bhv®

«Серия "Недостающее руководство" — просто самая разумная и полезная серия руководств».
КЕВИН КЕЛЛИ, СОУЧРЕДИТЕЛЬ ЖУРНАЛА «WIRED»

Создание Web-сайта

недостающее руководство

Книга, которая должна быть на полке

3-е издание с материалами по HTML5



O'REILLY®

Мэтью Мак-Дональд

Creating a Website

the missing manual®

The book that should have been in the box®

Creating a Website

3rd Edition

the missing manual®

The book that should have been in the box®

Matthew MacDonald

O'REILLY®

Beijing | Cambridge | Farnham | Köln | Sebastopol | Tokyo

Мэтью Мак-Дональд

Создание Web-сайта

недостающее руководство

3-е издание

Санкт-Петербург
«БХВ-Петербург»
2013

УДК 004.738.5
ББК 32.973.202
М15

Мак-Дональд М.

М15 Создание Web-сайта. Недостающее руководство. — 3-е изд.: Пер. с англ. — СПб.: БХВ-Петербург, 2013. — 624 с.: ил.

ISBN 978-5-9775-0806-3

Простым и понятным языком написано, как спланировать и создать Web-сайт, разместить его в Интернете. Рассмотрены основы работы с редакторами Web-страниц (Expression Web, Dreamweaver), дан обзор бесплатных редакторов. Подробно разобраны тонкости создания таблиц стилей, добавления графики, связывания страниц, разработки макета страницы. Также рассмотрены страницы с многократно используемыми элементами. В книге уделено пристальное внимание проблеме продвижения сайта: регистрация в поисковиках, каталогах, работа с Google Analytics, сервисом Google Places, Google AdWords, ведение сопровождающего блога в Facebook, Twitter, Google Groups. Описано создание собственного блога на платформе Blogger. Показано, как зарабатывать деньги на своем сайте (Google AdSense, Amazon Associates, PayPal). Описаны элементы украшения Web-сайта (добавка интерактивности с помощью JavaScript, привлекательные кнопки и меню, вставка аудио и видео). В книге имеется краткий справочник по основным HTML-элементам.

Для начинающих Web-разработчиков

УДК 004.738.5
ББК 32.973.202

Группа подготовки издания:

Главный редактор	*Екатерина Кондукова*
Зам. главного редактора	*Евгений Рыбаков*
Зав. редакцией	*Екатерина Капалыгина*
Перевод с английского	*Татьяны Коротяевой*
Компьютерная верстка	*Ольги Сергиенко*
Корректор	*Зинаида Дмитриева*
Оформление обложки	*Марины Дамбиевой*

Authorized translation of the Creating a Website: The Missing Manual by Matthew MacDonald, ISBN: 978-1-449-30172-9, Copyright © 2011 O'Reilly Media, Inc. This translation is published and sold by permission of O'Reilly Media, Inc., the owner of all rights to publish and sell the same.

Авторизованный перевод английской редакции книги: ISBN: 978-1-449-30172-9, Copyright © 2011 O'Reilly Media, Inc. Перевод опубликован и продается с разрешения O'Reilly Media, Inc., собственника всех прав на публикацию и продажу издания.

Подписано в печать 31.01.13.
Формат 70×100 1/16. Печать офсетная. Усл. печ. л. 50,31.
Тираж 1500 экз. Заказ №
"БХВ-Петербург", 191036, Санкт-Петербург, Гончарная ул., 20.

Первая Академическая типография "Наука"
199034, Санкт-Петербург, 9 линия, 12/28

ISBN 978-1-449-30172-9 (англ.)
ISBN 978-5-9775-0806-3 (рус.)

© 2011 O'Reilly Media, Inc.
© Перевод на русский язык "БХВ-Петербург", 2013

Оглавление

Об авторе ... **1**

Благодарности ... **1**

Введение ... **3**
Что потребуется для того, чтобы начать ... 4
Об этой книге ... 5
Macintosh и Windows ... 5
Несколько слов о структуре книги ... 5
Об | этих | разделителях ... 6
О сетевых ресурсах ... 7
Отсутствующий CD ... 7
Обратная связь ... 7
Список ошибок и опечаток ... 8
Информационная рассылка ... 8
Safari® Books Online ... 8

Глава 1. Подготовка к вступлению в Web ... **9**
Введение в World Wide Web ... 9
Web-браузеры ... 10
Web-серверы ... 14
Планирование Web-сайта ... 15
Виды сайтов ... 15
Время жизни вашего сайта ... 17
Приемы хорошего дизайна ... 18
Компоненты Web-сайта ... 20

Глава 2. Создание вашей первой Web-страницы ... **23**
Введение в HTML: язык Всемирной паутины ... 23
Как устроен HTML-файл ... 24
Просмотр HTML-кода реальной Web-страницы ... 26
Создание HTML-файла ... 26
HTML, XHTML и HTML5 ... 29
HTML-теги ... 30
Что такое тег? ... 31
Представление об элементах ... 32
Вложенные элементы ... 33

HTML-документ ... 35
Определение типа документа ... 35
Базовая структура ... 37
Вставка контента ... 39
Структурирование текста ... 40
А где же все картинки? ... 45
10 самых важных элементов (и несколько дополнительных) ... 47
Поиск ошибок на ваших страницах ... 52

Глава 3. Размещение вашей страницы в Web ... 55
Как работает Web-хостинг ... 55
Из чего состоит URL ... 55
Как браузеры анализируют URL ... 58
Имена доменов ... 60
Выбор удачного имени домена ... 60
Поиск имени ... 63
Регистрация вашего имени ... 64
Бесплатные имена доменов ... 68
Получение места в Web ... 69
Определение ваших потребностей ... 71
Выбор вашего хоста ... 74
Бесплатные Web-хосты ... 79
Перенос файлов ... 80
FTP ... 81
Выгрузка файлов с помощью браузера ... 83

Глава 4. Специализированные средства ... 87
Выбор программного средства ... 88
Разновидности редакторов Web-страниц ... 89
Поиск бесплатного редактора Web-страниц ... 90
Профессиональные HTML-редакторы ... 95
Работа в HTML-редакторе ... 96
Запуск ... 97
Множественные режимы отображения ... 97
Создание Web-страницы в режиме отображения исходного кода ... 99
Создание Web-страницы в режиме WYSIWYG ... 100
Управление Web-сайтом ... 103
Задание сайта в Expression Web ... 104
Выгрузка сайта в Expression Web ... 106
Задание сайта в Dreamweaver ... 109
Обновление сайта в Dreamweaver ... 111

Глава 5. Текстовые элементы ... 115
Представление о тексте и Web-пространстве ... 115
Логическая структура вместо физического форматирования ... 116
CSS (каскадные таблицы стилей) ... 118
HTML-элементы для основного текста ... 119
Абзацы ... 121
Переносы строк ... 122
Заголовки ... 124
Горизонтальные линии ... 126
Текст в заданном формате ... 127

Цитаты ... 128
Разделы и оболочки ... 130
HTML-элементы для создания списков ... 132
Нумерованные списки ... 132
Маркированные списки ... 134
Списки определений ... 135
Вложенные списки ... 135
HTML-элементы для таблиц ... 137
Простая таблица ... 137
Охват ячеек ... 139
Внутристрочное форматирование ... 142
Форматирование текста: курсив и жирное начертание ... 142
Несколько дополнительных элементов форматирования ... 143
Специальные символы ... 144
Иностранные языки ... 147

Глава 6. Таблицы стилей ... 151
Основы таблиц стилей ... 152
Три вида стилей ... 152
Анатомия правила ... 154
Присоединение таблицы стилей к странице ... 155
Применение внутренней таблицы стилей ... 159
Применение встроенных стилей ... 159
Каскад ... 161
Наследование ... 161
Более мощные селекторы ... 163
Селекторы классов ... 164
Селекторы идентификаторов ... 165
Цвета ... 166
Задание цвета ... 167
Поиск нужного цвета ... 168
Выравнивание и разгонка текста ... 170
Выравнивание ... 171
Разгонка ... 173
Пробелы ... 174
Основные шрифты ... 175
Задание шрифта ... 177
Поиск подходящего шрифта ... 178
Размеры шрифтов ... 180
Встроенные шрифты ... 184
Web-форматы для шрифтов ... 184
Применение комплектов шрифтов ... 185
Применение ваших собственных шрифтов ... 189
Границы ... 190
Основные границы ... 191
Создание улучшенных границ ... 192
Применение границ в таблицах ... 193
Руководство по созданию таблицы стилей ... 195
Создание хорошо структурированной таблицы стилей ... 195
Сокращение работы с помощью элемента * * ... 199
Сокращение работы с помощью контекстного селектора ... 201

Глава 7. Добавление графики ... 203
Основные сведения об изображениях ... 203
Элемент <*img*> ... 204
Альтернативный текст ... 204
Размер изображения ... 206
Форматы графических файлов ... 208
Размещение картинок на цветном фоне ... 212
Изображения и стили ... 215
Встраивание изображений в текст ... 215
Границы ... 216
Обтекание текста вокруг изображения ... 217
Вставка подписей к изображениям ... 221
Фоновые изображения ... 222
Приемы работы с графикой ... 226
Графический текст ... 227
Фон для других элементов ... 228
Графические маркеры в списке ... 230
Поиск бесплатной графики ... 231

Глава 8. Связывание страниц ... 235
Представление о якоре или точке привязки ... 235
Внутренние и внешние ссылки ... 237
Относительные ссылки и папки ... 239
Изменения цвета ссылок и подчеркивания ... 245
Почтовые ссылки ... 246
Изображения-ссылки и карты изображений ... 249
Вставка закладок ... 252
Когда хорошие ссылки портятся ... 254
Управление Web-сайтом ... 255
Средства проверки ссылок ... 257
Использование перенаправлений ... 260

Глава 9. Макет страницы ... 263
Проблемы экранного пространства ... 263
Макет на основе стилей ... 267
Структурирование страниц с помощью элемента <*div*> ... 267
Плавающие блоки ... 268
Фиксированные блоки ... 269
Создание макета с несколькими колонками ... 271
Построение макета с фиксированной шириной ... 272
Создание макета с изменяемым размером ... 275
Максимальная ширина: подстраховка ... 277
Увеличение высоты колонки ... 279
Сверхгибкие сайты: дзэн Web-дизайна ... 281
Несколько дополнительных приемов верстки ... 283
Применение слоев ... 283
Комбинирование абсолютного и относительного позиционирования ... 285
Задание размеров таблиц ... 287

Глава 10. Страницы с многократно используемыми элементами ... 293
Представление о страницах с многократно используемыми элементами ... 293
Серверные включения ... 295

Шаблоны страниц ... 299
Общее представление о шаблонах страниц ... 299
Создание нового шаблона страницы ... 301
Анатомия шаблона страницы ... 302
Применение шаблона страницы ... 306

Глава 11. Обнародование вашего сайта ... 311
План продвижения вашего Web-сайта ... 311
Сделайте ваш сайт удобным для поисковых машин ... 313
Выбирайте информативные названия страниц ... 314
Вставляйте описание страницы ... 315
Предоставляйте замещающий текст ко всем своим изображениям ... 317
Применяйте информативный текст для ссылок ... 317
Не пытайтесь жульничать ... 318
Регистрация в каталогах ... 319
Каталог Open Directory Project ... 319
Каталог Yahoo ... 322
Регистрация с помощью поисковых машин ... 323
Повышение ранга в результатах поиска ... 325
Как спрятаться от поисковых машин ... 326
Google Webmaster Tools ... 329
Отслеживание посетителей ... 332
Описание Google Analytics ... 333
Регистрация в Google Analytics ... 335
Просмотр вашего Web-трафика ... 337

Глава 12. Продвижение Web-сайта ... 345
Рассказать всем ... 346
Беззастенчивая самореклама ... 346
Google Places ... 347
Google AdWords ... 349
Вернувшиеся посетители ... 351
Превращение сайта в сообщество ... 353
Взращивание Web-сообщества ... 353
Инструменты Web-сообществ ... 354
Электронные информационные рассылки ... 356
Twitter ... 358
Группы ... 361
Facebook ... 371

Глава 13. Блоги ... 383
Общее представление о блогах ... 383
Популярные блоги ... 386
Агрегирование ... 387
Хостинг блога и программное обеспечение ... 390
Начало работы с Blogger ... 392
Создание блога ... 392
Создание форматированных сообщений ... 398
Управление блогом ... 400
Настройка общих параметров ... 403
Настройка вашего профиля в Blogger ... 406
Присвоение вашему блогу имени пользовательского домена ... 407

Настройка вашего шаблона ... 409
Управление комментариями ... 416

Глава 14. Как зарабатывать деньги на своем сайте ... 421
Зарабатывание денег в Интернете ... 422
Google AdSense ... 423
Регистрация в AdSense ... 426
Интерфейс AdSense ... 428
Создание рекламного объявления ... 430
Размещение рекламных объявлений на ваших Web-страницах ... 436
Улучшенный поиск Google ... 439
Amazon Associates ... 444
Регистрация в качестве партнера ... 445
Генерация партнерских ссылок ... 446
Услуги для торговцев системы PayPal ... 452
Регистрация в PayPal ... 454
Прием платежей ... 455
Создание корзины для покупок ... 461
Вывод ваших денег ... 466

Глава 15. JavaScript: добавка интерактивности ... 467
Основные сведения о JavaScript ... 467
Серверное и клиентское программирование ... 468
Языки сценариев ... 469
JavaScript 101 ... 471
Переменные ... 475
Функции ... 479
Внешние файлы сценариев ... 483
Dynamic HTML ... 485
HTML-объекты ... 485
События ... 491
Графические ролловеры ... 494
Сворачивающийся текст ... 496
Интерактивные формы ... 500
Сценарии в Web-пространстве ... 505
Поиск простого сценария ... 506
Библиотеки на JavaScript ... 510
Web-виджеты ... 510

Глава 16. Привлекательные кнопки и меню ... 513
Причудливые кнопки ... 513
Создание изображений ваших кнопок ... 515
Создание эффекта ролловера с помощью CSS ... 517
Кнопки с изображением и текстом ... 520
Кнопки без изображений ... 522
Привлекательные меню ... 523
Сворачивающиеся меню, сделанные самостоятельно ... 524
Чужие меню ... 529

Глава 17. Аудио и видео ... 535
Представление о мультимедиа ... 536
Связывание, встраивание и хостинг ... 536
Типы файлов мультимедиа ... 538

Стандартная фоновая музыка ... 540
Элемент <*embed*> ... 541
Параметры встроенного звука ... 542
Flash-плейеры MP3 ... 544
Плейер Premiumbeat: воспроизведение списка мелодий ... 544
Плейер Yahoo: воспроизведение всех ваших ссылок ... 547
Лупы формата Flashtrak ... 549
Видеоклипы ... 552
Подготовка видеофайлов ... 552
Связанное и встроенное видео ... 554
Flash-видео ... 555
Видео в HTML5 ... 556
Выгрузка вашего видео на YouTube ... 557
Регистрация на YouTube ... 558
Подготовительный этап ... 559
Выгрузка видео ... 560
Просмотр видео ... 564

Приложение 1. Краткий справочник по HTML ... 567
HTML-элементы ... 567
<*a*> (элемент-якорь) ... 568
<*address*> ... 569
<*area*> (карта изображения) ... 569
<*b*> (жирный шрифт) ... 570
<*base*> (базовый URL) ... 570
<*blockquote*> (блочная цитата) ... 570
<*body*> (тело документа) ... 571
<*br*> (перенос строки) ... 571
<*button*> (кнопка) ... 571
<*caption*> (заголовок таблицы) ... 572
<*cite*> (опубликованный источник) ... 572
<*code*> ... 572
<*dd*> (словарное определение) ... 573
<*del*> (удаленный текст) ... 573
<*dfn*> (определяемый термин) ... 573
<*div*> (типовой блочный контейнер) ... 573
<*dl*> (словарный список) ... 574
<*dt*> (словарный термин) ... 574
<*em*> (акцент) ... 574
<*embed*> (встроенный объект) ... 574
<*form*> (интерактивная форма) ... 575
<*h1*>, <*h2*>, <*h3*>, <*h4*>, <*h5*>, <*h6*> (заголовки) ... 575
<*head*> (шапка документа) ... 575
<*hr*> (горизонтальная линия) ... 576
<*html*> (документ) ... 576
<*i*> (наклонный шрифт) ... 576
<*iframe*> (внутристрочный фрейм) ... 577
<*img*> (изображение) ... 577
<*input*> (элемент ввода) ... 578
<*ins*> (вставленный текст) ... 579
<*li*> (пункт списка) ... 579
<*link*> (взаимосвязи документа) ... 580

<*map*> (карта изображения) ... 580
<*meta*> (метаданные) ... 581
<*noscript*> (замещающий контент сценария) ... 581
<*object*> (встроенный объект) ... 581
<*ol*> (нумерованный список) ... 581
<*option*> (пункт меню) ... 582
<*p*> (абзац) ... 582
<*param*> (параметр объекта) ... 583
<*pre*> (предварительно отформатированный текст) ... 583
<*q*> (короткая цитата) ... 583
<*samp*> (стандартный вывод) ... 584
<*script*> (клиентский сценарий) ... 584
<*select*> (список выбора) ... 584
<*small*> (мелким шрифтом) ... 585
<*span*> (типовой строчный контейнер) ... 585
<*strong*> (особая важность) ... 585
<*style*> (внутренняя таблица стилей) ... 585
<*sub*> (нижний индекс) ... 586
<*sup*> (верхний индекс) ... 586
<*table*> (таблица) ... 586
<*td*> (ячейка данных в таблице) ... 587
<*textarea*> (многострочное текстовое поле ввода) ... 587
<*th*> (ячейка шапки таблицы) ... 587
<*title*> (название документа) ... 588
<*tr*> (строка таблицы) ... 588
<*ul*> (маркированный список) ... 588
Семантические элементы HTML5 ... 588
<*article*> (статья) ... 590
<*aside*> (боковая панель) ... 591
<*figcaption*> (подрисуночная надпись) ... 591
<*figure*> (рисунок) ... 592
<*footer*> (нижний колонтитул или подвал) ... 592
<*header*> (верхний колонтитул) ... 592
<*hgroup*> (группа верхнего колонтитула) ... 593
<*mark*> (выделенный текст) ... 593
<*nav*> (навигационные ссылки) ... 593
<*section*> (секция) ... 594
<*time*> (дата или время) ... 594
Символьные HTML-последовательности ... 595
Названия цветов в HTML ... 597

Приложение 2. Полезные Web-сайты ... 598
Ссылки по главам ... 598

Предметный указатель ... 607

Об авторе



Мэтью Мак-Дональд пишет о науке и технологиях. Он — автор более дюжины книг. Фанатики электронных таблиц могут выполнять математические вычисления вместе с ним в книге Excel 2010: "The Missing Manual" (Excel 2010: недостающее руководство). Страстные любители данных могут последовать за автором в головокружительный мир баз данных с книгой "Access 2010: The Missing Manual" (Access 2010: недостающее руководство). И человеческие существа любого толка могут узнать о том, насколько они действительно необычны из книг "Your Brain: The Missing Manual" (Ваш мозг: недостающее руководство) и "Your Body: The Missing Manual" (Ваше тело: недостающее руководство).

Благодарности

Ни один автор не сможет завершить книгу без небольшой армии помощников. Я глубоко признателен всей команде "Missing Manual", особенно моему редактору Питеру Макки, не дававшему мне сойти с дистанции относительно нежными подталкиваниями, и гуру HTML, техническому рецензенту Шелли Пауерс, одолжившей свое глубокое понимание всего, что связано с Web-технологиями. Я также сердечно благодарен тем, кто оставил свои замечания о предыдущих изданиях книги, включая Сару Мильштейн (Sarah Milstein), Питера Мейерса (Peter Meyers) и технических рецензентов Джима Гудинафа (Jim Goodenough), Рею Ховард (Rhea Howard), Марка Левита (Mark Levitt), Тони Раско (Tony Ruscoe) и Меган Соренсен (Megan Sorensen). Как всегда, я глубоко признателен многочисленным помощникам, кто трудился за кулисами, составляя указатель, готовя рисунки и проверяя ошибки в окончательном варианте.

В заключение хочу сказать, что я не написал бы ни одной книги без поддержки моих родителей Норы и Пола, родителей жены Рази и Хамида и моей жены Фарии. (Я бы, наверное, написал гораздо больше, если бы не проблемы, связанные с моими милыми дочками, Майей и Бренной.) Спасибо всем!

Мэтью Мак-Дональд

Введение

В наши дни невозможно найти человека, ничего не слышавшего об Интернете. Компании создают Web-сайты до того, как разработают бизнес-планы. Политические деятели игнорируют дебаты и разносят своих оппонентов в сети. Даже бабушки, еще недавно опасавшиеся всякого рода технических новшеств, тратят часы на общение со старыми друзьями по электронной почте (и продавая причудливые семейные реликвии на eBay). Интернет даже изменил наш язык: например, появились новые глаголы гуглить и зафрендить, а твит не имеет никакого отношения к птицам (по-английски tweet — птичий щебет).

Как вы без сомнения знаете, установить свое присутствие в Web можно разными способами. Вы можете болтать с друзьями на странице Facebook, делиться своими фотографиями с близкими по духу фотографами на Flickr, выложить свое домашнее видео на YouTube или писать короткие дневниковые заметки в блоге, хранящемся на сервисе, таком как Blogger. Но если вы достаточно амбициозны и выбрали эту книгу, вы нацелились на золотой стандарт Web: свой собственный настоящий Web-сайт.

Что же вы сможете делать такого с помощью Web-сайта, чего нельзя добиться средствами электронной почты, социальных сетей и других Web-сервисов? Кратко: *все что угодно*.

Ваш персональный Web-сайт — просто постоянное место хранения вашего резюме или центр электронной оптовой торговли именной одеждой? (Так появился не один миллионер.) Главное в том, что ваш собственный Web-сайт дает вам власть решать, каким ему быть и возможность изменять все по собственному желанию. Если вы уже пользуетесь какими-либо Web-сервисами, например YouTube и Facebook, как вы узнаете из этой книги, их можно сделать частью своего Web-сайта. Почему не поместить YouTube-видео вашего кота, играющего в бильярд, прямо внутрь Web-сайта, рядом с вашими уникальными товарами для кошек?

Конечно, вместе с большой властью приходит и серьезная ответственность, т. е. если вы решили создать собственный сайт, именно вам придется добиваться того, чтобы он выглядел не так фальшиво и примитивно, как портреты в календарях 1960 гг. Вот тут и пригодится эта книга. Имея ее под рукой, вы узнаете следующее.

- *Как создавать Web-страницы.* HTML (HyperText Markup Language, Язык гипертекстовой разметки) — это язык Web-пространства. За последнее десятилетие

модернизированная версия HTML, названная XHTML (Extensible HyperText Markup Language, Расширяемый язык гипертекстовой разметки) постепенно вытеснила HTML и сейчас объединена с еще одной новой версией, названной HTML5. В этой книге вы познакомитесь с названными стандартами и узнаете, как писать самые современные, надежные Web-страницы.

- *Как сделать страницы привлекательными с помощью CSS (Cascading Style Sheets, Каскадные таблицы стилей).* Технология CSS начинается там, где заканчивается HTML, добавляя средства форматирования, способные превратить неряшливейшие из сайтов в семейство согласованных страниц, которые кажутся спроектированными профессионально. И лучше всего то, что, когда вы поймете, как правильно применять CSS, вы сможете изменить внешний вид всего сайта, откорректировав единственный файл.
- *Как поместить ваш Web-сайт во Всемирную паутину.* Величайший Web-сайт в мире не слишком полезен, если его никто не видит. Именно поэтому вы узнаете, как найти самую лучшую компанию Web-хостинга, выбрать подходящее *имя домена* (например, **www.HotToTrotHorses.com**) и поместить ваш шедевр в Интернет. Не беспокойтесь — огромное множество компаний, предлагающих дешевый Web-хостинг, готовы показывать ваш сайт за несколько центов в день.
- *Как привлечь посетителей.* Вы узнаете, как убедиться в том, что ваш сайт можно найти с помощью популярных поисковых машин, и как создать сетевое сообщество, поддерживающее повторные визиты.
- *Как разбогатеть (или по крайней мере заработать немного разменных денег).* Web-пространство — двигатель электронной коммерции, но даже обычные люди могут зарабатывать, распространяя свои любимые товары (через Amazon), продавая собственные изделия (с помощью платежной системы, такой как PayPal) или отображая рекламные объявления (с помощью Google). Вы узнаете, как начать действовать.
- *Как добавить излишеств.* У любого Web-сайта с изюминкой есть собственные хитрости. Вы узнаете, как ослепить посетителей причудливыми кнопками, отшлифованными меню и другими блестящими элементами, предупредительностью JavaScript. Вы даже узнаете, как с помощью фоновой музыки развлечь посетителей.

Что потребуется для того, чтобы начать

Эта книга создавалась в расчете на то, что у вас есть относительно современный компьютер и честолюбивые замыслы. Существуют десятки хорошо оснащенных программ Web-редактирования, способных помочь в построении Web-сайта, но для этой книги вам не понадобится ни одна из них. Если воспользоваться Web-редактором до того, как вы разберетесь в принципах работы Web-сайтов, вы создадите себе проблем больше, чем сможете решить за это время. Из-за своей готовности помочь эти программы уведут вас в сторону от изучения принципов хорошего

дизайна сайтов, которые демонстрируют разницу между привлекательным, легким в сопровождении Web-творением и кошмаром неорганизованного дизайна.

Освоив азы, вы сможете воспользоваться искусным редактором Web-страниц, таким как Microsoft Expression Web или Adobe Dreamweaver. В этой книге приводится обзор принципов работы этих программ и описание нескольких бесплатных альтернатив (см. *главу 4*).

Примечание

Ни при каких обстоятельствах вам не потребуется знание сложных технологий Web-программирования, например, Java или ASP.NET. Вам также ничего не нужно знать о базах данных или языке XML. Эти темы очень увлекательны, но их безумно трудно освоить без солидного опыта программирования. В этой книге вы узнаете, как создать наилучший Web-сайт, не становясь программистом. (Тем не менее вы узнаете о языке JavaScript достаточно для того, чтобы применять многие из бесплатных образцов сценариев, которые можно найти в сети.)

Об этой книге

У Web-пространства нет владельца. В результате никто не отвечает за обучение принципам его использования или приемам построения собственного места обитания в сети. Вот тут данная книга и приходит на помощь. Если бы у Web-пространства было справочное руководство, подробно и тщательно описывающее основные компоненты, экономящие время приемы и впечатляющие дополнения, необходимые каждому сайту, — им могла бы стать эта книга.

Macintosh и Windows

Одна из замечательнейших характеристик World Wide Web (Всемирная паутина) — ее подлинная всемирность. Где бы вы ни жили, от Арубы до Замбии, Web с нетерпением ждет вашего участия. Это же распространяется и на компьютер, который вы используете для создания вашего сайта. Вы можете реализовать тактику, средства и приемы, описанные в этой книге, почти на любом имеющемся у вас типе компьютера, начиная с ранних моделей Windows PC и заканчивая самыми последними и наилучшими компьютерами Mac. (Конечно, некоторые программы предпочитают ту или иную операционную систему, но вы узнаете об особенностях, как только они возникнут.) Самое замечательное в том, что эта книга подходит и может быть полезна для компьютеров всех мастей.

Несколько слов о структуре книги

Эта книга разделена на пять частей, каждая из которых включает несколько глав.

1. *Приглашение в Web.* В этой части вы начнете планировать ваш Web-сайт *(глава 1)*. Вы изучите основы HTML, языка Web-пространства *(глава 2)*; поместите свои страницы в сеть с помощью уважаемой компании, предоставляющей Web-

хостинг *(глава 3)*. И наконец, узнаете, как облегчить жизнь, применив программы редактирования Web-страниц *(глава 4)*.

2. *Построение улучшенных Web-страниц.* В этой части показано, как вставлять на ваши страницы основные компоненты. Сначала вы узнаете, как пользоваться CSS-стандартом, позволяющим задать привлекательные цвета, шрифты и границы *(глава 6)*. Далее вы вставите изображения на ваши страницы *(глава 7)* и создадите полностью Web-сайт из связанных страниц *(глава 8)*. В заключение вы овладеете несколькими тщательно разработанными макетами *(глава 9)* и узнаете, как распространить их на весь ваш сайт *(глава 10)*.
3. *Связь с вашей аудиторией.* В третьей части книги поясняется, как заставить поисковые машины, например Google, заметить ваш сайт *(глава 11)* и как увеличить Web-трафик *(глава 12)*. Вы также познакомитесь с блогами (сокращение от web log — Web-журнал) и бесплатным программным обеспечением, которое помогает их создавать *(глава 13)*. И наконец, вы узнаете, как встать на путь Web-обогащения, отображая рекламные объявления и продавая собственные товары *(глава 14)*.
4. *Элементы украшения Web-сайта.* Теперь, когда вы можете создать профессиональный работающий Web-сайт, почему не украсить его привлекательными элементами, такими как ярко светящиеся кнопки или всплывающие меню? Вы не найдете головоломных подробностей о том, как стать завзятым программистом на JavaScript, но узнаете достаточно, чтобы применять на своих собственных страницах бесплатные мини-программы на JavaScript для выполнения основных задач (*главы 15* и *16*). Вы также поиграете с видеоклипами и вставите музыкальный MP3-плейер на обычную Web-страницу *(глава 17)*.
5. *Приложения.* В конце книги вы найдете два приложения. Первое предлагает краткий справочник по языку HTML. В нем описаны основные HTML-элементы и приводятся ссылки на конкретную главу книги с более подробными обсуждениями. Второе приложение перечисляет полезные ссылки, которые помогут вам узнать больше, получить бесплатные материалы и зарегистрироваться на полезных сервисах. Не волнуйтесь, вам не придется вводить все указанные Web-ссылки вручную в окне вашего браузера. Все уже готово для вас на странице **Missing CD** (Отсутствующий CD) по адресу **www.missingmanuals.com/cds/caw3**.

Об | этих | разделителях

На протяжении всей книги вам будут встречаться следующие предложения: "Для удаления элементов управления форматированием Word выберите **Edit | Clear | Clear Formatting**". Это не что иное, как сокращенная форма записи более длинной инструкции, требующей последовательного открытия трех меню, например: "Откройте меню **Edit**, щелкните кнопкой мыши команду **Clear** для открытия второго меню. В этом меню для завершения процесса щелкните кнопкой мыши команду **Clear Formatting**. На рис. В.1 показаны подробности.

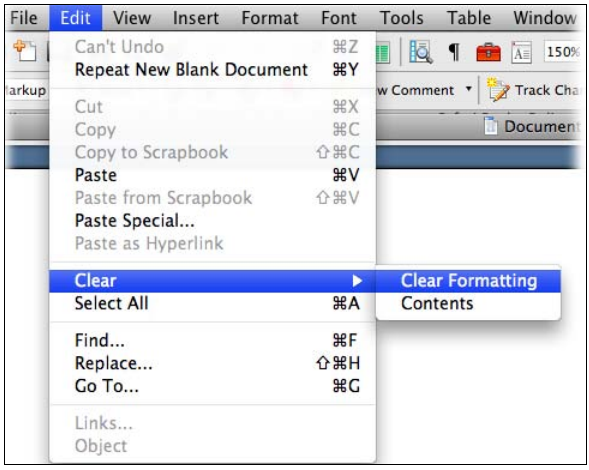


Рис. В.1. В этой книге записи с разделителями помогают упростить инструкции переходов в меню и папках. Например "Выберите **Edit | Clear | Clear Formatting**" — это более компактный способ записи инструкции: "Из меню **Edit** выберите команду **Clear**; из появившегося подменю выберите команду **Clear Formatting**"

О сетевых ресурсах

Купив данное руководство, вы получаете больше, чем просто книгу для чтения. В сети вы найдете файлы примеров, а также советы, статьи, видео, файлы. Вы также можете связаться с издательской командой и сообщить нам о том, что вам понравилось (или не понравилось) в книге. Просто зайдите на сайты **www.missingmanuals.com** и **www.bhv.ru**.

Отсутствующий CD

У книги нет сопроводительного компакт-диска, приклеенного к внутренней стороне спинки обложки, но вы ничего не упускаете. Перейдите на страницу **www.missingmanuals.com/cds/caw3** и загрузите примеры дизайна Web-страниц, упомянутые в книге, и дополнительную информацию. Таким образом вы не сотрете пальцы, набирая длинные Web-адреса, страница **Missing CD** предлагает перечень реагирующих на щелчок мыши ссылок на Web-сайты, упомянутые в книге.

> ***Совет***
>
> Если ищете конкретный пример, далее описано, как его быстро найти: посмотрите на соответствующий рисунок в книге. Имя файла обычно видно в конце текста, набранного в строке адреса. Например, если вы видите URL D:\Code\Creating a Website\Chapter 2\popsicles.htm (см. рис. 2.2), имейте в виду, что соответствующий файл с примером — popsicles.htm.

Обратная связь

Возникли вопросы? Нужна дополнительная информация? Видите себя в роли рецензента книг? На нашей странице **Feedback** (Обратная связь) вы сможете полу-

чить ответы специалистов на вопросы, появившиеся у вас во время чтения книги, поделиться своими мыслями о руководстве и найти группы людей, разделяющих ваш интерес к созданию собственных Web-сайтов. Чтобы сказать свое слово, перейдите на страницу **www.missingmanuals.com/feedback** или страницу русской книги на сайте **www.bhv.ru**.

Список ошибок и опечаток

Для того чтобы книга оставалась актуальной и корректной, насколько возможно, при каждой допечатке тиража мы вносим все проверенные корректировки, предложенные вами. Мы также сообщаем об этих изменениях на Web-сайте, посвященном английской версии книги, поэтому вы можете, если захотите, внести важные изменения в собственный экземпляр книги. Перейдите на страницу **http://tinyurl.com/4ulhb8l**, чтобы сообщить об ошибке и просмотреть имеющиеся исправления.

Информационная рассылка

Наша бесплатная информационная рассылка по электронной почте держит вас в курсе того, что происходит на территории серии Missing Manual. Вы сможете встретиться с авторами и редакторами, в качестве бонуса посмотреть видео и выдержки из книг и т. д. Для регистрации перейдите на страницу **http://tinyurl.com/MMnewsletter**.

Safari® Books Online

Safari® Books Online — цифровая библиотека по требованию, позволяющая легко находить быстрые ответы на свои вопросы в более чем 7 500 справочных книгах и видео о современных технологиях.

Благодаря подписке вы сможете прочесть любую страницу и посмотреть любое видео из нашей сетевой библиотеки. Читайте книги на вашем мобильном телефоне и других мобильных устройствах. Обращайтесь к новым изданиям до их выхода из печати, получайте исключительный доступ к рабочим рукописям и публикуйте сообщения для авторов. Копируйте и вставляйте примеры программного кода, создавайте группы Избранное, загружайте главы, делайте закладки на ключевые разделы, создавайте пометки, печатайте страницы и получайте выгоду от множества средств, сберегающих ваше время.

Издательство O'Reilly Media выложило эту книгу на английском языке на сервисе Safari® Books Online. Для получения электронного доступа к этой книге и другим книгам с похожей тематикой издательства O'Reilly и других издателей оформите бесплатную регистрацию на сайте **http://my.safaribooksonline.com**.

ГЛАВА 1

Подготовка к вступлению в Web

Всемирная паутина (или Web) — замечательное место. Здесь каждый день обрабатываются миллионы финансовых транзакций, публикуются самые свежие новости и сплетни о знаменитостях, предоставляются места для собраний самых разных сообществ, начиная с политических анархистов и заканчивая поклонниками вампирских шоу.

Поскольку вы читаете эту книгу, вы, вероятно, решили приобщиться к Web. Поздравляю! Но так же как в реальной жизни, прежде чем подыскать жилье, вы разрабатываете предварительный план, для перехода в интерактивное окружение необходимо выполнить некоторые подготовительные действия. В этой главе вы получите представление о том, как устроена Всемирная паутина или Web и узнаете, какие вам нужны компоненты для создания собственного Web-сайта.

Введение в World Wide Web

Несмотря на то что сеть Интернет скрывает свой возраст, она не так молода, как вы думаете. Компьютерные гении, создавшие ее, начали разрабатывать основные концепции в начале 1960-х гг. Первая передача данных по Интернету между компьютером Калифорнийского университета в Лос-Анджелесе и одним из компьютеров Стэнфордского исследовательского института состоялась в 1969 г. В первые моменты похвастаться особенно было нечем — компьютер завис, как только дошел до буквы G в слове "LOGIN". И все же революция свершилась.

В раннем Интернете в основном путешествовали представители научных и правительственных кругов. Сеть действовала как средство для научных исследований и сотрудничества, позволяя ученым в разных точках земного шара обмениваться информацией. В 1993 г. появился первый Web-браузер. В последующие годы новые пользователи заселили Интернет, включая продавцов книг, любителей новостей, обладателей хобби и множество программистов-одиночек.

ПРИМЕЧАНИЕ

Любители истории могут познакомиться с подробным описанием раннего Интернета на Web-страницах **www.isoc.org/internet/history** и **www.walthowe.com/navnet/history.html**.

Конечно, ранний Интернет мало походил на современный. В 1969 г. интернет-сообщество состояло из четырех компьютеров и все это были адские машины, о которых могли мечтать (или которые могли себе позволить) только государственная лаборатория или научное учреждение. В 1981 г. Интернет населяли менее 200 больших вычислительных машин и использовали их в основном компьютерные специалисты или научные сотрудники для своей повседневной работы. Сегодня во Всемирной сети гораздо более 100 миллионов Web-сайтов и еще больше Web-энтузиастов. Неудивительно, что вы получаете так много мусора по электронной почте.

ЧАСТО ЗАДАВАЕМЫЙ ВОПРОС

Отличие Web от Интернета

Есть ли разница между Всемирной паутиной (Web) и Интернетом?

Дикторы информационных программ, политики и обычные люди часто используют эти термины как взаимозаменяемые. Строго говоря, это разные вещи — и путать их все равно, что ставить на одну доску высококлассных компьютерных специалистов и компьютерных фанатов с высокой самооценкой.

Интернет — это сеть соединенных между собой компьютеров, покрывающая земной шар. Эти компьютеры связывают друг с другом для совместного использования информации, но делать это можно разными способами, включая электронную почту, обмен мгновенными сообщениями, передачу файлов по FTP-протоколу (File Transfer Protocol) и загрузку пиратских копий голливудских блокбастеров с помощью пиринговых программ (чего вы, конечно же, не делаете).

Всемирная паутина (*World Wide Web)* — один из множества способов обмена информацией в Интернете. И как люди получают доступ к этой информации? Для посещения Web-сайтов и Web-страниц, разбросанных по всему миру, как вы уже догадались, они применяют специальные программы, называемые *Web-браузерами*.

Web-браузеры

Web-браузер, как вы без сомнения знаете, — это программное обеспечение, позволяющее перемещаться по Web-страницам и отображать их на экране. Без браузеров Всемирная паутина все равно бы существовала, но вы не могли бы включить компьютер и заглянуть в нее.

Работа браузера удивительно проста: на самом деле большая ее часть состоит из выполнения двух задач. Во-первых, браузер запрашивает Web-страницы, когда вы набираете адрес Web-сайта (например, **www.google.com**) или щелкаете кнопкой мыши ссылку на Web-странице. Браузер посылает этот запрос на удаленный компьютер, именуемый *Web-сервером*. Сервер, как правило, гораздо мощнее домашнего компьютера, поскольку он должен обрабатывать одновременно запросы многочисленных браузеров. Сервер внимательно отслеживает эти запросы и посылает обратно содержимое нужных Web-страниц.

Когда браузер получает это содержимое, он запускает вторую задачу и визуализирует или прорисовывает Web-страницу. Технически это означает, что браузер преобразует обычный текст, полученный с сервера, в экранный документ, основанный на инструкциях форматирования, встроенных в текстовую страницу. Конечный



Рис. 1.1. Web-браузер создан для того, чтобы по-настоящему хорошо делать две вещи: связываться с удаленными компьютерами для запрашивания Web-страниц и затем отображать эти страницы на вашем компьютере

результат — графически привлекательно оформленная страница с разными шрифтами, цветами и ссылками. Процесс показан на рис. 1.1.

Выбор вашего браузера

Выбор Web-браузера — это в зависимости от особенностей вашей личности а) нудное занятие или б) важное дело, отражающее ваш характер, индивидуальность и в целом компьютерную грамотность. Если вы попадаете в последнюю категорию, то возможно уже остановили свой выбор на любимом браузере. Но если вы ищете в браузере другие возможности или хотите узнать, какие еще есть браузеры, загляните в приведенный далее обзор, в нем перечислены возможные варианты ваших браузеров.

Даже если вы не собираетесь менять браузер, неплохо познакомиться с самыми популярными из них (рис. 1.2). Это важно, потому что, создавая свой Web-сайт, вы должны рассчитывать на широкую аудиторию, пользующуюся разными браузерами. Для того чтобы ваши стильные странички при просмотре их другими людьми не превратились в неопрятные и примитивные, неплохо протестировать собственный сайт на других компьютерах с разными размерами экранов и другими Web-браузерами.

> ***Совет***
>
> У всех разработчиков Web-сайтов должны быть как минимум экземпляры Internet Explorer, Firefox и Google Chrome, вне всяких сомнений самых популярных сегодня браузеров. Применяя их для проверки созданных вами Web-страниц, вы сможете увидеть, как будет выглядеть ваша напряженная работа для 90 процентов пользователей в мире.

В следующем списке перечислены самые популярные на сегодня браузеры.

- Internet Explorer — все еще самый распространенный браузер в мире. Плохо ли, хорошо ли, но он устанавливает стандарты, которым все остальные браузеры должны следовать. Явное преимущество от применения Internet Explorer (сокращенно IE) заключается в том, что вы с вероятностью примерно 60% не набредете на Web-страницу, которую не сможете прочесть. IE просто настолько популярен, что разработчики сайтов не могут его игнорировать. Недостаток со-

стоит в том, что корпорация Microsoft очень медленно и с большой осторожностью обновляет IE, поэтому часто новые средства сначала появляются в других браузерах и включаются в IE много месяцев или лет спустя. Загрузить обновленную версию Internet Explorer можно с Web-страницы **www.microsoft.com/ie**.

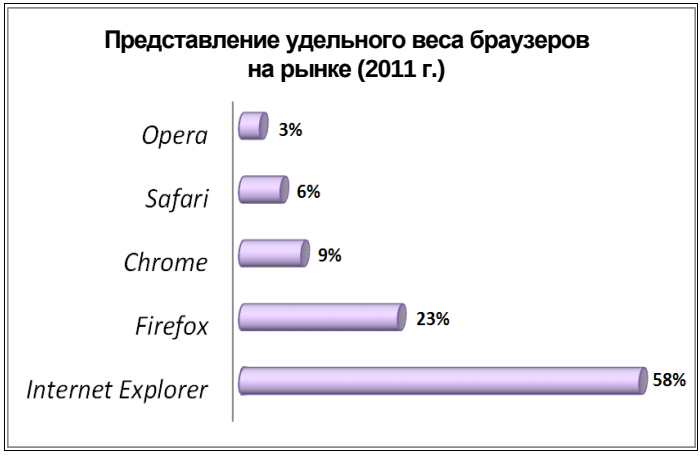


Рис. 1.2. Статистический показатель использования браузеров, вычисляемый как процент людей, использующих каждый из основных браузеров, меняется в зависимости от того, какие сайты вы проверяете и как вы подсчитываете количество посетителей, но во время написания книги это была единственная приемлемая оценка. Столь же важны тенденции применения браузеров, показывающие, что популярность Firefox, Chrome и Safari устойчиво растет, а Internet Explorer падает. (Текущий показатель использования браузеров можно посмотреть на Web-сайте **http://en.wikipedia.org/wiki/Usage_share_of_web_browsers**.)

Примечание

Несмотря на то что Microsoft в течение многих лет экспериментировала с Mac-версией Internet Explorer, они забросили эту работу, когда корпорация Apple создала встроенный браузер Safari. Поэтому если вы обладатель компьютера Mac, придется одолжить у друга на время компьютер PC с ОС Windows, чтобы посмотреть, как выглядят ваши страницы в Internet Explorer. На ваших собственных компьютерах вы возможно пользуетесь самыми популярными Mac-браузерами Safari или Firefox.

Расширенный список браузеров для компьютеров Mac, включающий браузеры, работающие с устаревшими операционными системами, можно найти на Web-странице **www.knutson.de/mac/www/browsers.html**.

- Firefox появился как современный ответ Internet Explorer. Это небольшой, надежный и более чем просто современный Web-браузер. Некоторые инновационные средства, включая многовкладочный просмотр (вы можете просматривать несколько Web-страниц на разных оконных "вкладках"), и блокировка всплывающих окон (для запрета таких надоедливых всплывающих рекламных объявлений) в нем появились гораздо раньше, чем в Internet Explorer. Firefox до сих пор лучше других браузеров подходит для игр благодаря своим невероятно гибким дополнениям (add-on), маленьким программкам, разработанным разными людьми для включения в Firefox дополнительных возможностей, таких как уведомитель электронной почты (Web-mail notifier) и свернутые значки сайтов

(thumbnails), отображаемые на странице результатов поиска Google. В настоящее время Firefox — культовый браузер в среде компьютерных профи и энтузиастов и растущего числа разочарованных старожилов Internet Explorer. Самое замечательное явление — армия добровольных программистов, поддерживающих Firefox на пике современности. Копию Firefox можно загрузить с Web-страницы **www.mozilla.org/firefox.**

- Google Chrome — новое детище в этом блоке. Несмотря на то что он никак не заявлял о себе до конца 2008 г., этот браузер уже занимает по популярности третье место на программном рынке и распространяется по планете с самой высокой скоростью. Несмотря на то что его пользовательский интерфейс немного грубоват, продвинутым Web-фанатам нравится, как стремительно в нем запускается JavaScript (фрагменты программного кода, поддерживающие интерактивные Web-страницы). Опробовать Google Chrome можно на Web-странице **google.com/chrome.**
- Safari — браузер, разработанный корпорацией Apple и входящий в состав операционной системы Mac OS X. Браузер Safari используется в таких продуктах Apple, как iPhone, iPad и iPod Touch. Он быстр, элегантен и содержит ряд полезных средств, например проверку орфографии при заполнении интерактивных форм. Хотя первоначально Apple разрабатывала Safari исключительно для компьютеров Mac, сейчас вы можете загрузить и вариант браузера для ОС Windows. Тем не менее Safari все еще гораздо популярнее среди пользователей Mac. Продолжить знакомство с Safari можно на Web-странице **www.apple.com/safari.**
- Opera — легко устанавливаемый браузер малого размера, который был популярен более десяти лет как противоядие против Internet Explorer с его раздутым размером и бесконечными излишествами. Рост популярности Opera многие годы сдерживал один недостаток — за версию без рекламы нужно было платить. Сегодня Opera, как и другие браузеры в данном перечне, распространяется свободно и версия, свободная от рекламы, тоже. У браузера есть небольшой круг верных ему поклонников и по популярности он занимает скромное пятое место среди других Web-браузеров. Найти Opera можно на Web-сайте **www.opera.com.**

ДЛЯ ТЕХ, КТО ПОНИМАЕТ

Невероятный спад популярности IE

Internet Explorer все еще лидирует в мире с большим преимуществом. Но в среде Web-фанатов его популярность неуклонно падает.

Когда вышло из печати первое английское издание этой книги (в 2005 г.), Internet Explorer удерживал 95% рынка браузеров. Разработчики Web-страниц редко задумывались о каких-либо других браузерах. Три года спустя, когда на прилавках появилось второе издание книги, доля IE на рынке уменьшилась до все еще внушительных 80%. Сегодня у Internet Explorer менее 60% Web-аудитории и все свидетельствует о том, что этот спад будет продолжаться.

Интересно проследить изменения долей рынка разных браузеров в зависимости от страны Web-путешественников и типов Web-сайтов. Например, в Германии Firefox предпочитают 60% Web-путешественников. В Белоруссии побеждает Opera с 49% аудитории. На Web-сайте TechCrunch (популярный новостной сайт для компьютерных

фанатов) 27% предпочитают Firefox, другие 27% используют Chrome, в то время как Internet Explorer пользуются только 16%.

Еще один показатель — устройство, применяемое для путешествий в Web. В предыдущих статистических показателях отражено использование на обычных персональных компьютерах, а в смартфонах и планшетах применяются специализированные браузеры. Например, те, кто пользуются новейшими продуктами Apple, такими как iPhone, iPad и iPod Touch почти всегда путешествуют в Интернете с помощью Safari. Точно так же, если приживется Google OS (операционная система Google для нетбуков и других Web-устройств), может вырасти количество людей, использующих Chrome.

Все эти подробности сводятся к одному простому выводу: вы выиграете, если ваши Web-страницы можно будет просматривать в разных Web-браузерах. Поэтому не забывайте о следующих двух советах.

- *Строго придерживайтесь Web-стандартов.* Для того чтобы все видели корректное отображение ваших Web-страниц, пишите строгий, основанный на стандартах HTML-код и не забывайте включать во все Web-страницы `doctype`, как описано в *главе 2*.
- *Проверяйте ваш Web-сайт в разных браузерах.* Прежде чем начать разработку свежего проекта для нового сайта, создайте тестовую страницу и опробуйте ее в самых популярных браузерах (сейчас это IE, Firefox и Chrome). В этом случае вы не создадите нечаянно Web-страницы, содержащие странности ваших любимых браузеров. После завершения вашего первоначального проекта можно провести тестирование только в двух браузерах: IE и Firefox или Chrome (т. к. визуализация страниц обычно совместима в Firefox и Chrome).

Web-серверы

Когда вы вводите Web-адрес в браузере, Web-сервер получает запрос от браузера и отправляет обратно нужную Web-страницу. На загруженном Web-сайте эта обычная задача может потребовать большого объема работ, поэтому Web-серверы, как правило, — это компьютеры промышленного уровня. Тем не менее на среднем ПК с ОС Windows и правильно подобранными установочными параметрами можно разместить Web-сайт, как правило, без особых усилий (см. приведенную далее врезку). Но обычно большинство людей находят коммерческую компанию, именуемую Web-хостом и предоставляющую им место на своих серверах, как правило, за помесячную плату. Другими словами, вам необходимо арендовать какое-то пространство в Web.

Иногда вы можете арендовать пространство в той же компании, услугами которой пользуетесь для выхода в Интернет; они даже могут включать предоставление места на сервере в ваш пакет услуг по подключению Интернета. В любом случае вам придется скопировать ваши вновь созданные Web-сайты на какие-то удаленные компьютеры, чтобы дать возможность аудитории во всем мире насладиться вашими талантами.

В *главе 3* вы узнаете больше о том, как Web-браузер путешествует в Web и находит нужную страницу. Сейчас же представим себе картину в целом, чтобы вы могли начать разработку плана вашего первого Web-сайта.

ЧАСТО ЗАДАВАЕМЫЙ ВОПРОС

Превращение в Web-хост

Могу ли я запустить Web-сервер?

Теоретически, определенно можете. Всемирная паутина разработана как открытое сообщество, и никто вас не может остановить. На практике это совсем не просто, независимо от количества родственников, отлично знающих компьютер.

Несколько мгновенно возникающих проблем мешают большинству, за исключением самых амбициозных людей, запускать свои собственные серверы. Во-первых, вам нужен надежный компьютер, работающий 24 часа в сутки. На этом компьютере также необходимо запустить специальное программное обеспечение для Web-хостинга, которое сможет предоставлять Web-страницы по запросам браузеров.

Следующая проблема состоит в том, что вашему компьютеру потребуется соединение с Интернетом специального типа, называемое *фиксированным IP-адресом*.

IP-адрес — это число, идентифицирующее ваш компьютер в Web. (IP — это сокращенно Internet Protocol, очень удачный стандарт, основанный на правилах, управляющих способом взаимодействия разных устройств в сети.)

Для того чтобы иметь компьютер, способный открывать ваш Web-сайт, и быть уверенным, что другие смогут его найти, ваш IP-адрес обязательно должен быть фиксированным, другими словами, вам нужно заблокировать его так, чтобы он не менялся непрерывно. Большинство ISP (Internet Service Providers, Интернет-провайдеров) назначают IP-адреса случайным образом, когда они требуются, и произвольно меняют их, т. е. большинство пользователей не могут размещать на своих компьютерах постоянные Web-сайты, по меньшей мере без специального программного обеспечения. Если вам все еще интересно, можете позвонить своему интернет-провайдеру и спросить, предоставляют ли они обслуживание фиксированных IP-адресов и сколько это стоит.

Планирование Web-сайта

Последнее, что нужно сделать перед тем, как начать создавать Web-сайт, — обрушить на себя лавину теоретических знаний. Но любой начинающий автор сбережет время и силы, предварительно разработав небольшой план. В следующих разделах даны краткие рекомендации, чтобы вывести вас на верную дорогу.

Виды сайтов

Вы вряд ли создадите удачный сайт, если не решите для чего он. У некоторых людей есть вполне конкретная цель (например, получить работу или создать рекламу книге), а другие намерены просто в полной мере самовыразиться. В любом случае познакомьтесь со следующим перечнем, чтобы составить представление о разных видах Web-сайтов, которые вы могли бы создать.

- *Персональные сайты* целиком ваших рук дело. По мере того как мир становится все более Web-осведомленным, кажется, что все стремятся построить свои жилища во Всемирной сети. Для того чтобы показать фотографии детей родственникам, описать свое путешествие в Куала Лумпур или просто поделиться последними раздумьями и идеями, персональный сайт — самое подходящее место.

  При создании персонального Web-сайта меру честолюбия определяете вы сами. Если вам нужны только подборка ссылок для бесед с друзьями или место для

размещения фотографий, вы можете уберечь себя от лишних хлопот, зарегистрировавшись в социальной сети, например, Facebook (**www.facebook.com**). Если вы хотите создать сайт в виде дневника для описания своей жизни, работы или страстного увлечения коллекционированием шляп, возможно, вам вполне подойдет персональный блог (см. например Web-сайт **www.blogger.com**, который обсуждается в *главе 13*). Если вы более честолюбивы (скажем, хотите создать генеалогическое дерево пяти поколений вашей семьи) или мечтаете о полном контроле над дизайном сайта (забудьте синий цвет в Facebook), или же вам необходимо единое Web-пространство для того, чтобы собрать вместе разные интересы, самостоятельно созданный персональный сайт — признак человека с собственным стилем.

- *Сайты резюме* — это специальный тип персонального сайта и мощный инструмент карьерного роста. Вместо того чтобы создавать цифровые копии множества бумажных резюме, почему бы не разослать электронные письма и не раздать визитные карточки со ссылкой на интерактивное резюме? Немного поработав над планом, можно включить в вашу интерактивную краткую автобиографию гораздо больше подробностей по сравнению с ее бумажным двойником, например, ссылки на прежние компании, интерактивный портфолио и даже звучащее фоном музыкальное сопровождение с песней "YMCA" (что, конечно же, не рекомендуется).
- *Тематические сайты* сосредоточены на определенном предмете, интересующем вас. Если вас больше, чем обсуждение собственной жизни, интересует создание Web-сайта о ваших любимых музыке, картинах, книгах, еде, политических движениях или участниках телепередачи "American Idol", тематический сайт — то, что вам нужно.

  Прежде чем создавать сайт, убедитесь, захотят ли другие люди со схожими интересами посещать его, и просмотрите уже существующие сайты по этой теме. Лучшие тематические Web-сайты привлекают других людей с теми же интересами. Худшие содержат десятки одних и тех же ссылок, которые вы можете найти где-нибудь еще. Помните, что в Web — море информации. Последнее, что здесь нужно — еще одна коллекция самых разных вещей Меган Фокс, собранных ее фанатами.
- *Событийные сайты* создаются не на годы — они приурочены к конкретному событию. Самый обычный пример — свадебный Web-сайт. Главные участники события создают его, чтобы предоставить инструкции сопутствующую информацию, ссылки на журналы регистрации подарков и несколько романтических фотографий. Когда свадьба проходит, сайт исчезает или превращается во что-то другое (например, персональный сайт, подробно описывающий медовый месяц). Аналогичным образом можно представить и другие события, такие как семейные встречи, костюмированные вечеринки, организуемые вами марши протеста.
- *Рекламные сайты* идеальны, если вы хотите похвастаться подготовленным вами компакт-диском или только что вышедшей из печати книгой. Они создаются для того, чтобы сообщить о конкретном предмете, будь то сделанные вручную ке-

рамические изделия или ваша собственная программа. Иногда эти Web-сайты превращаются в сайты малого бизнеса, на которых вы действительно продаете свои изделия (см. *следующий пункт перечня*).

- *Сайты для малого бизнеса* (или интернет-коммерции) демонстрируют в самом выгодном свете торговлю во Всемирной паутине практически всем, начиная с портативных музыкальных плейеров и заканчивая лекарствами, отпускаемыми по рецептам. Сайты интернет-коммерции очень широко распространены и сейчас трудно поверить в то, что во времена зарождения Web никому и в голову не могло прийти делать деньги в Интернете.

  Создание полномасштабного сайта электронной коммерции, такого как Amazon.com или eBay, не под силу отдельному человеку. Такие сайты нужно поддерживать с помощью сложных программ и компьютерных языков программирования незаурядного уровня. Но если вы идете в Web зарабатывать деньги, не оставляйте надежды! Современные передовые компании, например Pay-Pal и Yahoo, предлагают сервисы, способные помочь вам разработать торговый сайт с расчетами по электронным картам и принимать платежи с кредитных карт. Кроме того, можно разместить рекламные объявления Google или настойчиво предлагать товары с Web-сайта Amazon и таким образом зарабатывать деньги.

Когда вы точно определите суть и смысл вашего Web-сайта, легче будет понять, кто станет его посещать. Для создания эффективного сайта очень важно представлять себе его аудиторию. (И даже не думайте соглашаться с мыслью, что вы просто создаете сайт для себя — если это так, нет никакого смысла помещать его в Интернет!)

Время жизни вашего сайта

Web-пространство непрерывно меняется. Сейчас оно не то же, что в прошлом году и даже не такое, как 15 секунд назад.

Заслуживают внимания два суждения о времени жизни Web-сайтов.

- Лучшие Web-сайты постоянно совершенствуются. Их создатели добавляют поддержку новых средств браузеров, изменяют внешний вид сайта в соответствии с современными стилевыми тенденциями и, что важнее всего, постоянно обновляют информационное наполнение.
- Если Web-сайт перестает меняться, ему нужны реанимационные мероприятия. Многие замечательные Web-сайты погибли из-за отсутствия внимания.

Вспомните ваши любимые сайты. Скорее всего, они меняются ежедневно. Хороший Web-сайт не тот, к которому вы обратились однажды и забыли о нем, а сайт, на котором вы сделали закладку и к которому периодически возвращаетесь. В каком-то смысле Web-сайт похож на телевизионный канал. Если создатели не пополняют его новой информацией, на сайте вы видите повторный показ.

Создание Web-сайта — очень непростая задача, а постоянно обновлять информационное содержимое еще труднее. Далее приведено несколько рекомендаций, которые могут помочь.

- Решайте задачи поэтапно. Когда вы разместите свой первый Web-сайт во Всемирной паутине, он наверняка не будет доделан до конца. Воспринимайте его как версию 1 и сразу начинайте продумывать небольшие изменения, которые будут внесены в следующую версию. Мало-помалу, шаг за шагом вы сможете включить в него все, что сайт по-вашему должен содержать.
- Выберите компоненты, которые сможете изменять постоянно, а остальные оставьте в покое. Вы не сможете просматривать и обновлять весь Web-сайт еженедельно. Гораздо лучше определить разделы, которые вы хотите изменять регулярно. Например, на персональном сайте можно разместить новости на отдельной Web-странице и обновлять только ее. На Web-сайте для малого бизнеса можно сосредоточиться на главной странице, на ней вы сможете рекламировать новые товары и специальные предложения.
- Проектируйте Web-сайт так, чтобы его легко можно было менять. Этому принципу следовать труднее всего, поскольку здесь требуется не только планирование, но и немного опыта, приобретенного нелегким трудом. Став более опытным разработчиком Web-сайтов, вы научитесь создавать легко обновляемые страницы. Один из способов — разделение информации на несколько страниц, позволяющее добавлять новое содержимое без реорганизации всего сайта целиком. Другой вариант — применение каскадных таблиц стилей для отделения параметров форматирования страницы от ее информационного наполнения (см. *главу 6*). В этом случае вы легко сможете добавить новый материал, и при этом не понадобится полностью форматировать его, чтобы добиться согласованности с остальной частью страницы.

Приемы хорошего дизайна

Каждый год сотни Web-сайтов побеждают в конкурсах на самый отвратительный сайт. На одних из рассматриваемых сайтов размещены вертящиеся глобусы и ярко-розовый текст на фоне цвета лайма. У других сайтов корявые поисковые системы и карикатурно мигающие фоновые подложки. Но независимо от огрехов дизайна плохих, ужасно плохих Web-сайтов на удивление очень много.

Может быть дело в том, что создание Web-сайта — не такая уж трудная задача. А может быть в каждом из нас живет скрытое желание поиграть с цветом, фактурой и звуком, и замечательные Web-инструменты будят в нас самые уродливые инстинкты. Широко распространенные ошибки дизайна можно увидеть на Web-странице **www.angelfire.com/super/badwebs** (рис. 1.3). Можно также заглянуть на Web-сайт **www.worstoftheweb.com**, представляющий новых нарушителей ежемесячно.

Эта книга не сделает вас профессиональным Web-дизайнером. Но она сможет научить вас крайне важному искусству "Как не создавать плохие Web-сайты". В книге вы найдете полезные советы, предложения и предупреждения о пользе дизайна и проблемах, с ним связанных (обращайте особое внимание на врезки с заголовком "Уголок дизайнера"). Пока приведу несколько общих принципов, которые помогут вам не попасть в список худших в Web (если, конечно, вы сами не захотите этого).



Рис. 1.3. Web-сайт с сознательно допущенными ошибками. **www.angelfire.com** делает полезное дело, показывая заголовки с полосами прокрутки, сумасшедше мигающий фон и неразборчивый текст, все те элементы, которых следует избегать на ваших собственных Web-страницах

- *Не усложняйте сайт (и не раздражайте своих посетителей).* Вы можете наполнить Web-страницу множеством ненужных украшений и примочек. Но советую отказываться от них, если в этом нет необходимости. Вы убедитесь, что проявляя сдержанность, можно несколькими искусными штрихами создать остроумный и продвинутый сайт. Множество причудливых приемов могут сделать ваш сайт заумным и бредовым. Чем меньше всяких фокусов, тем больше вероятность того, что графическая мишура не затмит информационное содержимое и не заставит посетителей в раздражении покинуть ваш сайт.
- *Будьте последовательны.* Как бы по вашему мнению не был логичен ваш Web-сайт, найдется множество посетителей, мыслящих по-другому. Чтобы избежать путаницы, используйте страницы с похожей структурой, аналогично оформленными заголовками и ссылками, одинаковыми графическими приемами, общим навигационным меню и т. д. Такой подход позволит посетителям на вашем Web-сайте чувствовать себя как дома.
- *Составьте представление о своей аудитории.* У каждого типа сайтов есть свои неписаные правила. Не следует применять на сайте интернет-магазина тот же дизайн, что и на Web-странице, рекламирующей экспериментальный инстру-

ментальный ансамбль электрических губных гармошек. Для того чтобы понять, что годится, а что нет, просмотрите множество разных сайтов, имеющих дело с содержимым, подобным вашему.

НА ПРОФЕССИОНАЛЬНОМ УРОВНЕ

Расширяйте аудиторию, насколько это возможно

Нужно не только понимать свою аудиторию, но также представлять ее технические возможности. Хорошие Web-дизайнеры стараются не применять компьютерные хитрости, если не все еще научились ими пользоваться. Ни что так не огорчает, как посещение сайта, обещающего отличную анимацию и отображающего пустой экран из-за того, что на вашем компьютере нет нужного подключаемого модуля, плагина (дополнительной программы, снабжающей ваш браузер новыми средствами). Никого не позабавит найденный Web-сайт, разукрашенный для широкоэкранных мониторов и невыносимо зажатый (или того хуже, частично обрезанный) на маленьком экране нетбука.

Создатели большинства популярных Web-сайтов непременно учитывают такого рода проблемы. К примеру, не забывайте о тех людях, чьи компьютеры не позволяют им купить книгу на Amazon.com, сделать предложение на eBay или провести поиск в Google. (Думаете ли вы об этой пусть и очень малочисленной категории?) Чтобы сделать свой Web-сайт столь же легко доступным, как перечисленные лучшие сайты, необходимо строго придерживаться общепринятых Web-стандартов, следовать советам из этой книги и проверять свой сайт на разных компьютерах.

Общеизвестно, что среднестатистический Web-дизайнер проходит три этапа в своем развитии: 1) "Я только учусь, поэтому не буду усложнять"; 2) " Я — Web-гуру и докажу это, нагромоздив множество примочек"; 3) "Меня достали проблемы совместимости разных браузеров, поэтому не буду усложнять".

Компоненты Web-сайта

Теперь, когда вы знаете, Web-сайт какого сорта вы хотите создать, пора приступать к практическим занятиям. Построение Web-сайта — это не только выбор подходящего дизайна и написание основательного контента. Вы должны связаться с другими компаниями, чтобы поместить ваш Web-сайт во Всемирную паутину (World Wide Web) (и присвоить ему легко запоминающийся адрес, например, **www.StylinViolins.com**).

В приведенном далее кратком списке перечислено все, что нужно любому Web-сайту, и указано, в каком месте книге можно узнать об этом.

- *Web-страницы.* Любой Web-сайт формируется из отдельных Web-страниц. Для создания базовой Web-страницы вы должны знать HTML (HyperText Markup Language, Язык гипертекстовой разметки) — язык Всемирной паутины. В следующей главе вы создадите свою первую Web-страницу.
- *Web-пространство.* Создание Web-страниц — увлекательное занятие, но для того, чтобы другие могли их увидеть, необходимо поместить страницы на Web-сервер. В *главе 3* вы узнаете о том, как сделать вашу первую Web-страницу интерактивной с помощью платного сервиса или бесплатной альтернативы.

- *Имя домена.* Между Web-адресами **www.inetConnections.com/Users/~jMallone012/web** и **www.JackieMallone.com** существует огромная разница. Вы можете получить свое персональное имя домена, если оно не занято. Стоит это недорого — около 10 или 15 долларов в год. Если вы хотите поместить адрес вашего Web-сайта на визитную карточку или буклет для малого бизнеса, собственное имя домена — лучший выбор. В *главе 3* вы узнаете, как его купить.

> ***ПРИМЕЧАНИЕ***
>
> Имя домена — это первая составная часть Web-адреса, которая обозначает Web-сервер, хранящий и обслуживающий ваш сайт. В URL-адресе **www.ebay.com/help/index.html** имя домена — **www.ebay.com**. В *главе 3* вы узнаете гораздо больше об именах доменов и URL-адресах (сокращение для Universal Resource Locator, Унифицированный указатель ресурсов, также называемый Web-адресом) и о том, как они работают.

- *Средства Web-дизайна*. Создание Web-страницы с чистого листа — это замечательный способ приобретения новых навыков, но он очень уж медленный и болезненный, когда разрабатывается целиком весь Web-сайт. Для перехода на следующий уровень вам придется применить профессиональное средство для Web-дизайна. Если у вас есть коммерческая программа, например Dreamweaver, вы в надежных руках. Даже если ее у вас нет, могут помочь многочисленные бесплатные или условно-бесплатные программы. В *главе 4* рассматриваются возможные варианты средств и предлагается помощь по их освоению.
- *Гиперссылки*. Сама по себе Web-страница может предложить лишь кое-что. Реальное волшебство начинается, когда вы связываете страницы друг с другом с помощью ссылок. В *главе 8* описан механизм создания ссылок, позволяющий посетителям перемещаться по вашему сайту.
- *Необходимые дополнения*. После знакомства с основами создания Web-страниц и Web-сайтов у вас появится множество новых непокоренных вершин. Вы можете включить ваш сайт в поисковую машину (см. *главу 11*), организовать собственное сообщество, форум (см. *главу 12*) и даже продавать свои вещи (см. *главу 14*). Хотите еще? Почему бы не создать интерактивные страницы с помощью программного кода на JavaScript (см. *главу 15*), разработать раскрывающееся меню (см. *главу 16*) и добавить аудио- и видеоданные (см. *главу 17*)? Все эти средства и приемы помогут не ограничиваться основами Web-дизайна и стать настоящим высококлассным Web-мастером.

