемся списке **Enctype** — пункт **application/x-www-form-urlencoded**. Все, наша форма готова. Можно приступать к наполнению ее содержимым.

Поставьте текстовый курсор внутрь формы. Наберите строку "Имя" и поставьте после него пробел. После этого текста поместите поле ввода. (Для этого либо нажмите кнопку **Text Field** панели объектов, либо выберите пункт **Text Field** подменю **Form Objects** меню **Insert**.) У вас должно получиться то же самое, что показано на рис. 16.34.

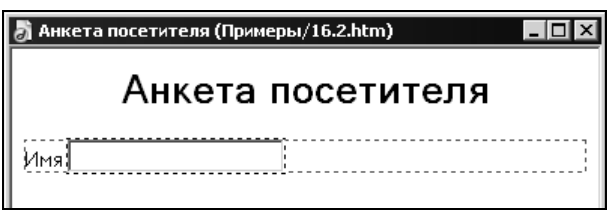


Рис. 16.34. Надпись и поле ввода имени, расположенные в одну строку

Не очень красиво... Но мы сейчас это исправим. Поместите текстовый курсор между надписью и полем ввода, уберите пробел и вставьте разрыв строки, нажав клавиши <Shift>+<Enter>. Посмотрите на то, что у нас получилось (рис. 16.35).

Задайте имя поля ввода `Name1`, максимальное количество символов 40. Длину поля ввода можете задать любой, следите только, чтобы она не превысила заданного нами максимального количества символов (40).

Теперь поставьте текстовый курсор после поля ввода имени и поставьте два разрыва строки, нажав комбинацию клавиш <Shift>+<Enter> дважды. Наберите слово "Фамилия", поставьте разрыв строки и поместите еще одно поле ввода, в которое будет вводиться фамилия. Задайте для него имя `Name2`

и максимальное количество символов также 40. И поставьте еще два разрыва строки. Результат показан на рис. 16.36.

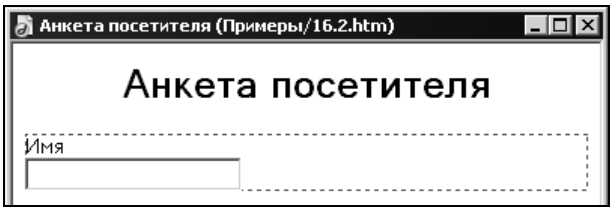


Рис. 16.35. Надпись и поле ввода имени, расположенные в две строки

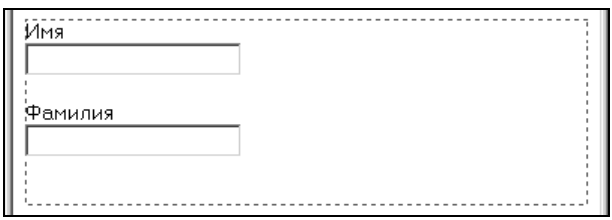


Рис. 16.36. Поля ввода имени и фамилии

Настала пора группы переключателей. Поставьте текстовый курсор после поля ввода фамилии. Наберите слово "Пол", поставьте разрыв строки и поместите на страницу первый переключатель набора. Сразу же после него наберите слово "Муж." (подпись переключателя) и поставьте два или три пробела, чтобы отделить его от другого переключателя. После этого поставьте второй переключатель, наберите слово "Жен." и поставьте два разрыва строки. Вот и все.

Выделите первый переключатель и задайте его параметры: имя набора — `Gender`, значение — `M`, включен изначально. У второго переключателя будут такие параметры: имя набора то же, что у первого, значение — `F`. (Вы можете сделать изначально выделенным второй переключатель — это не принципиально.) Результат показан на рис. 16.37.

Далее поместите в форму поле ввода года рождения. Вы уже знаете, как это делается. Задайте имя этого поля `YoB` (Year of Birth — год рождения), длина поля и максимальное количество символов 4. Можете также задать значение по умолчанию, например `1970`.

Настал черед раскрывающегося списка стажа работы в Интернете. Нет смысла подробно рассказывать, как это сделать, — действуйте по аналогии. Задаете имя раскрывающегося списка `Work`, в диалоговое окно **List Values** введите пункты "`Новичок`", "`Опытный пользователь`" и "`Гуру`" и присвойте им в качестве значений номера от 1 до 3. Изначально выбранным сделайте первый пункт ("Новичок").

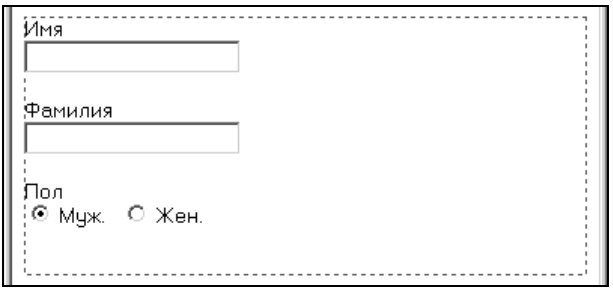


Рис. 16.37. Группа переключателей задания пола

Осталось поместить в форму кнопки **Отправить** (Submit) и **Сброс** (Reset). Сделайте это. Задайте для них имена `Submit` и `Reset` и надписи "Отправить" и "Сброс", соответственно.

Готовая форма изображена на рис. 16.38. Сохраните ее, откройте в Web-обозревателе и попробуйте ввести в нее какие-нибудь данные. Как видите, она работает.

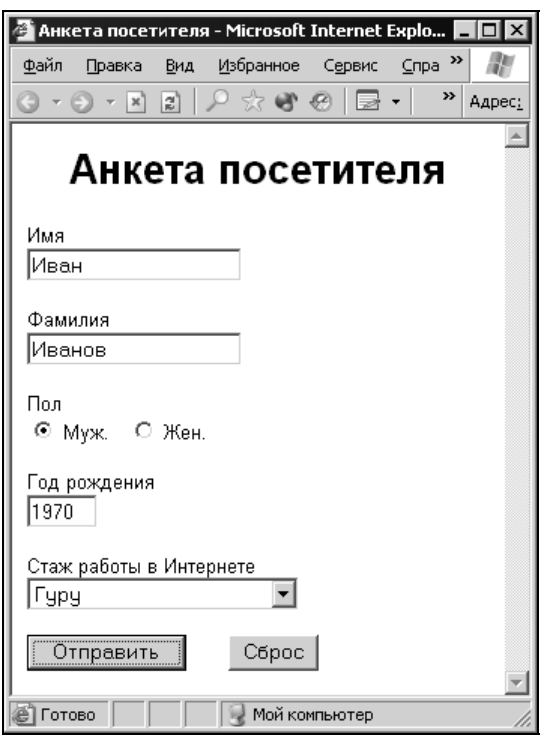


Рис. 16.38. Готовая форма в окне Web-обозревателя

Единственная проблема — некому обрабатывать введенные нами данные. Ну нет у нас программы program.exe, нет!

Использование таблиц и стилей для создания форм

Первая наша форма получилась не очень привлекательной на вид. Это естественно: ведь нашей задачей было научиться создавать формы, а не осваивать хитрости их форматирования. Но теперь-то мы умеем создавать формы и хотели бы научиться делать их покрасивее. Для этого можно использовать мощнейшие средства — таблицы и стили. Таблицы позволят нам располагать элементы управления в формах так, как нам необходимо, а стили заставят их выглядеть так, как нам нужно.

Давайте переделаем нашу форму так, чтобы не стыдно было поместить ее в Сеть. Сохраним предыдущий, "некрасивый" вариант и создадим новую страницу под названием 16.3.htm. Поместим в нее форму и зададим значения ее параметров, как у предыдущей формы, после чего на время оставим ее в покое.

Итак, что же мы хотим получить?

Давайте сделаем нашу новую форму в виде реальной печатной анкеты. Пусть она будет содержать две колонки: в левой будут помещаться надписи, в правой — элементы управления для ввода соответствующих данных. Такая форма будет неплохо выглядеть. Кроме того, у нас появилась фантазия сделать элементы управления инверсными — светлый текст на темном фоне.

Первое реализуется с помощью таблицы. Второе — посредством стилей.

Всего у нас шесть элементов управления (группу переключателей будем считать за один элемент управления, так же, как и пару кнопок **Отправить** (Submit) и **Сброс** (Reset)). Значит, нам понадобится таблица из шести строк и двух столбцов. Поместим такую таблицу в форму. Сделаем ей невидимую границу, а остальные параметры пока оставим без изменения. Можно также задать для таблицы значение ширины, отличное от 100%, либо убрать его совсем. Полученный результат показан на рис. 16.39.

Дальше уже совсем просто. В ячейки левого столбца помещаем надписи, а в ячейки правого — соответствующие им элементы управления. Вместо того чтобы создавать все заново, вы можете копировать надписи и элементы управления со страницы 16.2.htm, открыв ее в отдельном окне документа. В последнюю строку поместим кнопки: в левую ячейку — кнопку **Отправить**, а в правую — **Сброс**. Чтобы как-то их выделить, можете для последней строки таблицы задать выравнивание по правому краю.

Готовая форма будет выглядеть так, как показано на рис. 16.40.

Рис. 16.39. Таблица в форме

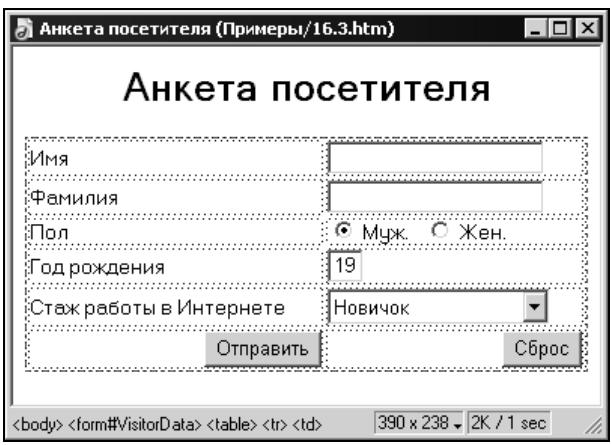


Рис. 16.40. Готовая форма, созданная на основе таблицы

Как видите, форма, построенная на основе таблицы, получилась значительно компактнее нашей первой формы. В этом проявилось одно из главнейших преимуществ таблиц — они позволяют представить объемные данные в удобном виде на небольшой площади.

Ну а уж задать "негативное" изображение для элементов управления проще простого! Создадим внутреннюю таблицу стилей и переопределим в ней с помощью специальных стилей теги `<INPUT>` (поля ввода и переключатели) и `<SELECT>` (раскрывающийся список). Как это делается, вы уже знаете.

Здесь есть одна тонкость. После того как вы переопределите теги `<INPUT>` и `<SELECT>`, изображение в окне документа не изменится. Почему-то Dreamweaver не отображает результат применения стилей к этим тегам. Поэтому, чтобы просмотреть окончательный результат, сохраните страницу 16.3.htm и откройте ее в Web-обозревателе. У вас должно получиться нечто, похожее на рис. 16.41.

Вот и все о формах и элементах управления.

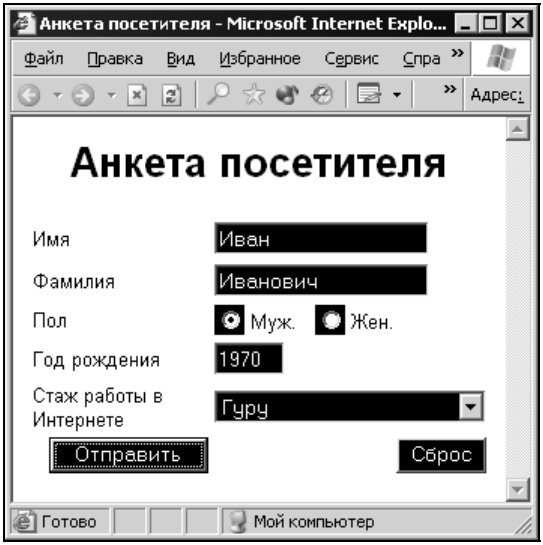


Рис. 16.41. Готовая форма, созданная на основе таблицы и стилей

Основные принципы разработки форм

Итак, вы познакомились с формами и элементами управления, выяснили, как эти формы создаются и используются. Настало время завершить разговор о формах некоторыми общими рекомендациями. А касаются эти рекомендации создания привлекательного и удобного интерфейса пользователя. Иначе говоря, привлекательных и удобных Web-форм.

Эти рекомендации пригодны не только для Web-форм, но и для обычных Windows-приложений. Фирма Microsoft, разработчик Windows, составила целый набор таких рекомендаций, где все высчитано и вымерено; доступны эти рекомендации на сайте **http://www.microsoft.com**. Однако нет смысла приводить их здесь: данные рекомендации слишком объемны, чтобы поместить их все в одну не слишком толстую книгу, вдобавок их без проблем можно найти в Сети. Ограничимся только несколькими замечаниями.

И еще раз повторимся: смотрите на реальные формы, реальные Windows-приложения. И учитесь на готовых примерах.

- Элементы управления в форме должны располагаться на достаточном расстоянии друг от друга, чтобы форма не выглядела скученной. Однако это расстояние не должно быть слишком большим, чтобы форма была компактной.
- Элементы в форме должны располагаться парами "надпись — элемент управления" по вертикали. (Если форма невелика, можно разместить их

и по горизонтали.) Расстояние между этими парами должно быть несколько больше, чем между надписью и элементом управления. Учтите, что группа переключателей считается в этом случае одним элементом управления.

- Объединяйте элементы управления, задающие однотипные данные, в группы. Например, вы можете выделить в отдельную группу поля ввода имени и фамилии посетителя.
- Кнопки **Отправить** (Submit) и **Сброс** (Reset) должны располагаться отдельно от других элементов управления, чтобы посетитель случайно на них не нажал. Неплохо выделить их визуально, скажем, оформить как отдельную группу. Кроме того, расстояние между этими кнопками должно быть достаточно велико, чтобы посетитель случайно не нажал одну кнопку вместо другой.
- Сведите количество элементов управления в форме к минимуму. Например, подумайте, нужны ли отдельные поля ввода для задания имени и фамилии посетителя, не обойтись ли одним полем ввода? Нужна ли кнопка **Сброс**? (Если форма состоит из одного-двух элементов управления, так и не нужна.) Это позволит также уменьшить объем пересылаемых по Сети данных.
- Форма должна быть визуально отделена от остального содержимого страницы. Задайте для нее особый фон или поместите ее в рамку. Если форма достаточно велика, вынесите ее на отдельную страницу.
- Сделав форму, попробуйте сами ввести в нее данные. Определите, удобно ли вам. Если неудобно, переделайте. Только после этого публикуйте форму в Интернете.

И еще несколько рекомендаций, посвященных выбору элемента управления, наиболее подходящего для ввода каких-либо данных.

- Поля ввода прекрасно подходят для ввода данных, количество различных вариантов которых бесконечно велико (имени и фамилии, адреса электронной почты, домашние адреса и т. п.).
- Флажки замечательно подходят для задания признаков, которые либо могут быть, либо нет (классический пример — вопрос, хочет ли посетитель получать по электронной почте сводку новостей сайта или нет).
- Группы переключателей пригодятся в том случае, если нужно предоставить посетителю возможность выбирать один вариант из нескольких имеющихся, причем он должен видеть все представленные варианты выбора одновременно (пол, социальный статус и т. п.).
- Используйте списки, если количество вариантов выбора достаточно велико и группа переключателей в этом случае заняла бы слишком много

места на форме (любимый телеканал, фирма-производитель принтера и т. п.).

- Используйте раскрывающиеся списки, если размер формы критичен, а посетителю необязательно видеть все представленные варианты (те же самые случаи).

Нет смысла говорить здесь о кнопках, обычных и графических, полях ввода файла и скрытых полях. У них достаточно специфическая область использования, и разговор на эту тему уже был.

Что дальше?

Вот вы и узнали, как делаются формы. Ничего сложного! Нужно только создать форму и правильно выставить все параметры ее элементов.

Но что-то автор все-таки забыл... Ах да — еще нужно написать серверную программу! Проклятый склероз... Ладно, это уже тема для следующей главы.

ГЛАВА 17

Простейшие серверные приложения

Дождались! Свершилось! Ура!!!

Мы будем писать серверные программы! Долго, очень долго мы шли к этому счастливому моменту. Изучали язык HTML, принципы Web-дизайна, таблицы стилей, формы и основы серверного программирования. Но все это в прошлом. Теперь мы готовы к будущему.

Мы полны сил. Нами движут лучшие помыслы. Наш Dreamweaver рвется в бой. Наш Web-сервер установлен и правильно настроен. (О том, как настраивать Microsoft Personal Web Server и Internet Information Server, читайте в поставляемой с ними документации.) Наш "статичный" Web-сайт давно работает и пользуется огромной популярностью. Наш кофе давно сварен и медленно остывает в чашке. Все готово к тому, чтобы сделать первый шаг в серверное программирование.

Ух, что мы сейчас сделаем! Какие мечты воплотим в жизнь! Мечтать — так мечтать, как говорил один из героев Александра Грина.

Ну, что ж, начнем, пожалуй. Только сначала подготовим почву для экспериментов. Давайте создадим гостевую книгу для нашего первого сайта Sample site 1. Сделаем так, чтобы посетители его смогли оставлять на особой страничке свои замечания и пожелания. Соответственно, для этого нам понадобится создать еще две Web-страницы: страницу, где выполняется ввод пожеланий и замечаний, и страницу, где уже введенные пожелания и замечания будут отображаться. Разумеется, это будут активные серверные страницы.

Подготовка к созданию серверных приложений

Давайте еще раз вспомним, что нам нужно для создания серверных приложений (точнее, активных серверных страниц) в среде Dreamweaver.

1. Полностью настроенный Web-сервер. Учтите, что, возможно, вам придется включить в его настройках поддержку серверных страниц. Хотя, если вы пользуетесь одним из Web-серверов фирмы Microsoft, вам не нужно это делать — в их настройках это включено по умолчанию.
2. База данных, если ваше приложение будет работать с данными. Возможно, вам придется заполнить ее какими-то данными, хотя бы для отладки создаваемых вами серверных приложений.
3. Правильно занесенные в Dreamweaver сведения о Web-сайте.

Итак, что мы уже имеем?

Как говорилось ранее, мы имеем настроенный Web-сервер. (Не будем описывать его установку и настройку. Учитесь читать инструкции — хороший системный администратор просто обязан это уметь.) Он нормально работает со статичными Web-страницами, а большего от него в данный момент и не требуется.

Вообще-то Web-сервер для разработки серверных страниц не очень-то и нужен. Чтобы просто "рисовать" страницы, вам будет достаточно только Dreamweaver. Web-сервер понадобится, когда вы начнете тестировать работу ваших страниц, или если вы захотите воспользоваться режимом просмотра "живых" данных Dreamweaver (об этом режиме мы поговорим позже). Так что пока можете особо не торопиться с Web-сервером (хотя и мешкать с его установкой тоже не стоит).

Теперь о базах данных. Ими мы займемся потом, чуть позже. Предположим, что вы умеете пользоваться какой-либо СУБД (системой управления базами данных), например Microsoft Access. Именно Access мы и будем пользоваться для создания баз данных, благо это лучшая на сегодняшний момент клиентская СУБД. Впрочем, если вы предпочитаете другую программу, можете пользоваться ей.

И, напоследок, о регистрации Web-сайта в Dreamweaver.

Когда вы регистрируете сайт, Dreamweaver предполагает, что он не будет содержать активные серверные страницы. Это правильно — в конце концов, не все сайты используют серверные программы в том или ином виде. Поэтому изначально стоит скрыть возможности по их созданию, если Web-дизайнер специально не "попросит".

Давайте же "попросим" Dreamweaver разрешить нам использовать в нашем сайте серверные страницы. Точнее, потребуем. Да так, чтобы он не смог отказаться.

"Просьба" наша будет заключаться в том, чтобы задать некоторые дополнительные настройки нашего сайта Sample site 1. Выберем его в списке панели **Site** и вызовем диалоговое окно **Site Definition**. Как это сделать, подробно описано в *главе 6*.

Прежде всего, переключимся на вкладку **Local Info** и запишем в поле ввода **HTTP Address** интернет-адрес нашего сайта. Поскольку Web-сервер, на котором он будет опубликован, находится на нашем же компьютере, введем в это поле ввода строку `http://localhost/`. Это обозначение локального Web-сервера, работающего на нашем компьютере.

Далее переключимся на вкладку **Testing Server**. То, что мы увидим, показано на рис. 17.1.

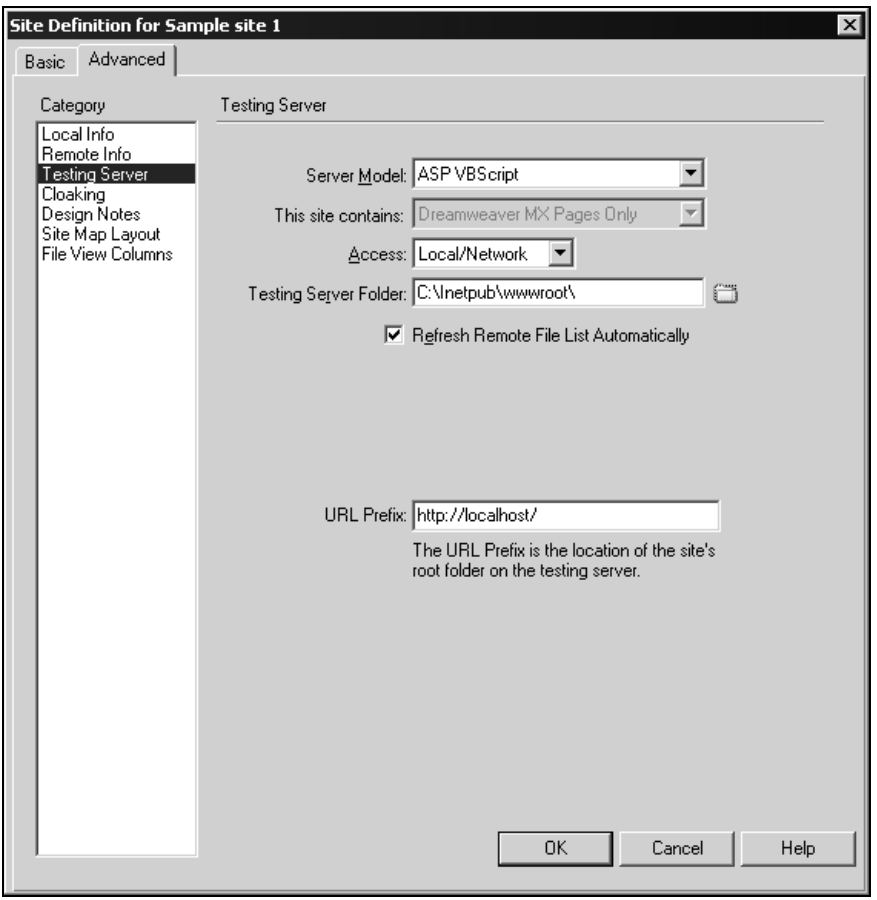


Рис. 17.1. Диалоговое окно **Site Definition** (вкладка **Testing Server**)

С помощью раскрывающегося списка **Server Model** задается используемая нами технология серверных страниц. Здесь доступны следующие пункты:

- **None** — технология серверных страниц не используется (значение по умолчанию);

- **ASP JavaScript** — используется Microsoft ASP и язык программирования JavaScript;
- **ASP VBScript** — используется Microsoft ASP и язык программирования VBScript;
- **ASP.NET C#** — используется Microsoft ASP.NET и язык программирования C#;
- **ASP.NET VB** — используется Microsoft ASP.NET и язык программирования VBScript;
- **ColdFusion** — используется Macromedia ColdFusion;
- **JSP** — используется Netscape JSP;
- **PHP MySQL** — используется PHP совместно с сервером баз данных MySQL.

Выберите в этом списке пункт **ASP JavaScript**.

Раскрывающийся список **This site contains** доступен только в случае, если в списке **Server Model** выбран пункт **ColdFusion**. Мы не будем его здесь рассматривать.