2

ГЛАВА 2

Создание вашей первой Web-страницы

Web-страницы — основные элементы Web-сайта, все Web-сайты состоят из набора страниц, одной или нескольких. Идеальная Web-страница содержит достаточно информации, чтобы заполнить все окно браузера в ширину, но при этом не вынуждает пользователя прокручивать окно браузера с утра до полудня, чтобы добраться до конца страницы. Другими словами, на идеальной странице достигается баланс: нет ощущения пустоты от излишка свободного места, и в то же время не возникает стресс от лавины предлагаемой информации.

Наилучший способ составить представление о том, что должна содержать Web-страница, — заглянуть на ваши любимые Web-сайты. Новостной сайт, например, **www.nytimes.com** отображает каждую новостную статью на отдельной странице (и разбивает длинные истории на несколько страниц). В интернет-магазине, таком как **www.amazon.com**, у каждого товара есть своя страница. Персональный Web-сайт можно было бы разделить на отдельные Web-страницы со следующими заголовками: "Обо мне", "Отпускные фотографии" и "Совершенно секретные рецепты приготовления свинины".

Пока не беспокойтесь о том, как разбить ваш Web-сайт на отдельные страницы; мы вернемся к этой проблеме в *главе 8*, когда начнем связывать страницы друг с другом. Сейчас ваша основная задача — понять, как работает базовая Web-страница и как создать свою собственную. Этим вы и займетесь в данной главе. Попутно вы познакомитесь ближе с самым важным стандартом Web-дизайна — языком HTML.

Введение в HTML: язык Всемирной паутины

Web-страницы описываются на языке гипертекстовой разметки — HTML (HyperText Markup Language). Неважно, содержит ли ваша страница серии текстовых блогов, десятки фотографий вашего ручного лемура или серьезно отформатированный сценарий фильма, — если вы просматриваете их в окне браузера, скорее всего перед вами HTML-страница.

Язык HTML играет на Web-страницах ключевую роль: он сообщает браузерам, как отображать Web-страницу, применяя для определения начала абзаца, выделения слова курсивом или вывода на экран изображения специальные команды (называемые *тегами*). Для того чтобы создавать свои собственные Web-страницы, следует научиться применять это семейство тегов.

HTML — настолько важный стандарт, что немало страниц этой книги посвящено его приемам, хитростям и недостаткам. Любая Web-страница, которую вы создадите попутно, будет не чем иным, как настоящим HTML-документом.

ПРИМЕЧАНИЕ

Стандарт HTML никак не может повлиять на способ извлечения Web-страницы из Всемирной паутины. Эта задача HTTP (HyperText Transport Protocol, гипертекстовый транспортный протокол), низкоуровневой системы коммуникации, которая позволяет двум компьютерам обмениваться данными в Интернете. Если воспользоваться аналогией с телефонным разговором, телефонная линия — это HTTP, а пикантные новости светской хроники, которые вы обсуждаете с тетей Мартой, — HTML-документы.

Как устроен HTML-файл

Внутри HTML-страница — не что иное, как обычный текстовый файл. Это значит, что любая Web-страница целиком состоит из букв, цифр и нескольких специальных символов (таких как пробелы, знаки пунктуации и любые другие знаки, которые можно найти на вашей клавиатуре). HTML-файл принципиально отличается от того, что вы найдете в типичном *двоичном файле* на вашем компьютере. Двоичный файл содержит элементы настоящего машинного языка, последовательности нулей и единиц. Если какая-нибудь достаточно глупая программа попытается преобразовать двоичную информацию в текст, вы увидите абракадабру.

Для того чтобы понять разницу между двоичным и текстовым файлами, взгляните на рис. 2.1, отображающий традиционный документ Word (DOC-файл) со всеми деталями. Сравните его с тем, что показано на рис. 2.2, на котором препарирован HTML-документ с тем же содержимым.

Для просмотра HTML-документа достаточно обычного текстового редактора. В состав Windows входит один из них, называемый Notepad (Блокнот). Для того чтобы открыть его в Windows Vista или Windows 7, щелкните мышью кнопку **Start** (Пуск), введите в строку поиска `Notepad` и затем щелкните кнопкой мыши появившуюся пиктограмму **Notepad** (Блокнот). На компьютере Mac воспользуйтесь его стандартным текстовым редактором TextEdit, который найдете в меню **Applications | TextEdit** (Программы | TextEdit).

В выбранном текстовом редакторе открыть ваш HTML-файл можно с помощью простой последовательности команд меню **File | Open** (Файл | Открыть). Если вы загрузили из Интернета сопроводительные материалы к книге (все их можно найти на странице **Missing CD** (Отсутствующий CD) по адресу **www.missingmanuals.com/cds/caw3**), то сможете попытаться открыть файл popsicles.htm, показанный на рис. 2.2.

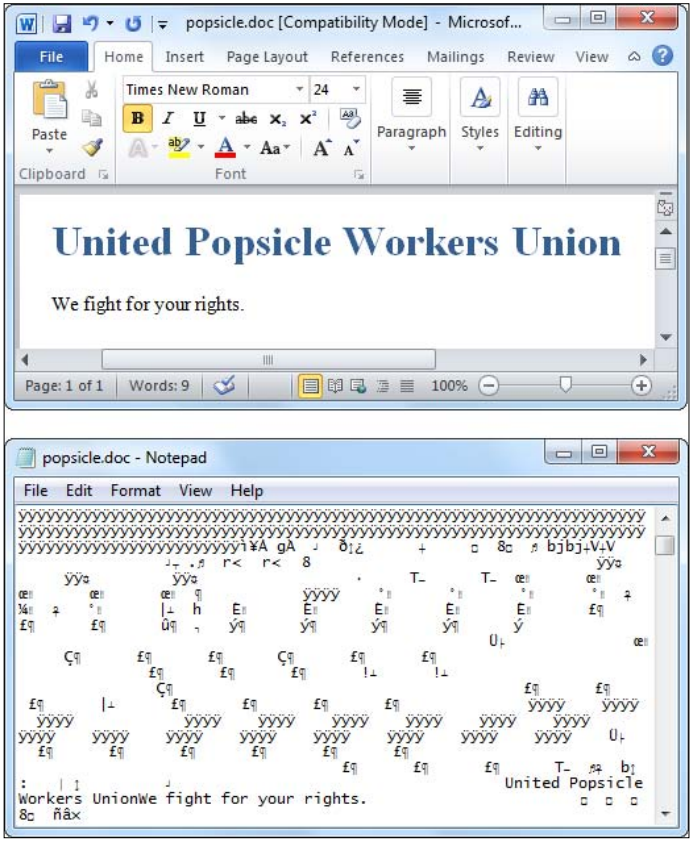


Рис. 2.1. Даже если ваш документ кажется довольно простым в редакторе Word (верхнее окно), DOC-файл не выглядит столь же привлекательно, если уйти из программы Word и открыть его в текстовом редакторе, например, Notepad (нижнее окно). Обычно текстовые редакторы преобразуют файловую строку из нулей и единиц в бессмысленный поток пугающей тарабарщины. Настоящий текст там где-то есть, но он глубоко спрятан в непонятном машинном языке

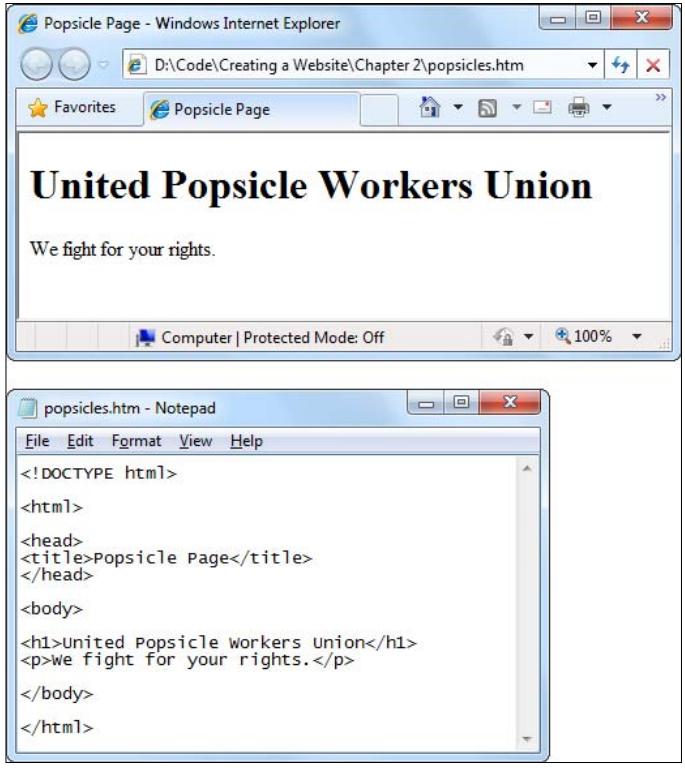


Рис. 2.2. HTML-документы хранятся как обычный текст. Верхнее окно: В Web-браузере отображается документ Word, переписанный в виде HTML-документа. Нижнее окно: Когда вы открываете HTML-файл в текстовом редакторе, легко можно найти весь текст из исходного документа, сопровождаемый несколькими дополнительными символами в угловых скобках (<>). Это HTML-теги, передающие сведения о структуре документа и его форматировании

Просмотр HTML-кода реальной Web-страницы

Большинство текстовых редакторов не позволяет открывать Web-страницу непосредственно в Интернете. Для этого они должны уметь отправить в сети запрос Web-серверу, а эту работу лучше оставить Web-браузеру. Но у браузеров самих есть возможность показать вам исходный HTML-код Web-страницы. Для этого нужно выполнить следующие действия:

1. Откройте выбранный вами Web-браузер.
2. Перейдите на Web-страницу, которую хотите просмотреть.
3. В вашем браузере щелкните страницу правой кнопкой мыши и выберите команду **View Source** (Просмотр HTML-кода) (в Internet Explorer) или **View Page Source** (Исходный код страницы) (в Firefox и Chrome).

После выбора команды появится новое окно, отображающее исходный HTML-код Web-страницы, которую вы видите.

> ***СОВЕТ***
>
> У браузера Firefox есть удобное средство, которое позволяет увидеть фрагмент HTML-кода сложной страницы. Просто выделите на самой Web-странице интересующий вас текст, щелкните его правой кнопкой мыши и выберите команду **View Selection Source** (Исходный код выделенного фрагмента).

Большинство Web-страниц существенно сложнее файла примера popsicles.htm (см. рис. 2.2), поэтому вам придется одолеть множество HTML-тегов. Но освоив гору информации, вы получите очень полезную возможность заглядывать под покров любой Web-страницы. В жизни профессиональные Web-разработчики часто пользуются этим приемом, чтобы познакомиться с работой своих конкурентов.

Создание HTML-файла

Один из самых важных секретов написания Web-страниц: для того чтобы начать создавать собственные Web-страницы, вовсе не нужен реальный Web-сайт. Вот почему вы легко можете создавать тестовые страницы, пользуясь только своим компьютером. На самом деле, вам даже не требуется интернет-соединение.

Базовый метод прост.

1. Запустите ваш любимый текстовый редактор.
2. Начните писать HTML-код.

   Конечно, этот пункт выполнить сложновато, поскольку вы до сих пор не познакомились с HTML-стандартом. Держитесь! Помощь рядом, в следующих разделах. Пока вы можете использовать приведенный далее фрагмент HTML-кода. Просто наберите его точь-в-точь, как в книге; текст, слэши, угловые скобки и все остальное:

```
<h1>United Popsicle Workers Union</h1>
<p>We fight for your rights.</p>
```

На деле в этом документе из двух строк пропущено несколько структурных деталей, которые должны быть у любой уважающей себя Web-страницы. Тем не менее все браузеры смогут прочитать этот HTML-фрагмент и корректно отобразить то, что вам нужно: а именно две строки отформатированного текста (см. рис. 2.2).

3. Когда закончите набор, сохраните документ.

   В Notepad (Блокнот) выберите команды меню **File | Save** (Файл | Сохранить). В нижней части диалогового окна **Save As** (Сохранить как) в раскрывающемся списке **Encoding** (Кодировка) выберите строку **UTF-8**. (Ваша страница будет функционировать, даже если вы пропустите этот шаг, но лучше осчастливьте специальные программы, валидаторы.)

   В редакторе TextEdit сначала нужно выбрать команды **Format | Make Plain Text** (Формат | Конвертировать в простой текст), чтобы быть уверенным в том, что программа сохранит вашу страницу как обычный текстовый файл (а не как форматированный или двоичный документ). В дальнейшем для сохранения файла можно будет пользоваться командами **File | Save** (Файл | Сохранить). В поле **Plain Text Encoding** (Кодировка простого текста) выберите **Unicode (UTF-8)**.

   Присваивая файлу имя, используйте расширение htm или html (см. следующее примечание). Например, типичное имя HTML-файла выглядит так: LimeGreenPyjamas.html. Редактор TextEdit может поинтересоваться, действительно ли вы хотите применить расширение htm или html вместо стандартного для текстового файла расширения txt. Щелкните кнопкой мыши ответ **Use .htm** (Использовать .htm). В Notepad (Блокнот) этот шаг отсутствует. Но если вы захотите увидеть свои HTML-файлы в списке файлов, придется в поле **Save as type** (Тип файла) выбрать строку **All Files (*.*)** (Все файлы (*.*)) вместо строки **Text Documents (*.txt)** (Текстовые документы (*.txt)).

   Примечание

   В принципе можно использовать любое расширение в имени файла. Но применение htm или html устраняет путаницу (вы сразу определите, что этот файл — Web-страница) и помогает избежать других распространенных проблем. Например, указав расширение htm или html, вы можете быть уверены, что при двойном щелчке кнопкой мыши имени файла ваш компьютер будет знать наверняка, что открывать его следует в Web-браузере, а не в какой-то другой программе. Кроме того, использовать расширение htm или html важно, если вы планируете загружать ваши файлы на Web-сервер; несговорчивые серверы могут отказаться принимать страницы с нестандартными файловыми расширениями.

4. Для того чтобы посмотреть на свою работу, откройте файл в Web-браузере.

   Если вы использовали расширение htm или html, открыть файл обычно легко: нужно дважды щелкнуть кнопкой мыши его имя. Если нет, придется ввести полное имя файла в строку адреса в вашем Web-браузере, как показано на рис. 2.3.

Помните, что когда вы формируете HTML-документ в текстовом редакторе, вы не можете просмотреть его в отформатированном виде. Вы увидите лишь обычный текст и команды HTML-форматирования.

СОВЕТ

Если вы измените файл и сохраните после того, как открыли его в вашем Web-браузере, можно просмотреть последние изменения, обновив страницу. В большинстве браузеров это сделать просто: нужно щелкнуть страницу правой кнопкой мыши и выбрать команду **Refresh** (Обновить) или **Reload** (Перезагрузить).

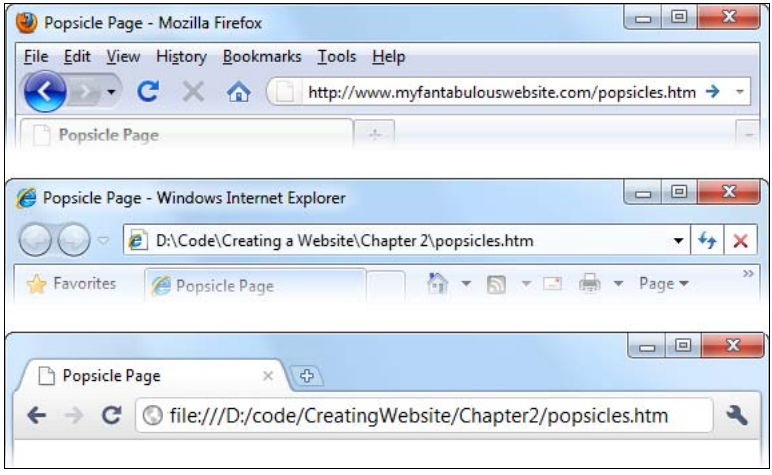


Рис. 2.3. В адресной строке браузера показано истинное местоположение текущей Web-страницы. Если увидите в адресе **http://**, она считана с Web-сервера в Интернете (верхнее изображение). Если вы просматриваете Web-страницу, находящуюся на вашем собственном компьютере, то увидите обычное полное имя файла (на среднем изображении в Internet Explorer показано местоположение файла в ОС Windows) или URL, начинающийся с префикса **file://** (на нижнем изображении показано местоположение файла в браузере Chrome). Вид локальных адресов зависит от используемых браузера и операционной системы

РАБОТА НАД ОШИБКАМИ

Если ваша первая Web-страница отображается некорректно...

... причина вероятно в способе сохранения файла.

Например, очень часто ваш документ появляется в Web-браузере без форматирования и с отображением HTML-тегов. Другими словами, в вашем браузере документ выглядит так же, как в вашем текстовом редакторе. Подобную ошибку может вызвать одна из следующих причин.

- *Вы использовали неверное расширение.* Когда вы открываете файлы, находящиеся непосредственно на вашем компьютере (а не на удаленном Web-сайте), ваш браузер пытается определить тип файла по его расширению. Если у файла вашей Web-страницы расширение txt, браузер полагает, что это обычный текстовый файл и просто выводит на экран текстовое содержимое файла. Чтобы избежать подобных ошибок, применяйте расширение htm или html.
- *Вы сохранили документ в текстовом процессоре.* Текстовые процессоры автоматически преобразуют HTML-символы, такие как угловые скобки, в символьные величины (см. *разд. "Специальные символы" в главе 5*). Например, текстовый процессор преобразует тег `<h1>` в текст "<h1>", не видя в нем команду HTML-форматирования. Чтобы избежать этого, создавайте ваш HTML-документ в текстовом редакторе.

- *Вы не сохранили его как текстовый файл.* Некоторые текстовые редакторы позволяют сохранять документы в разных форматах. Notepad (Блокнот) — нет, а TextEdit — да, и если вы нечаянно сохраните Web-страницу в формате rtf, браузер не воспримет ее как Web-страницу. Для того чтобы избежать подобной проблемы в TextEdit, убедитесь, что перешли к режиму **Format | Make Plain Text** (Формат | Конвертировать в простой текст). В других программах проверьте настройки в окне **File | Save As** (Файл | Сохранить как).

HTML, XHTML и HTML5

С момента своего появления язык HTML эволюционировал и иногда в новых неожиданных направлениях. В последнее время ситуация осложнилась. HTML — исходный язык Всемирной паутины. Последняя полная версия HTML (версия 4.01) была узаконена в 1999 г. За последнее десятилетие HTML постепенно вытеснила родственная улучшенная версия языка, названная XHTML (*eXtensible* HyperText Markup Language, Расширяемый язык гипертекстовой разметки).

> ***Примечание***
>
> Поскольку XHTML — это вариант HTML, и поскольку HTML был первым стандартом для Web, принято считать, что все Web-страницы писались и пишутся на языке HTML (что и делается в книге). В этом смысле термин "HTML" означает любую версию языка HTML или XHTML.

Теперь неприятная часть. Сегодня применяют версию XHTML 1.0 или (гораздо реже) XHTML 1.1. Предполагалось, что разработчики заменят обе эти версии языка XHTML более новым стандартом, названным XHTML 2.

К сожалению, создатели XHTML 2 утратили рациональное зерно. Они создали стандарт, который был теоретически безупречен, а практически превратился в ночной кошмар. Web-разработчиков и крупных поставщиков браузеров испугала строгость XHTML 2 и отсутствие в нем обратной совместимости с уже существующими Web-страницами. Другими словами, не существовало легкого способа превратить приличную страницу на XHTML 1.0 в крутую новинку на языке XHTML 2. Более того, в стандарте XHTML 2 отсутствовали продвинутые новые средства, в которых *действительно* нуждались многие Web-разработчики, особенно создатели сверхсложных Web-приложений.

Последний удар по XHTML 2 был нанесен, когда конкурирующая группа начала представлять первые версии стандарта, названного HTML5. Несмотря на то, что HTML5 — это по сути новый, отличающийся от прежних и до сих пор не полностью поддерживаемый всеми мировыми браузерами стандарт, он был встречен с огромным воодушевлением. Консорциум W3C (организация, разрабатывающая и внедряющая стандарты и официально отвечающая за HTML и XHTML) недавно остановила работу над обреченным стандартом XHTML 2 и начала формализацию HTML5, объявив его официальным, т. е. HTML5 — это будущее Web.

> ***Примечание***
>
> В названии HTML5 между "HTML" и "5" нет пробела. Это просто странность в названии стандарта.

Сбиты с толку? Не беспокойтесь — создатели HTML5 были очень умны. Они заставили его работать со всеми теми средствами, которые сегодня применяются в XHTML. Это значит, что вы можете писать документы на HTML5 прямо сейчас (что вы и будете делать, читая книгу), не применяя те новомодные средства, которые еще не поддерживаются браузерами, и при этом все будет отлично работать. Но главное, вы будете готовы к недалекому будущему, в котором Землю заполонят новые браузеры, и вы наконец сможете начать применять некоторые искусные средства, присущие только HTML5.

Примечание

HTML5 поддерживает все, что возможно в XHTML, причем без каких-либо изменений. Однако синтаксис XHTML строже синтаксиса HTML5, и есть несколько инструкций, которые поддерживаются HTML5 и не поддерживаются в XHTML. Web-разработчики разделились: одни считают эти инструкции — ужасно скверным стилем, а другие — всего лишь удобными краткими формами. Чтобы привить вам хорошие манеры, в этой книге всегда используется более строгий синтаксис XHTML.

АНОНС HTML5

Кое-что еще об HTML5

Освоить HTML5 не так просто. Отчасти проблема состоит в том, что под названием HTML5 скрывается последняя версия языка HTML и ряд стандартов на стадии становления, все еще находящихся на этапе разработки. Дополнительное разнообразие вносит и то, что часть этих стандартов действует в современных самых популярных браузерах, а другие до сих пор не включены даже в самые свежие версии HTML5.

Интересно отметить, что HTML5 предлагает лишь несколько дополнений к стандартному семейству тегов, с которыми вы познакомитесь в этой главе. Большинство из них предоставляют Web-браузерам, поисковым машинам и другим автоматизированным средствам дополнительную структурную информацию о ваших Web-страницах. Например, можно применять специальные теги, чтобы сообщить браузеру, в какой части вашей страницы содержатся навигационные ссылки.

Еще чудесней то, что в HTML5 также включен новый способ вставки видео на вашу Web-страницу, не требующий плагина, как в большинстве современных браузеров (см. дополнительную информацию об этом средстве в *главе 17*). Но главная задача HTML5 и связанных с ним стандартов — предоставить набор мощных средств для разработки Web-приложений, Web-страниц, с которыми вы взаимодействуете. Примерами могут служить хост для рисования плоских изображений, геолокационный инструмент, указывающий точное географическое местонахождение вашего посетителя, и технология, позволяющая интерактивным Web-страницам работать автономно. Все они крепко обосновались на переднем крае Web-дизайна, но ни одно не работает в версиях Internet Explorer, предшествующих IE9.

Если вас интересует будущее, просмотрите врезки этой книги с заголовками "Анонс HTML5". Можно также пролистать *приложение 1* с кратким обзором некоторых новых элементов в языке HTML5.

HTML-теги

Теперь, когда вы знаете, как заглянуть в имеющийся HTML-файл и как создать свой собственный, следующий шаг — постараться понять, что происходит *внутри* типичного HTML-файла. В нем все вращается вокруг одной структуры — тегов.

HTML-теги — это команды форматирования, сообщающие браузеру, как превратить обычный текст в нечто визуально привлекательное. Если вы уберете все теги из HTML-документа, на странице останется обычный неформатированный текст и ничего больше.

Что такое тег?

Тег легко обнаружить по угловым скобкам, двум специальным символам: < и >. Для создания тега между угловыми скобками вводится HTML-код. Этот код предназначен только для браузеров; Web-посетители никогда не видят его (если не воспользуются последовательностью команд меню **View | Source** (Вид | HTML-код) для просмотра HTML-кода). По существу код — это команда, сообщающая браузеру, как форматировать следующий далее текст.

Например, существует простой тег , сокращение от слова "bold" (жирный) (названия тегов всегда набираются строчными буквами). Когда браузер обнаруживает этот тег, он переключается на форматирование жирным шрифтом, которое воздействует на текст, следующий за тегом. Далее приведен пример.

```
Этот текст не жирный. <b>Этот текст жирный.
```

Но тега недостаточно; его называют *начальным* или *открывающим тегом*, что означает включение некоторого эффекта (в данном случае написание жирным шрифтом). У большинства начальных тегов есть пара в виде *конечного* или *закрывающего тега*, который отключает эффект.

Вы легко узнаете конечный тег. Он выглядит так же, как начальный, но начинается с прямого слэша. Конечные теги выглядят как </, в отличие от начальных <. Итак, конечный тег для выделения жирным шрифтом — . Далее приведен пример.

```
Это не жирный шрифт <b>Внимание!</b> Теперь мы вернулись к обычному шрифту.
```

Браузер отобразит его следующим образом:

Это не жирный шрифт **Внимание!** Теперь мы вернулись к обычному шрифту.

В этом примере отражается еще один важный принцип работы браузеров: они всегда обрабатывают теги по порядку их следования в вашем тексте. Для того чтобы получить форматирование в нужном месте текста, следует правильно разместить теги и .

Как вы успели заметить, у браузера очень простая работа. Он последовательно просматривает HTML-документ в поисках тегов и устанавливает и сбрасывает разные параметры форматирования. Браузер принимает остальное содержимое (все, что не является тегом) и отображает его в своем окне.

> ***Примечание***
>
> Вставка тегов в обычный неформатированный текст называется *разметкой* документа, а теги — знаками разметки. Когда вы просматриваете исходный HTML-документ, вас может интересовать содержимое, контент (текст, вложенный в теги) или разметка (сами теги).

Представление об элементах

Как вы видели, теги применяют парами. Когда вы используете начальный тег (например, для жирного шрифта), вы должны вставить конечный тег (такой как). Эта комбинация начального и конечного тегов и текста между ними обозначает *HTML-элемент*.

Суть заключается в том, что элементы представляют собой контейнеры (рис. 2.4). Вы помещаете некоторый контент (например, текст) внутрь контейнера. Если применяются теги и , создается контейнер, задающий жирное начертание текста, находящегося внутри контейнера. Создавая свои Web-страницы, вы будете использовать разные контейнеры для хранения различных фрагментов текста. Если воспринимать элементы таким образом, легче будет не пропустить конечный тег.



Рис. 2.4. Для получения жирного шрифта необходимо начать с корректного контейнера. Это уже знакомый элемент

Примечание

Когда кто-то упоминает элемент , имеется в виду все вместе: начальный тег, конечный тег и содержимое между ними. Если же кто-то ссылается на тег , это просто означает команду, выполняющую определенное форматирование.

Конечно, жизнь не была бы такой непредсказуемой (а книги о компьютерах такими толстыми), если бы не было исключений. На самом деле есть *два* типа элементов.

- *Элементы-контейнеры* — бесспорно самый распространенный вид элементов. Они задают форматирование содержимого, находящегося между начальным и конечным тегами.
- *Пустые элементы* не задают и не отменяют форматирование. Они вставляют что-либо, например изображение, на страницу. Одним из примеров может служить элемент <br>, который выполняет перенос строки на странице. Пустые элементы в отличие от элементов-контейнеров не имеют парных тегов и в них нельзя вставить текст.

В книге во все пустые элементы включен символ слэша перед закрывающей скобкой, как будто открывающий и закрывающий теги свернуты в один. Поэтому на страницах книги вы увидите перенос строки, обозначенный как
 вместо <br>. Такая форма записи, называемая *синтаксисом пустых элементов*, удобна, т. к. четко отделяет элементы-контейнеры от пустых элементов, и вы никогда не запутаетесь.

Примечание

В недавнем прошлом Web-разработчиков призывали применять синтаксис пустых элементов, т. к. он был официальной частью языка XHTML. Но в HTML5 это необязательный компонент и в пустых элементах разрешается использовать тот же синтаксис, что и в открывающих тегах (т. е. для обозначения переноса строки можно применять как `<br />`, так и `<br>`).

На рис. 2.5 для сравнения показаны элементы обоих типов.

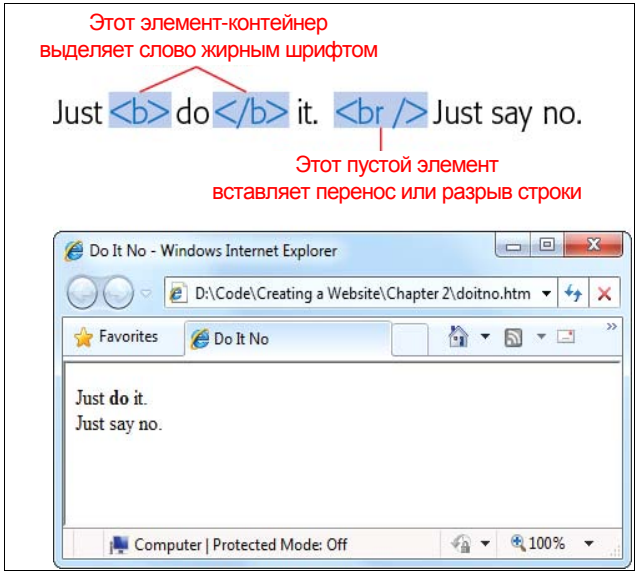


Рис. 2.5. Вверху: этот фрагмент HTML-кода содержит элемент-контейнер и пустой элемент. Внизу: браузер отображает получающуюся в результате Web-страницу

Вложенные элементы

В предыдущем примере применялся простой элемент `<b>` для выделения текста жирным шрифтом. Вы поместили нужный вам текст между тегами `<b>` и `</b>`. Но между начальным и конечным тегами можно поместить не только текст. Вы также можете вложить один элемент *внутрь* другого. На самом деле *вложенные элементы* — это один из базовых методов блочного построения Web-страниц. Вложение позволяет применять к тексту более подробные команды форматирования (например, жирный текст с курсивным начертанием), соединяя все нужные элементы в едином наборе команд. Вы также можете применять вложенные элементы для создания более сложных компонентов, таких как маркированные списки.

Для того чтобы увидеть вложенные элементы в действии, вам потребуется для работы еще один элемент. В следующем примере рассмотрим помимо элемента `<b>` элемент `<i>`, задающий курсивное начертание текста.

Вопрос заключается в том, как поступить, если вы хотите вывести курсивом часть текста с жирным шрифтом? В языке HTML нет одного элемента для этой цели, поэтому придется применить комбинацию двух элементов.

Далее приведен пример.

```
Это <b><i>слово</i></b> отображается курсивом и жирным шрифтом.
```

Когда браузер "прожует" этот обрывок HTML-кода, он выведет текст, который выглядит так:

Это ***слово*** отображается курсивом и жирным шрифтом.

Между прочим, порядок тегов <i> и <b> не имеет значения. Следующий HTML-код формирует тот же самый результат.

```
Это <i><b>слово</b></i> отображается курсивом и жирным шрифтом.
```

Но всегда помните, что закрывать теги следует в обратном порядке по сравнению с открытием. Другими словами, если вы задаете курсив, а затем жирный шрифт, то отменять нужно сначала жирный шрифт, а затем курсив. Далее приведен пример, нарушающий это правило.

```
Это <i><b>слово</i></b> отображается курсивом и жирным шрифтом.
```

В этом случае браузеры, как правило, могут восстановить нужный порядок и догадаются, чего вы хотели, но такое нарушение может стать опасной привычкой в случае написания более сложного HTML-кода.

Как вы увидите в последующих главах, язык HTML предоставляет много вариантов вложения элементов. Например, можно вложить один элемент в другой, а затем этот другой вложить в еще один элемент и т. д. до бесконечности.

Примечание

Если вы приверженец графического дизайна, вам наверняка хочется применить элементы более сложного форматирования, например, изменить выравнивание, интерлиньяж и шрифты. К сожалению, в мире Web у вас нет полного контроля над всеми элементами. В *главе 5* изложена вся нелицеприятная информация, а *глава 6* знакомит с наилучшим решением, таблицами стилей.

ЧАСТО ЗАДАВАЕМЫЙ ВОПРОС

Команда браузеру игнорировать тег

Что если я хочу, чтобы на моей Web- странице появился текст "<b>"?

Система тегов отлично работает, пока вы не захотите включить в ваш текст символы угловых скобок (< или >). Тут вы окажетесь в сложной ситуации.

Представим себе, что вы хотите написать следующий фрагмент текста, как доказательство вашего замечательного озарения:

```
Выражение 5 < 2 очевидно ложно, т. к. 5 больше, чем 2.
```

Когда браузер доходит до символа меньше чем (<), он приходит в полное недоумение. Его первое побуждение: решить, что вы начинаете тег и текст `2 очевидно ложно...` — это часть тега с длинным именем. Ясно, что это не то, на что вы рассчитывали, но то, что озадачивает браузер.

Для решения данной проблемы нужно заменить угловые скобки соответствующей специальной *символьной HTML-последовательностью*. Такая последовательность символов всегда начинается с амперсанда (`&`) и заканчивается точкой с запятой (`;`). Символьная последовательность для знака "меньше чем" — `&lt`, поскольку `lt` сокра-

щение от "less than". Точно так же символьная последовательность `&gt` — обозначение знака "больше чем".

Далее приведен корректный пример.

```
Выражение 5 &lt; 2 очевидно ложно, т. к. 5 больше, чем 2.
```

В вашем текстовом редакторе эта строка не будет выглядеть так, как вы хотите. Когда же браузер будет отображать этот документ, он автоматически заменит `<` на знак <, не путая его с тегом. Дополнительную информацию о символьных последовательностях см. в *разд. "Специальные символы" главы 5*.

Между прочим, замещение символов — одна из причин возникновения сбойных ситуаций, когда вы пытаетесь создавать ваши HTML-документы в текстовом процессоре. Когда ваш документ сохраняется в текстовом процессоре как текстовый файл, программа преобразует все специальные символы в соответствующие символьные последовательности. В результате в браузере вы видите обычный текст, а не текст, отформатированный так, как задумали.

HTML-документ

До сих пор вы изучали HTML-фрагменты — части полного HTML-документа. В этом разделе вы научитесь собирать их вместе и создадите свою первую настоящую Web-страницу.

Определение типа документа

Во времена зарождения Интернета у каждого браузера были свои странности. И их нужно было учитывать при разработке Web-страниц. Например, браузеры могли вычислять поля вокруг плавающих текстовых блоков немного по-разному, что приводило к корректному отображению страницы в одном браузере и к ее искажению в другом.

Спустя годы правила HTML (и CSS, стандарта каскадных таблиц стилей, который рассматривается в *главе 6*) были стандартизованы более тщательно. Применяя эти новые правила, любой браузер будет отображать страницу одним и тем же способом. Но это изменение стало серьезной головной болью для браузеров-долгожителей, например, Internet Explorer, переживших эпоху раннего средневековья языка HTML. Эти браузеры вынуждены были как-то поддерживать новые стандарты и при этом продолжать корректно отображать уже существующие Web-страницы, включая те, которые базировались на старых каверзах.

Web-сообщество нашло простое решение. Когда разрабатывается новая современная Web-страница, этот факт отмечается фрагментом кода, называемым *определением типа документа* (Document Type Definition, DTD) или *doctype* и расположенным в самом начале вашего HTML-документа (рис. 2.6).

Когда браузер обрабатывает страницу с типом документа, он переходит в *стандартный режим*. Затем он воспроизводит страницу самым согласованным и стандартизованным способом. В итоге страница выглядит практически одинаково в любом современном браузере.

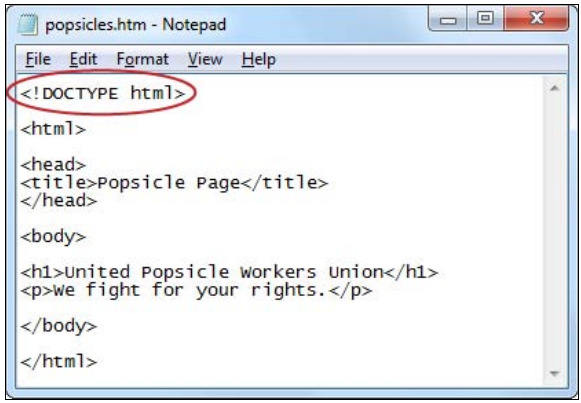


Рис. 2.6. Определение типа документа (DTD) — это первая порция данных в HTML-файле. Она сообщает браузеру, какой стандарт разметки применялся при написании страницы

Но если браузер обнаруживает HTML-документ без `doctype`, все соглашения отбрасываются. Некоторые, например Internet Explorer, переходят в *режим обратной совместимости* (quirks mode). В этом режиме Internet Explorer пытается вести себя так, как делали его ранние версии лет 10 назад с выкрутасами и странностями. Такой подход гарантирует, что старые Web-страницы будут отображаться так же, как во времена их создания, даже если они опирались на древние ошибки и каверзы браузера, просуществовавшие довольно долго до того, как были устранены. К сожалению, у каждого браузера свои индивидуальные непереносимости. Поэтому, когда вы просматриваете в разных браузерах страницу без `doctype`, вы рискуете получить текст разного размера, несогласованные поля и границы и некорректно размещенный контент. По этой причине Web-страницы без `doctype` — скверная штука и вам не следует их создавать.

Web-дизайнеры могут применять различные типы документов для обозначения стандарта разметки, которым они пользуются (например, XHTML, HTML5 или по-настоящему старый HTML 4.01). Однако у браузеров есть один неприличный секрет, касающийся типа документа: в действительности их не заботит то, о чем сообщает им ваш `doctype`. Они просто следят за тем, чтобы у вас он *был*. Дело в том, что почти все варианты `doctype` включают стандартный режим. Поэтому вы можете задать на вашей Web-странице `doctype` для XHTML 1.0, а поместить на ней содержимое в стандарте HTML5, и ваш браузер будет совсем не против.

В наши дни большинство Web-разработчиков используют тип документа HTML5, который выглядит следующим образом:

```
<!DOCTYPE html>
```

Вы заметили, что в этом типе документа не указан номер версии "5". В нем только приведен язык HTML. Это не опечатка. Напротив, такая запись отражает основной принцип HTML — поддержка и старых, и новых документов. Это также свидетельствует о том, что по мере добавления в язык HTML новых средств, все они автоматически становятся доступными на ваших уже имеющихся Web-страницах. (Это

реальный метод работы, уже применяемый в браузерах. HTML5 просто сделал его официальным.)

Для сравнения далее приводится гораздо более многословное описание типа документа для XHTML 1.0, который вы наверняка обнаружите на слегка устаревших Web-страницах.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

Даже опытные Web-разработчики вынуждены копировать `doctype` XHTML 1.0 с существующих Web-страниц, чтобы не ошибиться при наборе.

В этой книге во всех примерах используется тип документа для HTML5. Несмотря на то, что нет браузеров, поддерживающих все новое наполнение HTML5 (а некоторые версии браузеров не поддерживают ничего), все они принимают описание типа документа для HTML5 и при виде его переключаются в стандартный режим. Это означает, что у любой Web-страницы с типом документа HTML5 будет единообразное стабильное отображение, аналогичное тому, которое есть у страницы с традиционным типом документа для XHTML 1.0.

> ***ПРИМЕЧАНИЕ***
>
> Преимущество применения определения типа документа для HTML5 заключается в подготовке ваших страниц для будущего использования. Простая вставка типа документа не означает, что вы можете применять новые свойственные HTML5 средства. В действительности вам нужно постараться изо всех сил не использовать большинство из них в настоящее время, потому что у них еще нет широкой поддержки.

Базовая структура

Теперь, когда выбран `doctype`, можно писать остальную часть вашей Web-страницы.

Для создания настоящего HTML-документа начните с трех элементов-контейнеров: `<html>`, `<head>` и `<body>`. Эти три элемента, действующие вместе, описывают основную структуру вашей страницы.

`<html>`

В этот элемент включается все (кроме `doctype`), что есть на вашей Web-странице.

`<head>`

Приведенный элемент определяет *шапку* вашего документа. Он может содержать некоторую необязательную информацию о Web-странице, включая обязательное название страницы (которое ваш браузер отобразит в строке заголовка окна), необязательные ключевые слова для поиска и таблицу стилей (о которой вы узнаете в *главе 6*).

`<body>`

Этот элемент содержит сердцевину вашей страницы, включая истинное содержимое, которое вы хотите представить миру.

Комбинировать перечисленные элементы можно единственным способом. Далее приведена правильная их организация с типом документа для HTML5 в начале страницы.

```
<!DOCTYPE html>

<html>

<head>
...
</head>

<body>
...
</body>

</html>
```

Такой каркас применяется на любой Web-странице. Многоточия (...) показывают, куда вставлять дополнительную информацию. Вставленные пустые строки не обязательны, они помогают вам лучше увидеть структуру элемента.

После того как базовый каркас страницы готов, необходимо добавить в структуру несколько элементов-контейнеров. Каждой Web-странице нужен элемент `<title>`, который мы поместим в шапку. Затем вам нужно создать контейнер для текста в секции страницы `<body>`. Один из универсальных текстовых контейнеров — элемент `<p>`, обозначающий абзац.

Далее приведено более подробное описание элементов, которые следует добавить.

`<title>`

Этот элемент задает название вашей Web-страницы. У названия несколько ролей. Во-первых, Web-браузеры отображают название в верхней строке окна браузера. Во-вторых, когда Web-посетитель делает закладку на вашей странице, браузер использует название для обозначения закладки в меню **Bookmarks** (Закладки) (или **Favorites** (Избранное)). В-третьих, когда ваша страница попадает в поиск, поисковая машина, как правило, отображает в первой строке результатов название, за которым следует фрагмент информационного наполнения страницы.

`<p>`

Этот элемент обозначает абзац. Web-браузеры не делают отступов для абзацев, они просто увеличивают расстояние между смежными абзацами, чтобы отделить их друг от друга.

Далее приведен HTML-код страницы с этими двумя новыми элементами.

```
<!DOCTYPE html>

<html>
```

```
<head>
<title>Everything I Know About Web Design</title>
</head>

<body>
<p></p>
</body>

</html>
```

Если вы откроете этот документ в Web-браузере, то увидите, что страница пуста, но название у нее появилось (рис. 2.7).

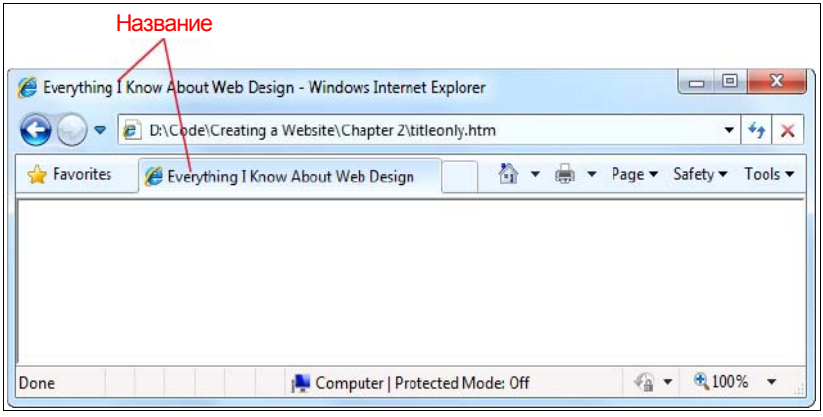


Рис. 2.7. Когда браузер отображает Web-страницу, он выводит в верхней строке своего окна ее название и короткий дополнительный текст в конце. Если в вашем браузере для отображения применяются вкладки (как в Internet Explorer), название также появится на вкладке

Приведенный HTML-документ, что понятно уже сейчас, — отличный шаблон для будущих страниц. Базовая структура уже есть и вам остается только изменить название и включить какой-то текст.

Вставка контента

Вы полностью подготовились для превращения этой HTML-заготовки в реальный документ с помощью информационного наполнения (или контента). Именно этот компонент самый важный и именно этой задаче посвящена бóльшая часть книги.

Предположим, что вы пишете простую страницу с резюме. Далее показан первый этап работы на ней.

```
<!DOCTYPE html>

<html>

<head>
<title>Hire Me!</title>
</head>
```

```
<body>
<p>I am Lee Park. Hire me for your company, because my work is <b>off
the hizzle</b>.</p>
</body>

</html>
```

В примере выделены (жирным шрифтом) изменения, внесенные в базовую HTML-структуру — другое название и одна строка текста. На странице внутри абзаца использован только один элемент `<b>` для того, чтобы чуть-чуть приукрасить страницу. Перед тем как двигаться дальше, вы, возможно, захотите создать этот файл-пример в вашем текстовом редакторе (используя знания, полученные из *разд. "Создание HTML-файла"* данной главы) и открыть ее в вашем любимом Web-браузере (рис. 2.8).

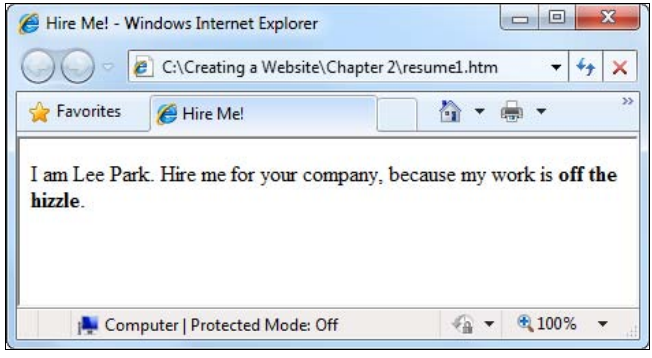


Рис. 2.8. Добро пожаловать во Всемирную паутину! На этой странице совсем немного HTML-штучек (и она вряд ли поможет Ли устроиться на работу), но это одна из простейших HTML-страниц, которые вы можете создать

С помощью HTML-хитростей, описанных в следующих разделах, вы сможете усовершенствовать этот пример и предложить Ли Парку лучшее резюме.

> ***Совет***
>
> Даже если у вас есть мощный HTML-редактор, например Dreamweaver, пока не применяйте его. Если начали изучать HTML, старайтесь все делать вручную, так вы поймете каждую деталь на вашей Web-странице. Позже, когда освоите азы и сможете создавать более сложные Web-страницы, возможно, появится желание перейти к другим программным средствам, обсуждаемым в *главе 4*.

Структурирование текста

Начав создавать более сложные Web-страницы, вы очень быстро обнаружите, что разработка страницы не так проста, как, скажем, создание страницы в текстовом процессоре Microsoft Word. Например, вы решите улучшить страницу резюме, создав перечень своих профессиональных навыков. Далее приведена приемлемая первая попытка.

```
<!DOCTYPE html>

<html>

<head>
<title>Hire Me!</title>
</head>

<body>
<p>I am Lee Park. Hire me for your company, because my work is <b>off
the hizzle</b>.</p>

My skills include:
*Fast typing (nearly 12 words/minute).
*Extraordinary pencil sharpening.
*Inventive excuse making.
*Negotiating with officers of the peace.</p>
</body>

</html>
```

Проблема возникнет, когда вы откроете этот, казалось бы, безобидный документ в вашем Web-браузере (рис. 2.9).

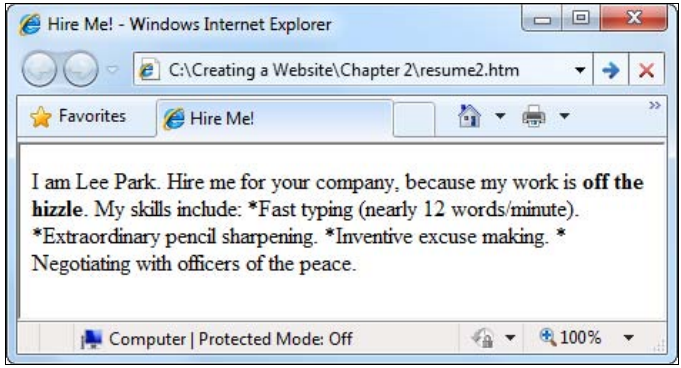


Рис. 2.9. HTML игнорирует переносы строк и следующие друг за другом пробелы, поэтому текст, аккуратно организованный в вашем HTML-файле, может превратиться в свалку при отображении в окне браузера

Все дело в том, что HTML игнорирует дополнительные пропуски. К ним относятся табуляции, переносы строк и дополнительные пробелы (любое количество следующих подряд пробелов). Когда это случится впервые, вы, возможно, оторопело уставитесь на экран и будете недоуменно размышлять о том, почему Web-браузеры устроены таким образом. Но вы поймете, что в этом есть смысл, когда вспомните, что HTML должен работать как *универсальный стандарт*.

Скажем, вы настроили гипотетическую Web-страницу с нужными пропусками, отступами и шириной строк для вывода на монитор *вашего* компьютера. Но на дру-

гом мониторе эта страница может выглядеть не так хорошо. Например, часть текста может уйти за правый край экрана и это затруднит чтение. А разные мониторы — это только часть проблемы. Сегодня Web-страницы должны отображаться на *устройствах* разных типов. Будущий босс Ли Парка может просматривать резюме Парка на чем-нибудь вроде современнейшего широкоэкранного ноутбука, планшетного компьютера или смартфона.

Для того чтобы справиться с таким диапазоном устройств вывода, HTML содержит элементы, определяющие *структуру* вашего документа. Вместо команды браузеру: "Здесь перейди на новую строку, а здесь вставь четыре дополнительных пробела", HTML извещает браузер: "Здесь есть два абзаца и один маркированный список". Именно браузер отвечает за отображение страницы с помощью инструкций, которые вы включили в ваш HTML-код.

Для исправления примера с резюме вам понадобятся дополнительные элементы абзацев и два новых элемента-контейнера.

`<ul>`

Этот элемент обозначает начало маркированного списка. Такой список — наилучший вариант для перечисления навыков Ли.

`<li>`

Этот элемент обозначает отдельный пункт маркированного списка. Ваш браузер создает отступ для каждого пункта списка, а для предложений, не помещающихся на одной строке, создает такие отступы для строк-продолжений, что они выравниваются относительно первой строки. Кроме того, перед каждым пунктом вставляется маркер (•). Элемент `<li>` можно применять только внутри элемента списка, подобного `<ul>`. Другими словами, любой пункт списка должен быть частью списка.

Далее приведена исправленная Web-страница (рис. 2.10) со структурными элементами, выделенными жирным шрифтом.

```
<!DOCTYPE html>

<html>

<head>
<title>Hire Me!</title>
</head>

<body>
<p>I am Lee Park. Hire me for your company, because my work is <b>off the
hizzle</b>.</p>
<p>My skills include:</p>
<ul>
  <li>Fast typing (nearly 12 words/minute).</li>
  <li>Extraordinary pencil sharpening.</li>
  <li>Inventive excuse making.</li>
```

```
  <li>Negotiating with officers of the peace.</li>
</ul>
</body>

</html>
```



Рис. 2.10. При наличии нужных элементов (как показано в коде примера) браузер понимает структуру вашего HTML-документа и знает, как его отображать

Нежелание браузера обращать внимание на переносы строк можно обратить себе на пользу. Для того чтобы облегчить чтение ваших HTML-документов, вставляйте переносы строк и пробелы в любое место. Web-специалисты часто пользуются отступами для наглядного представления вложенных элементов. В примере с резюме вы можете видеть этот прием в действии. Обратите внимание на то, с каким отступом располагаются пункты списка (строки, начинающиеся с тега `<li>`). На отображение в браузере это не влияет, но помогает лучше увидеть структуру HTML-документа и проверить, как браузер ее отобразит.

На рис. 2.11 HTML-документ изображен в виде древовидной структуры. С помощью этой схемы удобно знакомиться с анатомией Web-страницы, поскольку она дает общее представление обо всей структуре целиком. Но когда ваши Web-страницы станут достаточно сложными, построение древовидной структуры может превратиться в излишне трудоемкую задачу.

Если вы — мазохист, можете не использовать пробелы вообще. Предыдущий пример — точный эквивалент следующего, гораздо менее наглядного HTML-документа.

```
<!DOCTYPE html><html><head><title>Hire Me!</title></head><body><p>I am
Lee Park. Hire me for your company, because my work is <b>off the hizzle
</b>.</p><p>My skills include:</p><ul><li>Fast typing (nearly 12
words/minute).</li><li> Extraordinary pencil harpening.</li><li>Inventive
```

```
excuse making.</li><li> Negotiating with officers of the peace.</li></ul>
</body></html>
```

Понятно, что человек не может в таком виде написать HTML-код без ошибок.

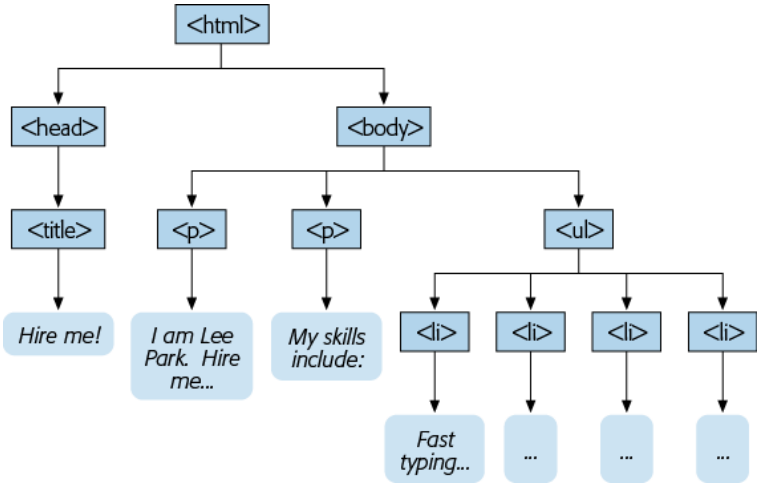


Рис. 2.11. Это другой способ представления созданного вами HTML-документа. На древовидной структуре показано, как вложены HTML-элементы. Двигаясь по стрелкам, вы увидите, что элемент верхнего уровня `<html>` содержит элементы `<head>` и `<body>`. Внутри элемента `<head>` находится элемент `<title>`, а внутри элемента `<body>` есть два абзаца и маркированный список из четырех пунктов. Если присмотреться к древовидной структуре повнимательней, можно понять, почему в языке HTML все эти элементы — контейнеры

СЕКРЕТЫ РЕМЕСЛА

Есть что скрывать?

Во время работы над сложной Web-страницей может возникнуть желание удалить на время элемент или раздел контента. Это очень удобный прием, если страница отображается не совсем правильно, и вы хотите выяснить, какие элементы создают проблемы. Во-первых, это можно сделать с помощью старого доброго инструмента "вырезать-и-вставить" в вашем текстовом редакторе. Вырежьте раздел, который по вашему мнению содержит ошибки, сохраните файл и откройте его в вашем браузере. Если раздел не виноват, верните его на место, а затем снова сохраните файл. Повторяйте этот процесс до тех пор, пока не найдете виновника.

Но язык HTML предлагает более простое решение — *комментарии*. С их помощью вы можете оставить всю страницу без изменений. Когда вы включаете в комментарий раздел страницы, HTML ее игнорирует.

HTML-комментарий создается с помощью последовательности символов `<!` для обозначения начала комментария и `-->` для обозначения его конца. Ваш браузер будет игнорировать все, что находится между этими двумя метками, будь то контент или теги. Метки комментария могут помещаться на одной строке или использоваться для скрытия целого раздела вашего HTML-документа. Но не пытайтесь вложить один комментарий в другой, т. к. подобная схема не сработает.

Далее приведен пример, скрывающий два пункта списка. Когда вы откроете ваш Web-браузер, в вашем списке останется только два последних пункта: "Inventive excuse

making" (придумывать изобретательные предлоги и отговорки) и "Negotiating with officers of the peace" (договариваться с представителями сил охраны правопорядка).

```
<ul>
<!  <li>Fast typing (nearly 12 words/minute).</li>
  <li>Extraordinary pencil sharpening.</li>
-->
  <li>Inventive excuse making.</li>
  <li>Negotiating with officers of the peace.</li>
</ul>
```

Когда решите вернуть списку первоначальное великолепие, просто удалите метки комментария.

А где же все картинки?

Web-мир был бы очень унылым местом без картинок, будь то график состояния запасов, эмблема вашей неформальной любительской группы или подретушированное фото любимой знаменитости. До сих пор вы видели, как в HTML-документ вставляется текст, а что делать, если вам нужно изображение?

Как это ни странно, хранить изображения внутри HTML-файла нельзя. Для этого есть множество веских причин — файлы ваших Web-страниц стали бы очень большими, было бы трудно изменять изображения или делать что-то другое с этими файлами и вы бы прокляли все на свете, редактируя такие страницы в текстовом редакторе, т. к. графические данные создали бы мешанину в HTML-документах. Выход — хранить ваши картинки, как отдельные файлы, и ссылаться на них в вашем HTML-документе. В этом случае браузер извлечет картинки и поместит их на вашу Web-страницу именно туда, куда вы хотите.

Реализует этот прием элемент `<img>` (сокращение для слова "image"). Он указывает на файл изображения, который браузер считывает и вставляет на страницу. Вы можете поместить файл изображения в ту же папку, где хранится ваша Web-страница (это самое легкое решение), или на совершенно другой Web-сайт.

Несмотря на то, что вы узнаете все о Web-графике в *главе 7*, прямо сейчас стоит рассмотреть простейший пример. Для того чтобы перейти к нему, под руками должно быть готовое Web-изображение. (Самые распространенные типы поддерживаемых графических файлов — JPEG, GIF и PNG.) Если вы загрузили из Интернета сопроводительные материалы к книге (со страницы **Missing CD** по адресу **www.missingmanuals.com/cds/caw3**), можно использовать файл leepark.jpg. Если вы поместили этот файл в ту же папку, что файл вашей Web-страницы, отобразить картинку можно с помощью следующего элемента `<img>`.

```
<img src="leepark.jpg" alt="Lee Park Portrait" />
```

Как и элемент `<br>`, элемент `<img>` — непарный или элемент без содержимого. Поэтому есть смысл воспользоваться синтаксисом пустого элемента и вставить прямой слэш перед закрывающей угловой скобкой.

Но между элементами `<br>` и `<img>` есть заметная разница. Хотя элемент `<img>` и пустой элемент, он несамодостаточен. Для того чтобы этот элемент что-то означал,

необходимо задать два вида данных: имя файла с изображением и какой-либо замещающий текст, применяемый в тех случаях, когда ваш браузер не может загрузить или вывести на экран изображение (см. *разд. "Альтернативный текст" главы 7*). Для включения этой дополнительной информации в элемент язык HTML использует *атрибуты*. Атрибуты — это дополнительные данные, вводимые после имени элемента, но перед закрывающей скобкой >.

Пример элемента <img> включает два атрибута, разделенных пробелом. У каждого из них два компонента: имя (сообщающее браузеру назначение атрибута) и значение (меняющиеся данные, которые задаете вы). Имя первого атрибута элемента <img> — src, сокращение от слова "source" (источник); он сообщает браузеру, где взять нужное вам изображение. В данном примере значение атрибута src — leepark.jpg, имя файла с фотографией Ли Парка.

Имя второго атрибута элемента <img> — alt, сокращение от "alternate text" (альтернативный или замещающий текст), он сообщает браузеру о том, что в случае невозможности загрузки изображения вы хотите, чтобы он вывел текст. Значение этого атрибута — тот самый текст, который вы хотите отобразить на экране, в данном примере это "Lee Park Portrait" (Портрет Ли Парка).

После подробного разбора элемента <img> вы можете использовать его в вашем HTML-документе. Просто поместите его внутрь имеющегося абзаца, туда, где он должен находиться по смыслу.

```
<!DOCTYPE html>

<html>

<head>
<title>Hire Me!</title>
</head>
<body>
<p>I am Lee Park. Hire me for your company, because my work is <b>off the
hizzle</b>.
<img src="leepark.jpg" alt="Lee Park Portrait" />
</p>
<p>My skills include:</p>
<ul>
  <li>Fast typing (nearly 12 words/minute).</li>
  <li>Extraordinary pencil sharpening.</li>
  <li>Inventive excuse making.</li>
  <li>Negotiating with officers of the peace.</li>
</ul>
</body>

</html>
```

На рис. 2.12 показано, где именно обоснуется картинка.

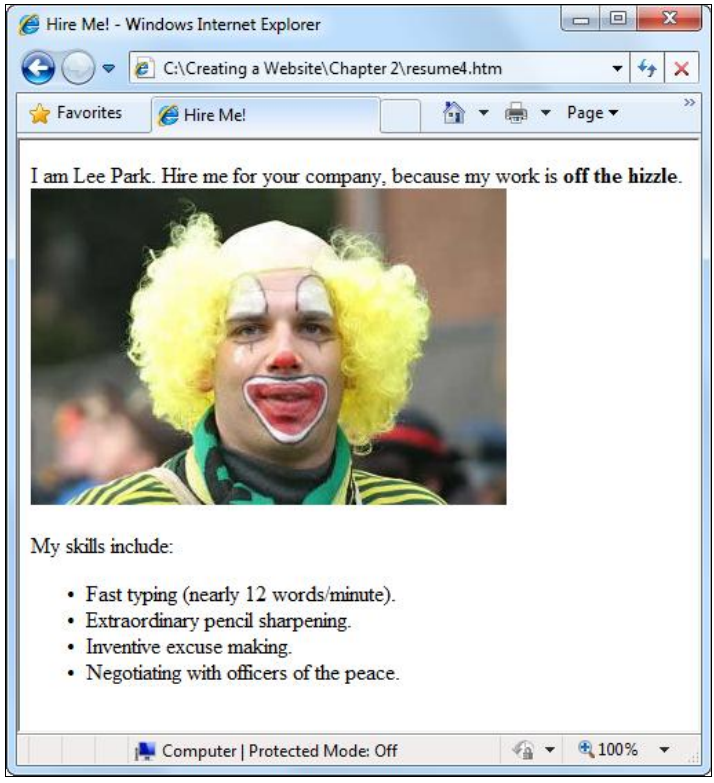


Рис. 2.12. Это Web-страница, в которую встроена картинка благодаря способности элемента `<img>` устанавливать связь. Для отображения этого документа браузер выполняет отдельный запрос на считывание файла с изображением. В итоге ваш браузер может вывести текст Web-страницы до того, как загрузит графический файл — все зависит от времени загрузки (как правило, это доли секунды)

Примечание

В *главе 7* вы узнаете гораздо больше приемов и хитростей для Web-графики, включая способы изменения размера картинок и обтекания текста вокруг них.

10 самых важных элементов (и несколько дополнительных)

В данный момент вы усвоили достаточно для того, чтобы создать базовый HTML-документ, и уже имеете в запасе несколько элементов. Вы знаете основы — осталось расширить ваши знания за счет изучения дополнительных элементов.

В языке HTML довольно ограниченный набор элементов. Регулярно вы будете использовать меньше 25 из них. В этом и кроется основное достоинство языка, делая его популярным, общедоступным и легко принимаемым всеми.

Примечание

В HTML-документе нельзя определять собственные элементы и затем применять их, т. к. Web-браузеры не будут знать, как их интерпретировать.

Некоторые элементы, например элемент <p>, задающий формат абзаца, очень важны для разметки страницы в целом. Их называют *элементами блочного уровня* или *блочными элементами*. Элементы блочного уровня можно помещать прямо в раздел вашей Web-страницы <body> или внутрь другого блочного элемента. В табл. 2.1 приведен краткий обзор ряда самых основных элементов блочного уровня, причем некоторые вы уже видели. В ней также указано, какие из элементов являются контейнерами, а какие — пустыми элементами (как написано в *разд. "Представление об элементах"*, элементам-контейнерам требуются начальный и конечный теги, а пустые обходятся только одним тегом).

Таблица 2.1. *Элементы блочного уровня*

Элемент	Название	Тип	Описание
<p>	Абзац	Контейнер	Абзац — основной блок структурирования текста. Если вы используете несколько элементов абзаца в одной строке, браузер вставляет определенный объем свободного пространства между ними, немного больший, чем обычная пустая строка. Подробности см. в *главе 5*
<h1>, <h2>, <h3>, <h4>, <h5>, <h6>	Заголовки	Контейнер	Заголовки помогают структурировать вашу Web-страницу и выделять названия. Они отображают текст большими буквами и жирным шрифтом. Чем меньше номер заголовка, тем большего размера его текст, так <h1> задает самый крупный заголовок. По мере того как вы дойдете до уровня <h5>, размер уменьшится до размера обычного текста, а <h6> выводит текст жирным шрифтом, но его размер меньше обычного текста
<hr>	Горизонтальная линия	Пустой элемент	Горизонтальная линия поможет отделить один раздел вашей Web-страницы от другого. Линия автоматически принимает размер, равный ширине окна браузера. (Или, если вы помещаете линию в другой элемент, например ячейку таблицы, становится равна ширине своего контейнера)
,	Маркированный список, пункт списка	Контейнер	Эти элементы позволяют формировать стандартные маркированные списки. Браузер автоматически помещает каждый пункт списка на отдельной строке и создает нужный отступ для каждого. Для быстрого внесения изменений можно заменить элемент элементом и получить вместо маркированного списка автоматически нумерованный список (ol сокращение от "ordered list" (нумерованный список))

Остальные элементы предназначены для работы с более мелкими структурными деталями — например, фрагментами текста с жирным шрифтом или курсивным начертанием, переносами строк, ссылками на другие Web-страницы и изображе-

ниями. Эти элементы называют *строчными элементами* (inline elements). Строчные элементы нельзя поместить непосредственно в элемент <body>. Вместо этого вы должны вложить их *внутрь* элемента блочного уровня. В табл. 2.2 перечислены самые часто используемые строчные элементы.

***Таблица 2.2.** Основные строчные элементы*

Элемент	Название	Тип	Описание
<b>, <i>	Жирный шрифт и курсив	Контейнер	Эти два элемента задают начертание шрифта: жирное или курсивное. (Строго говоря, <i> означает "текст другим голосом", а <b> — "стилистически выделенный текст" и, как вы увидите в *главе 6*, существуют приемы для изменения задаваемого ими форматирования. Но в действительности почти все Web-разработчики считают, что <i> означает курсив, а <b> — жирный шрифт)
 	Перенос строки	Пустой	Порой вам нужен просто текст, разделенный переносами строки, а не элементами абзаца. Это позволяет располагать смежные строки текста ближе друг к другу. О верстке текста вы узнаете больше в *главе 5*
<img>	Изображение	Пустой	Применяйте этот элемент для вывода изображения внутри вашей Web-страницы. Не забудьте указать атрибут src для указания имени файла, содержащего изображение, которое должен вывести браузер
<a>	Якорь	Контейнер	Элемент-якорь — отправная точка для создания гиперссылок, позволяющих посетителям Web-сайта переходить с одной страницы на другую. Вы узнаете об этом незаменимом элементе в *главе 8*

Для того чтобы сделать пример с резюме более представительным, воспользуйтесь несколькими элементами из табл. 2.1 и 2.2.

```
<!DOCTYPE html>

<html>

<head>
  <title>Hire Me!</title>
</head>

<body>
  <h1>Hire Me!</h1>
  <p>I am Lee Park. Hire me for your company, because my work is <b>off the
hizzle</b>. As proof of my staggering computer skills and monumental work
ethic, please enjoy this electronic resume.</p>
```

```
  <h2>Indispensable Skills</h2>
  <p>My skills include:</p>
  <ul>
    <li>Fast typing (nearly 12 words/minute).</li>
    <li>Extraordinary pencil sharpening.</li>
    <li>Inventive excuse making.</li>
    <li>Negotiating with officers of the peace.</li>
  </ul>
  <p>And I also know XHTML!</p>

  <h2>Previous Work Experience</h2>
  <p>I have had a long and illustrious career in a variety of trades. Here are
some highlights:</p>
  <ul>
    <li>2005-2008 — Worked as a typist at <i>Flying Fingers</i></li>
    <li>2008-2009 — Performed cutting-edge web design at <i>Riverdale
     Farm</i></li>
    <li>2009-2011 — Starred in Chapter 2 of <i>Creating a Website: The
     Missing Manual</i></li>
  </ul>

  <hr />
</body>

</html>
```

На рис. 2.13 показана эта обновленная версия Web-страницы.

СЕМИНАР ДЛЯ ОПЫТНЫХ ПОЛЬЗОВАТЕЛЕЙ

Плагины: выход за пределы HTML

Создатели языка HTML разрабатывали его преимущественно для размещения научных статей и других редко меняющихся документов в Web. Они не предвидели интернет-аукционов, магазинов электронной коммерции и браузерных игр. Для этой работы профессиональные программисты приспособили слегка причудливый язык JavaScript (см. *главу 15*), который действует рука об руку с HTML. JavaScript очень подходит для выполнения таких повседневных задач, как сбор пользовательской информации и отправка ее на Web-сервер или обновление информации на Web-странице. Но решать более сложные задачи, например, предоставить интерактивную карту земного шара или 3D-модель изделия, крайне трудно или просто невозможно (даже если вы работаете в Google).