В раскрывающемся списке **Access** выбирается способ отправки серверных страниц Web-серверу. В нем доступны три пункта:

- **None** — серверные страницы не отправляются Web-серверу;
- **FTP** — отправка страниц по протоколу FTP;
- **Local/Network** — отправка страниц по локальной сети. Также выбирается, если Web-сервер работает на клиентском компьютере, как в нашем случае.

Выберите пункт **Local/Network**.

Остальные элементы управления были описаны в *главе 6*. Поэтому мы не будем на них останавливаться.

Задав нужные параметры, нажмите кнопку **OK**. Все, теперь Dreamweaver готов к работе с серверными страницами.

Установление соединения с базой данных

Следующее, что нам нужно сделать, — это установить соединение с базой данных. Это надо для того, чтобы Dreamweaver смог создать необходимые для работы с данными сценарии. (Имейте в виду, что в данном случае речь идет о серверных сценариях.) Процесс соединения с базой данных мы рассмотрим пошагово.

Но сначала нам нужно подготовить саму базу данных, где будут храниться записи нашей гостевой книги. Создайте ее в Microsoft Access. Поместите

в нее единственную таблицу `Guestbook`, содержащую единственное текстовое поле `Content`. Сам файл базы данных назовите так же, как таблицу — Guestbook.mdb. Поместите ее в какую-либо папку, но не в папку, где хранятся файлы локальной копии вашего сайта, иначе Dreamweaver опубликует ее на Web-сервере "за компанию" с Web-страницами и прочими файлами.

Внимание!

Никогда не публикуйте файлы базы данных на Web-сервере, особенно если она содержит секретные данные! Если вы все же это сделаете, любой достаточно грамотный пользователь Интернета сможет загрузить ее на свой компьютер и прочитать все, что в ней хранится. База данных не должна быть доступна извне!

Внимание!

Пользователям операционных систем Windows NT, 2000 и XP. При установке Web-сервер Internet Information Server создает пользователя под именем `IUSR_<сетевое имя вашего компьютера>`. Проверьте, имеет ли этот пользователь полные права для доступа к папке, где хранится файл вашей базы данных, иначе при попытке доступа к ней из серверной страницы вы получите сообщение об ошибке. (Учтите — ко всей папке, а не к самому файлу базы данных!) Как дать пользователю права доступа к папке, описано в документации Windows.

Создание источника данных ODBC

Для доступа к базам данных Web-серверы фирмы Microsoft, да и многие другие программы, используют механизм *ODBC* (Open DataBase Connectivity — открытое соединение с базой данных). ODBC — универсальный механизм доступа к базам данных любого формата, встроенный в операционные системы Windows. А раз он встроен, то грех им не воспользоваться, тем более что он предоставляет достаточно мощные средства для извлечения данных и управления ими.

Чтобы получить доступ к данным с помощью ODBC, сначала нужно создать так называемый *источник данных* ODBC — своего рода "пропуск" к базе данных, которым будут пользоваться использующие ее программы. Источник данных ODBC содержит сведения о том, где находится файл или файлы базы данных, и в каком формате она (база данных) хранится. Кроме того, там же содержатся имя и пароль пользователя, используемые для подключения к серверу данных, и некоторые другие необязательные параметры.

Сейчас давайте рассмотрим, как создается источник данных ODBC.

Само собой, здесь приводится только краткое описание и уж тем более не показаны все параметры, хранящиеся в источнике. Если вы хотите узнать

больше о ODBC, поищите соответствующую документацию на сайте **http://msdn.microsoft.com**. А здесь ограничимся только кратким описанием, без которого в данном случае нам не обойтись, и опустим все "излишества".

Итак, первым делом нужно вызвать апплет Панели управления Windows под названием **Источники данных (ODBC)**. (В других операционных системах он может называться по-другому, например, **ODBC**; в любом случае, эта аббревиатура должна присутствовать в его названии.) Просто вызовите окно Панели управления и щелкните по значку с соответствующей надписью.

После этого на экране появится окно **Администратор источников данных ODBC**. Сразу же переключитесь на вкладку **Системный DSN** (рис. 17.2).

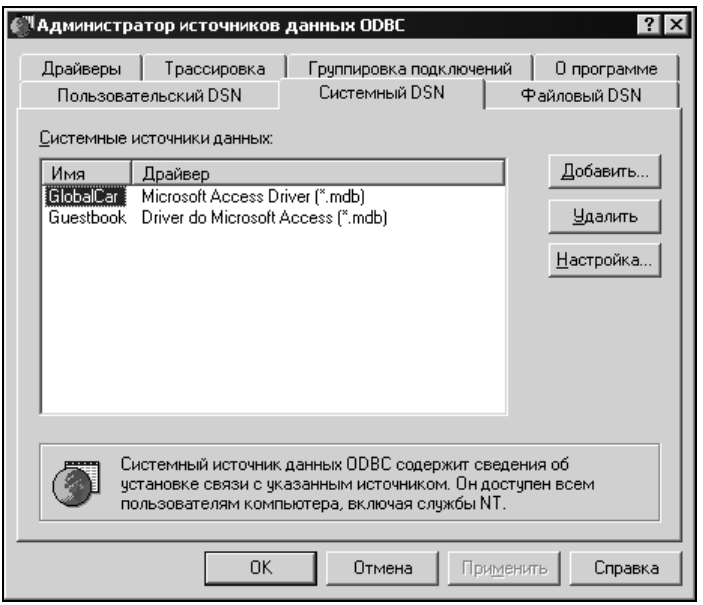


Рис. 17.2. Диалоговое окно **Администратор источников данных ODBC** (вкладка **Системный DSN**)

Чтобы добавить новый источник данных, нажмите кнопку **Добавить**. После этого на экране появится диалоговое окно **Создание нового источника данных**, показанное на рис. 17.3.

В списке, занимающем все это окно, выберите драйвер, соответствующий формату вашей базы данных. (*Драйвер* — это небольшая программа, непосредственно работающая с базой данных. Для каждого формата баз данных существует свой драйвер ODBC, в то время как остальные компоненты ODBC едины для всех форматов.) В нашем случае — это пункт **Microsoft Access Driver (*.mdb)**. Выбрав его, нажмите кнопку **Готово**.

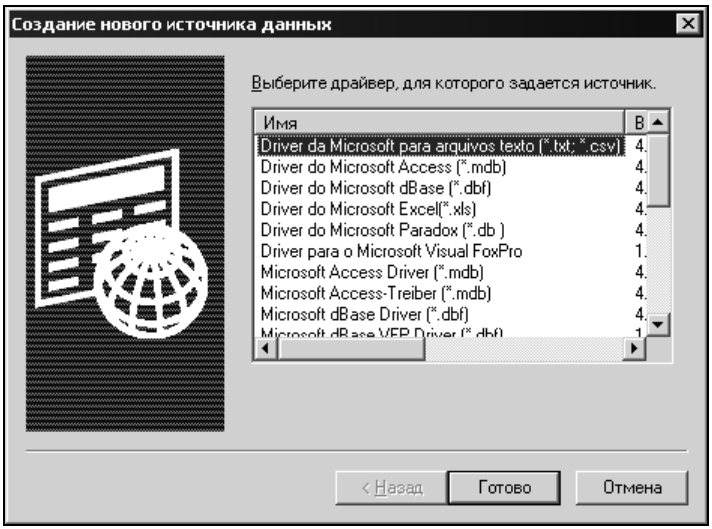


Рис. 17.3. Диалоговое окно **Создание нового источника данных**

Диалоговые окна сыплются на нас, как осенние листья. Следующее — это окно **Установка драйвера ODBC для Microsoft Access**, показанное на рис. 17.4. В нем задаются собственно параметры вашей базы данных.

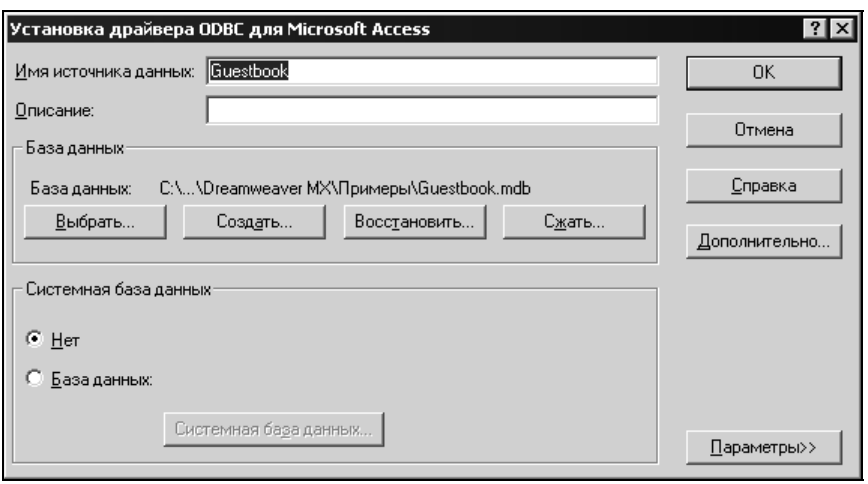


Рис. 17.4. Диалоговое окно **Установка драйвера ODBC для Microsoft Access**

В поле ввода **Имя источника данных** задается уникальное имя источника данных. Оно необходимо для того, чтобы программы, использующие эту базу данных, получили доступ к ее источнику. Введите в него что-нибудь наподобие `Guestbook`.

В поле ввода **Описание** вы можете ввести развернутое описание создаваемого источника данных. Вообще-то, это необязательно.

Осталось только задать путь к файлу базы данных. Нажмите кнопку **Выбрать**, выберите нужный файл в появившемся на экране диалоговом окне открытия файла Windows и нажмите кнопку открытия.

Задав все данные, нажмите кнопку **OK**. После этого созданный вами источник данных появится в списке **Системные источники данных**, занимающем почти всю вкладку **Системные DSN** (см. рис. 17.2). Теперь можно закрыть окно **Администратор источников данных ODBC**, нажав кнопку **OK**.

Если вы случайно ввели не те данные, выберите в списке **Системные источники данных** нужный пункт и нажмите кнопку **Настройка**. Также вы можете просто дважды щелкнуть по нужному пункту этого списка. На экране появится диалоговое окно **Установка драйвера ODBC для Microsoft Access**; исправьте данные и нажмите кнопку **OK**. Чтобы удалить ненужный источник данных, выберите необходимый пункт в списке и нажмите кнопку **Удалить**. Далее на экране появится небольшое окно-предупреждение; нажмите кнопку **Да**.

На этом создание источника данных ODBC можно считать законченным.

Регистрация базы данных в Dreamweaver

Наш следующий шаг — *регистрация базы данных* в среде Dreamweaver. Это нужно для того, чтобы Dreamweaver выяснил структуру базы данных, с которой мы будем работать, и смог создать необходимые серверные сценарии.

Но сначала давайте создадим нашу первую серверную страницу. Это будет страница AddRecord.asp, служащая для добавления записи в гостевую книгу. Чтобы создать ее, выберите пункт **New** в меню **File**, выберите в списке **Category** диалогового окна **New Document** пункт **Dynamic Page**, а в правом списке — пункт **ASP JavaScript**. После этого нажмите кнопку **Create** — и активная серверная страница будет создана.

Для того чтобы зарегистрировать базу данных в Dreamweaver, вам понадобится панель **Databases**. Если ее нет на экране, включите пункт-выключатель **Databases** меню **Window** или нажмите комбинацию клавиш <Ctrl>+<Shift>+<F10>. Сама эта панель показана на рис. 17.5.

Большую часть этой панели (как и многих других панелей Dreamweaver) занимает список уже зарегистрированных баз данных. Вы можете выбрать любой пункт этого списка и произвести над соответствующей ему базой данных различные манипуляции...

Нет, постойте! Ведь мы еще не зарегистрировали ни одной базы данных! Панель **Databases** показывает нам текст, описывающий шаги, необходимые для регистрации базы данных в Dreamweaver. Но, поскольку мы эти шаги уже выполнили, перейдем сразу к процессу регистрации.

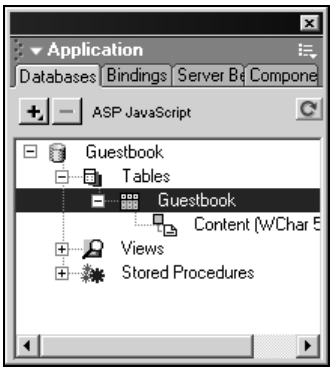


Рис. 17.5. Панель **Databases**

Нажмите кнопку со знаком "плюс", расположенную над списком, и выберите в появившемся на экране меню пункт **Data Source Name (DSN)**. На экране появится диалоговое окно **Data Source Name (DSN)** (рис. 17.6).

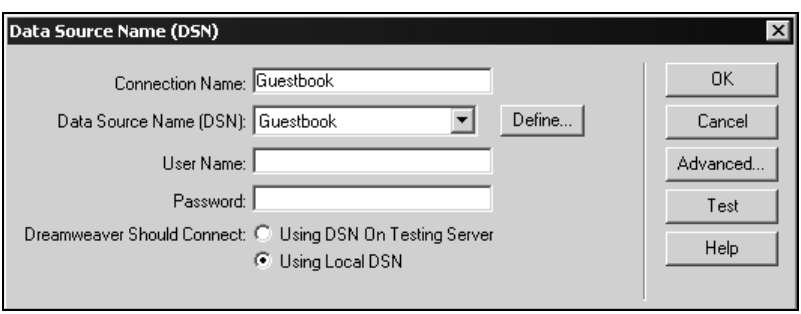


Рис. 17.6. Диалоговое окно **Data Source Name (DSN)**

В поле ввода **Connection Name** вводится уникальное имя зарегистрированной в Dreamweaver базы данных. Чтобы не ломать голову, введите здесь то же имя, что и у источника данных ODBC.

Сам источник данных выбирается в раскрывающемся списке **Data Source Name (DSN)**. Нажав кнопку **Define**, вы можете вызвать окно **Администратор источников данных ODBC** и создать его, если не сделали этого ранее.

В поля ввода **User Name** и **Password** вводятся соответственно имя пользователя и пароль для подключения к серверу баз данных. В нашем случае они не нужны — оставьте эти поля пустыми.

Чтобы проверить правильность ввода данных, нажмите кнопку **Test**. После этого Dreamweaver попытается подключиться к вашей базе данных и выведет соответствующее окно-предупреждение. Если попытка подключения оказалась неудачной, исправьте введенные данные и повторите ее. Возмож-

но, вам придется вызвать окно **Администратор источников данных ODBC** и исправить параметры самого источника данных ODBC.

Введя все данные и проверив их на правильность, нажмите кнопку **OK**. Если же вы передумали регистрировать базу данных, нажмите кнопку **Cancel**.

После этого зарегистрированная вами база данных появится в списке панели **Databases** в виде ветви иерархического списка (см. рис. 17.5). Вы можете развернуть ее и просмотреть ее содержимое. Все созданные вами таблицы находятся в подветви **Tables**. Также вы можете просмотреть, какие поля содержит та или иная таблица, ведь таблица тоже представляет собой ветвь списка.

Чтобы изменить параметры регистрации базы данных, выделите нужную ветвь списка и выберите пункт **Edit Connection** контекстного меню. На экране появится диалоговое окно **Data Source Name (DSN)**, в котором вы сможете изменить эти параметры.

Чтобы удалить ненужную базу данных (в смысле, регистрацию), выделите необходимую ветвь списка и нажмите кнопку со знаком "минус" или выберите пункт **Delete Connection** контекстного меню. На экране появится окно-предупреждение; нажмите кнопку **Да** (**Yes**).

Если вы хотите зарегистрировать еще одну базу данных, сведения о которой не сильно отличаются от уже зарегистрированной, выделите уже зарегистрированную базу данных и выберите пункт **Duplicate Connection** контекстного меню. На экране появится диалоговое окно **Data Source Name (DSN)**, в котором вы сможете задать параметры новой базы данных.

Пункт **Test Connection** позволит вам протестировать подключение к базе данных, аналогично кнопке **Test** диалогового окна **Data Source Name (DSN)**.

Чтобы просмотреть данные, содержащиеся в какой-либо таблице зарегистрированной базы данных, выделите ветвь списка, соответствующую этой таблице, и выберите пункт **View Data** контекстного меню. На экране появится диалоговое окно **View Data** (рис. 17.7), в котором вы и увидите все содержимое этой таблицы. Сами данные отображаются в списке, занимающем почти все это окно. Нажимая кнопки **Previous 25** и **Next 25**, вы можете просмотреть соответственно предыдущие или последующие 25 записей таблицы. Чтобы закрыть это окно, нажмите кнопку **OK**.

Примечание

Dreamweaver создает в папке Connections корневой папки сайта активные серверные страницы, соответствующие каждой зарегистрированной вами базе данных. Эти страницы содержат только серверные сценарии и хранятся в файлах, чьи имена совпадают с именами, данными вам базам данных при регистрации. Не удаляйте их, иначе все зарегистрированные вами базы данных пропадут.

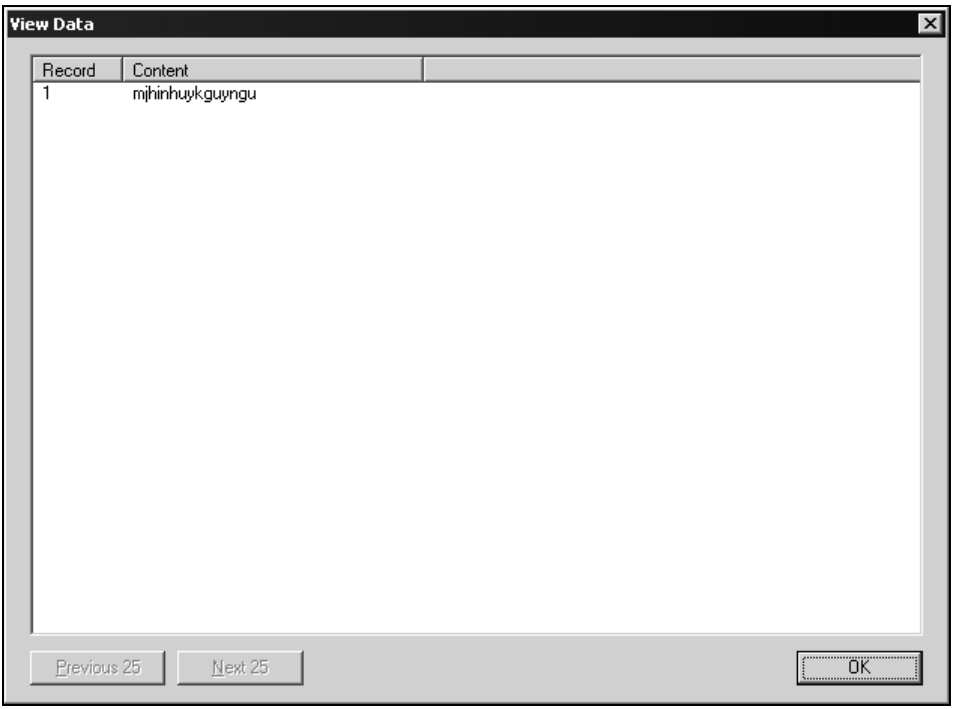


Рис. 17.7. Диалоговое окно **View Data**

Вот, собственно, и все. Мы подключились к базе данных и теперь можем приступить к созданию самой серверной страницы.

Создание серверных страниц в Dreamweaver

Сохраните готовую Web-страницу в файле AddRecord.asp и поместите ее в каталог HTMLs нашего сайта. Вы можете задать для нее те же свойства, что и для других страниц этого же сайта, и поместить в нее какой-либо поясняющий текст. Нет смысла описывать, как это делается — уж что-что, а делать обычные Web-страницы вы уже научились.

Создание простейших серверных страниц

Итак, мы хотим создать страницу, с помощью которой посетитель сайта может добавить новую запись в гостевую книгу. Для этого нам понадобится форма, состоящая из одного поля ввода, и серверный сценарий, точнее, серверное поведение, добавляющее введенный посетителем текст в поле

Content таблицы Guestbook. Максимальная длина этого текста составит 50 символов (или какую там длину текстового поля Content вы задали).

Давайте создадим форму, содержащую одно поле ввода и одну кнопку отправки данных. Назовем поле ввода Content, а саму форму — AddRecord. Так мы не запутаемся в многочисленных названиях. Остальные параметры формы мы задавать не будем — это сделает за нас Dreamweaver.

Итак, создайте эту форму. Она будет очень проста — см. рис. 17.8.

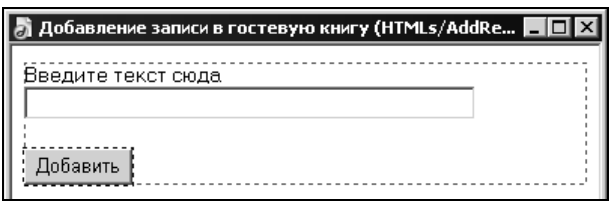


Рис. 17.8. Форма добавления новой записи в гостевую книгу

Следующий шаг — создание серверного поведения, выполняющего добавление записи в таблицу.

Для работы с поведениями нам понадобится панель **Server Behaviors** (рис. 17.9). Чтобы вызвать ее на экран, включите пункт-выключатель **Server Behaviors** в меню **Window** или нажмите комбинацию клавиш <Ctrl>+<F9>.

Рис. 17.9. Панель **Server Behaviors**

Как обычно, большую часть этой панели занимает список уже созданных поведений. Также эта панель имеет кнопки со знаками "плюс" и "минус", соответственно, добавляющую и удаляющую поведения.

Чтобы создать новое серверное поведение, нужно щелкнуть кнопку со знаком "плюс" и выбрать необходимый пункт в появившемся на экране меню серверных поведений. Поведение, добавляющее запись в таблицу, создается

выбором пункта **Insert Record**. После выбора этого пункта на экране появляется диалоговое окно **Insert Record**, показанное на рис. 17.10.

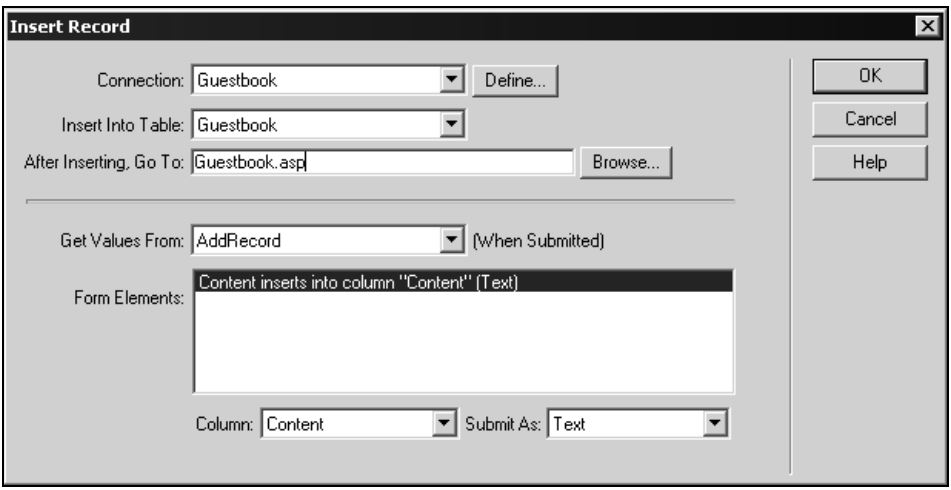


Рис. 17.10. Диалоговое окно **Insert Record**

В раскрывающемся списке **Connection** выбирается нужная база данных, а в раскрывающемся списке **Insert Into Table** — таблица, куда будет добавлена запись.

В поле ввода **After Inserting, Go To** вводится интернет-адрес Web-страницы, на которую будет выполнен переход после успешного добавления записи в таблицу. Вы также можете нажать кнопку **Browse** и выбрать нужный файл в диалоговом окне **Select File**.

В раскрывающемся списке **Get Values From** выбирается форма, из которой берутся значения для новой записи.

В списке **From Elements** перечислены все элементы управления, находящиеся в выбранной форме. Вы можете выбрать любой из них и задать для него поле таблицы, куда будут помещены содержащиеся в нем данные, и тип отправляемых данных (текст, число, дата и т. п.).

Поле таблицы выбирается в раскрывающемся списке **Column**. Пункт **<ignore>** этого списка позволяет проигнорировать значение этого элемента управления (т. е. оно не будет помещено в поле таблицы).

Тип отправляемых данных указывается в раскрывающемся списке **Submit As**. В нем доступны следующие пункты:

- **Text** — текстовые данные;
- **Numeric** — числовые данные;
- **Date** — значение даты;

- **Date MS Access** — значение даты, предназначенное для Microsoft Access;
- **Checkbox Y,N** — логическое значение, преобразуемое в символы "Y" ("да") или "N" ("нет");
- **Checkbox 1,0** — логическое значение, преобразуемое в числа 1 ("да") или 0 ("нет");
- **Checkbox -1,0** — логическое значение, преобразуемое в числа −1 ("да") или 0 ("нет");
- **Checkbox MS Access** — логическое значение, предназначенное для Microsoft Access.

Задав все нужные данные, не забудьте нажать кнопку **OK**. После этого в списке панели **Server Behaviors** появится новое поведение. А готовая форма добавления записи в гостевую книгу будет подобно рис. 17.11. Как видите, Dreamweaver поместил в него скрытое поле, специально для своих нужд.

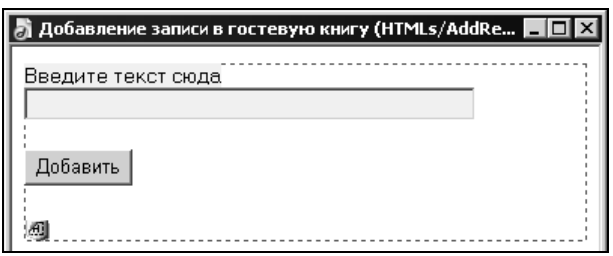


Рис. 17.11. Готовая форма добавления записи в гостевую книгу

Теперь вы можете проверить созданную Web-страницу в действии. Сохраните ее, запустите Web-сервер, если он еще не запущен, и откройте страницу в Web-обозревателе. То, что вы увидите, показано на рис. 17.12.

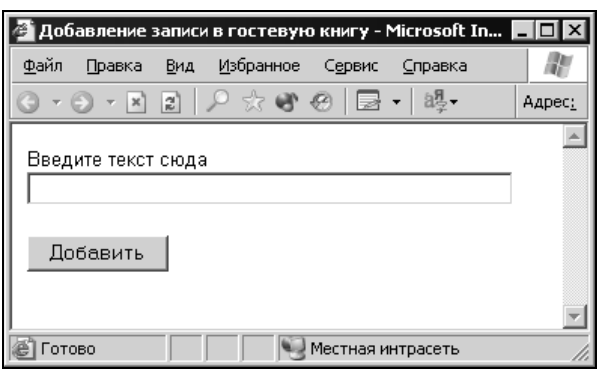


Рис. 17.12. Страница AddRecord.asp в окне Web-обозревателя

Попробуйте ввести в поле ввода какой-нибудь текст и нажать кнопку **Добавить**. После этого откройте базу данных Guestbook.mdb в Access и посмотрите, что получилось. В таблице `Guestbook` должна появиться новая запись, введенная вами.

Если хотите, можете просмотреть исходный код созданной нами серверной страницы. Видите, сколько сценариев JavaScript добавил в нее умница Dreamweaver, чтобы мы смогли добавить запись в таблицу! Пользуясь им и документацией по языку JavaScript и технологии ASP, вы сможете изучить принципы написания серверных сценариев вручную.

Но все это потом. Теперь нам пора сделать следующий шаг к вершинам серверного программирования.

Более сложные серверные страницы

Какой толк в гостевой книге, если посетители сайта не могут просмотреть все записи, добавленные в нее ранее! (Если, конечно, это не "закрытая" книга, только для "своих".) Поэтому давайте сделаем страницу для просмотра этих записей. Создадим новую серверную страницу ASP и сохраним ее под именем Guestbook.asp. И приступим...

Создание набора данных

Но прежде чем приступим собственно к дизайну, нам нужно выполнить одно подготовительное действие — создать набор данных. *Набор данных* — это условие выборки данных с указанием таблиц и полей, из которых будут извлекаться данные. При этом также возможно задание *фильтра* — условия отбора данных из записей. После создания набора данных уже можно приступать к серверной странице, отображающей данные.

Для работы с наборами данных нам понадобится еще одна панель, не рассмотренная ранее, — **Bindings** (рис. 17.13). Чтобы вызвать эту панель на экран, включите пункт-выключатель **Bindings** меню **Window** или нажмите комбинацию клавиш <Ctrl>+<F10>.

Эта панель также состоит, в основном, из списка уже созданных к данному времени наборов данных. Изначально она, однако, содержит только нравоучительный текст, перечисляющий нерадивым Web-программистам шаги, которые они должны пройти перед тем, как смогут создать свой первый набор данных. Но мы-то их уже прошли! Поэтому сразу же перейдем к созданию нашего первого набора данных.

Нажмите неизменную кнопку со знаком "плюс", находящуюся в этой панели, и выберите в появившемся на экране меню пункт **Recordset (Query)**. Также вы можете нажать кнопку **Recordset** (рис. 17.14) вкладки **Application** панели объектов или выбрать пункт **Recordset** подменю **Application Objects** меню **Insert**. На экране появится диалоговое окно **Recordset** (рис. 17.15).

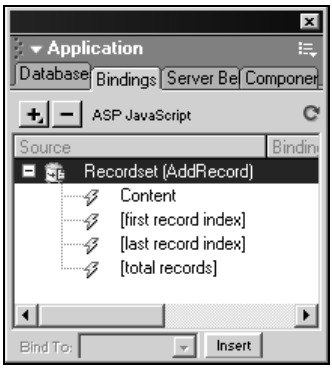


Рис. 17.13. Панель **Bindings**

Рис. 17.14. Кнопка **Recordset** панели объектов

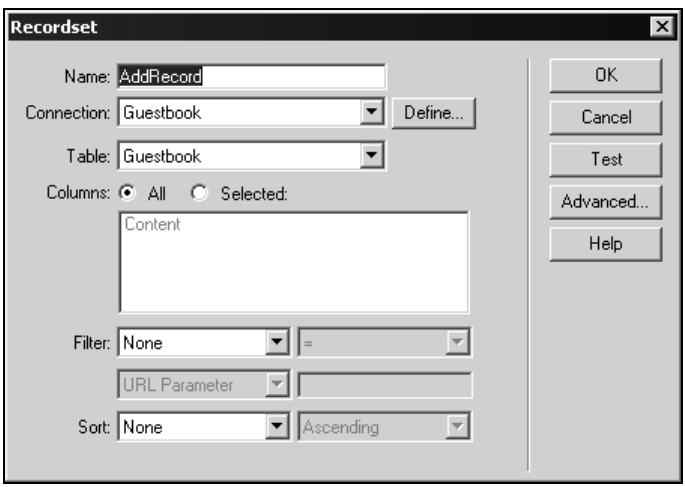


Рис. 17.15. Диалоговое окно **Recordset**

В поле ввода **Name** вводится уникальное имя создаваемого набора данных. Введите в него, например, `AddRecord`. Так будет сразу понятно, зачем нужен этот набор данных.

В раскрывающемся списке **Connection** выберите зарегистрированную базу данных. Вы также можете нажать кнопку **Define**; на экране появится диалоговое окно **Data Source Name (DSN)**, с помощью которого вы сможете зарегистрировать базу данных, если не сделали это ранее.

В раскрывающемся списке **Table** выберите нужную таблицу зарегистрированной базы данных. Впрочем, она у нас всего одна — `Guestbook`.

Группа переключателей **Columns** позволит вам извлечь данные из всех полей выбранной таблицы или только из некоторых. Если вы включите переключатель **All** (вообще-то он включен по умолчанию), данные будут извлечены из всех полей таблицы. Чтобы извлечь данные только из некоторых полей, включите переключатель **Selected** и выберите нужные поля в списке, расположенном ниже. Чтобы выбрать сразу несколько пунктов, щелкайте по ним, удерживая нажатой клавишу <Ctrl>.

Чтобы проверить, правильно ли введены параметры, нажмите кнопку **Test**. После этого на экране должно появиться диалоговое окно **Test SQL Statement**, похожее на уже знакомое вам окно **View Data**. Если параметры набора данных введены неправильно, Dreamweaver выдаст вам соответствующее окно-предупреждение.

Элементы управления, расположенные в нижней части окна **Recordset** и предназначенные для задания фильтров, мы рассмотрим позднее. Сейчас же просто нажмите кнопку **OK**.

После того как вы создадите новый набор данных, он появится в панели **Bindings** в виде ветви иерархического списка. Эта ветвь будет содержать все поля созданного набора данных, а также некоторые дополнительные поля, которые не берутся из таблицы, а вычисляются самим процессором баз данных. Такие поля называются *системными*. Давайте их перечислим.

- **[first record index]** — номер первой записи набора, показываемой на текущей странице.
- **[last record index]** — номер последней записи набора, показываемой на текущей странице.
- **[total records]** — количество записей в наборе.

Вы можете использовать системные поля на своих страницах наряду с обычными полями таблицы.

Если вы хотите исправить параметры какого-либо набора данных, дважды щелкните по нужной строке списка. Учтите, что щелкать надо по начальному пункту ветви. После этого на экране появится диалоговое окно **Recordset**, в котором вы сможете изменить необходимые параметры.

Чтобы удалить ненужный набор данных, выделите соответствующий пункт и нажмите кнопку со знаком "минус".

А теперь поговорим о свойствах набора данных. Чтобы увидеть их, вам сначала нужно выделить соответствующий пункт списка панели **Bindings**. Но обычно этого мало — редактор свойств так же будет отображать параметры страницы, открытой в окне документа. Поэтому дважды щелкните по нужному набору данных в списке, чтобы вызвать диалоговое окно **Recordset**,

и тут же закройте его, щелкнув кнопку **Cancel**. Только после этого редактор свойств наконец-то покажет параметры набора данных (рис. 17.16).

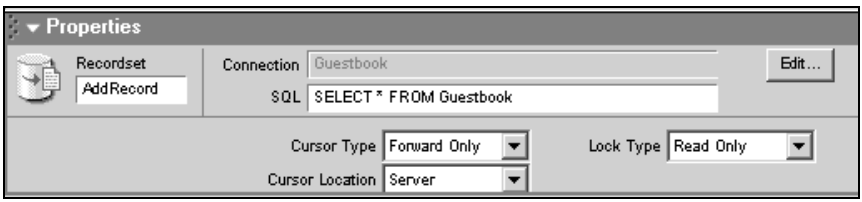


Рис. 17.16. Вид редактора свойств при выделенном наборе данных

В поле ввода **Recordset** вводится имя набора данных.

Текстовое поле **Connection** отображает имя зарегистрированной базы данных, из которой будут извлекаться записи данного набора. Вы можете нажать кнопку **Edit**; на экране появится диалоговое окно **Recordset**, в котором вы сможете изменить параметры набора данных.

В поле ввода **SQL** отображается текст запроса на языке SQL, сформированный Dreamweaver. В данный момент там находится следующий текст:

```
SELECT * FROM Guestbook
```

Разберем его пословно:

- `SELECT` — ключевое слово, задающее набор полей;
- `*` (звездочка) обозначает все поля таблицы;
- `FROM` — ключевое слово, задающее таблицу;
- `Guestbook` — таблица.

Как видите, запросы на языке SQL напоминают фразы английского языка. И, в общем-то, они довольно понятны, "прозрачны", как говорят.

Раскрывающийся список **Cursor Type** служит для задания *типа* набора данных. В нем доступны четыре пункта:

- **Static** — *статический набор данных*, содержащий только те записи таблицы, которые присутствовали в ней на момент выборки. Доступен только для чтения;
- **Forward Only** — *однонаправленный статический набор данных*, по которому можно перемещаться только по направлению от первой записи к последней, но не обратно. Обеспечивает более высокое быстродействие, чем простой статический набор данных. Этот пункт выбран по умолчанию;
- **Dynamic** — *динамический набор данных*, содержащий все записи таблицы и отражающий все изменения, добавления и удаления, сделанные други-

ми пользователями базы данных. Доступен как для чтения, так и для записи;

- **Keyset** — *ограниченный динамический набор данных*, не отражающий добавления и удаления, сделанные другими пользователями базы данных.

Раскрывающийся список **Cursor Location** задает местонахождение набора данных, в смысле, где он обрабатывается: на стороне клиента (пункт **Client**) или на стороне сервера (пункт **Server**). По умолчанию выбран пункт **Server**, и это правильно — ведь мы создаем серверное приложение.

Раскрывающийся список **Lock Type** задает *тип блокировки*. *Блокировка* используется для того, чтобы ограничить доступ к записи, изменяемой каким-либо пользователем базы данных, другим пользователям. Если блокировку не использовать, может произойти конфликт изменения данных, когда одновременно несколько пользователей будут пытаться сохранить изменения, сделанные в одной и той же записи. Здесь доступны четыре пункта:

- **Read Only** — записи набора доступны только для чтения;
- **Pessimistic** — *пессимистическая блокировка*, когда запись блокируется, как только пользователь начинает изменять значения ее полей. Как только пользователь сохранит изменения, блокировка снимается;
- **Optimistic** — *оптимистическая блокировка*, когда запись блокируется только на момент записи в нее новых значений;
- **Batch Optimistic** — разновидность оптимистической блокировки, используемой при операциях изменения сразу нескольких записей.

Задайте для нашего набора данных динамический тип (пункт **Dynamic** раскрывающегося списка **Cursor Type**) и оптимистическую блокировку (пункт **Optimistic** раскрывающегося списка **Lock Type**), т. к. мы будем создавать страницу, добавляющую в таблицу новую запись. Остальные элементы управления пока не трогайте.

Страница, отображающая данные

Серверную страницу, отображающую данные, создать проще простого. После того как мы создали набор данных, нам остается только перетаскивать в страницу нужные поля из панели **Bindings** — а Dreamweaver сделает все остальное.

Итак, откройте страницу Guestbook.asp, если вы ее еще не открыли. И поместите в нее *динамический текст*, отображающий содержимое поля таблицы `Guestbook`. Можете считать, что динамический текст — своего рода изменяемая область шаблона, содержимое которой будет взято из заданного вами поля таблицы.

Как это сделать? Как уже было сказано, просто перетащив нужное поле (в нашем случае — **Content**) из панели **Bindings** на Web-страницу. Если вы

уже закрыли эту панель, то можете щелкнуть кнопку **Dynamic Text** (рис. 17.17) вкладки **Application** панели объектов или выбрать пункт **Dynamic Text** подменю **Application Objects** меню **Insert**. После этого на экране появится диалоговое окно **Dynamic Text**, показанное на рис. 17.18. Выберите нужный пункт в иерархическом списке **Field** и нажмите кнопку **OK**.

Рис. 17.17. Кнопка **Dynamic Text** панели объектов

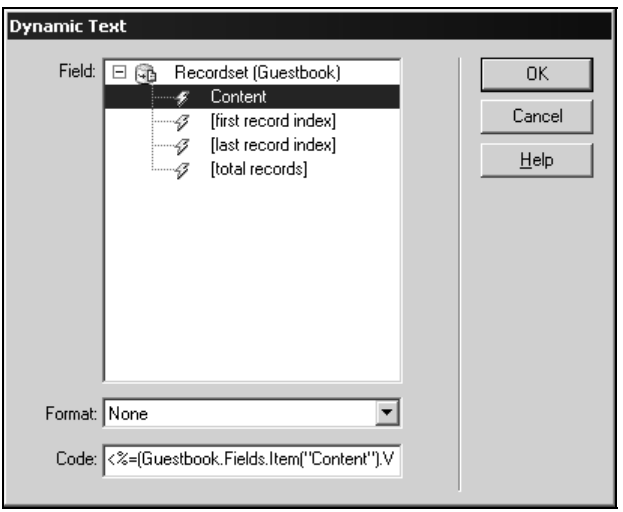


Рис. 17.18. Диалоговое окно **Dynamic Text**

Созданный нами динамический текст выглядит не очень вдохновляюще — см. рис. 17.19. Просто какой-то текст, заключенный в фигурные скобки и выделенный голубым цветом. Чтобы познать его скрытую силу, нам нужно наполнить его реальными данными. А для этого необходимо загрузить страницу в Web-обозревателе...

Рис. 17.19. Динамический текст, отображающий содержимое поля `Content` таблицы `Guestbook`

Но не хочется нам открывать этот Web-обозреватель! Не хочется, и все тут. Ведь Dreamweaver имеет встроенные средства просмотра серверных Web-страниц в том виде, в каком они будут отображены в Web-обозревателе,

вместе со всеми данными. Это так называемый *режим показа "живых" данных*. И сейчас мы выясним, как в него переключиться.

А сделать это проще простого. Нажмите кнопку-выключатель **Live Data View** (рис. 17.20) в инструментарии документа. Если вы убрали этот инструментарий с экрана, включите пункт-выключатель **Live Data** в меню **View** или нажмите комбинацию клавиш <Ctrl>+<Shift>+<R>. После этого Dreamweaver запросит данные из базы и выведет открытую в окне документа страницу в своем "настоящем" виде (рис. 17.21).

Рис. 17.20. Кнопка **Live Data View** инструментария документа

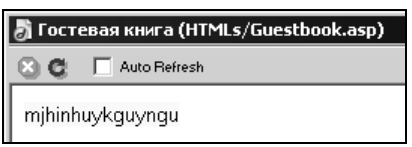


Рис. 17.21. Страница Guestbook.asp, отображаемая при включенном режиме показа "живых" данных

Тарабарщина, которую вы видите на этом рисунке, — содержимое поля `Content` первой записи таблицы `Guestbook`. Это отладочные данные, введенные автором, чтобы проверить, как работает серверная страница.

Да, но где остальные записи?

Дело в том, что динамический текст показывает только содержимое одной — текущей — записи набора данных. Чтобы увидеть все остальные записи, вам нужно будет переместиться на них. Как это сделать? Сейчас мы узнаем.

Создание навигатора

Для перемещения по записям набора данных нужен особый элемент Web-страницы, называемый *навигатором*. Он состоит из набора гиперссылок или кнопок, выполняющих перемещение на первую, предыдущую, последующую и последнюю записи набора. Если вы много работаете в Microsoft Access, вы должны помнить нечто подобное, расположенное в самом низу окна таблицы, а именно — набор из четырех вышеупомянутых кнопок и поля ввода номера записи.

Dreamweaver предлагает нам встроенные средства создания навигатора, состоящего из гиперссылок. Давайте воспользуемся ими. Но сначала поставим текстовый курсор в конец созданного нами ранее динамического текста и нажмем клавишу <Enter>.

Чтобы создать гиперссылку и серверное поведение, выполняющие переход на первую запись набора, выберите пункт **Move To First Record** в подменю **Recordset Paging** меню серверных поведений панели **Server Behaviors**. На экране появится диалоговое окно **Move To First Record** (рис. 17.22).

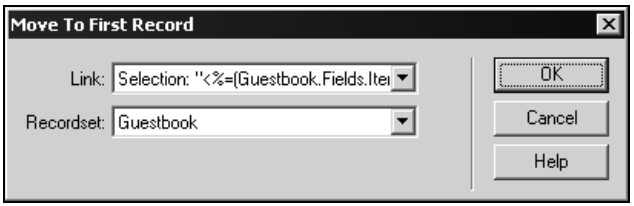


Рис. 17.22. Диалоговое окно **Move To First Record**

В раскрывающемся списке **Recordset** выберите нужный набор данных и нажмите кнопку **OK**. Dreamweaver создаст в месте, где находится текстовый курсор, гиперссылку с текстом "First". Замените ее текст на что-нибудь русскоязычное (например, "Первая") или более понятное ("<<"). Все, на этом создание гиперссылки, выполняющей переход на первую запись набора, закончено.

Поставьте после гиперссылки "<<" два-три неразрывных пробела, нажав нужное количество раз комбинацию клавиш <Ctrl>+<Shift>+<пробел>. Так мы отделим одну гиперссылку навигатора от другой.

Чтобы создать поведение, выполняющее переход на предыдущую запись набора, выберите пункт **Move To Previous Record** в подменю **Recordset Paging** меню серверных поведений. На экране появится диалоговое окно **Move To Previous Record**, аналогичное окну **Move To First Record**. Точно так же выберите в раскрывающемся списке **Recordset** нужный набор данных, нажмите кнопку **OK** и замените текст созданной гиперссылки на "<". Поставьте после второй гиперссылки два-три неразрывных пробела. И продолжим.

Поведения, выполняющие переход на последующую и последнюю записи набора, создаются путем выбора пунктов **Move To Next Record** и **Move To Last Record** соответственно. Одноименные диалоговые окна полностью схожи с окном **Move To First Record**. Так что никаких неожиданностей не должно возникнуть.

Результат наших трудов показан на рис. 17.23. Проверьте, все ли у вас получилось.

Поведения для навигатора можно создавать и по-другому. Напишите текст, который станет одной из гиперссылок навигатора, выделите его и выберите в меню серверных поведений нужный пункт. После этого введите необходимые параметры в появившееся на экране диалоговое окно и нажмите

кнопку **OK**. Так даже проще — вам не придется исправлять в конце текст гиперссылки.

Рис. 17.23. Готовая страница Guestbook.asp с навигатором

Сохраните готовую серверную страницу. К сожалению, проверить ее в режиме просмотра "живых" данных у нас не получится: хоть Dreamweaver и заменит динамический текст реальными данными, гиперссылки навигатора работать не будут. Поэтому придется все-таки открыть ее в окне Web-обозревателя. Попробуйте пощелкать по ссылкам и посмотрите, как меняется содержимое динамического текста.

Создание строки статуса набора данных

Вспомните окно таблицы Microsoft Access. Возле навигатора там отображается небольшое текстовое поле с номером текущей записи и общим количеством записей в таблице. Давайте сделаем такое же поле и на нашей серверной страничке, т. е. создадим *строку статуса набора данных*.

А для этого нам понадобятся системные поля этого самого набора данных. Напомним, что это такое. *Системным полем набора данных* называется поле, не взятое из таблицы, на основе которой создан этот набор, а созданное самим процессором баз данных и отображающее какую-либо служебную информацию. Таких системных полей Dreamweaver предлагает три: `[first record index]` (номер первой записи набора, показываемой на текущей странице), `[last record index]` (номер последней записи, показываемой на текущей странице) и `[total records]` (общее количество записей в наборе). И все они отображаются в панели **Bindings** в виде одноименных пунктов (см. рис. 17.13).

Итак, поставьте текстовый курсор после самой последней гиперссылки навигатора и нажмите клавишу <Enter>. В новом текстовом абзаце наберите текст "Запись" и поставьте после него неразрывный пробел. Именно здесь мы и создадим новый динамический текст, связанный с системным полем `[first record index]`. Как это сделать, вы знаете. Впрочем, если вы хотите, можете привязать этот динамический текст к полю `[last record index]` — все равно на нашей странице отображается только одна запись.

Теперь поставьте обычный пробел, введите текст "`из`", поставьте неразрывный пробел и создайте второй динамический текст, привязанный к системному полю `[total records]`. Вот и все.

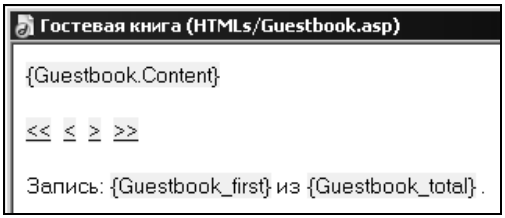


Рис. 17.24. Готовая страница Guestbook.asp с навигатором и строкой статуса набора данных

Готовая страница Guestbook.asp с навигатором и строкой статуса показана на рис. 17.24. Сохраните ее и откройте в Web-обозревателе, чтобы проверить работу строки статуса.