Для преодоления ограничений, подобных названным, высокоразвитые технологические компании (например, Microsoft, Google, Apple, Adobe и многие другие) создали *плагины* для браузеров. Плагин — это маленькая программка, выполняющаяся прямо внутри вашего Web-браузера. Например, если у вас есть плагин Adobe PDF, вы можете просматривать документ в формате Adobe PDF в окне вашего браузера. Более впечатляющие плагины, такие как Microsoft Silverlight и Adobe Flash, позволяют вашему браузеру выполнять мощные Web-приложения, отображающие анимацию (например, в рекламных объявлениях), проигрывать видеоролики (как в YouTube) и даже создавать интерактивные видеоигры (примеры некоторых вызывающих психологическую зависимость Flash-игр можно найти на странице **www.ferryhalim.com/orisinal**).

Вы всегда можете определить, работает ли что-либо отличное от HTML, щелкнув правой кнопкой мыши подозрительную область контента на странице. Например, щелчок правой кнопкой мыши Flash-зоны выводит на экран раскрывающееся меню с вариантами настройки Flash или получения дополнительной информации. Если у вас нет Flash-плагина, но вашей странице он необходим, на странице появится предложение загрузить самую свежую версию плагина (в *главе 17* вы примените Flash для размещения превосходного музыкального проигрывателя на вашей Web-странице).

Несмотря на то, что плагины впечатляют и способны делать гораздо больше, чем простой HTML, они сталкиваются с проблемами совместимости, особенно в столь быстро растущем мире всевозможных устройств, подключаемых к Web. Например, несмотря на то, что Flash — самый популярный из существующих плагинов (установленный на 99% компьютеров, блуждающих в Интернете), iPad, iPhone и iPod Touch компании Apple до сих пор его не поддерживают.

Подобные проблемы — одна из причин, по которой Web-разработчики и компании, выпускающие браузеры, с нетерпением ожидают HTML5 и новый набор стандартов для Web-приложений. Хотя многие из этих разрабатываемых технологий появятся нескоро, со временем они позволят профессиональным разработчикам Web-сайтов создавать мощные Web-приложения, не выходя за пределы HTML.



Рис. 2.13. Снабженный дополнительными заголовками, списками и горизонтальной чертой этот HTML-документ делает резюме более привлекательным

Поиск ошибок на ваших страницах

Даже Web-разработчик с наилучшими намерениями может написать плохой файл разметки и нарушить правила HTML. И хотя браузеры должны вылавливать эти ошибки, фактически они этого не делают. Вместо этого они изо всех сил стараются игнорировать ошибки и выводить бракованные документы.

На первый взгляд такой подход кажется удачной находкой — в результате он сглаживает любые мелкие оплошности, случайно сделанные вами. Но есть и темная сторона у терпимости к ошибкам. В частности, такое поведение позволяет серьезным ошибкам на ваших Web-страницах остаться незамеченными. Что значит серьезная ошибка? Это проблема, не возникающая при просмотре страницы в вашем любимом браузере, но приводящая в замешательство при отображении страницы в другом браузере; ошибка, которая не обнаруживается, пока вы не отредактируете код, и которая проявляется случайно при просмотре страницы в браузере после ее корректировки; или ошибка, не влияющая на вывод страницы, но мешающая автоматическим средствам (например, поисковой машине) считывать страницу.

К счастью, есть способ выявлять такие проблемы. Вы можете воспользоваться средством валидации (проверки правильности), которое прочтет вашу Web-страницу и проверит ее разметку. Если вы применяете профессиональное программное обеспечение для Web-дизайна, например Dreamweaver, можно воспользоваться встроенным средством поиска ошибок (подробности см. в *главе 4*). Если же вы создаете страницу вручную, в текстовом редакторе можно применить бесплатное интерактивное средство валидации.

Далее перечислены некоторые возможные проблемы, которые валидатор сможет обнаружить:

- пропущенные обязательные элементы (например, элемент `<title>`);
- начальный тег без соответствующего конечного тега;
- некорректно вложенные теги;
- теги с пропущенными атрибутами (например, элемент `<img>` без атрибута `src`);
- элементы или контент в неподходящем месте (например, текст, помещенный непосредственно в раздел `<head>`).

В Интернете можно найти огромное множество средств валидации. Далее написано, как применять валидатор, предлагаемый организацией по разработке стандартов W3C.

1. Убедитесь, что у вашего документа правильный `doctype`.

   Ваш `doctype` сообщает валидатору, какие правила применять при проверке правильности вашего документа. Например, если вы используете `doctype` для HTML5 (см. *разд. "Определение типа документа"* этой главы), валидатор проверит ваш документ на соответствие правилам стандарта HTML5 и сочтет правильными некоторые свойства, неприемлемые в более старом стандарте XHTML 1.0.

ПРИМЕЧАНИЕ

Поскольку стандарт HTML5 все еще совершенствуется, валидаторы HTML5 еще не так отшлифованы, как их варианты для XHTML. Например, во время написания книги валидатор для HTML5 консорциума W3C рассматривался как "опытный образец".

2. В вашем Web-браузере перейдите на сайт **http://validator.w3.org** (рис. 2.14).

   Валидатор W3C предлагает три варианта, представленные на трех отдельных вкладках: **Validate by URI** (Проверить URI) — для страницы, которая уже размещена в Web; **Validate by File Upload** (Проверить загруженный файл) — для страницы, хранящейся в файле на вашем компьютере; **Validate by Direct Input** (Проверить набранный текст) — для разметки, которую вы вводите непосредственно в соответствующее поле.

3. Выберите нужную вкладку и задайте ваш HTML-контент.

   Вариант **Validate by URI** (Проверить URI) позволяет проверить имеющуюся Web-страницу. Просто введите ее URL в строку адреса (например, **www.MySloppySite.com/Flawed Page.html**).

   Вариант **Validate by File Upload** (Проверить загруженный файл) позволяет загрузить любой файл с вашего компьютера. Сначала щелкните мышью кнопку **Browse** (Обзор) (которая в разных браузерах называется по-разному, например, в Chrome это **Choose File** (Выберите файл)), чтобы вызвать на экран диалоговое окно **Open** (Открыть). Затем найдите папку с вашим HTML-файлом и щелкните мышью кнопку **Open** (Открыть).

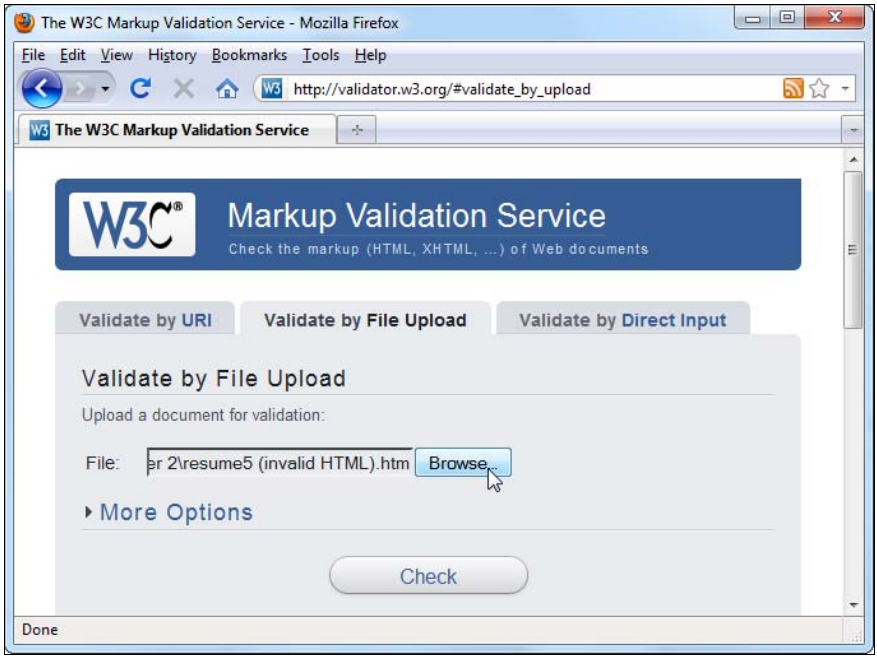


Рис. 2.14. Web-сайт **http://validator.w3.org** предлагает вам три варианта для проверки корректности HTML-кода. Вы можете ввести адрес другой Web-страницы, загрузить свой файл (этот вариант показан) или непосредственно ввести разметку

Вариант **Validate by Direct Input** (Проверить набранный текст) позволяет проверить любую разметку — вам нужно ввести ее в специальное поле. Самый легкий способ для этого варианта — скопировать разметку из вашего текстового редактора и вставить ее в специальное поле на странице валидации W3C.

Перед тем как продолжить, можно щелкнуть мышью кнопку **More Options** (Дополнительные параметры), чтобы задать другие параметры, но, возможно, вы и не захотите этого делать. Лучше дать валидатору автоматически определить тип документа, в этом случае он воспользуется `doctype`, заданным на вашей Web-странице. Точно так же, если в документе используется английский язык и у валидатора не будет проблем при определении подходящего набора символов, разрешите ему автоматически определить набор символов.

4. Щелкните мышью кнопку **Check** (Проверить).

Ваша HTML-страница будет переслана валидатору W3C. После короткой задержки валидатор сообщит, прошел проверку ваш документ или нет, и какие ошибки обнаружил валидатор в случае неудачи (рис. 2.15).

Для полностью корректного HTML-документа валидатор может вывести несколько безобидных предупреждений, включая сообщение об автоматическом определении набора символов и предупреждение о том, что служба валидации для HTML5 считается опытным образцом, и работа над этим средством не завершена.

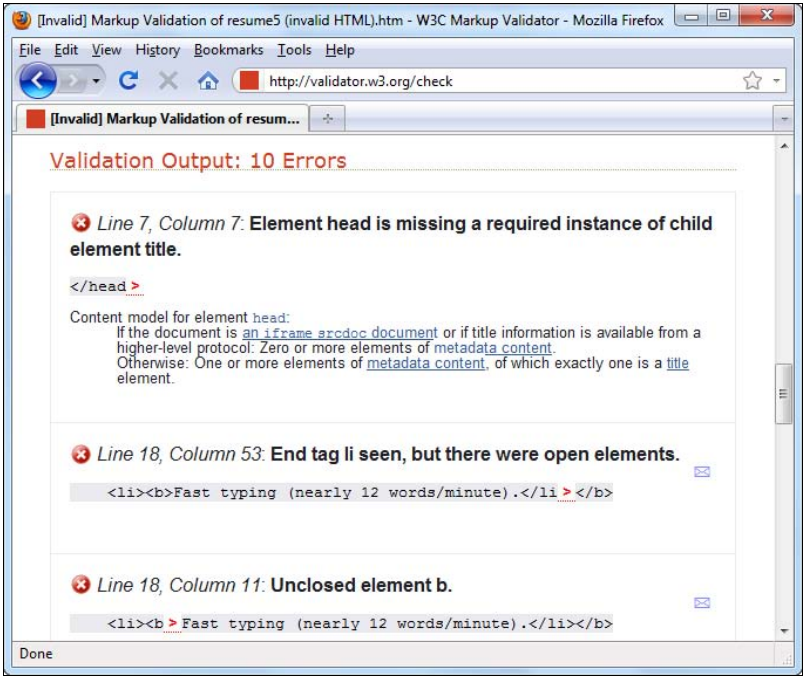


Рис. 2.15. В этом файле валидатор обнаружил 10 ошибок, порожденных двумя оплошностями в коде. Во-первых, на странице пропущен обязательный элемент `<title>`. Во-вторых, элемент `<li>` закрыт прежде, чем закрыт вложенный в него элемент `<b>`. (Для решения этой проблемы вам следует заменить порядок следования `</li></b>` на `</b></li>`.) Тем не менее этот файл довольно близок к корректному, так что браузеры могут отобразить его правильно

ГЛАВА 3

Размещение вашей страницы в Web

В предыдущей главе вы познакомились с основами HTML и создали простое одностраничное резюме. Вы можете сделать гораздо больше для улучшения этой страницы, но прежде чем двигаться дальше, остановимся на одном из важнейших фрагментов Web-мозаики: превращении ваших страниц в интерактивные.

В этой главе вы узнаете, как работают Web-серверы, и как вы можете заставить их работать на вас. Вооружившись этими знаниями, вы сможете начать поиск своего *Web-хоста*, компании, которая разрешит вам поместить ваш сайт на ее Web-сервере. Вам нужно всего лишь сформулировать свои требования, посмотреть хосты, предлагающие то, что вам нужно, и сравнить цены.

Как работает Web-хостинг

Как вы уже знаете из *главы 1,* Web не хранится на одном компьютере и не принадлежит никакой компании. Напротив, его отдельные фрагменты (Web-сайты) разбросаны по миллионам компьютеров (Web-серверов). И хотя кажется, что все они — часть единой вычислительной среды, это всего лишь хитроумная иллюзия. В действительности Интернет — это просто набор стандартов, который позволяет независимым компьютерам общаться друг с другом.

Итак, как же ваш любимый браузер перемещается по этой запутанной сети компьютеров в поисках именно той Web-страницы, которая нужна вам? Все дело в URL (Uniform Resource Locator, Унифицированный указатель ресурсов), или адресе Web-сайта, который вы вводите в ваш браузер, например, **www.google.com**.

Из чего состоит URL

URL состоит из нескольких компонентов. Некоторые из них необязательные, т. к. браузер или Web-сервер могут заполнить отдельные пропуски автоматически. Другие же обязательны. На рис. 3.1 разобран на компоненты URL **http://www.SellMyJunkForMillions.com/Buyers/listings.htm**.

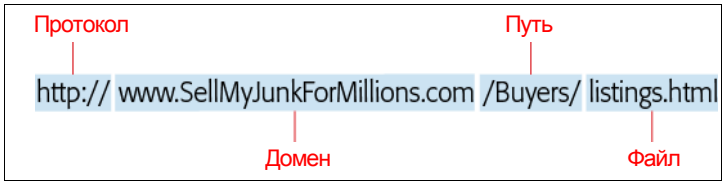


Рис. 3.1. Обычный URL состоит из четырех порций информации. Первая часть (протокол) задает Web-стандарты, которым следует страница. Вторая часть (домен) обозначает Web-сервер, с которым вы связываетесь. В третьей и четвертой частях указаны путь к странице и имя файла на означенном сервере

В Web-адресах, в одной текстовой строке упакован большой объем информации.

- *Протокол* указывает на выбранный вами способ передачи — другими словами, как ваш браузер должен взаимодействовать с Web-сервером. Web-сайты всегда используют HTTP (HyperText Transport Protocol, Гипертекстовый транспортный протокол), т. е. часть URL Web-сайта, относящаяся к протоколу, всегда будет **http://** или **https://**. (Последний вариант устанавливает сверхзащищенное соединение по протоколу http, которое шифрует важную для вас информацию, такую как номера кредитных карт или пароли.) В большинстве браузеров вы можете не вводить эту часть URL. Например, когда вы вводите **www.google.com**, ваш браузер автоматически преобразует его в полный URL, **http://www.google.com**.

  Примечание

  Несмотря на то, что **http://** — это способ перемещения в Web, для других задач вы можете применять другие протоколы, но это зависит от вашего браузера. Общими примерами могут служить **ftp://** (File Transfer Protocol, Протокол передачи файлов) для загрузки в Интернет и скачивания из Интернета файлов и **file:///** для извлечения файла непосредственно с жесткого диска вашего собственного компьютера.

- Имя *домена* обозначает сервер, хранящий сайт, который вы хотите увидеть. В соответствии с принятым соглашением имена серверов обычно начинаются с **www** для обозначения их как серверов World Wide Web (Всемирной паутины). Кроме того, как вы узнаете чуть позже из этой главы, выглядящие дружелюбно имена доменов, такие как **www.google.com** или **www.microsoft.com**, на самом деле просто заменители того, что действительно нужно вашему браузеру для определения местонахождения сервера, а именно числового адреса (см. следующий *разд. "Как браузеры анализируют URL"*).

- *Путь* содержит имя папки, в которой сервер хранит конкретную Web-страницу, которую вы ищете. У этого компонента URL может быть столько уровней вложенности, сколько потребуется. Например, путь /MyFiles/Sales/2011/ указывает на папку MyFiles, которая содержит папку Sales, содержащую в свою очередь папку с именем 2011. Поклонники Windows, обратите внимание — слэши в этом компоненте URL прямые, а не обратные, как в полных именах файлов (например, c:\MyFiles\Current). Это правило соответствует заданию пути к файлу или полного имени файла, используемого на компьютерах под управлением ОС Unix, которые были первыми машинами, предоставлявшими услуги хостинга. Это же правило применяется в современных операционных системах Macintosh.

ПРИМЕЧАНИЕ

Многие браузеры достаточно сообразительны и самостоятельно исправляют неправильный тип слэшей. Но на это не следует полагаться, потому что подобная леность может испортить созданные вами Web-страницы.

- *Имя файла* — последний компонент пути, задающий конкретную Web-страницу, которая вам нужна. Часто его можно распознать по расширению файла htm или html (и то, и другое сокращение от HTML).

ПРИМЕЧАНИЕ

Web-страницы часто заканчиваются расширением htm или html, но это необязательно. Например, если вы смотрите на URL и видите странное расширение blackpudding, с вероятностью 50%, вы все же смотрите на HTML-документ. В большинстве случаев браузеры игнорируют расширение до тех пор, пока могут интерпретировать информацию, содержащуюся в файле. Просто сохраняйте благоразумие — это единственное правило, которое не следует нарушать.

- *Фрагмент* — необязательная часть URL, обозначающая определенную область на Web-странице. Фрагмент легко узнать, потому что он всегда начинается со знака решетки (#) и указывается после имени файла. Например, в URL **http://www.LousyDeals.com/index.html#New** входит фрагмент **#New**. Если щелкнуть URL кнопкой мыши, вы попадете в область страницы index.html, на которой создатель страницы поместил закладку **New**. Как вставлять закладки на страницу, вы узнаете в *разд. "Вставка закладок" главы 8*.

- *Строка запроса* — необязательная часть URL, которую некоторые Web-сайты используют для отправки дополнительной информации с одной Web-страницы на другую. Определить строку запроса легко, потому что она начинается с вопросительного знака (?) и указывается после имени файла. Для того чтобы увидеть строку запроса в действии, перейдите на страницу поисковой машины Microsoft **www.bing.com** и выполните поиск строки "pet platypus" (ручной утконос). Когда вы щелкнете мышью кнопку **Поиск** (Search), вы будете перенаправлены на URL **http://www.bing.com/search?q=pet+platypus&form=QBLH&filt=all&qs=n**. Его непросто анализировать, но найдя вопросительный знак, вы обнаружите, что находитесь на странице с именем "search". Информация справа от вопросительного знака сообщает о том, что выполняется англоязычный поиск страниц с наличием двух ключевых слов "pet" и "platypus" одновременно. Когда вы выполните запрос этого URL, специализированная программа поиска корпорации Microsoft проанализирует данную строку запроса и определит, какой тип поиска ей следует выполнить.

ПРИМЕЧАНИЕ

На ваших собственных Web-страницах вы не будете применять строку запроса, потому что она предназначена для сверхмощных программ, таких как управляющая поиском Bing (или Google). Но разобравшись со строкой запроса, вы составите некоторое представление о том, как работают другие Web-сайты.

Как браузеры анализируют URL

Понятно, что URL концентрирует в одном месте большой объем информации. Но как же браузер использует URL для поиска нужной вам Web-страницы? Для того чтобы понять это, заглянем за кулисы (рис. 3.2).

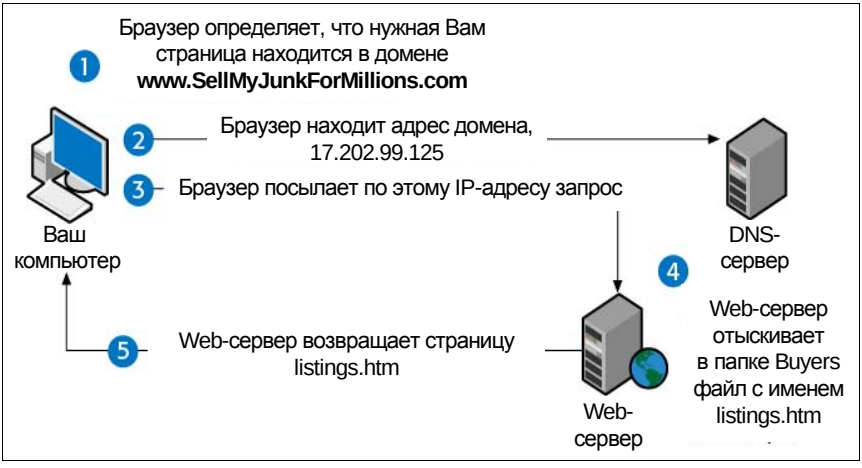


Рис. 3.2. Простой Web-запрос вовлекает в действие группу компьютеров, контактирующих друг с другом в указанном порядке. Первый компьютер (DNS-сервер, номер 2) дает вам крайне важный IP-адрес, позволяющий добраться до второго компьютера (Web-сервера, номер 4), который предоставит вам нужную Web-страницу (номер 5)

Например, вот что произойдет после того, как в строке адреса вы наберете **http://www.SellMyJunkForMillions.com/Buyers/listings.htm** и нажмете клавишу <Enter>.

1. Браузер извлечет из URL имя домена и определит, с каким Web-сервером связаться.

   В этом примере имя домена — **www.SellMyJunkForMillions.com**.

2. Для того чтобы найти сервер с именем **www.SellMyJunkForMillions.com**, браузер преобразует имя домена в понятное компьютеру число, называемое IP-адресом.

   В Web у каждого компьютера, начиная с Web-серверов и заканчивая вашей собственной машиной, есть собственный уникальный IP-адрес. Для поиска IP-адреса сервера браузер ищет доменное имя сервера в гигантском сетевом каталоге, именуемом DNS (Domain Name System, Система доменных имен). IP-адрес выглядит как набор из четырех чисел, разделенных точками. К примеру, у Web-сайта **www.SellMyJunkForMillions.com** может быть IP-адрес 17.202.99.125.

Примечание

Каталог DNS не хранится на вашем компьютере, поэтому ваш браузер получает эту информацию из Интернета. У такого подхода есть преимущества. В обычных обстоятельствах имя домена для электронного бизнеса никогда не меняется; это имя используют и помнят клиенты. Но за кулисами его IP-адрес может измениться, т. к. компания могла перенести свой Web-сайт с одного сервера на другой. Если компания об-

новит DNS, перенос не вызовет никаких нарушений. К счастью, если сайт перемещается на другой сервер, вам не нужно самим заботиться об управлении DNS, поскольку компания, хранящая ваш сайт, автоматически выполнит все необходимое.

3. Браузер отправляет запрос страницы на только что полученный IP-адрес Web-сервера.

   Реальный маршрут запроса трудно предсказать. На пути он может пройти ряд других серверов.

4. Сервер получает запрос и просматривает путь и имя файла в URL.

   В данном случае сервер видит, что запрашивается файл с именем listings.htm из папки Buyers. Сервер находит этот файл и отправляет его обратно запросившему Web-браузеру. Если файла не существует, сервер посылает назад сообщение об ошибке.

5. Браузер получает HTML-страницу, которую он ждал (listings.htm), и затем выводит ее на экран для вашего удовольствия.

   URL **http://www.SellMyJunkForMillions.com/Buyers/listings.htm** — типовой пример. В реальной жизни вы иногда будете встречать URL, которые выглядят гораздо проще. Например, рассмотрим **http://www.amazon.com**. В нем четко задано имя домена (**www.amazon.com**), но не указана информация о пути или имени файла. Что делать Web-серверу в этом случае? Он видит, что вы не запрашиваете конкретный файл, поэтому посылает вам опорную страницу сайта — его страницу, заданную *по умолчанию*, как говорят компьютерные профи, и часто называемую index.htm или index.html, а также известную вам, как домашняя или исходная страница сайта. (Но Web-администратор может сделать любую страницу страницей по умолчанию.)

Теперь, когда вы знаете, как устроены URL, можно встраивать ваши собственные страницы в структуру Web. Первая ваша задача — получить хорошее имя домена.

СЕМИНАР ДЛЯ ОПЫТНЫХ ПОЛЬЗОВАТЕЛЕЙ

Отличие Интернета от интранета

Как вы уже знаете, Интернет — это громадная компьютерная сеть, раскинувшаяся на весь земной шар. *Интранет* намного меньше; это сеть в конкретной компании, организации или доме, объединяющая гораздо меньшее число компьютеров. В действительности интранет может объединять всего лишь два компьютера.

В применении интранета есть смысл, если вы хотите, чтобы Web-сайт был доступен лишь небольшому числу людей, находящихся в одном месте. Например, компания может использовать интранет для предоставления Web-страниц с маркетинговыми данными (или самыми свежими офисными сплетнями). В вашем собственном доме вы можете позволить своим соседям просматривать ваши Web-творения на их компьютерах. Единственное ограничение — Web-сайт в интранете доступен только компьютерам этой сети. Другие Web-посетители не смогут его увидеть.

Настройка Web-сайта для интранета проще, чем для Интернета, поскольку вам не нужно регистрировать имя домена. Вместо него вы можете использовать сетевое имя компьютера. Например, если у вашего компьютера сетевое имя — SuperServer, вы сможете добраться до Web-страницы с помощью, скажем, такого URL: **http://SuperServer/MySite/MyPage.htm**. Конечно, это будет работать, если вы настроите домашнюю сеть (легко) и некоторое серверное программное обеспечение (немного сложнее). Например, на компьютере с ОС Windows вы сможете воспользоваться

службами IIS (Internet Information Services), входящими в состав большинства версий Windows, но программу придется запустить и настроить вручную. Эти задачи не обсуждаются в данной книге. Дополнительную информацию поищите на хорошем сетевом ресурсе.

Имена доменов

Шекспир написал известную фразу "Что в имени? Как розу ни зови, В ней аромат останется все тот же" (Вильям Шекспир "Ромео и Джульетта", перевод Д. Л. Михаловского). Но мог бы посмотреть на вещи иначе, если бы ему пришлось набирать в его браузере **www.thesweetsmellingredflowerwiththorns.biz** вместо **www.rose.com**. Короткие, яркие имена доменов привлекают внимание и хорошо запоминаются. Сегодня любому создателю Web-сайтов доступны дешевые адреса доменов. Если вы решите завести свой собственный домен, не жалейте времени на его выбор.

ПРИМЕЧАНИЕ

Корректные имена доменов включают только буквы, цифры и дефисы.

Выбор удачного имени домена

Вы быстро выясните, что самые короткие и остроумные комбинации слов больше не доступны для выбора в качестве имени домена. Даже если они не используются, *доменные сквоттеры* — предприимчивые дельцы, покупающие и придерживающие популярные имена в надежде продать их позже невероятно богатым соискателям — уже давно предъявили права на эти имена. Откажитесь от имени **www.worldsbestchocolate.com** — оно уже занято. Но вам может повести при выборе немного более длинных или более специфических имен (**www.worldsbestmilkchocolate.com**), используйте названия местности или имена людей (**www.bestvermontchocolate.com** или **www.anniesbestchocolate.com**) или вставляйте искусственно составленные слова (**www.chocolatech.com**). Во время написания книги все эти имена доменов были доступны.

Нет четких и однозначных рекомендаций по выбору удачного имени, но существует невероятное множество примеров неудачных имен. Далее перечислены ошибки, которых следует избегать.

- *Обилие дефисов.* Легко может возникнуть соблазн получить именно то имя домена, какое хотите, добавив в него несколько лишних символов, таких как дефисы между словами. Например, у вас нет шансов получить имя **www.popularbusiness.com**, но имя **www.popular-business.com** все еще свободно. Подумайте, как следует. Одних людей дефисы могут запутать, а другие могут их пропустить. (Другие символы, например знаки подчеркивания, гораздо хуже, всеми силами старайтесь их избегать.) Некоторые мастера Web-дизайна считают, что имя домена с одним дефисом вполне приемлемо, но с несколькими дефисами выглядит как сайт, рассылающий спам, и от него следует отказаться.

- *Фразы, приводящие в замешательство при наборе строчными буквами.* Имена доменов не чувствительны к регистру символов, и если вы набираете имя домена в браузере, он все преобразует в символы нижнего регистра (строчные). Проблема состоит в том, что некоторые слова при наборе строчными символами могут сливаться, особенно если они начинаются с гласных букв. Смотрите, что произойдет, если компания по изготовлению документации Prose Xact поместит свое коммерческое название из строчных букв в имя домена: **www.prosexact.com**. Поняли? (Это как раз тот случай, когда захочется вернуться к дефисам.)

> ***СОВЕТ***
>
> Несмотря на то, что имена доменов не чувствительны к регистру символов, ничто не мешает использовать заглавные буквы в визитных карточках, рекламе и маркетинговых материалах для увеличения наглядности имени домена. Наберут клиенты в своих браузерах **www.google.com** или **wWw.gOOgLE.cOm**, они попадут на один и тот же сайт.

- *Имена, не соответствующие коммерческому названию вашей компании.* Это классическая ошибка коммерсантов. Вы создаете цветочный магазин в Нью-Йорке с названием "Roses are Red". К сожалению, домен **www.rosesarered.com** уже занят, поэтому вы переходите к следующему возможному варианту **www.newyorkflorist.com**. Что же вы сделали на самом деле? Создали два разных коммерческих названия и какое-то шизофреническое раздвоение вашей компании. Чтобы избежать подобных проблем, когда начинаете новый бизнес, старайтесь одновременно выбирать коммерческое название своей компании и имя домена для нее, добиваясь их соответствия. Если коммерческое название у вас уже есть, остановите свой выбор на URL, который длиннее на одно-два слова, например, **www.rosesareredflorist.com**. Это имя не так коротко, как **www.newyorkflorist.com**, но оно поможет избежать неминуемой путаницы и раздвоения.

- *Необычные домены верхнего уровня.* Последние несколько букв (компонент после последней точки) в имени домена называются *доменом верхнего уровня*. Для коммерческих целей всем нужен адрес, заканчивающийся **.com**, и в результате это имя домена получить труднее всего. Конечно, есть и другие домены верхнего уровня, такие как **.net**, **.org**, **.biz**, **.info** и **.name**. Но проблема в том, что большинство Web-посетителей рассчитывает на **.com**. Если у вас имя домена **www.SuperShop.biz**, с вероятностью 50% кто-нибудь, пытаясь найти ваш сайт, наберет **www.SuperShop.com**. Эта ошибка легко может привести ваших поклонников к конкуренту (или на крайне низкосортный Web-сайт). Другими словами, порой стоит выбрать не самое лучшее имя, но получить наилучший домен верхнего уровня (т. е. **.com**).

> ***ПРИМЕЧАНИЕ***
>
> Домен верхнего уровня **.org** первоначально предназначался для некоммерческих организаций. Теперь он доступен всем для использования и злоупотреблений. Тем не менее, если вы создаете собственную некоммерческую организацию, **.org** подойдет больше, чем **.com**, а узнаваем он точно так же.

За время, прошедшее с момента выхода первого английского издания книги, получить приличное имя домена стало еще труднее. В прошлом вашими единственными конкурентами были только другие люди, собиравшиеся организовать Web-сайт, и перепродавцы, скупавшие ходовые имена и рассчитывавшие перепродать их с большой выгодой. Теперь же нечистые на руку люди без колебаний покупают просто любое имя домена, создают временную Web-страницу, заполненную рекламными объявлениями, и ждут несколько месяцев, чтобы узнать, сколько доверчивых посетителей попадется на их пути. Такую практику называют *опробованием домена* (domain tasting), и она на удивление прибыльна. Что же в итоге? Пока еще возможно сформировать привлекательное имя домена, которое есть в наличии, но от вас потребуется умение идти на компромисс и весь ваш творческий потенциал.

ПРИМЕЧАНИЕ

Существует масса новых доменов верхнего уровня, которые запланированы, но все еще не разрешены для повсеместного использования. К ним относятся **.eco**, **.love**, **.god**, **.sport**, **.gay** и почти все остальное, что еще могут придумать соискатели. Такие домены не появятся до 2012 г., и это по самым оптимистическим оценкам.

ЧАСТО ЗАДАВАЕМЫЙ ВОПРОС

Иностранные имена доменов

Некоторые имена доменов заканчиваются кодом страны. Следует мне получать такой домен?

Адрес **.com** — лучший друг создателя Web-сайтов. Но если вы не можете получить нужный вам **.com**, другое подходящее решение — согласиться на домен верхнего уровня, ориентированный на конкретную страну, например **.us** (США) или **.ru** (Россия).

Если вы предлагаете уроки игры на фортепиано в Англии, имя домена **www.pianolessons.co.uk** — неплохой выбор. Если же вы собираетесь продавать товары интернациональной аудитории, адрес **www.HotRetroRecords.co.uk** мог бы отпугнуть заинтересовавшихся покупателей, которые решат, что иметь дело с британским продавцом слишком хлопотно.

Для регистрации имен, ориентированных на конкретную страну, существуют специальные правила. Например, некоторые имена доступны только людям, живущим в соответствующей стране или имеющим зарегистрированный в этой стране бизнес. В соответствии с этими ограничениями некоторые компании Web-хостинга не продают определенные территориально-зависимые имена. Если при регистрации нужного вам имени, ориентированного на конкретную страну, возникли какие-то проблемы, можно воспользоваться поиском в Google и найти регистратора, который поддержит ваш выбор. Например, для того чтобы найти регистратора австралийских доменов, поищите "Australian domain names" (австралийские имена доменов).

У любой страны в мире есть собственный уникальный домен верхнего уровня. Но некоторые из них доступны для нестандартного использования другими людьми. Например, домен верхнего уровня **.tv** был создан для крошечной страны Тувалу, но теперь он доступен любому, кто хочет создать Web-сайт, посвященный телевидению (при этом правительство Тувалу получает маленькую долю от прибыли). К аналогично перепрофилированным доменам относятся **.me**, **.cd**, **.tm** и **.ws**. Дополнительную информацию о нестандартных альтернативах применения этих имен доменов можно найти в Википедии по адресу **http://tinyurl.com/rnlmf**.

Поиск имени

Обзаведясь несколькими вариантами имени домена, начинайте проверять их доступность. Делать это можно, даже если вы еще не выбрали Web-хост. На самом деле Web изобилует сервисами, позволяющими проверить доступность любого имени домена.

> ***Совет***
>
> Поиски имени домена — это существенный этап подготовки к работе. Постарайтесь придумать для имени как можно больше вариантов и необычных комбинаций. Стремитесь подобрать десяток подходящих имен, в этом случае у вас будет больше выбор. После того как список составлен, почему бы вечерком не сделать несколько телефонных звонков и не попристовать к друзьям и родственникам, чтобы проверить их первые впечатления?

Почти все компании Web-хостинга предлагают собственные средства для поиска имени домена. На рис. 3.3 показано одно из них от **www.godaddy.com**. Для начала введите имя домена, которое вам нравится больше других, и щелкните мышью кнопку **Search** (Поиск). (Обратите внимание на то, что скорее всего вам не придет-

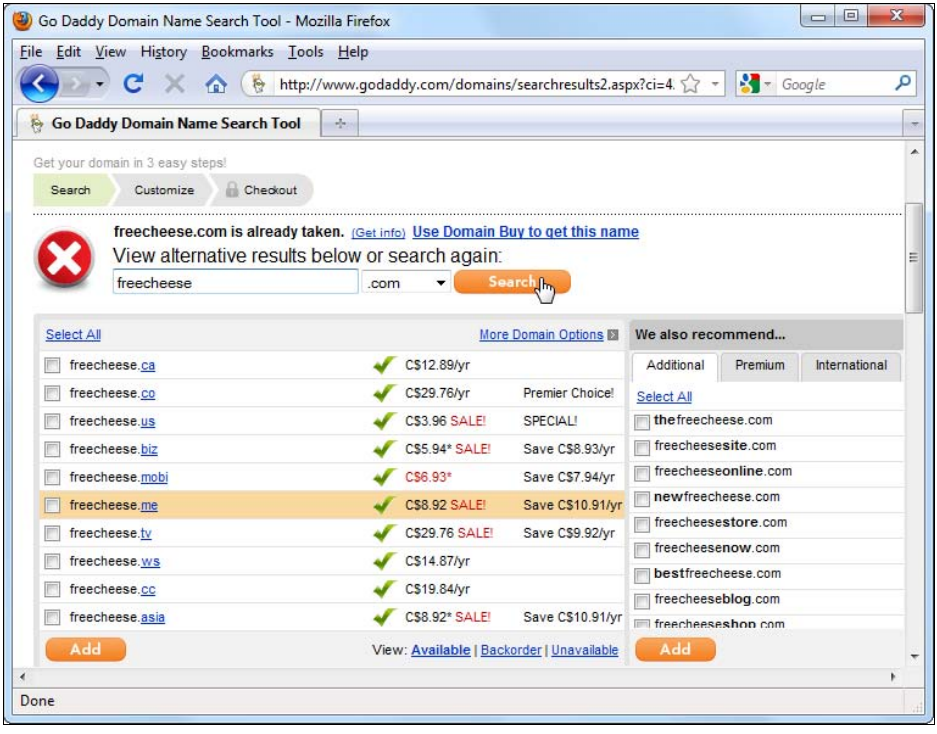


Рис. 3.3. Поиск показывает, что ваш первый вариант **www.freecheese.com** занят. Все, что осталось, — это менее привлекательный **www.freecheese.biz** и территориально-зависимые домены. В списке **Additional** (Дополнительные), расположенном справа, сгенерированы несколько похожих имен доменов, которые свободны (в данном случае это имена **www.thefreecheese.com** и **www.freecheesesite.com**). Вы также можете щелкнуть мышью ссылку **Premium** (Премиум-класс), чтобы увидеть имена доменов, продаваемые их владельцами; но приготовьтесь выложить кругленькую сумму (например, более 1 000 долларов за **www.fatfreecheese.com** и больше 3 000 долларов за **www.cheesenow.com**)

ся набирать **www** в начале имени домена или **.com** в конце, поскольку эти составляющие предполагаются по умолчанию.)

Когда вы выполните поиск и найдете свободное имя домена, компания, предоставляющая хостинг, предложит вам купить его. Но пока ничего не предпринимайте, потому что вам еще нужно найти лучший вариант хостинга.

ПРИМЕЧАНИЕ

Возможно, вы думаете, что можете определить доступность имени домена, просто набрав его в своем Web-браузере. Но такой метод проверки отнимает больше времени и не дает окончательного ответа. Кто-то мог купить имя домена без создания Web-сайта, поэтому, даже если вы не нашли сайт с помощью браузера, домен может быть занят.

Регистрация вашего имени

После того как свободное имя найдено, возможно вы захотите подождать с его регистрацией до подписки на хостинг (о котором вы узнаете в следующем разделе). Многие Web-хосты, если вы арендуете у них место, предлагают бесплатную регистрацию имени домена или скидку на нее. Кроме того, выполнение регистрации и оформление хостинга одновременно — самый легкий способ настройки вашего имени домена, потому что связь между вашим именем домена и вашим Web-сайтом при этом устанавливается автоматически.

Однако возможны ситуации, когда удобнее регистрировать имя домена отдельно от вашего пакета на Web-хостинг. Далее перечислено несколько таких примеров.

- Вы не хотите создавать Web-сайт прямо сейчас, но имя зарегистрировать хотите, чтобы никто не мог прибрать его к рукам (тактика известна как *паркинг* или *парковка домена*). Когда-нибудь в будущем вы сможете разработать Web-сайт, использующий это имя.
- У вас уже есть зарезервированное место в Web, возможно, через вашего провайдера интернет-услуг (Internet Service Provider, ISP). И вам нужно лишь сделать Web-сайт более профессиональным, получив персональное имя домена. Этот вариант может оказаться сложнее и возможно потребует процедуры, называемой *переадресацией* или *перенаправлением домена* (об этом вы прочтете буквально через несколько минут).
- Ваша компания Web-хостинга не может зарегистрировать тот тип домена, который вы хотите. Это может произойти, если вам нужно имя домена с территориально-ориентированным доменом верхнего уровня.

Если вы не попадаете ни в одну из этих конкретных категорий, переходите сразу к *разд. "Получение места в Web"*, чтобы начать поиск подходящего Web-хоста. В противном случае продолжайте чтение, чтобы ознакомиться с подробностями, касающимися регистрации имени домена и управления им.

ПРИМЕЧАНИЕ

Все компании, предоставляющие Web-хостинг, позволяют регистрировать несколько имен домена для одного и того же Web-сайта. Это значит, что вы можете зарегистри-

ровать два адреса: **www.FancyPants.com** и **www.FancyPants.biz** и сделать так, чтобы оба они указывали на один Web-сайт. Конечно, придется заплатить за регистрацию дополнительного имени домена. (В действительности этой стратегией пользуются крупные Web-компании, чтобы учесть опечатки, которые могут сделать потенциальные посетители. Например, проверьте, куда приведут вас адреса **amzn.com** и **googel.com**.)

ЧАСТО ЗАДАВАЕМЫЙ ВОПРОС

Хост здесь, домен там

Могу ли я купить имя домена и место в Web у разных компаний и заставить их работать сообща?

Простейший вариант — получить и то, и другое в одной компании, но это не всегда возможно. Быть может, вы купили имя домена до того, как начали формировать Web-сайт, и не хотите платить за перенос домена. Или у вас есть территориально-ориентированное имя домена (например, **www.CunningPets.co.uk**), которое ваш провайдер Web-хостинга не может зарегистрировать. А может вам просто нужна гибкость, позволяющая часто менять Web-хосты и получать наилучшее обслуживание или самые низкие тарифы.

Для исполнения виртуозного танца с участием нескольких компаний вам необходима некоторая техническая поддержка от вашего провайдера Web-хостинга. Свяжитесь с их службой поддержки и сообщите, что собираетесь делать. Они дадут вам конкретные рекомендации и настроят свои серверы имен (больше об этом через минуту), используя корректные данные, относящиеся к вашему домену.

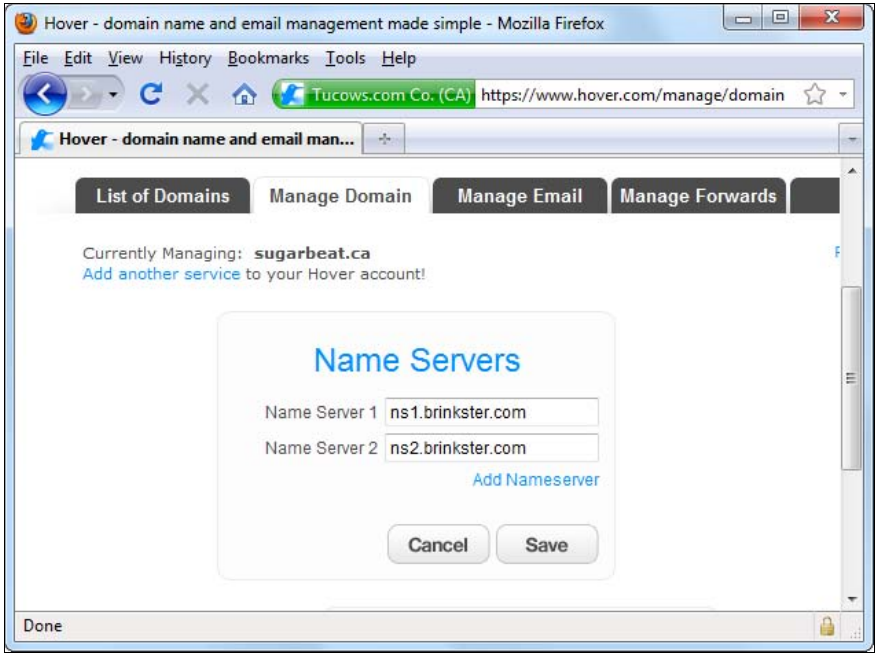


Рис. 3.4. Владелец Web-сайта зарегистрировал свой домен (sugarbeat.ca) с помощью компании **www.hover.com**, но хранит свой контент на сервере компании **www.brinkster.com**. Чтобы проделать это, он передает управление своим именем домена серверам преобразования имен ns1.brinkster.com (**http://ns1.brinkster.com**) и ns2.brinkster.com (**http://ns2.brinkster.com**)

Следующая задача — изменить регистрационные данные вашего домена. Для этого выполните следующее:

1. Выясните имена DNS-серверов в вашей компании, предоставляющей Web-хостинг. DNS-серверы— это компьютеры, преобразующие имена доменов в числовые IP-адреса (см. *разд. "Как браузеры анализируют URL" данной главы*). Информацию о них может вам дать обслуживающий персонал компании.
2. Обратитесь в компанию, где вы зарегистрировали имя домена, и обновите параметры регистрации. Измените имя сервера в соответствии с именами серверов, которые вы выяснили в п. 1 (как показано на рис. 3.4).

В зависимости от способа работы DNS-серверов для внесения изменений может потребоваться 24 часа или более.

После того как изменение внесено, фактически можно сказать, что теперь за выдачу IP-адреса вашему Web-сайту отвечает ваш Web-хост. Когда кто-нибудь наберет имя вашего домена в своем браузере, последний обратится к DNS-серверу вашего провайдера Web-хостинга для получения IP-адреса. С этого момента дальнейшее ваше плаванье будет спокойным и гладким.

После модификации регистрации имени домена у вас будут все те же два счета для оплаты. Вы заплатите вашему провайдеру Web-хостинга за хостинг и внесете годовую плату за регистрацию имени домена на счет компании, в которой зарегистрировали свое имя домена.

Парковка домена

Парковка домена означает, что вы зарегистрировали имя домена, но пока не купили никакие другие сервисы, такие как аренда некоторого объема долговременной памяти в Web.

Большинство пользователей применяют парковку доменов для резервирования имени. В активно заселяемом мире Web многие пользуются ею для защиты своих имен (например, **www.samanthamalone.com**). Парковка домена также полезна, если вы хотите защитить некоторые коммерческие названия для последующего использования.

Совет

Если вы пока только резервируете имя домена, не поленитесь и выберите также компанию, услугами которой вы хотели бы пользоваться для размещения своего Web-сайта. Вы, конечно, можете переключать имена доменов с одного Web-хоста на другой, но все же это дополнительная головная боль. Свяжитесь с хостом, которым пользуетесь в данный момент, чтобы получить конкретные рекомендации.

Истинная привлекательность парковки домена в ее дешевизне. Вы платите символическую плату за регистрацию домена (не более 5 долларов в год) и можете беречь для себя имя столько времени, сколько готовы за него платить.

Переадресация домена

Переадресация или перенаправление домена позволяет сочетать персонифицированное имя домена с местом в Web-пространстве, которое у вас уже есть. Например, у вас может быть свободное место на Web-серверах вашего провайдера интернет-услуг, в вашей школе, на работе, от службы бесплатного Web-хостинга или

у сумасбродного дядюшки с Web-сервером в подвале его дома. В таких ситуациях вы можете сэкономить деньги, потому что не придется платить за хранение вашего сайта. Но у вас не будет возможности выбрать свое имя домена. Если у вас есть место у вашего провайдера, вы привязаны к URL наподобие **http://member.magicisp.com/members/personalwebspace/~henryj420/home**, далеко не такому легко запоминающемуся, как **www.HenryTheFriendly.com**.

Не отчаивайтесь! Даже если у вас есть бесплатный Web-хостинг, вы все равно можете присвоить вашему Web-сайту привлекательный URL. Нужно просто отдельно купить URL и применить переадресацию домена, чтобы ваш совершенно новый URL указывал на ваш уже существующий Web-сайт. В предыдущем примере вы могли бы купить имя домена **www.HenryTheFriendly.com** и воспользоваться переадресацией домена для ссылки на ваш сайт **www.HenryTheFriendly.com**, размещенный по адресу **http://member.magicisp.com**.

Совет

Даже если URL от вашего провайдера не так плох, как в предыдущем примере, купить собственное имя домена все равно стоит, по нескольким причинам. Если вы используете URL от вашего провайдера и последний решил изменить конфигурацию или вы сменили провайдера, ваши Web-посетители больше не смогут найти ваш сайт. Но если вы пользуетесь собственным URL и переадресацией домена, достаточно обновить настройку имени домена, чтобы учесть в ней новую конфигурацию провайдера или просто другого провайдера. В любом случае ваше персонифицированное имя домена продолжает действовать. Никто даже не заметит изменений.

Для применения переадресации домена прежде всего нужно зарегистрировать имя домена, а переадресация в этом случае будет сопутствующим сервисом. Затем вы сможете войти в систему и настроить параметры переадресации (рис. 3.5).

Многие хосты, предлагающие переадресацию, позволяют вам замаскировать ваш URL (см. установленный флажок на рис. 3.5). Если вы так и поступаете, посетители не догадываются, что ваш хост перенаправляет их, потому что URL в их браузерах не меняется. Поэтому когда вы отправите ваших посетителей с сайта **www.HenryTheFriendly.com** на какой-нибудь длинный громоздкий URL, в строке адреса браузера останется первоначальный адрес **www.HenryTheFriendly.com**. Маскировка не только придает сайту более опрятный и профессиональный вид, она также гарантирует, что в случае установки пользователем закладки, закладка получит первоначальный URL (имя домена, купленное вами), а не URL перенаправления (на котором вы храните свой контент).

С другой стороны, когда вы используете переадресацию домена для отправки посетителей с альтернативного Web-сайта (например, **.biz** сайта) на ваш основной Web-сайт, не следует скрывать URL. В этом случае имеет смысл сохранять в строке адреса URL вашего основного сайта.

Примечание

Маскировку URL иногда называют скрытыми переадресацией или перенаправлением.

Однако переадресация доменов — это прием не без изъянов, и порой они просматриваются. Во-первых, переадресация домена может запутать поисковые машины.

Например, когда ваш Web-сайт появляется в результатах поиска Google, для него может выводиться реальный URL вашего Web-сайта (тот, на который выполняете переадресацию), а не купленное вами имя домена. Кроме того, переадресация может вызывать проблемы при отслеживании ваших посетителей. Все, кто попадает на ваш Web-сайт, будут казаться пришедшими с вашего перенаправленного домена, и вы потеряете информацию о том, как они первоначально попали на него (пришли ли они по ссылке, размещенной на другом Web-сайте, или через Google). И наконец, переадресация URL добавляет крошечную, почти незаметную задержку при выводе вашей страницы, поскольку браузерам посетителей вашего Web-сайта приходится общаться с двумя разными Web-серверами.

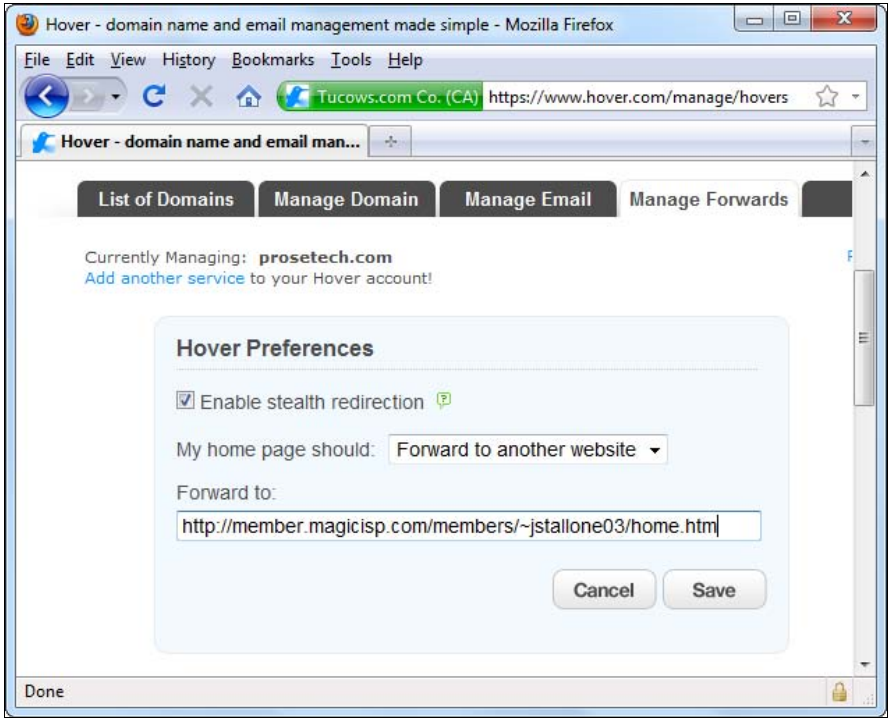


Рис. 3.5. Служба переадресации домена отправляет посетителей, набравших адрес **www.HenryTheFriendly.com** на гораздо более неудобный URL, на котором действительно размещен Web-сайт (нижнее поле ввода). Обычно Web-хосты так реализуют переадресацию домена, что в строке адреса браузера отображается первоначальное имя домена, несмотря на то, что хост посылает посетителей на другой сайт

Бесплатные имена доменов

Как вы уже наверняка догадались, Web — замечательное место для бережливых покупателей. Вы можете прилично сэкономить не только на покупке щегольского iPod и подержанного дивана-кровати, но и выбрать имя домена за кругленькую сумму в ноль долларов. Секрет — в регистрации на бесплатном сервисе *поддоменов*.

Поддомены — это расширения существующего имени домена. Бесплатный сервис поддоменов может владеть доменом **www.net.tc** и позволить вам зарегистрировать поддомен **www.acebusiness.net.tc**. Что и говорить, конечная составляющая имени домена (**.net.tc**) немного нескладная, но зато она бесплатна. Дополнительный плюс состоит в том, что включить популярные слова в имя поддомена гораздо проще, чем в имя отдельного домена. Например, даже не пытайтесь прибрать к рукам следующие версии имени: **www.chocolates.com** или **.net**, **.org**, **.biz**, **.us**, **.ca**, поскольку все они уже разобраны. Но если вы обратитесь к популярному сервису поддоменов **www.smartdots.com** (рис. 3.6), вы сможете зарегистрировать имя **www.chocolates.net.tc**, не потратив ни копейки.

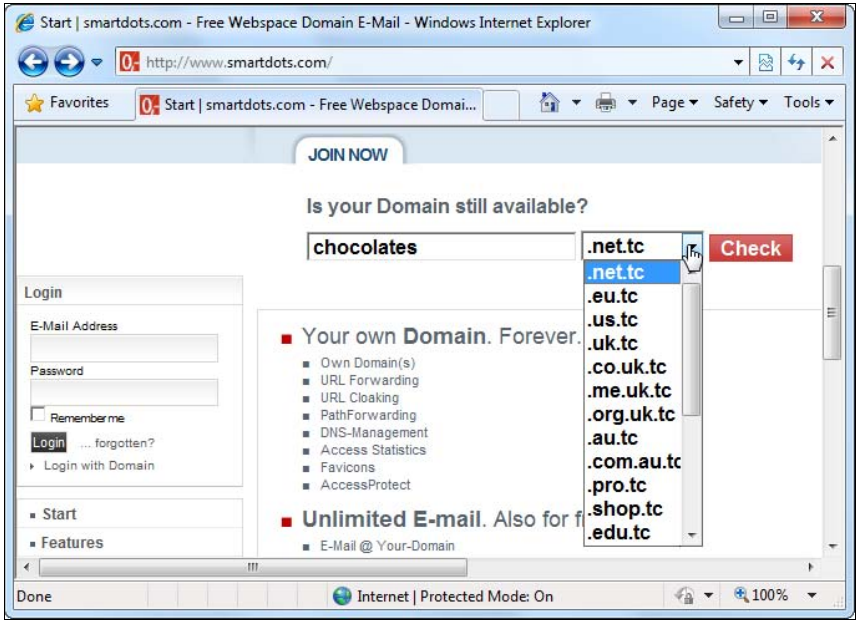


Рис. 3.6. На Web-сайте smartdots вы можете зарегистрировать бесплатный поддомен и связать его с вашим Web-сайтом. Выбирать можно из нескольких корневых каталогов доменных имен, включая стандартные .net.tc корневые каталоги, предлагающие территориально-ориентированные имена (например, .us.tc и .uk.tc) или определенный тип контента (например, .pro.tc, .shop.tc и .edu.tc)

Получение места в Web

Для того чтобы достичь статуса суперзвезды в Web, вам нужно всего лишь имя домена и небольшой объем дисковой памяти на Web-сервере. При выборе Web-хоста нет одного универсального решения. Вы выбираете компанию, предоставляющую хостинг, исходя из своего бюджета, назначения вашего Web-сайта и собственных необъяснимых фантазий (посмотрим правде в глаза — у одних компаний, предлагающих хостинг, имена более классные, чем у других).

Вам уж точно не нужен Web-хост, предлагающий специальные программы, которые обещают помочь создать Web-сайт в два-три простых приема (рис. 3.7). По-

добные средства бывают только посредственные или ужасные. Помимо всего прочего, если вы согласны довольствоваться созданием типового Web-сайта, как у многих других, то, вероятно, не заинтересовались бы HTML и, наверное, не стали бы читать эту книгу.

Вместо этого вам нужен стандартный хостинг Web-сайта, где вам предложат некоторый объем памяти на сервере, чтобы вы могли распоряжаться им по своему усмотрению. Вы создаете ваши страницы на своем компьютере и затем копируете их файлы на сервер, чтобы другие могли на них посмотреть. Для освоения предлагаемого в книге материала вам понадобится именно этот тип Web-хостинга.

Пакеты Web-хостинга обычно оплачивают помесячно. Для типового хостинга месячная плата колеблется от приемлемых 5 до 10 долларов в месяц. Конечно, стоимость может резко вырасти в зависимости от услуг, которые вам потребуются от вашего хоста.

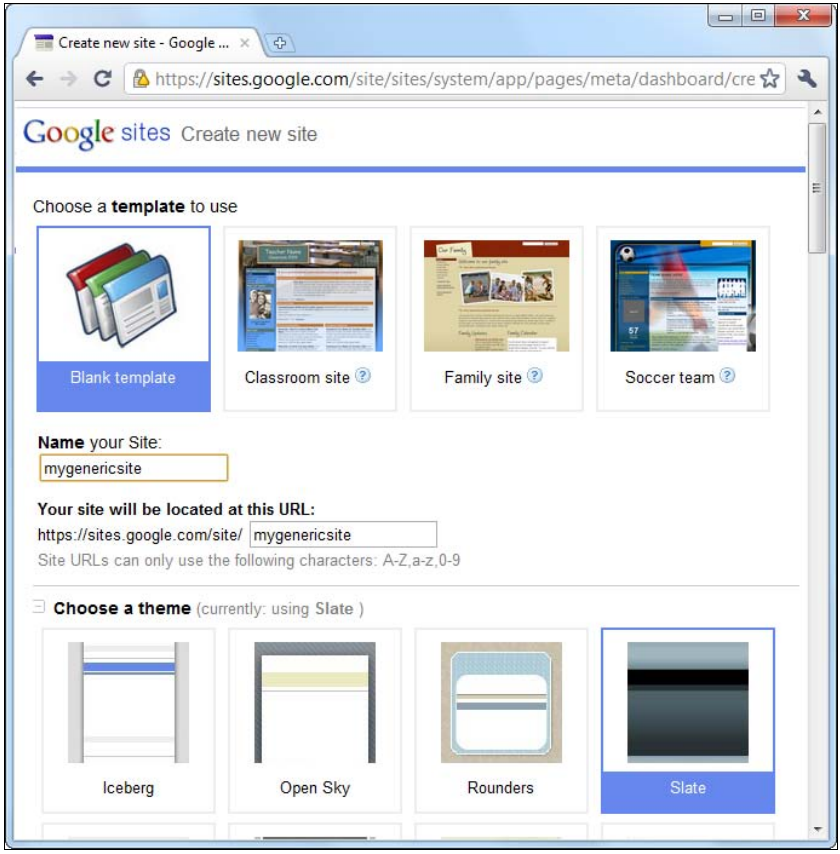


Рис. 3.7. Мастера создания Web-страниц предоставляют легкий и быстрый способ создания типовых страниц. Для истинных лентяев показанный сервис Google Sites ((**http://sites.google.com**) — возможно лучший выбор из этого класса программ. Но если вам нужен настоящий контроль, истинный стиль или оригинальный штрих, гораздо полезнее выучить HTML и CSS. Если вы все еще мечтаете облегчить себе жизнь, опробуйте гораздо более мощное программное средство для Web-дизайна, о котором пойдет речь в *главе 4*

Определение ваших потребностей

Прежде чем принимать решение о хосте, задайте себе очень важный вопрос: "Какие услуги мне нужны?" Web-хосты торопятся напичкать свои рекламные объявления и Web-сайты профессиональным жаргоном, но не скажут вам, какие сервисы по-настоящему полезны. Далее приводится краткий обзор того, что продают хосты и что вам нужно знать обо всех их предложениях. Если хотите иметь перед глазами перечень всех нужных вам показателей, можно воспользоваться памяткой, приведенной далее в *разд. "Контрольный список для выбора Web-хоста" данной главы*.

- *Место в Web* — определенный объем дискового пространства на сервере, который вы арендуете для хранения вашего Web-сайта. Несмотря на то что HTML-страницы очень невелики, вам может понадобится больше дискового пространства для размещения изображений или файлов, предназначенных для скачивания посетителями, например, видеоролик вашей свадьбы. Скромный сайт может легко прожить на мизерных 100 Мбайтах (это не гигабайты, а мегабайты), если вы не набьете его до краев картинками и видеороликами. Большинство Web-хостов предлагают в 10 или 100 раз бóльший объем, зная, что вы, скорее всего, никогда не будете использовать его весь.

> ***Примечание***
>
> Для любителей цифр 1 гигабайт (Гбайт) равен 1024 мегабайтам (Мбайт). Для сравнения, у современных жестких дисков может быть объем 500 Гбайт и больше, т. е. достаточно места для размещения десятков тысяч Web-сайтов.

- *Пропускная способность* (или, как иногда говорят, Web-трафик) — максимальное количество информации, которое вы можете предоставить любому посетителю вашего сайта в течение месяца. В большинстве случаев вам будет достаточно минимальной пропускной способности, которую предлагает вам компания, предоставляющая хостинг. Дополнительную информацию см. далее во врезке *"На профессиональном уровне. Загадка пропускной способности"*.
- *Имя домена* — адрес пользовательского Web-сайта, например **www.HenryTheFriendly.com**. Если вы решите получить персональное имя домена, совсем не обязательно получать его в той же компании, где вы разместили свой Web-сайт. Получение и того, и другого из одного источника облегчает жизнь, и компании хостинга часто предлагают имена доменов по сниженным ценам, если вы подписываетесь на Web-хостинг.
- *Адреса электронной почты*. С вероятностью 50% у вас уже есть таковые. Но возможно вы хотите получить адрес электронной почты, соответствующий адресу вашего Web-сайта, особенно если вы платите за индивидуальное имя домена. Например, если у вас есть имя домена **www.HenryTheFriendly.com**, вам захочется иметь адрес электронной почты **Hank@HenryTheFriendly.com**. Большинство компаний, предоставляющих хостинг, разрешают создавать десятки или сотни адресов электронной почты на базе вашего домена (что очень удобно, если вы создаете небольшую компанию) и, кроме того, позволяют читать вашу электронную почту в вашем браузере или отправляют ее в программу настольного компьютера, например, Microsoft Outlook.

- *Возможность выгрузки файлов в Интернет.* Легкость передачи ваших файлов на ваш сервер — это еще одна важная деталь. Как отмечалось в предыдущей главе, вы можете совершенствовать ваши Web-страницы на собственном компьютере, прежде чем передать их в Интернет. Но после того как вы подготовили ваш Web-сайт для массового просмотра, нужен удобный способ копирования всех ваших файлов на ваш сервер. Подавляющая часть коммерческих Web-хостов предлагает поддержку FTP-протокола (File Transfer Protocol, Протокол передачи файлов), позволяющего легко копировать несколько файлов одновременно (подробности см. в *разд. "FTP" данной главы*).
- *Программная поддержка.* Сегодня большинство Web-хостов поддерживают хотя бы несколько из десятков имеющихся серверных языков программирования. При поиске Web-хоста вы столкнетесь с их именами: ASP, PHP, CGI, Perl, Python, Ruby и т. д. Несмотря на то что это мощные языки программирования, они требуют квалификации профессионального программиста и не обсуждаются в этой книге.

> ***ПРИМЕЧАНИЕ***
>
> Хотя в этой книге подробно не обсуждаются серверные программы, в *главе 15* вы узнаете о *клиентских сценариях*. Клиентские сценарии или скрипты выполняются прямо в браузерах посетителей Web-сайтов и обладают гораздо более ограниченными возможностями по сравнению с серверными программными платформами. Клиентские сценарии широко применяются для создания специальных эффектов, таких как анимированные кнопки. Их достоинство заключается в том, что даже новичок в программировании может поместить простой скрипт на свою Web-страницу и наслаждаться его работой. Но сейчас не стоит о них беспокоиться, поскольку, в отличие от серверного программирования, клиентским скриптам не требуется специальная поддержка от вашей компании, предоставляющей Web-хостинг.

- *Навороты*. Для того чтобы привлечь вас на свою сторону, Web-хосты часто включают уйму наворотов и примочек в свои тарифные планы. Например, иногда они расхваливают свои замечательные быстрые и легкие средства создания Web-сайтов. На самом деле они приглашают вас воспользоваться для создания вашего собственного сайта корявым компонентом программного обеспечения, размещенного на их Web-сайте. Все кончится созданием типового сайта с минимумом возможностей для самовыражения. Держитесь подальше от этих бесполезных средств. Гораздо нужнее предлагаемая компанией Web-хостинга статистика сайта — подробная информация о том, сколько посетителей собирается на вашем сайте за день или ежемесячно. (В *главе 11* вы познакомитесь с бесплатным средством отслеживания посетителей, которое намного лучше того, что предлагают большинство хостов.)

НА ПРОФЕССИОНАЛЬНОМ УРОВНЕ

Загадка пропускной способности

Большинство компаний, предоставляющих Web-хостинг, устанавливают цены на свои услуги, исходя как минимум из нужных вам объема дискового пространства и пропускной способности. Это может стать проблемой, потому что среднестатистический создатель Web-сайта не представляет, как вычислять эти показатели, не говоря уже о получении их реальных ориентировочных значений.

К счастью, вы можете сэкономить уйму времени и усилий, усвоив один маленький секрет: для среднестатистического персонального сайта или коммерческого сайта для малого бизнеса не нужно большого дискового объема и огромной пропускной способности. Вы, скорее всего, можете выбрать наименьшие значения из предлагаемых любой компанией Web-хостинга и жить припеваючи.

Если вы все же настаиваете на вычислении пропускной способности, вот как это делается. Предположим, вы создаете относительно скромный Web-сайт с 50 Web-страницами, пакетом небольших, оптимизированных для Web изображений и несколькими документами, предназначенными для скачивания из Интернета (скажем, PDF-файлами или документами в формате Word). Со всеми компонентами такой Web-сайт занимает 10 Мбайт дискового пространства.

Допустим, ваш сайт хорошо сделан и принимает около 30 посетителей в день. Если каждый посетитель просмотрит все страницы (другими словами, загрузит весь ваш Web-сайт целиком), ваша дневная пропускная способность вырастет до 300 Мбайт (30 посетителей × 10 Мбайт). Ваша ежемесячная пропускная способность будет равна 9 Гбайт (30 дней × 300 Мбайт). Полученное значение существенно меньше типичных установочных значений в планах Web-хостинга, которые, как правило, равны 100 Гбайт в месяц или больше.

Контрольный список для выбора Web-хоста

- *Объем Web-пространства.* 50 Мбайт вполне приемлемо, если вы получаете бесплатно место на Web-сервере вашего провайдера интернет-услуг. Если вы платите ежемесячно за предоставляемое место в Web, вы почти наверняка получите несколько гигабайтов или больше. Вероятно, это гораздо больше, чем вы сможете использовать, даже если будете хранить фотографии, звуковые данные или другие большие файлы. В действительности многие Web-хосты предлагают сотни гигабайтов пространства, зная, что почти ни у кого нет достаточно времени, чтобы занять весь этот объем.
- *Пропускная способность.* Слишком большая пропускная способность вам не нужна. Большинство пользователей могут вообще игнорировать этот показатель. Исключением может стать нереально популярный Web-сайт, желание хранить очень большие файлы и разрешать посетителям их скачивать или демонстрация громадного каталога цифровых фотографий, музыки или видео и отказ от использования сторонних сервисов, таких как Flickr или YouTube. Если вы относитесь к меньшинству пользователей, нуждающихся в громадном Web-трафике, поищите Web-хост, который предлагает неограниченную пропускную способность. В этом случае вам не придется беспокоиться о том, что вы превысили установленный лимит.
- *Имя домена.* Это ваша идентификация — **www.You.com**. Идеально, если ваш Web-хост включит в тарифный план имя домена бесплатно.
- *Адреса электронной почты.* Они предоставляются вместе с именем домена. Ищите Web-хост, который предлагает пять или больше адресов электронной почты (иногда называемых псевдонимами электронной почты), поскольку это позволит вам применять отдельные адреса для связи с разными людьми или использовать их для разных целей. Также выбирайте наличие доступа к электронной почте из Web, что позволит вам читать электронную почту на любом подключенном к Web компьютере.

- *FTP-доступ.* Он позволит вам легко выгружать свои файлы в Web и, как правило, входит в стандартный набор услуг, предлагаемых большинством компаний Web-хостинга.
- *Техподдержка.* Лучшие компании обеспечивают 24-часовую техподдержку, в идеале по бесплатному номеру телефона или с помощью средства обмена сообщениями в режиме реального времени, что позволит вам задавать вопросы прямо из своего браузера.
- *Серверные скрипты.* Несмотря на то что серверное Web-программирование слишком сложно для непрофессионалов, это средство позволит вам самосовершенствоваться. Если ваш Web-сайт поддерживает технологию серверного программирования (такую как CGI, ASP или ASP.NET), вполне вероятно, что вы сможете применить на ваших Web-страницах разработанный кем-то другим скрипт, позволяющий выполнить сложную задачу, например, заполнить форму информацией о посетителе.
- *Расширения FrontPage.* Если вы создаете свой сайт с помощью программы Expression Web (альтернативы FrontPage корпорации Microsoft), вы можете применить серверные расширения программы FrontPage, чтобы получить несколько дополнительных наворотов. Например, можно сосчитать количество посетителей для конкретной страницы и позволить посетителям загружать файлы на сервер.
- *Купоны на рекламу вашего Web-сайта.* Некоторые Web-хосты включают в свои предложения чеки на платные сервисы, например кредит в 25 долларов на использование программы Google AdWords. Как вы узнаете из *главы 12*, для привлечения нужных людей на ваш сайт можно воспользоваться рекламными объявлениями Google AdWords.

Выбор вашего хоста

Теперь, держа в голове свои потребности, можно приступить к поиску Web-хоста. В следующих разделах описаны возможные варианты.

Ваш ISP (провайдер интернет-услуг)

Как вы уже поняли, у вашего ISP — компании, обеспечивающей ваше подключение к Интернету, могут быть собственные службы хостинга. На самом деле порой провайдеры включают эту услугу в вашу абонентскую плату, т. е. может оказаться, что у вас уже есть выделенный объем дискового пространства, о котором вы даже не подозреваете. Если вы оказались в подобной ситуации, поздравляю. Никаких дополнительных шагов не требуется. Если не уверены, быстро звоните своему провайдеру и получите дополнительную информацию. Подтвердите, что вам требуется "место в Web для личных нужд", т. к. многие ISP также предоставляют за месячную плату пакеты Web-хостинга большого объема. Включенный в ваш тарифный план объем Web-пространства всегда значительно меньше того, который вы получаете от компании, специализирующейся на Web-хостинге, и никогда, как правило, в него не включено имя домена (хотя его можно купить и отдельно).

Примечание

Часто ваш провайдер может предлагать Web-хостинг, от которого вы решите отказаться. Например, предоставляется мало места, или вас заставляют применять их ограниченные программы создания Web-сайтов (что уж точно в тягость). По этим причинам, как правило, лучше заплатить отдельной компании, предоставляющей Web-хостинг.

Компании, предоставляющие Web-хостинг

Строго говоря, любая компания, предоставляющая место в Web, — это Web-хост, но есть класс компаний, специализирующихся на Web-хостинге и не делающих ничего другого. Найти такие компании можно в Интернете или компьютерных магазинах. К сожалению, компании Web-хостинга всегда взимают помесячную оплату. Бесплатно вы ничего не получите.

Печальная истина заключается в том, что в Интернете проанализировать сведения о компаниях, предлагающих Web-хостинг, практически невозможно, поскольку во Всемирной паутине объявлений о предоставлении хостинга больше чем о фармацевтических товарах по сниженной цене. К счастью, есть много проверенных вариантов, некоторые перечислены в табл. 3.1 — с них можно начать. Если вы любознательны, познакомьтесь с их сайтами, сравните цены и предлагаемые услуги.

Таблица 3.1. *Примеры интернет-фирм, предоставляющих Web-хостинг*

Название	**URL**
Brinkster	**www.brinkster.com**
DreamHost	**www.dreamhost.com**
GoDaddy	**www.godaddy.com**
HostGo	**www.hostgo.com**
Insider Hosting	**www.insiderhosting.com**
OCS Solutions	**www.ocssolutions.com**
Pair Networks	**www.pair.com**
Sonic.net	**www.sonic.net**

Не рассчитывайте на быстрый поиск во Всемирной паутине объективных обзоров Web-хостинга. Большинство Web-сайтов, предлагающих обзор и ранжирование Web-хостов, просто рекламируют несколько компаний, которые платят за рекомендации. Популярные технические сайты годами не дают обзоров Web-хостов, т. к. тщательный анализ даже части из них потребовал бы многочисленных людских ресурсов. Их старые обзоры принесут также мало пользы, поскольку качество компании, предоставляющей хостинг, может измениться очень быстро.

Но в ваших поисках Web-пространство не совсем бесполезно. Вы можете получить информацию о компаниях хостинга из интернет-форумов, где начинающие пользователи Web-хостов беседуют с более опытными. Один из лучших — WebHostingTalk,

который можно найти по адресу **http://tinyurl.com/5zffwp**. Этот форум особенно полезен, если вы уже свели свой выбор к нескольким компаниям и хотите задать вопрос или услышать об опыте других.

При знакомстве с разными хостами вам придется разбираться с невообразимым количеством показателей на различных Web-сайтах. В следующих двух разделах вы узнаете, как прорываться сквозь рекламный туман и находить важную информацию на Web-сайтах двух взятых для примера компаний, предоставляющих хостинг.

НА ПРОФЕССИОНАЛЬНОМ УРОВНЕ

Приручение длинных URL с помощью TinyURL

Зоркий глаз заметит, что URL, упомянутый на предыдущей странице и приводящий вас на форум WebHostingTalk, начинается с казалось бы не связанного с ним имени домена TinyURL.com (**http://tinyurl.com**). URL форума WebHostingTalk был намеренно укорочен с помощью TinyURL, Web-сайта, предоставляющего бесплатный сервис переадресации URL. TinyURL — очень удобное средство, которое можно применять всегда, случайно встретившись с URL, невероятно длинным или запутанным настолько, что у вас с самого начала нет шансов записать его, набрать на клавиатуре или прокричать по телефону.

Вот как применяется TinyURL. Сначала скопируйте ваш ужасно длинный адрес. Затем перейдите на Web-сайт **http://tinyurl.com**, в текстовое поле на домашней странице введите или вставьте из буфера ваш URL и щелкните мышью кнопку **Make TinyURL** (Создать TinyURL). Сайт наградит вас гораздо более коротким URL, который начинается с **http://tinyurl.com**. В предыдущем примере TinyURL заменил возмутительно сложный URL **http://www.webhostingtalk.com/forumdisplay.php?s= aa8768ada1bc5ddbd96e0578584cffce&f=1** на **http://tinyurl.com/5zffwp**.

Несмотря на то, что новый URL ничего не означает, он намного короче. Но самое главное — короткий URL действует так же исправно, как и исходный. Введите его в любом браузере и попадете на форум WebHostingTalk.

Как же эта система работает? Когда вы набираете короткий URL, ваш браузер переносит вас на Web-сайт TinyURL. Там хранится длинный список сложных URL, предоставленных пользователями, а также новые более короткие URL, которые сайт создает вместо них. Когда вы запрашиваете страницу, сайт ищет короткий URL в списке, находит исходный URL и перенаправляет ваш браузер на тот сайт, который вам действительно нужен. Еще одно краткое замечание: весь процесс разворачивается так быстро, что вы даже не догадаетесь о его присутствии, если не прочли эту врезку.

Обзор Web-хоста № 1

На рис. 3.8 показано, как можно попасть на домашнюю страницу популярной компании Web-хостинга Aplus.Net. Компания предлагает выделенные серверы, стандартный хостинг, регистрацию имени домена и сервисы для Web-дизайна. Все четыре услуги разработаны, чтобы помочь вам оказаться в Web-пространстве, но вам нужен только Web-хостинг. *Выделенный сервер* — это услуга Web-хостинга премиум-класса. Она означает, что ваш Web-сайт запускается на собственном сервере, отдельном компьютере, не содержащем больше ни одного сайта. Она в первую очередь должна интересовать участников крупного бизнеса с мощными сайтами, активно поглощающими компьютерные ресурсы. Большинство персональных сайтов и сайтов малого бизнеса существуют на серверах совместного использования и запускаются без каких-либо заметных задержек.

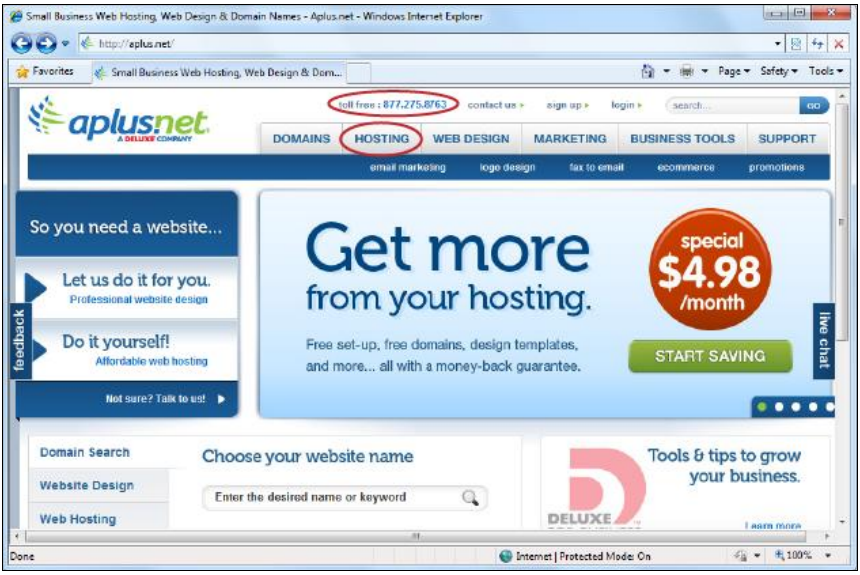


Рис. 3.8. На этой странице представлен большой объем информации. Щелкните кнопкой мыши вкладку **Hosting**, чтобы познакомиться с тарифными планами компании Aplus (рис. 3.9). В верхней части страницы Aplus приводит бесплатный номер телефона для оформления покупки или техподдержки. Щелкните мышью кнопку **Chat Now** (Поговорить сейчас), расположенную у правого нижнего края страницы, и откроется окно для интерактивного общения. Введите свой вопрос, и специалист Aplus даст ответ немедленно. Если вы всерьез подумываете подписаться на тарифный план Aplus, опробуйте обе эти возможности, так вы сможете оценить качество технической поддержки

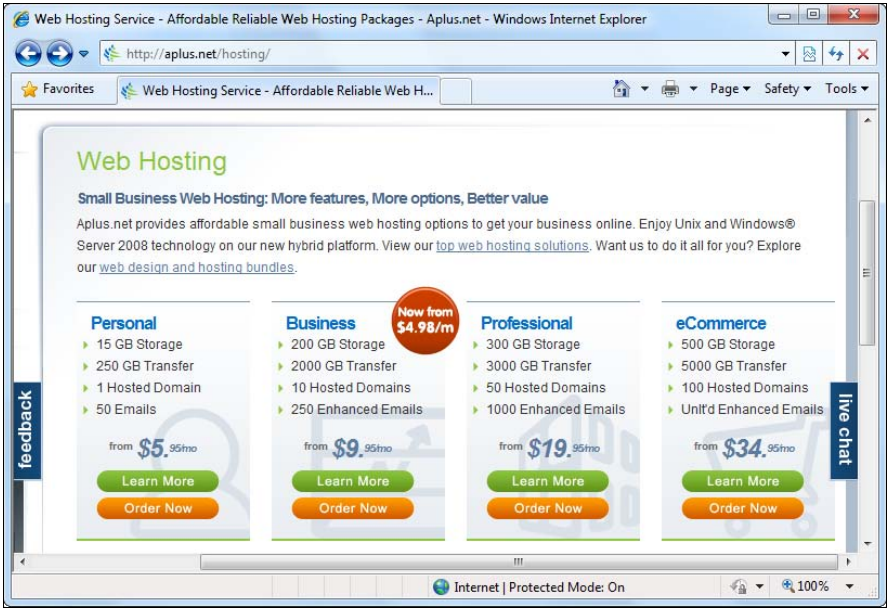


Рис. 3.9. Просмотрев все, вы выясните, что самый дешевый тарифный план — 6 долларов в месяц за 15 Гбайт дискового пространства для вашего Web-сайта и пропускную способность 250 Гбайт. Aplus включает в план бесплатное имя домена и 50 адресов электронной почты, а также поддержку FTP-протокола. Узнать дополнительные технические подробности можно, щелкнув мышью кнопку **Learn More** (Узнать больше)

В верхней части Web-страницы Aplus расположено несколько вкладок. Вкладка **Domains** (Домены) позволяет передать имеющееся имя домена или припарковать домен для использования в будущем. На вкладке **Hosting** (Хостинг) вы увидите тарифные планы хостинга компании Aplus (рис. 3.9), как раз то, что вам нужно. Вкладка **Web Design** (Web-дизайн) главным образом может быть интересна пользователям, питающим стойкое отвращение к языку HTML. На ней можно заплатить группе Web-дизайна за создание всех HTML-страниц и графических файлов для вашего Web-сайта. Но что в этом интересного?

Обзор Web-хоста № 2

В итоге поиск на Aplus.Net показал солидное предложение по разумной цене, но разборчивые Web-покупатели возможно надеются сэкономить несколько долларов или получить чуть больше места. На рис. 3.10 представлена еще одна компания Web-хостинга — Brinkster. Целевая аудитория Brinkster — создатели персональных сайтов, малый бизнес и разработчики, а не учреждения и организации. В итоге вы можете найти здесь более выгодное предложение для вашего Web-сайта.

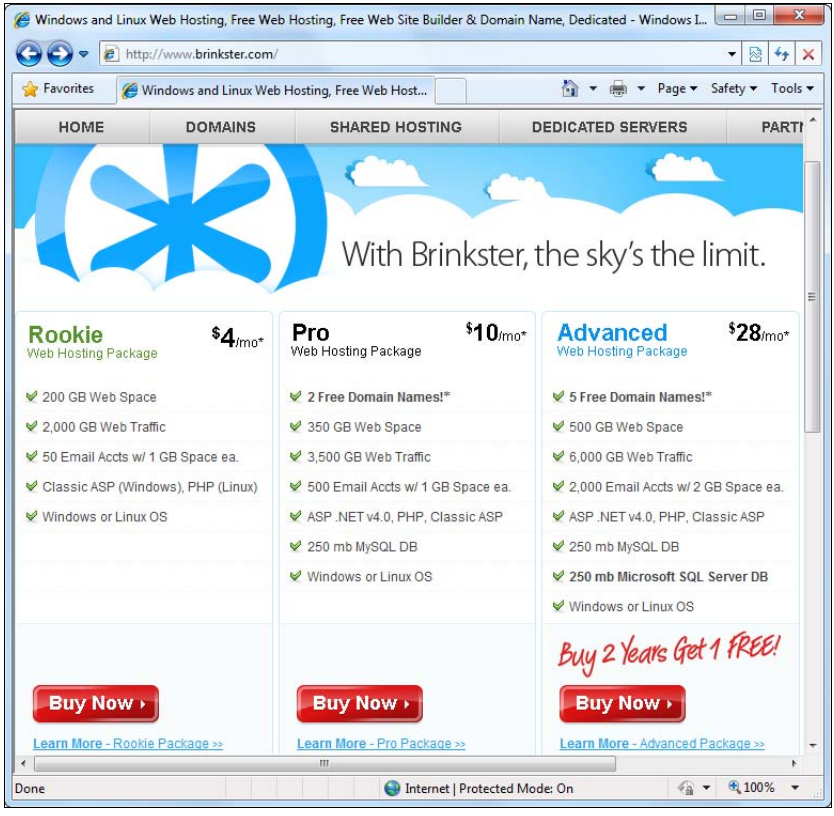


Рис. 3.10. Домашняя страница компании Brinkster сразу переходит к делу, отображая три базовых тарифных плана или пакета услуг. Пакет Rookie отлично подойдет начинающим создателям Web-сайтов. За 4 доллара в месяц он предлагает 200 Гбайт дискового пространства для гигантского Web-сайта, внушительную помощь пропускной способности (2 000 Гбайт), плюс бесплатное имя домена и адреса электронной почты. Для перехода к общению в режиме реального времени прокрутите страницу до отображения правого нижнего угла (не показанного на рисунке)

У Brinkster есть еще одно нововведение — вы можете выбрать не только вид тарифного плана, но и операционную систему на сервере, где будет обитать ваш Web-сайт. Если вы не программист, намеревающийся создавать Web-приложение, запускаемое на сервере, нет оснований беспокоиться об операционной системе, установленной на сервере. Считая, что компания хостинга делает свою работу и распределяет хранящиеся у них Web-сайты по многим компьютерам, ваш сайт будет быстрым и надежным под управлением любой операционной системы. Задайте себе вопрос: когда я последний раз задумывался о том, под управлением какой операционной системы работает Web-сайт eBay (Windows) или Amazon (Linux)?

Присмотревшись повнимательнее, вы обнаружите в верхней части страницы Brinkster вкладку **Dedicated Servers** (Выделенные серверы). Это предложение премиум-класса предоставляет в ваше распоряжение целый сервер. Крупные компании пользуются этим типом хостинга для обеспечения железобетонной стабильности, но немногие согласны платить дополнительную цену, которая может быть значительно выше 100 долларов в месяц.

После обзора сайтов двух компаний Web-хостинга вы достаточно подготовлены к тому, чтобы оценить некоторые из них самостоятельно. Если же вы до сих пор чувствуете неуверенность, можете выложить свой сайт с помощью одной из компаний, о которых шла речь. От вас потребуется только пара щелчков кнопкой мыши, и через несколько часов вы окажетесь во Всемирной паутине.

Примечание

Если Web-хост обманул ваши ожидания, не паникуйте. Перейти на другой хост не трудно. Главное, что следует помнить, — когда вы меняете хост, вы по существу отказываетесь от одного сервера и размещаете сайт на другом. Скопировать ваши Web-страницы на новый сервер — это ваша задача, никто за вас это не сделает. Пока копия вашего Web-сайта есть на вашем персональном компьютере (а она должна быть), это не составит труда. Если вы несколько скептически относитесь к выбранной вами компании, понаблюдайте за ее работой 30 дней, срок, в течение которого гарантируется возврат денег.

Бесплатные Web-хосты

Все еще не выбрали в Web ни одну из сотен компаний, предлагающих Web-хостинг? Не прельщает предложение от вашего ISP с недостаточным объемом дискового пространства? Если вы решили сэкономить на помесячной абонентской плате, есть решение, но, возможно, оно не стоит полученного разочарования.

В Web есть большое количество бесплатных Web-хостов. Это компании, которые предоставляют маленькое дисковое пространство, не требуя с вас никакой платы за него. Порой это объясняется тем, что компании надеются заставить вас перейти на платный сервис, если вы превысите строгие ограничения бесплатного пакета. В других случаях они могут быть заинтересованы в доходах от рекламы. Поскольку некоторые бесплатные хосты требуют помещения на ваши страницы назойливых рекламных баннеров.

Прежде чем соглашаться на бесплатный хостинг, ознакомьтесь с проблемами, которые могут у вас возникнуть.

- *Рекламные баннеры.* Худшие бесплатные хосты заставляют помещать на ваши страницы свои рекламные объявления. Если вы согласны наполнить вашу страницу назойливой рекламой кредитных карт, можно пойти по этому пути. Если нет, ищите другое место. В итоге вполне реально найти бесплатные хосты, не навязывающие подмигивающую баннерную рекламу, поэтому не останавливайтесь на том, который поступает именно так.
- *Ненадежность.* Бесплатные хосты могут чаще находиться в неисправном состоянии, что означает периодическое исчезновение вашего Web-сайта из Всемирной паутины. Или же серверы, используемые хостом, могут тормозить из-за плохого технического сопровождения или Web-сайтов других пользователей, заставляющих ваш сайт ползти на самом малом ходу.
- *Эксплуатационные ограничения.* Некоторые бесплатные Web-хосты вынуждают соглашаться с их политикой, ограничивающей информационное содержимое, которое вы помещаете на свой сайт. Например, они могут запрещать интернет-коммерцию, продажу места для рекламных объявлений или выгрузку в Интернет файлов определенных типов (таких как музыкальные записи, кинофильмы или файлы большого объема).
- *Ограниченная техподдержка.* Многие профессиональные координаторы Web-сайтов говорят, что хорошим хост делает не громадный объем свободного места или гигантский лимит для пропускной способности, а присутствие человека у телефонного аппарата, позволяющее в любое время решать неожиданно возникающие проблемы. Бесплатные Web-хосты не могут позволить себе нанять команду профессионалов для обслуживания клиентов, поэтому вы вынуждены будете ждать помощи, если вообще ее получите.
- *Затрудненные выгрузки в Интернет*. У многих бесплатных хостов нет поддержки для легкой выгрузки файлов по FTP-протоколу (см. *следующий раздел*). Без этой услуги вы будете вынуждены применять длительную выгрузку страниц.

Несмотря на все эти возможные проблемы, многие бережливые хранители своих кошельков слепо доверяют своим бесплатным хостам (и этикетке с месячной платой 0 долларов). Если у вас есть время для экспериментов и вашему бизнесу не нужна железобетонная надежность и срочное присутствие в Web, можете опробовать несколько. Просмотрите на сайте **www.free-webhosts.com** громадный каталог бесплатных хостов, в котором подробно расписаны объемы предоставляемого дискового пространства и условия, на которых настаивают хосты. Вы также найдете тысячи пользовательских обзоров. Но помните, недобросовестные Web-хосты могут раздувать рейтинги в ими же написанных обзорах, и любой бесплатный хост способен неожиданно поменять свои предложения.

Перенос файлов

После подписки на Web-хостинг можно переносить некоторые ваши файлы на отведенное вам в Web место. Для выполнения этого упражнения воспользуемся резюме Ли Парка (Lee Park) из предыдущей главы (которое вы можете загрузить со

страницы **Missing CD** (Отсутствующий CD), находящейся по адресу **www.missingmanuals.com/cds/caw3**). Имя окончательной версии файла — resume5.htm.

FTP

Идеально, если ваш Web-хост предоставляет доступ по FTP-протоколу. С помощью FTP вы можете переносить группы файлов со своего компьютера на сервер (и в обратном направлении) во многом точно так же, как копировать файлы из одной папки в другую в Проводнике ОС Windows или в файловом менеджере Finder на компьютерах Mac.

Перед переносом файлов по FTP-протоколу необходимо выяснить адрес FTP-сервера, а также имя пользователя и пароль. Последние обычно совпадают с именем и паролем вашей учетной записи для хостинга, но не всегда.

Для выгрузки в Интернет файлов по FTP-протоколу может потребоваться отдельная FTP-программа. Но в наши дни это не обязательно. В ОС Windows включен собственный встроенный FTP-браузер, отлично выполняющий эту задачу. Вот как он работает.

1. В ОС Windows откройте программу Explorer (Проводник).

   Щелкните правой кнопкой мыши кнопку **Start** (Пуск) и затем выберите команду **Explorer** (Проводник).

2. Введите в строку адреса Explorer нужный FTP-адрес. Убедитесь, что URL начинается с **ftp://**.

   Иначе говоря, если вам нужен адрес **ftp.myhost.com**, введите URL **ftp://ftp.myhost.com**, а не **http://ftp.myhost.com**, который ошибочно отправит ваш компьютер искать Web-страницы.

3. Введите ваши регистрационные данные (рис. 3.11).

   После регистрации вы увидите папки и файлы вашего сайта на Web-сервере; вы сможете копировать, удалять, переименовывать и перемещать их во многом так же, как вы поступаете с локальными папками и файлами. Поскольку вы до сих пор ничего еще не выгружали на сервер, папка может оказаться пустой или содержать стандартный файл index.htm, выводящий сообщение "under construction" (в разработке), если кто-нибудь случайно попытается просмотреть страницу.

4. Скопируйте ваши файлы на сервер. Легче всего перетащить с нажатой кнопкой мыши файлы из открытого окна папки и поместить в FTP-окно.

5. На рис. 3.12 показаны шаги, необходимые для выгрузки файла резюме. Убедитесь в том, что оба файла, resume5.htm и связанное с ним изображение leepark.jpg, перенесены в одну и ту же папку вашего сайта.

> ***Совет***
>
> Переносить файлы можно не только с помощью перетаскивания мышью. Вы можете применять все известные упрощенные формы команд Windows, включая команды **Cut** (Вырезать), **Copy** (Копировать) и **Paste** (Вставить) из меню **Edit** (Правка) и клавиатурные комбинации <Ctrl>+<C> (Копировать) и <Ctrl>+<V> (Вставить).

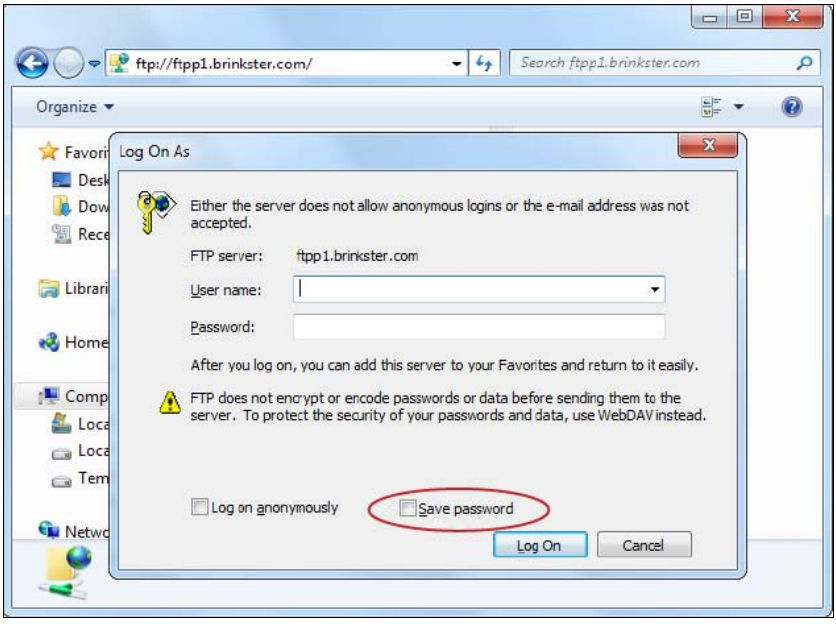


Рис. 3.11. Когда вы впервые вводите адрес FTP-сайта, Explorer в ОС Windows может попытаться зарегистрировать вас как анонима и аварийно завершиться. Затем он может предложить вам ввести имя пользователя и пароль (как показано) или отобразит сообщение об ошибке. Если увидите сообщение об ошибке, щелкните мышью кнопку **OK** и затем выберите команды меню **File | Login As** (Файл **|** Войти как). Если установить флажок (обведен) **Save password** (Сохранить пароль), не придется повторять эту процедуру при последующих визитах

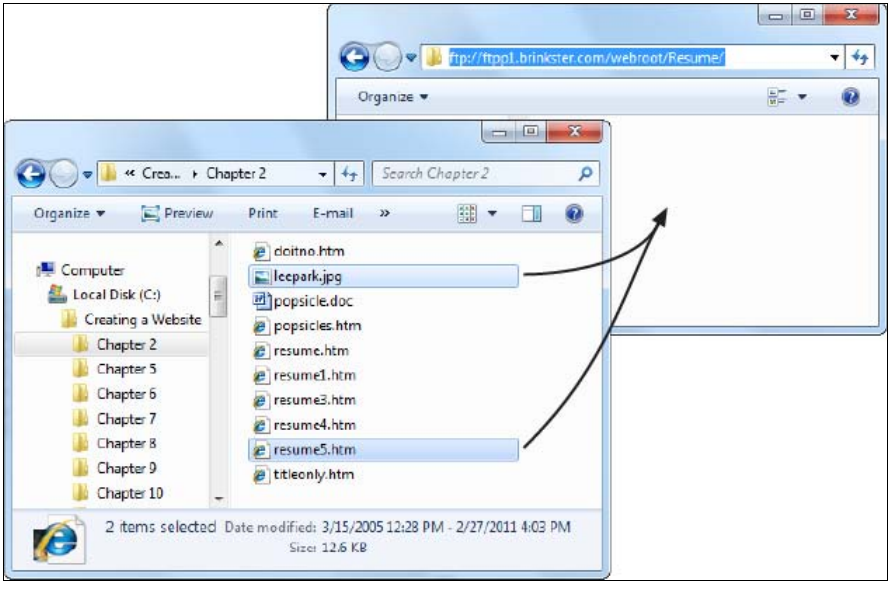


Рис. 3.12. Для вывода Ли Парка в Интернет начните с открытия окна в вашей файловой системе, используя программу Explorer (Проводник). Далее перейдите в нужную папку на вашем компьютере. Когда найдете загруженные ранее из Интернета файлы resume5.htm и leepark.jpg, выделите их и перетащите мышью в FTP-окно, чтобы начать выгрузку файлов

Если вы работаете на компьютере Mac, необходимо применить отдельную FTP-программу. К счастью, есть варианты бесплатных версий для скачивания, включая очень простую и удобную программу Rbrowser (есть на сайте **www.rbrowser.com**). Она очень похожа на своего близнеца из Windows. Сначала запустите Rbrowser. Программа спросит, хотите ли вы обновить ее до последней лицензионной версии; щелкните мышью вариант **FTP Only** (Только FTP). В следующем окне введите адрес FTP-сайта; как только вы сделаете это, Rbrowser предоставит возможность зарегистрироваться как аноним, т. е. оставить пустыми поля **Username** (Имя пользователя) и **Password** (Пароль). Далее щелкните мышью кнопку **Connect** (Подключиться) (рис. 3.13). Когда покончите с этим, можно переносить файлы, перетащив их из папки на вашем компьютере Mac в окно программы Rbrowser.

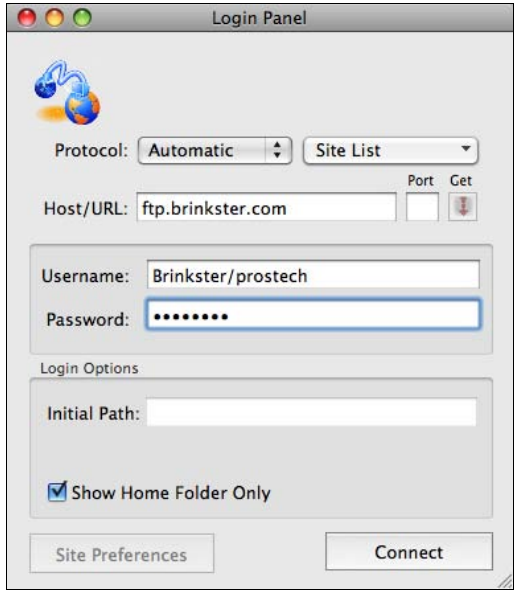


Рис. 3.13. Для регистрации в программе Rbrowser введите имя FTP-сервера (в поле **Host/URL**), а также ваше имя пользователя и пароль. Не беспокойтесь о задании начального пути; в нужную папку на вашем Web-сайте вы всегда сможете перейти после того, как установлено соединение с сервером. В заключение щелкните мышью кнопку **Connect** (Подключиться), чтобы "подтвердить сделку"

После выгрузки файла можете протестировать свою работу. Просто введите ваше имя домена и следом имя Web-страницы. Например, если вы выгрузили на ваш сайт **www.supersavvyworker.com** пример с резюме, попробуйте запросить в вашем браузере **www.supersavvyworker.com/resume5.htm**. Ждать вам не придется. После выгрузки на ваш сервер файл почти мгновенно становится доступен любому браузеру, запросившему сайт.

Выгрузка файлов с помощью браузера

Выгрузка файлов из браузера невероятно проста, но не всегда удобна. Суть в том, что вы переходите на специальную Web-страницу сайта вашего хоста, на которой задаете файлы на вашем жестком диске, предназначенные для переноса на сервер.

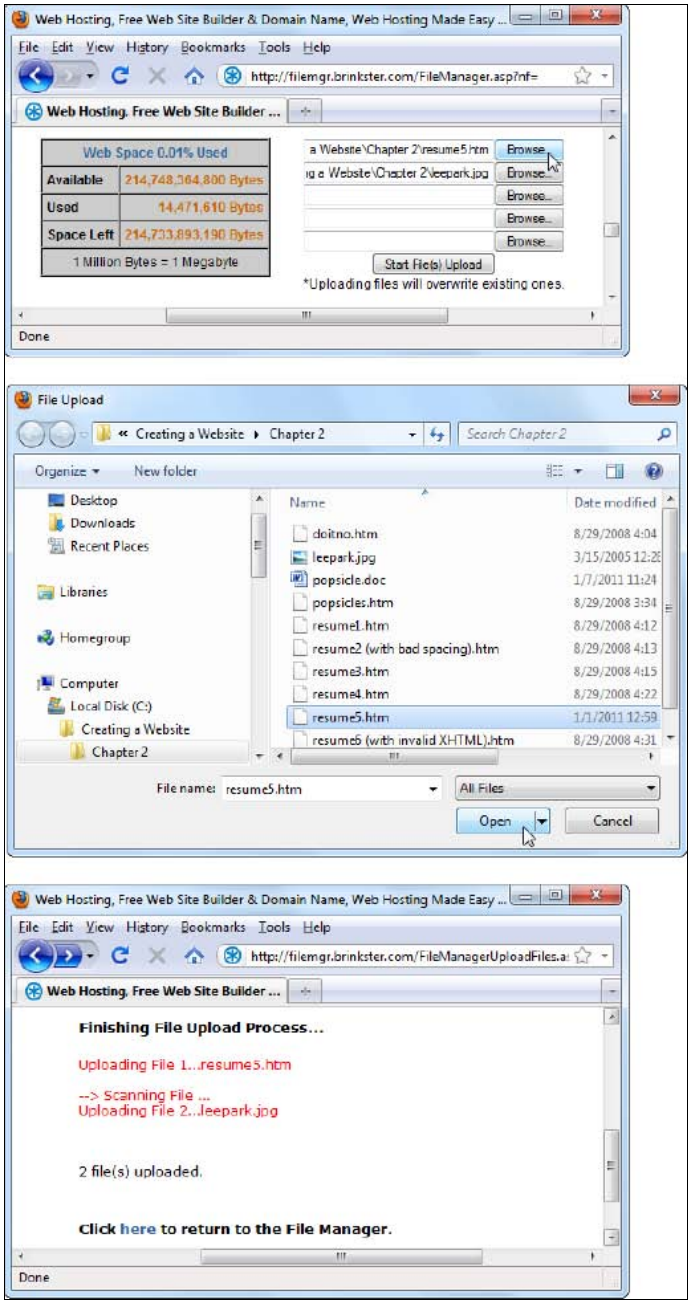


Рис. 3.14. Все сайты Web-хостов немного отличаются друг от друга, но у многих есть страница выгрузки файлов, состоящая из набора текстовых полей. Все они действуют одинаково. Сначала вы щелкаете мышью кнопку **Browse** (Обзор) (верхнее окно), которая выводит на экран диалоговое окно **Open File** (Открыть файл) (среднее окно). Найдите нужный файл, выделите его и щелкните мышью кнопку **Open** (Открыть). Если у вас несколько файлов, предназначенных для выгрузки, повторите этот процесс, используя каждый раз другое текстовое поле. Когда все нужные файлы выбраны (или просто не осталось свободных текстовых полей ввода), щелкните мышью кнопку **Start File(s) Upload** (Начать выгрузку файлов) и ждите, пока файлы не скопируются, и вы не получите сообщение с подтверждением выполнения выгрузки (нижнее окно)

Многие компании хостинга предоставляют как FTP-, так и браузерную выгрузку. Если вы пользуетесь самым дешевым тарифным планом или бесплатным Web-хостом, у вас может вообще не быть выгрузки по FTP-протоколу. Взамен вы будете выгружать ваши файлы с помощью вашего браузера.

Для выгрузки файлов из браузера посетите Web-сайт компании, предоставляющей вам Web-хостинг, и зарегистрируйтесь с именем пользователя и паролем, созданными во время оформления хостинга. Затем просмотрите по очереди пиктограммы и найдите среди них страницу управления файлами.

У всех Web-хостов разные страницы управления файлами. Некоторые выглядят как настоящая FTP-программа, часто с Flash-плагином для создания удобного интерфейса, поддерживающего одновременное перетаскивание мышью группы файлов. Другие же более традиционны и заставляют выбирать файлы по очереди (рис. 3.14).

К сожалению, возможности для ошибок при выгрузке файлов в браузере безграничны. Самая распространенная ошибка возникает, когда вы копируете сразу много файлов. Не только тратится время на выбор каждого из них, но и возрастает вероятность забыть какой-нибудь. Кроме того, могут возникнуть проблемы при попытках выгрузить файлы в разные папки или при необходимости переименования или удаления некоторых файлов после того, как вы их выгрузили.

4

ГЛАВА 4

Специализированные средства

В *главе 2* вы создали вашу первую HTML-страницу с помощью только текстового редактора и сильной нервной нагрузки — именно так начинают свою карьеру молодые талантливые Web-дизайнеры. Для того чтобы по-настоящему понять язык HTML (и выработать свой HTML-стиль), необходимо начинать с чистого листа.

Однако очень немногие авторы накрепко привязаны к простым текстовым редакторам и используют их для создания чего-то, отличного от простых текстовых страниц. Даже типовая HTML-страница состоит из множества элементов. Пытаясь написать каждый абзац, перенос строки и теги форматирования вручную, вы можете допустить ошибку в процессе создания страницы. Даже если это не так, трудно наглядно представить себе законченную страницу, если весь день смотришь на угловые скобки. Это особенно справедливо, когда имеешь дело с гораздо более сложными страницами, например с включенной графикой или многоколоночным макетом.

Оборотной стороной роста вашего профессионализма (когда редакторов Notepad (Блокнот) или TextEdit будет недостаточно) станут дополнительные расходы. Профессиональные средства для Web-дизайна могут стоить сотни долларов. В какой-то момент компании, разрабатывающие программное обеспечение, планировали включить базовые Web-редакторы в состав операционных систем, таких как Windows и Mac OS. И действительно, некоторые более ранние версии Windows поставлялись с усеченным Web-редактором FrontPage Express. (Большинство Mac OS содержат несколько урезанный редактор iWeb, ограничивающий вас готовыми шаблонами и не позволяющий вставить даже строку HTML-кода.) Но если вам нужен полнофункциональный редактор Web-страниц, способный находить ваши ошибки, помогающий вспомнить важные HTML-элементы и позволяющий управлять всем сайтом, придется искать его самостоятельно. К счастью, есть бесплатные альтернативы даже для самых безденежных творцов.

В этой главе вы узнаете, как работают редакторы Web-страниц и как выбрать из них тот, который подходит именно вам. Вы также вскоре познакомитесь с лучшими бесплатными или условно-бесплатными предложениями. Большинство редакторов Web-страниц удивительно похожи, поэтому данная глава поможет начать работу

в предпочитаемом вами редакторе, будь то Adobe Dreamweaver, Microsoft Expression Web или лакомый кусочек свободного программного обеспечения.

Выбор программного средства

Выбор таких средств, как Notepad (Блокнот) и TextEdit, совсем не плох для начала. В них просто создавать Web-страницы, и они не портят ваш HTML-код (как текстовые процессоры). Для того чтобы увидеть результат вашей работы, достаточно просто обновить окно браузера. Почему же вы должны перерасти ваш любимый текстовый редактор? По ряду причин, включая следующие.

- *Никто не совершенен.* При работе в текстовом редакторе рано или поздно вы сделаете ошибку, например, набрав `<b>` вместо `</b>`. К сожалению, вы можете ее не заметить даже при просмотре вашей страницы в браузере. Браузеры стараются изо всех сил компенсировать погрешности HTML-кода, даже если это приводит к затушевыванию реальных проблем на вашей странице. Хороший редактор Web-страниц может выделить ошибку в HTML и помочь вам ее исправить.
- ***Edit** (Правка)-**Save** (Сохранить)-**Refresh** (Обновить). И так 1 000 раз.* Текстовые редакторы хороши для маленьких страниц. Но что если вам нужно точно подобрать размер картинки или выровнять столбец в таблице? Вы должны перескакивать из текстового редактора в браузер и обратно, каждый раз сохраняя и обновляя вашу страницу, и процесс этот может длиться без преувеличения часами. В хорошем редакторе Web-страниц у вас появляются удобные инструменты, такие как тонкая настройка страницы с помощью перетаскивания объектов мышью: делаете несколько поправок, и ваш редактор соответствующим образом подправляет HTML-код. У редакторов также есть режим предварительного просмотра, который позволяет без обращения к браузеру немедленно увидеть эффект ваших корректировок на Web-странице.
- *Помогите, я тону в HTML.* Один из замечательных приемов в редакторе Web-страниц — разноцветный HTML-код. Разноцветное кодирование позволяет выделять эти противные теги в море текста. Без этого свойства вы бы часами сводили глаза "в кучку".
- *Просто наберите* `<ul><li></li></ul>`**.** Конечно, для создания маркированного списка. Надеюсь, вы еще не забыли это, не так ли? Надо признаться, что большинство Web-творцов не помнят все имеющиеся HTML-элементы. Благодаря редактору Web-страниц в этом и нужды не будет. Если вы что-то забудете, для восполнения пробела всегда есть ссылка на файл помощи или команду меню. Без средства сопровождения вы предоставлены сами себе.

Конечно, применение графического редактора Web-страниц не лишено недостатков. Вот почему вы начинали в простом текстовом редакторе, и бóльшая часть этой книги посвящена изучению языка HTML (и нескольких сопутствующих стандартов). Если вы не поймете HTML как следует, то рискуете угодить в целый ряд ловушек.

Например, вы можете применить хитроумный редактор страниц для использования изысканных шрифтов на ваших страницах. Каково же будет ваше удивление, когда

вы увидите свою работу на компьютере, у которого эти шрифты не установлены; ваша страница будет отображаться ужасным трудно читаемым шрифтом (дополнительную информацию см. в *главе 6*). Аналогичным образом редактор может нечаянно вынудить вас вставить элементы, которые приводят к ошибкам в некоторых браузерах, или графику, которая не будет корректно отображаться на других компьютерах. И наконец, даже с лучшим редактором вы потратите много времени на просмотр HTML-кода, чтобы разобраться в том, что делается, устранить путаницу или просто скопировать полезные фрагменты и вставить их на другие страницы.

Разновидности редакторов Web-страниц

Есть редакторы Web-страниц на любой вкус, но все их можно отнести к одной из следующих трех категорий.

- В *редакторах с текстовым режимом просмотра* вы должны работать с текстом и тегами чистого языка HTML. Разница между обычным текстовым редактором (таким как Notepad) и текстовым HTML-редактором только в удобстве. В отличие от Notepad (Блокнот) или TextEdit текстовые HTML-редакторы обычно содержат кнопки, позволяющие быстро вставлять часто используемые HTML-элементы или их комбинации, и предоставляют возможность одним щелчком мыши сохранить ваш файл и открыть его в отдельном окне браузера. По сути текстовые HTML-редакторы — это обычные текстовые редакторы со встроенным в них набором средств, полезных для редактирования Web-страниц. Создатели Web-сайтов предпочитают HTML-редакторы, в которых максимальное количество подобных средств, но при этом обеспечивается только текстовый режим работы.
- *Редакторы с двойным окном* также заставляют вас писать HTML-код вручную, но одновременно с набором кода в отдельном окне вы видите результаты вашей работы. Другими словами, у вас есть предварительный просмотр в режиме реального времени, поэтому не нужно все время переключаться с текстового редактора на браузер и обратно, чтобы увидеть чего вы добились.
- *Редакторы WYSIWYG (What You See Is What You Get) или полного соответствия* во многом работают как текстовые процессоры. Вместо написания HTML-тегов вы вводите текст страницы, форматируете его и вставляете картинки, точно так же, как делаете это в программе текстового процессора. Редактор WYSIWYG генерирует вашу HTML-разметку за кадром.

Любая разновидность этих редакторов — хорошая замена обычного текстового редактора. Конкретный выбор зависит от того, сколько инструментов вам нужно, как вы предпочитаете работать и сколько денег вы готовы потратить. В лучших программах редактирования Web-страниц границы между описанными разновидностями размыты и у вас есть возможность переключаться с просмотра HTML-кода в режим WYSIWYG и обратно.

Независимо от того, какой разновидностью редактора вы пользуетесь, для получения нужных вам результатов необходимо как следует выучить язык HTML. Даже если у вас есть редактор WYSIWYG, почти всегда у вас будет желание подоррек-

тировать разметку вручную. Знание хитростей HTML позволить понять, что можно сделать, а чего нельзя, а также выработать стратегию, которой нужно следовать, чтобы добиться наилучших результатов. Даже в редакторе WYSIWYG вы наверняка будете заглядывать во внутренности ваших Web-страниц.

ЧАСТО ЗАДАВАЕМЫЙ ВОПРОС

Сохранить как HTML

У моей программы текстовой обработки/верстки страниц/создания электронных таблиц есть средство сохранения документов в виде Web-страниц. Можно ли им пользоваться?

За последнее десятилетие Интернет стал повсеместно самым злободневным ключевым словом маркетинга. Любая мыслимая компьютерная программа отчаянно похваляется своими новыми Web-инструментами. Например, практически у всех современных текстовых процессоров есть средство для экспорта ваших документов в HTML. Не пользуйтесь им.

Средства экспорта в HTML работают плохо. Часто программа пытается втиснуть документ, разработанный для одного средства передачи (обычно печати) в другое (Web-пространство). Но документы текстовых процессоров отличаются от Web-страниц. У них, как правило, бóльшие отступы, изящнее шрифты, больше текста, больше свободного места вокруг текста, нет ссылок и совершенно другой макет. Когда вы экспортируете документ, ваш текстовый процессор старается сохранить эти детали в процессе создания страниц, которые при просмотре в Web-браузере получаются неудобочитаемые и совсем непривлекательные.

Другая проблема состоит в том, что средства экспорта в HTML часто формируют ужасно сложную разметку. В результате вы получите громоздкую Web-страницу, которую почти невозможно редактировать из-за обилия параметров форматирования. (В редакторе Dreamweaver даже есть средство, предназначенное для чистки разметки; загляните в команду меню **Commands | Clean Up Word HTML** (Команды | Очистить Word HTML.) И если вы захотите сделать HTML-код одной из этих страниц строже и чище, придется поработать руками.

Вывод? Если возможно, избегайте применения команды **Save as HTML** (Сохранить как HTML). Гораздо лучше скопировать и вставить содержимое вашего документа как обычный текст, а затем отформатировать его с помощью HTML-тегов собственноручно.

Поиск бесплатного редактора Web-страниц

Если вы не относитесь к немногочисленным везунчикам, обладающим самым современным редактором Web-страниц, таким как Dreamweaver или Expression Web, то, наверное, гадаете, как найти хорошую программу за минимальную цену. В конце концов, в Web можно найти массу бесплатных продуктов. Пока вы не нашли бесплатный клон коммерческой программы Dreamweaver, *можно* получить хороший базовый редактор, не открывая кошелька. Вот как это делается.

- Если вы любите проводить изыскания самостоятельно (всегда неплохая мысль) и не собираетесь устанавливать на ваш компьютер несколько десятков программ, пока не найдете подходящую, отправляйтесь на самый популярный сайт условно-бесплатного программного обеспечения **www.download.com**. Для того чтобы найти несколько кандидатов, введите в строку поиска "HTML editor" (HTML-редактор).

Примечание

Условно-бесплатное программное обеспечение (shareware) — это программное обеспечение, которое можно бесплатно опробовать и раздать друзьям. Если вам оно понравилось, вас вежливо попросят заплатить за него (или невежливо заблокируют его по истечении испытательного срока). *Бесплатное программное обеспечение* (free software) — это программное обеспечение, которое совсем ничего не стоит, и если понравится, оно ваше! Как правило, вы не получите такой мелочи, как техподдержка. Некоторые бесплатные программы поддерживаются добровольными пожертвованиями. Для того чтобы быть уверенным, что ваше условно-бесплатное программное обеспечение не заражено вирусами или шпионскими программами, загружайте его с проверенных сайтов, таких как **www.download.com**.

- Можно также просмотреть обширный список редакторов Web-страниц в Википедии по адресу **http://en.wikipedia.org/wiki/Comparison_of_HTML_editors**. Этот список заслуживает упоминания благодаря включенной в него исчерпывающей информации. Например, в нем приводится цена каждого редактора и операционная система, под управлением которой он работает. И самое важное: в нем указаны редакторы, которые можно получить бесплатно.

Вы быстро поймете, что существует масса бесплатных редакторов. У многих из них ужасное, корявое оформление кнопок и командных меню. В некоторых есть явные ошибки. Найти подходящий вам удастся не сразу.

В следующих разделах описаны четыре заслуживающих внимания редактора, которые не будут вам стоить ни копейки.

BlueGriffon

BlueGriffon — остроумный, небольшой редактор, которым невероятно легко пользоваться. Самое замечательное, что у BlueGriffon есть версии для всех трех основных операционных систем: Windows, Mac OS X и Linux.

У BlueGriffon тот же механизм визуализации, что и у браузера Firefox. Это продукт с открытым программным кодом (open source), т. е. вы можете не только бесплатно скачать его из Интернета и скопировать, но если вы программист, то можете просматривать его исходный код и предлагать любые усовершенствования.

Как человеку, увлеченному Web, вам, скорее всего, понравятся множественные режимы отображения страниц, позволяющие вносить изменения как в текстовое представление **Source** (Исходный код), так и во внешнее представление WYSIWYG, визуализирующее ваши страницы (рис. 4.1).

У BlueGriffon несколько запутанная история. Большинство его разработчиков сначала, до 2006 г., трудились над его предшественником, популярным бесплатным редактором Nvu. Когда работа над Nvu прекратилась, другая группа программистов использовала Nvu как основу для сделанного на скорую руку редактора Web-страниц KompoZer. Но теперь на сцене BlueGriffon, превзошедший оба упомянутых редактора.

Главный недостаток BlueGriffon в его новизне и быстрой эволюции. С другой стороны, это позволит вам получить крепкую поддержку новых средств языка HTML5. (На Web-сайте BlueGriffon регулярно предоставляются бета-компоновки самых

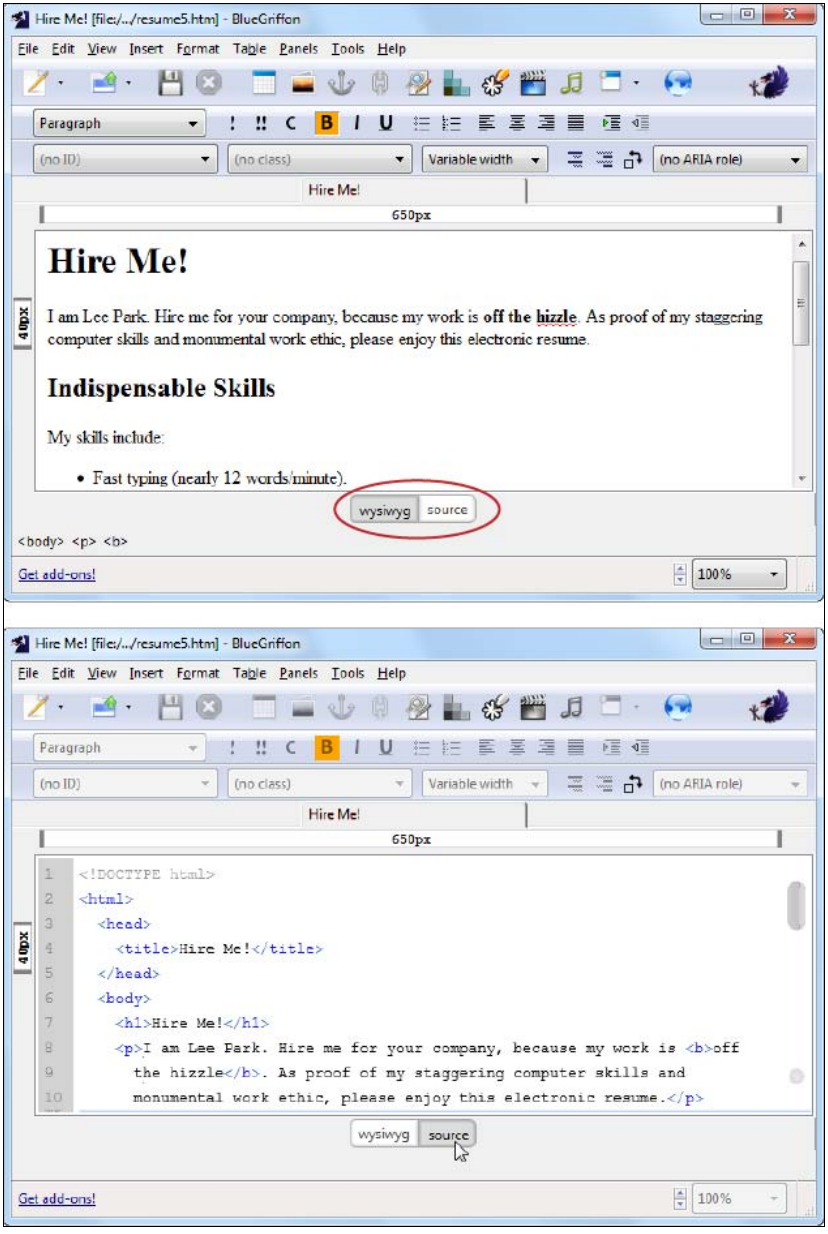


Рис. 4.1. *Верхнее окно*: вместо работы с HTML-кодом представление WYSIWYG редактора BlueGriffon позволяет форматировать текст так, как будто вы работаете в текстовом процессоре. Для переключения из одного режима просмотра в другой используйте кнопки в нижней части окна (обведены). *Нижнее окно*: для тонкой настройки вашей разметки перейдите в режим просмотра **Source** (Исходный код). BlueGriffon предложит пронумерованные для удобства строки кода и выделение цветом HTML-элементов

последних, непротестированных версий редактора, в которые встроены новейшие средства; ищите их по ссылке сайта **Fresh Meat** (Все самое свежее).) Определенный недостаток — вероятность ошибок и несущественных глюков. Во время написания книги текущей была версия BlueGriffon 0.8, указывающая на то, что группа разработчиков еще не добилась соблюдения стандартов стабильности и надежности, соответствующих профессиональной версии с номером 1.0.

Для скачивания редактора BlueGriffon или чтения последних новостей о нем посетите Web-сайт **www.bluegriffon.org**.

> ***Совет***
>
> В BlueGriffon, как во многие более усовершенствованные (и более дорогие) Web-редакторы, включена проверка орфографии. Если BlueGriffon находит незнакомое слово (как "hizzle" на рис. 4.1), он подчеркивает его красной чертой. Затем вы можете щелкнуть его правой кнопкой мыши и выбрать нужный вариант из списка предлагаемых исправлений.

Amaya

Amaya — редактор Web-страниц с настоящей историей. Он существует более десяти лет. Этому способствует то, что World Wide Web Consortium, солидная международная организация по стандартизации, помогающая выработать стандарты в Web, поддержала Amaya. Как и BlueGriffon, редактор Amaya работает в трех больших операционных системах, что само по себе редкость. Большинство бесплатных редакторов, включая те два, о которых вы скоро узнаете, предназначены только для компьютеров под управлением Windows. Возможности представления Web-страниц в BlueGriffon лучше, чем в Amaya (см. рис. 4.2).

Amaya не так отшлифован, как профессиональный редактор страниц, и поиск нужной вам команды в его загроможденных меню и на панелях инструментов часто становится тяжелой задачей. Отсутствие свежих обновлений также огорчает, особенно поклонников HTML5, которым непременно нужно вставить новые теги на место их собственных. И все же для бесплатного редактора Amaya — очень удачное приобретение. Чтобы опробовать Amaya, посетите сайт **www.w3.org/Amaya.**

HTML-Kit

HTML-Kit появился как немного экстравагантный редактор Web-страниц для Windows с видом экрана, который мог понравиться только матушке Билла Гейтса. Та версия, теперь именуемая HTML-Kit 292, все еще существует. Она бесплатная, относительно надежная и нелепо настраиваемая. Но компания-разработчик сейчас предлагает модернизированную версию, названную по непонятной причине HTML Kit Tools (59 долларов). В новой версии более приятный интерфейс и лучше поддержка HTML5.

В отличие от BlueGriffon и Amaya, в HTML-Kit нет режима редактирования WYSIWYG. Вместо него есть редактор с предварительным просмотром, т. е. вы можете просматривать внешний вид вашего HTML-документа в процессе его редактирования. Для загрузки бесплатной версии (или пробной новой предлагаемой за плату) посетите Web-сайт **www.htmlkit.com.**

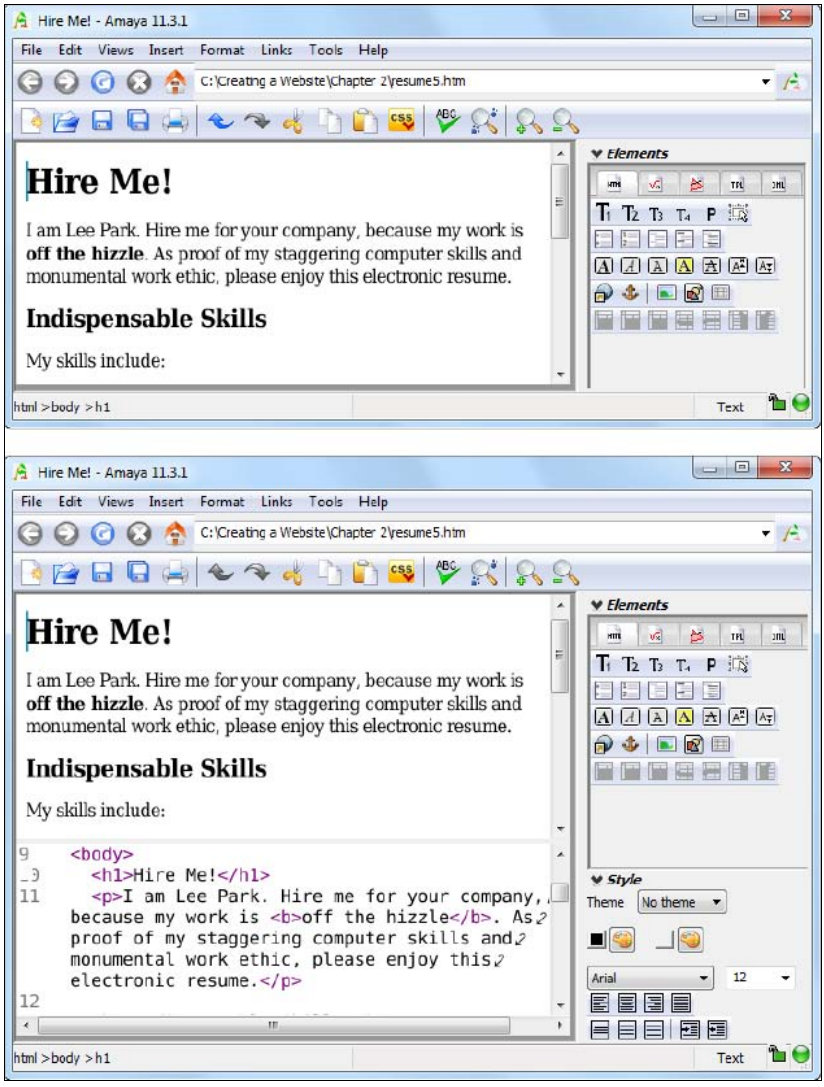


Рис. 4.2. *Верхнее окно*: откройте Web-страницу в Amaya и вы попадете в режим просмотра **Design** (Конструктор).
Нижнее окно: выберите в меню команды **Views | Show Source** (Представления | Показывать код), и Amaya отобразит разделенное окно с разными режимами просмотра, позволяющее перейти из представления **Design** (Конструктор) к вашей HTML-разметке. Самое главное: вы можете вносить изменения в любом окне. Если у вас широкоэкранный монитор, необязательно выбирать разделение на верхнее и нижнее окно; выберите команды **Views | Split View Vertically** (Представления | Разделить на представления вертикально) и два представления страницы расположатся рядом друг с другом

CoffeeCup Free HTML Editor

CoffeeCup Free HTML Editor — урезанная версия полнофункционального продукта для ОС Windows с одноименным названием (за исключением слова "Free" (бесплатный)). В полную версию (49 долларов) включены и текстовый режим, и режим

WYSIWYG, а в бесплатной версии режим WYSIWYG отключен. Получить копию можно на сайте **www.coffeecup.com/free-editor**.

Профессиональные HTML-редакторы

Надоело примериваться к маломощным редакторам Web-страниц и их интерфейсам, спроектированным, по-видимому, М. К. Эшером? Если вы готовы перейти к профессиональным пакетам для Web-дизайна, соберитесь с духом, — ваш выбор удивительно прост. Сегодня на рынке программных средств вы найдете только два действительно высококлассных редактора.

- Adobe Dreamweaver предпочитают дизайнеры-графики и закоренелые HTML-мастера. Напичканный инструментами он обеспечивает тончайший контроль каждого HTML-компонента. Поскольку этот редактор — лидер на рынке и инструмент, выбираемый большинством профессионалов, у него внушительный ценник: 400 долларов за стандартную версию.
- Microsoft Expression Web — полностью переработанная версия долгожителя Microsoft, редактора FrontPage. Он предлагает многие из тех мощных средств, которые есть в Dreamweaver за менее чем полцены (150 долларов во время написания книги). Если вы не собираетесь производить впечатление на толпу дизайнеров в черных водолазках, это лучшее вложение ваших денег.

> ***Примечание***
>
> Как ни странно, у Microsoft Expression Web есть дальний родственник с большинством тех же самых инструментов. У него длинное, в стиле Microsoft название Microsoft Office SharePoint Designer. Как и Expression Web, SharePoint Designer — потомок устаревшего FrontPage. Но в отличие от Expression Web у него есть необязательные средства, работающие с Microsoft SharePoint, программой, ориентированной на работу в Web и используемую крупными компаниями для совместной обработки документов и сотрудничества их персонала. Если вы покупаете редактор Web-страниц корпорации Microsoft за свои деньги, вы почти наверняка остановите выбор на Expression Web. Но если у вас уже есть доступ к SharePoint Designer (скажем, в компании, где вы работаете), он может стать вполне равноценной заменой.

Одна из причин, по которой эти продукты гораздо лучше своих конкурентов, состоит в том, что они содержат множество инструментов, которые вам наверняка понадобятся, когда вы начнете разрабатывать Web-страницы. Например, оба позволяют создавать таблицы стилей (современное средство, о котором вы узнаете в *главе 6*), изменять размеры изображений и перемещать их по вашим страницам и управлять всем сайтом целиком. Expression Web даже содержит инструмент для генерации привлекательных кнопок.

Другая причина — ими чертовски легко пользоваться. Несмотря на то что у обоих редакторов множество сложных инструментов, редактировать простой HTML-файл невероятно легко.

В прошлом Dreamweaver считался достаточно сложным, чтобы отпугнуть начинающих Web-дизайнеров. С другой стороны, FrontPage (предшественник Expression

Web) славился легкостью использования, но при этом обладал несколькими дурными особенностями — вставлял ненужные элементы в ваш код или полагался на хитрости, которые срабатывали, только если ваш Web-сервер поддерживал серверные расширения FrontPage. Однако последние версии обеих программ побороли эти слабости. Теперь Dreamweaver фактически так же легко использовать, как Expression Web, а Expression Web такой же хорошо продуманный и гармоничный, как Dreamweaver. На самом деле типовые задачи в обеих программах решаются удивительно похоже. Вывод? Вы не прогадаете, выбрав любую из них.

Если все еще сомневаетесь, для спокойствия испытайте 30-дневные пробные версии обоих продуктов. С Web-сайта **www.microsoft.com/Expression/try-it** скачайте бесплатную пробную версию Expression Web, а с сайта **www.adobe.com/products/dreamweaver** — работающую демоверсию Dreamweaver. А для тщательного изучения всех инструментов Dreamweaver посмотрите книгу Dreamweaver CS5: The Missing Manual.

Совет

Ищете лучшее и при этом стараетесь сэкономить? Adobe и Microsoft предлагают радующее снижение цен на обновления. Что касается Adobe, чтобы получить скидку, вы должны обладать предыдущей версией Dreamweaver или GoLive, а Microsoft предоставляет скидку при наличии купленной в розницу версии программ Adobe Creative Suite, Microsoft Expression или Microsoft Office. В противном случае у вас есть возможность найти у специализированных розничных продавцов академическую (или "студент и преподаватель") версию обеих программ. Эти версии урезаны, но вполне функциональны и у них цены гораздо ниже. Для получения права на них нужно доказать, что вы студент или сотрудник высшего учебного заведения.

Работа в HTML-редакторе

После того как редактор Web-страниц выбран, познакомимся с ним. Компании-разработчики программного обеспечения потратили последнее десятилетие на копирование инструментов своих конкурентов и в результате самые распространенные операции в Expression Web, Dreamweaver и многих бесплатных редакторах Web-страниц удивительно похожи. Независимо от того, какой программой вы пользуетесь, в следующих разделах вы научитесь основным приемам: созданию типового HTML-документа и превращению его в интерактивный непосредственно в вашем удобном редакторе. Когда вы будете свободно ориентироваться в вашем редакторе, можно двигаться дальше и узнать больше о том, как работает язык HTML.

Примечание

Несмотря на то что в последующих главах вы не найдете пошагового руководства для любого из этих редакторов, дополнительные быстрые клавиши, приемы и методы для вашего любимого редактора ищите во врезках и советах. Единственное исключение — крайне важное средство работы с шаблонами страниц, имеющееся как в Dreamweaver, так и в Expression Web, будет рассматриваться в *главе 10*.

Запуск

Первый шаг для запуска вашего редактора Web-страниц — двойной щелчок кнопкой мыши нужной пиктограммы на рабочем столе или быстрый переход к меню **Start** (Пуск) (на компьютерах с Windows) и последовательность команд **Finder | Applications** (Finder | Программы) (для Mac OS).

Некоторые редакторы, такие как Amaya, запускают окно с советом дня. Закройте это окно, чтобы перейти в главное окно программы.

После того как программа запущена, можно открыть файл и начать редактирование. Обычно это означает применение последовательности команд **File | Open** (Файл | Открыть), как и в любой другой уважающей себя программе. Для испытания вашего редактора откройте один из файлов-примеров, с которыми вы работали в *главе 2* (также доступными на странице **Missing CD** (Отсутствующий CD) по адресу **www.missingmanuals.com/cds/caw3**). В примерах данной главы используется файл resume5.htm.

Множественные режимы отображения

Как вы уже знаете, HTML-документ в зависимости от вашего желания можно просматривать разными способами, например, получить удобный макет, подобный создаваемому текстовым процессором, или полностью управлять страницей непосредственно из HTML-разметки. Большинство редакторов Web-страниц предлагают выбрать режим отображения и позволяют быстро переключаться из одного режима в другой. Для включения режимов отображения страницы нужно найти ряд кнопок, обычно расположенных над или под документом, с которыми вы работаете. Рисунок 4.3 поможет обнаружить эти кнопки. Единственное исключение — редактор Amaya. У него нет этих удобных кнопок и переключить режим отображения можно только командами меню **Views** (Представления).