Страница для одновременного просмотра нескольких записей

Ох и хорошая у нас получилась серверная страничка! (Не смотрите, что выглядит она весьма коряво, ведь сейчас наша задача — научиться писать серверные страницы. Впрочем, Web-дизайн мы уже "прошли", так что можете по ходу дела заняться украшательством.) Одно плохо — показывает она только одну запись набора. Конечно, мы создали навигатор и строку статуса набора данных, но толку от них не очень много. Вот как бы нам "запихать" на одну страничку сразу несколько записей!..

Что, вы хотите несколько записей на одной странице? Запросто!

Создайте новую серверную страницу и сохраните ее в файле Guestbook_table.asp. Именно ее мы и превратим в чудо современного программистского искусства, показывающего несколько записей одновременно.

Первым делом создадим набор данных с такими же, как и у страницы Guestbook.asp, параметрами. Назовем его так же — `Guestbook`. После этого поместим на страницу динамический текст, привязанный к полю `Content` набора данных. И напоследок поставим текстовый курсор в конце этого динамического текста и нажмем клавишу <Enter>. Все это вам уже знакомо.

А теперь остановимся. И поговорим.

Что мы видим в окне документа? Ничего странного — просто серверная Web-страница, похожая на созданную нами ранее Guestbook.asp. Набор данных, динамический текст — и все. Если теперь открыть ее в Web-обозревателе, она отобразит нам одну-единственную запись. Мы уже это наблюдали.

Да, но нам нужны несколько записей на одной странице!

Для этого нам поможет уже знакомая вам возможность, предлагаемая Dreamweaver специально для таких случаев, — повторяющаяся область.

Да-да, нашу серверную страничку можно уподобить шаблону! В самом деле, в Dreamweaver мы создали своего рода образец Web-страницы, который в дальнейшем будет наполняться полезным содержимым. Только делать это будет не Web-дизайнер, а серверный сценарий. И данные он будет брать не из своей головы (или головы Web-писателя), а из базы данных.

Чтобы создать повторяющуюся область, сначала выделите фрагмент страницы, который станет ее содержимым. В нашем случае — это динамический текст, привязанный к полю `Content` набора данных. Затем выберите пункт **Repeat Region** меню серверных поведений панели **Server Behaviors**. После этого на экране появится диалоговое окно **Repeat Region** (рис. 17.25).

Рис. 17.25. Диалоговое окно **Repeat Region**

В раскрывающемся списке **Recordset** выберите набор данных, из которого будут извлекаться записи.

Если вы хотите, чтобы на странице отображались все записи набора данных, включите переключатель **All Records** группы **Show**. Это стóит делать тогда, когда вы уверены, что записей в вашем наборе не очень много, или если содержимое их невелико по размеру. В противном случае лучше всего показывать содержимое набора данных по частям; для этого включите верхний переключатель группы **Show** и введите количество одновременно отображаемых записей в поле ввода **Records at a Time**.

Давайте сделаем нашу страничку как можно меньше. Для этого зададим количество одновременно отображаемых записей, равное трем. Это позволит нам не вводить слишком много отладочных записей в таблицу `Guestbook` базы данных Guestbook.mdb.

Введя все данные, нажмите кнопку **OK**. Если вы передумали создавать повторяющуюся область, нажмите кнопку **Cancel**. Готовая повторяющаяся область показана на рис. 17.26.

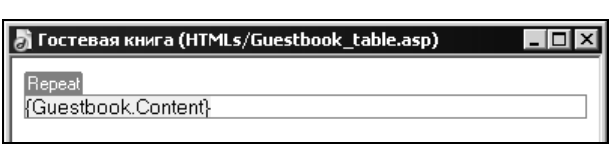


Рис. 17.26. Повторяющаяся область серверной страницы

Если теперь включить режим показа "живых" данных Dreamweaver, мы увидим следующее — см. рис. 17.27. Как видите, она "выдала" нам содержимое всех трех записей, что есть в таблице `Guestbook`.

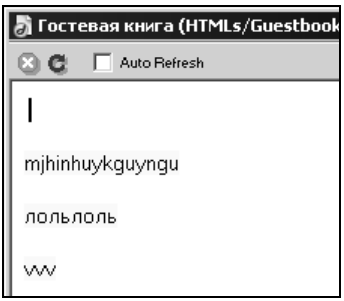


Рис. 17.27. Повторяющаяся область серверной страницы при включенном режиме показа "живых" данных

Теперь осталось добавить к нашей странице навигатор. Конечно, это будет не такой навигатор, который мы создали ранее. В данном случае нам нужно перемещаться не от записи к записи, а от страницы к странице. Но вам не нужно об этом беспокоиться — все хлопоты по отсчету необходимого количества записей возьмет на себя Dreamweaver. Поэтому просто создайте навигатор, как мы делали это раньше.

Если хотите, вы можете создать также строку статуса для набора данных. Здесь вам понадобятся оба системных поля, отображающих количество показываемых на странице записей: `[first record index]` и `[last record index]`. Как вы помните, первое из них показывает номер первой записи, присутствующей на странице, а вторая — номер последней из показываемых записей.

Осталось сказать о параметрах повторяющейся области. Чтобы задать их, выделите нужную повторяющуюся область и посмотрите на редактор свойств (рис. 17.28).

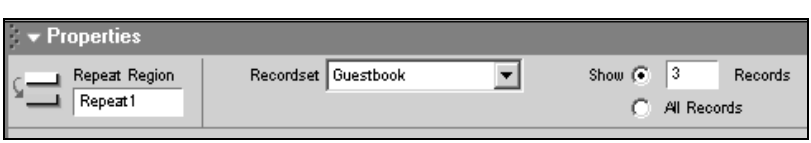


Рис. 17.28. Редактор свойств при выделенной повторяющейся области

Единственный элемент управления, который стоит здесь рассмотреть, — это поле ввода **Repeat Region**. В нем задается имя повторяющейся области. Остальные элементы управления знакомы вам по диалоговому окну **Repeat Region**.

Привязка элементов управления к данным

А сейчас мы рассмотрим еще одну очень интересную возможность — *привязку элементов управления к данным*. Это значит, что вы можете, скажем, заполнять списки значениями, взятыми из записей набора данных, или заимствовать оттуда же начальные значения для полей ввода. Первое, во всяком случае, очень часто бывает нужно.

Давайте добавим в нашу гостевую книгу второе поле — тип записи. Тип записи будет показывать, что хотел сказать посетитель сайта: похвалить его разработчика, поругать или сделать нейтральное замечание. Соответственно, это поле может принимать три значения: положительное, нейтральное или отрицательное замечание.

Откроем базу данных Guestbook.mdb в Access и добавим в таблицу `Guestbook` еще одно поле `Sign`. Зададим для него числовой тип. После этого создадим еще одну таблицу под названием `Signs`, содержащую два поля: счетчик `ID` и текстовое `Desc`. В поле `ID` будет помещаться уникальный номер каждой записи, а в поле `Desc` — описание соответствующей оценки. Сделаем поле `ID` ключевым. После этого привяжем поле `Sign` таблицы `Guestbook` к полю `ID` таблицы `Signs`.

Далее откроем таблицу `Signs` и создадим в ней три записи: "Положительно", "Нейтрально" и "Отрицательно". Access автоматически проставит в полях `ID` уникальные номера этих записей. Теперь осталось открыть таблицу `Guestbook` и проставить в поле `Sign` для всех записей какое-либо значение, неважно, какое. На этом подготовительные действия можно считать законченными. Закройте Access и вернитесь в Dreamweaver.

Откройте страницу AddRecord.asp. Поместите в форму новый элемент управления — раскрывающийся список `Sign`. Вставьте его между полем ввода `Content` и кнопкой отправки данных. И сохраните страницу.

Откуда будут браться значения для заполнения нашего списка? Правильно, из набора данных. Страница AddRecord.asp не содержит никаких наборов данных, поэтому нам нужно его создать.

Выберите в меню кнопки со знаком "плюс" панели **Bindings** пункт **Recordset (Query)**. В диалоговом окне **Recordset**, которое появится на экране после этого, задайте имя создаваемого набора данных `Signs`, выберите базу данных `Guestbook` и таблицу `Signs`. После нажатия кнопки **OK** набор данных будет создан.

Теперь выделите только что созданный нами раскрывающийся список `Signs` и нажмите кнопку **Dynamic**, которая появится в редакторе свойств (рис. 17.29). На экране появится диалоговое окно **Dynamic List/Menu**, показанное на рис. 17.30.

Рис. 17.29. Кнопка **Dynamic** редактора свойств

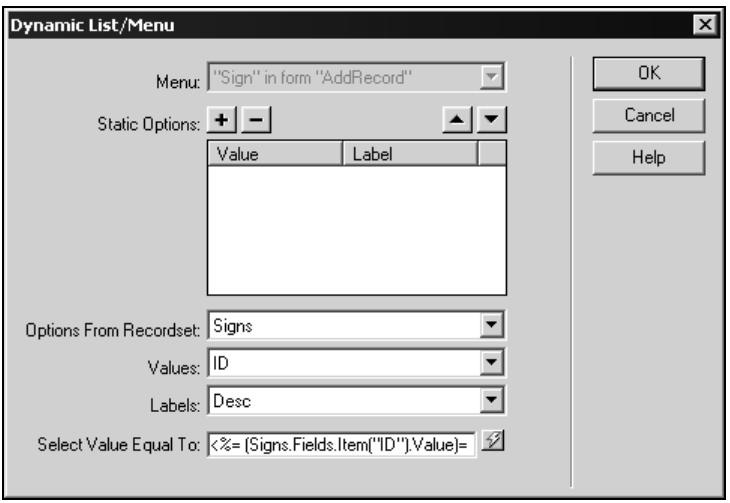


Рис. 17.30. Диалоговое окно **Dynamic List/Menu**

В раскрывающемся списке **Menu** выбирается нужный список. Но, поскольку мы его уже выделили на форме, этот список недоступен.

В раскрывающемся списке **Options From Recordset** выбирается набор данных, из которого будут браться значения для заполнения выделенного списка. Пункт **None** позволит вам "отвязать" список от наборов данных, т. е. создать обычный список с фиксированным набором пунктов. Поле набора данных, из которого будут браться значения для создания пунктов, выбирается в раскрывающемся списке **Labels**, а поле, из которого будут браться значения этих пунктов, — в списке **Values**.

Выберите в списке **Options From Recordset** набор данных `Signs`, в списке **Labels** — поле `Desc`, а в списке **Values** — поле `ID`. И продолжим разговор об окне **Dynamic List/Menu**.

Если вы хотите, чтобы при открытии страницы какой-то пункт списка отображался изначально выбранным, воспользуйтесь полем ввода **Select Value Equal To**. В него вводится значение пункта, который должен быть изначально выбранным. В нашем случае это число 1 — значение поля `ID` первой записи таблицы `Signs`.

Если вы хотите поместить в список свои собственные пункты, не взятые из набора данных *(статические)*, воспользуйтесь списком **Static Options**. Этот список состоит из двух колонок: **Value** (значение пункта) и **Label** (название

пункта). Чтобы ввести в этот список новый пункт, нажмите кнопку со знаком "плюс", и пункт будет добавлен. Поочередно щелкните по значениям, находящимся в обеих колонках, введите нужный текст и не забудьте нажать клавишу <Enter>. Точно так же вы можете изменить необходимое значение в колонке любого уже созданного пункта.

Если вы хотите удалить ненужный пункт из этого списка, выберите его и нажмите кнопку со знаком "минус". А кнопки ↑ и ↓ позволят вам переместить выбранный пункт соответственно выше или ниже на одну позицию.

Введя все данные, нажмите кнопку **OK**. Вот и все.

Точнее, не все. Нам еще нужно изменить параметры поведения `Insert Record` так, чтобы значение выбранного в списке `Sign` пункта помещалось в поле `Sign` таблицы `Guestbook`. Для этого откройте панель **Server Behaviors** и дважды щелкните по пункту **Insert Record (from "AddRecord")**, после чего на экране появится уже знакомое вам диалоговое окно **Insert Record**. Выберите в списке **Form Elements** пункт **Sign <ignore>**, после чего выберите в раскрывающемся списке **Column** пункт **Sign**, а в раскрывающемся списке **Submit As** — пункт **Numeric** (впрочем, сам Dreamweaver автоматически выберет этот пункт). И, конечно, нажмите кнопку **OK**.

Вот теперь действительно все. Можете проверить страницу в действии.

Создание сложных наборов данных

Осталось изменить страницу Guestbook_table.asp так, чтобы она отображала не только содержимое поля `Content`, но и содержимое поля `Sign`. Давайте сделаем это и попутно изучим построение сложных SQL-запросов и создание сложных наборов данных.

На первый взгляд, задача, стоящая перед нами, очень проста. В самом деле, мы уже знаем, как создать динамический текст и привязать его к полю набора данных. Мы уже, собственно, выполнили это, когда делали страницы Guestbook.asp и Guestbook_table.asp. Теперь же нам ничего не стоит добавить в повторяющуюся область последней страницы новый динамический текст, привязанный к полю `Sign`.

Давайте так и сделаем. И что получится в результате? А вот что.

Когда мы откроем доработанную страницу Guestbook_table.asp в Web-обозревателе, то увидим, что вместо слов "Положительно", "Отрицательно" или "Нейтрально" стоят цифры от 1 до 3. Почему? Да потому, что созданное нами поле `Sign` имеет числовой тип, и именно число, находящееся в нем, отображается на странице. Но мы-то хотим увидеть слова, а не цифры! Что делать?

Создать *сложный набор данных*, извлекающий данные не из одной, а из нескольких таблиц. А для этого нам придется построить сложный SQL-запрос, и сделать это нужно вручную.

Откройте страницу Guestbook_table.asp и переключитесь на панель **Bindings**. Дважды щелкните по пункту **Recordset (Guestbook)**, представляющему созданный нами ранее простой набор данных. (*Простым набором данных* называется тот, который извлекает данные из одной таблицы.) На экране появится диалоговое окно **Recordset**. Щелкните по кнопке **Advanced** — и это окно изменит свой вид (рис. 17.31).

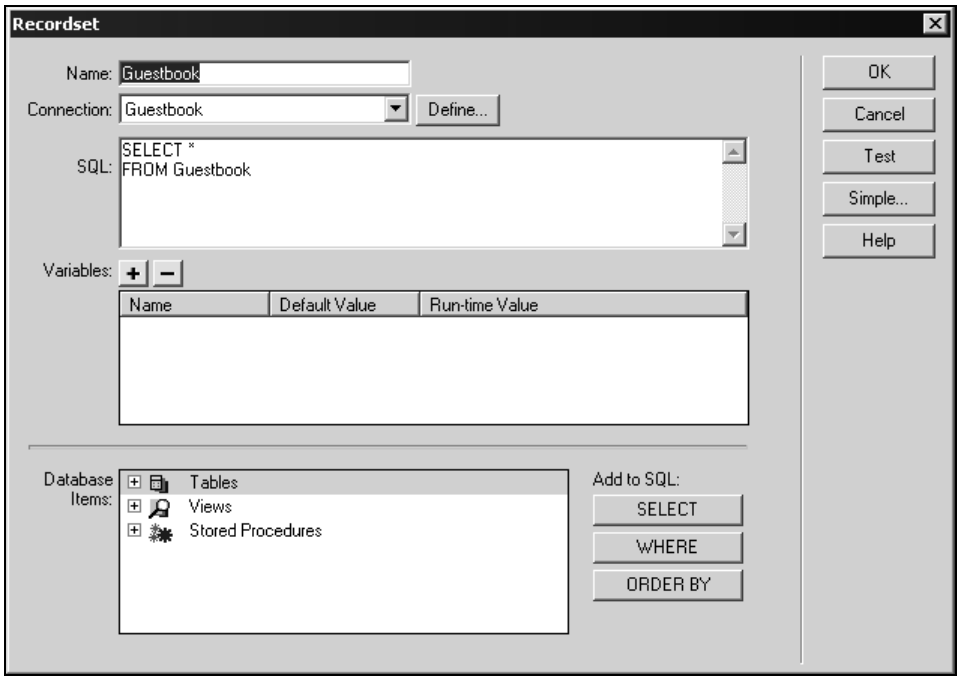


Рис. 17.31. Диалоговое окно **Recordset** (расширенное)

В этом диалоговом окне нам понадобятся поле ввода **SQL**, список **Database Items** и кнопки **SELECT**, **WHERE** и **ORDER BY**. Многоколоночный список **Variables** и все относящиеся к нему кнопки мы пока трогать не будем.

В поле ввода **SQL**, как вы поняли, вводится текст SQL-запроса. В настоящее время он таков:

```
SELECT * FROM Guestbook
```

Выше мы уже рассмотрели этот запрос по частям, поэтому не будем здесь повторяться. Напомним только, что этот запрос извлекает все поля всех записей из таблицы `Guestbook`.

Поскольку мы хотим создать новый запрос SQL, то давайте сначала удалим все содержимое поля ввода **SQL**. И начнем, что называется, с чистого листа.

Прежде всего поочередно откроем ветви **Tables** и **Guestbook** и выберем пункт **Content** в списке **Database Items**. Далее нажмем кнопку **SELECT**, чтобы поместить в поле ввода **SQL** ключевое слово `SELECT` с выбранным нами полем `Content`. В поле ввода **SQL** появится такой код:

```
SELECT Content FROM Guestbook
```

Затем поместим текстовый курсор после слов "SELECT Content", откроем ветви **Tables** и **Signs** и выберем пункт **Desc**. Опять нажмем кнопку **SELECT**. Код примет следующий вид:

```
SELECT Content, Desc FROM Guestbook, Signs
```

Как видите, мы извлекаем данные уже из двух таблиц: поле `Content` из таблицы `Guestbook` и поле `Desc` из таблицы `Signs`.

Осталось только добавить условие, связывающее запись таблицы `Guestbook` с записью таблицы `Signs`. Для этого добавим условие выборки записей. Выберем пункт **Sign** в подветви **Guestbook** ветви **Tables** и нажмем кнопку **WHERE**. Код в поле ввода **SQL** опять изменится:

```
SELECT Content, Desc FROM Guestbook, Signs WHERE Sign
```

Поставим текстовый курсор после текста "WHERE Sign", введем знак "=", выберем пункт **ID** ветви **Signs** в ветви **Tables** и опять нажмем кнопку **WHERE**. Код в поле ввода **SQL** примет свой окончательный вид:

```
SELECT Content, Desc FROM Guestbook, Signs WHERE Sign=ID
```

Здесь ключевое слово `WHERE` обозначает *условие отбора записей*. А код `Sign=ID` показывает, что в набор попадут только те записи таблицы `Guestbook`, значения поля `Sign` которых равно значению поля `ID` соответствующей записи таблицы `Signs`.

Чтобы проверить созданный SQL-запрос, нажмите кнопку **Test**. Если он правилен, на экране появится диалоговое окно **Text SQL Statement**, содержащее записи созданного набора данных.

Нажмите кнопки **OK** обоих диалоговых окон. И на всякий случай сохраните страницу. Сложный набор данных, извлекающий записи из двух таблиц, нами создан. Этот набор содержит два поля: `Content` (содержимое записи гостевой книги), взятое из таблицы `Guestbook`, и `Desc` (оценка сайта посетителем), взятое из таблицы `Signs`.

Что теперь? Может, создать новый динамический текст и привязать его к полю `Desc`? Можно, конечно. Но мы поступим по-другому. Как? Сейчас увидите.

Работа с динамическими атрибутами

Когда мы говорили о шаблонах *(см. главу 9)*, то выяснили, что атрибуты тегов можно сделать изменяемыми. В Web-страницах, созданных на основе

шаблона, мы можем задавать значения для этих атрибутов и, таким образом, менять их внешний вид в более широких пределах. Но атрибуты тегов можно сделать и *динамическими*, т. е. брать их значения из полей набора данных. Здесь мы рассмотрим, как работать с такими атрибутами.

Давайте сделаем так, чтобы содержимое разных записей набора данных `Guestbook` отображалось различным цветом в зависимости от значения, содержащегося в поле `Sign`. Пусть, например, содержимое поля `Content` отображается темно-синим цветом, если в поле `Sign` содержится единица ("Положительно"), темно-красным — если содержится тройка ("Отрицательно"), черным — в остальных случаях (двойка, "Нейтрально"). Соответствующие коды цветов в этом случае:

- темно-синий — `#000099`;
- темно-красный — `#990000`;
- черный — `#000000`.

Откроем базу данных Guestbook.mdb в Access и создадим новое поле в таблице `Signs`. Дадим этому полю имя `Color`, текстовый тип и установим его длину в 7 символов — этого хватит, чтобы хранить коды цветов. После чего откроем таблицу `Signs` и введем коды цветов в соответствующие записи таблицы. После этого закроем Access.

Далее переключимся в Dreamweaver, вызовем диалоговое окно **Recordset** для набора данных `Guestbook` и добавим в него вновь созданное поле. Попробуйте сделать это самостоятельно. Если же вы хотите сразу перейти к работе с динамическими атрибутами, просто введите в поле ввода **SQL** такой код:

```
SELECT Content, Desc, Color FROM Guestbook, Signs WHERE Sign=ID
```

Теперь выделим содержимое повторяющейся области страницы, щелкнув по тексту "{Guestbook.Content}", и зададим для него черный цвет. Для этого воспользуемся хорошо знакомым селектором цвета в редакторе свойств. В результате Dreamweaver поместит содержимое повторяющейся области в тег `<FONT>` с атрибутом `COLOR` — как вы помните, именно с их помощью задается цвет текста.

Теперь выделите содержимое тега `<FONT>`, щелкнув по соответствующей кнопке секции тегов. И переключитесь в режим отображения кода. Хотя в интерактивном руководстве и описано, как привязать атрибут к полю набора данных в режиме отображения страницы, но почему-то в данном случае Dreamweaver ведет себя очень странно. В частности, раскрывающийся список **Bind To**, с помощью которого и выполняется привязка поля набора данных к атрибуту тега, остается недоступным. Поэтому мы поступим иначе.

Выделите значение атрибута `COLOR` тега `<FONT>`. А теперь просто перетащите на него пункт **Color** панели **Bindings**. Значение атрибута `COLOR` примет такой вид (выделено полужирным шрифтом):

```
<FONT COLOR="<%=(Guestbook.Fields.Item("Color").Value)%>">
```

Код, который Dreamweaver подставил в качестве значения атрибута `COLOR`, извлекает из поля `Color` набора данных значение цвета и присваивает его атрибуту `COLOR`. Как видите, все достаточно просто и, опять же, "прозрачно".

Точно таким же образом вы можете создавать другие динамические атрибуты. Главное — "попасть" мышью в нужный фрагмент исходного кода. Но, вероятно, вы с этим справитесь и сами.

Вот и все. Теперь можете открыть вашу страницу в Web-обозревателе и посмотреть на получившийся результат.

Получение данных от другой Web-страницы

Мы выяснили, как передать данные, введенные в форму, в таблицу, создав, таким образом, новую запись. Но не всегда данные, переданные с другой страницы, нужно сохранять в базе. Иногда их необходимо обработать на другой странице, нигде не сохраняя.

Давайте сделаем следующее. Создадим небольшую статичную Web-страничку с тремя гиперссылками, направляющими посетителя на страницу Guestbook_table.asp и заставляющими последнюю показать только записи, удовлетворяющие определенному условию. Пусть первая из них будет показывать записи, значения поля `Sign` которых равно 1, вторая — записи с полем `Sign`, равным 2, третья — 3. Назовем эту страницу Selector.htm.

Интернет-адрес первой гиперссылки страницы Selector.htm будет таков:

```
Guestbook_table.asp?sign=1
```

Здесь мы передаем методом GET странице Guestbook_table.asp параметр `sign`, равный 1. (Вспомните *главу 15*, где описывались оба метода передачи данных.) Соответственно, адреса других гиперссылок будут отличаться от этого только значением параметра `sign`.

А для того чтобы отобрать нужные записи из набора данных `Guestbook` страницы Guestbook_table.asp, мы используем фильтр, сравнивающий значение поля `Sign` со значением параметра `sign`, переданного нам от страницы Selector.htm. Как видите, все очень просто и довольно наглядно.

Нет смысла подробно описывать, как создается страница Selector.htm — это вы уже знаете. Вместо этого сосредоточимся на странице Guestbook_table.asp и фильтре, отбирающем записи из таблиц в набор. Откройте данную страницу, если вы ее уже закрыли. И переключитесь в панель **Bindings**.

Для того чтобы получить данные от другой Web-страницы, нам нужно создать *параметр гиперссылки*. Этот параметр впоследствии примет значение, переданное от другой страницы методом GET. А уж принятое им значение мы сможем использовать где угодно.

Чтобы создать параметр гиперссылки, выберите в меню кнопки "плюс" панели **Bindings** пункт **Request Variable**. После этого на экране появится диалоговое окно **Request Variable**, показанное на рис. 17.32.

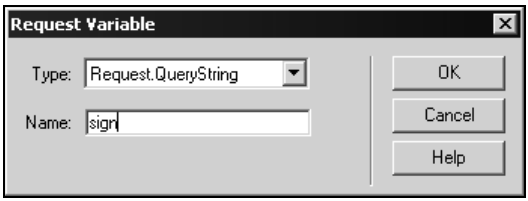


Рис. 17.32. Диалоговое окно **Request Variable**

В раскрывающемся списке **Type** выберите пункт **Request.QueryString**, а в поле ввода **Name** введите имя создаваемого параметра (в нашем случае — `sign`). Затем нажмите кнопку **OK**. После этого в списке панели **Bindings** появится новая ветвь **Request** с единственным пунктом **QueryString.sign**.

Внимание!

Вы не сможете изменить созданный вами параметр гиперссылки, так что вводите данные внимательно. Если вы все же ошиблись, удалите неверно заданный параметр и создайте его заново.

Если же вы передаете данные методом POST, вам будет нужно создать *параметр формы*. Для этого выберите в меню кнопки "плюс" все тот же пункт **Request Variable**, но в раскрывающемся списке **Type** диалогового окна **Request Variable** выберите пункт **Request.Form**.

Создание фильтров

А теперь можно приступить к созданию фильтра, отбирающего записи из таблицы в набор.

Помните, как мы создали сложный набор данных? Если не помните, рассмотрим код SQL-запроса, созданного нами:

```
SELECT Content, Desc, Color FROM Guestbook, Signs WHERE Sign=ID
```

Ключевое слово `WHERE` языка SQL задает условие, согласно которому из таблицы будут извлекаться записи. В приведенном выше примере мы уже фактически создали условие, отбирающее только те записи таблицы `Guestbook`, для которых в таблице `Signs` имеется "пара". Причем "пары" мы отбирали по значениям поля `Sign` таблицы `Guestbook` и поля `ID` таблицы `Signs`. Такие условия называются *условиями связи таблиц*.

Фильтром же называется *условие отбора записей*. Это условие используется не для связи таблиц, а для отбора записей из таблицы. Например, в приве-

денном ниже SQL-запросе используется именно условие отбора записей, иначе говоря, фильтр:

```
SELECT Content FROM Guestbook WHERE Sign=2
```

Этот запрос создаст набор данных, включающий только те записи таблицы `Guestbook`, поля `Sign` которых содержат значение 2 ("Нейтрально").

Чтобы создать фильтр, нам придется добавить условие отбора записей к уже существующему условию связи таблиц. Это совсем просто. Вызовите диалоговое окно **Recordset** для нашего набора данных `Guestbook`. Если оно отобразилось в обычном виде, щелкните кнопку **Advanced**, чтобы переключить его в расширенный вид. И посмотрите на содержимое поля ввода **SQL**.

Итак, нам нужно добавить в SQL-запрос новое условие. Мы будем сравнивать значение поля `Sign` и...

Но с чем мы его будем сравнивать?

Ах, да, с параметром `sign`, переданным страницей Selector.htm. Но как его сюда ввести?

Очень просто. Для этого нам понадобится создать *переменную SQL-запроса*, представляющую некий параметр, полученный от другой Web-страницы или в результате каких-то вычислений. И в этом нам поможет список **Variables** диалогового окна **Recordset** и относящиеся к нему элементы управления.

Нажмите кнопку со знаком "плюс". В списке появится новая строка. Щелкните по этой строке в районе колонки **Name** (имя создаваемой переменной) и введите это имя, в нашем случае — `sign`. После этого щелкните в районе колонки **Default Value** (значение переменной по умолчанию, присваиваемое ей изначально) и введите, скажем, `2` (т. е. по умолчанию будут выводиться только "нейтральные" записи). Теперь останется лишь ввести в колонку **Run-time Value** выражение, задающее истинное значение этой переменной. В нашем случае это будет следующий код:

```
Request.QueryString("sign")
```

Он извлекает из строки запроса, переданной методом GET, параметр `sign`. Если же вы передаете параметры методом POST, код будет несколько иным:

```
Request.Form("sign")
```

Вот и все. Мы создали переменную SQL-запроса и теперь можем ее использовать.

Откройте в списке **Database Items** ветви **Tables** и **Guestbook** и выберите пункт **Sign**. Поставьте текстовый курсор в поле ввода **SQL** после текста "WHERE Sign=ID" и нажмите кнопку **WHERE**. Содержимое поля ввода **SQL** станет таким:

```
SELECT Content, Desc, Color FROM Guestbook, Signs WHERE Sign= ID AND Sign
```

Поставьте текстовый курсор после текста "AND Sign" и введите вручную текст `=sign`. Таким образом, окончательный SQL-запрос будет выглядеть так:

```
SELECT Content, Desc, Color FROM Guestbook, Signs WHERE Sign=ID
  ↳AND Sign=sign
```

Все, наш новый запрос создан. Можете нажать кнопку **OK**. Перед этим, однако, лучше нажать кнопку **Test**, чтобы проверить правильность задания запроса. Так, на всякий случай.

Теперь сохраните страницу Guestbook_table.asp и опубликуйте ее на Web-сервере вместе со страницей Selector.htm. После этого откройте в Web-обозревателе страницу Selector.htm, набрав в строке адреса следующее:

```
http://localhost/HTMLs/Selector.htm
```

Пощелкайте по ссылкам и посмотрите, что из этого получится. Если же серверная страница упорно выдает вам список всех записей гостевой книги, то просто нажмите кнопку обновления вашего Web-обозревателя.

Необязательные области серверной страницы

Кроме повторяющихся областей, Dreamweaver поддерживает создание также областей необязательных. Вы, конечно, помните, что говорилось о необязательных областях в *главе 9*, но на всякий случай давайте повторим это. Необязательная область создается на серверной странице и может присутствовать или не присутствовать в зависимости от выполнения или невыполнения какого-либо условия. Все же вам стоит прочитать еще раз *главу 9*, где о необязательных областях говорилось более подробно.

Чтобы создать необязательную область, сначала выделите элементы страницы, которые должны стать ее содержимым. После этого выберите в подменю **Show Region** меню кнопки "плюс" панели **Server Behaviors** один из пунктов. Всего этих пунктов шесть:

- **Show Region If Recordset Is Empty** — создает необязательную область, отображаемую, если набор данных не содержит ни одной записи (пуст);
- **Show Region If Recordset Is Not Empty** — создает необязательную область, отображаемую, если набор данных содержит хотя бы одну запись (не пуст);
- **Show Region If First Record** — создает необязательную область, отображаемую, если текущей является первая запись набора;
- **Show Region If Not First Record** — создает необязательную область, отображаемую, если текущей является не первая запись набора;
- **Show Region If Last Record** — создает необязательную область, отображаемую, если текущей является последняя запись набора;

- **Show Region If Not Last Record** — создает необязательную область, отображаемую, если текущей является не последняя запись набора.

Давайте поместим повторяющуюся область, навигатор и строку статуса набора данных, т. е. все содержимое страницы, в необязательную область, отображаемую, если набор данных содержит записи. Для этого выделим все, что находится на странице, и выберем пункт **Show Region If Recordset Is Not Empty**. После этого на экране появится диалоговое окно **Show Region If Recordset Is Not Empty**, показанное на рис. 17.33.

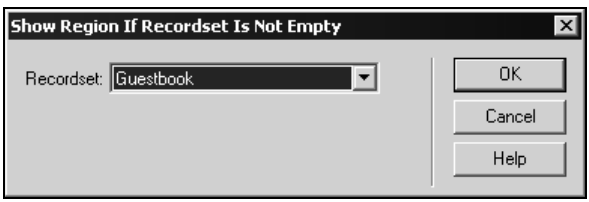


Рис. 17.33. Диалоговое окно **Show Region If Recordset Is Not Empty**

Единственное, что вам нужно сделать, — это выбрать в раскрывающемся списке **Recordset** нужный набор данных. И нажать кнопку **OK**. Созданная вами необязательная область показана на рис. 17.34.

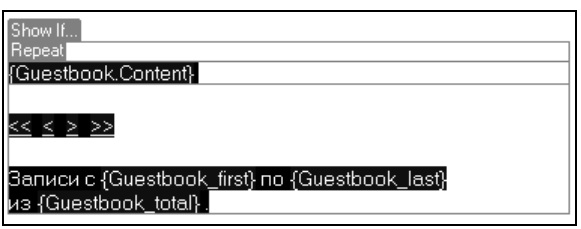


Рис. 17.34. Необязательная область

Теперь можете проверить измененную страницу Guestbook_table.asp. Переключитесь в режим показа "живых" данных или откройте ее в Web-обозревателе. Поскольку набор данных `Guestbook` содержит данные, содержимое страницы (оно же — содержимое необязательной области) будет отображено.

Теперь давайте создадим еще одну необязательную область, но отображаемую, если набор данных не имеет записей. Поместим в эту область текст "Гостевая книга не содержит записей". Это обычная практика в Web-дизайне: предупредить посетителя о том, что какой-то набор данных пуст.

Поместим текстовый курсор после созданной ранее необязательной области и нажмем клавишу <Enter>. В новом текстовом абзаце наберем текст "Гос-

тевая книга не содержит записей" и выделим его целиком, для чего щелкнем по соответствующей кнопке секции тегов.

Далее выберем в подменю **Show Region** меню кнопки "плюс" панели **Server Behaviors** пункт **Show Region If Recordset Is Empty**. На экране появится диалоговое окно **Show Region If Recordset Is Empty**, аналогичное уже знакомому вам окну **Show Region If Recordset Is Not Empty**. Выберем в раскрывающемся списке **Recordset** нужный набор данных и нажмем кнопку **OK**.

Также можно поместить гиперссылки, составляющие навигатор, в необязательные области, отображаемые и скрываемые в зависимости от того, является ли текущая запись первой или последней. Сделайте это сами — это будет хорошей практикой.

На этом рассказ о создании простейших серверных страниц можно считать почти законченным. Почти потому, что нам нужно рассмотреть еще одну возможность, предлагаемую Dreamweaver, — быстрое создание серверных страниц.

Быстрое создание серверных страниц

Очень часто бывает просто необходимо быстренько сляпать какую-либо серверную страницу: поместить в нее форму для ввода данных, навигатор, строку статуса и т. п. Специально для таких "пожарных" случаев Dreamweaver предлагает возможность быстрого создания серверных страниц. Воспользовавшись пунктами подменю **Application Objects** меню **Insert** или соответствующими им кнопками вкладки **Application** панели объектов и введя нужные данные в появившихся на экране диалоговых окнах, вы можете быстро создать различные элементы, присущие серверным страницам. Согласитесь — это неплохой выход для малоопытных программистов и торопыг.

Вы спросите, почему автор описал эту замечательную возможность в паре абзацев, да вдобавок в самом конце книги? Дело в том, что задача автора: научить вас серверному программированию в среде Dreamweaver. А для этого вам нужно самим "почувствовать" в руках если не сам программный код, то хотя бы поведения Dreamweaver и элементы страниц, к которым они привязаны. Вы должны сами знать, как что работает. Кроме того, настоящий творец все делает своими руками. (Можете считать это шуткой, но это правда.)

Хотя не в силах автора запретить вам пользоваться возможностями быстрого создания серверных страниц. Но только если действительно нет времени (или желания) творить.

Что дальше?

А дальше продолжим занятия серверным программированием.

В этой главе мы научились создавать простейшие серверные страницы. А в следующей — и последней — главе настоящей книги мы узнаем, как создаются простейшие интерактивные сайты. Мы узнаем, как объединить вместе серверные страницы, обрабатывающие данные из базы. А также изучим некоторые основные понятия, не затронутые в этой главе.

Ну, так что? Вперед, к интерактивности!

ГЛАВА 18

Создание интерактивных сайтов

Здесь мы подведем итог всего, что было сказано в трех предыдущих главах: научимся создавать целые интерактивные сайты, построенные на основе серверных Web-страниц. Мы объединим все полученные нами ранее знания в единое целое — Web-сайт, чьи страницы генерируются серверными программами на основе базы данных. И на этом закончим изучение Dreamweaver.

Да-да, это все! Конец книги близок. Автор рассказал вам о Dreamweaver все, что хотел рассказать. Остальное зависит только от вас, от вашего трудолюбия и желания учиться бóльшему.

Создание интерактивных Web-сайтов — высший пилотаж Web-дизайна. Чтобы сделать такой сайт, Web-дизайнер не только должен владеть языком HTML и обладать художественным чутьем, но и просто обязан знать "классическое" программирование и разбираться в базах данных. Конечно, современные программные продукты, наподобие Macromedia Dreamweaver, значительно облегчают труд Web-дизайнера, но отнюдь не берут всю его работу на себя. Уже говорилось, что Web-дизайн, как и "классическое" программирование — не наука, а искусство, иначе и программистов, и Web-дизайнеров давно бы уже заменили роботами. Или программами типа Dreamweaver.

Поэтому и программист, и Web-дизайнер, чтобы достичь вершин мастерства, должны работать. Программистами и Web-дизайнерами не рождаются — ими становятся. И данная книга — только первый шаг в овладении этими всегда перспективными профессиями.

В настоящее время профессии Web-дизайнера и Web-программиста слились в одну. Теперь любой Web-дизайнер просто обязан владеть навыками серверного программирования. А иначе и быть не может: интерактивные Web-сайты из разряда экзотики переходят в область ширпотреба. Сейчас можно сказать, что все сайты являются интерактивными. По крайней мере, боль-

шинство из них содержит хотя бы гостевую книгу, т. е. реализует начальный уровень взаимодействия с посетителями.

Об интернет-магазинах, расплодившихся, словно грибы после дождя, уже не стоит и говорить. Интернет-магазин — это большой интерактивный сайт, и создать его без серверного программирования невозможно. (Можно, конечно, но тогда у вас получится простейший сайт фирмы со списком товаров и сведениями о заказе.) Также без серверного программирования не удастся создать полноценный поисковый сайт, доску объявлений и даже полноценный архив программ.

А поэтому нам просто необходимо выяснить, как создаются такие вот интерактивные сайты. И начнем мы, как всегда, с теории.

Принципы создания интерактивных сайтов

Перечислим основные принципы разработки интерактивных Web-сайтов.

1. Все или почти все страницы сайта генерируются серверными программами. Статичные страницы хоть и присутствуют, но составляют весьма незначительный процент и содержат данные, не меняющиеся с течением времени, например сведения о владельце сайта или справочные данные о пользовании сайтом.
2. Данные для генерации страниц, как правило, берутся из баз данных, поскольку программирующее человечество пока что не придумало лучшего способа организованно хранить однотипные данные, чем помещение их в реляционные базы, которые вне конкуренции. Базы данных практически всегда работают под управлением сервера и очень часто — на отдельном серверном компьютере. Нередко также используется хранение данных в текстовых файлах, но это справедливо, в основном, для фрагментов Web-страниц (серверные включения).
3. Очень часто используется *разграничение доступа* к сайту. Разграничение доступа подразумевает, что разные посетители сайта имеют различные права на доступ к тем или иным данным, опубликованным на сайте. Так, обычные посетители могут только просматривать данные, в то время как "опытные" посетители имеют возможность также вносить в них некоторые изменения. Ну, а администраторы сайта, разумеется, могут творить с ними все, что захотят.
4. Если сайт использует разграничение доступа, то он обязательно должен предусматривать возможность *входа* (по-английски — logon). Чтобы войти на сайт, посетитель должен зайти на особую *страницу входа* и набрать свои имя и пароль. Если эти имя и пароль занесены в список посетителей сайта, серверная программа разрешает посетителю войти на сайт. В противном случае посетителю предлагается зарегистрироваться на сайте.

5. Сайт, использующий разграничение доступа, должен также предусматривать возможность *регистрации посетителя*. Если это общедоступный сайт, наподобие интернет-магазина, то посетитель регистрируется на нем, вводя данные о себе на особой *странице регистрации*, после чего они помещаются в список посетителей сайта. Если же это закрытый сайт, то для регистрации на нем надо будет обратиться к его администратору.
6. Сайт, использующий разграничение доступа, должен также предусматривать возможность *выхода с сайта*. При этом серверная программа "забывает", что данный посетитель вошел на сайт, очищая память, отведенную для хранения данных в нем, хотя соответствующая запись в списке посетителей, конечно же, сохраняется. Каждый посетитель, вошедший на сайт, обязательно должен с него выйти, чтобы освободить системные ресурсы серверного компьютера для других посетителей. Чтобы выйти с сайта, посетитель обычно должен щелкнуть специальную гиперссылку; реже используется особая *страница выхода с сайта*.
7. Разумеется, полностью общедоступные сайты типа поисковых машин не предусматривают разграничение доступа.

Осталось теперь ввести один термин. Давайте назовем процесс "общения" посетителя с Web-сайтом, начиная от входа на главную страницу и заканчивая "уходом" с него, *сессией*. В дальнейшем мы будем часто пользоваться этим термином, так что не забывайте его.

Вот теперь, кажется, все. Можно приступать к созданию нашего первого интерактивного Web-сайта. Пусть это будет библиотека различных афоризмов, умных, глупых или нейтральных. Причем доступна эта библиотека будет только после регистрации, т. е. она будет поддерживать разграничение доступа. Сами данные библиотеки будут храниться в базе данных Access, а доступ к ним будет производиться через особые серверные страницы. (Такой способ общения с базой данных называется *Web-интерфейсом*.)

Итак, приступим...

Административные страницы сайта

Первоначально создадим набор Web-страниц, служащих для администрирования нашего сайта. В самом деле, нашу базу данных нужно сначала наполнить содержимым — кому нужна пустая библиотека. А это работа администратора, которому необходимо предоставить соответствующие инструменты.

Как администрируются Web-сайты

Как вы знаете, администратором называется человек, осуществляющий поддержку Web-сайта или какой-либо серверной программы (Web-сервера,

сервера базы данных и пр.). Администратор настраивает программное обеспечение, поддерживает в порядке серверный компьютер, а в случае интерактивного Web-сайта также еще и следит за его базами данных. Очень часто администратор сайта является и его разработчиком, хотя это и не правило.

Администратор может быть один или работать в составе группы администраторов, отвечающих за различные составные части этого сайта. В последнем случае один администратор может следить за Web-сервером, другой — за базами данных, а третий — за сервером баз данных. Как правило, большие корпоративные сайты так и администрируются — один человек просто не может за всем уследить. Кроме того, большие Web-сайты разрабатываются отдельной командой Web-дизайнеров, Web-художников и Web-программистов, работающих совместно с администраторами. Сами понимаете, корпоративный сайт большой компании, такой как Microsoft, один человек просто не "потянет".

Другое дело — маленький сайт, такой как наш. Разрабатывает и поддерживает их один человек — автор, Web-дизайнер, Web-художник и Web-программист в одном лице.

Теперь поговорим о том, как администрируются интерактивные Web-сайты. Собственно, для этого могут применяться два способа, которые мы сейчас рассмотрим. Оба этих способа имеют право на существование, и выбор любого из них — дело вашего вкуса.

Первый способ заключается в том, что для работы с базой данных сайта — просмотра или изменения хранящихся в ней данных — используются сторонние программы. "Сторонние" в данном случае значит — не относящиеся к Web-серверу, т. е. Web-сервер никак не участвует в администрировании вашего сайта. Административные программы работают отдельно от него и не обращаются к нему для выполнения своих функций.

В *главе 17*, когда мы создавали базу данных для нашей доски объявлений, то применяли как раз такой подход. Поскольку наш сайт использовал базу данных формата Microsoft Access, то мы использовали эту самую СУБД для работы с этой базой. Web-сервер в этом никак не участвовал.

Если же вы храните данные в серверной базе, то можете пользоваться любой современной программой СУБД, умеющей "общаться" с вашим сервером данных, в том числе той же самой Access. (Впрочем, получать данные от сервера сейчас могут практически все СУБД.) Самый же простой случай — хранение данных в текстовых файлах; в этом случае вы можете воспользоваться любым текстовым редактором, например Блокнотом.

Преимущество первого способа администрирования сайта — снижение нагрузки на Web-сервер. Это преимущество особенно ощутимо, когда сайт администрируется достаточно часто. Недостаток: необходимость наличия дополнительных программ на компьютере администратора.

Второй способ администрирования сайта заключается в том, что вместо сторонних программ применяется специальный набор серверных Web-страниц. Эти страницы доступны только для администратора сайта и позволяют работать с данными, хранящимися в базе, без использования других программ. Такие Web-страницы часто называют *административными*.

У этого способа администрирования сайтов есть два достоинства и два недостатка. Достоинства: отсутствие нужды в сторонних программах и возможность разграничения доступа для администраторов сайта, если таких несколько. Недостатки: повышенная нагрузка на Web-сервер и необходимость создания двух разных наборов Web-страниц — административных и общего доступа.

Как уже писалось, вы можете использовать любой из этих способов. Просто взвесьте все достоинства и недостатки, предлагаемые обоими этими способами, и сделайте выбор. В принципе, ни один из них не предлагает каких-то неоспоримых достоинств, так что выбор — за вами.

Давайте выберем для нашего сайта второй способ администрирования, т. е. с использованием административных Web-страниц. Так мы сможем лучше изучить принципы построения современных интерактивных сайтов и сможем попрактиковаться в создании разных типов серверных страниц. А что до трудоемкости, то сейчас для нас важна любая практика.

Страница списка высказываний

Итак, приступим!

Давайте сначала подумаем над структурой нашей базы данных. Ведь если мы этого не сделаем, то ее (структуру) придется придумывать по ходу дела. А так никто не делает — вспомните, ведь строители перед тем, как начать строить что-либо, всегда сверяются с планом.

Пусть все высказывания, хранящиеся в нашей библиотеке, делятся на три категории: умные, глупые и нейтральные. (Возможно, потом появятся другие категории, но сейчас нам вполне хватит этих трех.) Таким образом, мы позволим посетителю нашего будущего сайта сразу же при входе выбрать то, что ему нужно, т. е. когда посетитель зайдет на наш сайт, он сразу же получит возможность выбрать на особой странице одну из категорий, после чего перейдет на другую страницу, где будут перечислены высказывания выбранной им категории.

Сейчас же мы не будем делить высказывания на категории, а выведем их в одном списке вперемешку. Не стоит слишком усложнять наш первый сайт, иначе мы толком ничему не научимся.

Вы спросите, а как же разграничение доступа? Где страница входа на сайт? Это успеется. Сначала сделаем основные страницы, отображающие и изме-

няющие информацию, хранящуюся в нашей базе, а потом займемся безопасностью. На данном этапе ей можно пренебречь.

Откроем Access и создадим базу данных под названием Library.mdb. В ней создадим таблицу `Categories` с полями `ID` (счетчик) и `Name` (текстовое, 30 знаков). Первое поле послужит нам ключевым, а во втором, как вы уже поняли, будут храниться имена категорий. После этого введем в эту таблицу три записи: "Умные", "Нейтральные" и "Глупые".

Далее создадим таблицу, где будут храниться наши высказывания. Эта таблица будет содержать следующие поля:

- `ID` — счетчик;
- `Content` — текст высказывания (Memo-поле);
- `Category` — категория высказывания (длинное целое, связанное с полем `ID` таблицы `Categories`).

Назовем эту таблицу `Sentences`. Введем какую-нибудь запись для отладочных целей, не обязательно осмысленную. И закроем Access — больше он нам не понадобится.