Большинство редакторов запускаются в режиме WYSIWYG, отображающем отформатированный HTML-документ, или приблизительное представление того, как будет выглядеть страница в Web-браузере. Когда вы переключаетесь в режим просмотра исходного HTML-кода, вы видите реальную версию, уже знакомое отображение в текстовом режиме выделенных цветом тегов и текста. Эти режимы просмотра представляют два основных элемента HTML-редактирования. Но, возможно, самый лучший вариант — разделенное окно, отображающее оба представления страницы одновременно. Вы чаще всего будете его использовать для редактирования HTML-тегов и одновременного предварительного просмотра результатов. Можно работать и иначе: редактировать представление WYSIWYG и смотреть, какие теги вставляет ваш редактор, — это чудесный способ выучить язык HTML.

Некоторые редакторы предлагают интересные гибридные режимы просмотра. Например, у BlueGriffon есть средство **HTML Tags** (HTML-теги), отображающее HTML-теги в перемещаемых желтых прямоугольниках. Чтобы увидеть их, переключитесь в режим WYSIWYG и выберите команды **View | All Tags Mode** (Вид | Все теги) (рис. 4.4). Amaya предлагает два специализированных режима

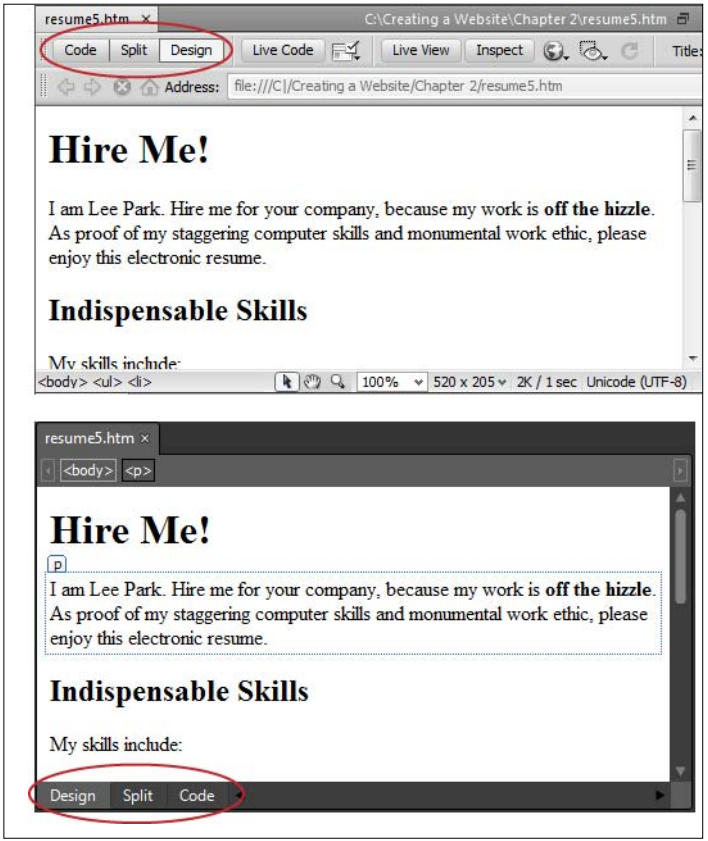


Рис. 4.3. Несмотря на разные имена и порядок размещения, большинство редакторов Web-страниц, включая Dreamweaver (вверху), Expression Web (внизу) и BlueGriffon (см. рис. 4.1), применяют похожие кнопки для переключения режимов просмотра или отображения разных представлений страницы

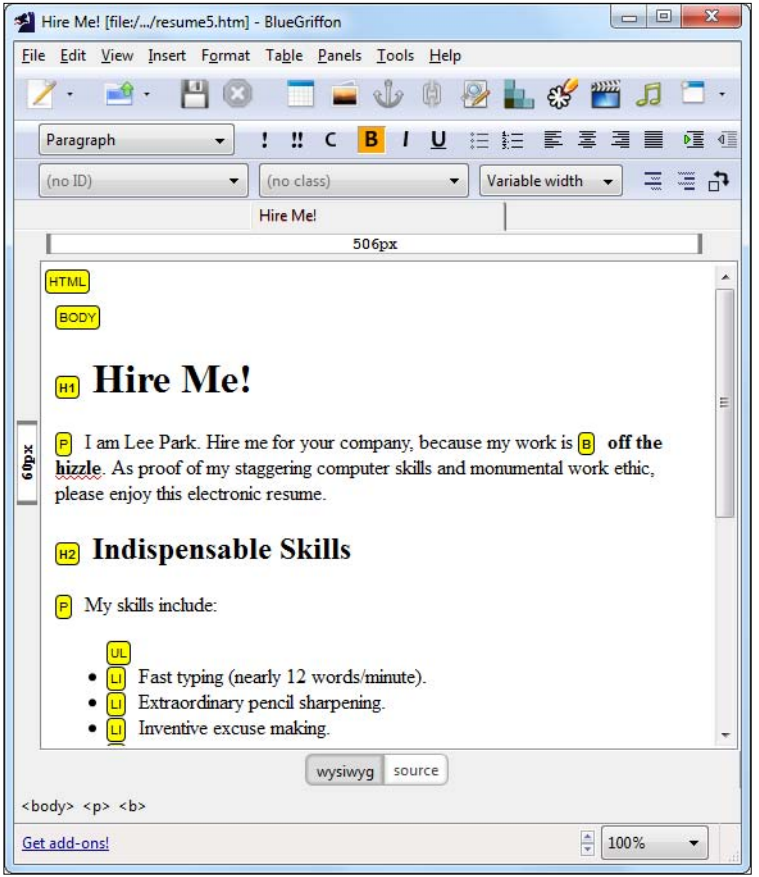


Рис. 4.4. Многие редакторы Web-страниц поставляются с полезными дополнительными инструментами. В BlueGriffon последовательность команд **View | All Tags Mode** (Вид | Все теги) вставляет желтые прямоугольники, отображающие ваши теги в представлении WYSIWYG. Если увидите какую-то ошибку, можно перейти в режим просмотра **Source** (Исходный код) и исправить ее

просмотра. Вы можете посмотреть структуру вашей страницы целиком с помощью команд **Views | Show Structure** (Представления | Показать структуру) или отобразить все ссылки на другие Web-страницы (см. *главу 8*) с помощью команд **Views | Show Links** (Представления | Показать ссылки).

Создание Web-страницы в режиме отображения исходного кода

Лучший способ понять, как работает ваш редактор Web-страниц — создать новый HTML-документ. Начнем с работы в режиме **Code** (Исходный код) и попробуем воспроизвести один из вариантов резюме, разработанных вами в *главе 2* (лучший вариант показан на рис. 2.10 и в тексте с кодом к нему).

Откройте программу и выполните следующие действия:

1. Для создания Web-страницы выполните команды **File | New** (Файл | Создать).

   Некоторые редакторы Web-страниц начинают с вывода пустой страницы, в этом случае пропустите все приведенные далее инструкции. Но большинство открывают окно вида **New Document** (Новый документ), в котором вы выбираете тип создаваемого файла.

2. Выберите тип страницы, которую хотите создать.

   Многие Web-редакторы предлагают набор готовых макетов страниц, чтобы помочь вам приступить к делу. Например, в Dreamweaver и Expression Web можно создать страницу с разными многоколоночными макетами (дополнительную информацию см. в *главе 9*). Лучше отказаться от них, пока вы не научитесь создавать собственные оригинальные макеты. Сейчас начнем с простейшей пустой HTML-страницы. В Dreamweaver выберите **Blank Page** (Пустая страница) и в списке **Page Type** (Тип страницы) задайте **HTML**. В Expression Web выберите группу **General** (Общие) и затем задайте **HTML.**

   Многие редакторы также позволяют выбрать тип документа, `doctype` (см. *разд. "Определение типа документа"*), для вашей новой Web-страницы. В Dreamweaver выбор делается в списке **DocType** в окне **New Document** (Новый документ). В Amaya используется похожий список **Document Profile** (Профиль документа) в окне **New XHTML Document** (Новый XHTML-документ). В Expression Web нужно выполнить дополнительное действие: сначала щелкните кнопкой мыши ссылку **Page Editor Options** (Параметры редактора страниц) в окне **New** (Создать), а затем в открывшемся окне **Page Editor Options** (Параметры редактора страниц) найдите список **Document Type Declaration** (Объявление типа документа). В BlueGriffon, чтобы выбрать `doctype`, вместо команд **File | New** (Файл | Создать) необходимо выбрать команды **File | New Wizard** (Файл | Мастер создания). Если вы не нашли нужную вам настройку в вашем редакторе, не волнуйтесь, после создания страницы вы легко сможете отредактировать `doctype` вручную.

3. Создайте документ.

Для того чтобы внесенные вами изменения были приняты и был создан документ, щелкните мышью кнопку **Create** (Создать) или **OK** (в зависимости от программы). Редактор отобразит ваш новый пустой документ, содержащий голый HTML-каркас. В режиме отображения **Code** (Исходный код) вы увидите базовые элементы `<html>`, `<head>` и `<body>`.

Убедитесь, что вы в режиме **Code** (Исходный код) и таким образом у вас есть полный контроль над вашей HTML-разметкой. Введите все теги и текст резюме от начала до конца так, как вы бы делали это в редакторах Notepad (Блокнот) или TextEdit. (Вы можете заглянуть в *главу 2* и вспомнить основные правила написания HTML-кода.) В процессе вы заметите несколько инструментов-ускорителей. Например, когда вы начинаете набирать тег в Dreamweaver или Expression Web появляется раскрывающееся меню с предлагаемыми вариантами. Вы можете выбрать подходящий HTML-тег из меню или продолжить набор. Кроме того, когда вы вставите начальный тег в элемент-контейнер (например, `<h1>` для заголовка), многие Web-редакторы автоматически добавят конечный тег (`</h1>`), так что вы не забудете его.

Создание Web-страницы в режиме WYSIWYG

Создание и форматирование страницы в режиме отображения WYSIWYG — гораздо более интересная и трудная задача, т. к. вам нужно знать, где в вашем редакторе искать различные параметры форматирования. Dreamweaver и Expression Web выручают, поместив большое количество элементов управления HTML-кодом прямо на свои панели инструментов. Для вставки элемента в режиме WYSIWYG вы сначала выбираете фрагмент текста, который хотите отформатировать, а затем щелкаете мышью соответствующую кнопку на панели инструментов. Далее можно перейти в режим отображения **Code** (Исходный код) и убедиться, что получен ожидаемый результат. Например, для выделения фрагмента текста жирным шрифтом выделите текст и щелкните мышью кнопку с меткой **B** на панели инструментов. За кадром (в вашу настоящую HTML-разметку) редактор вставит тег `<b>` непосредственно перед выделенным фрагментом и тег `</b>` сразу после фрагмента.

На рис. 4.5 показаны самые полезные кнопки панели инструментов в редакторе BlueGriffon. Но независимо от применяемого редактора вам придется потратить немного времени на просмотр панелей инструментов и меню, чтобы найти самый удобный способ получения результатов.

> ***Совет***
>
> Вам решать, создавать страницу в режиме **Code** (Исходный код) или в режиме WYSIWYG. В WYSIWYG всегда работать быстрее и комфортнее, особенно поначалу, но вдобавок к будущим осложнениям, возможно, придется проверять и просматривать большой объем HTML-кода.

Для того чтобы попрактиковаться в редактировании в режиме WYSIWYG, создайте заново одну из страниц-примеров из *главы 2* (отлично подойдет мини-резюме из *разд. "А где же все картинки?"*) . Вместо ввода тегов вручную введите текст и от-

форматируйте его с помощью кнопок панели инструментов или команд меню вашего редактора.

Создайте документ, как описано ранее, и выполните следующие действия:

1. Перейдите в режим WYSIWYG и введите заголовок "Hire me!" (Возьмите меня на работу!)

   Текст на экране будет обычного размера.

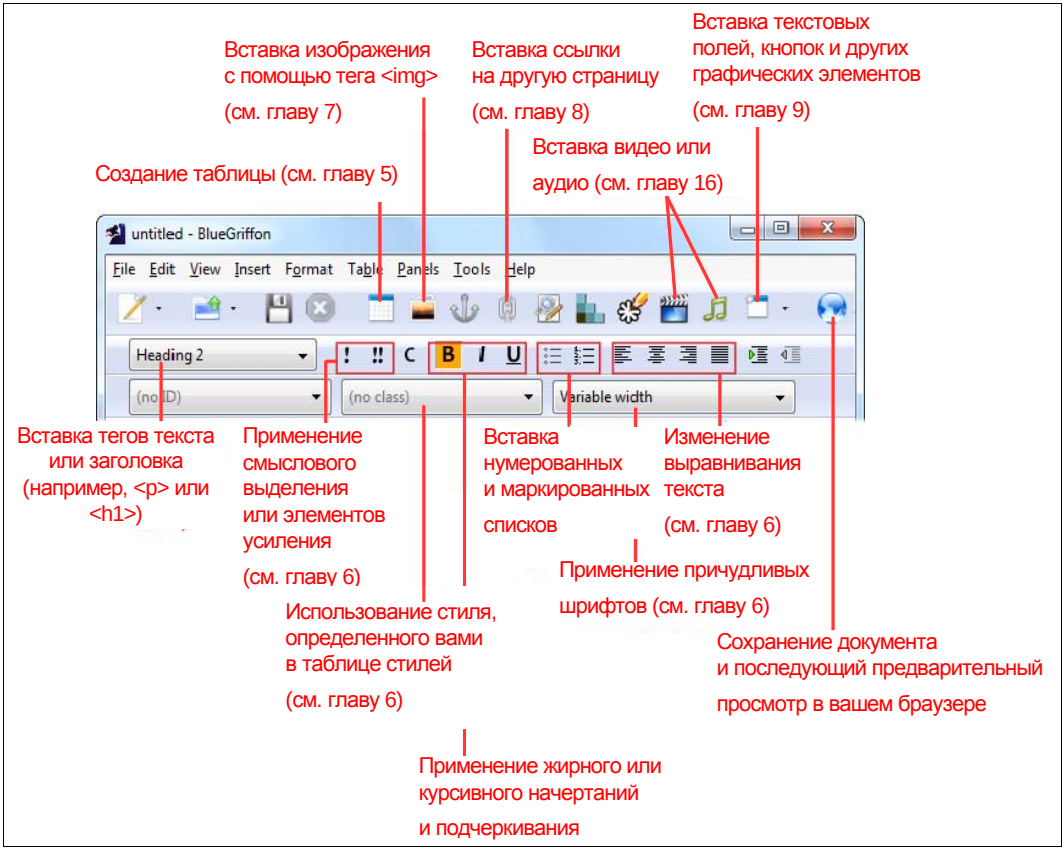


Рис. 4.5. BlueGriffon предлагает множество HTML-инструментов в одном месте. Но помните, есть разница между обнаружением инструмента на панели и реальным умением его применять. Чтобы узнать все хитрости и ловушки, на которые вы натолкнетесь при использовании изображений, шрифтов и таблиц, продолжайте чтение

2. Выделите текст и затем найдите на панели инструментов или в меню вариант, преобразующий текст в заголовок с помощью вставки тегов `<h1>` и `</h1>`.

   Для разметки текста вам потребуется более близкое знакомство с вашим редактором Web-страниц, командами его меню и кнопками на панелях инструментов.

   В Expression Web (и BlueGriffon) самый быстрый способ задать заголовок первого уровня — выбрать строку **Heading 1** из раскрывающегося списка **Style**

(Стиль). В Dreamweaver у вас два стоящих варианта: можно применить комбинацию клавиш <Ctrl>+<1> или использовать раскрывающийся список в окне **Properties** (Свойства) (если не найдете окно, выполните команду **Window | Properties** (Окно | Свойства)). В Amaya нужно щелкнуть мышью пиктограмму **T1** на панели, напоминающей панель инструментов, которая расположена в главном окне справа.

3. Нажмите клавишу <Enter> для перехода к следующему абзацу.

Размер шрифта станет обычным, и вы сможете набрать оставшийся текст документа. Следующая ваша задача — создание маркированного списка. В Dreamweaver, Expression Web, BlueGriffon и Amaya на панели инструментов есть кнопка для создания маркированных списков, если вы, конечно, сумеете ее разыскать.

В дебрях тегов и себя-то легко потерять. Для облегчения ориентировки Expression Web, Dreamweaver, BlueGriffon и Amaya содержат панель быстрого выбора элемента над или под вашим документом (рис. 4.6).

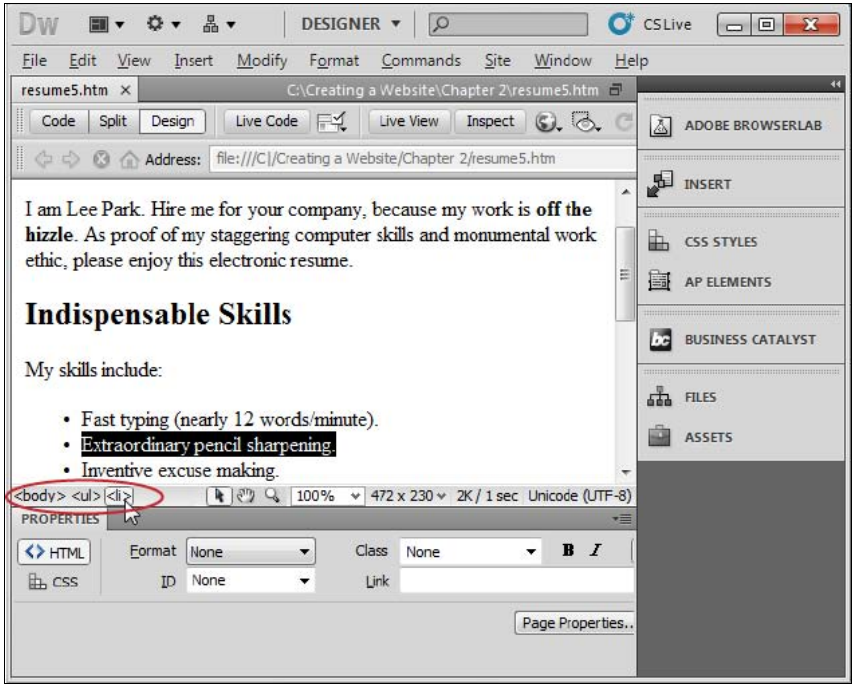


Рис. 4.6. Вы поймете, что селектор элементов очень удобен, когда потребуется редактировать ваш HTML-код. После того как найдете на странице фрагмент HTML, который нуждается в редактировании, дважды щелкните кнопкой мыши селектор элементов, чтобы подтвердить, что вы в нужном месте. Селектор перечислит все действующие элементы в текущем фрагменте. В данном примере вы поставили курсор внутрь элемента <b> (для применения жирного шрифта), который находится внутри элемента <li> (обозначающего пункт списка), который вложен в элемент <ul> (начало маркированного списка), в свою очередь вложенного в элемент <body>, который содержит контент всего HTML-документа. Вы можете быстро выбрать элемент, щелкнув кнопкой мыши его тег на панели селектора

4. Для разнообразия попробуйте вставить изображение.

   Кнопка **Image** (Изображение) в редакторе BlueGriffon позволяет сделать это очень легко. В Dreamweaver и Expression Web есть аналогичные кнопки, но легче сразу перейти в меню: в Dreamweaver и Amaya выберите **Insert | Image** (Вставка | Изображение), а в Expression Web — **Insert | Picture | From File** (Вставка | Рисунок | Из файла).

   ПРИМЕЧАНИЕ

   В этом примере следует поместить изображение в тот же каталог, что и Web-страницу. В противном случае некоторые редакторы могут вставить элемент `<img>`, связанный с конкретным местом на вашем жестком диске. Это проблема: Web-посетители не смогут получить доступ к вашему жесткому диску и поэтому не увидят изображение. Для того чтобы еще раз проверить, все ли в порядке, просмотрите элемент `<img>` в режиме HTML и убедитесь, что атрибут `src` не начинается с префикса `file:///`. Если же это так, отредактируйте его вручную, чтобы в атрибуте `src` было задано только имя файла, как в элементе `<img>`, который применялся в *разд. "А где же все картинки?" главы 2*.

5. Когда ваш редактор попросит вас выбрать файл с изображением, найдите файл leepark.jpg и выделите его. (Это изображение можно скачать со страницы **Missing CD** (Отсутствующий CD) по адресу **www.missingmanuals.com/cds/caw3**.)

   Программа добавит в ваш HTML-код соответствующий элемент `<img>`. Вставив изображение, вы по достоинству оцените режим WYSIWYG. Во всех описанных здесь редакторах вы можете с помощью мыши перемещать изображение или менять его размер.

Управление Web-сайтом

Большинство редакторов Web-страниц не ограничивают вас работой с одной страницей. Dreamweaver, Expression Web, BlueGriffon и Amaya позволяют открыть несколько документов одновременно и переключаться на любой из них с помощью вкладок в верхней части окна документа (рис. 4.7). Это особенно удобно, если вам нужно внести одно и то же изменение в целый набор страниц, или вы хотите вырезать фрагмент контента с одной страницы и вставить его на другую.

Помимо возможности одновременного редактирования нескольких страниц большинство редакторов позволяют управлять вашим Web-сайтом целиком. В следующих разделах описана эта процедура в редакторах Dreamweaver и Expression Web. В обоих случаях первый шаг — задание папок и файлов вашего сайта, чтобы редактор знал, какие именно документы формируют ваш сайт.

ПРИМЕЧАНИЕ

Web-сайт — это просто коллекция из одной или нескольких Web-страниц с сопутствующими файлами (например, файлами изображений). Часто в редакторе Web-страниц удобно управлять всеми этими файлами вместе. В некоторых программах при создании Web-сайта становятся доступны дополнительные инструменты, такие как шаблоны, проверка ссылок и выгрузка Web-сайта в Интернет.

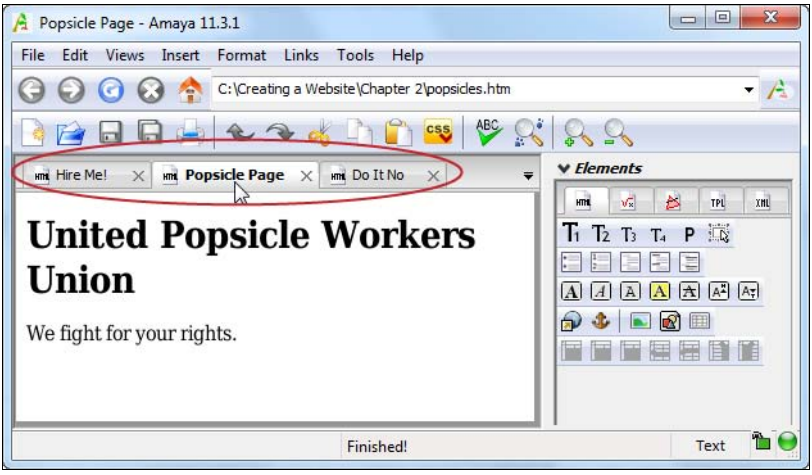


Рис. 4.7. Ваш редактор Web-страниц создает новую вкладку каждый раз, когда вы открываете файл. Вы можете переключаться на нужную страницу, щелкнув кнопкой мыши заголовок вкладки, обычно находящийся над страницей (и под строкой меню редактора и его панелями инструментов). В данном случае в Amaya открыты три страницы одновременно. Для того чтобы закрыть документ, щелкните мышью **X** справа от заголовка вкладки

Задание сайта в Expression Web

Для создания Web-сайта в Expression Web выполните следующие действия:

1. Выберите команды **Site | Open Site** (Сайт | Открыть сайт)**.**

   На экране появится диалоговое окно **Open Site** (Открыть сайт) (рис. 4.8)**.**

2. Щелкните мышью кнопку **Browse** (Обзор), чтобы найти папку Web-сайта.

   Открытие Web-сайта подобно поиску файла, за исключением того, что выводятся только папки, а не имена файлов.

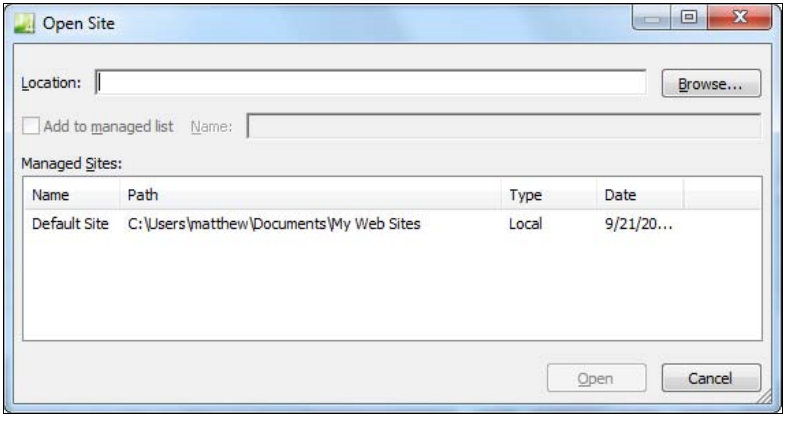


Рис. 4.8. В диалоговом окне **Open Site (**Открыть сайт) перечисляются все Web-сайты, о которых знает Expression Web. При первом запуске Expression Web этот список будет пустым (за исключением практически бесполезного стандартного Web-сайта, который создается при первоначальной установке Expression Web). Для обнаружения одного из ваших сайтов следует щелкнуть мышью кнопку **Browse** (Обзор)

3. Найдите папку, которую хотите открыть, выделите ее и щелкните мышью кнопку **Open** (Открыть).

   Можно загрузить папку Chapter 2 со страницы **Missing CD** (Отсутствующий CD) (**www.missingmanuals.com/cds/caw3**), чтобы получить готовую папку и набор файлов (файлы резюме, с которыми вы раньше работали).

   Щелчок мышью кнопки **Open** (Открыть) вернет вас в диалоговое окно **Open Site** (Открыть сайт). Если хотите запомнить местоположение этого Web-сайта, чтобы быстрее открыть его в следующий раз, установите флажок **Add to managed list** (Добавить в управляемый список). После этого Expression Web включит папку в свой список **Managed Sites** (Управляемые сайты).

4. Для загрузки вашего Web-сайта в Expression Web щелкните мышью кнопку **Open** (Открыть).

   После этого редактор отобразит вкладку **Site View** (Отображение сайта) со списком всех файлов вашего сайта (рис. 4.9).

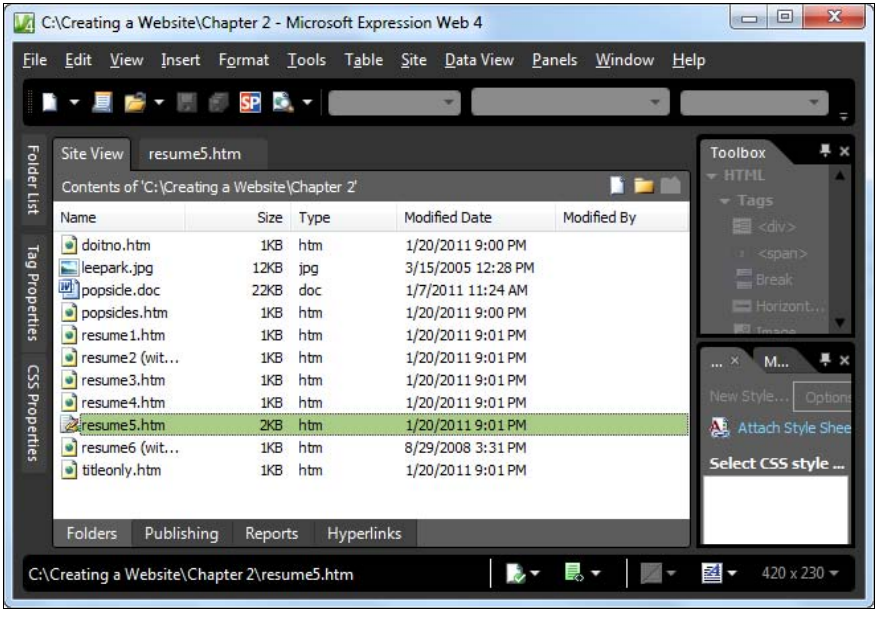


Рис. 4.9. Когда вы открываете папку Web-сайта (в данном случае папка C:\Creating a Website\Chapter 2), Expression Web открывает вкладку, на которой отображаются все файлы из этой папки. Вы можете применять к этим файлам базовые команды работы с файлами. Например, двойной щелчок файла правой кнопкой мыши открывает меню с командами переименования или удаления файла. Вы также можете дважды щелкнуть страницу левой кнопкой мыши и открыть ее для редактирования. Под вкладкой **Site View** (Отображение сайта) расположены кнопки, позволяющие опубликовать ваш Web-сайт, создать отчет или проверить ваши ссылки

5. Добавьте папки метаданных Expression Web. Для этого выберите команды **Site | Site Settings** (Сайт | Параметры сайта), а затем установите **Maintain the website using hidden metadata files** (Сопровождение Web-сайта с помощью скрытых файлов метаданных) и щелкните мышью кнопку **OK.**

Многим инструментам управления сайтом в Expression Web требуются данные сопровождения, которые программа хранит в скрытых подпапках. Но Expression Web не создает эти подпапки автоматически; вы должны дать согласие на их создание.

Слово *метаданные* означает "данные о данных". Другими словами, папки метаданных в Expression Web хранят данные о данных вашего Web-сайта. Если интересно, эти подпапки с именами _private, _vti_cnf и _vti_pvt можно увидеть в Windows Explorer (Проводник). (Акроним VTI — сокращение от Vermeer Technologies Inc., названия компании, первоначально создавшей FrontPage, а затем продавшей его корпорации Microsoft.)

У этих папок несколько назначений. Во-первых, они хранят сведения о том, какие файлы вы выгрузили на ваш Web-сервер. Такое отслеживание невероятно облегчает обновление Web-сайта, поскольку Expression Web передает на ваш сервер не целый сайт, а только те файлы, которые были изменены. В папках также хранится информация о страницах вашего сайта и ресурсах, которую Expression Web использует для очень удобных инструментов, таких как проверка ссылок (см. *разд. "Средства проверки ссылок" главы 8*) и шаблоны (см. *разд. "Создание нового шаблона страницы" главы 10*).

> ***Совет***
>
> Воспринимайте папки с метаданными как некую скрытую водопроводно-канализационную сеть. Их нужно иметь для функционирования некоторых инструментов, но после того, как вы их создали, можно больше о них не думать.

Выгрузка сайта в Expression Web

Одно из самых популярных средств Expression Web позволяет выгружать ваш Web-сайт в Интернет без отдельной FTP-программы, передающей файлы сайта на ваш сервер. Прежде чем воспользоваться этим преимуществом, необходимо выполнить несколько действий, перечисленных далее, чтобы сообщить Expression Web, что ваша папка с Web-страницами представляет полный Web-сайт. Затем можно обновлять ваш сайт с помощью следующих действий:

1. Выберите **Site | Publishing** (Сайт | Публикация) (или щелкните мышью кнопку **Publishing** (Публикация) в нижней части вкладки **Site View** (Отображение сайта)).

   Вы не сможете опубликовать ваш Web-сайт, пока не введете основные данные о вашей компании, предоставляющей Web-хостинг.

2. Щелкните кнопкой мыши крупную ссылку **Add a publishing destination** (Добавить местонахождение публикации), расположенную в средней части вкладки **Site View** (Отображение сайта).

   На экране появится окно **Connection Settings** (Параметры подключения) (рис. 4.10).

3. Введите данные, сообщающие Expression Web, как подключиться к вашему серверу, и щелкните мышью кнопку **Save** (Сохранить).

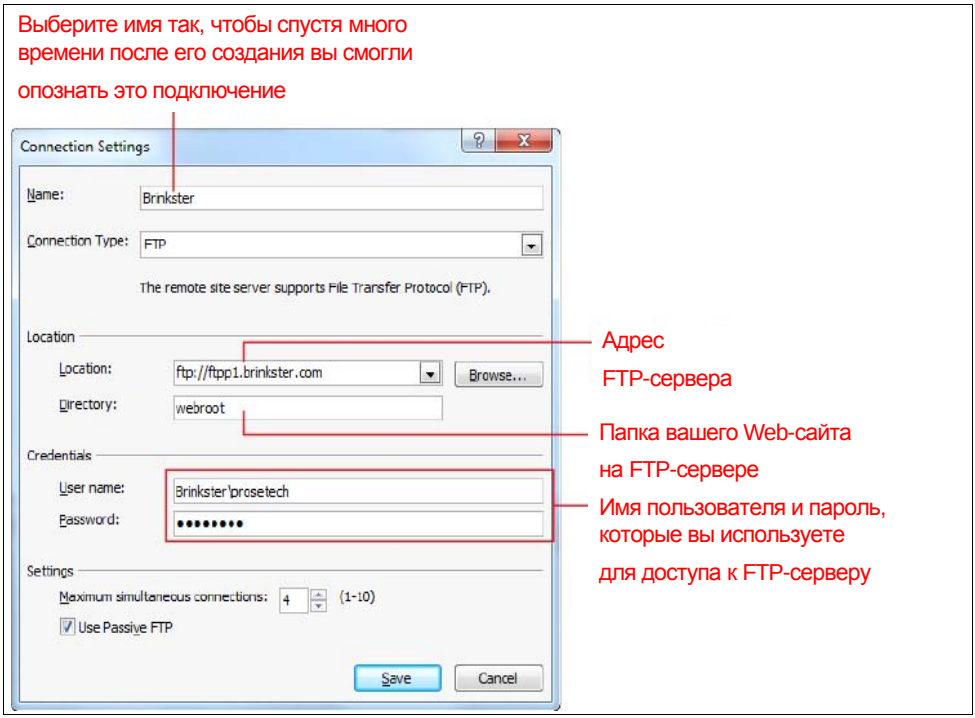


Рис. 4.10. Точные параметры, задаваемые в окне **Connection Settings** (Параметры подключения), должна сообщить вам компания, предоставляющая Web-хостинг. Обычно необходимо указать адрес FTP-сервера вашего хоста, имя каталога (папки) вашего Web-сайта на сервере и ваши имя пользователя и пароль. Эту форму придется заполнить один раз. Если все в порядке, Expression Web воспользуется этими данными при следующей публикации вашего сайта (хотя пароль придется вводить каждый раз)

Expression Web сохранит данные и подключится к вашему Web-хосту. В следующий раз, когда вы будете использовать средство публикации в Expression Web, вы увидите вместо ссылки **Add a publishing destination** (Добавить местонахождение публикации) ссылку **Connect to publishing destination** (Подключиться к местонахождению публикации). Для подключения щелкните ее один раз кнопкой мыши и все.

После того как соединение установлено, Expression Web отображает рядом для сравнения содержимое Web-сайта, хранящееся на вашем компьютере, и содержимое, размещенное на сервере, что позволяет сразу увидеть, какие файлы были изменены (рис. 4.11).

4. Для обновления содержимого на вашем Web-сервере выберите команды **Site | Publish Changed Files** (Сайт | Опубликовать измененные файлы).

   Эти команды запускают процесс публикации (рис. 4.12).

Вы также можете переносить отдельные файлы с помощью стрелок, расположенных между двумя списками файлов. Для передачи файлов с вашего компьютера на сервер выделите их в левом списке и щелкните мышью стрелку, направленную вправо. Для выгрузки файла с вашего сервера выделите его в правом списке и

щелкните мышью стрелку, направленную влево. Двунаправленная стрелка ниже (**Synchronize files** (Синхронизировать файлы)) выполняет обе операции одновременно; просматривается каждый выделенный файл и проверяется, нет ли его старых версий на вашем компьютере или на Web-сервере.

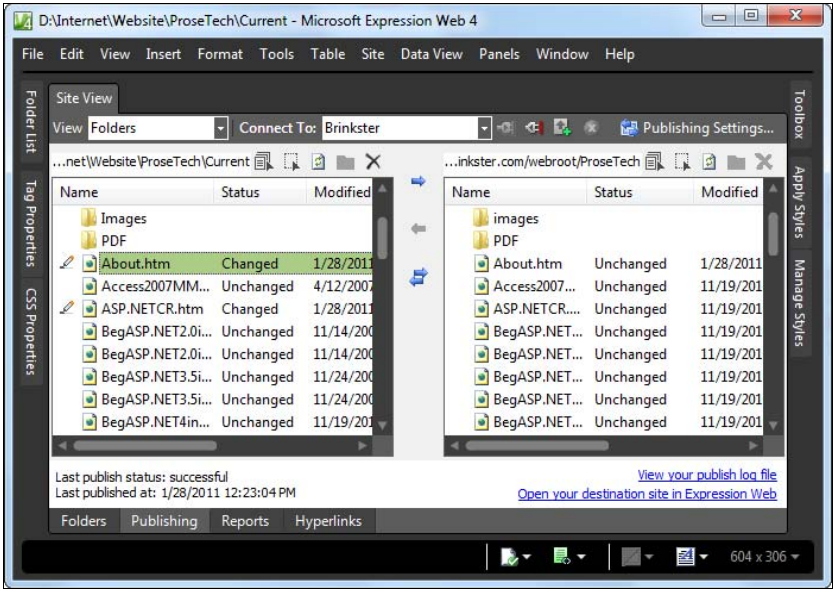


Рис. 4.11. Редактор Expression Web выделил два файла: About.html и ASP.NETCR.htm, которые вы обновили на своем компьютере, но не на Web-сервере

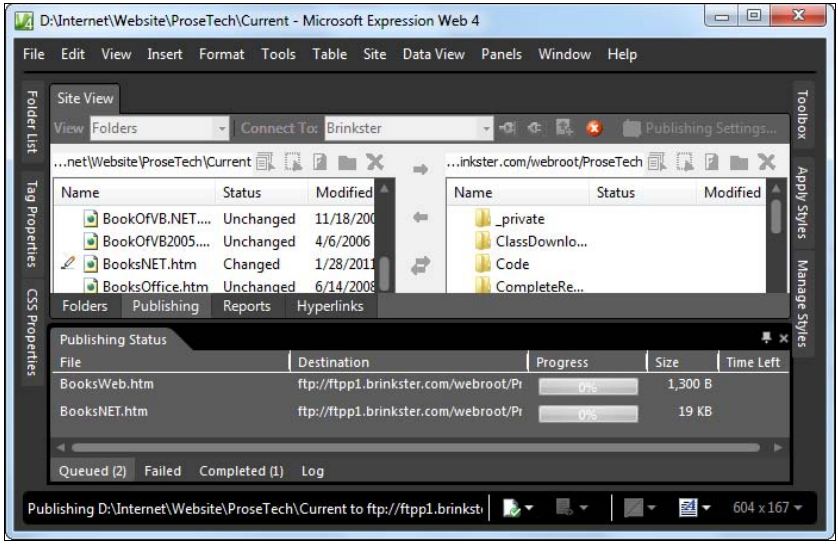


Рис. 4.12. Когда вы публикуете Web-сайт, Expression Web просматривает ваши файлы и копирует только те, которые были добавлены или изменены с момента последней публикации сайта. Индикатор выполнения указывает имя копируемого файла и время, необходимое для выполнения операции

Задание сайта в Dreamweaver

Dreamweaver предлагает два способа работы с Web-сайтом. Простейший — использовать панель **Files** (Файлы) для просмотра файлов в любой папке на вашем компьютере (рис. 4.13).

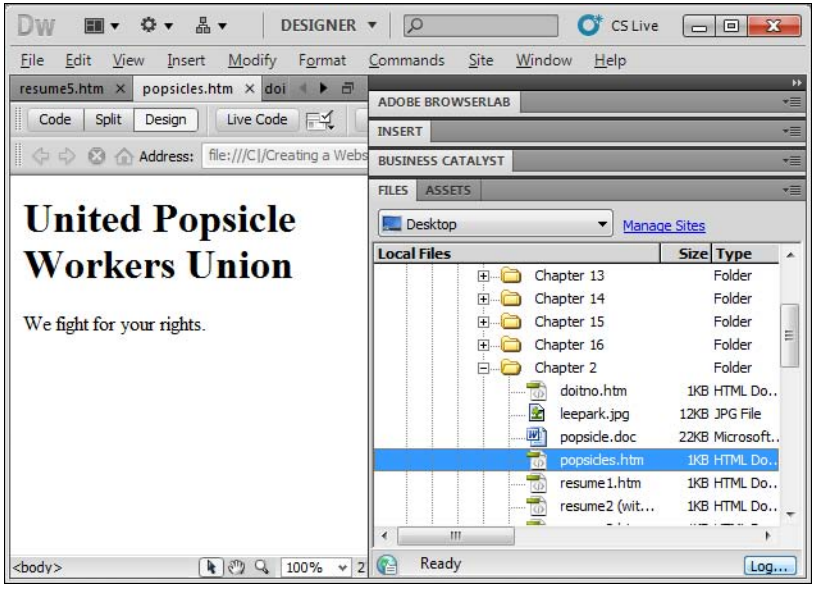


Рис. 4.13. С помощью панели **Files** (Файлы) редактора Dreamweaver вы можете просматривать все файлы вашего сайта, не покидая программу

Хотя использовать панель **Files** (Файлы) удобно, этот способ очень ограничен. Проблема в том, что Dreamweaver не может сообщить, какие файлы и папки формируют ваш Web-сайт. Для поддержки выгрузки Web-сайта и нескольких других инструментов необходимо задать папку в качестве Web-сайта. Вот как это делается.

1. Щелкните кнопкой мыши ссылку **Manage Sites** (Управление сайтами) на панели **Files** (Файлы) или выберите команды **Site | Manage Sites** (Сайт | Управление сайтами).

   Dreamweaver выводит на экран диалоговое окно **Manage Sites** (Управление сайтами) с перечнем всех Web-сайтов, которые вы настраивали до данного момента. Первоначально этот список пуст.

2. Для задания нового сайта щелкните мышью **New** (Создать).

   Dreamweaver запускает окно **Site Setup** (Настройка сайта), в которое вы вводите данные о вашем новом сайте.

3. Введите описательное имя в поле **Site Name** (Имя сайта).

   Имя сайта используется только для отслеживания вашего сайта, поэтому можно задать любое. В данном примере задано имя LeeParkSite. Имя сайта также выводится на панели **Files** (Файлы).

4. В поле **Local Site Folder** (Локальная папка сайта) введите полный путь к папке вашего Web-сайта (обычно что-то похожее на C:\Creating a Website\Chapter 2).

   Если вы не знаете, где именно находится папка вашего сайта, можно щелкнуть кнопкой мыши пиктограмму папки рядом с текстовым полем, чтобы просмотреть дерево папок и найти нужную. Можно также ввести путь к папке, которой еще нет; Dreamweaver создаст новую пустую папку, чтобы вы могли начать создавать свой новый Web-сайт.

5. На левой панели щелкните кнопкой мыши строку **Servers** (Серверы). Затем щелкните мышью маленький значок "+" в левом нижнем углу списка Web-серверов (рис. 4.14).

   На экране появится диалоговое окно, в которое вы введете параметры вашего подключения (рис. 4.15).

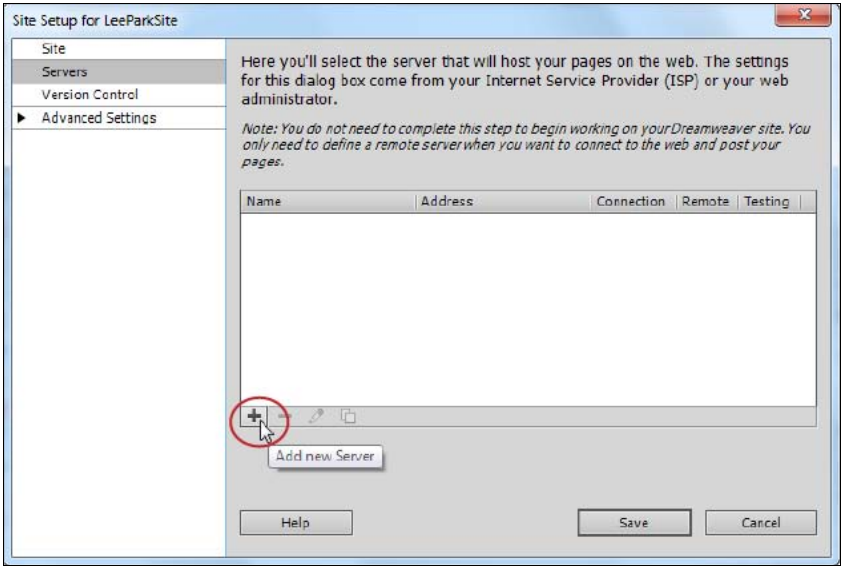


Рис. 4.14. Первоначально список Web-серверов Dreamweaver пуст. Вам нужно ввести данные вашего провайдера Web-хостинга

6. Когда введете все данные о вашем Web-сервере, щелкните мышью кнопку **Save** (Сохранить).

   Вы вернетесь в окно **Site Setup** (Настройка сайта). Dreamweaver сохранит данные о вашем подключении, поэтому снова их вводить не придется.

7. Еще раз щелкните мышью кнопку **Save** (Сохранить), чтобы закрыть окно **Site Setup** (Настройка сайта).

   Вы окажетесь в диалоговом окне **Manage Sites** (Управление сайтами).

8. Щелкните мышью кнопку **Done** (Выполнить).

   Вы вернетесь в главное окно программы Dreamweaver.

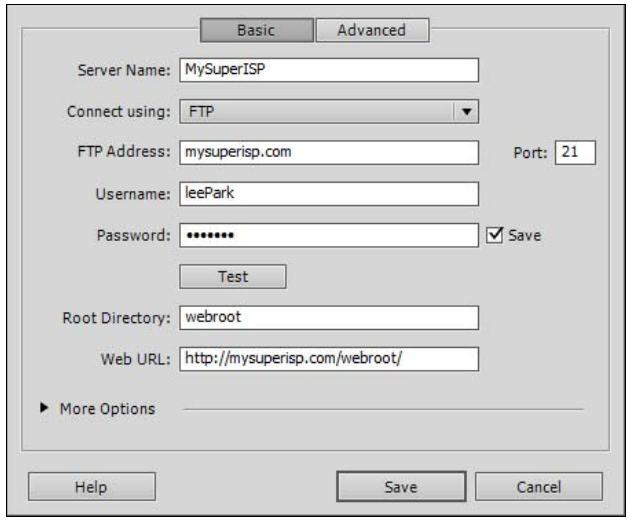


Рис. 4.15. Ваш Web-хост может снабдить вас всей необходимой информацией о соединении. К важным деталям относятся адрес вашего FTP-сервера, ваши имя пользователя и пароль и корневой каталог (имя папки) вашего Web-сайта на сервере

Обновление сайта в Dreamweaver

После того как вы задали ваш Web-сайт, вы увидите его на панели **Files** (Файлы) и сможете просматривать файлы на вашем удаленном Web-сервере или переносить файлы с вашего компьютера на сервер и обратно.

Для переноса файлов с вашего локального компьютера на ваш сервер применяется операция, на FTP-жаргоне называемая *выгрузкой*. Она выполняется следующим образом:

1. На панели **Files** (Файлы) выберите ваш Web-сайт из раскрывающегося меню, расположенного вверху слева.
2. Выберите строку **Local view** (Локальный компьютер) из раскрывающегося меню, находящегося вверху справа.

   На панели **Files** (Файлы) будут перечислены файлы на вашем компьютере (рис. 4.16).

   Совет

   Первоначально список файлов втиснут в небольшую область в углу окна Dreamweaver. Если хотите распространить эту область на все окно и видеть файлы на вашем локальном жестком диске и ваши файлы на Web-сервере, щелкните мышью кнопку расширения, показанную на рис. 4.16. Она сформирует окно, похожее на вкладку **Site View** (Отображение сайта) в Expression Web.

3. Выделите файлы, которые хотите перенести на Web-сервер.

   Можно выделить сразу несколько файлов, держа нажатой клавишу <Ctrl> и одновременно щелкая кнопкой мыши имя каждого файла.

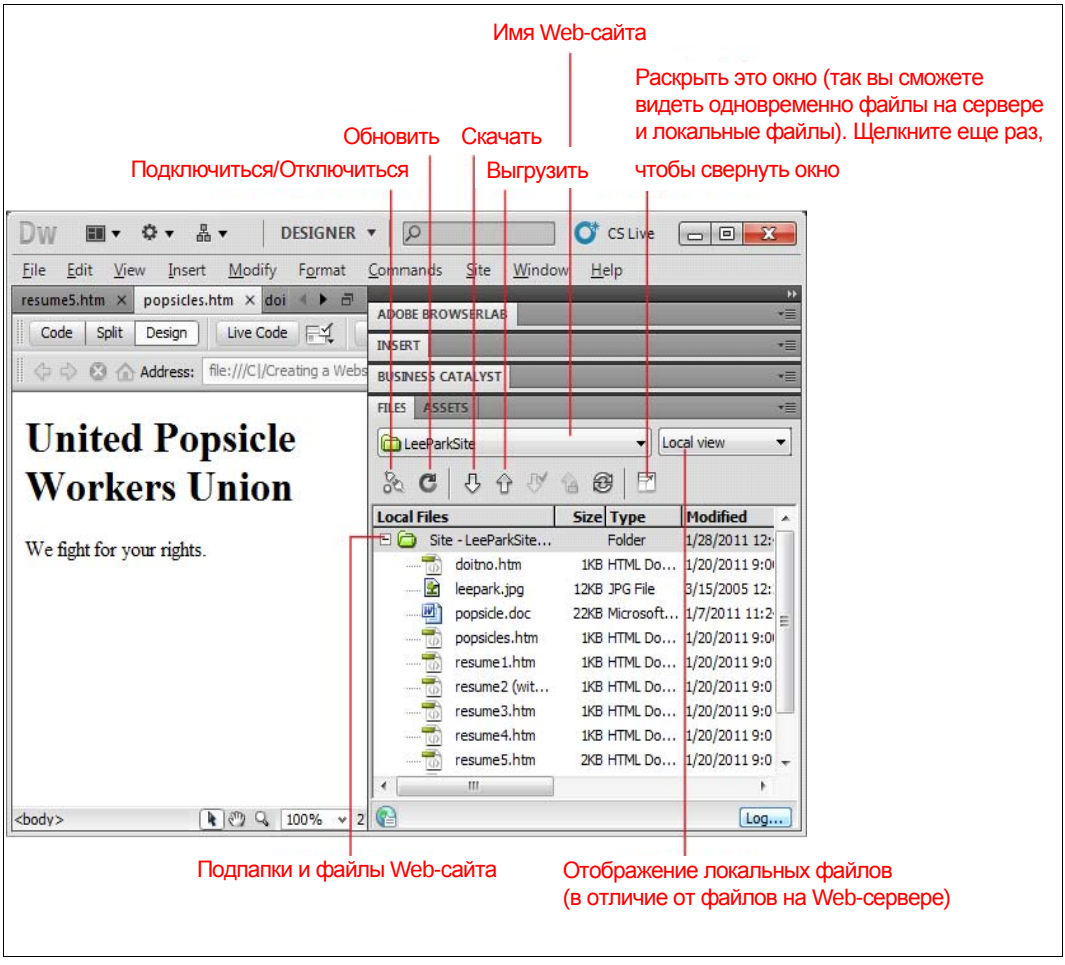


Рис. 4.16. Это локальное представление Web-сайта, названного LeeParkSite. В него включен список всех файлов из папки сайта на вашем компьютере. С помощью пиктограмм этого окна можно быстро передавать файлы на ваш сервер и обратно

4. Щелкните кнопкой мыши стрелку **Put** (Выгрузить) (голубая стрелка, направленная вверх) или щелкните правой кнопкой мыши файлы, а затем выберите команду **Put** (Выгрузить) в раскрывающемся меню.

   Dreamweaver спросит, хотите ли вы скопировать зависимые файлы.

5. Выберите кнопку **Yes** (Да), если хотите скопировать все файлы, связанные со страницами вашего сайта. Например, если вы выгружаете в Интернет страницу с элементами `<img>` для отображения графики, щелкните мышью кнопку **Yes**, чтобы Dreamweaver выгрузил не только сами страницы, но и графические файлы. Если у вас нет зависимых файлов, выбор кнопки ни на что не влияет.

   Dreamweaver подключится к вашему Web-серверу и перенесет файлы.

Для обратного переноса файлов с вашего сервера на ваш компьютер воспользуйтесь операцией **Get** (Скачать).

Выполните следующие действия:

1. На панели **Files** (Файлы) из раскрывающегося меню вверху слева выберите ваш Web-сайт.
2. Из раскрывающегося меню вверху справа выберите команду **Remote view** (Удаленный просмотр).

   Dreamweaver не отображает автоматически список файлов на вашем сервере, т. к. получение списка требует некоторого времени. Вам нужно специально запросить у программы обновленное представление списка файлов с вашего сервера, как это сделать, написано в следующем пункте.
3. Щелкните мышью кнопку **Refresh** (Обновить), пиктограмма которой выглядит как круговая стрелка.

   Dreamweaver подключится к Web-серверу, получит список файлов вашего сайта и выведет его на экран.
4. Выделите файлы, которые хотите перенести на ваш компьютер.

   Можно выделить сразу несколько файлов, держа нажатой клавишу <Ctrl> и одновременно щелкая кнопкой мыши имя каждого файла.
5. После выделения файлов щелкните кнопкой мыши стрелку **Get** (зеленая стрелка, направленная вниз) или правой кнопкой имена файлов и выберите команду **Get** (Скачать) из раскрывающегося меню.

   Dreamweaver спросит, нужно ли копировать зависимые файлы.
6. Выберите **Yes** (Да), если хотите скопировать связанные файлы. Например, если вы скачиваете из Интернета страницу с элементами `<img>` для отображения графики, щелкните мышью кнопку **Yes**, чтобы Dreamweaver скачал и графические файлы. Если у вас нет зависимых файлов, выбор кнопки ни на что не влияет.

   Dreamweaver подключится к вашему Web-серверу и скопирует файлы в папку сайта на вашем компьютере.

СОВЕТ

После того как научитесь переносить наборы файлов, можно испытать кнопку **Synchronize** (Синхронизировать) редактора Dreamweaver. Она действует так же, как инструмент публикации в Expression Web. При щелчке мышью этой кнопки Dreamweaver просматривает файлы Web-страниц на вашем компьютере, определяет, какие из них вы обновили, и передает именно эти файлы на ваш сервер.

ГЛАВА 5

Текстовые элементы

Помещать текст на Web-страницы легко. Вы просто открываете HTML-файл, вставляете в него контент и добавляете необязательный тег для форматирования этого контента. Но заставить текст отображаться так, как вы хотите, гораздо труднее.

Для того чтобы ваши Web-страницы выглядели привлекательно, нужно проделать некоторую организационную работу. В этой главе вы получите в свое распоряжение первый необходимый инструмент: HTML-элементы, позволяющие разбить неоформленный текст на четко выделенные заголовки, абзацы, списки и т. д. В примерах с резюме, в *главе 2*, вы уже заставили несколько таких элементов работать, но теперь вы попадете в HTML-мастерскую, чтобы увидеть все, что она может предложить. Завершив знакомство, можно перейти к следующей главе, которая научит вас форматировать строго структурированный контент с помощью *таблиц стилей* — мощных правил форматирования, позволяющих изменять внешний вид отдельных элементов, целых Web-страниц и даже всего вашего Web-сайта.

Представление о тексте и Web-пространстве

Рано или поздно все разработчики Web-сайтов понимают, что проектирование для Web сильно отличается от разработки чего-либо, предназначенного для печати. Прежде чем дать волю сидящему в вас талантливому дизайнеру графических Web-страниц, нужно устранить несколько концептуальных преград.

Определим разницу между HTML-страницей и страницей, созданной в текстовом процессоре. Программы обработки текста (текстовые процессоры) перед тем, как напечатать документ, показывают, как он будет выглядеть. Вы знаете, какой величины будут заголовки, каким шрифтом они будут отображаться, где ваш текст перейдет с одной строки на другую и т. д. Если вы видите что-то, что вам не нравится, вы изменяете это с помощью меню и команд форматирования. Другими словами, ваш текстовый процессор предоставляет вам полный контроль над каждой деталью вашей страницы.

В Web-пространстве гораздо меньше ограничений. Когда вы создаете HTML-документ, вы знаете некоторые, но не все, подробности о том, как будет выглядеть Web-страница на каком-то другом компьютере. Браузер одного пользователя может отображать шрифт большого размера вместо стандартного и растягивать текст для того, чтобы заполнить весь экран 28-дюймового широкоэкранного монитора, в то время как браузер другого пользователя может его сжать и засунуть в угол рабочего стола. А посетители, просматривающие ваш сайт на экранах своих телефонов, планшетных компьютеров или тостеров с выходом в Web, получат еще одно принципиально отличающееся отображение страницы. Итак, в Web вы не можете полностью управлять мелкими деталями отображения так, как вы это делаете при выводе на печать. Но вы *можете* предоставить всю информацию, необходимую Web-браузеру для корректного отображения ваших страниц. Делается это с помощью структурирования ваших страниц, позволяющего обрабатывать элементы вашей страницы неизменно, независимо от настроек браузера у ваших посетителей.

Логическая структура вместо физического форматирования

Прежде чем начать создавать Web-страницы с помощью HTML-разметки, следует усвоить еще одно принципиальное понятие — разницу между *структурированием* документа (разделением его контента на дискретные компоненты, такие как заголовки, абзацы, списки и т. д.) и *форматированием* документа (приданием этим компонентам привлекательного внешнего вида с помощью курсивного начертания, изменения размера шрифта, выделения цветом и т. д.). Начинающие Web-разработчики, не понимающие этой разницы, очень часто, когда нужно структурировать, применяют форматирование, что приводит к созданию запутанных HTML-страниц, которые трудно поддерживать.

Когда Web-пространство только появилось, язык HTML использовал разные элементы, чтобы подчеркнуть различие между структурой документа и его форматом.

- *Логические элементы* (также называемые *семантическими элементами*) определяют отдельные компоненты, формирующие вашу Web-страницу. Они задают заголовок на странице, абзац, список и т. д. Другими словами, они описывают *структуру* вашей страницы. При этом логические элементы не определяют формат компонентов, т. е. как будут выглядеть компоненты при выводе на экран.
- *Презентационные элементы* (также называемые *типографскими элементами*) заняты исключительно форматированием. Примерами могут служить курсивное, жирное или с подчеркиванием начертание шрифта и другие параметры шрифта, которым набран текст. Презентационные элементы не определяют структуру вашей страницы, они только сообщают браузерам, как форматировать ее контент.

> ***Примечание***
>
> Вы опробовали оба типа этих элементов в примере с резюме в *главе 2*. С помощью логических элементов в резюме были созданы заголовок, абзацы и маркированный

список. Презентационные элементы применялись для выделения жирным и курсивным шрифтом небольших фрагментов текста.

Логические элементы главенствовали в HTML с момента его возникновения. Язык XHTML 1.0 упрочил их доминирование, исключив многие презентационные элементы (например, `<font>`, изменяющий гарнитуру шрифта, и `<u>`, подчеркивающий текст). Стандарт HTML5 пошел даже дальше — в нем изменено определение двух презентационных элементов: `<b>` и `<i>`, которое упрямые Web-разработчики отказываются принимать. Теперь `<b>` означает "стилистически отличающийся текст", а не "жирный шрифт", а `<i>` — текст, "выделяемый голосом, например иностранные слова или технические термины", а не "курсивный или наклонный шрифт". На Web-разработчиков это не повлияло. Они всегда отображают текст `<b>` как жирный, а текст `<i>` — как курсив, если вы не применяете таблицу стилей, которая отказывается от этого. Но разница в словах подчеркивает принципиальное изменение в подходе: HTML-элементы должны описывать логическое назначение фрагментов текста, а не его типографическое представление.

Сейчас вы, наверное, недоумеваете: почему HTML-гуру так стремятся убрать форматирование из HTML-документов. Далее перечислены несколько веских причин.

- *Встроенное форматирование беспорядочно.* Увидев старые HTML-страницы с разметкой, напоминающей топкое болото, вы ни за что не захотите вернуться к ним. Для простейшего редактирования вам придется продираться сквозь трясину тегов форматирования, управляющих отступами, цветами, шрифтом и выравниванием.
- *Логические документы легко поддаются корректировке.* Удаление форматирования из разметки вашей страницы делает ее более гибкой. Для изменения способа отображения вашей Web-страницы можно применять таблицы стилей (см. *следующий раздел*), которые при желании можно менять в любой момент. Если же вам нужно отображать Web-страницу по-разному на различных устройствах с доступом в Web, это вы (или браузер) тоже можете сделать.
- *Логические элементы несут смысловую нагрузку.* Логические элементы позволяют внешним программам анализировать HTML-код страницы. Например, кто-то может создать программу автоматического поиска, просматривающую Web-страницу и извлекающую из нее только заголовки верхнего уровня для создания краткого содержания страницы. Или программа могла бы просматривать Amazon.com и искать только обзоры книг. Другая могла бы создать список рекламной рассылки, прочитав элементы `<address>`. Любая подобная программа, столкнувшаяся с Web-страницей, заполненной только презентационными элементами, не сможет получить столь же интересные результаты.
- *Логические элементы удобны для людей с ограниченными возможностями.* Программа считывания с экрана и другие автоматические средства работают гораздо лучше с логически структурированными страницами. Например, программа считывания с экрана (диктор) может помогать слабовидящему человеку знакомиться с вашей страницей, зачитывая заголовки или ссылки, а не читая вслух все подряд.

Примечание

Представление Web как пространства, в котором элементы описывают содержимое страницы (цены, объемы, адреса и т. д.), а не ее внешний вид, называется *семантическим Web-пространством*. По мнению провидцев, первых создателей Интернета, семантическое Web-пространство могло бы стать провозвестником золотой эры информационного доступа и сверхинтеллектуального поиска.

CSS (каскадные таблицы стилей)

В ходе обсуждения структуры страницы возникает вопрос: если при создании "чистейших" страниц предполагается использовать только элементы, которые просто задают структуру ваших страниц, то что же сделает их внешне привлекательными?

Допустим, вы хотите, чтобы все ваши заголовки отображались ярко-розовым шрифтом. Если в языке HTML есть элемент, задающий цвет шрифта, его пришлось бы включать столько раз, сколько заголовков на вашей странице. Теперь представьте, что позже вы решили, что вам больше подходит благородный темно-фиолетовый цвет заголовков. Для внесения изменений в ваш Web-сайт придется открыть и отредактировать каждую HTML-страницу, на которой есть заголовок. Представьте себе, насколько все станет проще, если вы сможете связать ваши заголовки с командой форматирования на главной ссылочной странице. В случае такой настройки вы сможете изменить цвет заголовков всего сайта, просто откорректировав команду форматирования для заголовков на одной главной ссылочной странице.

В конце концов HTML-специалисты пришли именно к такому решению. Они отделили структуру документа — элементы, создающие заголовки, абзацы, списки и т. д. — от команд форматирования и поместили каждую часть в отдельный файл. Вот как это работает. Сначала вы создаете HTML-документы так же, как вы делали это в *главе 2*. Эти документы включают в себя те же элементы для заголовков, абзацев и списков, которые вы изучили раньше. Это означает, что вам не нужно менять свой подход или отбрасывать базовые элементы, которые уже освоили.

Далее с помощью стандарта CSS (Cascading Style Sheets, Каскадные таблицы стилей) вы создаете отдельный документ. Он называется *таблицей стилей* и определяет, как отдельные элементы в вашем HTML-документе форматируются браузером. Например, таблица стилей может содержать такие команды, как "вывести каждый заголовок ярко-красным" или "установить во всех абзацах отступ 15 пикселов".

Механизм таблиц стилей обладает рядом преимуществ. Прежде всего, вы можете применять одну и ту же таблицу стилей ко всем вашим Web-страницам. Поскольку добиваться нужного форматирования приходится долго и нудно, это главное средство экономии времени. После того как вы улучшили внешний вид вашего Web-сайта, свяжите ваши страницы с этой таблицей стилей, и все страницы приобретут тот же внешний вид (рис. 5.1). Более того, когда будет готово новое оформление, не нужно возиться с вашими HTML-документами — просто подправьте таблицу стилей и всем связанным с ней страницам немедленно будет выполнена "подтяжка лица".

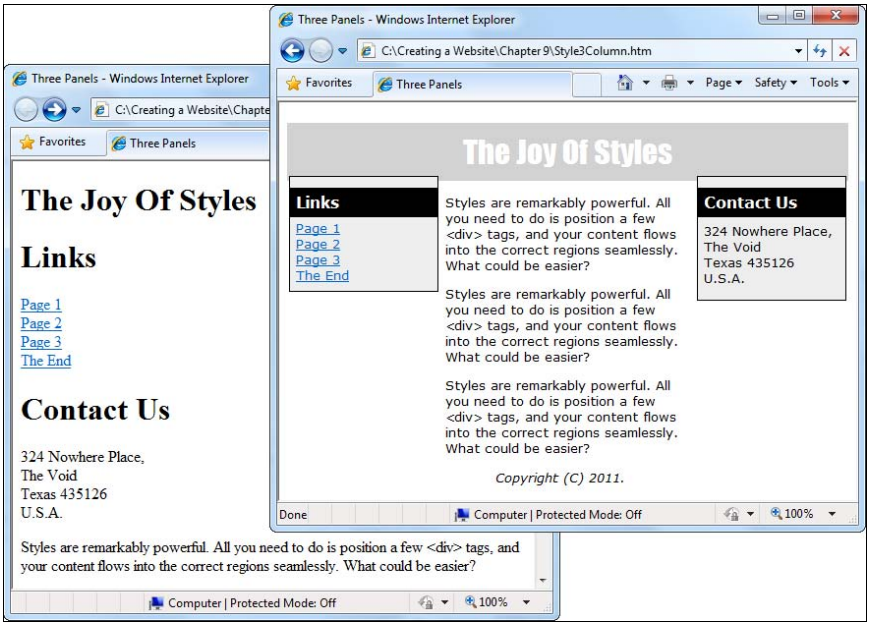


Рис. 5.1. *Слева*: на этой странице отображен обычный текст, но она готова к применению таблицы стилей — автор тщательно разделил страницу на логические разделы.
Справа: когда вы примените таблицу стилей к странице, ее форматирование и макет изменятся до неузнаваемости. Вы увидите этот пример в *главе 9*

Примечание

В языке HTML все еще остается несколько средств форматирования, возвращающих к трудным старым временам Web. Они не так искусны, как таблицы стилей, и гораздо грубее. Теперь, когда вы уже знаете о таблицах стилей, пора отвлечься от этих трудностей и сосредоточиться на освоении главных компонентов HTML-диеты — элементов для структурирования текста.

HTML-элементы для основного текста

Как вы узнали из *главы 2*, о каждом новом элементе, который вам встретился, следует выяснить две вещи. Для корректного применения элемента необходимо ответить на два вопроса:

- Это элемент-контейнер или пустой элемент?
- Это блочный элемент или строчный?

Первый вопрос говорит кое-что о синтаксисе, применяемом вами во время вставки элемента в документ. Элементам-контейнерам (например, элементу `<b>`, выделяющему текст жирным шрифтом) требуются начальный и конечный теги и контент, вложенный между ними. В пустых элементах (например, элементе `<img>`, вставляющем изображение на страницу) используется одиночный тег "все в одном". Если пустым элементам нужна дополнительная информация, например местонахождение файла-изображения, вы предоставляете ее с помощью атрибутов.

Второй вопрос говорит кое-что о том, куда можно поместить элемент. Блочные элементы (как, например, элемент <p>) находятся в главном элементе <body> или внутри других блочных элементов страницы. Когда вы приступаете к созданию общей структуры вашей страницы, то всегда начинаете с блочных элементов. Строчные элементы (например, элемент <img>) должны располагаться в блочных элементах. Строчные элементы сами по себе, находящиеся вне какого-либо контейнера, не имеют смысла.

> ***Совет***
>
> Для того чтобы быстро выяснить, является ли элемент контейнером или это пустой элемент, и понять блочный он или строчный, загляните в справочник в *приложении 1*.

Блочные элементы также влияют на объем свободного пространства в вашем контенте. По сути, каждый блочный элемент определяет фрагмент контента. Когда вы завершаете блочный элемент, ваш браузер автоматически вставляет перенос строки и немного дополнительного свободного пространства перед следующей порцией контента.

Например, рассмотрим следующий фрагмент HTML-кода.

```
<h1>Bread and Water</h1><p>This economical snack is really
all you need to sustain life.</p>
```

В этом фрагменте есть заголовок, отображаемый буквами крупного размера и жирным шрифтом, а сразу за ним следует абзац обычного текста. Можно предположить, что оба компонента (заголовок и обычный текст) расположатся на одной строке. Но элемент <h1> блочный. Когда вы закроете его с помощью тега </h1>, браузер наведет порядок и перед следующим элементом вставит перенос строки и немного свободного пространства. Текст абзаца начнется с новой строки, как показано на рис. 5.2.

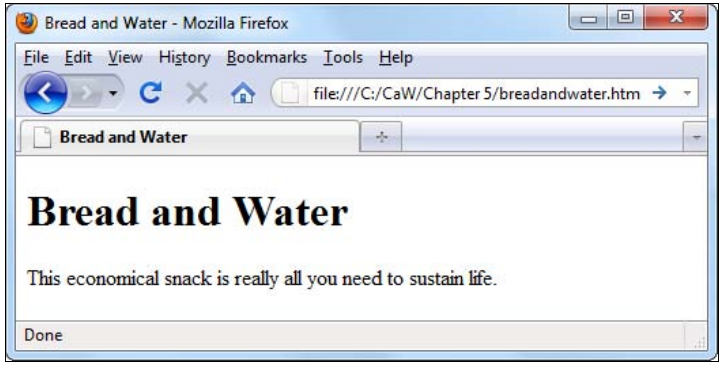


Рис. 5.2. HTML разделяет блочные элементы примерно полуторным интервалом (свободное пространство между "Bread and Water" (Хлеб и вода) и предложением, расположенным ниже)

Браузеры раздвигают блочные элементы, чтобы одна порция текста наверняка не наложилась на другую. Однако во многих случаях вас не обрадует такая автоматическая вставка свободного пространства. Например, на плотных, загруженных ин-

формацией страницах это стандартно добавляемое свободное пространство выглядит излишне щедрым. Для уплотнения текста и сжатия свободных промежутков между блочными элементами можно воспользоваться таблицами стилей и изменить параметры, задающие поля блочных элементов (см. табл. 6.2).

Пытаясь меньше говорить о форматировании и уделять больше внимания структуре, версия HTML5 предлагает новую терминологию. Так строчные элементы предлагается называть *фразовыми элементами* (phrasing elements), а блочные — *потоковыми элементами* (flow elements). Цель — особо подчеркнуть разницу в способах применения и размещения элементов этих типов и одновременно отвлечь внимание от их влияния на форматирование. (В конце концов, детали форматирования могут быть изменены с помощью таблиц стилей.) Но термины "блочный элемент" и "строчный элемент" так широко распространены, что потребуется некоторое время для замены их новым профессиональным жаргоном, если жаргон HTML5 вообще приживется.

Теперь, когда вы познакомились с основными типами элементов, самое время просмотреть весь комплект элементов.

Абзацы

Вы уже знаете базовый элемент абзаца `<p>`. Это блочный элемент, задающий абзац текста.

```
<p>It was the best of times, it was the worst of times...</p>
```

Путешествуя по Интернету, вы, конечно, заметили, что у HTML-абзацев, в отличие от документов, предназначенных для печати, нет абзацного отступа. Это стиль Web, который вы можете изменить с помощью таблиц стилей (см. табл. 6.2). На рис. 5.3 показан образец применения элементов абзаца.

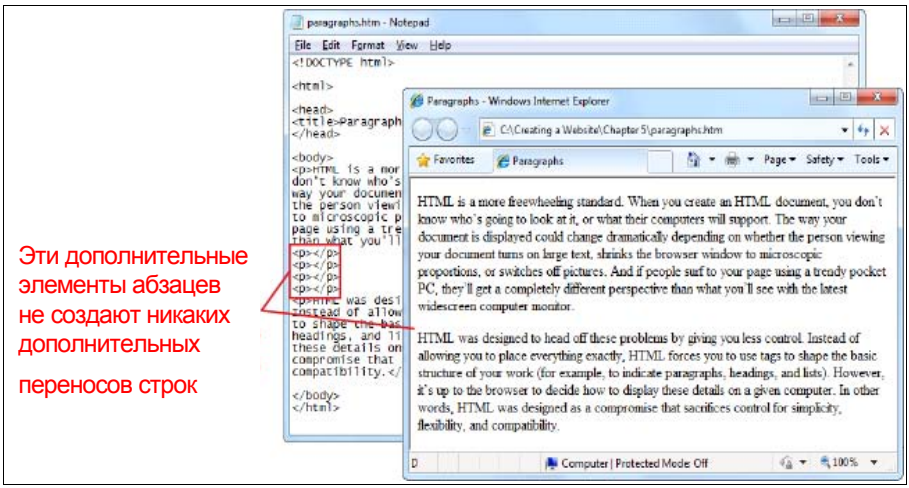


Рис. 5.3. Когда вы размещаете несколько абзацев подряд, браузер отделяет каждый из них пустым пространством, равным примерно полуторному интервалу. Пустые абзацы браузер вообще игнорирует и не добавляет для них никакого пустого пространства

Приучите себя воспринимать текст на ваших Web-страницах как набор абзацев. Иначе говоря, прежде чем набрать любой текст, вставляйте теги и , чтобы сохранить его как контейнер. Большую часть времени именно абзацы будут основным уровнем структуры, добавляемой на вашу страницу.

Web-браузеры не обращают внимания на *принудительные возвраты каретки* (переносы строк, создаваемые вами при нажатии клавиши <Enter>). Это означает, что во время набора текста вы можете отделить его несколькими пустыми строками, а браузер все равно разобьет его в соответствии с размерами своего окна. Строго говоря, браузеры трактуют переносы строки (как тот, что вы видите в конце текущей строки) как единичный пробел. Если браузер находит несколько пробелов, идущих друг за другом, он игнорирует все дополнительные пробелы. Если действительно хотите вставить разрыв между двумя строками вашего текста, прочтите следующий раздел.

НА ПРОФЕССИОНАЛЬНОМ УРОВНЕ

Создание широких пробелов

Игнорирование пробелов браузерами может вывести из себя. Что если вам действительно нужно вставить несколько пробелов подряд? Хитрость заключается в применении *неразрывного пробела* — ` ` — специальной символьной или Escape-последовательности HTML (см. *разд. "Специальные символы" данной главы*), которая вынудит браузер вставить пробел.

Когда браузер встречает такую последовательность, он трактует ее как пробел, который нельзя игнорировать. Итак, если вы создадите следующий абзац:

```
<p>Hello   Bye</p>
```

На экране появится такой текст:

```
Hello   Bye
```

Многие Web-редакторы при нажатии клавиши пробела в режиме **Design** (Конструктор) вставляют неразрывные пробелы, которые не исчезают. Старайтесь без необходимости не использовать неразрывные пробелы. (Если вы действительно хотите создать абзацный отступ, есть лучшее решение с применением таблиц стилей, о нем вы узнаете в *главе 6*.) И никогда не применяйте пробелы для выравнивания колонок текста — это всегда заканчивается плохо, т. к. браузер борется с вашими попытками. Вместо этого используйте инструменты верстки, описанные в *главе 9*.

Переносы строк

Порой вам нужно начать новую строку текста, но вы не хотите применять элемент абзаца, т. к. браузер добавляет пустое пространство между абзацами. Такая ситуация возникает, например, когда вы хотите вставить адрес компании на ваш сайт и хотите, чтобы он выводился в стандартном трехстрочном формате с одинарным интервалом. В этих случаях оказывается удобным пустой элемент переноса строки <br>.

Элементы переноса строки крайне просты: они сообщают браузеру, что нужно перейти к началу следующей строки (рис. 5.4). Это строчные элементы, поэтому вы должны применять их внутри блочного элемента, например абзаца.

```
<p>This paragraph appears<br />
on two lines</p>
```

Не злоупотребляйте переносами строк. Помните о том, что, если вы измените размер окна браузера, он переформатирует ваш текст, стараясь заполнить все доступное пространство. Если вы попытаетесь улучшить ваши абзацы с помощью переносов строк, вы, в конце концов, получите страницы, выглядящие эксцентрично в окнах разного размера. Полезное практическое правило — избегайте переносов строк в обычных абзацах. Используйте их для разрыва строк в адресах, содержаниях,

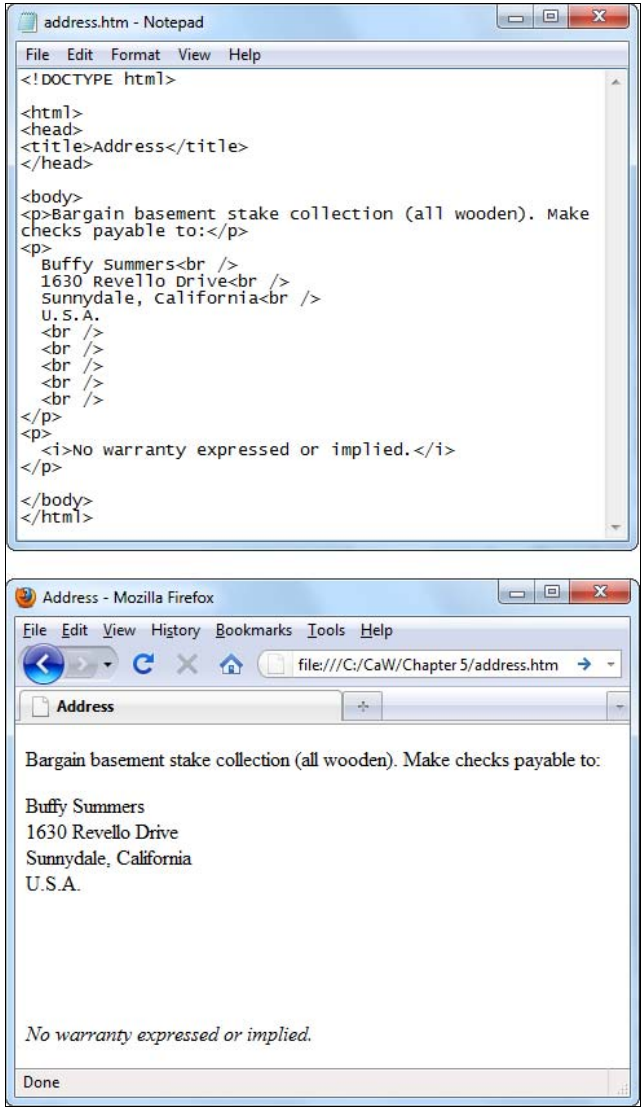


Рис. 5.4. Элемент переноса строки
 очень удобен для разделения строк в адресах. Если вы хотите пропустить несколько строк, можно применить последовательность стоящих рядом элементов
 (но лучше использовать пустые абзацы, как описано в следующей врезке)

стихах и других видах текста, требующих тонкой настройки объема окружающего их свободного пространства. В маркированных и нумерованных списках их также не следует применять. Далее в *разд. "HTML-элементы для создания списков"* вы узнаете об элементах, разработанных специально для списков.

В некоторых случаях нужно *помешать* переносу строки, например, если вы хотите длинноватое название компании или товара сохранить на одной строке. Для этого следует вставить код неразрывного пробела (` `), а не набивать полосу пробелов. Когда браузер встретит код ` `, он все равно отобразит пробел, но оставит в одной строке слова, находящиеся по обе стороны от неразрывного пробела (рис. 5.5).

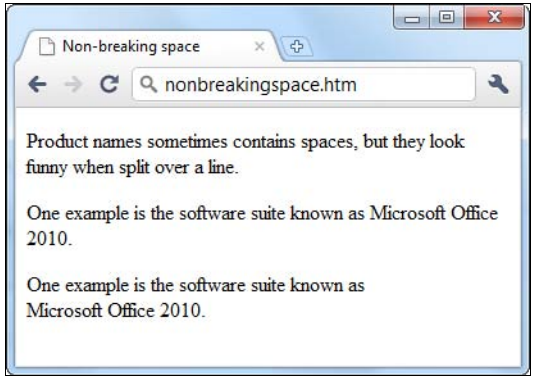


Рис. 5.5. В абзацах 2 и 3 показано, как код ` ` влияет на переносы строк. На самом деле абзац 3 размечен как `Microsoft Office 2010`. В результате браузер не хочет разбивать это название

КАК ЭТО ДЕЛАЕТСЯ?

Таинство пустых абзацев

В приложениях для создания Web-страниц, таких как Dreamweaver и Expression Web, если в режиме **Design** (Конструктор) нажать клавишу <Enter>, программа создаст новый абзац. Это кажется нелогичным, т. к. браузеры обычно игнорируют переносы строк (см. рис. 5.5).

Хитрость состоит в том, что при нажатии клавиши <Enter> обе программы вставляют абзац, содержащий неразрывный пробел. Вот как выглядит это создание:

```
<p> </p>
```

Этот абзац пустой, но браузер не будет его игнорировать, поскольку в него вставлен код ` `. Следовательно, браузер наделит его таким же окружающим свободным пространством, как и однострочный абзац, и сдвинет контент вниз. Между прочим, Dreamweaver и Expression Web разрешают даже в режиме **Design** (Конструктор) применять дополнительные элементы `<br>` вместо пустых абзацев. Для этого вместо клавиши <Enter> нажмите сочетание клавиш <Shift>+<Enter>.

Заголовки

Заголовки — это названия разделов, как слово "Заголовки" над этим абзацем. Браузеры отображают их жирным шрифтом разного размера в зависимости от *уровня*

заголовка. HTML поддерживает шесть уровней, начиная с `<h1>` (самый высокий) и заканчивая `<h6>` (самый низкий). Заголовки уровня `<h5>` и `<h6>` меньше размера обычного текста. На рис. 5.6 показаны заголовки всех уровней, которые вы можете применять.

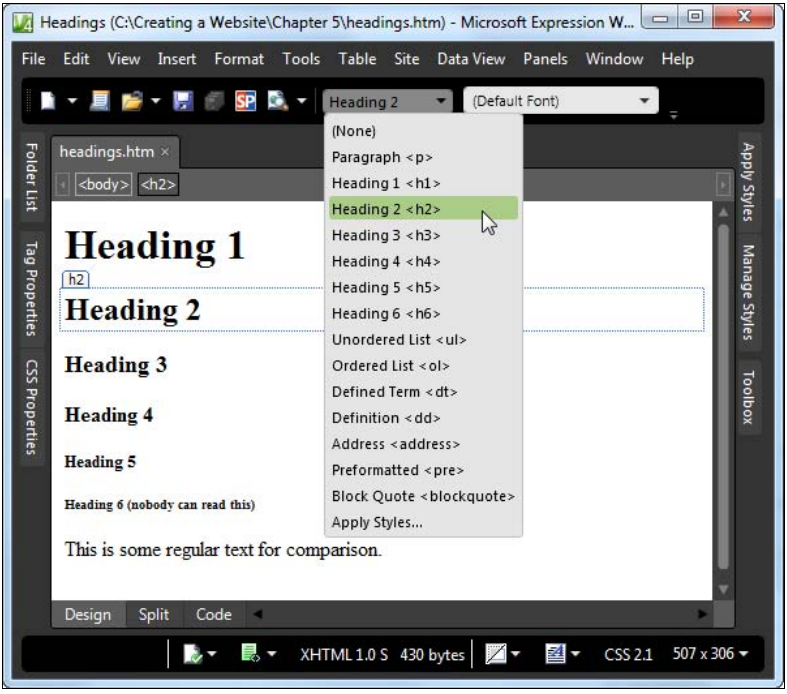


Рис. 5.6. Многие редакторы Web-страниц позволяют применять заголовки одним щелчком кнопки мыши. В Dreamweaver можно использовать удобные кнопки на вкладке **Text** (Текст) панели инструментов. В Expression Web можно найти раскрывающийся список, который позволяет превратить выделенный в данный момент текст в абзац или заголовок одного из шести уровней

Заголовки не просто полезны для форматирования — они помогают определить иерархию вашего документа. Крупные заголовки обозначают важные темы, а более мелкие задают менее важные разделы, относящиеся к более крупному заголовку. Для того чтобы ваш документ был логичен, начинайте с самых крупных заголовков (уровня 1) и затем постепенно переходите к следующим уровням. Не следует сразу начинать с заголовка уровня 3, только потому, что вам нравится, как он выглядит.

Примечание

Вам, наверное, приходило в голову, что, если все будут использовать заголовки одних и тех же уровней в одном и том же порядке, Web станет пресным как бублик в сетевом супермаркете. Без паники — это не так плохо, как кажется. Когда вы добавите в ваш сайт таблицы стилей, то увидите, что можете легко изменить внешний вид любого и каждого из использованных вами заголовков. Пока же старайтесь применять заголовки разных уровней в нужном порядке.

Горизонтальные линии

Абзацы и переносы строк — не единственные элементы, позволяющие отделять части текста. Другой остроумный прием — использование пустого элемента <hr>, который переводится как "горизонтальная линейка". Элемент горизонтальной линейки вставляет линию, которая тянется от одного края контейнера до другого, разделяя все сверху и снизу от нее.

> ***Примечание***
>
> Обычно вы помещаете горизонтальную линию между абзацами, и она тянется от одного края страницы до другого. Но вы также можете поместить ее в меньший контейнер, например отдельную ячейку таблицы, и в этом случае она не будет такой длинной.

Горизонтальные линейки — блочные элементы, поэтому их можно помещать между абзацами (рис. 5.7).

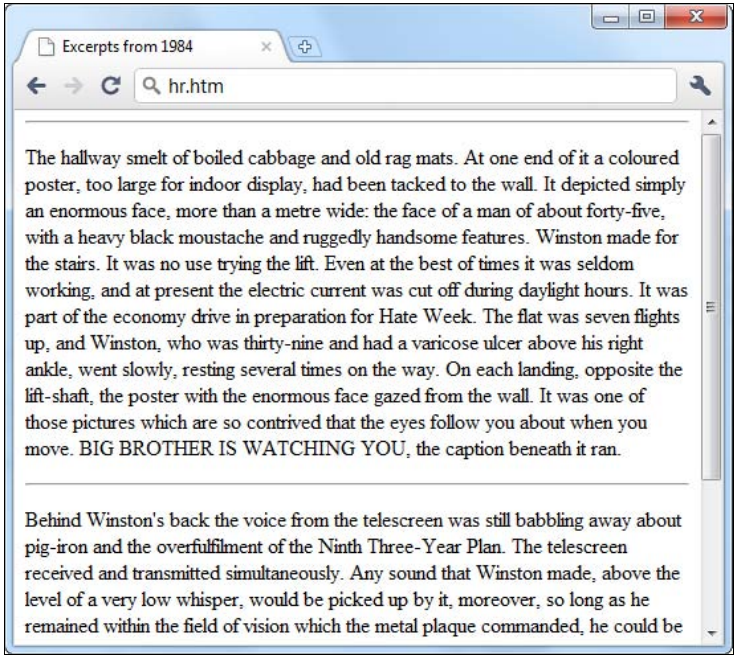


Рис. 5.7. В этом примере элемент <hr> находится между двумя абзацами, вставляя горизонтальную линию, которую вы видите

Понятно, что <hr> — это презентационный элемент (см. *разд. "Логическая структура вместо физического форматирования" данной главы*), потому что он всего лишь рисует линию на вашей странице. Авторы спецификации HTML5 попытались теснее связать его с логической структурой, описав не как "горизонтальную линейку", а как "тематический разрыв", другими словами, как основной способ разделения разных блоков контента, например, эпизодов в романе или разделов в научной статье. Но этот элемент все равно выглядит как линия.

Текст в заданном формате

Текст в заданном формате или *предварительно отформатированный текст* — это уникальное понятие в HTML, нарушающее уже известные вам правила. Как вы уже знаете, браузеры игнорируют множественные пробелы и размещают ваш текст так, чтобы он заполнил окно браузера по ширине. Хотя до известной степени это можно изменить с помощью переносов строк и неразрывных пробелов, с некоторыми видами контента все же справиться трудно.

Например, представьте себе, что вы хотите вывести на экран небольшое стихотворение. На выравнивание текста с помощью неразрывных пробелов вы потратите уйму времени и сделаете вашу разметку трудной для чтения. Элемент <pre> предлагает лучшее решение. Он заставляет браузер воспроизводить текст именно так, как вы его ввели, включая все пробелы и переносы строк, и вывести его в таком виде на экран. Кроме того, браузер отображает весь этот текст моноширинным шрифтом (как правило, Courier), в дальнейшем отделяя этот контент от остального содержимого вашей страницы. На рис. 5.8 показан пример.

> ***Примечание***
>
> В моноширинном шрифте все буквы занимают одно и то же пространство (например, таким шрифтом оформлен HTML-код данной книги). В HTML-документах и книгах, например в этой, используются пропорциональные шрифты, в которых буквы W и M гораздо шире букв l и i. Моноширинные шрифты удобны в предварительно отформатированном тексте, поскольку позволяют точно выровнять строки текста. Но результат, к сожалению, не так привлекателен.

Иногда профессиональные Web-разработчики применяют текст в заданном формате для цитирования программного кода или отображаемого на экране вывода компьютерной программы. В таких случаях можно сочетать элемент <pre> с элементом <code> (для представления фрагмента кода) или странно названным элементом <samp> (для представления компьютерного вывода).

```
<p>This is an infinite loop in Visual Basic:
<pre><code>
    Do
        X = X + 1
    Loop Until X < 0
</code></pre>
</p>
```

В элементах <code> и <samp> используется моноширинный шрифт так же, как в элементе <pre>. Но только элемент <pre> заставляет браузер вставлять все заданное свободное пространство. Вот почему нужно применять элементы <pre> и <code> вместе.

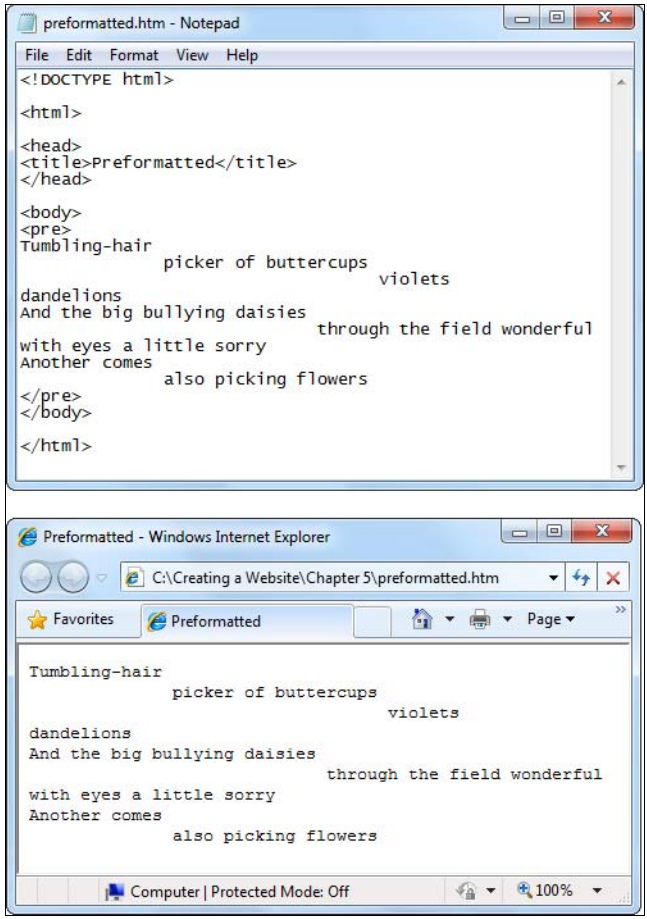


Рис. 5.8. В том, как отображается стихотворение Е. Е. Каммингса, нет никакой мистики. Поскольку оно помещено в блок `<pre>`, вы видите точно такое свободное пространство и переносы строк, какие заданы в вашем HTML-файле. Элемент `<pre>` также хорошо подходит для блоков программного кода

Цитаты

Редко встретишь Web-страницу, на которой литературные цитаты льются рекой, но разработчики HTML создали блочный элемент `<blockquote>` специально для длинных цитат. Когда вы применяете этот элемент, браузер задает отступы для текста слева и справа.

Далее приведен пример.

```
<p>Some words of wisdom from "A Tale of Two Cities":</p>
<blockquote>
<p>It was the best of times, it was the worst of times, it was the age of
wisdom, it was the age of foolishness, it was the epoch of belief, it was the
epoch of incredulity, it was the season of Light, it was the season of
Darkness, it was the spring of hope, it was the winter of despair, we had
everything before us, we had nothing before us, we were all going direct
```

```
to Heaven, we were all going direct the other way—in short, the period was
sofar like the present period, that some of its noisiest authorities insisted
on its being received, for good or for evil, in the superlative degree
of comparison only.</p>
</blockquote>
<p>It's amazing what you can fit into one sentence...</p>
```

На рис. 5.9 показано, как этот текст выглядит в браузере.

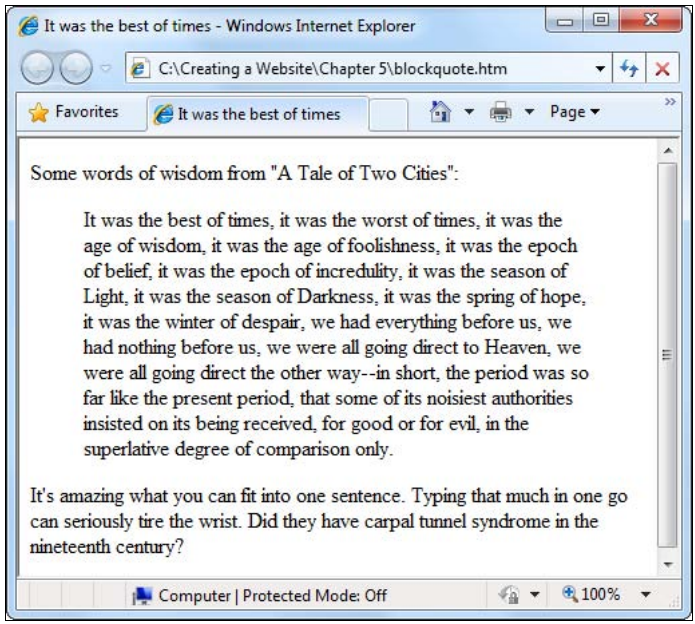


Рис. 5.9. Элемент `<blockquote>` задает отступы для среднего абзаца

Порой разработчики используют элемент `<blockquote>` исключительно из-за его возможностей форматирования — им нравится, как он выводит на экран текст. Конечно, это дискредитирует идею элемента и лучше применять таблицы стилей для получения аналогичного результата.

Элемент `<blockquote>` — блочный, т. е. он всегда вставляется наравне с другими блочными элементами, такими как абзац или заголовки. У элемента `<blockquote>` есть одно дополнительное ограничение: он может содержать только другие блочные элементы, что означает необходимость помещать ваш текст в абзацы, а не просто вводить его между начальным и конечным тегами `<blockquote>`.

Если вместо длинной цитаты из одного или нескольких абзацев вы хотите вставить простую однострочную цитату, HTML предоставит все необходимое. В нем определен строчный элемент для коротких цитат, которые можно вложить в блочный элемент. Это элемент `<q>` (сокращение от "quotation" — цитата):

```
<p>As Charles Dickens once wrote, <q>It was the best of times, it was the
worst of times</q>.</p>
```

Одни браузеры, например Firefox, заключают в кавычки текст в элементе <q>, другие, например Internet Explorer, ничего не делают. Если вы хотите, чтобы ваша цитата выделялась из окружающего текста в любом браузере, можно добавить другое форматирование для нее, например курсивное начертание. Делается это с помощью правила в таблице стилей (см. *главу 6*).

Если вы хотите включить в ваши цитаты дополнительную информацию, можно вставить URL, указывающий на источник цитаты (предположим, что он есть в Web) с помощью атрибута cite.

```
<p>As Charles Dickens once wrote,
<q cite="http://www.literature.org/ authors/dickens-charles/two-cities">It
was the best of times, it was the worst of times</q>.</p>
```

Глядя на этот пример, вы, возможно, рассчитываете, что браузер предоставит вам некое средство, перемещающее вас на указанный Web-сайт (например, при двойном щелчке кнопкой мыши абзаца). Нет, такого средства нет. Если вы хотите связать ваш текст со ссылкой, нужно изучить элемент-якорь в *главе 8*.

На самом деле информация из атрибута cite вообще не появится на вашей странице. Она доступна только программам, анализирующим вашу Web-страницу, например, автоматизированным программам, сканирующим страницы и формирующим список ссылок, или поисковым машинам, использующим эту информацию для улучшения результатов поиска. Большую часть времени эта ссылка приносит мало пользы за исключением того, что она хранит дополнительную порцию информации, которая вам как разработчику Web-сайта может понадобиться позже для перепроверки своих источников.

Если вы действительно хотите сослаться на источник цитаты, можно использовать элемент <cite>, он указывает на заголовок источника, например, так:

```
<p>The quote <q>It was the best of times, it was the worst of times</q> is from
<cite>A Tale of Two Cities</cite>.</p>
```

В этом случае браузер отображает текст в элементе <cite> курсивным шрифтом, как если бы вы применили элемент <i>.

Разделы и оболочки

Последний блочный элемент, с которым вы познакомитесь — — наименее интересный, по крайней мере, на первый взгляд. Сам по себе он ничего не делает.

Элемент применяется для группировки одного или нескольких блочных элементов. Это означает, что вы можете сгруппировать несколько абзацев или абзац и заголовок и т. д. Далее приведен пример.

```
<div>
    <h1>...</h1>
    <p>...</p>
    <p>...</p>
</div>
<p>...</p>
```

Принимая во внимание тот факт, что элемент ничего не делает, вы, возможно, удивитесь тому, что он вообще существует. Непритязательный тег станет гораздо интереснее, если вы используете его вместе с таблицами стилей, поскольку сможете задать команды форматирования непосредственно для элемента . Например, если элемент содержит три абзаца, можно отформатировать все три одновременно, просто отформатировав элемент .

Примечание

Читая эту книгу, вы не раз убедитесь в том, что элемент — незаменимый универсальный контейнер. Его применяют для формирования макета страницы, создания колонок текста, навигационных меню, плавающих изображений и т. д.

У элемента есть важный родственник — элемент . Как и элемент , сам по себе ничего не делает, но если вы поместите его *внутрь* блочного элемента и зададите его атрибуты в таблице стилей, его можно использовать для форматирования части абзаца, что очень удобно. Далее приведен пример.

```
<p>In this paragraph, some of the text is wrapped in a span element. That
<span>gives you the ability</span> to format it in some fancy way later on. </p>
```

Вы научитесь правильно применять элементы и в последующих главах.

АНОНС HTML5

Усиленная структура

Элемент невероятно гибок, но он мало что говорит о реальной структуре вашей страницы. Например, один элемент может представлять заголовок в верхней части вашей страницы. Другой может включать в себя набор навигационных ссылок. По этой причине мало поможет пользователю (или программе) при просмотре вашей разметки и выяснении ее смысловой нагрузки. И хотя это не фатальный дефект элемента , но все же досадный недостаток.

Как вы узнали из *разд. "Логическая структура вместо физического форматирования" данной главы*, идеальная модель будущего Интернета невозможна без встраивания в ваши страницы структурных компонентов, позволяющих браузерам, поисковым машинам и автоматизированным средствам понять структуру страниц и находить важные порции информации. Для облегчения задачи HTML5 предлагает несколько новых элементов, способных заменить традиционные элементы . К ним относятся <header>, <footer>, <article>, <section>, <aside> и <nav>. У этих элементов, как и у элемента , нет встроенного форматирования. Это просто контейнеры, куда вы можете поместить содержимое (и затем точно разместить его и отформатировать с помощью таблиц стилей, о которых узнаете в *главе 6*). Но в отличие от элемента , эти элементы несут бóльшую смысловую нагрузку. По этой причине борцы за чистоту HTML5 предпочитают помещать группу навигационных ссылок в элемент <nav>, а не в стандартный элемент , несмотря на то, что конечный результат в браузере будет одинаковым.

Конечно, поскольку элементы HTML5 новые, не все браузеры способны интерпретировать их. Но это не большая проблема — в конце концов, элементы, заменяющие элемент , не содержат никакого форматирования, поэтому ничего страшного, если браузер решит игнорировать их. Но вот в чем трудность: Internet Explorer не разрешает применять стили к нераспознанным элементам. Для устранения этой пробле-

мы примените "заплатку" на JavaScript. В *приложении 1* описана эта "заплатка" и приведены подробные описания каждого из семантических элементов стандарта HTML5. Пользователи, не рвущиеся на передовые рубежи, могут подождать некоторое время и оставаться приверженцами проверенного временем элемента `<div>`.

HTML-элементы для создания списков

Освоив базовые текстовые элементы, можно приниматься за HTML-элементы для создания списков. Язык HTML позволяет создавать списки трех типов.

- *Нумерованные списки* присваивают каждому пункту списка последовательный номер (например, 1, 2, 3). Они удобны, когда важна последовательность, например, при перечислении шагов, помогающих вашим родственникам проехать к вашему дому.
- *Ненумерованные или маркированные списки* — списки, в которых каждый пункт начинается с маркера. Вы можете до некоторой степени управлять отображением маркера. Маркированный список вы читаете прямо сейчас.
- *Списки определений* удобны для вывода терминов, за которыми следуют определения или описания. Например, словарь — это один громадный список определений. В списке определений на Web-странице ваш браузер выравнивает по левому краю термины и располагает определения под ними с отступом.

В следующих разделах вы узнаете, как создавать списки всех трех типов.

Нумерованные списки

В нумерованном списке HTML нумерует каждый пункт последовательно, начиная с некоторого значения (обычно 1). Удачная находка в нумерованных HTML-списках состоит в том, вам не нужно поддерживать нумерацию. Браузер автоматически вставляет соответствующий номер перед каждым пунктом списка (похоже на средство автонумерации в Microsoft Word). Это удобно по двум причинам. Во-первых, это позволяет вставлять и удалять пункты списка, не нарушая нумерации. Во-вторых, HTML тщательно выравнивает номера и пункты списка, что нелегко сделать вручную.

Для создания нумерованного списка используйте блочный элемент `<ol>` (сокращение от "ordered list" (нумерованный список)). Далее в элемент `<ol>` для каждого пункта списка помещается элемент `<li>` (сокращение от "list item" (пункт списка)).

Далее в качестве примера приведен нумерованный список из трех пунктов.

```
<p>To wake up in the morning:</p>
<ol>
    <li>Rub eyes.</li>
    <li>Assume erect position.</li>
    <li>Turn on light.</li>
</ol>
```

В браузере вы увидите следующее:

```
To wake up in the morning:
     1. Rub eyes.
     2. Assume erect position.
     3. Turn on light.
```

HTML вставляет некоторый объем свободного пространства между абзацем, предшествующим списку, и списком как таковым, как между любыми другими блочными элементами. Далее он присваивает каждому пункту списка номер.

Нумерованный список становится более интересным, если в него вставить атрибуты `start` и `type`. Атрибут `start` позволяет начать нумерацию не с 1, а с другого значения. Далее приведен пример, в котором нумерация начинается с 5.

```
<p>To wake up in the morning:</p>
<ol start="5">
...
</ol>
```

Этот список содержит номера 5, 6 и 7. К сожалению, в настоящее время нет способа задать обратную нумерацию или автоматически продолжить нумерацию предыдущего списка, размещенного на странице.

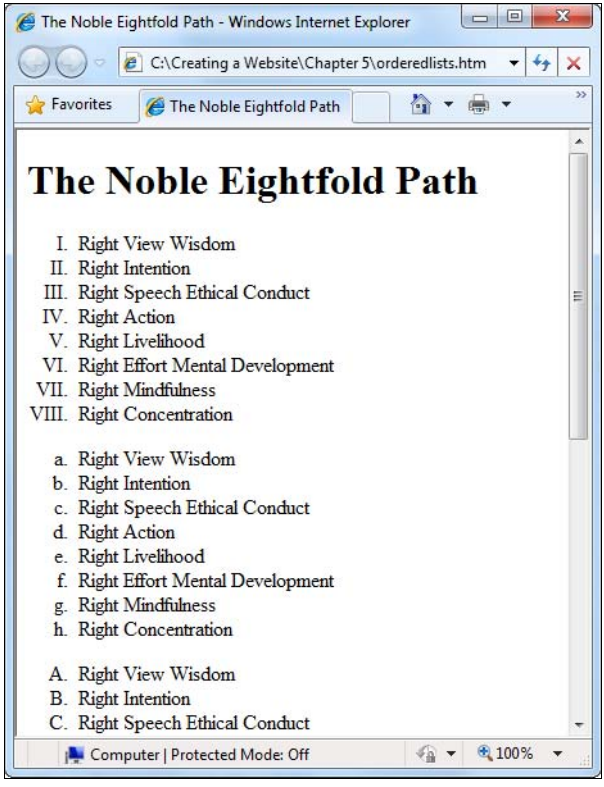


Рис. 5.10. Атрибут `type` в действии. Например, код для начала первого списка выглядит так:

```
<ol type="I">
```

HTML разрешает применять в нумерованных списках не только числа. В атрибуте `type` можно задать стиль нумерации. Можно использовать буквы в алфавитном порядке или римские цифры, как показано в табл. 5.1. На рис. 5.10 приведено несколько примеров.

Таблица 5.1. *Типы нумерованных списков*

Атрибут `type`	Описание	Образец
`1`	Числа	1, 2, 3, 4...
`a`	Строчные буквы	a, b, c, d...
`A`	Заглавные буквы	A, B, C, D...
`i`	Строчные римские цифры	i, ii, iii, iv...
`I`	Заглавные римские цифры	I, II, III, IV...

Маркированные списки

Маркированные списки похожи на нумерованные только без последовательной нумерации пунктов. Элемент-контейнер — `<ul>`, в него вы вставляете пункт списка, заключенный в элемент `<li>`. Браузер формирует отступ для каждого пункта списка и автоматически отображает маркеры.

Самый интересный компонент маркированных списков — атрибут `type`, позволяющий изменять значок маркера. Вы можете использовать кружок (черный кружок, выбираемый автоматически), окружность (не закрашенный кружок) или квадратик (закрашенный квадратик). На рис. 5.11 показаны списки с разными стилями.

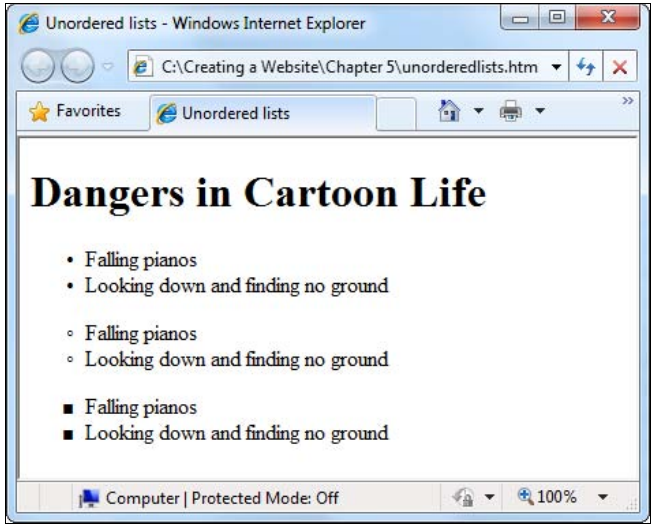


Рис. 5.11. Изменение маркера позволяет создать три варианта одного и того же списка

СОВЕТ

В современных редакторах Web-страниц есть очень удобные инструменты списков, поэтому вы можете быстро создать списки разных типов. В Dreamweaver ищите пиктограммы "ul" и "ol" на вкладке **Text** (Текст) панели инструментов **Insert** (Вставка).

Списки определений

Списки определений очень удобны для создания вашего собственного словаря терминов. У каждого пункта списка есть две части: *термин* (для которого браузер не создает отступа) и *определение* (которое браузер с отступом помещает под термином). В списках определений применяется несколько иная система разметки, чем в нумерованных и маркированных списках. Во-первых, вы помещаете весь список в элемент `<dl>` (словарный список). Затем вы помещаете каждый термин в элемент `<dt>` (словарный термин), а каждое определение — в элемент `<dd>` (словарное определение). Далее приведен пример.

```
<dl>
<dt>eat</dt>
<dd>To perform successively (and successfully) the functions of mastication,
humectation, and deglutition.</dd>
<dt>eavesdrop</dt>
<dd>Secretly to overhear a catalogue of the crimes and vices of another or
yourself.</dd>
<dt>economy</dt>
<dd>Purchasing the barrel of whiskey that you do not need for the price
of the cow that you cannot afford.</dd>
</dl>
```

В браузере вы увидите следующее:

```
eat
     To perform successively (and successfully) the functions of
     mastication, humectation, and deglutition.
eavesdrop
     Secretly to overhear a catalogue of the crimes and vices of
     another or yourself.
economy
     Purchasing the barrel of whiskey that you do not need for the
     price of the cow that you cannot afford.
```

Вложенные списки

Списки и сами по себе очень полезны, но можно добиться большего, поместив целиком один список в другой. Этот прием называется *вложением* списков, и он позволяет создать многоуровневые оглавления и подробные наборы инструкций.

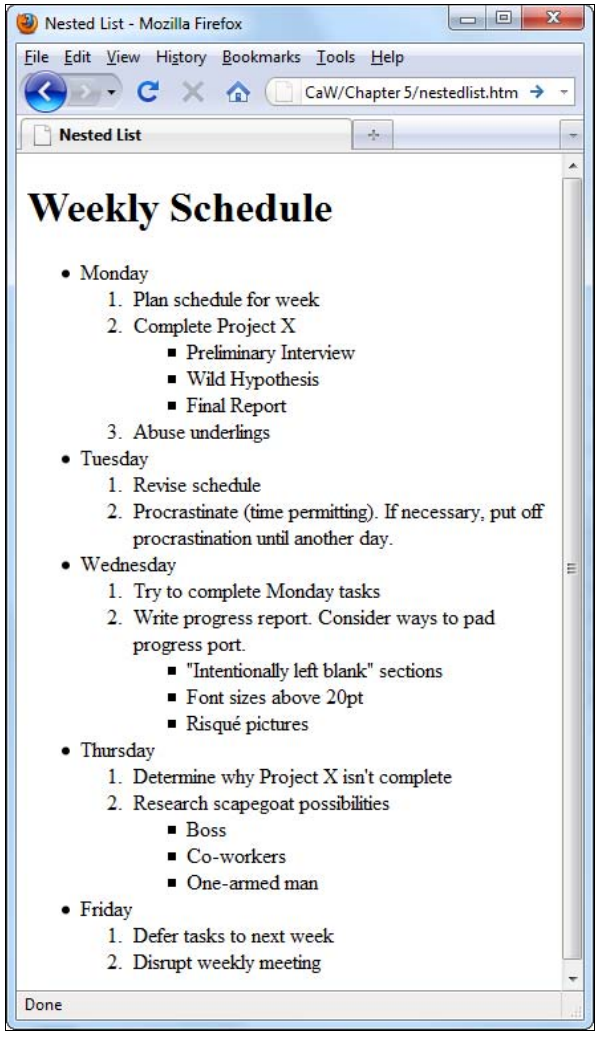


Рис. 5.12. Когда списки вкладываются друг в друга, браузеры создают отступы для списков каждого уровня. Хотя количество уровней не ограничено, в конечном счете вы займете все пространство окна вашего браузера и ваш текст упрется в правый край страницы

Для вложения списка опишите новый список внутри элемента `<li>` уже имеющегося списка. Например, в следующем списке повседневных дел три уровня.

```
<ul>
  <li>Monday
    <ol>
      <li>Plan schedule for week</li>
      <li>Complete Project X
      <ul style="square">
        <li>Preliminary Interview</li>
        <li>Wild Hypothesis</li>
        <li>Final Report</li>
```

```
      </ul>
      </li>
      <li>Abuse underlings</li>
    </ol>
  </li>
  <li>Tuesday
    <ol>
      <li>Revise schedule</li>
      <li>Procrastinate (time permitting). If necessary, put off
      procrastination until another day.</li>
    </ol>
  </li>
  <li>Wednesday
...
</ul>
```

Совет

Если вы применяете вложенные списки, неплохо в вашем HTML-документе использовать отступы, чтобы можно было сразу увидеть элементы списков разных уровней. Иначе трудно будет определить, какому списку принадлежит каждый элемент.

Во вложенном списке сразу становятся полезными разные стили списков, позволяющие различать разные уровни. На рис. 5.12 показан результат приведенного примера.

HTML-элементы для таблиц

HTML-таблица — это сетка из ячеек, формирующих строки и столбцы. В старые добрые времена умелые Web-дизайнеры использовали невидимые таблицы для выравнивания изображений и создания колонок текста. Теперь таблицы стилей делают это с помощью первоклассных средств верстки (как описано в *главе 9*) и HTML-таблицы снова становятся просто таблицами.

Простая таблица

Вы можете быстро создать таблицу, применяя следующие HTML-элементы.

- Элемент `<table>` включает в себя все. Это отправная точка любой таблицы.
- Элемент `<tr>` представляет одну строку таблицы. Каждый табличный элемент (`<table>`) содержит один или несколько элементов `<tr>`.
- Элемент `<td>` задает ячейку таблицы ("td" сокращение от "table data" — табличные данные). Для каждой ячейки, которую вы хотите поместить в строку таблицы, следует добавить один элемент `<td>`. Внутрь этого элемента вы помещаете текст (или числа, или элементы `<img>`, или почти любой код HTML, который вам нравится), отображаемый в этой ячейке. Если в ячейку вставляется текст, он отображается тем же шрифтом, что и основной текст тела документа.

- Элемент <th> — необязательный табличный элемент; его применяют, если хотят задать заголовки столбцов таблицы. Этот элемент можно использовать вместо любого элемента <td>, но, как правило, в нем больше смысла, если он размещен в первой строке таблицы. Браузеры форматируют текст в элементе <th> почти так же, как текст в элементе <td>, за исключением того, что они автоматически отображают его жирным шрифтом и центрируют (до тех пор, пока вы не зададите другие правила форматирования в таблице стилей).

На рис. 5.13 показан пример простейшей таблицы.

Далее приведен HTML-код, формирующий эту таблицу.

```
<table>
  <tr>
    <th>Rank</th>
    <th>Name</th>
    <th>Population</th>
  </tr>
  <tr>
    <td>1</td>
    <td>Rome</td>
    <td>450,000</td>
  </tr>
  <tr>
    <td>2</td>
    <td>Luoyang (Honan), China</td>
    <td>420,000</td>
  </tr>
  <tr>
    <td>3</td>
    <td>Seleucia (on the Tigris), Iraq</td>
    <td>250,000</td>
  </tr>
...
</table>
```

В коде разметки этой таблицы элементы таблицы приведены с отступом, чтобы отчетливее показать структуру таблицы. Создание таких отступов — всегда полезный прием, т. к. помогает обнаружить непарные теги. В этом примере в элементах <td> размещен единственный вид контента — обычный текст. Но в ячейки можно помещать и другие HTML-элементы, включая изображения (элемент <img>).

> ***Примечание***
>
> Если вы хотите добавить границы таблицы или управлять размером столбцов таблицы, необходимо воспользоваться таблицей стилей. В *разд. "Применение границ в таблицах" главы 6* показано, как использовать свойства таблицы стилей для создания границ таблицы, а в *разд. "Задание размера столбца" главы 9* демонстрируется, как их применять для управления шириной столбцов таблицы.

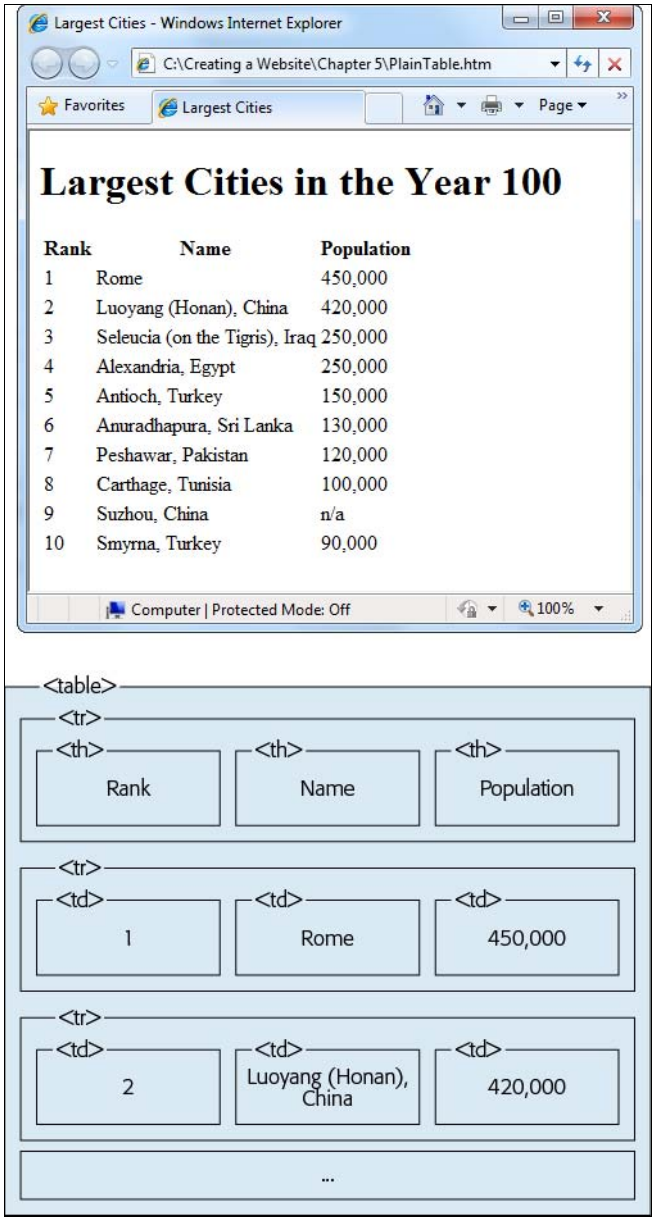


Рис. 5.13. *Вверху*: у этой простой таблицы нет никаких границ (стандартный вариант), но вы все равно можете увидеть отличительный признак таблицы: текст четко выровнен в строках и столбцах. *Внизу*: вид внутренней HTML-структуры, формирующей таблицу, показанную вверху, включает в себя элементы `<table>`, `<tr>`, `<th>` и `<td>` для первых трех строк

Охват ячеек

Таблицы поддерживают *охват* ячеек, средство, позволяющее одной ячейке растянуться на несколько столбцов или строк. Охват — это HTML-версия средства **Merge Cells** (Объединение ячеек) в программах Microsoft Word и Excel.

Охваченные ячейки позволяют видоизменять таблицы самым причудливым образом. Например, можно задать охват столбцов, чтобы растянуть ячейку на два или несколько столбцов, расположенных следом за ней. Просто вставьте атрибут `colspan` в элемент `<td>`, который хотите растянуть, и задайте количество столбцов, которое нужно пропустить.

Далее приведен пример, в котором ячейка растягивается на два столбца.

```
<table>
  <tr>
    <td>Column 1</td>
    <td>Column 2</td>
    <td>Column 3</td>
    <td>Column 4</td>
  </tr>
  <tr>
    <td> </td>
    <td colspan="2">Look out, this cell spans two columns!</td>
    <td> </td>
  </tr>
...
</table>
```

На рис. 5.14 показан этот прием в действии.

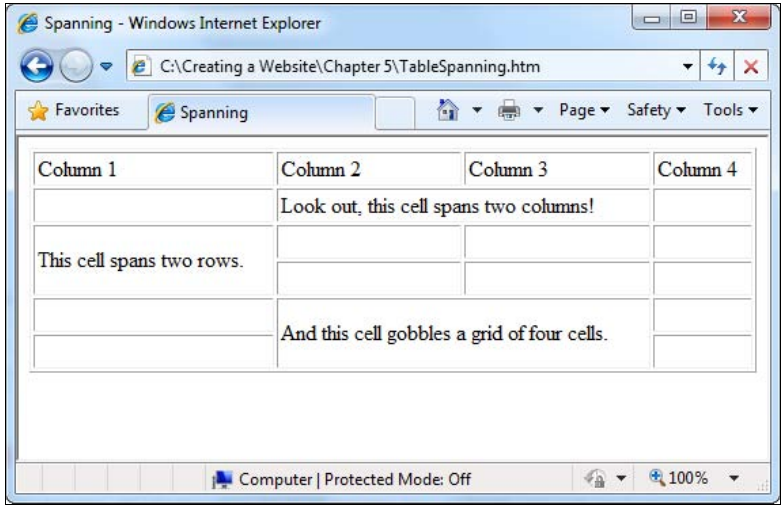


Рис. 5.14. Таблицей с охватом строк и столбцов трудно управлять. В данном случае создатель страницы включил отображение границ таблицы (см. *разд. "Применение границ в таблицах" главы 6*), поэтому вы можете увидеть очертание каждой ячейки

Для того чтобы не исковеркать таблицу, нужно следить за общим количеством столбцов, с которыми следует работать в каждой строке. В предыдущем примере первая строка начинается с определения четырех основных столбцов.

```
  <tr>
    <td>Column 1</td>
    <td>Column 2</td>
    <td>Column 3</td>
    <td>Column 4</td>
  </tr>
```

В следующей строке второй столбец растягивается до третьего столбца, благодаря охвату ячеек (см. разметку далее). В результате третий элемент `<td>` на самом деле представляет четвертый столбец. Это означает, что для заполнения всей ширины таблицы вам нужно только три элемента `<td>`.

```
<tr>
  <!—Этот элемент заполняет столбец 1 -->
  <td> </td>

  <!-- Этот элемент заполняет столбцы 2 и 3 -->
  <td colspan="2">Look out, this cell spans two columns!</td>

  <!-- Этот элемент заполняет столбец 4 -->
  <td> </td>
</tr>
```

Тот же принцип применяется для охвата строк и задания атрибута `rowspan`. В следующем примере первая ячейка в строке переходит на вторую строку.

```
<tr>
  <td rowspan="2">This cell spans two rows.</td>
  <td> </td>
  <td> </td>
  <td> </td>
</tr>
```

В следующей строке ячейка сверху уже занимает первую ячейку, поэтому первый объявляемый вами элемент `<td>` на самом деле относится ко второму столбцу. Следовательно, в этой строке должно быть только три элемента `<td>`.

```
<tr>
  <td> </td>
  <td> </td>
  <td> </td>
</tr>
```

Если вы ошибетесь и вставите слишком много ячеек в строку, вы получите лишний столбец в конце вашей таблицы.

> ***СОВЕТ***
>
> Многие Web-редакторы позволяют создавать охваты с помощью объединения ячеек. В Dreamweaver и Expression Web выделите группу ячеек, щелкните ее правой кнопкой мыши и выберите команду **Merge Cells** (Объединить ячейки).

Внутристрочное форматирование

Как вы уже узнали из этой главы, лучше всего не форматировать усиленно HTML-код. Для наилучшего контроля внешнего вида ваших страниц и более легкого обновления представления вашего Web-сайта в будущем следует сразу переходить к таблицам стилей (как описано в следующей главе). Тем не менее, несколько базовых элементов форматирования по-настоящему полезны. Вы обязательно столкнетесь с ними и возможно захотите использовать их на ваших собственных страницах. Все эти элементы *строчные*, поэтому используются внутри блочного элемента, такого как абзац, заголовок или список.

Форматирование текста: курсив и жирное начертание

Из *главы 2* вы уже знаете элементы для задания курсивного (<i>) и жирного начертания шрифта (<b>). Это основные элементы в мире HTML, позволяющие быстро форматировать фрагменты текста. Далее приведен пример.

```
<p>
<b>Stop!</b> The mattress label says <i>do not remove under penalty
of law</i> and you <i>don't</i> want to mess with mattress companies.
</p>
```

Браузер отобразит этот HTML-код следующим образом.

Stop! The mattress label says *do not remove under penalty of law* and you *don't* want to mess with mattress companies.

Как лекарство от скуки в язык HTML введен второй набор элементов, которые на первый взгляд делают то же самое. Это элементы (для выделения текста) и .

Предыдущий пример можно переписать, применив элементы и .

```
<p>
<strong>Stop!</strong> The mattress label says <em>do not remove under penalty
of law</em> and you <em>don't</em> want to mess with mattress companies.
</p>
```

Обычно элемент задает курсивное начертание шрифта, так же как элемент <i>. Аналогично элемент заставляет браузер выводить текст жирным шрифтом, как и элемент <b>. Разница философская.

Вот, что предлагает последняя версия HTML-стандарта.

- Используйте для текста, имеющего важное значение. Слово "Stop" из приведенного кода может служить хорошим примером.
- Используйте <b> для текста, который должен быть представлен жирным шрифтом, но при этом не важнее остального вашего текста. Он может включать ключевые слова, названия товаров и все остальное, что следует выделить жирным шрифтом при печати. Например, по стилистическим соображениям вы можете

решить выделять жирным шрифтом первые три слова в каждом абзаце вашего документа. Это вряд ли будет привлекательно, но <b> для этого больше подходит, чем .

- Применяйте для выделения текста, у которого при произнесении вслух было бы логическое ударение. Например, слово "don't" в предыдущем примере.
- Применяйте <i> для текста выделяемого голосом, такого как иностранные слова, технические термины и все то, что вы выделили бы курсивом при печати. Хороший образец — метка на подушке сиденья автомобиля из предыдущего примера, запрещающая удалять ее.

> ***Примечание***
>
> Честно говоря, очень немногие Web-дизайнеры не пренебрегают этими рекомендациями хотя бы изредка. Лучший совет — будьте последовательны в своей работе. И помните, что вы можете изменить способ выделения вашего текста спустя много времени после того, как применили свою разметку. Например, вы можете решить, что текст в элементе должен быть не жирным, а другого цвета или выводиться другим шрифтом или шрифтом другого размера. Как вы увидите в *главе 6*, с помощью таблиц стилей можно изменить форматирование любого элемента.

Несколько дополнительных элементов форматирования

Несколько дополнительных элементов способны изменить внешний вид маленьких фрагментов текста (рис. 5.15), хотя вы будете применять их редко, если вообще будете. И все-таки потратьте время на знакомство с ними, чтобы быть во всеоружии в тех редких случаях, когда вам понадобится один из них.

Для начала два элемента, изменяющие размер и местоположение текста. Вы можете использовать элемент <sub> для нижних индексов — текста, который меньше текста текущей строки и расположен у нижнего края строки. Элемент <sup>, сокращение от superscript (верхний индекс), — мелкий текст у верхнего края текущей строки.

Следующий элемент <del> обычно представляет удаленный текст в переработанном документе или текст, который больше не применяется (например, устаревший закон). Браузер выводит этот текст, но зачеркивает его горизонтальной линией (в HTML5 введен дополнительный элемент <mark> для выделения текста, но он пока еще слишком новый, чтобы рассчитывать на его твердую поддержку браузерами.)

И в заключение элемент <small> — прямой способ форматирования "мелкой печати" — например, юридических тонкостей в нижней части контракта, которую вы можете не заметить, на что очень рассчитывает деловой партнер. В прошлом элемент <small> просто означал "текст, выглядящий мелким", и Web-дизайнеры редко им пользовались. Но HTML5 дал ему вполне приемлемое смысловое значение и благодаря этому вернул право на жизнь.

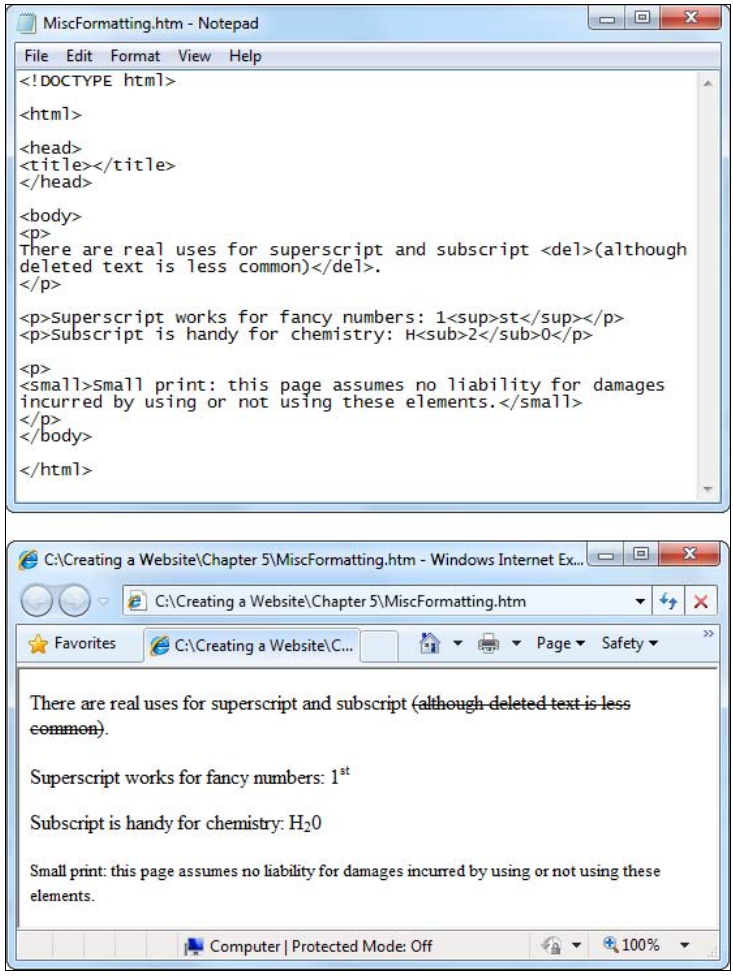


Рис. 5.15. Удаленный текст, верхний и нижний индексы и мелкая печать в действии

Примечание

В языке HTML также есть элемент `<u>` для подчеркнутого текста, но Web-разработчики считают его устаревшим. Если вы действительно хотите подчеркнуть текст, поместите его в элемент `<span>` (который сам по себе не задает никакого форматирования) и затем воспользуйтесь стилевым свойством `text-decoration`, чтобы отформатировать элемент `<span>` (см. табл. 6.3).

Специальные символы

Не все символы есть на вашей клавиатуре. Например, что если вам нужно вставить знак охраны авторского права (©), знак абзаца (¶) или акцентированную e (é). Хочу вас обрадовать: HTML поддерживает их все наряду с другими почти 250 родственниками, включая математические знаки и исландские буквы. Но чтобы вставить их, вам понадобится некоторая ловкость рук. Хитрость состоит в применении сим-

вольных HTML-последовательностей — специальных кодов, которые распознаются как запросы на вывод необычных символов. В табл. 5.2 приведены несколько часто используемых знаков и горстка акцентированных символов (символов с диакритическими знаками).

Таблица 5.2. *Широко распространенные специальные символы*

Символ	Название	HTML-код
©	Знак охраны авторского права	`©`
®	Зарегистрированная торговая марка	`®`
¢	Знак цента	`¢`
£	Фунт стерлинг	`£`
¥	Знак йены	`¥`
€	Знак евро	`€` (у кода `&#8364` бóльшая поддержка)
°	Знак градуса	`°`
±	Плюс или минус	`±`
÷	Знак деления	`÷`
×	Знак умножения	`×`
µ	Микро	`µ`
¼	Дробь, одна четверть	`¼`
½	Дробь, одна вторая	`½`
¾	Дробь, три четверти	`¾`
¶	Знак абзаца	`¶`
§	Знак параграфа	`§`
"	Левая угловая кавычка, галочка влево	`«`
"	Правая угловая кавычка, галочка вправо	`»`
¡	Перевернутый восклицательный знак	`¡`
¿	Перевернутый вопросительный знак	`¿`
æ	Строчный дифтонг ae (лигатура)	`æ`
ç	Строчная c, седиль	`ç`
è	Строчная e, гравис	`è`
é	Строчная e, восходящий акцент	`é`
ê	Строчная e, знак циркумфлекс	`ê`
ë	Строчная e, диереза или умляут	`ë`
ö	Строчная o, диереза или умляут	`ö`
É	Заглавная E, восходящий акцент	`É`

Символьные HTML-последовательности существуют не только для не английских букв и экзотических символов. Они вам понадобятся для работы с символами, имеющими специальное значение в соответствии со стандартом HTML, а именно угловыми скобками (<, >) и амперсандом (&). Эти символы не следует непосредственно вводить на Web-страницу, потому что браузер решит, что вы пытаетесь ввести специальную команду. Вместо этого упомянутые символы нужно заменить их эквивалентными символьными последовательностями, как показано в табл. 5.3.

Таблица 5.3. *Символьные HTML-последовательности*

Символ	Название	HTML-код
<	Левая угловая скобка	`<`
>	Правая угловая скобка	`>`
&	Амперсанд	`&`
"	Знак кавычек	`"`

Строго говоря, эти символьные последовательности нужны вам не всегда. Например, вполне можно вставить в ваш текст обычные кавычки, набрав их на клавиатуре — только не помещайте их в имена атрибутов. У браузеров также хватает интеллекта для корректной обработки знака амперсанда (&), встретившегося в тексте, но лучшим стилем считается применение кода `&`, чтобы браузер наверняка не спутал чистый амперсанд с началом специальной символьной последовательности. И наконец, для угловых скобок, встречающихся в тексте, символьные последовательности совершенно и безусловно необходимы.

Далее приводится испорченный фрагмент текста, который не будет отображаться корректно.

```
I love the greater than (>) and less than (<) symbols. Problem is,
when I type them my browser thinks I'm trying to use a tag.
```

А далее приводится исправленная версия с HTML-последовательностями. Когда браузер обрабатывает и выводит на экран этот текст, он заменяет символьные последовательности нужными вам символами.

```
I love the greater than (&gt;) and less than (&lt;) symbols. Problem is,
when I type them my browser thinks I'm trying to use a tag.
```

Большинство редакторов Web-страниц вставляют нужные символьные последовательности во время набора текста, если вы находитесь в режиме **Design** (Конструктор), а не в режиме **Code** (Исходный код).

СОВЕТ

Более полный перечень специальных символов и их отображение в вашем браузере можно найти на Web-странице **www.webmonkey.com/reference/Special_Characters**.

Иностранные языки

Несмотря на то, что символьные последовательности отлично себя зарекомендовали, частое обращение к ним затрудняет работу. Например, рассмотрим знаменитое французское высказывание "Мы произносим речь, чтобы скрыть свои мысли":

```
La parole nous a été donnée pour déguiser notre pensée.
```

Вот как она будет выглядеть, если все буквы с диакритическими знаками заменить символьными последовательностями:

```
La parole nous a &eacute;t&eacute; donn&eacute;e pour d&eacute;guiser notre
pens&eacute;e.
```

Франкоязычные Web-разработчики вряд ли будут довольны, если им часто придется сталкиваться с подобным набором. К счастью есть решение, называемое *кодировкой Unicode* (Unicode encoding). По существу Unicode — это система, преобразующая символы в байты, которые компьютеру понятны и которые он может правильно отобразить в HTML-коде и в тексте, отображаемом на вашей Web-странице. Применяя кодировку Unicode, вы сможете создавать буквы с диакритическими знаками так же легко, как если бы они были клавишами на вашей клавиатуре.

Как это работает? Прежде всего, вам нужен способ вставки акцентированных символов на вашу Web-страницу. Далее перечислено несколько вариантов.

- *Ввести их с клавиатуры.* У многих не англоговорящих пользователей есть клавиатуры с акцентированными символами.
- *Применить утилиту.* В ОС Windows вы можете запустить маленькую утилиту с именем *charmap* (сокращение от Character Map (Таблица символов)), которая позволяет выбрать из набора специальных символов и скопировать в буфер выделенный вами символ и после этого его можно вставлять в другую программу. Для запуска таблицы символов щелкните мышью кнопку **Start** (Пуск) и введите `charmap` в строку поиска. Когда программа появится на экране, щелкните ее кнопкой мыши (рис. 5.16).
- *Применить ваш редактор Web-страниц.* В состав некоторых редакторов Web-страниц входят собственные встроенные таблицы символов. В Dreamweaver можно использовать команды **Insert | HTML | Special Characters | Other** (Вставить | HTML | Специальные символы | Другие). В Expression Web выберите **Insert | Symbol** (Вставка | Символ). Обычно эта процедура вставляет символьные последовательности, а не символы Unicode. Но конечный результат тот же — ваша HTML-разметка будет включать в себя кучу специальных кодов символов.

При использовании кодировки Unicode следите за корректным сохранением вашей Web-страницы. Трудностей не возникает в профессиональном редакторе Web-страниц, который достаточно умен, чтобы с первого раза сделать все правильно, но текстовые редакторы Unicode может сбить с толку. Например, в редакторе Notepad (Блокнот) ОС Windows необходимо выбрать команды **File | Save As** (Файл | Сохранить как), а затем указать **UTF-8** в списке **Encoding** (Кодировка) (рис. 5.17). В редакторе TextEdit на компьютерах Mac выберите команды **Format | Make Plain Text**

(Формат | Конвертировать в простой текст), затем перейдите к окну и полям **Preferences | Open and Save | Plain Text File Encoding | Saving Files** (Настройки | Открытие и сохранение | Кодировка файла простого текста | Сохранение файлов) и выберите из раскрывающегося списка вариант **Unicode (UTF-8)**. После этого при всех последующих сохранениях файла Notepad и TextEdit будут кодировать его правильно.