Вы уже знаете, как создается источник данных ODBC. Если не знаете, обратитесь к *главе 17*, где это подробно описано. Наш новый источник данных будет называться `Library` и предоставлять доступ к базе данных Library.mdb.

После этого создадим новую папку Sample4. В ней будут находиться файлы нашего нового сайта Sample site 4. В папке Sample4 создадим вложенную папку Admin, в которой будут храниться административные страницы сайта. Зарегистрируем наш новый сайт в Dreamweaver, задав те же данные, что и для сайта Sample site 1 в *главе 17*. Зарегистрируем базу данных Library.mdb в Dreamweaver, воспользовавшись только что созданным источником `Library`. На этом подготовительные действия закончились.

Создав таблицу, приступим к созданию серверной страницы. Назовем ее так же, как таблицу, — Sentences.asp. Введем какой-нибудь пояснительный текст, если вдруг у нас появится такое желание. И сохраним ее в папке Admin, вложенной в корневую папку нашего сайта.

После этого создадим набор записей `Sentences`, извлекающий данные из только что созданной таблицы. Причем этот набор будет сложным: он свяжет вместе таблицы `Sentences` и `Categories`, "заимствуя" из первой таблицы поля `ID` и `Content`, а из второй — поле `Name`. Связь будет осуществляться по полю `Category` таблицы `Sentences` и полю `ID` таблицы `Categories`. Создайте этот набор записей, а в случае затруднений справьтесь в *главе 17*. Не задавайте для этого набора ни фильтра, ни сортировки. Текст SQL-запроса должен быть таким:

```
SELECT Sentences.ID, Content, Name FROM Sentences, Categories
  ↳WHERE Sentences.Category=Categories.ID
```

Запись вида "Sentences.ID" означает, что мы выбираем значения поля `ID` таблицы `Sentences`.

Поскольку мы будем отображать одновременно два значения, будет лучше создать для этого таблицу. Эта таблица должна содержать две строки и четыре столбца. В первой строке будет находиться "шапка", а во второй — собственно содержимое. (Как вы поняли, вторую строку придется поместить в повторяющуюся область.) В первом столбце таблицы будет отображаться содержание высказывания, а во втором — категория (в смысле, название категории), а последние два столбца мы пока оставим пустыми. Создайте такую таблицу и отформатируйте, как хотите.

Теперь поместите в первую ячейку второй строки этой таблицы динамический текст, привязанный к полю `Content` набора записей, а во вторую ячейку — динамический текст, привязанный к полю `Name`. Далее выделите всю вторую строку, щелкнув по соответствующей кнопке секции тегов, и создайте "вокруг" нее повторяющуюся область, связанную с набором данных `Sentences`. Задайте количество одновременно отображаемых записей равным двадцати.

Поскольку мы создали повторяющуюся область, отображающую за раз фиксированное количество записей, нам нужно также создать навигатор и строку статуса. Сделайте это, сверяясь с *главой 17*, где создание навигатора и строки статуса было подробно описано, и поместите их под таблицей.

Теперь остается только создать "вокруг" таблицы, навигатора и строки статуса необязательную область, показываемую только тогда, когда в наборе `Sentences` есть записи. И поместить на странице еще одну необязательную область, отображаемую при отсутствии записей в наборе и содержащую текст, говорящий об этом. Создайте эти области.

Вот и все. Наша страница готова. Можете проверить ее, переключив Dreamweaver в режим показа "живых" данных или загрузив страницу в Web-обозревателе.

К несчастью, в таблице `Sentences` находится только одна запись. Но мы сейчас это исправим.

Страница добавления записи

Давайте подумаем, что нам нужно сделать теперь, когда страница списка высказываний готова и работает?

Таблица `Sentences`, как вы уже заметили, девственно-пуста. (Отладочная запись, введенная нами в Access, не считается.) Нужно заполнить ее данными. А для этого нам необходимо создать страницу добавления записи.

Создайте новую серверную страницу, введите в нее поясняющий текст и сохраните опять же в папке Admin под именем AddSentence.asp. Создайте в этой странице форму и назовите ее `Sentence`. Внутри формы создайте область редактирования и назовите ее `Content`, не забудьте также о кнопке отправки данных.

Что-то мы забыли... Ах, да! Нужно еще поместить в форму раскрывающийся список `Category` и привязать его к набору записей `Categories`. Но перед этим необходимо создать сам набор записей `Categories`, из которого будут взяты пункты списка `Category`. Задайте выборку всех записей таблицы `Categories`. После этого задайте *сортировку* записей по полю `Name`, для чего выберите это поле в раскрывающемся списке **Sort** диалогового окна **Recordset** (см. рис. 17.15). А раскрывающийся список, расположенный правее, позволит вам задать порядок сортировки записей: по возрастанию (пункт **Ascending**) или по убыванию (пункт **Descending**). Задав параметры набора данных, нажмите кнопку **OK**.

Теперь можно создать поведение `Insert Record`. (Как это делается, было описано в *главе 17*.) Привяжите область редактирования `Content` к полю `Content` таблицы `Sentences`, а список `Category` — к полю `Category`. В качестве страницы, на которую будет осуществлен переход после добавления записи, задайте Sentences.asp.

Сохраните готовую страницу AddSentence.asp. После этого откройте страницу Sentences.asp и поместите в нее гиперссылку, указывающую на страницу AddSentence.asp. Вот и все.

Теперь можете проверить созданную нами страницу добавления записи. Опубликуйте сайт на Web-сервере, откройте страницу Sentences.asp в Web-обозревателе и щелкните по гиперссылке "Добавить". Внесите в таблицу `Sentences` несколько записей. Они нам пригодятся для отладки.

Страница изменения записи

Наряду с добавлением новых записей часто приходится изменять уже существующие. Давайте создадим страницу, с помощью которой можно изменять уже существующие в наборе записи.

Создадим новую страницу, введем в нее поясняющий текст и сохраним под именем EditSentence.asp в папке Admin. Здесь мы впоследствии создадим форму с элементами управления, предназначенными для правки значений полей выбранной нами записи. А пока закроем эту страницу — нас ждут другие дела.

Сейчас мы добавим на страницу Sentences.asp гиперссылку, ведущую на страницу изменения записи. Откройте страницу, поместите текстовый курсор в третью ячейку второй строки таблицы (теперь вы поняли, зачем она

нужна?) и наберите в ней слово "Изменить". Далее выделите это слово и наберите в поле ввода интернет-адреса редактора свойств такой текст:

```
EditSentence.asp?ID=
```

Далее переключитесь в режим отображения кода и перетащите из панели **Bindings** в место, находящееся точно после знака "=", поле `ID` набора записей. Интернет-адрес гиперссылки примет такой вид:

```
EditSentence.asp?ID=<%=(Sentences.Fields.Item("ID").Value)%>
```

Давайте выясним, что это значит. Код

```
Sentences.Fields.Item("ID").Value
```

извлекает значение поля `ID` данной записи набора. Далее это значение помещается в гиперссылку, которая принимает вид:

```
EditSentence.asp?ID=<значение поля ID">
```

и передается странице EditSentences.asp в качестве значения параметра гиперссылки `ID`. Поскольку значение поля `ID` уникально для каждой записи набора (поле `ID` — счетчик записей, как вы помните), страница EditSentences.asp может однозначно идентифицировать запись, которую вы хотите изменить, и заполнить значениями ее полей элементы управления формы, т. е. подготовить запись к правке.

Теперь сохраните страницу Sentences.asp и откройте только что созданную нами страницу EditSentence.asp. Пора заняться ей.

Для того чтобы отредактировать запись, нам понадобится форма. Создайте ее по аналогии со страницей AddSentence.asp. Никаких особенностей она иметь не будет, поэтому вы можете просто скопировать уже готовую форму со страницы добавления новой записи и вставить ее в страницу EditSentence.asp. Но перед этим создайте на этой странице набор данных `Categories`, также по аналогии со страницей AddSentence.asp.

Следующий шаг — создание набора записей, из которого будет взята запись для заполнения элементов управления созданной нами формы. Этот набор должен содержать одну-единственную запись, поэтому нам придется задать фильтр, отбирающий нужную запись. Она будет выбрана по значению поля счетчика `ID` таблицы `Sentences`, которое должно равняться значению параметра гиперссылки `ID`, переданному от страницы Sentences.asp *(см. выше)*.

Чтобы создать фильтр, выберите название поля `ID` в раскрывающемся списке **Filter** диалогового окна **Recordset**, а в раскрывающемся списке, расположенном правее, — пункт = (равенство значений поля и фильтра). Далее в списке, расположенном ниже, выберите пункт **URL Parameter** (параметр, переданный методом GET), а в поле ввода, расположенном ниже и правее, введите имя этого параметра — `ID`.

Далее нам нужно заполнить элементы управления формы соответствующими им значениями. Эти значения, как вы поняли, будут взяты из только что созданного нами набора записей. Для разных элементов управления действия будут различными, и мы их сейчас опишем.

Сначала выделите область редактирования и найдите в редакторе свойств кнопку с изображением молнии — она находится правее области редактирования **Init Val**. Щелкните по этой кнопке. На экране появится диалоговое окно **Dynamic Data**; выберите в иерархическом списке **Field** пункт **Content**, соответствующий одноименному полю, и нажмите кнопку **OK**. Все, теперь область редактирования получит значение из поля `Content` набора записей.

С раскрывающимся списком все несколько сложнее. Выделите его и нажмите кнопку **Dynamic**. В появившемся на экране диалоговом окне **Dynamic List/Menu** щелкните кнопку с изображением молнии, расположенную правее поля ввода **Select Value Equal To**. После этого на экране опять же появится диалоговое окно **Dynamic Data**; выберите в иерархическом списке **Field** пункт **Category**, соответствующий одноименному полю, и нажмите кнопку **OK**. Теперь остается только нажать кнопку **OK** окна **Dynamic List/Menu**.

Можно сделать еще проще — перетащить нужное поле из панели **Bindings** на соответствующий ему элемент управления. Автор рекомендует вам так и поступать в дальнейшем.

Теперь нужно создать поведение `Update Record`, которое, собственно, и реализует заполнение формы и обновление значений полей редактируемой записи. Выберите пункт **Update Record** меню серверных поведений панели **Server Behaviors**. На экране появится диалоговое окно **Update Record** (рис. 18.1).

Выберите в раскрывающемся списке **Connection** базу данных, в списке **Table To Update** — таблицу, запись которой будет обновляться, а в списке **Select Record From** — набор записей, из которого будет взята запись для заполнения формы. В нашем случае это будет соответственно `Library`, `Sentences` и `Sentences` (соответственно, одноименные таблица и набор записей).

В поле ввода **Unique Key Column** задается поле набора записей, по которому нужная нам запись будет однозначно идентифицироваться. В нашем случае — это ключевое поле `ID`. Если это числовое поле (а поле `ID` — числовое), включите флажок **Numeric**, в противном случае — отключите его.

В поле ввода **After Updating, Go To** вводится интернет-адрес страницы, на которую будет осуществлен переход после успешного изменения записи. Введите в него `Sentences.asp`. Вы также можете щелкнуть кнопку **Browse** и выбрать нужную страницу в диалоговом окне **Select File**.

В раскрывающемся списке **Get Values From** выбирается форма, из которой будут взяты значения для обновления записи.

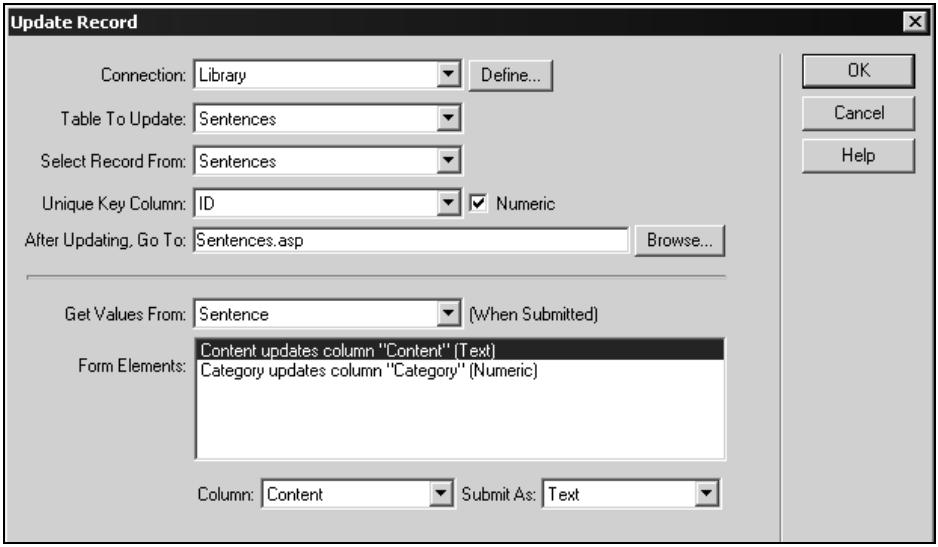


Рис. 18.1. Диалоговое окно **Update Record**

В списке **From Elements** перечислены все элементы управления, находящиеся в выбранной форме. Вы можете выбрать любой из них и задать для него поле таблицы, куда будут помещены содержащиеся в нем данные, и тип отправляемых данных (текст, число, дата и т. п.).

Поле таблицы выбирается в раскрывающемся списке **Column**. Пункт **<ignore>** данного списка позволяет проигнорировать значение этого элемента управления (т. е. оно не будет помещено в поле таблицы).

Тип данных указывается в раскрывающемся списке **Submit As**. В нем доступны следующие пункты:

- **Text** — текстовые данные;
- **Numeric** — числовые данные;
- **Date** — значение даты;
- **Date MS Access** — значение даты, предназначенное для Microsoft Access;
- **Checkbox Y,N** — логическое значение, преобразуемое в символы "Y" ("да") или "N" ("нет");
- **Checkbox 1,0** — логическое значение, преобразуемое в числа 1 ("да") или 0 ("нет");
- **Checkbox -1,0** — логическое значение, преобразуемое в числа -1 ("да") или 0 ("нет");
- **Checkbox MS Access** — логическое значение, предназначенное для Microsoft Access.

Задав все нужные данные, не забудьте нажать кнопку **OK**.

Сохраните готовую страницу EditSentence.asp и проверьте ее в действии. Для этого опубликуйте сайт на Web-сервере, откройте страницу Sentences.asp в Web-обозревателе и попробуйте отредактировать какое-либо высказывание. Если страница изменения записи почему-то не работает, проверьте, все ли правильно вы сделали.

Средства удаления записи

Осталось создать средства удаления ненужных высказываний из таблицы `Sentences`. Что ж, и такое иногда приходится проделывать...

Чтобы удалить ненужную запись, нам также понадобится серверная страница. Мы поместим на этой странице содержимое удаляемой записи (это хороший тон Web-программирования) и кнопку, запускающую процесс удаления записи. А для перехода на эту страницу мы поместим на странице Sentences.asp еще одну гиперссылку.

Откройте страницу Sentences.asp. Поставьте текстовый курсор в четвертую ячейку второй строки таблицы и наберите текст "Удалить". Выделите только что введенное слово и наберите в поле ввода интернет-адреса редактора свойств такой текст:

```
DeleteSentence.asp?ID=
```

Далее переключитесь в режим отображения кода и перетащите из панели **Bindings** в место, находящееся точно после знака "=", поле `ID` набора записей. Интернет-адрес гиперссылки примет такой вид:

Мы уже знаем, что означает этот код.

```
DeleteSentence.asp?ID=<%=(Sentences.Fields.Item("ID").Value)%>
```

Теперь сохраните и закройте страницу Sentences.asp. Создайте новую серверную страницу, введите в нее поясняющий текст и сохраните под именем DeleteSentence.asp в папке Admin. Но не закрывайте только что созданную страницу — мы ей сейчас займемся.

Сначала нужно создать набор записей, из которого будет взята запись для отображения на этой странице. Этот набор также должен содержать одну-единственную запись, поэтому нам придется задать фильтр по значению поля счетчика `ID` таблицы `Sentences`, которое должно равняться значению параметра гиперссылки `ID`, переданному от страницы Sentences.asp. Кроме того, этот набор записей должен быть сложным, аналогичным тому, что мы создали на странице Sentences.asp. Создайте его, а если у вас возникнут затруднения, обратитесь к спасительной *главе 17*. Текст SQL-запроса будет таков:

```
SELECT Sentences.ID, Content, Name FROM Sentences, Categories
  ↳WHERE Sentences.ID = MMColParam AND Sentences.Category=Categories.ID
```

Здесь `MMColParam` — имя параметра SQL-запроса, соответствующего параметру гиперссылки `ID`. Его значение, соответственно, будет `Request.QueryString("ID")`.

Теперь поместим на страницу динамический текст, привязанный к полю `Content` набора записей. В следующем абзаце наберем слово "Категория", поставим неразрывный пробел и поместим динамический текст, привязанный к полю `Name`.

В следующем абзаце создадим форму и назовем ее `Delete`. Эта форма нужна нам для того, чтобы поместить кнопку, запускающую процесс удаления записи. Поместим же эту кнопку в форму. И сохраним страницу.

Теперь нам нужно создать поведение `Delete Record`. Для этого выберем одноименный пункт в меню серверных поведений панели **Server Behaviors**. На экране появится диалоговое окно **Delete Record**, показанное на рис. 18.2.

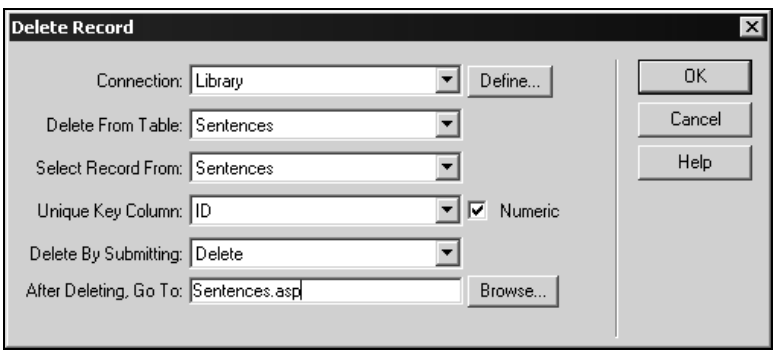


Рис. 18.2. Диалоговое окно **Delete Record**

Выберите в раскрывающемся списке **Connection** базу данных, в списке **Delete From Table** — таблицу, запись которой будет удалена, а в списке **Select Record From** — набор записей, из которого будет взята запись для отображения ее содержимого на странице. В нашем случае это будет соответственно `Library`, `Sentences` и `Sentences` (таблица и набор записей у нас называются одинаково).

В поле ввода **Unique Key Column** задается поле набора записей, по которому нужная нам запись будет однозначно идентифицироваться. В нашем случае — это ключевое поле `ID`. Если это числовое поле (а поле `ID` — числовое), включите флажок **Numeric**, в противном случае — отключите его.

В раскрывающемся списке **Delete By Submitting** выбирается кнопка, после нажатия которой произойдет удаление записи.

В поле ввода **After Deleting, Go To** вводится интернет-адрес страницы, на которую будет осуществлен переход после успешного удаления записи. Вве-

дите в него Sentences.asp. Вы также можете щелкнуть кнопку **Browse** и выбрать нужную страницу в диалоговом окне **Select File**.

После ввода всех данных не забудьте нажать кнопку **OK**. Сохраните страницу DeleteSentence.asp и проверьте ее. Введите какую-нибудь запись и попробуйте ее удалить. Если она почему-то не удалится, проверьте, все ли вы сделали правильно.

Страница входа на сайт

Поскольку наш сайт будет поддерживать разграничение доступа, нам нужно обязательно сделать страницу входа на сайт. А, кроме того, нам необходимо создать в базе данных Library.asp таблицу, которая будет хранить список зарегистрированных посетителей.

Первым делом нужно, разумеется, создать таблицу, чтобы нам было с чего начинать создание страницы входа. Эта таблица будет содержать следующие поля:

- Name — имя посетителя (текстовое, 20 символов, ключевое);
- Password — пароль посетителя (текстовое, 20 символов);
- Grants — права посетителя (текстовое, 20 символов).

С первыми двумя полями все ясно. Поговорим о третьем поле — Grants. Dreamweaver, точнее, его поведения, "отвечающие" за разграничение доступа, принимают текстовые значения, говорящие о *правах посетителя*. Таких значений может быть, как минимум, три:

- "Administrator" — администратор сайта, полный доступ ко всем данным сайта;
- "User" — посетитель, доступ к открытым данным сайта, предназначенным для "обычных" посетителей сайта;
- "Guest" — *"гость" сайта*, минимальный доступ к данным сайта.

Основываясь на этих значениях, поведения Dreamweaver принимают решение, допускать ли посетителя до какой-либо страницы сайта. Конечно, вы можете задать свои значения, но эта троица присутствует практически во всех списках доступа.

Чтобы не усложнять чрезмерно наш сайт, давайте ограничимся двумя типами прав посетителей: администратором (значение "Administrator" поля Grants) и обычным посетителем (значение "User"). Администратор будет один и может править любые данные, сохраненные в базе Library.asp, а обычный посетитель сможет только их просматривать.

Итак, мы создали новую таблицу. Но подождите ее сохранять. Задайте для поля Grants значение по умолчанию "User", чтобы вновь зарегистрировав-

шиеся посетители сразу же получили соответствующие права доступа. Вот теперь можете закрывать и сохранять новую таблицу. Назовите ее `Users`.

После этого откройте только что созданную таблицу и введите в нее одну запись. Значения ее полей будут такими:

- поле `Name` — "`Admin`";
- поле `Password` — "`Admin`" (не будем заботиться о секретности пароля — все равно наш сайт учебный);
- поле `Grants` — "`Administrator`".

Теперь можно приступать к созданию страницы входа на сайт. Введите в нее поясняющий текст и назовите Login.asp. Причем сохраните ее в корневой папке сайта, а не в папке Admin, — мы используем ее для входа на сайт и администраторов, и обычных посетителей.

Теперь нам нужно создать форму с двумя полями ввода — соответственно, для ввода имени и пароля. Ну и, конечно, понадобится кнопка отправки данных. Создайте форму и назовите ее `Login`. Поля ввода назовите соответственно `Name` и `Password`. Не забудьте текстовые подписи, иначе посетитель вашего сайта не поймет, что куда нужно вводить. После этого сохраните страницу.

Теперь создадим поведение `Log In User`. Для этого выберите одноименный пункт в подменю **User Authentication** меню серверных поведений панели **Server Behaviors**. На экране появится диалоговое окно **Log In User**, показанное на рис. 18.3.

В раскрывающемся списке **Get Input From Form** выбирается форма, из которой будут взяты введенные посетителем имя и пароль. Поле ввода имени выбирается в раскрывающемся списке **Username Field**, а поле ввода пароля — в списке **Password Field**.

В раскрывающемся списке **Validate Using Connection** выбирается база данных, зарегистрированная в Dreamweaver. В списке **Table** выбирается таблица, содержащая список посетителей, в списке **Username Column** — поле, в котором хранятся имена посетителей, а в списке **Password Column** — поле, в котором хранятся пароли.

Очень часто на главной странице сайта делается гиперссылка на страницу входа (а иногда сама страница входа делается главной). В этом случае посетитель после успешного входа перенаправляется на страницу с "закрытыми" данными. В поле ввода **If Login Succeeds, Go To** как раз и вводится интернет-адрес страницы, на которую будет осуществлен переход, если посетитель успешно войдет на сайт. Введите в это поле имя страницы списка высказываний Admin/Sentences.asp. Вы также можете нажать кнопку **Browse** и выбрать нужную страницу в диалоговом окне **Select File**.