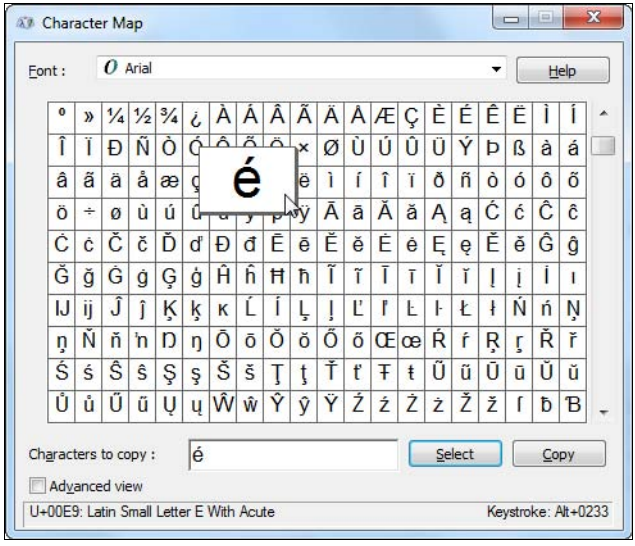


Рис. 5.16. Утилита Charmap предоставляет исчерпывающий перечень специальных символов, которые можно скопировать и вставить в любую программу. Дважды щелкните кнопкой мыши любой нужный вам символ, а затем, когда выделите все, что нужно, щелкните мышью кнопку **Copy** (Копировать). При вставке этих символов в другую программу вы будете иметь дело с реальным символом в кодировке Unicode, а не с зашифрованной символьной последовательностью

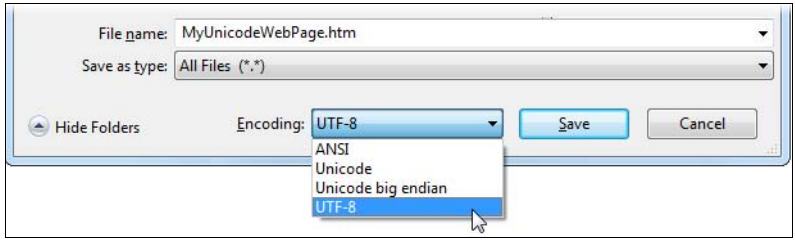


Рис. 5.17. UTF-8, всепобеждающий стандарт Web, — это усеченная версия Unicode, экономящая память при передаче обычных символов. Но в редакторе Notepad при сохранении Web-страницы, содержащей специальные символы, например, буквы с диакритическими знаками, следует явно задать кодировку UTF-8

УГОЛОК ДИЗАЙНЕРА

Подготовка вашего текста для Web

Как вы уже узнали из этой главы, текст, предназначенный для Web-пространства, отличается от печатного текста. Но порой трудно отказаться от старых привычек. Далее приведено несколько неписаных правил, которые помогут вам грамотно использовать текст на ваших Web-страницах.

- *Разбивайте ваш текст на небольшие части.* Web-страницы (и их читатели) неодобрительно относятся к длинным абзацам.
- *Создавайте короткие страницы.* Если страница занимает более двух экранов, разбейте ее на две. Это не только облегчит чтение ваших страниц, но и позволит увеличить число страниц, что поможет выполнить следующую рекомендацию.
- *Делите ваш контент на несколько страниц.* Далее следует связать эти страницы друг с другом (см. *главу 8*). Такой подход позволит читателям выбирать, что они хотят прочесть и в каком порядке.
- *Помещайте самую важную информацию на первый экран.* Суть заключается в том, чтобы дать возможность посетителям прочесть что-либо привлекающее внимание или заинтересовавшее без прокрутки страницы. Этот метод проектирования называется проектированием *верхней половины полосы* или *видимости без прокрутки*. Профессионально сверстанные газеты пользуются той же стратегией; она дает возможность посетителям газетных киосков прочесть самое интересное, не разворачивая сложенную газету, отсюда и термин "верхняя половина полосы".
- *Вычитывайте, вычитывайте и вычитывайте.* Опечатки и безграмотность убивают впечатление профессионализма и крутизны вашего Web-сайта.
- *Не зацикливайтесь на форматировании, пока не разберетесь в таблицах стилей.* Если вы проигнорируете это правило, то получите огромную свалку, которую позже именно вам придется разгребать.

ГЛАВА 6

Таблицы стилей

В предыдущей главе вы узнали маленький неблаговидный секрет языка HTML: у него не слишком много возможностей для форматирования. Если вы хотите, чтобы ваши Web-страницы выглядели стильно, необходимо добавить таблицы стилей.

Таблица стилей — это просто документ, заполненный правилами форматирования. Браузеры читают эти правила и применяют их при выводе Web-страниц на экран. Например, в правиле форматирования может быть записано: "Выводить все заголовки этого сайта жирным шрифтом пурпурным цветом и обводить рамкой".

Помещать правила форматирования в таблицы стилей, а не в код Web-страницы следует по нескольким причинам. Самая очевидная — многократное использование. Например, благодаря таблицам стилей вы можете создать одно правило для форматирования заголовков 3-го уровня, и все заголовки этого уровня на всех Web-страницах вашего сайта будут ему соответствовать. Вторая причина заключается в том, что таблицы стилей помогают писать аккуратные, легко читаемые и управляемые HTML-файлы. Поскольку таблицы стилей управляют форматированием всего вашего сайта, HTML-документ не должен этим заниматься. Его единственная задача — формирование логических разделов на ваших страницах. (Разницу между структурированием и форматированием Web-страницы см. в *разд. "Логическая структура вместо физического форматирования" главы 5*.) И в заключение таблицы стилей предоставляют более широкие возможности форматирования, чем сам по себе язык HTML. Применяя таблицы стилей, вы можете управлять цветом, рамками, полями, выравниванием и (до некоторой степени) шрифтами.

В этой главе вы сначала узнаете о том, как работают таблицы стилей и как присоединить таблицу стилей к вашим Web-страницам, далее вы познакомитесь с обширным набором свойств таблиц стилей, которые сможете применять для получения нужных визуальных эффектов.

Примечание

В этой главе описаны не все свойства CSS. Некоторые из них применяются в первую очередь к изображениям или макетам; вы узнаете о них в последующих главах.

Основы таблиц стилей

Таблицы стилей используют стандарт, названный CSS (Cascading Style Sheets, Каскадные таблицы стилей). CSS — система для определения стилевых правил. Эти правила изменяют внешний вид элементов, размещенных на ваших Web-страницах, выполняют тонкую настройку их деталей, таких как цвет, шрифт, размер, границы и местоположение.

Когда вы на вашей Web-странице применяете CSS, браузер читает и HTML-файл страницы, и правила его таблицы стилей. Затем он применяет прочитанные правила для форматирования страницы. На рис. 6.1 показан этот процесс.

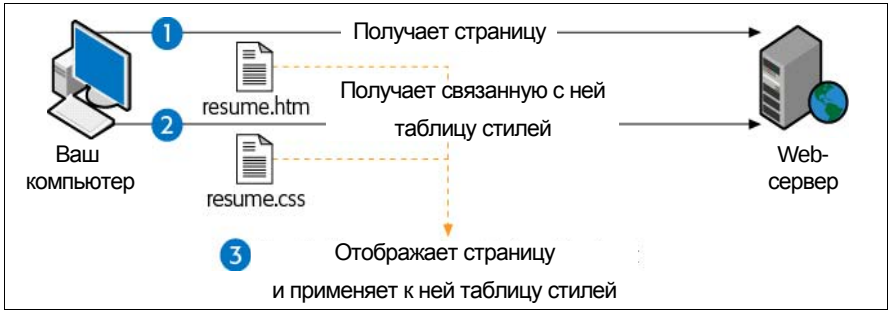


Рис. 6.1. Когда вы переходите на Web-страницу, использующую таблицу стилей, происходит следующее. 1. Ваш браузер запрашивает HTML-страницу с Web-сервера. 2. На HTML-странице браузер находит инструкцию, указывающую на то, что Web-страница использует таблицу стилей. Затем с помощью отдельного запроса браузер захватывает эту таблицу стилей. 3. Браузер "пережевывает" HTML-код Web-страницы и применяет правила из таблицы стилей для настройки отображения страницы

Такой подход предоставляет Web-разработчикам максимум возможностей — разнообразные способы форматирования страниц и сохранение чистоты HTML-кода. В идеале HTML-документ описывает только структуру вашей страницы (что служит заголовком, что абзацем, а что списком и т. д.), а таблица стилей форматирует вашу страницу, придавая ей привлекательный внешний вид.

Три вида стилей

Прежде чем учиться писать CSS-правила, необходимо подумать о том, куда поместить их. Стандарт CSS предлагает три варианта применения таблицы стилей к вашей странице.

- Вы сохраняете *внешнюю таблицу стилей* в отдельном файле. Это самый эффективный способ, т. к. он полностью отделяет правила форматирования от ваших HTML-страниц. Он также позволяет легко применять одни и те же правила ко многим страницам.
- Вы встраиваете *внутреннюю таблицу стилей* в HTML-документ (она помещается в раздел `<head>` на вашей странице). У вас все еще сохраняется разделение информации о стилях и HTML-кода и при желании, с помощью операций копирования и вставки, вы можете перенести встроенную таблицу стилей с одной

страницы на другую (хотя будет трудно сохранять идентичность всех копий, если в дальнейшем вы будете вносить изменения). Внутреннюю таблицу стилей применяют, если хотят передать кому-либо полную Web-страницу в едином файле, например, послать кому-нибудь по электронной почте вашу главную Web-страницу. Вы также можете применить внутреннюю таблицу стилей, если знаете, что не будете использовать ее правила на другой странице.

- *Встроенный стиль* — это способ вставки языка таблицы стилей прямо в начальный тег HTML-элемента. На первый взгляд этот вариант выглядит сомнительно. Вы уже усвоили, что следует избегать встроенных инструкций форматирования на Web-странице, т. к. описание форматирования, как правило, длинно и неуклюже. И это правда, но иногда можно применить встроенный стиль для однократного форматирования на скорую руку. Такой подход нельзя назвать строгим или структурным, но он срабатывает.

Перечисленные возможности обеспечивают гибкость и позволяют следовать CSS-идеологии без оговорок (с помощью внешних таблиц стилей) или изредка идти на компромисс (применяя внутренние таблицы стилей или встроенные стили). Поскольку язык таблиц стилей один и тот же даже при половинчатом подходе, таком как внутренние таблицы, вы всегда можете вырезать и вставить правила во внешнюю таблицу стилей, когда будете готовы добиваться большей структуризации.

НА ПРОФЕССИОНАЛЬНОМ УРОВНЕ

CSS-совместимость браузеров

Прежде чем использовать таблицы стилей, неплохо было бы убедиться в том, что они работают во всех браузерах, применяемых посетителями вашего сайта. Это не так легко выяснить, как кажется на первый взгляд, потому что существует несколько версий CSS-стандарта: первоначальная CSS1, слегка улучшенная CSS2, откорректированная CSS2.1 и почти готовая CSS3. Но настоящая проблема в том, что браузеры необязательно поддерживают весь CSS-стандарт целиком, независимо от его версии.

В этой книге мы сосредоточимся на свойствах CSS1 и CSS2, про которые известно, что они хорошо поддерживаются всеми основными браузерами. В некоторых редких случаях будут рассматриваться свойства, до сих пор не имеющие широкой поддержки браузерами (например, изящное свойство `font`, описанное далее). В такой ситуации вы встретите в тексте предупреждение о риске совместимости и способы получения наилучших, максимально унифицированных результатов. Как уже говорилось, не забывайте тестировать ваши страницы в разных браузерах, чтобы убедиться в корректности их вывода.

Если вас все же беспокоит, поддерживает ли конкретная версия браузера определенное свойство CSS, можно поинтересоваться таблицами CSS-совместимости браузеров, доступными в Web. Далее приведены два полезных источника.

- Сайт QuirksMode (**www.quirksmode.org/css/contents.html**) содержит наборы таблиц, скрупулезно описывающих, какие браузеры могут обрабатывать те или иные свойства CSS. Особенно удобно, что таблицы цветные и включают разные версии популярных браузеров, таких как Firefox, Chrome, Safari, Opera и Internet Explorer с версии 5.5 до версии 9.
- Сайт When can I use (**www.caniuse.com**) содержит информацию о поддержке браузерами новейших инструментов из CSS3, HTML5 и других родственных стандартов. Вы можете выбрать интересующее вас средство, и Web-сайт сгенерирует нужную вам таблицу.

Анатомия правила

Таблицы стилей содержат только *правила*. Каждое правило — это команда форматирования, применяемая к части вашей страницы. Таблица стилей может включать одно правило или содержать десятки (и даже сотни) правил.

Далее приведено простое правило, заставляющее браузер отображать все заголовки `<h1>` синим цветом.

```
h1 { color: blue }
```

Правила CSS выглядят не так, как все, что вы видели в HTML-разметке, но у вас не возникнет проблем, если понять, что любое правило использует только три вида компонентов: *селекторы* (selector), *свойства* (property) и *значения* (value). Далее приведен формат, которому следует любое правило.

```
selector { property: value }
```

И вот, что означает каждый его компонент.

- Селектор обозначает тип контента, который вы хотите отформатировать. Браузер будет искать все фрагменты Web-страницы, соответствующие заданному селектору. Пока сосредоточимся на селекторах, которые соответствуют каждому экземпляру конкретного элемента страницы, например заголовку. Позже, в этой главе (см. *разд. "Более мощные селекторы"*), вы узнаете, как создавать более сложные селекторы, которые действуют только на отдельные части вашей страницы.
- Свойство задает тип форматирования, который вы хотите применить. С его помощью вы выбираете, что изменять — цвета, шрифты, выравнивание или что-то еще.
- Значение задает конкретное значение свойства, определенного выше. Именно с его помощью вы добиваетесь нужного результата. Например, если свойство — цвет, значением может быть светло-синий.

Конечно, редко бывает достаточно отформатировать лишь одно свойство HTML-элемента. Обычно нужно одновременно задать формат в нескольких свойствах. В таблице стилей это делается с помощью следующего правила.

```
h1 { text-align: center;
color: black; }
```

В приведенном примере изменяется цвет и центрируется текст в элементе `<h1>`. Вот почему в правилах определения стиля используются смешные фигурные скобки — `{` и `}` — с их помощью можно сгруппировать столько команд форматирования, сколько хотите. Одно свойство от другого отделяется точкой с запятой (`;`). Последнее свойство заканчивать точкой с запятой необязательно, но Web-профессионалы часто ставят ее, чтобы упростить добавление новых свойств в конец правила.

> ***Совет***
>
> Как и в HTML-файле, в CSS-файлах вы можете применять пробелы, отступы и переносы строк практически в любом месте. Разработчики часто помещают каждую команду форматирования в отдельную строку (как в ранее приведенном примере), чтобы облегчить чтение таблиц стилей.

С другой стороны, может возникнуть желание создать одну команду форматирования, воздействующую на несколько элементов. Например, вы хотите, чтобы у заголовков первых трех уровней, от h1 до h3, были синие буквы. Вместо того чтобы писать три отдельных правила, можно создать селектор, включающий все три элемента, разделенные запятыми. Далее приведен пример.

```
h1, h2, h3 { color: blue }
```

Хотите верьте, хотите нет, но селекторы свойства и значения — суть CSS. Разобравшись с этими тремя компонентами, вы очень быстро станете специалистом по разработке таблиц стилей.

Далее перечислены несколько побочных эффектов системы таблиц стилей, о которых вы, возможно, пока не догадываетесь.

- Одно правило может форматировать значительную часть HTML-разметки. Если вы применяете правило к виду селекторов, описанному выше (называемому *селектором типа*), правило действует на каждый такой элемент. Поэтому, если вы задаете синие заголовки уровня h1, как в приведенном ранее примере, все элементы h1 на вашей странице станут синими.
- Только вам решать, какую часть вашего контента форматировать. Можно заняться тонкой настройкой каждого HTML-элемента на вашей странице, или с помощью метода, обсуждаемого в *разд. "Более мощные селекторы"*, написать правила, воздействующие только на один элемент.
- Можно создать два разных правила для одного и того же элемента. Например, вы можете создать правило, изменяющее шрифт заголовков всех уровней (<h1>, <h2>, <h3> и т. д.), а затем вставить еще одно правило, которое изменяет цвет лишь элементов <h1>. Только следите за тем, чтобы не задавать многократно одно и то же свойство с конфликтующими значениями, иначе будет трудно предсказать результаты.
- У некоторых элементов есть встроенные правила стилей. Например, браузеры отображают жирным шрифтом текст в элементе <b>, даже если для этого нет правила в таблице стилей. Аналогичным образом текст в заголовке <h1> браузеры отображают крупным шрифтом, не нуждаясь в специальном правиле в таблице стилей. Но вы можете переопределить некоторые или все встроенные правила с помощью пользовательских стилевых правил. Например, можно явно установить размер шрифта в заголовке <h1> так, что он окажется *мельче* обычного текста. Точно так же можно убрать подчеркивание ссылок, заставить элемент <b> курсивить текст, а не делать его жирным и т. д.

Не старайтесь запоминать свойства и значения, которые они могут принимать. Позже в этой главе вы узнаете, как работают таблицы стилей, а затем познакомитесь со всеми командами форматирования, которые сможете использовать.

Присоединение таблицы стилей к странице

Пора посмотреть на таблицы стилей в действии. Прежде чем двигаться дальше, найдите файл resume.htm, с которым вы работали в *главе 2* (он также есть на стра-

нице **Missing CD** (Отсутствующий CD) по адресу **www.missingmanuals.com/cds/caw3**). Вы воспользуетесь им для тестирования новой таблицы стилей. Выполните следующие действия.

1. Создайте таблицу стилей, открыв новый файл в любом текстовом редакторе, например Notepad или TextEdit.

   Создание таблицы стилей не отличается от создания HTML-страницы — и то, и другое — просто текст. Во многих HTML-редакторах есть встроенная поддержка таблиц стилей (дополнительную информацию см. далее во *врезке "Секреты ремесла. Создание таблиц стилей с помощью редакторов Web-страниц"*).

2. Введите следующее правило в вашу таблицу стилей.

   ```
   h1 { color: fuchsia }
   ```

   Это правило заставляет браузер отображать все элементы `<h1>` яркими пурпурными (цвета фуксии) буквами.

3. Сохраните таблицу стилей с именем resume.css.

   Как и HTML-документу, файлу с таблицей стилей можно присвоить любое имя. Однако в соответствии с соглашением таблицы стилей почти всегда используют расширение css. В нашем примере убедитесь, что вы сохранили таблицу стилей в той же папке, что и вашу HTML-страницу.

4. Затем откройте файл resume.htm.

   Если под руками нет файла resume.htm, можно протестировать данную таблицу стилей с помощью любого HTML-файла, в котором есть хотя бы один элемент `<h1>`.

5. Вставьте элемент `<link>` в ваш HTML-файл.

   Элемент `<link>` указывает вашему браузеру, где искать таблицу стилей, написанную для ваших страниц. Элемент `<link>` следует поместить в раздел `<head>` вашей HTML-страницы. Далее приведен исправленный раздел `<head>` в файле resume.htm с вставленным элементом `<link>`.

   ```
   <head>
     <title>Hire Me!</title>
     <link rel="stylesheet" href="resume.css" />
   </head>
   ```

   В элемент `<link>` включены два компонента. Атрибут `rel` свидетельствует о том, что ссылка указывает на таблицу стилей. Атрибут `href` — важный компонент; он определяет местонахождение вашей таблицы стилей (href — сокращение от "hypertext reference" (гипертекстовая ссылка)). Если вы поместили вашу таблицу стилей в ту же папку, что и HTML-файл, достаточно задать имя файла. (Если оба файла находятся в разных папках, необходимо указать местонахождение CSS-файла с помощью полного имени файла, см. *разд. "Относительные ссылки и папки" главы 8*.)

6. Сохраните HTML-файл, а затем откройте его в браузере.

   Вот что происходит при этом. Браузер начинает обрабатывать HTML-документ и находит элемент <link>, который заставляет найти связанную с документом таблицу стилей и применить все ее правила. Далее браузер читает первое (и единственное в данном случае) правило в таблице стилей. Для применения этого правила он начинает анализировать селектор, указывающий на все элементы <h1>. Далее браузер находит на HTML-странице все элементы <h1> и применяет форматирование пурпурным цветом.

Таблица стилей в данном примере не слишком выразительна. В действительности кажется, что для получения розового заголовка требуется много работы. Но после создания базовой модели вы быстро начнете извлекать из нее выгоду. Например, можно отредактировать таблицу стилей для изменения шрифта ваших заголовков <h1> в файле resume.htm. Или можно добавить правила для форматирования других частей документа. Просто сохраните исправленную таблицу стилей и затем обновите отображение Web-страницы, чтобы увидеть эффект от изменения или вставки правил.

Чтобы увидеть, как это работает, измените таблицу стилей в файле resume.css, включив в нее приведенные далее правила.

```
body {
  font-family: Verdana,Arial,sans-serif;
  font-size: 83%;
}

h1 {
  border-style: double;
  color: fuchsia;
  text-align: center;
}

h2 {
  color: fuchsia;
  margin-bottom: 0px;
  font-size: 100%;
}

li {
  font-style: italic;
}

p {
  margin-top: 2px;
}
```

Эти правила меняют шрифт для всего документа (с помощью правила для элемента <body>), модифицируют заголовки двух уровней, делают курсивным начертание

пунктов списка и уменьшают свободное пространство между абзацами. Хотя вы не поймете все эти правила сразу, общая модель сохраняется как в предыдущем примере с резюме (контент на Web-странице, форматирование в таблице стилей). На рис. 6.2 показан результат.

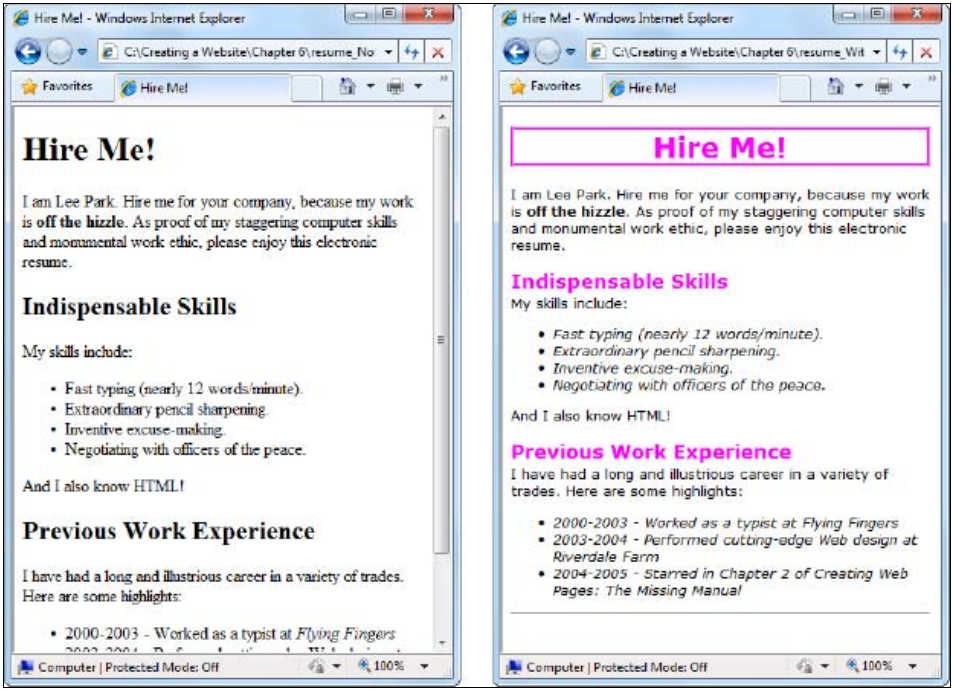


Рис. 6.2. Слева: к этому времени вы уже можете распознать типовую Web-страницу. Справа: таблица стилей реконструирует всю страницу

СЕКРЕТЫ РЕМЕСЛА

Создание таблиц стилей с помощью редакторов Web-страниц

У некоторых редакторов Web-страниц, таких как Dreamweaver и Expression Web, есть удобные средства для редактирования таблиц стилей. Для их испытания откройте уже имеющуюся таблицу стилей или создайте новую. Для создания таблицы стилей в Dreamweaver выберите команды **File | New** (Файл | Создать), задайте **CSS** в списке **Page Type** (Тип страницы), а затем щелкните мышью кнопку **Create** (Создать). В Expression Web для создания таблицы стилей выберите **File | New | CSS** (Файл | Создать | CSS).

До этого момента вы не увидите ничего удивительного. Самое интересное начнется, когда вы попробуете редактировать вашу таблицу стилей. Во время ввода правила редактор Web-страниц раскроет список возможных свойств и значений (рис. 6.3). Если копнете глубже, то обнаружите, что в обоих редакторах есть окна, позволяющие формировать стили с помощью указателя мыши и щелчков ее рабочей кнопки, а также удобных клавиатурных ускорителей команд для применения стилей к элементам страницы.

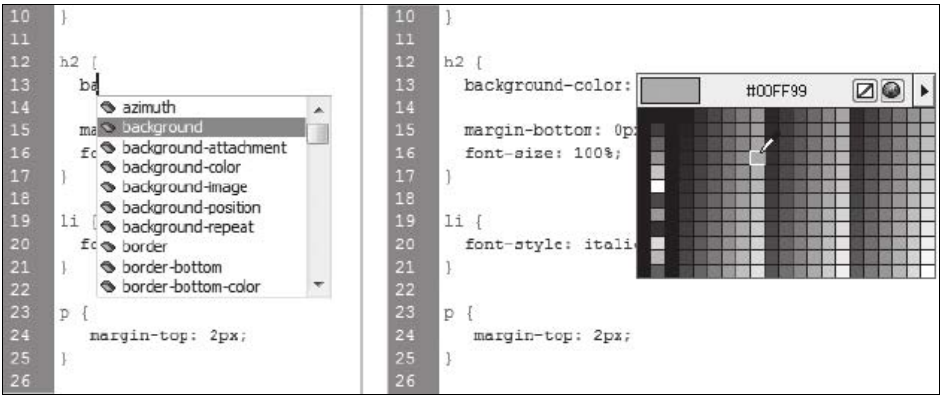


Рис. 6.3. Когда вы редактируете таблицу стилей в Dreamweaver, редактор раскрывает список возможных стилевых свойств (слева) и их значений. Если вы работаете с цветами, вы даже получите в свое распоряжение удобную палитру (справа), которая преобразует коды, обозначающие цвета в вашей таблице стилей, в реальный цвет и отображает результаты. Это чудесное подспорье для слабой памяти, избавляющее от немалого количества нажатий клавиш

Применение внутренней таблицы стилей

Пример resume.css демонстрирует использование внешней таблицы стилей. *Внешние таблицы стилей* — самый популярный способ применения CSS, т. к. позволяют связать единственную любовно созданную таблицу стилей с любым количеством Web-страниц. Но бывают периоды, когда вы не работаете над целым Web-сайтом и были бы рады найти менее масштабное решение.

Внутренняя таблица стилей — это таблица стилей, которую вы не связываете со страницей, а *встраиваете* в код вашей Web-страницы, в раздел `<head>`. Да, она удлиняет ваши страницы, и вам придется для каждой страницы создавать отдельную таблицу стилей. Но порой иметь один файл, содержащий и вашу страницу, и ее стилевые правила, настолько удобно, что такой подход заслуживает внимания.

Для замены в предыдущем примере внешней таблицы стилей на внутреннюю удалите элемент `<link>` из вашей HTML-разметки и вставьте стилевые правила в элемент `<style>` внутри раздела `<head>` вашей страницы, как показано далее.

```
<head>
  <title>Hire Me!</title>
  <style>
     h1 { color: fuchsia }
  </style>
</head>
```

Применение встроенных стилей

Если вы совсем не хотите писать таблицу стилей, можно воспользоваться еще одним приемом. *Встроенные стили* позволяют вставить свойство и значение из правила таблицы стилей прямо в начальный тег HTML-элемента. Селектор при этом

задавать не нужно, т. к. браузеры понимают, что вы хотите отформатировать только тот элемент, в который вставили правило.

Далее показано, как использовать встроенный стиль для форматирования одного заголовка.

```
<h1 style="color: fuchsia">Hire Me!</h1>
```

Приведенное правило влияет только на элемент <h1>, в который вы его вставили; другие заголовки <h1> на странице останутся без изменений.

На первый взгляд встроенные стили могут показаться привлекательными, они понятны и просты. Вы задаете команду форматирования именно там, где хотите ее применить. Но если попытаться отформатировать всю страницу таким способом, вы поймете, почему Web-разработчики с осторожностью пользуются ими. Все очень просто. Типовое правило форматирования CSS длинное. Если вам придется помещать его рядом с вашим контентом и копировать для каждого добавляемого элемента, очень скоро ваша Web-страница будет изуродована до неузнаваемости. Далее приведен пример с более сложным заголовком, которому требуется несколько стилевых правил.

```
<h1 style="border-style: double; color: fuchsia; text-align: center">Hire
Me!</h1>
```

Даже если такая вставка встретится в документе всего один раз, она все равно выглядит неряшливым уродством. Поэтому старайтесь, если можно, не применять встроенные стили.

ДЛЯ ТЕХ, КТО ПОНИМАЕТ

Увеличение скорости за счет таблицы стилей

Внешние таблицы стилей — более эффективный способ форматирования Web-сайта по сравнению с внутренними таблицами стилей и встроенными стилями, потому что браузеры используют *кэширование*. Кэширование — это метод повышения производительности, заставляющий браузеры хранить копию некоторой загруженной из Интернета информации на вашем компьютере и избегать повторной загрузки.

Когда браузер загружает Web-страницу, ссылающуюся на таблицу стилей, он формирует отдельный запрос на эту таблицу стилей (см. рис. 6.1). Но если браузер загрузит другую страницу, использующую ту же таблицу стилей, он сообразит, что уже имеет под руками нужный CSS-файл. В результате ему не придется формировать второй запрос. Вместо этого он воспользуется таблицей стилей из кэша, что немного сократит время загрузки страницы. (Конечно, браузеры хранят содержимое кэша недолго. Если вы отправитесь на тот же сайт завтра, браузерам придется снова запрашивать таблицу стилей.)

Если вы встроите таблицу стилей во все ваши Web-страницы, браузер всегда будет загружать полную страницу с дубликатом таблицы стилей. У него нет способа узнать, что вы применяете один и тот же набор правил снова и снова. Хотя разница в скорости загрузки невелика, она может сказаться при загрузке Web-сайта с большим числом страниц. Скорость — еще одна причина, по которой опытные Web-разработчики предпочитают внешние таблицы стилей.

Каскад

Вас, наверное, заинтересовало, что означает слово "каскадные" в названии "каскадные таблицы стилей". Оно связано со способом определения браузерами приоритета правил при использовании нескольких наборов правил.

Например, если во внешней таблице стилей указано, что заголовки <h1> должны выводиться синими буквами, а затем вы примените жирное начертание с помощью встроенного стиля, в результате получится сумма двух изменений: заголовок синего цвета с жирным начертанием шрифта. Но что произойдет, если правила конфликтуют друг с другом? Например, одно правило задает синий текст, а другое настаивает на красном, какой цвет победит?

Для того чтобы узнать, у какого правила наивысший приоритет, просмотрите приведенный далее перечень. Он определяет порядок применения стилей в браузере. У пунктов списка с бóльшим номером приоритет выше: браузер применяет их после того, как применит стили из предыдущих пунктов, и они переопределяют любое ранее выполненное форматирование.

1. Стандартные настройки браузера.
2. Внешняя таблица стилей.
3. Внутренняя таблица стилей (внутри элемента <head>).
4. Встроенный стиль (внутри HTML-элемента).

Итак, если внешняя таблица стилей конфликтует с внутренней таблицей стилей, внутренняя таблица победит.

Основываясь на этом, вы можете решить, что из этого каскадного поведения можно извлечь выгоду и определять общие правила во внешних таблицах стилей, а затем переопределять их за редким исключением с помощью встроенных стилей. Действительно вы можете так поступать, но есть решение гораздо лучше. Вместо того чтобы форматировать отдельные элементы с помощью встроенных свойств, для форматирования конкретных частей страницы вы можете применить *селекторы класса*, о которых вы узнаете чуть позже в этой главе.

> ***Примечание***
>
> Слово "каскадные" в каскадных таблицах стилей немного обманчиво, поскольку в большинстве случаев вы не будете применять несколько типов таблиц стилей (просто потому, что это может быстро привести к путанице). Большинство Web-дизайнеров предпочитают внешние таблицы стилей раз и навсегда.

Наследование

Наряду с каскадными стилями есть и еще тесно связанное понятие — *наследование* стиля. Для того чтобы понять наследование стилей, нужно вспомнить, что в HTML-документе один элемент может содержать другие элементы. Помните элемент маркированного списка ? Он содержит элементы пунктов списка <li>. Аналогичным образом элемент абзаца <p> может содержать элементы символьного форматирования, например, <b> и <i>, а элемент <body> содержит весь документ.

Благодаря наследованию, когда вы применяете команды форматирования к элементу, содержащему *другие* элементы, правило форматирования применяется к *каждому из этих других элементов*. Например, если для элемента <body> вы задаете шрифт Verdana (как в таблице стилей для резюме, приведенной ранее), все элементы внутри элемента <body>, включая все заголовки, абзацы, списки и т. д., будут выводиться шрифтом Verdana.

> ***Примечание***
>
> Элементы наследуют большинство, но не все стилевые свойства. Например, элементы никогда не наследуют от другого элемента поля. Во всех таблицах этой главы обращайте внимание на столбец "Наследуется?", чтобы понять, передает ли CSS настройку свойства из одного элемента в другой с помощью наследования.

Однако существует небольшая хитрость. Порой правила форматирования могут накладываться. В таком случае самое узкоизбирательное правило, т. е. то, которое иерархически ближе всего к элементу, и побеждает. Например, свойства, заданные вами для элемента <h1>, переопределят свойства, заданные для элемента <body> и касающиеся всех заголовков уровня 1. Рассмотрим следующую таблицу стилей.

```
body {
  color: black;
  text-align: center;
}

ul {
  color: fuschia;
  font-style: italic;
}

li {
  color: red;
  font-weight: bold;
}
```

Приведенные правила накладываются. В типичном документе (рис. 6.4) вы помещаете <li> (пункт списка) внутрь элемента списка <ul>, который в свою очередь находится внутри элемента <body>. В данном случае текст в каждом пункте списка будет красным, т. к. правило для <li> переопределяет правила для <ul> и <body>, вступающие в силу первыми.

Умелые разработчики таблиц стилей могут обернуть такое поведение себе на пользу. Например, задать шрифт в элементе <body> так, чтобы все элементы на вашей Web-странице, заголовки, абзацы текста, списки и т. д. отображались одним шрифтом. Затем вы можете сознательно переопределить этот шрифт лишь для нескольких элементов, применив правила форматирования, относящиеся к конкретным элементам.

> ***Примечание***
>
> Несмотря на то, что вы вряд ли слишком часто увидите таблицы стилей в действии, почти наверняка вы будете использовать наследование.

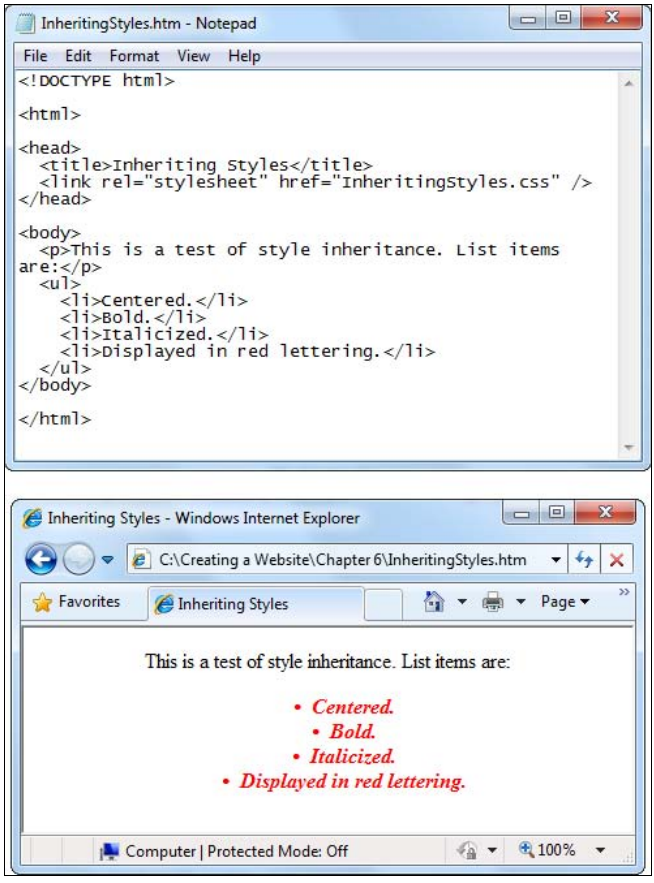


Рис. 6.4. Когда правила вступают в конфликт, побеждает самое конкретное из них. В этом примере ваш браузер отображает пункты списка красным цветом, потому что правило для элемента `<li>` переопределяет свойства, унаследованные из элементов `<ul>` и `<body>` (вверху). Но элемент сохраняет стиль унаследованного правила, если оно не конфликтует с другим правилом. В этом примере это означает, что элемент `<li>` будет выводиться курсивом и центрироваться благодаря наследованию (внизу)

Более мощные селекторы

До сих пор вы видели правила таблицы стилей, применяемые к каждому вхождению конкретного HTML-элемента. Селекторы в этих универсальных стилях называют *селекторами типов*.

Селекторы типов — мощное средство, но недостаточно гибкое. Порой вам нужно изменить лишь один раздел вашего HTML-документа или даже единственный элемент. Можно применить встроенные стили и поместить форматирование в тег реального элемента, но это грубо. К счастью, таблицы стилей предлагают два практических решения этой проблемы: селекторы классов и селекторы идентификаторов.

Селекторы классов

Селекторы классов — это одно из лучших средств таблицы стилей. Сначала вы отбираете конкретные элементы на вашей странице, присвоив им одно и то же *имя класса*. Затем вы заставляете браузер применить форматирование ко всем элементам с заданным именем класса.

Для проведения эксперимента начнем с выбора описательного имени класса. Вы можете выбрать любое имя, состоящее из букв, цифр и дефисов, но первый символ в нем всегда должен быть буквой. В следующем примере используется имя класса `FancyTitle`.

После выбора имени класса необходимо в вашей таблице стилей определить правило для этого класса. Это правило выглядит как любое другое, за исключением того, что в нем вместо имени тега в качестве селектора используется имя класса, которому предшествует точка (`.`).

```
.FancyTitle {
  color: red;
  font-weight: bolder;
}
```

Но как браузер узнает, когда применять правило, использующее селектор класса? Браузер никогда не применяет правила для класса автоматически; вы должны вставить имена классов в элементы, которые хотите форматировать. Далее приведен пример, использующий класс `FancyTitle` в заголовке.

```
<h3 class="FancyTitle">Learning to Embroider</h3>
```

Если имя класса в элементе совпадает с именем класса в таблице стилей, браузер применяет форматирование. Если браузер не может найти стиль, связанный с каким-либо именем класса, ничего не происходит.

> ***Примечание***
>
> Правила для классов действуют в дополнение к любым другим правилам. Например, если вы создали правило для элемента `<p>`, оно применяется ко всем абзацам, включая те, которые относятся к специализированному классу. Но если правило для класса конфликтует с любыми другими правилами, это правило побеждает.

Вы также можете создать правило, у которого есть имя класса и задан тип элемента. Например, если известно, что вы хотите применить класс `FancyTitle` только в элементе `<h3>`, можно написать следующее правило.

```
h3.FancyTitle {
  color: red;
  font-weight: bolder;
}
```

Большинство Web-дизайнеров применяют как элементозависимые правила для классов, так и более общие правила классов (те, в которых не задан тип элемента). Несмотря на то, что можно оставаться приверженцем исключительно общих правил, если известно, что определенный набор форматирования будет использоваться

только в элементе конкретного типа, неплохая идея четко обозначить этот факт, применив элементозависимое правило класса. В этом случае вы легче вспомните назначение правила, когда в дальнейшем будете редактировать вашу таблицу стилей.

Селекторы идентификаторов

У селекторов классов есть очень близкий родственник — *селектор идентификатора*. Как и селектор класса, селектор идентификатора позволяет выбрать описательное имя. Но вместо точки вы используете знак решетки (#), как показано далее.

```
#Menu {
  border-width: 2px;
  border-style: solid;
}
```

Прямо сейчас это правило `Menu` можно применить к любому элементу. Но вы можете ограничить сферу действия селектора идентификатора конкретным типом элемента, поместив имя элемента перед знаком решетки.

```
div#Menu {
  ...
}
```

В этом примере определен селектор идентификатора с именем `Menu`, который может применяться только к элементам `<div>`.

Как и в случае правил для классов, браузеры не применяют правила для идентификаторов, если вы не сообщите им об этом явно в вашем HTML-коде. Но вместо включения правила с помощью атрибута `class` вы делаете это посредством атрибута `id`. Например, далее описан элемент `<div>`, использующий стиль `Menu`.

```
<div id="Menu">...</div>
```

Сейчас вы, наверное, думаете, зачем нужен селектор идентификатора, кроме всего прочего, он кажется почти точно таким же, как селектор класса. Но есть одно отличие: вы можете присвоить заданный идентификатор только *одному* элементу на странице. В приведенном примере это означает, что с помощью идентификатора `Menu` можно пометить только один элемент `<div>`. У имен классов нет такого ограничения, их можно применять столько раз, сколько захотите.

Селектор идентификатора — удачный выбор, если нужно отформатировать единственный, никогда не повторяющийся элемент на вашей странице. Достоинство подобного решения в том, что селектор идентификатора ясно подчеркивает особую важность элемента. Например, если на странице есть селектор идентификатора с именем `Menu` или `NavigationBar`, Web-дизайнер сразу же понимает, что на странице отображается только одно меню или навигационная панель. Конечно, у вас никогда не возникнет острой необходимости в применении селектора идентификатора. Некоторые Web-разработчики используют селекторы классов для всего, независимо от уникальности или повторяемости раздела. Это только дело вкуса.

Примечание

В последнем разделе этой главы вы познакомитесь с еще одним типом селектора — контекстными селекторами. Но сейчас пора обратить внимание на другое. В следующих разделах вы преумножите добытые с трудом знания о таблицах стилей, изучив десятки свойств форматирования, которые сможете изменять.

Цвета

Добавить немного цвета на ваши Web-страницы совсем нетрудно. В правилах таблицы стилей есть два свойства, связанных с цветами (табл. 6.1). В следующих разделах вы узнаете о типах значений, которые сможете применять при настройке цветов (имена цветов, их коды и RGB-значения).

Таблица 6.1. *Свойства цвета*

Свойство	Описание	Стандартные значения	Наследуется?
`color`	Цвет текста. Это свойство — удобный способ выделения заголовков или текста с логическим ударением	Имя цвета, код цвета или RGB-значение цвета	Да
`background-color`	Цвет фона только для данного элемента	Имя цвета, код цвета или RGB-значение цвета. Вы также можете использовать слово "transparent" (прозрачный)	Нет[1]

[1] Свойство `background-color` не наследуется. Если вы зададите в разделе страницы `<body>` синий фон и затем поместите на страницу заголовок, заголовок не унаследует синий фон. Но есть одна хитрость. Если вы не присваиваете явно элементу цвет фона, его цвет прозрачен. Это означает, что сквозь него виден цвет фона содержащего его элемента, т. е. тот же эффект, что и при наследовании. Поэтому заголовок в данном примере будет отображаться на синем фоне.

Свойство `color` понятно; это цвет текста. Свойство `background-color` несколько более необычно.

Если применить свойство цвета фона к элементу `<body>` на Web-странице, как вы и ожидаете, вся страница приобретет синий цвет. Но если задать цвет фона для отдельного элемента, такого как заголовок, результаты будут немного странными. И все потому, что CSS считает, что каждый элемент заключен в невидимую прямоугольную рамку. Когда вы применяете цвет фона к элементу, CSS закрашивает цветом фона именно этот прямоугольник.

Например, в следующей таблице стилей на странице задаются разные цвета фона для заголовков, абзацев и любого жирного начертания текста.

```
body {
  background-color: yellow;
}
```

```
h1 {
  color: white;
  background-color: blue;
}

p {
  background-color: lime;
}

b {
  background-color: white;
}
```

На рис. 6.5 показан результат.

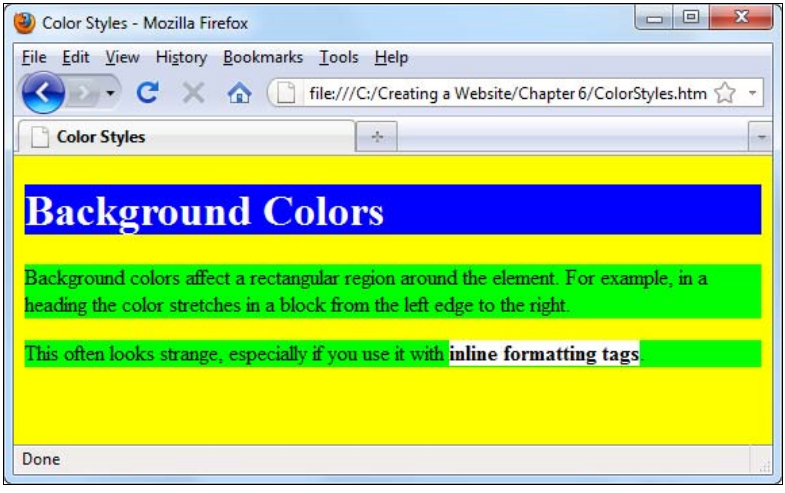


Рис. 6.5. Если вы задаете цвет фона в таком элементе, как <h1>, правило CSS закрашивает только данную строку. Если вы применяете цвет фона в строчном элементе, таком как <b> или , он воздействует только на слова в этом элементе. Оба результата выглядят странно — кажется, что здесь разгулялся некто с набором фломастеров. Гораздо лучше задавать цвет фона для всей страницы в элементе <body> или подкрашивать большую прямоугольную область страницы (например, боковую панель) с помощью элемента-контейнера, такого как

Задание цвета

Задать цвет — это значит найти код, обозначающий точный оттенок вашего любимого ярко-синего цвета. Добиться этого можно несколькими способами. Во-первых, вы можете задать ваш цвет простым названием на английском (lime — зеленый). К сожалению, эта система работает лишь с небольшим набором из 16 цветов (aqua — цвета морской волны, black — черный, blue — синий, fuchsia — пурпурный, gray — серый, green — зеленый, lime — зеленый, цвета лайма, maroon — темно-бордовый, navy — темно-синий, olive — оливковый, purple — лиловый, red — красный, silver — серебристый, teal — бирюзовый, white — белый и yellow — жел-

тый). Некоторые браузеры принимают и другие названия, но нет гарантии их одинаковой поддержки в разных браузерах, поэтому лучше воспользоваться другим подходом. CSS предлагает еще две возможности: шестнадцатеричные значения цветов и RGB-значения.

Шестнадцатеричные значения цветов

В случае шестнадцатеричных значений цветов вы пользуетесь странно выглядящим кодом, начинающимся со знака решетки (#). Строго говоря шестнадцатеричные значения цвета включают в себя три числа, представляющие количества красного, зеленого и синего цветов, участвующие в формировании цвета (комбинируя разные количества этих трех основных или первичных цветов, можно создать любой цвет.) Но в шестнадцатеричных значениях цветов эти три ингредиента превращаются в тайный код, абсолютно понятный компьютерам и невероятно сложный для нормальных людей. Запись цвета в шестнадцатеричной форме вы легко найдете в разных частях Web, потому что это первоначальный формат задания цвета в языке HTML. Но эта форма так же понятна, как набор нулей и единиц, приводящий в действие ваш компьютер.

Далее приведен пример.

```
body {
  background-color: #E0E0E0
}
```

Даже компьютерный суперспециалист не скажет вам, что `#E0E0E0` означает применение светло-серого фона. Для того чтобы выяснить код вашего любимого цвета, прочтите *разд. "Поиск нужного цвета"*.

RGB-значения цветов

Еще один способ задания цвета — RGB-значения. В соответствии с этим более логичным подходом вы просто указываете, сколько нужно добавить красного, зеленого и синего цветов "в смесь" для создания окончательного цвета. Каждый компонент задается значением из диапазона 0—255. Например, цвет, составленный из красного, зеленого и синего со значениями 255 каждый, получается белым; с другой стороны, если задать для всех компонентов 0, генерируется черный.

Далее приведен пример для формирования приятного зеленого цвета.

```
body {
  background-color: rgb(177,255,20)
}
```

Поиск нужного цвета

Таблицы стилей способны справиться с любым цветом, который только можно вообразить. Но как найти код цвета для прекрасного оттенка закатного оранжевого (или цвета красной рыбы), если он вам нужен?

К сожалению, нет способа в этой черно-белой книге показать вам ваши возможности. Но в Интернете существует множество отличных программ выбора цвета (палитр). Например, зайдите на Web-сайт **www.colorpicker.com**, где вам придется только перемещать вашу мышь по градиенту нужного вам цвета (и отмечать его шестнадцатеричный код). Или проверьте страницу **www.colorschemer.com/online.html**, на которой сгруппированы вместе дополняющие цвета, что особенно полезно для создания Web-сайтов, производящих впечатление разработанных профессионалами. Если вы пользуетесь средствами Web-дизайна, такими как Dreamweaver или Expression Web, у вас есть еще более легкий вариант — встроенная в программу интеллектуальная палитра (см. рис. 6.3).

Примечание

RGB-система позволяет выбрать любой из 16.7 миллионов цветов, т. е. ни один Web-сайт для выбора цвета не сможет показать все возможные RGB-значения цветов (если же ему удастся сделать это, ни в коем случае не нажимайте кнопку **Print** (Печать); даже разместив по 10 цветов в строке, вы получите тысячи страниц). Взамен большинство сайтов ограничиваются репрезентативной выборкой из всего множества цветов. Это срабатывает, потому что многие цвета настолько близки, что их почти невозможно отличить.

Стандарт RGB-цветов широко применяется и в других компьютерных программах. Например, если вы видите ваш любимый цвет в профессиональной графической программе, такой как Photoshop (или менее профессиональной графической программе Windows Paint), с вероятностью 50% существует способ получения значений красного, зеленого и синего компонентов для этого цвета. Это позволяет с легкостью добиться совпадения цвета текста на вашей Web-странице с цветом на изображении. И теперь, когда есть такое средство, будет доволен даже самый придирчивый дизайнер интерьера.

УГОЛОК ДИЗАЙНЕРА

Как добиться привлекательного цвета

Для создания четких, ясных, легко читающихся Web-страниц со своей индивидуальностью нет ничего лучше черного текста на белом фоне. Эта сочетание черного с белым отлично подходит и для страниц с множеством полноцветных изображений. Неслучайно почти все первоклассные Web-сайты, новостные сайты (**www.cnn.com**), поисковые машины (**www.google.com**), магазины электронной торговли (**www.amazon.com**) и аукционные дома (**www.ebay.com**) используют победное сочетание черного на белом.

Но что если вы слишком любите цвет, чтобы оставить вашу Web-страницу традиционно черно-белой? Лучший совет — следуйте золотому правилу применения цвета: *сохраняйте сдержанность*. Если вы не разрабатываете сайт, возрождающий шестидесятые годы или страницу, посвященную группе Led Zeppelin, вряд ли вы захотите, чтобы ваши страницы вышли из-под контроля от многоцветия. Вот несколько советов, позволяющих добавить красочное цветное пятно на вашу страницу и в то же время не дать цвету целиком захватить власть на ней.

- Придерживайтесь одного цвета. Это означает применение черного, белого и еще одного темного цвета. Используйте новый цвет для выделения важного конструктивного элемента, например подзаголовков в статье. Web-сайт журнала "Time" применяет фирменный красный цвет для некоторых ссылок и фрагментов текста.

- Применяйте едва окрашенные фоны светлых оттенков. Порой слабый акварельный фон — все, что нужно для оживления сайта. Например, легкий бронзовый или золотистый цвета могут придать элегантность и изысканность (см. сайт библиотеки Гарвардского университета по адресу **http://lib.harvard.edu**). Светло-розовые или желтые тона могут подготовить посетителей к покупке ночных рубашек или другой женской одежды на сайте Victoria's Secret (**www.victoriassecret.com**).
- Используйте текст в прямоугольной рамке. Web-дизайнеры часто применяют закрашенные прямоугольники для выделения важных областей Web-страницы (посмотрите в Википедии вот эту страницу **http://en.wikipedia.org**). Чуть позже в этой главе вы узнаете, как создавать прямоугольные рамки.
- С осторожностью пользуйтесь белым текстом. Белый текст на черном или темно-синем фоне может быть поразительным — его очень трудно читать. Практическое правило гласит: избегайте его, если не стремитесь создать футуристический, нетрадиционный и сумрачный облик вашего Web-сайта. (Даже если вы попадаете в одну из этих категорий, можно добиться большего эффекта с помощью белого фона и нескольких хорошо подобранных графических изображений с эффектными волнующими цветами.)

Выравнивание и разгонка текста

CSS содержит множество свойств, которые позволяют управлять внешним видом текста на Web-странице. Если вас когда-либо интересовало, как создать абзацные отступы, растянуть строки или отцентрировать заглавие, вам нужны именно эти средства.

В табл. 6.2 приведено подробное описание всех вариантов выравнивания.

Таблица 6.2. *Свойства выравнивания и разгонки текста*

Свойство	Описание	Типичные значения	Наследуется?
`text-align`	Выравнивает текст по одному или обоим краям страницы	`left`, `right`, `center`, `justify`	Да
`text-indent`	Создает отступ в первой строке текста (обычно абзаца)	Значение в пикселах (задает величину отступа) или в процентах от ширины содержащего текст элемента	Да
`margin`	Задает поле, пробельную зону вокруг блочного элемента (см. *разд. "HTML-элементы для основного текста" главы 5*). Для изменения поля только с одной стороны используйте похожие свойства `margin-bottom`, `margin-left`, `margin-right` и `margin-top`	Значение в пикселах или процентах, задающее величину пробельной зоны, окружающей элемент	Нет
`padding`	Задает пробельную зону внутри блочного элемента. Обладает тем же эффектом, что и `margin`, если вы не применяете рамку или цвет фона	Значение в пикселах или процентах, определяющее величину пробельной зоны, окружающей элемент	Нет

Таблица 6.2 *(окончание)*

Свойство	Описание	Типичные значения	Наследуется?
`word-spacing`	Задает расстояние между словами	Значение в пикселах или процентах	Да
`letter-spacing`	Задает расстояние между буквами	Значение в пикселах или процентах	Да
`line-height`	Задает интервал или расстояние между строками	Значение в пикселах или процентах. Можно также применять множитель (например, 2 для задания двойного интервала)	Да
`white-space`	Сообщает браузеру, как устанавливать пробельные зоны в вашем тексте	`normal`, `pre`, `nowrap`	Да

Например, если хотите создать страницу с абзацами и красной строкой (как в романе или газете), используйте следующее правило в таблице стилей.

```
p {
  text-indent: 20px
}
```

В нижеприведенных разделах вы увидите примеры, в которых используются свойства выравнивания и задания полей.

Выравнивание

По умолчанию весь текст на Web-странице выравнивается по левому краю окна браузера. С помощью свойства `text-align` можно отцентрировать этот текст или выровнять его по правому краю или по ширине страницы. На рис. 6.6 показаны возможные варианты.

Самый интересный тип выравнивания — выравнивание по ширине (или выключка), которое заставляет текст выводиться вплотную к левому и правому полям страницы, как, например, в данной книге. Задается выравнивание по ширине с помощью значения `justify`. Первоначально принтеры предпочитали выравнивание по ширине, т. к. оно позволяло разместить больше слов на одной странице, снижая общее количество страниц в книге и, следовательно, ее печатную цену. В наши дни это стиль жизни. Многие люди считают, что текст с выровненными обоими краями выглядит четче и яснее, чем текст с рваным краем, даже несмотря на то, что по результатам тестов простой невыровненный текст читается легче.

В мире Web выравнивание по ширине выглядит не так хорошо, как при печати. Главная проблема — отсутствие правил, делящих длинные слова на слоги, вставляющих мягкие переносы и переносящих их непоместившиеся части на новую строку. Браузеры пользуются довольно упрощенным методом выравнивания текста

по ширине. По сути они добавляют слова в строку по очереди, пока в строке не останется места для нового слова, в этот момент они вставляют дополнительные пробелы между словами, чтобы растянуть строку на полную длину. Для сравнения, лучшие системы верстки страниц для печати анализируют весь абзац и для каждой строки выбирают наиболее подходящую стратегию выравнивания по ширине. В сложных случаях может потребоваться помощь опытного наборщика для установки вручную переносов строк. На этом фоне Web-браузеры выглядят безнадежно отсталыми, как вы увидите на рис. 6.7.

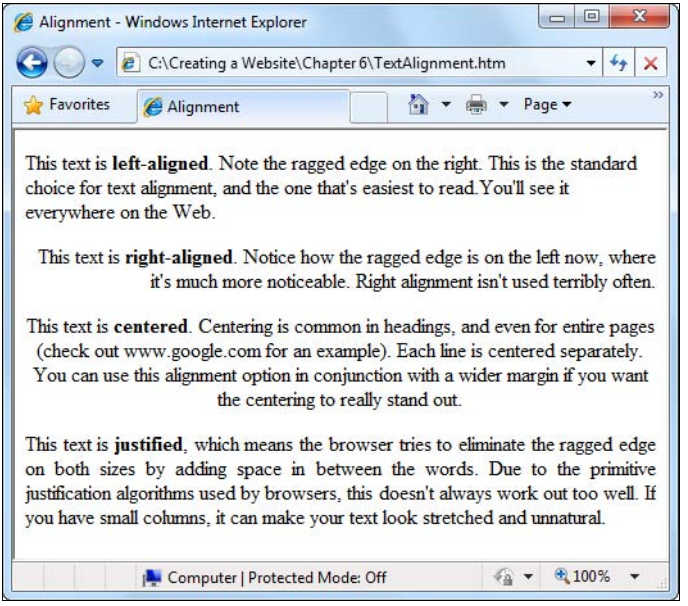


Рис. 6.6. На странице показаны типовые варианты выравнивания текста

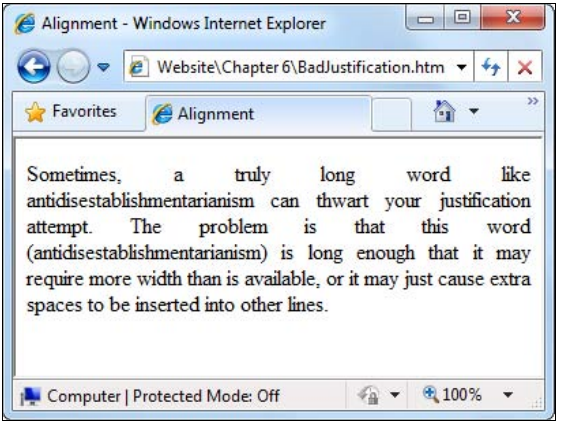


Рис. 6.7. Если вы решили добиться полного выравнивания по ширине на Web-странице, убедитесь, что ваши абзацы достаточно длинны. В противном случае вы очень быстро получите пробельные зазоры и коридоры. На очень немногих Web-сайтах применяется выравнивание по ширине

Разгонка

Для настройки величины пробельной зоны вокруг любого элемента пользуйтесь свойством `margin`. Далее приведен пример правила, которое вставляет фиксированную пробельную зону шириной 8 пикселов со всех сторон от абзаца:

```
p {
  margin: 8px;
}
```

Это конкретное правило не оказывает большого эффекта, поскольку 8 пикселов — это величина поля, которое Web-браузеры вставляют со всех сторон от блочного элемента, обеспечивая базовый объем свободного пространства. Если вы хотите создать страницы с большим объемом информации, то предоставление такого пространства вам может показаться излишне расточительным. Поэтому многие разработчики Web-сайтов ищут способы, немного ужать поля.

Один из распространенных приемов — уменьшение зазора между заголовками и следующим за ними текстом. Далее приведен пример, в котором это сжатие достигается за счет применения встроенных стилей.

```
<h2 style="margin-bottom: 0px">This heading has no bottom margin</h2>
<p style="margin-top: 0px">This paragraph has no top margin.</p>
```

Вы, наверное, заметили, что это правило использует уточняющие свойства `margin-top` и `margin-bottom`, чтобы сосредоточить внимание поочередно на одном поле. Для задания боковых полей можно применять свойства `margin-left` и `margin-right`. На рис. 6.8 показано несколько разных вариантов полей.

Многие свойства CSS поддерживают сокращенную форму записи, сжимающую несколько свойств в одно правило. Например, вместо задания каждого из четырех полей отдельно (`margin-top`, `margin-bottom`, `margin-left` и `margin-right`) вы можете настроить их все с помощью следующего правила.

```
p {
  margin: 5px 10px 15px 20px;
}
```

Это правило задает верхнее поле 5 пикселов, правое поле 10 пикселов, нижнее поле 15 пикселов и левое поле 20 пикселов. Главное обязательно отделить каждое число от следующего пробелом — не стесняйтесь, добавьте дополнительные запятые или точки с запятой.

Если вы отважны, можно даже применить *отрицательные* поля. И если довести их до крайности, два элемента могут наложиться друг на друга. Но лучший способ наложения элементов — абсолютное позиционирование, стилевой прием, описание которого вы найдете в *разд. "Фиксированные блоки" главы 9*.

Примечание

Элементы никогда не наследуют поля, в отличие от большинства других свойств CSS. Это означает, что если изменить поля одного элемента, другие элементы внутри этого не будут затронуты.

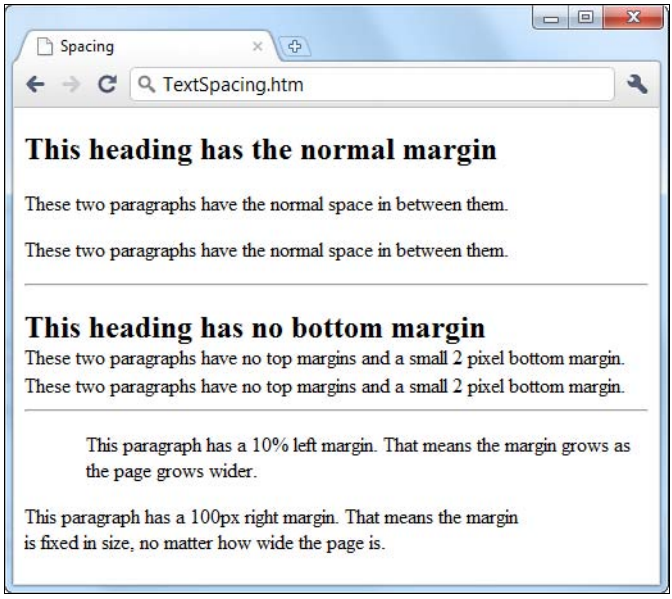


Рис. 6.8. Если хотите изменить пробельную зону между элементами страницы, такими как заголовки и абзацы, необходимо рассматривать и элемент, расположенный выше, и элемент, находящийся ниже. Например, если вы кладете один абзац на другой, заслуживают внимания два фактора — величина нижнего поля верхнего абзаца и верхнего поля нижнего абзаца. Браузеры используют большее из этих двух значений. Это означает, что нет смысла сжимать верхнее поле нижнего элемента, если вы не сжали нижнее поле верхнего абзаца. С другой стороны, если нужно расширить свободное пространство, достаточно увеличить поле одного из двух элементов

Пробелы

Как вы узнали из предыдущих глав, в языке HTML существует хитрый способ обработки пробелов. Если поместить несколько пробелов в ряд, HTML трактует первый из них как реальный символ пробела, а остальные игнорирует. Это позволяет писать легко читаемую разметку, т. к. можно вставлять пробелы, куда захотите, не беспокоясь об их влиянии на внешний вид Web-страницы.

Вы также уже знаете два способа изменения стандартной обработки пробелов браузерами: символьную последовательность ` ` и элемент `<pre>`. С помощью свойства таблицы стилей `white-space` можно заменить оба эти приема.

Сначала рассмотрим символьную последовательность ` `. Она служит двум целям: позволяет вставлять пробелы, которые браузер не будет игнорировать, и не дает браузеру разрывать строку в середине названия компании или какого-либо другого важного термина. Далее приведен пример, демонстрирующий только что описанный метод.

```
<p>You can trust the discretion of
Hush Hush Private Plumbers</p>
```

Этот прием срабатывает (на странице отображается текст "Hush Hush Private Plumbers", в котором название компании не разрывается и не переносится частично

на следующую строку), но такую разметку трудно читать. Далее приведен эквивалент из таблицы стилей со свойством white-space, значение которого равно nowrap.

```
<p>You can trust the discretion of
<span style="white-space: nowrap">Hush Hush Private Plumbers</span></p>
```

Для того чтобы предложенный прием сработал, следует в HTML-коде название компании поместить в контейнер, к которому и применяется форматирование. Хорошо подойдет для этого элемент (см. *разд. "Разделы и оболочки" главы 5*), т. к. он не задает никакого форматирования кроме того, которое вы явно в него вставите.

Теперь рассмотрим элемент <pre>, сообщающий браузеру о необходимости отображать все пробелы, содержащиеся в контенте. В *разд. "Текст в заданном формате" главы 5* вы видели, как можно использовать <pre> для вставки подходящей пробельной зоны в стихотворение Каммингса. Можно добиться того же эффекта, присвоив свойству white-space элемента (скажем, , или <p>) значение, равное pre.

```
<p style="white-space: pre">Your browser won't ignore these
                            s   p   a   c   e   s .</p>
```

Если в свойстве white-space используется значение pre, браузер отображает все пробелы, табуляции и жесткие переносы (разрывы строк, которые создаются при нажатии клавиши <Enter>). Но в отличие от элемента <pre> значение pre в свойстве white-space не изменяет шрифт текста. Если вы хотите применить моноширинный шрифт, например Courier, для пропорционального расположения букв и пробелов, необходимо вставить свойство font-family (см. *следующий раздел*).

Основные шрифты

С помощью CSS-свойств, относящихся к шрифтам, можно выбрать гарнитуру, насыщенность шрифта (его жирность) и размер (табл. 6.3). Однако приготовьтесь к небольшой неопределенности Web-стиля, это единственная ситуация, делающая жизнь труднее, чем она кажется.

Таблица 6.3. *Свойства для задания шрифта*

Свойство	Описание	Стандартные значения	Наследуется?
font-family	Список названий шрифтов. Браузер просматривает список до тех пор, пока не найдет шрифт, который установлен на компьютере вашего посетителя. Если он не находит поддерживаемого шрифта, применяется стандартный шрифт, который всегда используется браузером	Название шрифта (например, Verdana, Times или Arial) или общее название семейства шрифтов: serif, sans-serif, monospace	Да

Таблица 6.3 *(окончание)*

Свойство	Описание	Стандартные значения	Наследуется?
`font-size`	Задает размер шрифта	Конкретный размер или одно из следующих значений: `xx-small`, `xsmall`, `small`, `medium`, `large`, `x-large`, `xx-large`, `smaller`, `larger`	Да
`font-weight`	Задает насыщенность шрифта (насколько жирным он отображается)	`normal`, `bold`, `bolder`, `lighter`	Да
`font-style`	Позволяет задать курсивное начертание	`normal`, `italic`	Да
`font-variant`	Позволяет применить малые прописные буквы (капитель), превращая строчные буквы в малые прописные (КАК ЭТИ)	`normal`, `small-caps`	Да
`text-decoration`	Задает несколько разных изменений текста, таких как подчеркивание или зачеркивание. Строго говоря, они не являются частью шрифта (их вставляет браузер)	`none`, `underline`, `overline`, `line-through`	Да
`text-transform`	Преобразует текст так, что он весь состоит только из прописных или только из строчных букв	`none`, `uppercase`, `lowercase`	Да

Несмотря на то, что большинство CSS-свойств, касающихся шрифтов, понятны и просты, со свойством `font-family` связана неприятная неожиданность — оно не всегда срабатывает. Неизбежная проблема, с которой вы столкнетесь, — нет двух компьютеров, на которых установлен один и тот же набор шрифтов, поэтому шрифты, используемые вами для оформления Web-страницы, не обязательно будут именно теми шрифтами, которые есть на компьютерах ваших посетителей. В результате любые заданные вами параметры шрифтов, — не что иное, как всего лишь рекомендации. Если на компьютере нет нужного вам шрифта, браузер перейдет на стандартный шрифт, который он применяет при отображении сайтов, не содержащих команд форматирования шрифтов.