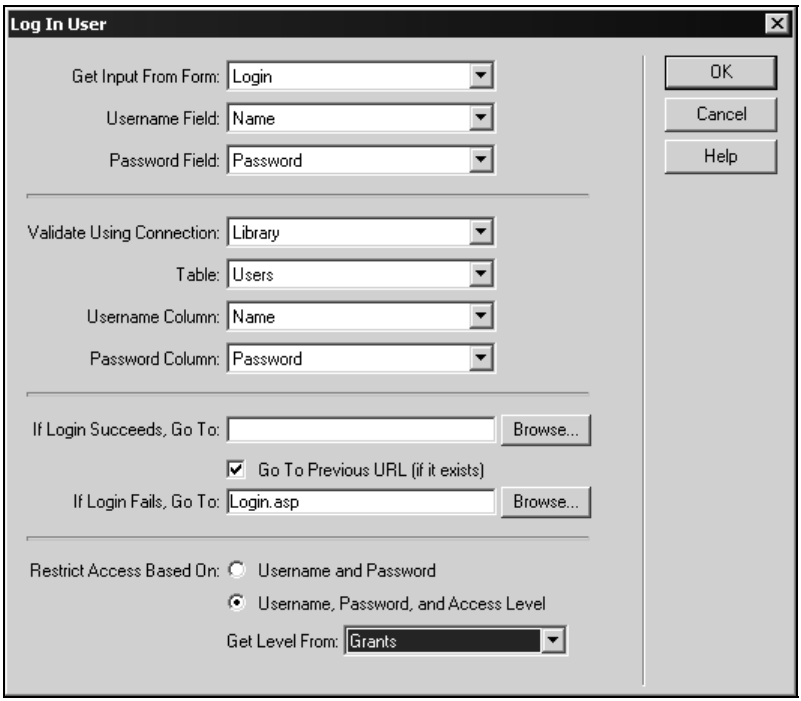


Рис. 18.3. Диалоговое окно **Log In User**

Иногда система разграничения доступа функционирует следующим образом. При попытке захода на не доступную "посторонним" страницу посетитель автоматически перенаправляется на страницу входа на сайт. После ввода имени и пароля, если этот посетитель занесен в список, он возвращается на страницу, на которую хотел войти изначально. Для такого случая предусмотрен флажок **Go To Previous URL (if it exists)**. Включите его, если хотите, чтобы система разграничения доступа работала таким образом.

В поле ввода **If Login Fails, Go To** вводится интернет-адрес страницы, на которую будет осуществлен переход, если посетителю не удалось войти на сайт. (Такое может быть, например, из-за того, что посетитель не внесен в список или неправильно набрал имя или пароль.) Введите в это поле имя страницы входа Login.asp; таким образом, после неудачного входа посетитель окажется на той же странице. Вы также можете нажать кнопку **Browse** и выбрать нужную страницу в диалоговом окне **Select File**.

С помощью группы переключателей **Restrict Access Based On** вы можете задать, на основе чего производится разграничение доступа к Web-страницам. Если включен переключатель **Username and Password**, разграничение доступа производится на основе имени и пароля. Если же вы хотите "вовлечь" в этот процесс еще и права доступа посетителя, включите переключатель **Username,**

Password, and Access Level. В этом случае становится доступным раскрывающийся список **Get Level From**, в котором вы сможете выбрать поле таблицы, выбранной в списке **Table**, откуда будут взяты права посетителя. В нашем случае — это поле `Grants`.

Введя все данные, нажмите кнопку **OK**.

Вы, наверно, подумали, что сейчас мы будем проверять вновь созданную страницу в работе. Но делать это пока еще рано. Нам нужно защитить остальные страницы от *несанкционированного доступа*, т. е. попытки войти на них, миновав страницу входа на сайт. Без этого наши старания по защите сайта от "случайных людей" пойдут насмарку.

Защита страниц от несанкционированного доступа

В нашем случае стоит защитить только страницу Sentences.asp, т. к. лишь на ней отображаются какие-то данные. Остальные страницы (AddSentence.asp, EditSentence.asp и DeleteSentence.asp) выполняют вспомогательные функции и без страницы Sentences.asp работать не будут.

Итак, откроем страницу Sentences.asp. Чтобы защитить ее от непрошенных посетителей, создадим поведение `Restrict Access To Page`. Выберем одноименный пункт в подменю **User Authentication** меню серверных поведений панели **Server Behaviors**. На экране появится диалоговое окно **Restrict Access To Page**, показанное на рис. 18.4.

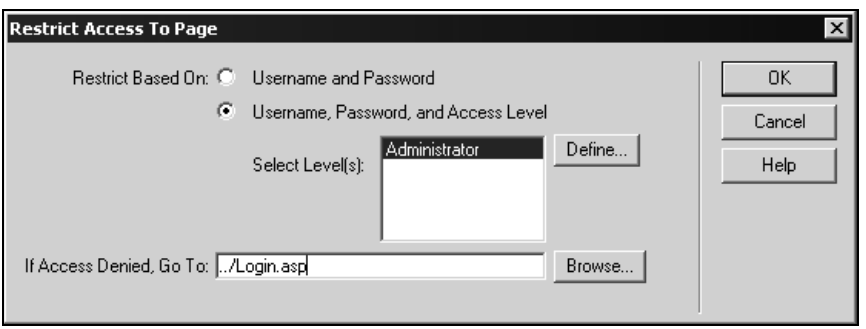


Рис. 18.4. Диалоговое окно **Restrict Access To Page**

С помощью группы переключателей **Restrict Based On** вы можете задать, на основе чего производится разграничение доступа к данной странице. Если включен переключатель **Username and Password**, разграничение доступа производится на основе только имени и пароля. Если же вы хотите "вовлечь" в этот процесс еще и права доступа посетителя, включите переключатель **Username, Password, and Access Level**. В этом случае становится доступным

список **Select Level(s)**, в котором вы сможете выбрать права для посетителей, которые смогут увидеть эту страницу.

Изначально список **Select Level(s)** пуст. Поэтому нам придется его заполнить. Щелкните кнопку **Define**, и на экране появится диалоговое окно **Define Access Levels**, показанное на рис. 18.5. В этом окне и задаются права доступа.

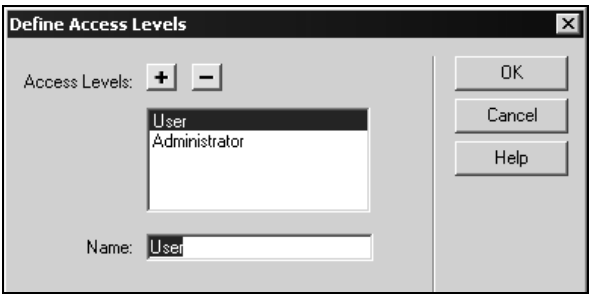


Рис. 18.5. Диалоговое окно **Define Access Levels**

Большую часть этого окна занимает список уже созданных вами прав. Чтобы добавить в него новый пункт, введите его имя в поле ввода **Name** и нажмите кнопку со знаком "плюс". Чтобы изменить пункт, выберите соответствующее имя в списке, измените его в поле ввода **Name** и переключитесь на любой другой пункт списка. Если же вам требуется удалить ненужный пункт, выберите соответствующее имя и нажмите кнопку со знаком "минус". К сожалению, эту операцию вам придется выполнять почти всегда: Dreamweaver почему-то часто добавляет в список пустой пункт. Введя все права посетителей, нажмите кнопку **OK**.

Введя в список **Select Level(s)** права посетителей, вы можете их выбрать, щелкнув мышью по нужному пункту. В нашем случае, выберите пункт **Administrator**. Если вам необходимо выбрать сразу несколько пунктов, щелкните по первому из них, а потом щелкайте по остальным, удерживая нажатой клавишу <Ctrl>.

В поле ввода **If Access Denied, Go To** вводится интернет-адрес страницы, на которую будет осуществлен переход, если посетителю запрещено входить на эту страницу или если посетитель не выполнил процедуру входа на сайт. Введите в это поле имя страницы входа `Login.asp`. Вы также можете нажать кнопку **Browse** и выбрать нужную страницу в диалоговом окне **Select File**.

Введя все данные, нажмите кнопку **OK**.

Вот теперь можете проверить созданную нами систему разграничения доступа. Опубликуйте сайт на Web-сервере и попробуйте открыть страницу Sentences.asp. Если она все же откроется, выполните ее перезагрузку в окне Web-обозревателя. После этого вы уж точно окажетесь на странице входа на сайте Login.asp.

Реализация выхода с сайта

Осталось реализовать выход с сайта — и работу над административными страницами можно считать законченной. Как вы помните, каждый посетитель, вошедший на сайт с разграничением доступа, должен по окончании работы с него выйти, чтобы освободить системные ресурсы серверного компьютера. Так что выход с сайта так же важен, как и вход на него.

Выход с сайта можно реализовать двумя способами. Во-первых, можно предусмотреть на каждой из страниц, отображающей "закрытые" данные, особую гиперссылку, запускающую процесс выхода. Во-вторых, для этого можно предусмотреть особую серверную Web-страницу, вход на которую автоматически вызовет выполнение действий по выходу с сайта. Вы можете использовать любой из этих способов, мы же выберем первый.

Откройте страницу Sentences.asp. Поместите где-либо на ней текст "Выход с сайта". Выделите его и выберите в подменю **User Authentication** меню серверных поведений панели **Server Behaviors** пункт **Log Out User**. На экране появится диалоговое окно **Log Out User**, показанное на рис. 18.6.

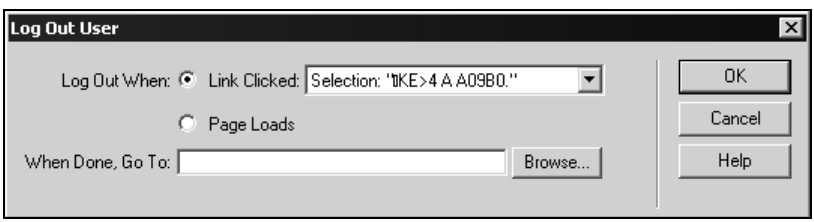


Рис. 18.6. Диалоговое окно **Log Out User**

С помощью набора переключателей **Log Out When** выбирается момент времени, когда выполняется выход посетителя с сайта. Если вы хотите, чтобы выход выполнился при щелчке по гиперссылке (это как раз наш случай), включите переключатель **Link Clicked**. Если же вы хотите, чтобы выход выполнился при входе на данную Web-страницу, включите переключатель **Page Loads**.

В поле ввода **When Done, Go To** вводится интернет-адрес страницы, на которую будет осуществлен переход после выполнения процедуры выхода с сайта. Введите в это поле имя страницы входа Login.asp. Вы также можете нажать кнопку **Browse** и выбрать нужную страницу в диалоговом окне **Select File**.

Введя все данные, нажмите кнопку **OK**.

Проверьте готовую страницу в работе. Для этого опубликуйте сайт на Web-сервере, откройте в Web-обозревателе страницу Sentences.asp, войдите на сайт и тут же выйдите с него.

Все, на этом создание административных страниц можно считать законченным. Теперь приступим к созданию страниц общего доступа, предназначенных для обычных посетителей нашего сайта.

Страницы общего доступа

Страницы общего доступа предназначены для обычных посетителей сайта. Как правило, они создаются после административных страниц; в этом случае вы можете использовать последние как образцы для создания "обычных" страниц. В самом деле, "обычные" страницы часто в общих чертах повторяют административные, и вам остается только удалить некоторые возможности, не нужные обычным посетителям. Так мы и сделаем.

Страница списка категорий

Давайте сделаем так, чтобы при входе на сайт посетитель мог выбрать категорию высказываний (зря мы их, что ли, вводили!). А для этого нам понадобится создать страницу списка категорий, чем мы сейчас и займемся.

Создадим новую серверную страницу ASP и сохраним ее в файле Categories.asp в корневой папке сайта. И остановимся, чтобы немного подумать.

Что нам нужно? Правильно, вывести список категорий. А еще? Нам необходимо, чтобы название каждой категории в этом списке представляло собой гиперссылку. Щелкнув по ней, посетитель сайта попадет на другую страницу, на которой будут перечислены все высказывания этой категории.

Как это сделать? Очень просто: надо воспользоваться поведением `Go To Detail Page`, выполняющим переход на связанную страницу.

Создадим новый набор записей под именем `Categories`. Задайте выборку всех записей таблицы `Categories`. После этого задайте *сортировку* записей по полю `Name`, для чего выберите это поле в раскрывающемся списке **Sort** диалогового окна **Recordset** (см. рис. 17.15). Раскрывающийся список, расположенный правее, позволит вам задать порядок сортировки записей: по возрастанию (пункт **Ascending**) или по убыванию (пункт **Descending**).

Напишите на странице Categories.asp какой-либо пояснительный текст, если хотите. После этого создайте текстовый абзац и поместите в него динамический текст, связанный с полем `Name` набора данных `Categories`. Далее выделите этот текст и создайте "вокруг" него повторяющуюся область, привязав ее к созданному нами набору данных `Categories`. Пусть она отображает сразу все записи набора — их не так много, чтобы городить огород с навигатором и строкой статуса.

Собственно, на этом создание страницы списка категорий почти закончено. Осталось только добавить поведение, реализующее переход на страницу со списком высказываний, относящихся к выбранной категории.

Чтобы нам самим было проще в дальнейшем, давайте введем еще пару терминов. Назовем страницу, отображающую список категорий, *первичной*, а страницу, отображающую список высказываний, относящихся к выбранной категории, — *вторичной*. В литературе встречаются другие названия таких страниц, но мы будем использовать эти.

Выделите содержимое повторяющейся области, а именно динамический текст. После этого выберите в меню серверных поведений панели **Server Behaviors** пункт **Go To Detail Page**. На экране появится диалоговое окно **Go To Detail Page** (рис. 18.7).

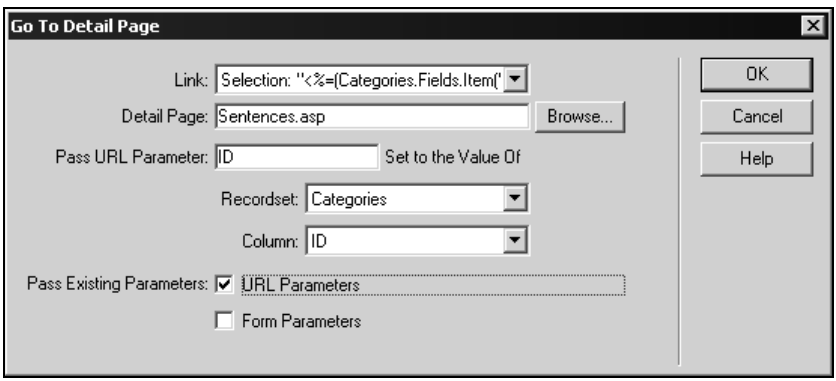


Рис. 18.7. Диалоговое окно **Go To Detail Page**

В поле ввода **Detail Page** введите интернет-адрес вторичной страницы, отображающей список высказываний. Вы также можете щелкнуть по кнопке **Browse** и выбрать нужную страницу в диалоговом окне **Select File**. Мы создадим эту страницу позднее и назовем ее так же, как уже созданную административную страницу списка высказываний, — Sentences.asp, только сохраним в корневой папке сайта.

В поле ввода **Pass URL Parameter** введите имя параметра, передаваемого из первичной страницы во вторичную. По умолчанию Dreamweaver подставляет в него имя первого поля ввода набора записей, и, если это то поле, которое вам нужно, менять его не стоит.

В раскрывающемся списке **Recordset** выбирается набор записей, чье поле используется для связи первичной и вторичной страниц. Само же это поле выбирается в раскрывающемся списке **Column**. В нашем случае это набор записей `Categories` и его поле `ID`.

Теперь остается задать метод передачи параметра из первичной страницы во вторичную. Это делается с помощью группы флажков **Pass Existing Parameters**. Флажок **URL Parameters** задает использование метода GET, а флажок **Form Parameters** — метода POST. Нам лучше выбрать метод GET, т. к.

объем передаваемых данных очень невелик, а хранить их в секрете необязательно.

Введя все данные, нажмите кнопку **OK**. После этого Dreamweaver преобразует выделенный фрагмент страницы в гиперссылку, выполняющую переход на вторичную страницу. И, разумеется, создаст поведение `Go To Detail Page`.

Теперь нужно ограничить доступ к этой странице. Для этого, как вы помните, нужно создать поведение `Restrict Access To Page`. Задайте разграничение доступа по имени и паролю (переключатель **Username and Password** диалогового окна **Restrict Access To Page**). Таким образом, на эту страницу смогут попасть все зарегистрированные посетители вне зависимости от прав доступа.

На этом создание страницы списка категорий можно считать законченным. Наш следующий шаг — создание страницы списка высказываний.

Страница списка высказываний

В данном случае лучше всего создать страницу списка высказываний "с нуля". Если же мы решим создать ее на основе созданной ранее административной страницы Sentences.asp, нам придется вносить в нее слишком много изменений. Поэтому создадим новую серверную страницу, напишем на ней поясняющий текст и сохраним под именем Sentences.asp в корневой папке сайта.

После этого создадим набор записей `Sentences`, извлекающий данные из только что созданной таблицы. Зададим для него фильтр по полю `Category`. Для этого выберите название этого поля в раскрывающемся списке **Filter** диалогового окна **Recordset**, а в раскрывающемся списке, расположенном правее, — пункт = (равенство значений поля и фильтра). Далее в списке, расположенном ниже, выберите пункт **URL Parameter** (параметр, переданный методом GET), а в поле ввода, расположенном ниже и правее, введите имя этого параметра — `ID`.

Кроме того, мы можем не извлекать из таблицы все поля, а ограничимся только полем `Contents`. Включим переключатель **Selected** в группе **Column** диалогового окна **Recordset** и выберем в расположенном ниже списке пункт **Content**, соответствующий одноименному полю таблицы. Поля `ID` и `Category` нам в данном случае не нужны, потому что их содержимое никак не отображается на странице. Это позволит нам уменьшить нагрузку на серверный компьютер, который и без того сильно загружен, ведь для хранения содержимого полей таблиц расходуется его оперативная память.

Теперь поместите в текстовом абзаце динамический текст, привязанный к полю `Content` набора записей, выделите весь абзац, щелкнув по соответст-

вующей кнопке секции тегов, и создайте "вокруг" него повторяющуюся область. Затем сделайте навигатор и строку статуса набора данных.

После этого поместите повторяющуюся область, навигатор и строку статуса внутрь необязательной области, отображающейся только при непустом наборе записей `Sentences`. И создайте еще одну необязательную область, отображающуюся, наоборот, при пустом наборе записей и содержащую текст "Список высказываний пуст". Как видите, мы повторяем шаги, проделанные нами при создании административной страницы Sentences.asp.

Последний шаг (его мы тоже проделывали) — это защита страницы от несанкционированного доступа и предоставление возможности выхода с сайта. Создайте соответствующие поведения и гиперссылку для выхода. Когда будете создавать поведение `Restrict Access To Page`, задайте разграничение доступа по имени и паролю (переключатель **Username and Password** диалогового окна **Restrict Access To Page**). После этого сохраните готовую страницу.

Теперь проверьте готовые страницы. Для этого опубликуйте сайт на Web-сервере, откройте в Web-обозревателе страницу Categories.asp, войдите на сайт (под именем "Admin" — других посетителей в списке нет), выберите категорию и просмотрите отобранные серверной программой высказывания. После этого выйдите с сайта. Если вы все сделали правильно, все должно работать.

Страница регистрации посетителя

Последняя из страниц общего доступа, которую нам нужно создать, — это страница регистрации нового посетителя нашего сайта. В самом деле, если посетитель не сможет зарегистрироваться на нашем сайте, занеся сведения о себе в список посетителей, как же он сможет добраться до содержимого этого сайта?

Создайте новую серверную страницу, поместите на ней поясняющий текст и сохраните под именем Register.asp в корневой папке нашего сайта. Поместите в нее форму и назовите `User`. В этой форме создайте поля ввода `Name` (имя посетителя) и `Password` (пароль) и кнопку отправки данных.

Для регистрации посетителя в списке используется хорошо вам знакомое поведение `Insert Record`. В самом деле, если список посетителей представляет собой таблицу базы данных, а сведения о каждом посетителе — запись в этой таблице, то именно этим поведением и стоит пользоваться. Создайте же его. Пусть содержимое поля ввода `Name` заносится в поле `Name` таблицы `Users`, а содержимое поля ввода `Password` — в поле `Password`. Поле `Grants`, в котором записываются права посетителя, получит значение по умолчанию "`User`" (обычный посетитель). А в качестве страницы, на которую будет выполнен переход после успешного добавления новой записи, задайте

Categories.asp — в этом случае новоиспеченный посетитель сразу сможет зарегистрироваться и просмотреть плоды нашего ума.

Вроде бы все прекрасно. Посетитель заходит на эту страницу, вводит данные о себе и сразу же получает доступ к сайту. Однако это та самая простота, которая хуже воровства. Почему? Сейчас увидим.

Предположим, какой-то посетитель успешно зарегистрировался на нашем сайте, получил к нему доступ и успешно же вышел с него. После этого на сайте регистрируется второй посетитель. И вводит то же самое имя, под которым уже зарегистрировался первый посетитель!

Если вы не предусмотрите специальных средств для недопущения совпадения имен посетителей (например, не сделаете поле таблицы, где хранится имя, уникальным), это может породить конфликт. А такие конфликты ни к чему хорошему не ведут — они ведут только к плохому!

Именно для недопущения таких случаев Dreamweaver предусматривает особое поведение `Check New Username`. Это поведение проверяет таблицу списка посетителей на предмет совпадения введенного в форме регистрации имени с уже имеющимися в таблице. Если введенного новым посетителем имени нет в таблице, поведение `Check New Username` запускает поведение `Insert Record`, успешно добавляющее в список нового посетителя.

Давайте создадим поведение `Check New Username`. Но сначала проверим, действительно ли мы создали поведение `Insert Record`.

Выберите в подменю **User Authentication** меню серверных поведений панели **Server Behaviors** пункт **Check New Username**. На экране появится диалоговое окно **Check New Username**, показанное на рис. 18.8.

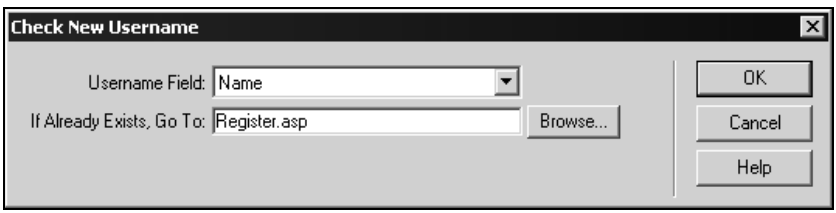


Рис. 18.8. Диалоговое окно **Check New Username**

В раскрывающемся списке **Username Field** выберите поле ввода, в котором вводится имя посетителя.

В поле ввода **If Already Exists, Go To** вводится интернет-адрес страницы, на которую будет осуществлен переход, если в списке будет найден посетитель с таким именем. Введите в это поле имя страницы регистрации `Register.asp`. Вы также можете нажать кнопку **Browse** и выбрать нужную страницу в диалоговом окне **Select File**.

Завершив ввод данных, нажмите кнопку **OK**. И сохраните готовую страницу.

Осталось сделать совсем немного. А именно — поместить на страницу входа на сайт Login.asp гиперссылку, указывающую на страницу регистрации. Это нужно для того, чтобы новый посетитель сайта смог сразу же после открытия его зарегистрироваться в списке. Сделайте это.

Теперь можете проверить готовую страницу в действии. Опубликуйте сайт на Web-сервере, откройте в Web-обозревателе страницу входа Login.asp, зарегистрируйтесь на сайте под другом именем, скажем, "User", и попробуйте войти на сайт.

Реализация поиска высказываний

Все сайты, публикующие информацию из баз данных, предусматривают возможность *поиска записей* на основе какого-либо критерия. Давайте и мы сделаем возможность поиска высказываний по введенному посетителем слову.

Чтобы посетитель смог ввести слово для поиска, нам нужно создать где-то форму. Давайте поместим ее на странице списка категорий Categories.asp под собственно списком категорий, в новом текстовом абзаце. Назовем форму `Search`, поле ввода слова для поиска — `Content` и не забудем о кнопке отправки данных.

Теперь выделим форму и с помощью редактора свойств введем следующие параметры:

- **Action** — имя страницы, отображающей результаты поиска, а именно — Result.asp;
- **Method** — метод отправки данных, в нашем случае — GET;
- **Enctype** — кодировка отправляемых данных, а именно — `application/x-www-form-urlencoded`.

Теперь, если посетитель введет слово и нажмет кнопку отправки данных, это слово будет отправлено странице Result.asp как параметр гиперссылки `Content`.

Далее нужно создать страницу отображения результатов поиска. Создадим ее на основе уже существующей страницы списка высказываний Sentences.asp. Для этого откроем ее и сохраним под именем Result.asp в корневой папке сайта. После этого откроем новую страницу.

В панели **Bindings** удалим параметр гиперссылки `ID`, если он там есть. После этого откроем диалоговое окно **Recordset** и изменим кое-какие параметры фильтра. В раскрывающемся списке **Filter** выберем пункт **Content** (имя поля, по которому мы будем осуществлять поиск) а в раскрывающемся спи-

ске, расположенном правее, — пункт **contains** (поле должно содержать введенное посетителем слово). Далее в списке, расположенном ниже, выберем пункт **URL Parameter** (параметр, переданный методом GET), а в поле ввода, расположенном ниже и правее, введем имя этого параметра — `Content`. И нажмем кнопку **OK**.

Сохраним страницу Result.asp и опубликуем сайт на Web-сервере. Войдем на сайт, введем в форму поиска какое-либо слово и нажмем кнопку отправки данных. В результате мы получим страницу со списком найденных высказываний.

Все, наш сайт готов!

Вы можете расширить его. Например, сделать Web-интерфейс для работы со списками категорий и посетителей. Также вы можете добавить дизайнерские изыски — этим мы не занимались вообще. В общем, поработайте над сайтом, практика пойдет вам на пользу.

Что дальше?

А дальше... все! Книга закончилась.

Заключение

"А напоследок я скажу..." — поется в старинном русском романсе. Что же мне сказать напоследок?..

Книга о Dreamweaver закончена. Я рассказал об этой программе все, что знал и что мог найти в поставляемой с ним электронной документации и других информационных ресурсах. Я поделился своими соображениями насчет принципов Web-дизайна и различных интернет-технологий, еще современных и уже устаревших. Вы были озадачены огромным количеством новых терминов, имеющих хождение среди интернетчиков и Web-дизайнеров. Я продемонстрировал вам множество фрагментов исходного кода на HTML и JavaScript и дал множество полезных (надеюсь) советов. Кроме того, я успел надоесть многим из вас своими остротами. Но книга закончена, и я больше не буду вам докучать.

Но значит ли это, что я рассказал о Macromedia Dreamweaver MX все? Нет.

Любой достаточно сложный программный продукт таит в себе обилие возможностей. Их так много, что им посвящают толстенные книги, в которых, опять же, много остается неописанным. Возьмите ту же самую Microsoft Windows — сколько чудес она преподносит даже опытным пользователям! (Не надо говорить про ее ошибки — в других операционных системах ошибок не меньше, если не больше.) Дома у меня лежит толстенная тысячестраничная книга по Windows 2000, но, даже прочитав ее от корки до корки, я бы не сказал, что разобрался в ней досконально. И так обстоит дело с любым объемным программным продуктом.

В том числе, и с Dreamweaver.

Dreamweaver — мощный и сложный программный пакет. Я просто не мог описать все его возможности в одной книге.

- Я не описал механизма автоматизации выполнения повторяющихся задач. Этот механизм позволяет вам выполнять весьма сложные действия одним щелчком мыши.

- Я не описал механизма *расширений* Dreamweaver. Этот механизм позволяет вам создавать свои собственные способы поведения, модули расширения, вызываемые из меню главного окна программы и т. п.
- Я даже не описал многих возможностей, предлагаемых Dreamweaver для разработчиков серверных страниц. Серверное программирование — это целый мир, описать который в четырех главах не представляется никакой возможности — для этого нужно писать еще одну книгу, толщиной никак не менее этой. (Хотя, основные возможности серверного программирования в Dreamweaver я все же описал.)

Да, я много о чем не рассказал. Но все это описано в электронной документации. Правда, она полностью английская, но опытный компьютерщик обязан знать этот язык.

Кроме того, существует Интернет, Всемирная сеть, в которой можно найти все, что угодно. Сейчас я приведу список интернет-ресурсов, посвященных Web-дизайну, HTML и другим интернет-технологиям. И, конечно, самому Dreamweaver (табл. 3.1).

Таблица 3.1. *Список интернет-ресурсов по теме книги*

Интернет-адрес ресурса	Описание
	Зарубежные ресурсы
http://www.macromedia.com/	"Домашний" сайт фирмы Macromedia, разработчика Dreamweaver. Посвящен как самому Dreamweaver, так и другим продуктам этой фирмы (Fireworks, Flash, Shockwave и др.)
http://msdn.microsoft.com/ie/	"Дом" Internet Explorer. Исчерпывающее описание Web-обозревателя фирмы Microsoft, особенностей поддержки им HTML, CSS, JavaScript и многого другого. К несчастью, читать эту документацию можно только с сайта
http://developer.netscape.com/	"Дом" Netscape Navigator. Исчерпывающее описание Web-обозревателя фирмы Netscape, особенностей поддержки им HTML, CSS, JavaScript и многого другого. Почти вся документация доступна для загрузки в виде ZIP-архивов
http://www.w3c.org/	Сайт самого великого и ужасного комитета WWWC. Все интернет-стандарты в удобном для чтения виде
http://webreview.com/	Огромное количество документации по HTML, CSS, JavaScript, программам Web-обозревателей

Таблица 3.1 *(окончание)*

Интернет-адрес ресурса	Описание
	Отечественные ресурсы
http://docs.rinet.ru/	Огромное количество документации, в том числе, по HTML и JavaScript. Правда, по-английски
http://www.citforum.ru/	"Город" электронной документации по компьютерным и интернет-технологиям. Все по-русски. В числе прочего, есть и переводы стандартов с сайта комитета WWWC. Настоятельно рекомендую с ними ознакомиться
http://webims.virtualave.net/	Начала Web-дизайна и Web-программирования для самых начинающих. Большое количество примеров. Все по-русски
http://subscribe.ru/	Сервер почтовых рассылок. Поищите рассылки, связанные с интернет-технологиями, – их там очень много. Там есть и рассылка, посвященная Dreamweaver

Ну, вот и все. Книга закончилась, и я с вами прощаюсь. До новых встреч!

Владимир Дронов, **vlad@vgi.volsu.ru**

Предметный указатель

A

ASP 611
AVI 154

B

BMP 124

C

CGI 609
COM 571
◊ клиент 571
◊ контейнер 571
◊ сервер 571
CSS 385
CSV 182

D

DOM 513

F

Flash 125, 140
◊ изображение 146
◊ кнопка 144
◊ надпись 141
FTP 107
◊ пассивный 224

G

GIF 124
◊ анимированный 125
◊ однопиксельный 194
GUID 151

H

HTML 18
HTML-редактор 30
HTML-стиль 98
◊ абзаца 99
◊ символа 99

I

ISAPI 610

J

Java 470
◊ апплет 572
◊ файл класса 573
JavaScript 470
JPEG 124
JScript 470
JSP 611

M

MIDI 155
MIME 149

N

NSAPI 610

O

ODBC 660
◊ драйвер 661
◊ источник данных 660

P

PNG 124

S

Shockwave 146
SQL 621
SSH 224

V

VBScript 470
VML 124
VRML 123

W

WAV 151
Web-дизайнер 13
Web-интерфейс 697
Web-обозреватель:
◊ Internet Explorer 28
◊ Mozilla 28
◊ Navigator 28
◊ Opera 28
Web-редактор 29
◊ визуальный 30
◊ гибридный 30
◊ невизуальный 30
Web-сайт 213
◊ локальная копия 221
◊ планирование 212
◊ полезное содержимое 215
◊ регистрация 221
◊ удаленная копия 223
Web-страница:
◊ административная 699
◊ архив новостей сайта 215
◊ вторичная 715
◊ входа на сайт 696, 708
◊ выхода с сайта 697
◊ главная 27, 214
◊ динамическая 609
◊ заголовок 20
◊ заставка 214
◊ интерактивная 468
◊ карта сайта 216
◊ название 20
◊ неинтерактивная 468
◊ новости сайта 215
◊ общего доступа 714
◊ общие свойства 110
◊ первичная 715
◊ по умолчанию 27
◊ просмотр 62
◊ регистрации посетителя сайта 697, 717
◊ сведения о контакте 216
◊ сведения о разработчиках 215
◊ создание 75, 235
◊ статическая 609
◊ тело 20
◊ уникальное содержимое 348
World Wide Web 23
◊ Consortium 22

X

XHTML 22
XML 22

Z

z-индекс 439

А

Автономный режим 27
Администратор 27, 697
Активная серверная страница 609
Активы 255
◊ категория 255
Алгоритм 499
Анимация 472, 481, 494
◊ дорожка 478, 481
◊ зацикленная 474, 492
◊ канал 481
◊ канал поведений 564
◊ реального времени 477
◊ частота кадров 492
Архитектура 26
◊ двухзвенная 26
◊ клиент-сервер 26
◊ однозвенная 26
◊ одноранговая 26
Атрибут 20
◊ без значения 94
◊ динамический 687
◊ изменяемый 368
◊ необязательный 24
◊ обязательный 23
◊ серверной директивы 597
◊ стиля 386

Б

База данных 619
◊ закрытие 620
◊ открытие 620
◊ процессор 619
◊ регистрация 663
◊ реляционная 619
Байт-код 572
Баннер 480
Библиотека 259
Блокировка 674
◊ оптимистическая 674
◊ пессимистическая 674
◊ тип 674
Брандмауэр 228

В

Ветвление 504
Виртуальная машина 470
Вложенность тегов 20, 92
◊ уровень 21
Врезка 307
Вход на сайт 696
Выравнивание 79
Выражение 502
◊ составное 505
Выход с сайта 697, 713

Г

Гиперссылка 23
◊ внешняя 243
◊ замена по сайту 243
◊ изображение 132
◊ мертвая 242
◊ оборванная 242
◊ создание 102, 123, 132, 236
◊ цель 104, 292
◊ якорь 108
Горячая область 135
"Гость" 708

Д

Дизайн Web-страниц:
◊ схема 269
◊ табличный 306
◊ фреймовый 268
Док 38

Е

Единица измерения CSS 401

З

Замещение 286
Запись 619
◊ поиск 620
◊ текущая 620
Запрос клиентский 26
Захват 448
Значение:
◊ цикла конечное 506
◊ цикла начальное 506

И

Идентификатор интервала 522
Иерархия тегов 20

Изображение:
◊ активное 134
◊ векторное 125
◊ вставка 126
◊ выравнивание 130
◊ изменение размеров 128
◊ карта 135
◊ перекрывающее 134
◊ плавающее 131
◊ распорка 330
◊ растровое 124
◊ составное 204
◊ фоновое 147
◊ черновое 130
Имя:
◊ нового окна Web-обозревателя 545
◊ пользователя 107
◊ формы 623
◊ фрейма 284
◊ элемента Web-страницы 514
Индексирование 579
Инкремент 506
Инструментарий 39
◊ документа 39
◊ отладчика 568
◊ панели Site 229
◊ стандартный 39
Интернет 17
◊ сервис 106
Интернет-адрес 23
◊ абсолютный 237
◊ относительный 237
Интерпретатор 470
Интерфейс:
◊ многодокументный 35
◊ однодокументный 35
История Web-обозревателя 77

К

Каскадная таблица стилей 385
Каскадность 394
Каталог ссылок 578
Класс 509
◊ конструктор 510
◊ системный 511
◊ создание 510
◊ экземпляр 509
Клиент 26
◊ FTP 107
◊ новостей 107
Ключевое слово 504, 576
Кнопка 628
◊ графическая 636
◊ надпись 628
◊ отправки данных 629
◊ переключатель 79
◊ потайная 51
◊ сброса формы 629
Кодировка 66
Комментарий 115
Компилятор 471
Константа 502
Курсор текстовый 43
Кэш 27, 223, 228

Л

Линейка 196
◊ измерительная 320
Линия:
◊ горизонтальная 92
◊ траектории 485
Литерал 59
Логин 622

М

Маркер:
◊ выделенного кадра 484
◊ изменения размера 128
◊ плавающего изображения 131
◊ точки останова 569
Массив 504
◊ двумерный 504
◊ индекс 504
◊ ключевых точек траектории 477
◊ одномерный 504
◊ элемент 504
Меню:
◊ гиперссылок 553
◊ главное 35
◊ панели дополнительное 37
◊ поведений 531
◊ серверных поведений 667
◊ событий 530
◊ строки ширин 332
Метатег 580
◊ базовый интернет-адрес 587
◊ ключевые слова 586
Продолжение рубрики см. на с. 728

Метатег *(прод.)*:
◊ описания Web-страницы 585
◊ перезагрузки 589
◊ связи 590
◊ специальный 593
Метод 508
◊ создание 510
◊ кодирования данных 613, 614
◊ передачи данных 615
▫ POST 616
Мини-редактор HTML 49
Модератор 578
Модуль расширения 151
◊ создание 151
Мультимедиа 148
◊ поддержка 149

H

Набор данных 670
◊ динамический 673
▫ ограниченный 674
◊ простой 685
◊ сложный 684
◊ статический 673
▫ однонаправленный 673
◊ тип 673
Набор записей 620
Набор фреймов 268
◊ вложенный 274
◊ внешний 274
◊ внутренний 274
◊ простой 274
◊ сложный 274
Навигатор 676
Настройка 65
◊ русификация 66
Начертание 83
Несанкционированный доступ 711
◊ защита 711
Новости 107

O

Область:
◊ видимая 452
◊ невидимая 356
◊ редактирования 626
Область шаблона:
◊ заголовок 354
◊ изменяемая 348
◊ неизменяемая 348
◊ необязательная 371, 691
◊ необязательная изменяемая 377
◊ повторяющаяся 373, 679
◊ создание 353
◊ табличная повторяющаяся 378
Обработчик события 471, 517
◊ привязка 471, 518
Объект 508
◊ внедренный 18
◊ системный 510
◊ создание 509
◊ удаление 509
Объектная модель документа 513
Окно:
◊ Change Attribute 138
◊ Change Link Sitewide 243
◊ Change Object 494
◊ Choose Spacer Image 330
◊ Convert Layers to Table 464
◊ CSS Style Definition 399
◊ Define Access Levels 712
◊ Design Notes 252
◊ Dynamic List/Menu 682
◊ Dynamic Text 675
◊ Edit Style Sheet 424
◊ Find and Replace 54
◊ Grid Settings 321
◊ Inconsistent Region Names 360
◊ Link External Style Sheet 425
◊ New CSS Style 397
◊ Parameters 153
◊ Rename Timeline 492
◊ Reports 240
◊ Save As Template 349
◊ Select Image Source 127
◊ Synchronize Files 246
◊ Tag Chooser 52
◊ Template Properties 369
◊ Update Files 234
◊ View Data 665
◊ выбора цвета 90
◊ главное 34, 36
◊ документа 35
◊ модальное 54
◊ немодальное 53
◊ правки тега 51
◊ приглашение 35
◊ родительское 35
◊ списка синхронизируемых файлов 247

Операнд 502
Оператор 502
◊ возврата значения 507
◊ завершения цикла 507
◊ инкремента 506
◊ конкатенации 509
◊ объявления 503
◊ переключатель 505
◊ присваивания 502
◊ продолжения цикла 507
◊ слияния строк 509
◊ создания объекта 509
◊ условный 504
◊ цикла 506
Операция файловая 42
Оптимизация 200
◊ по размеру 200
◊ по скорости 200
Ответ серверный 26
Отладка 566, 567
Отступ 80, 195, 409
Ошибка:
◊ 404 242
◊ логическая 566
◊ синтаксическая 565

П

Палитра 90
◊ безопасная 91
◊ режим 91
Панель 35
◊ Behaviors 529
◊ Code Inspector 48
◊ CSS Styles 397
◊ Databases 663
◊ Frames 281
◊ HTML Styles 99
◊ Layers 453
◊ Link Checker 243
◊ Reference 64
◊ Search 58
◊ Site 229
◊ Site Reports 241
◊ Tag Inspector 51
◊ Target Browser Check 120
◊ Timelines 481
◊ группа 36
◊ заголовка 585
◊ заголовок 35
◊ закрытие 38
◊ объектов 38
◊ плавающая 36
◊ развертывание 37
◊ ручка 36
◊ сжатие 37
◊ скрытие 41
Папка:
◊ корневая 217
◊ создание 235
◊ списка избранных 258
◊ файловой системы 216
Параметр:
◊ гиперссылки 688
◊ формы 689
◊ функции 507
Пароль:
◊ пользователя 107
◊ пустой 107
Паук 577
Перевод строки:
◊ жесткий 627
◊ мягкий 627
Переключатель 630
Переменная 502
◊ SQL-запроса 690
◊ Web-сервера встроенная 598
◊ глобальная 503
◊ локальная 507
◊ объявление 503
◊ счетчик 506
Плашка 196
Поведение 525, 617
◊ создание 667
Подсказка по коду 50
Подстрока 54
Поиск записей 719
Поисковая машина 576
◊ классическая 577
Поисковик 577
Поисковый агент 577, 579
Поисковый робот 577
Поле 619
◊ имя 619
◊ ключевое 619
◊ системное 672
◊ счетчика 619
Поле ввода 625
◊ имени файла 635
◊ многострочное 626
◊ пароля 626

Полоса навигации 293
Порт 228
Порядок вложенности тегов 21
Пошаговое выполнение 567
Права посетителя 708
Представление данных 384
Привязка к данным элементов управления 682
Примечание 250
Примитив 125
Приращение 473
Пробел неразрывный 96
Проверка:
◊ HTML-кода 240
◊ гиперссылок 242
Программа:
◊ CGI 609
◊ клиентская 609
◊ серверная 609
Программирование событийно-управляемое 472
Прозвище 258
Просвет 313
Протокол 26
◊ FTP 107
◊ HTTP 26
◊ NNTP 107
◊ TCP/IP 26
◊ высокоуровневый 26
◊ низкоуровневый 26
Псевдостиль гиперссылок 394
Публикация 27
Пункт повторяющейся области 373
Пункт меню:
◊ выключатель 41
◊ переключатель 48
Пункт списка статический 683

P

Рабочая среда 32
◊ выбор 33
◊ старый стиль 33
◊ стиль MX 33
Разграничение доступа 696
Разрыв строк 95
Рамка 412
Раскрутка 576
Растр 124
Расширение Web-сервера 609
Расширения Dreamweaver 722
Регистрация посетителя на сайте 697
Регулярное выражение 57, 58
Редактор свойств 38
Режим:
◊ показа "живых" данных 676
◊ правки стилей 422
◊ применения стилей 422
Режим отображения 46
◊ разметка страницы 319
◊ страницы 47, 48
◊ страницы и кода 47
Режим панели Site:
◊ дерева 230
◊ списка 230
Результат функции 507
Реклама:
◊ активная 575
◊ пассивная 575
◊ черная 584

C

Свойство 508
◊ создание 510
Секция 44
◊ заголовка 181
◊ поддона 181
◊ размера окна 45
◊ размера страницы 45
◊ тегов 44
◊ тела 181
Сервер 26
◊ FTP 107
◊ Web 26
◊ баз данных 621
◊ новостей 107
◊ полномочий 227
◊ почтовый 105
◊ прокси 227
Серверная директива 595
◊ стандартный набор 596
Серверное включение 597
Сессия 697
Сетевой экран 228
Сетка координатная 321
Сжатие 124
◊ без потерь 125
◊ с потерями 124

Символ:
◊ допустимый с точки зрения протокола HTTP 613
◊ код 66
◊ коммерческое эт 105
◊ специальный 95
Синхронизация 246
Скаляр 504
Скобки 503
Скрипт 469
◊ скриптовый язык 470
Скрытое поле 638
Событие 471
◊ внешнее 471
◊ внутреннее 471
Содержимое тега 19
Сортировка 179, 702, 714
◊ по буквенному значению 180
◊ по числовому значению 180
Список 633
◊ гиперссылок 641
◊ маркированный 81
◊ меню 82
◊ нумерованный 81
◊ определений 82
◊ папок 82
◊ стиль 82
◊ тип 82
Справка по Dreamweaver 63
Стиль 98, 385
◊ внедренный 390
◊ внутренний 390
◊ встроенный 390
◊ гибридный 388
◊ конфликт 392
▫ неразрешимый 426
◊ переопределения тега 387
◊ приоритет 394
◊ селектор 388
◊ стилевой класс 386
Столбец текущий 167
Строка:
◊ параметры 173
◊ текущая 166
◊ ширин 327
Строка статуса 44
◊ набора данных 678
Структура:
◊ логическая 213
◊ физическая 216
Сценарий 469
◊ загрузочный 517
◊ серверный 609, 611
◊ язык написания 470

T

Таблица:
◊ выделение 170
◊ заголовок 327
◊ закрытие 620
◊ открытие 620
◊ параметры 174
◊ разметки 309
◊ создание 163
Таблица стилей 385
◊ внедренная 390
◊ внешняя 390
◊ внутренняя 390
◊ времени просмотра 430
◊ времени редактирования 430
◊ временная 430
◊ глобальная 391
◊ импорт 425
◊ привязанная 390
◊ привязка 425, 426
◊ экспорт 427
Таймер системный 476
Тег 18
◊ дочерний 21
◊ закрывающий 19
◊ логического форматирования 19, 85
◊ невидимый 20
◊ неизвестный 157
◊ нестандартный 22
◊ одинарный 20
◊ открывающий 19
◊ парный 20
◊ родительский 21
◊ стандартный 22
◊ физического форматирования 19, 85
Текст:
◊ альтернативный 130
◊ базовая линия 131
◊ вывод 160
◊ динамический 674
◊ поток 130, 306
◊ центральная линия 131
Точка:
◊ начала отсчета 320
Продолжение рубрики см. на с. 732

Точка *(прод.)*:
◊ останова 567
◊ траектории ключевая 478
◊ траектории конечная 474
◊ траектории начальная 474
◊ угловая 138
Траектория 473
◊ длина 473

У

Указатель текущей записи 620
Условие 504, 506
◊ отбора записей 686, 689
◊ связи таблиц 689

Ф

Файл:
◊ включения 597
◊ данных 149
◊ дополнительных данных 157
◊ сирота 243
Фильтр 670, 689, 716
Флажок 629
Форма 612, 622
Формат:
◊ абзаца 78
◊ предопределенный 177
◊ таблиц текстовый 182
◊ фиксированный 78, 160
Фрейм 268
◊ граница 275
◊ заголовок сайта 271
◊ набор гиперссылок 270
◊ невидимый 273
◊ нулевой 273
◊ полезное содержание 269
◊ сведения об авторских правах 272
◊ создание 275
◊ текущий 275
Функция 507
◊ траектории 473, 477

Ц

Цвет:
◊ задание 90
◊ прозрачный 125
◊ селектор 90
◊ схема 114
Цикл:
◊ завершение 507
◊ продолжение 507
◊ проход 506
◊ с условием 506
◊ со счетчиком 506

Ч

Чистка HTML-кода 118

Ш

Шаблон 75, 348, 429
◊ базовый 380
◊ вложенный 380
◊ создание 349
Шкала временная 481
Шрифт:
◊ задание 87
◊ моноширинный 162
◊ пропорциональный 162
◊ стандартный 88

Э

Экспорт 365
Элемент:
◊ активов 255
◊ библиотеки 260
◊ избранный 257
Элемент ActiveX 151, 571
◊ создание 155
Элемент Web-страницы 18
◊ блочный 407
◊ встраивающийся 407
◊ встроенный 407
◊ значок 449
◊ выравнивание 452
◊ компактный 407
◊ поведение 407
◊ позиция на траектории 473
◊ свободно позиционируемый 439
◊ свободный 439
◊ фиксированный 440

Я

Язык программирования:
◊ Perl 611
◊ интерпретируемый 471
◊ классический 468
◊ компилируемый 471
◊ объектно-ориентированный 509
◊ процедурный 509

Ячейка:
◊ заголовка 172
◊ параметры 171
◊ разметки 323
◊ саморастягивающаяся 329
◊ слияние 184
◊ фиксированной ширины 329