Имея в виду это предупреждение, вы, возможно, решите, что вообще не стоит беспокоиться о настройке шрифтов. Не торопитесь, есть и хорошие новости. Вместо запроса шрифта и слепой надежды на его доступность в браузере, можно создать список предпочтительных шрифтов (font preferences). В этом случае браузер попытается найти соответствие первому шрифту в вашем списке и, если потерпит неудачу, перейдет ко второму шрифту и т. д. В конце списка следует указать один из нескольких стандартных шрифтов, которые поддерживаются почти всеми компьютерами. В следующем разделе вы увидите этот прием в действии.

Если непременно нужно применить необычный шрифт на Web-странице, у вас остается два варианта. Для небольших порций текста (например, заголовков или кнопок) можно использовать изображение, как описано в *разд. "Графический текст" главы 7*. Для больших фрагментов текста можно применить встраивание или внедрение шрифтов (описанное далее в *разд. "Встроенные шрифты"*), но имейте в виду, что это средство следующего поколения и более старые браузеры могут его не поддерживать.

Задание шрифта

Для выбора шрифта применяется атрибут `font-family`. Далее приведен пример, в котором меняется шрифт всей Web-страницы.

```
body {
  font-family: Arial;
}
```

Arial — это шрифт *без засечек* (семейство sans-serif), который можно найти на всех современных компьютерах, включая работающие под управлением ОС Windows, Mac OS, Unix и Linux. (Разницу между шрифтами с засечками и без см. на рис. 6.9.)

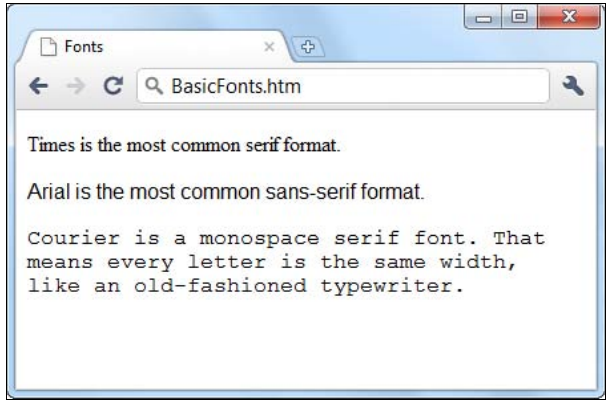


Рис. 6.9. Шрифты с засечками (семейство serif) легче читать в печатных документах. Например, в этой книге в качестве основного шрифта применяется шрифт семейства serif. Если посмотреть повнимательнее на букву "S" в основном тексте, вы увидите крошечные завитки (засечки) в правом верхнем и левом нижнем углах. С другой стороны, у шрифтов семейства sans-serif строгая плавная форма. Они заставляют страницы выглядеть менее книжно и формально, а более современно и свежо

Для безопасности при создании списка шрифтов всегда вставляйте в конец название типового семейства шрифтов. Все компьютеры поддерживают стандартные шрифты с именами семейств `serif`, `sans-serif` и `monospace`.

Далее приведено измененное правило.

```
body {
  font-family: Arial, sans-serif;
}
```

В этот момент вам захотелось сделать это правило чуть изобретательней, добавив поддержку менее распространенного шрифта семейства sans-serif. Далее приведен пример.

```
body {
  font-family: Eras, Arial, sans-serif;
}
```

Если шрифт Eras похож на шрифт Arial, то такой прием не приведет к проблемам. Но если шрифты сильно отличаются, это плохая идея.

Во-первых, применяя нестандартный шрифт, вы создаете страницу, внешний вид которой может значительно изменяться в зависимости от шрифтов, установленных на компьютере посетителя. Если страницы изменчивы, гораздо труднее с помощью тонкой настройки добиться их совершенства, поскольку вы точно не знаете, как они будут отображаться на разных компьютерах. Разные шрифты занимают разный объем пространства, и если текст растягивается или сжимается, верстка других элементов (например, изображений) тоже меняется. Кроме того, так ли уж приятно долгое время читать шрифты KidzzFunScript или SnoopDawg?

Вы столкнетесь и с более коварной проблемой, если на компьютере посетителя есть шрифт с тем же именем, но выглядящий совершенно иначе. Более того, браузеры могут обратиться к интерактивной базе шрифтов, чтобы попытаться найти шрифт похожий на тот, который уже установлен на компьютере вашего гостя. Такой подход очень быстро может все испортить и, что хуже всего, любая из этих проблем может привести к выводу нечитаемого текста.

> ***Совет***
>
> Большинство редакторов Web-страниц не предупреждают об опасности при использовании нестандартного шрифта, поэтому будьте настороже. Если ваш шрифт не входит в небольшой набор широко распространенных Web-шрифтов (об этом через несколько минут), его применять не следует.

Поиск подходящего шрифта

Если хотите, чтобы ваша Web-страница отображалась корректно, используйте стандартный, общедоступный шрифт. Какие же шрифты относятся к стандартным? К сожалению, Web-специалисты не достигли полного консенсуса.

Любой из перечисленных далее шрифтов поможет стать истинным консерватором и не даст сбиться с пути:

- Times
- Arial
- Helvetica
- Courier

Конечно, все они безумно скучны. Если хотите рискнуть, можно применить один из следующих шрифтов, установленных почти на всех компьютерах с Windows и

компьютерах Mac (но необязательно в других операционных системах, например Unix):

- ❒ Verdana
- ❒ Georgia
- ❒ Tahoma
- ❒ Comic Sans MS
- ❒ Arial Black
- ❒ Impact

Если в имени шрифта есть пробелы или специальные символы, лучше заключить его в кавычки или апострофы, т. е. следует писать:

```
body {
    font-family: "Comic Sans MS";
}
```

а не

```
body {
    font-family: Comic Sans MS;
}
```

Большинство браузеров не придают этому значения, но такая форма записи поможет избежать потенциальных проблем.

Сравнить названные шрифты можно на рис. 6.10.

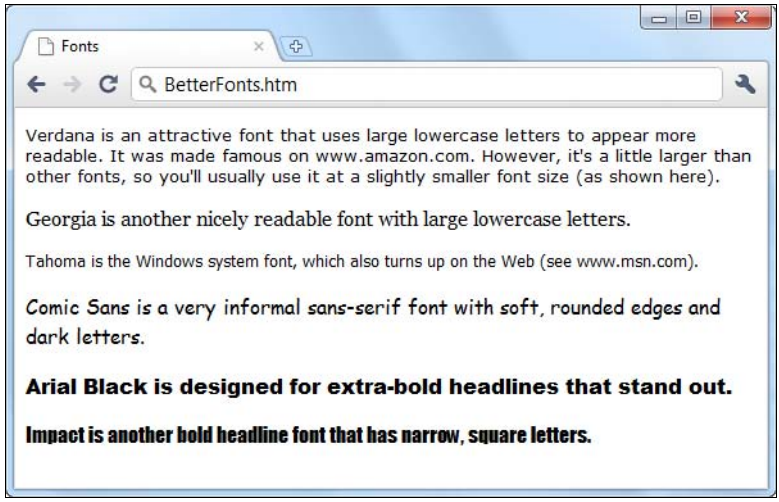


Рис. 6.10. Встречали вы эти шрифты на просторах Web?

Шрифты Verdana, Georgia и Tahoma могут придать вашим Web-страницам более современный облик. Но символы в шрифтах Verdana и Tahoma немного увеличены, поэтому обычно приходится уменьшать их на один уровень (методика описана в следующем разделе).

Интересное обсуждение шрифтов, платформ, их надежно поддерживающих, всех за и против для каждого семейства шрифтов (например, некоторые шрифты хорошо смотрятся на экране, но отвратительны в напечатанном виде) см. на страницах **http://tinyurl.com/cr9oyx** и **http://tinyurl.com/325f9qs**.

Размеры шрифтов

После того как разобрались в трудной проблеме выбора шрифта, возможно, вы захотите изменить его размер. Важно выбрать размер, обеспечивающий удобочитаемость и привлекательный внешний вид. Не поддавайтесь желанию уменьшить или увеличить текст в соответствии с вашими личными предпочтениями. Вместо этого стремитесь, чтобы ваш текст соответствовал стандартному размеру, который вы видите на других популярных Web-сайтах.

Вопреки ожиданиям у вас нет полного контроля над размерами шрифтов на Web-страницах. Большинство посетителей сайта пользуются браузерами, позволяющими уменьшать и увеличивать размеры шрифтов, добиваясь увеличения объема текста на экране или, чаще всего, облегчения чтения текста на мониторе с высоким разрешением. В Internet Explorer и Firefox вы найдете эти параметры в меню **View | Text Size** (Вид | Размер шрифта).

Параметры настройки размера шрифта в браузерах не полностью переопределяют размер, который вы задаете на вашей Web-странице. Вместо этого они немного изменяют шрифт, увеличивая или уменьшая его. Например, если на вашей Web-странице применяется шрифт большого размера (что соответствует примерно 15 пунктам в текстовом процессоре) и посетитель, использующий Internet Explorer, выбирает последовательность команд **View | Text Size | Larger** (Вид | Размер шрифта | Крупный), размер текста вырастет примерно на 20%, до 18 пунктов.

То, что у посетителей есть такой вариант управления размером текста, — еще одна причина, по которой не следует на ваших страницах применять особенно мелкий или крупный шрифт. Если соединить их с настройками по умолчанию в браузерах, крупноватый текст может стать гигантским, а слегка мелковатый превратится в нечитаемый. Лучшая защита от таких проблем — тестирование ваших страниц в разных браузерах и с разными настройками по умолчанию для размера текста.

Как вы узнаете из следующих разделов, задать размер шрифта можно несколькими способами.

Задание размера ключевым словом

Простейший способ задания размера вашего текста — использование одного из обозначений размера, перечисленных в табл. 6.2. Например, для создания действительно крупного заголовка и смехотворно мелкого текста можно применить два следующих правила.

```
body {
  font-size: xx-small;
}
h1 {
  font-size: xx-large;
}
```

Эти ключевые слова, определяющие размер, часто называют *абсолютными размерами*, т. к. они формируют текст точного размера. "Какого именно размера?" —

спросите вы. Вот тут начинаются сложности. Детали установки размера не увековечены в камне; разные браузеры вольны интерпретировать их по-разному. Основное практическое правило — размер шрифта `medium` (средний) соответствует в браузере тексту стандартного размера (12 пунктов). Именно его применяет браузер к Web-сайту, не содержащему настроек размера текста. При каждом подъеме на один уровень к размеру добавляется около 20%. (Любители вычислений могут убедиться, что при каждом переходе на уровень вниз теряется 17%.)

В большинстве браузеров стандартный размер шрифта — 12 пунктов (хотя текст этого размера обычно выглядит мельче на компьютерах Mac по сравнению с компьютерами PC). Это означает, что текст размера `large` (крупный) примерно равен 15 пунктам, `x-large` — 18 пунктам и `xx-large` — 27 пунктам.

На рис. 6.11 показаны основные размеры, из которых вы можете выбирать.



Рис. 6.11. Семь стандартных размеров текста, от `xx-large` до `xx-small`. Вы можете задать размер шрифта, указав также значение в пикселах (см. далее *разд. "Размер в пикселах"*)

Примечание

Перед тем как использовать ключевые слова, определяющие размер шрифта, убедитесь, что на вашей Web-странице есть определение типа документа (doctype). Если его нет, Internet Explorer будет отображать страницу в наводящем ужас режиме обратной совместимости, увеличивающем текст на один уровень по сравнению со стандартным размером. В результате в Internet Explorer ваша страница будет выглядеть иначе, чем в других браузерах, например в Firefox.

Размер в процентах

Еще один вариант задания значения в свойстве `font-size` — использование вместо ключевых слов процентов. Например, если хотите, чтобы наверняка выводился текст обычного размера, примените следующее правило.

```
body {
  font-size: 100%;
}
```

Если же хотите сделать текст мельче, примените такое правило.

```
body {
  font-family: Verdana,Arial,sans-serif;
  font-size: 83%;
}
```

Так же легко увеличить текст.

```
h1 {
  font-size: 120%;
}
```

Но помните, что 100% всегда относится к стандартному размеру обычного текста абзаца, а не к стандартному размеру текста в элементе, стиль которого вы создаете, например, такого как заголовок `<h1>`. Таким образом, если вы формируете заголовок с текстом размера 120%, ваш заголовок станет немного крупнее обычного текста абзаца, который значительно мельче нормального размера заголовка `<h1>`.

Применение размеров в процентах — самый безопасный и надежный способ задания размера текста. Он не только обеспечивает согласованные результаты во всех браузерах, но и действует вместе с описанными ранее настройками размеров по умолчанию в браузерах.

Относительные размеры

Еще один вариант задания размера шрифта — применение одного из двух значений относительного размера: `larger` или `smaller`. В этом случае берется текущий размер текста в элементе и немного увеличивается или уменьшается.

Легче всего понять, как это работает, на примере следующей таблицы стилей, содержащей два правила.

```
body {
  font-size: xx-small;
}

b {
  font-size: larger;
}
```

Первое правило применяет абсолютный размер `xx-small` ко всему тексту на странице. Если ваша страница содержит элемент `<b>`, текст внутри этого элемента `<b>`

наследует размер xx-small (подробности см. в *разд. "Наследование"*), а затем второе правило увеличивает размер текста на один уровень до x-small.

Теперь посмотрим, что произойдет, если отредактировать приведенный стиль текста страницы и задать более крупный шрифт.

```
body {
  font-size: x-small;
}

b {
  font-size: larger;
}
```

Теперь весь текст, отображаемый жирным шрифтом, станет на один уровень крупнее x-small, т. е. small.

Относительные размеры применять немного сложнее, чем абсолютные. Вероятнее всего, вы воспользуетесь ими при наличии таблицы стилей с большим набором разных размеров текста. Например, для текста с жирным шрифтом можно использовать относительный размер, если хотите, чтобы текст, набранный жирным шрифтом, всегда был немного крупнее окружающего текста. Если бы вы вместо относительного применили абсолютный размер, текст с жирным шрифтом отображался бы крупнее, чем мелкий текст абзаца, но не отличался бы от текста заголовка, набранного крупным шрифтом.

> ***Совет***
>
> Применяя абсолютные или относительные размеры, вы создаете более гибкие страницы. Если посетитель увеличит размер текста с помощью стандартных настроек браузера, браузер пропорционально изменит размеры всех остальных шрифтов на странице.

Размер в пикселах

В большинстве случаев следует полагаться на абсолютные и относительные размеры текста. Но если нужен более тонкий контроль, можно задать точный размер текста, указав его в пикселах. Размеры в пикселах могут меняться в широком диапазоне, при стандартном размере текста в элементе <body>, равном примерно 12 и 14 пикселам. Для задания размера в пикселах укажите число и сразу следом за ним буквы px, как показано далее.

```
body {
  font-size: 11px;
}
h1 {
  font-size: 24px;
}
```

> ***Примечание***
>
> Не вставляйте пробел между числом и буквами px. Если сделать это, правило будет действовать в Internet Explorer, а другие браузеры приведет в сильное замешательство.

Как всегда нужно проверять, улучшать и снова проверять выбор шрифта, чтобы добиться подходящего размера. Некоторые шрифты выглядят крупнее остальных, поэтому для них следует задавать меньший размер. Есть шрифты, отлично выглядящие при больших размерах и превращающиеся в трудноразличимые при уменьшении размера.

Internet Explorer интерпретирует заданный в пикселах шрифт иначе, чем текст, заданный другим способом. Если вы используете его команды **View | Text Size** (Вид | Размер шрифта), Internet Explorer исправляет весь текст *за исключением* заданного в пикселах. С другой стороны, средство масштабирования IE (**View | Zoom** (Вид | Масштаб)) воздействует на все компоненты страницы, включая изображения и текст. У других браузеров нет особой трактовки текста, заданного в пикселах. В итоге размер текста в пикселах приведет к несогласованным результатам.

Встроенные шрифты

Большую часть времени существования Всемирной паутины Web-разработчики вынуждены были жить с ограниченными возможностями свойства `font-family`. Они научились извлекать максимум из очень маленького набора стандартных шрифтов. Но в стандарт CSS3 вводится новое средство `@font-face`, предоставляющее браузерам способ загрузки шрифтов Web-страницы на лету (помещая их во временный кэш страниц и изображений) и использования шрифтов только для данного Web-сайта или страницы. В результате Web-страницы могут использовать практически любую компьютерную гарнитуру.

> ***Примечание***
>
> Строго говоря, свойство `@font-face` не новое. Оно было частью стандарта CSS 2, но исчезло из версии CSS 2.1, когда создатели браузеров не смогли договориться. Теперь в CSS3 вновь появился стимул сделать свойство `@font-face` универсальным стандартом.

На первый взгляд, правило `@font-face` кажется отличным решением проблем, связанных со шрифтами и знакомых всем Web-разработчикам. К сожалению, вы столкнетесь с тем, что существование на переднем крае Web-дизайна не бывает безболезненным. Первая проблема — получение поддержки формата вашего шрифта.

Web-форматы для шрифтов

Несмотря на то что современные браузеры поддерживают `@font-face`, не все они поддерживают один и тот же *тип* файлов шрифтов. Internet Explorer, поддерживавший правило `@font-face` многие годы, работает (все версии, предшествующие IE 9) только с файлами шрифтов формата EOT (Embedded OpenType). У формата EOT есть ряд достоинств — он применяет сжатие для уменьшения размера файла шрифта и разрешает строгое лицензирование Web-сайтов, так что шрифт не может быть украден с одного Web-сайта и использоваться на другом. Но формат EOT так

и не завоевал популярности, и другие браузеры его не применяют. Остальные браузеры (до недавнего времени) были привязаны к более привычным стандартам шрифтов, применяемым в приложениях настольных компьютеров, к ним относятся форматы TTF (TrueType) и OTF (OpenType PostScript). Но история будет неполной без двух дополнительных акронимов. В табл. 6.4 для сравнения перечислены все форматы шрифтов.

Таблица 6.4. *Форматы встроенных шрифтов*

Формат	Описание	Применение
TTF (TrueType), OTF (OpenType PostScript)	Ваш шрифт возможно сначала появится в одном из этих обычных настольных форматов	В Firefox (до версии 3.6), Chrome (до версии 6), Safari и Opera
EOT (Embedded Open Type)	Особый формат корпорации Microsoft, не завоевавший популярность в других браузерах, кроме Internet Explorer	В Internet Explorer (до IE 9)
SVG (Scalable Vector Graphics)	Универсальный графический формат, который можно применять для шрифтов. Он дает хорошие, но не блестящие результаты (отображается медленнее и формирует текст более низкого качества)	В Safari Mobile (в iPhone и iPad до iOS 4.2[1]) и мобильных устройствах, использующих операционную систему Android
WOFF (Web Open Font Format)	Единый формат возможно в будущем. В настоящее время поддерживается только самыми современными браузерами	Только в Internet Explorer 9, Firefox 3.6, Chrome 6

[1] Операционная система iOS 4.2 была выпущена в конце 2010 г.

Примечание

Дополнительную информацию, современный взгляд на поддержку шрифтов в разных браузерах можно найти на Web-сайте **http://tinyurl.com/cluxzz**.

И последнее, если вы хотите применять правило `@font-face` прямо сейчас, придется одновременно поддерживать несколько стандартов. Как минимум нужно предоставить ваш шрифт в форматах TTF или OTF (подойдет любой), формате EOT и формате SVG. Неплохо (но не обязательно) также предоставить шрифт в формате WOFF, который в будущем, вероятно, станет более популярным и лучше поддерживаемым. Но даже если вы выполните все требования и предоставите все необходимые форматы шрифтов, затруднений не избежать: например, пользователи сообщают, что в некоторых браузерах возникают проблемы при печати определенных шрифтов.

Применение комплектов шрифтов

Сейчас вас, наверное, интересует, где взять файлы нужных вам шрифтов. Самый простой вариант — загрузить готовый *комплект шрифтов* из Web. В этом случае

вы получаете файлы всех необходимых шрифтов. К сожалению, ваш выбор ограничен тем, что вам удастся найти в Интернете.

Одно из лучших мест для поиска комплектов шрифтов — Web-сайт Font Squirrel; его тщательно подобранные шрифты можно увидеть на странице **www.fontsquirrel.com/fontface** (рис. 6.12).

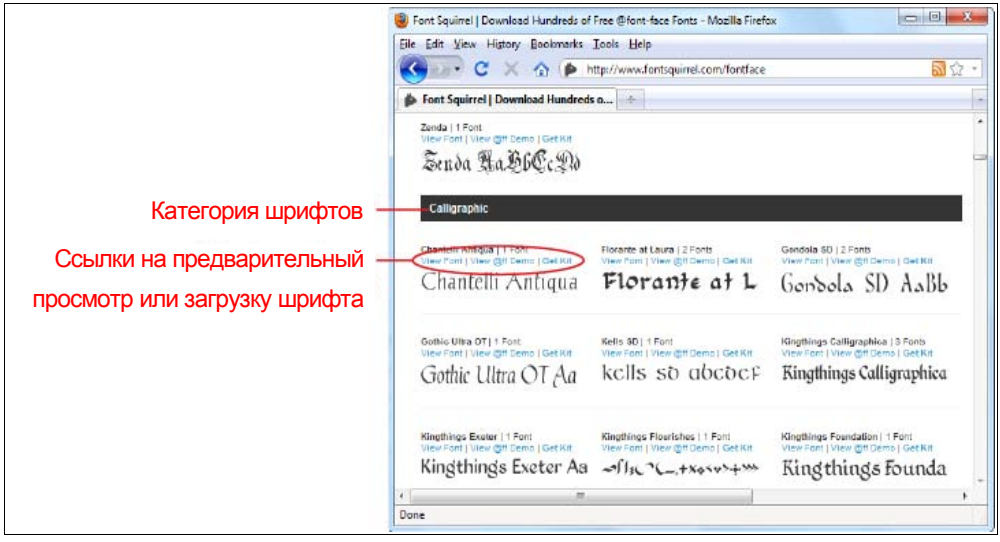


Рис. 6.12. Web-сайт Font Squirrel предлагает несколько сотен высококачественных шрифтов, разделенных на категории по типу (например, **Calligraphic**, **Grunge** и **Retro**). Самое замечательное то, что все шрифты можно применять бесплатно и для создания документов на вашем персональном компьютере, и для разработки Web-страниц во Всемирной паутине. Когда подберете шрифт, щелкните кнопкой мыши ссылку **View Font** (Просмотреть шрифт), чтобы получше разглядеть каждую букву, ссылку **View @ff Demo** (Просмотреть образец), чтобы увидеть шрифт на тестовой Web-странице, полученной с помощью правила `@font-face`, и ссылку **Get Kit** (Получить комплект), чтобы загрузить шрифт на ваш компьютер

В результате загрузки комплекта шрифта вы получите сжатый ZIP-файл, содержащий в свою очередь несколько файлов. Например, загрузив шрифт Chantelli Antiqua, показанный на рис. 6.12, вы получите следующие файлы:

- Bernd Montag License.txt
- Chantelli_Antiqua-webfont.eot
- Chantelli_Antiqua-webfont.svg
- Chantelli_Antiqua-webfont.ttf
- Chantelli_Antiqua-webfont.woff
- demo.html
- stylesheet.css

Текстовый файл (Bernd Montag License.txt) содержит лицензионную информацию, как правило, сообщающую о том, что вы можете использовать шрифт бесплатно, но не имеете права продавать его. Файлы Chantelli_Antiqua-webfont предоставляют файлы четырех разных форматов. (В зависимости от выбранного вами шрифта вы

можете получить дополнительные файлы для разных вариаций шрифта, например, в жирном, курсивном начертаниях и в стиле extra-dark.) И наконец, файл stylesheet.css содержит правило таблицы стилей, которое необходимо для использования шрифта на вашей Web-странице, а файл demo.html отображает шрифт на тестовой Web-странице.

Для применения шрифта Chantelli_Antiqua необходимо скопировать все файлы Chantelli_Antiqua-webfont в папку, где хранится ваша Web-страница. Далее шрифт следует зарегистрировать, для того чтобы он был доступен для применения в вашей таблице стилей. Для этого в начале таблицы стилей применяется сложное правило `@fontface`, выглядящее следующим образом (строки пронумерованы, чтобы было легче на них ссылаться).

```
1  @font-face {
2    font-family: 'ChantelliAntiquaRegular';
3    src: url('Chantelli_Antiqua-webfont.eot');
4    src: local('Chantelli Antiqua'),
5      url('Chantelli_Antiqua-webfont.woff') format('woff'),
6      url('Chantelli_Antiqua-webfont.ttf') format('truetype'),
7      url('Chantelli_Antiqua-webfont.svg') format('svg');
8  }
```

Для того чтобы понять, что происходит в этом правиле, рассмотрим его строка за строкой.

- Строка 1: `@font-face` — это средство для официальной регистрации шрифта, поэтому его можно помещать в любом месте таблицы стилей.
- Строка 2: шрифту можно присвоить любое имя. Именно им вы будете пользоваться, задавая шрифт.
- Строка 3: первым задаваемым форматом должен быть файл формата EOT. Делается это потому, что в противном случае Internet Explorer приходит в замешательство и игнорирует все остальные форматы. Функция `url()` — метод таблицы стилей, заставляющий браузер загружать еще один файл, находящийся по заданному вами адресу. Если поместить шрифт в ту же папку, что и ваша Web-страница, здесь можно указать только имя файла.
- Строка 4: следующий шаг — применение функции `local()`. Она сообщает браузеру имя шрифта и, если оказывается, что этот шрифт уже установлен на компьютере посетителя, браузер применяет его. Однако в редких случаях может возникнуть проблема (например, в Mac OS X либо на экране появится диалоговое окно системы безопасности, это зависит от того, куда ваш посетитель установил шрифт, либо на компьютере окажется другой шрифт с тем же именем). Поэтому Web-разработчики иногда применяют явно фиктивное имя, чтобы браузер наверняка не нашел локального шрифта. Как правило, используют бессмысленный символ, например, `local('☺')`.
- Строки 5—7: последний шаг — сообщить браузеру об остальных файлах шрифта, которые он может использовать. Если у вас есть файл шрифта формата WOFF, предложите его первым, т. к. он обеспечит наивысшее качество. Далее

сообщите браузеру о файлах формата TTF или OTF и в заключение о файле формата SVG.

> ***Совет***
>
> Конечно, нет необходимости вводить правило `@font-face` вручную (и безусловно, не нужно разбираться во всех технических тонкостях, описанных выше). Вы можете просто скопировать правило из файла stylesheet.css, входящего в комплект Web-шрифта.

После того как с помощью правила `@font-face` встраиваемый шрифт зарегистрирован, его можно применять в любой таблице стилей. Просто воспользуйтесь знакомым свойством `font-family` и укажите имя семейства шрифтов, которое вы задали с помощью правила `@font-face` (строка 2). Далее приведен пример, в котором опущены подробности полного описания правила `@font-face`.

```
@font-face {
  font-family: 'ChantelliAntiquaRegular';
  ...
}

body {
    font-family: 'ChantelliAntiquaRegular';
}
```



Рис. 6.13. На этой Web-странице применяется встроенный шрифт, придающий тексту законченный вид, что отличает его от других сайтов

Это правило задает шрифт для всей Web-страницы, хотя вы можете ограничить его применение несколькими определенными элементами или использовать классы (см. ранее *разд. "Селекторы классов" данной главы*). На рис. 6.13 показана простая страница, использующая шрифт Chantelli Antiqua.

ПРИМЕЧАНИЕ

Зарегистрировать шрифт с помощью `@font-face` следует до его применения в стилевом правиле. Если поменять местами эти два шага, шрифт не будет применяться корректно.

СОВЕТ

Еще одно отличное место для поиска бесплатных Web-шрифтов — Google Font Directory на странице **http://code.google.com/webfonts**. Здесь не такой большой выбор шрифтов, но ими легко пользоваться. Вместо загрузки файлов и копирования их на ваш Web-сайт, вы просто вставляете ссылку на таблицу стилей Google, которая с помощью правила `@font-face` регистрирует для вас выбранный шрифт. Затем его можно применять в вашей таблице стилей, не беспокоясь об особенностях разных форматов шрифта.

Применение ваших собственных шрифтов

Любители шрифтов, как известно, очень привередливы при выборе гарнитуры. Если у них в голове есть некое представление о шрифте, который должен выводиться на их Web-страницах, даже самой большой библиотеки бесплатных шрифтов будет недостаточно. К счастью, есть невероятно легкий способ адаптации шрифта для Web. Применяя подходящее средство, можно взять уже имеющийся файл шрифта формата TTF или OTF и создать другие нужные форматы (EOT, SVG и WOFF).

Но прежде чем ступить на этот путь, важно узнать об одной щекотливой детали. Обычно шрифты не бесплатны. Это значит, что не вполне законно взять шрифт, имеющийся на вашем компьютере, и применить его на вашем Web-сайте, если у вас нет явного разрешения от создателя шрифта. К примеру, корпорации Microsoft и Apple платят за включение ряда шрифтов в их операционные системы и приложения, поэтому вы можете использовать их, скажем, для создания информационных рассылок в текстовом процессоре. Но лицензия не разрешает помещать эти шрифты на Web-сервер и использовать их на ваших Web-страницах.

ПРИМЕЧАНИЕ

Стоимость лицензии на шрифт может быть очень разной. Первоклассные производители шрифтов в зависимости от величины потока информации, получаемой вашим Web-сайтом, могут требовать сотни долларов за предоставление лицензии. Другие создатели шрифтов могут разрешить использовать свои шрифты за символическую цену или бесплатно, требуя взамен соответствия некоторым критериям (например, вы вставляете на страницу набранное мелким шрифтом примечание об используемом шрифте или создаете некоммерческий Web-сайт, не предназначенный для получения вагонов денег). Если у вас есть любимый шрифт, единственный способ узнать, нужно ли за него платить, — связаться с компанией или человеком, создавшими его. У таких контактов есть и положительная сторона. Высокопрофессиональные создатели шрифтов часто предоставляют Web-оптимизированные версии своих творений, которые в Интернете смотрятся гораздо лучше стандартных шрифтов.

После того как вы узнали, что конкретный шрифт использовать разрешено, можно преобразовать его с помощью удобного средства, размещенного на Web-сайте Font Squirrel (это сайт, предлагающий отличные бесплатные комплекты Web-шрифтов). Для этого перейдите на страницу **www.fontsquirrel.com/fontface/generator**. На рис. 6.14 показан трехшаговый процесс, которому необходимо следовать.

Web-комплект, генерируемый Font Squirrel, точно такой же, как бесплатные комплекты шрифтов, описанные в предыдущем разделе. В него даже включена наряду с тестовой Web-страницей таблица стилей, содержащая нужное вам правило `@font-face`.

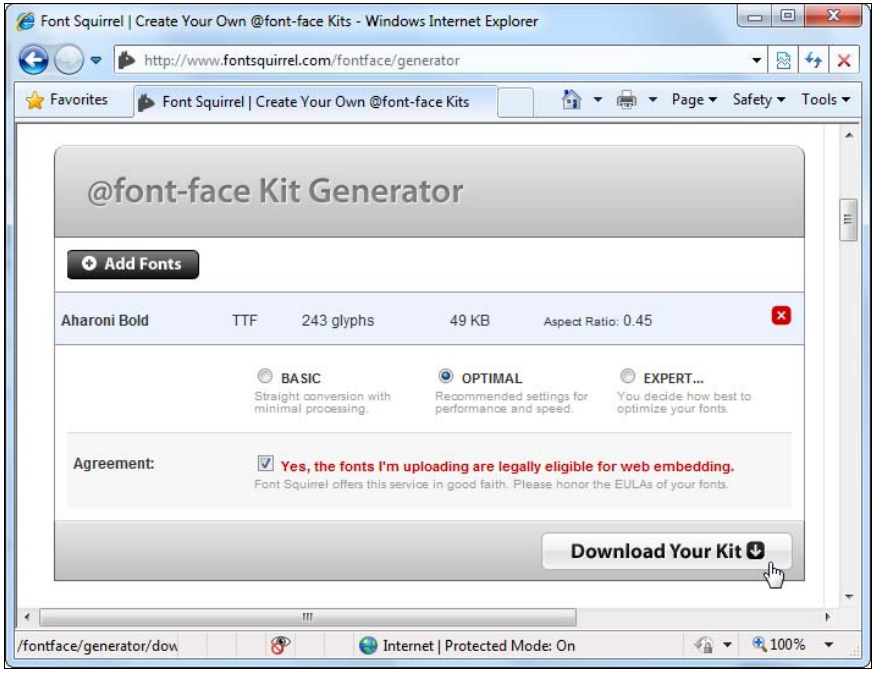


Рис. 6.14. Сначала щелкните мышью кнопку **Add Fonts** (Добавить шрифты) для выгрузки файла шрифта с вашего компьютера. Затем установите флажок **Yes the fonts I'm uploading are legally eligible for web embedding** (Шрифты, которые я выгружаю, могут встраиваться на законном основании) (при условии, что вы просмотрели их лицензионные требования). В заключение щелкните мышью кнопку **Download Your Kit** (Загрузить ваш комплект)

Границы

Последняя группа свойств таблиц стилей, о которой вы узнаете в этой главе, позволяет вставлять рамки или границы на вашу Web-страницу (рис. 6.15). Границы — прекрасный способ отделить маленькие фрагменты или целые блоки контента.

В табл. 6.5 перечислены три основных свойства форматирования границ.

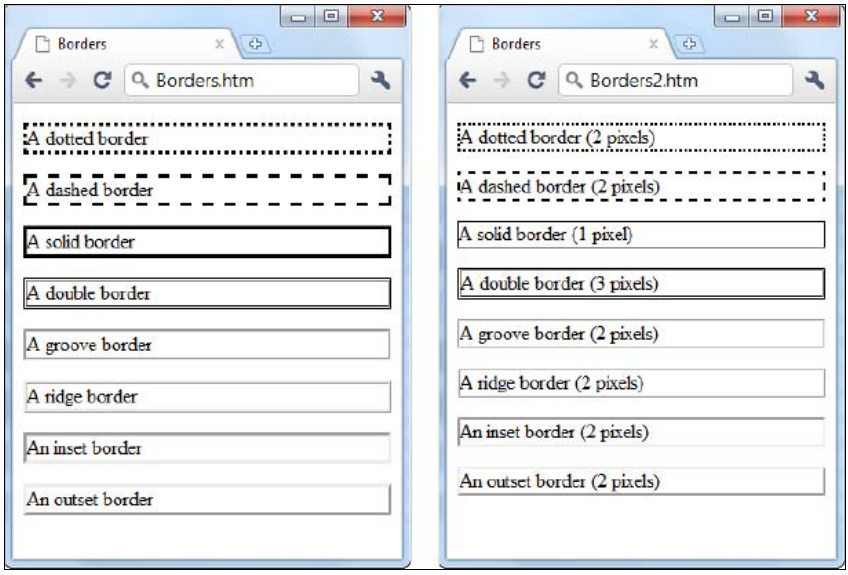


Рис. 6.15. *Слева*: базовые стили границ выглядят немного старомодно на современных приглаженных Web-сайтах.
Справа: сделайте эти границы на один-два пиксела у́же, и они будут смотреться гораздо лучше

Таблица 6.5. *Свойства границ*

Свойство	Описание	Стандартные значения	Наследуется?
`border-width`	Задает толщину линии границы. Обычно хочется ее урезать	Ширина в пикселах	Нет
`border-style`	У браузеров есть восемь встроенных стилей границ. Стиль границы определяет ее внешний вид	`none`, `dotted`, `dashed`, `solid`, `double`, `groove`, `ridge`, `inset`, `outset`	Нет
`border-color`	Цвет границы	Имя цвета, шестнадцатеричный код цвета или RGB-значение (см. ранее *разд. "Задание цвета" данной главы)*	Нет

Основные границы

Первое, что вы должны выбрать, создавая границу, — ее стиль. Можно использовать штриховую или точечную линию, желобок (или канавку) или гребень или просто обычную тонкую волосяную линию (которая часто выглядит привлекательнее остальных). Далее приведено правило, создающее штриховую границу.

```
p {
  border-style: dashed;
}
```

Толщина стандартной границы почти всегда слишком велика. Для того чтобы граница выглядела представительнее, уменьшите ее толщину на 1—2 пиксела в зависимости от стиля границы.

```
p {
  border-style: dashed;
  border-width: 2px;
}
```

Можно также применять такие свойства, как `border-top-style` и `border-left-width`, чтобы задавать разные стили, толщины и цвета для каждой из четырех сторон рамки, окружающей элемент. Одновременное применение большого количества свойств порой может привести к неожиданному эффекту, но, как правило, в такой детализации нет необходимости. В следующем разделе дается несколько советов по оптимизации границ.

Создание улучшенных границ

Границы на рис. 6.15 выглядят неплохо, но они слишком прижаты к тексту в рамке, сформированной из граничных линий, а также к краям страницы.

Для того чтобы сделать границу выразительной, рассмотрим применение свойства `border` и трех остальных свойств.

- Свойство `background-color` (см. *разд. "Цвета" данной главы*) задает фоновый цвет вашего элемента. Применяемое вместе со свойством `border`, оно заставляет ваш элемент выглядеть как перемещаемая панель, почти как боковая колонка в журнале.
- Свойство `margin` (см. табл. 6.2) позволяет задать величину пробельной зоны между рамкой элемента и остальной страницей. Выбирайте поля так, чтобы рамки, окружающие элементы, не теснили остальной контент страницы и не прижимались к боковым сторонам окна браузера.
- Свойство `padding` действует как свойство `margin`, но задает величину пробельной зоны внутри элемента, между его границами и его реальным содержимым. Выбирайте отступ так, чтобы было достаточно свободного пространства между границей элемента и текстом в нем.

На рис. 6.16 показана разница между свойствами `margin` и `padding`.

Далее приведен пример абзаца, отображаемого внутри подкрашенной прямоугольной области.

```
p {
  background-color: #FDF5E6;
  margin: 20px;
  padding: 20px;
  border-style: solid;
  border-width: 1px;
}
```

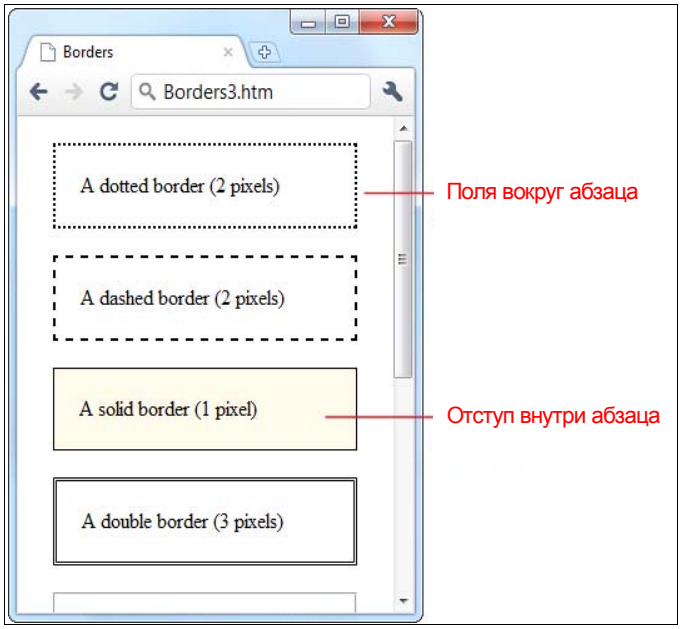


Рис. 6.16. Обычно невозможно понять разницу между полями и отступами, потому что края элемента не видны. Например, элемент <p> отображает абзац в невидимой прямоугольной области, границы которой вы не видите. Если добавить границу, все изменится. Для создания привлекательной границы необходимо увеличивать и поля и отступы. Для дополнительного эффекта добавьте светлый цвет фона (как у элемента с границами в виде сплошных тонких линий)

На рис. 6.16 показано, как свойства `margin`, `padding` и `background-color` превращают обычный абзац в выразительную панель.

Применение границ в таблицах

Как вы узнали в *главе 5*, стандартная таблица — это набор ячеек без границ. Но с помощью таблицы стилей легко можно добавить в таблицу выбранные вами границы. Нужно просто применить к комбинации элементов <tr>, <td>, <th> и <table> свойства, задающие границы.

В следующем примере таблица стилей задает тонкую синюю рамку вокруг каждой ячейки и толстую синюю рамку вокруг всей таблицы.

```
table {
  border-width: 3px;
  border-style: solid;
  border-color: blue;
}
td, th {
  border-width: 1px;
  border-style: solid;
  border-color: blue;
}
```

На рис. 6.17 показан результат.

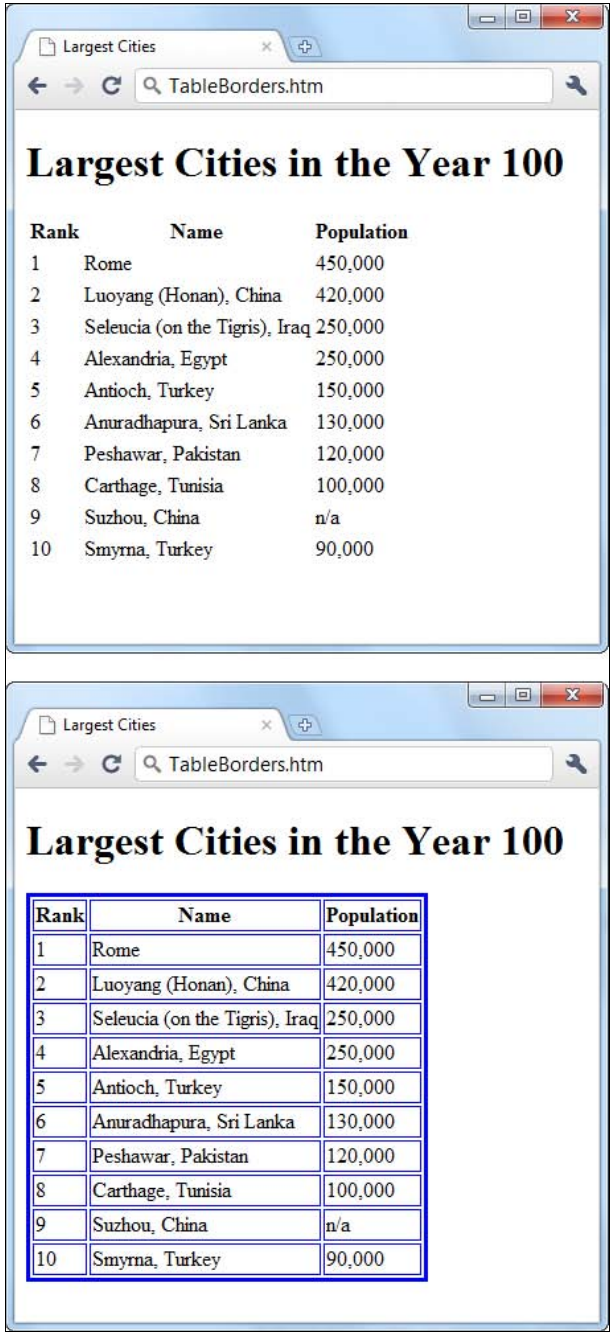


Рис. 6.17. Сравните стандартную HTML-таблицу без границ (вверху) с той же таблицей, снабженной границами, настраиваемыми с помощью таблицы стилей (внизу)

Руководство по созданию таблицы стилей

Вы изучили множество разных вариантов форматирования, которые можно применять на ваших Web-страницах, начиная с пронзительного цвета и заканчивая причудливыми шрифтами. Но даже зная возможности таблиц стилей, вы, наверное, не представляете себе, как создать настоящую таблицу стилей, которая сведет сложное форматирование к простому набору правил, но при этом останется достаточно гибкой, чтобы совершенствоваться вместе с вашим Web-сайтом.

Это естественно. Создание таблиц стилей — искусство, требующее некоторых практических навыков. Приведенное далее практическое руководство поможет "разложить все по полочкам".

Создание хорошо структурированной таблицы стилей

Прежде чем приступать к написанию каких-либо стилевых правил, необходимо хорошенько продумать Web-страницу и ее структуру. Предположим, что ваш Web-сайт содержит страницу с аннотациями книг, показанную на рис. 6.18.

Сначала имеет смысл зафиксировать стандартные детали. Например, вы можете выбрать для всего сайта единственный шрифт, для абзацев — базовые поля и отступы определенного размера и стандартный стиль для заголовков первого уровня. Далее приведена таблица стилей с описанием этих деталей.

```
/* Задание шрифта для всей страницы. */
body {
  font-family: Georgia,serif;
}

/* Задание стандартных полей для абзацев. */
p
{
  margin-top: 2px;
  margin-bottom: 6px;
}

/* Форматирование заголовков с помощью цвета фона. */
h1 {
  background-color: #FDF5E6;
  padding: 20px;
  text-align: center;
}
```

Эта таблица стилей содержит три правила для селекторов типов. Первое форматирует элемент <body>, таким образом задавая шрифт для всей страницы. Второе задает для всех элементов <p> одинаковые поля, а третье изменяет выравнивание и цвет фона заголовков <h1>. Все эти правила достаточно просты и в них используются селекторы типов.

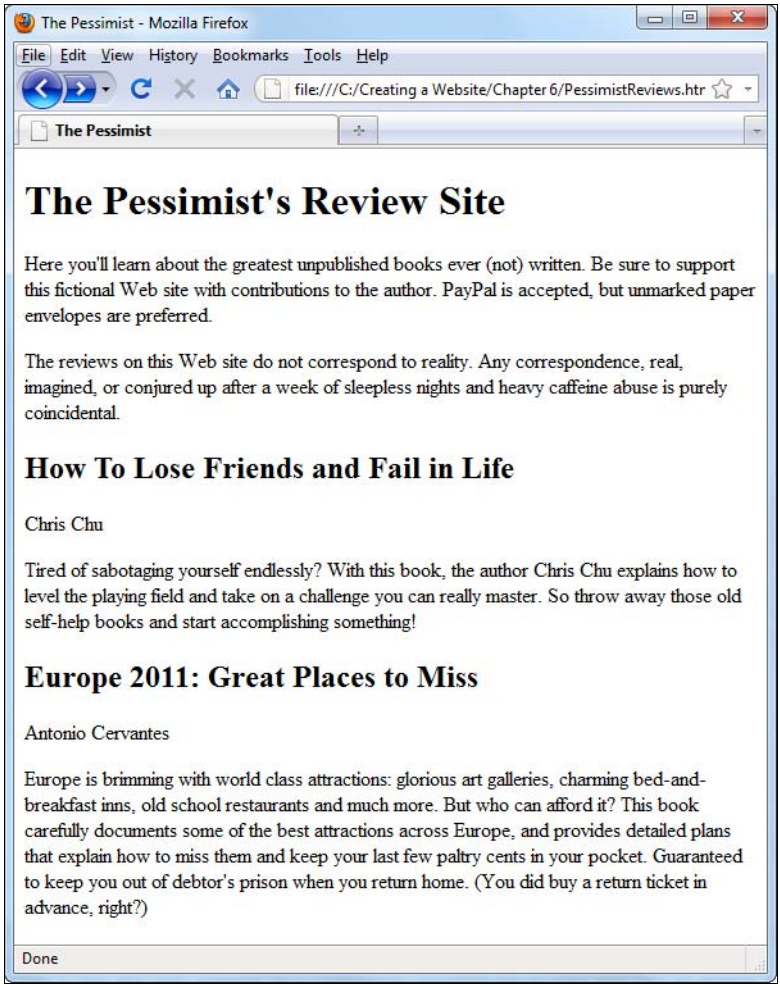


Рис. 6.18. Среднестатистический HTML-документ содержит массу похожих элементов, даже сложная страница часто сводится только к заголовкам и абзацам. На данной странице есть общее введение, за которым следует несколько аннотаций. Автор страницы выделил с помощью элементов <p> общее введение, названия книг и имена авторов и краткие аннотации к книгам, но у всех этих компонентов должно быть разное форматирование, т. к. они представляют разные виды контента. Лучше всего по-разному форматировать разные виды контента (заглавие книги, автора и ее описание)

Примечание

В этом примере вводится новое средство — CSS-комментарии. Они отличаются от HTML-комментариев. CSS-комментарии всегда начинаются с символов `/*` и заканчиваются символами `*/`. Они позволяют описать, что представляет каждый селектор типа. В сложной таблице стилей без комментариев очень легко забыть, что делает каждое правило, особенно если вы пользуетесь селекторами классов.

Покончив с базовыми правилами, самое время перейти к реальной работе по созданию таблицы стилей и начать писать правила для классов. Как вы прочли в *разд. "Селекторы классов" данной главы*, правила для классов придают таблице стилей настоящую гибкость и позволяют применять форматирование к любому

элементу и к любому количеству элементов. Они также помогают формировать более строгое и структурированное представление о ваших страницах. Вместо акцентирования внимания на использованных вами HTML-тегах, они помогают сосредоточиться на разных видах предоставленной информации.

Например, страница с книжными аннотациями делится на несколько составляющих: заголовок аннотации, подзаголовок (указывающий имя автора) и сам текст аннотации. Раз вы мысленно разделили страницу на эти компоненты, можно вставить правила классов, которые их отформатируют.

Например, для форматирования заголовка можно применить следующее правило.

```
/* Отображение заголовков аннотаций синим цветом */
h2.review {
  font-size: 100%;
  color: blue;
  margin-bottom: 0px;
}
```

Это правило применяется только к элементам `<h2>` и только к тем из них, у которых имя класса — `review`. Такой прием имеет смысл, т. к. на Web-сайте наверняка есть другие заголовки уровня 2, не содержащие заголовков аннотаций и поэтому нуждающиеся в другом форматировании.

> ***Совет***
>
> Напоминаю, что имя класса должно представлять собой краткое описание типа контента, который вы хотите отформатировать. В данном примере имя класса — `review`, т. к. вы хотите применить данный стиль к заголовкам аннотаций к книгам. Хорошие имена классов описывают назначение класса, а не его внешний вид. Например, WarningNote — хорошее имя, а BoldRedArialBox — нет. Проблема последнего имени в том, что оно лишится смысла, как только вы измените формат предупреждающих замечаний (например, решите выводить их красным цветом).

Последними в таблице стилей определены два класса, форматирующие элементы абзацев — один для подзаголовка с именем автора, а второй для текста аннотации.

```
/* Сделать подзаголовки мелкими и курсивными. */
p.byline {
  font-size: 65%;
  font-style: italic;
  border-bottom-style: outset;
  border-bottom-width: 1px;
  margin-bottom: 5px;
  margin-top: 0px;
}

/* Сделать текст аннотации немного мельче и выровнять по ширине. */
p.review {
  font-size: 83%;
  text-align: justify;
}
```

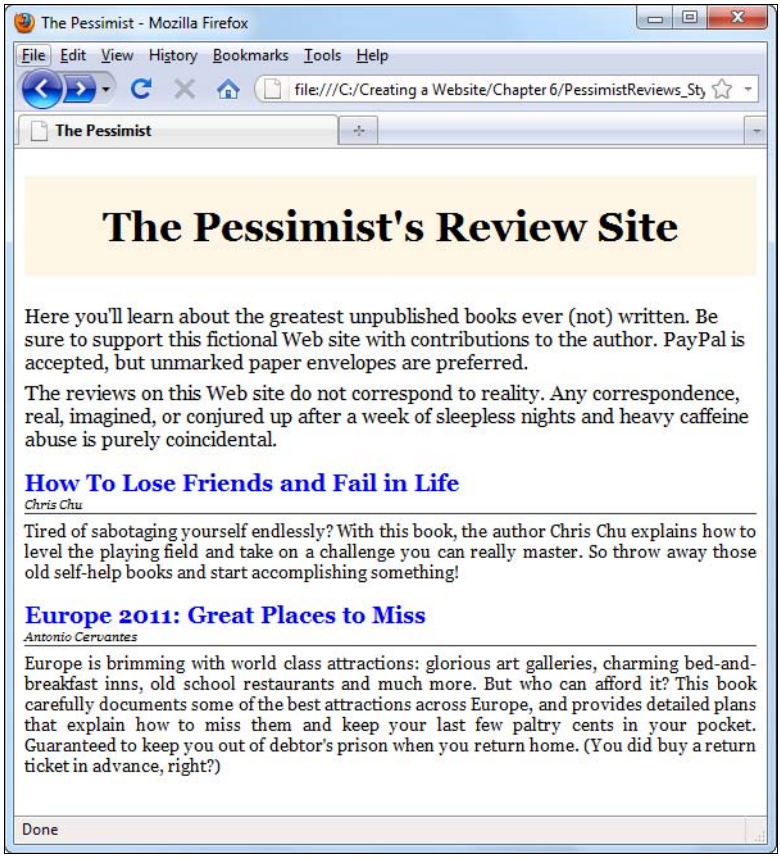


Рис. 6.19. Правила для классов позволяют по-разному форматировать различные части документа, несмотря на то что они используют один и тот же элемент (например, такой, как неизменно популярный элемент <p>)

Теперь нужно просто применить вашу хорошо структурированную таблицу стилей к странице с аннотациями к книгам. Далее приведена HTML-разметка, использующая описанные классы. (Для экономии места большая часть текста пропущена, но важные компоненты структуры сохранены.)

```
<!DOCTYPE html>

<html>

<head>
  <title>The Pessimist</title>
  <link rel="stylesheet" href="PessimistReviews1.css" />
</head>

<body>
  <h1>The Pessimist's Review Site</h1>
  <p>...</p>
  <p>...</p>
```

```
  <h2 class="review">How To Lose Friends and Fail in Life</h2>
  <p class="byline">Chris Chu</p>
  <p class="review">...</p>
  <h2 class="review">Europe 2009: Great Places to Miss</h2>
  <p class="byline">Antonio Cervantes</p>
</body>

</html>
```

На рис. 6.19 показан результат.

Сокращение работы с помощью элемента * *

Довольно утомительно вставлять на вашей Web-странице атрибут `class` в каждый элемент, который хотите отформатировать. Например, в каждой аннотации с четырьмя или пятью абзацами после подзаголовка придется вставлять его в каждый абзац. К счастью, можно существенно сократить работу благодаря остроумному элементу `<div>`.

Возможно, вы еще помните элемент `<div>` из предыдущей главы (см. *разд. "Разделы и оболочки"*). Это блочный элемент, позволяющий группировать любые разделы вашей Web-страницы. С помощью элемента `<div>` можно объединить сколько угодно элементов, включая заголовки, абзацы, списки и т. д.

Благодаря наследованию в таблице стилей (см. *разд. "Наследование" данной главы*) имя класса можно задать в элементе `<div>`, и определенный стиль автоматически унаследуют все вложенные элементы. Представьте, что произойдет, если вы замените правило таблицы стилей:

```
p.review {...}
```

на следующее:

```
div.review {...}
```

Можно будет взять аннотацию из четырех абзацев:

```
  <h2 class="review">...</h2>
  <p class="byline">...</p>
  <p class="review">...</p>
  <p class="review">...</p>
  <p class="review">...</p>
  <p class="review">...</p>
```

и упростить разметку следующим образом.

```
  <div class="review">
    <h2 class="review">...</h2>
    <p class="byline">...</p>
    <p>...</p>
    <p>...</p>
```

```
    <p>...</p>
    <p>...</p>
  </div>
```

В заголовке и подзаголовке все равно нужен атрибут `class`, который задает для них особое форматирование. Но остальные абзацы автоматически наследуют форматирование из элемента `<div>`, в который они вложены. И хотя некоторые свойства стилей (такие как `margin` и `padding`) не поддерживают наследование, большинство свойств наследуется.

На рис. 6.20 показан результат.



Рис. 6.20. В данном примере каждая аннотация в HTML-разметке помещена в элемент `<div>`, для которого заданы цвет фона и некоторые границы, отделяющие аннотации от остальной информации на странице. Такие приемы позволяют сформировать плотные страницы с большим объемом информации

Элемент `<div>` позволяет существенно сэкономить время. Web-специалисты пользуются им постоянно. Но вы можете еще усовершенствовать разметку, применив один контекстный селектор, как описано в следующем разделе.

Сокращение работы с помощью контекстного селектора

Контекстный селектор строже обычного селектора типа. Селектор типа соответствует определенному элементу, а контекстный селектор — элементу внутри другого элемента. Чтобы понять разницу, взгляните на следующий селектор типа.

```
b {
  color: red;
}
```

Этот селектор задает отображение красным цветом всего текста с жирным начертанием. Но что если вы хотите отформатировать только жирный шрифт, появляющийся в маркированном списке? Это можно сделать с помощью следующего контекстного селектора типа, который находит пункты маркированного списка (`<ul>`) и затем ищет в них элементы с жирным шрифтом. Если он находит такие, то отображает их красным цветом.

```
ul b {
  color: red;
}
```

Для создания контекстного селектора типа просто вставьте пробел между двумя элементами.

Контекстные селекторы очень полезны, но размышления о разных вариантах комбинирования элементов могут довести до легкого головокружения. Реальный выигрыш от применения контекстного селектора можно увидеть, если использовать его, чтобы добиться соответствия элементу конкретного типа, находящемуся внутри класса определенного типа. Например, посмотрим, что произойдет, если в следующем правиле таблицы стилей:

```
h2.review {...}
```

заменить селектор таким:

```
div.review h2 {...}
```

Первая часть данного селектора указывает на все элементы `<div>` на вашей странице. Вторая часть ограничивает найденные совпадения элементами `<div>` с именем класса `review` (таких всего один). Третья и последняя часть указывает на элементы `<h2>` внутри элемента `<div>`. В итоге соответствующее форматирование получают все заголовки уровня 2, относящиеся к книжным аннотациям, а заголовки на оставшейся части страницы не изменяются.

А самое главное, вы убираете атрибут `class` из элемента `<h2>`, тем самым формируя следующую более простую разметку.

```
<div class="review">
  <h2>...</h2>
  <p class="byline">...</p>
  <p>...</p>
```

```
  <p>...</p>
  <p>...</p>
  <p>...</p>
</div>
```

Этот прием широко применяется при задании разных правил форматирования различных областей страницы. Он также хорошо сочетается с приемами верстки на основе CSS, описанными в *главе 9* и применяющими элементы `<div>` для создания и размещения отдельных панелей контента.

СЕМИНАР ДЛЯ ОПЫТНЫХ ПОЛЬЗОВАТЕЛЕЙ
Создание таблицы стилей для всего Web-сайта

Правила для классов полезны не только для отделения разных видов контента. Они также удобны при написании правил для всего сайта в одной таблице стилей.

Типовой Web-сайт содержит страницы или группы страниц, которые форматируются по-разному. Например, у вас может быть страница с резюме, несколько страниц с хроникой вашей поездки на остров Гваделупа и еще одна группа страниц с фотогалереей. Вместо трех отдельных таблиц стилей можно создать одну, в которой описано все. Хитрость заключается в использовании разных имен классов при описании каждого раздела. Другими словами, создается класс `resume`, класс `trip` и класс `gallery`. Далее приведена базовая структура, описывающая такой подход.

```
/* Используется на страницах с резюме. */
.resume { ... }
.resume h1 { ... }
.resume h2 { ... }

/* Используется на страницах с дневником поездки. */
.trip { ... }
.trip h1 { ... }
.trip h2 { ... }

/* Используется на страницах с фотогалереей. */
.gallery { ... }
.gallery h1 { ... }
.gallery h2 { ... }
```

Понятно, что на отдельной странице будут применяться только некоторые из этих правил. Но зачастую гораздо легче сопровождать Web-сайт, если все ваши стили хранятся в одном месте.

7

ГЛАВА 7

Добавление графики

Можно с уверенностью сказать, что создатели Интернета никогда не представляли себе, что он будет выглядеть так, как сегодня, набитый картинками, рекламой, видеороликами и анимацией. Они разрабатывали место встречи ведущих научных умов; мы же пришли к тому, что больше напоминает базар в Шри-Ланке. Но никто не возражает, потому что без графики Web-пространство стало бы ужасно унылым местом.

В этой главе вы овладеете искусством Web-графики. Вы узнаете, как вставлять на Web-страницу обычные изображения и тщательно их позиционировать. Вы также поймете, что значит подготовить картинки для Web или найти подходящую им замену в Сети.

Основные сведения об изображениях

Для того чтобы понять, как изображения применяются в Web, необходимо знать следующее.

- Изображения не помещают в HTML-файлы. Каждое изображение хранится в отдельном файле.
- Для отображения картинок на странице в вашем HTML-документе применяется элемент `<img>`.

Вы будете пользоваться изображениями на всем вашем сайте. Возможно вам даже придется применять их для замены обычного текста, если понадобится какой-нибудь особый шрифт.

СОВЕТ

Если вы предполагаете, что какая-то часть контекста на странице — разновидность графики, попробуйте щелкнуть ее правой кнопкой мыши. Если это изображение, большинство браузеров в раскрывающемся меню предложат команду **Save Picture As** (Сохранить изображение как).

Элемент <*img*>

Картинки отображаются на ваших Web-страницах с помощью элемента <img>, который сообщает браузеру, где искать их. Далее приведен пример элемента <img>, который выводит на экран файл с именем photo01.jpg.

```
<img src="photo01.jpg" />
```

Изображения — непарные элементы (см. *разд. "HTML-элементы для основного текста" главы 5*), т. е. не нужно вставлять в элемент отдельные начальный и конечный теги. Вместо этого в конец тега, перед закрывающей угловой скобкой, помещается символ слэша (/).

Изображения — также строчные элементы (см. *разд. "HTML-элементы для основного текста" главы 5*), т. е. они помещаются внутрь других блочных элементов, например, абзацев.

```
<p><img src="photo01.jpg" /></p>
```

Когда браузер считает этот элемент <img>, он запросит файл photo01.jpg. После его извлечения браузер поместит файл на страницу, в то место, где находится элемент <img>. Если файл изображения большой (тяжелый) или подключение к Интернету медленное, можно увидеть, что выполняется этот двухшаговый процесс, т. к. легкие компоненты страницы, например текст, появятся до вывода изображения.

> ***Совет***
>
> Как правило, при наличии большого числа файлов, формирующих ваш сайт, возникает желание их упорядочить и создать в папке, содержащей Web-страницы, подпапку с изображениями. В *главе 8* вы узнаете, как это делается.

Как не удивительно, для вывода изображения необходим единственный компонент HTML, элемент <img>. Но для того чтобы добиться желаемых результатов, придется разобраться в некоторых тонкостях, включая использование альтернативного текста, изменение размеров ваших изображений, выбор формата файла и выравнивание изображений относительно остального контента страницы.

Альтернативный текст

Браузер может вывести на экран изображение, если у него есть атрибут src. Но есть еще один настоятельно рекомендуемый атрибут. Это атрибут alt, который содержит альтернативный или замещающий текст, выводимый браузером при невозможности отобразить само изображение.

Далее приведен пример.

```
<img src="photo01.jpg"
alt="There's no picture, so all you get is this alternate text." />
```

Альтернативный текст полезен не только в уже описанной ситуации, но и в некоторых других.

- Браузер не поддерживает вывод изображений (это, конечно, редкий в наши дни случай, но текстовый браузер Lynx до сих пор встречается на старых Unix-системах).

- Web-посетитель отключил в браузере вывод графики для сокращения времени загрузки страницы (тоже не очень типичная ситуация в наши дни).
- Браузер запрашивает картинку, но не находит ее. (Возможно, вы забыли скопировать ее на ваш Web-сервер.)
- Посетитель — слабовидящий человек и применяет программу экранный диктор (программу, проговаривающую текст, включая слова в атрибуте `alt`).
- Поисковая машина (например, Google) анализирует страницу и весь ее контент, чтобы проиндексировать его в каталоге поиска.

Последние две причины особенно важны. Web-специалисты всегда используют осмысленный текст в атрибуте `alt`, чтобы экранные дикторы и поисковые машины интерпретировали соответствующие изображения корректно.

Не путайте альтернативный текст с текстом всплывающей подсказки, необязательным сообщением, появляющемся на экране, когда посетитель наводит указатель мыши на изображение (рис. 7.1).

Для вставки всплывающего текста используйте атрибут `title`.

```
<img src="matador.jpg"
alt="A matador extends his cape in welcome."
title="Welcome to the ring." />
```

Разные браузеры по-разному отображают альтернативный текст. Internet Explorer выводит этот текст в поле изображения с пиктограммой пропущенной картинки

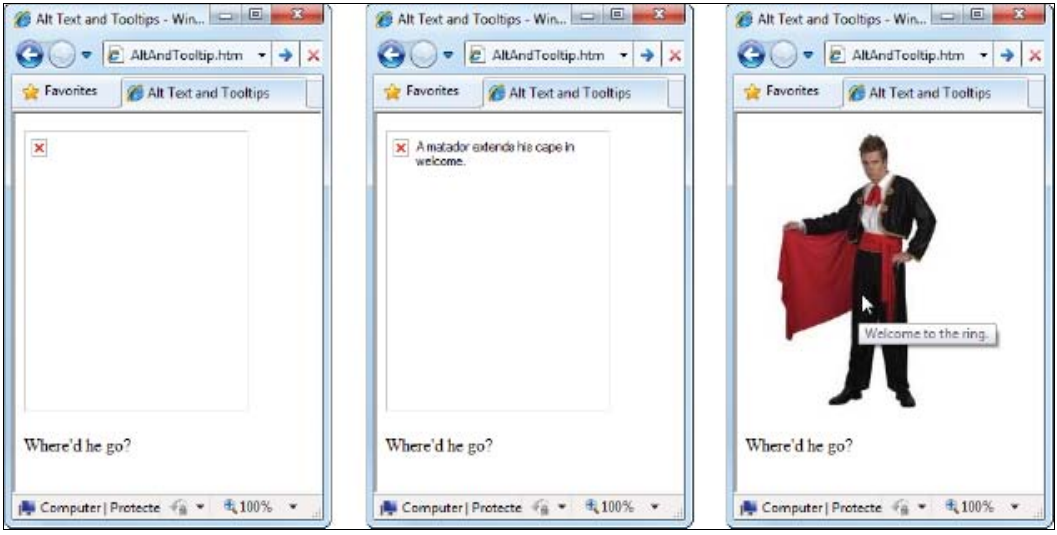


Рис. 7.1. *Слева*: для того чтобы этот элемент `<img>` сработал, следует поместить файл, на который он указывает, в ту же папку, где находится Web-страница. В противном случае вы увидите устрашающую пиктограмму поврежденной ссылки на изображение.
В центре: немного поможет альтернативный или замещающий текст — применяйте его, чтобы объяснить, что именно должен был увидеть посетитель.
Справа: также можно использовать атрибут `title` для задания всплывающего текста, который появится, когда указатель мыши будет наведен на изображение

(см. рис. 7.1, в центре). Firefox отображает замещающий текст как абзац обычного текста на странице, без поля изображения или пиктограммы пропущенной картинки.

Размер изображения

У термина "размер изображения" есть два значения: он может ссылаться на размеры картинки (сколько экранного пространства она занимает на Web-странице) или означать размер файла изображения (количество байтов, необходимых для его хранения). Для Web-разработчиков важны обе эти меры.

Размеры изображения заслуживают внимания, потому что определяют, сколько места на экране занимает изображение. Web-творцы измеряют графику в пикселах. Пиксел представляет одну крошечную точку на компьютерном экране. Web-мир не работает с фиксированными единицами измерения, такими как дюймы или сантиметры, поскольку вы заранее никогда не знаете, какого размера монитор у вашего посетителя и, следовательно, сколько пикселов на нем поместится. (В *главе 9* приводится подробное обсуждение размеров экрана и способов разработки страниц, удовлетворяющих максимальному количеству потенциальных программ-обозревателей.)

Размер графического файла также важен, потому что он определяет, сколько времени займет пересылка изображения в браузер. Большие изображения могут замедлить загрузку страницы, в особенности если на странице много изображений, и вы посылаете их пользователю с необычно медленным подключением к Интернету. (Для того чтобы оценить размер файла и способы управления размером, необходимо разобраться в разных форматах файлов изображений, используемых браузерами, — тема, которая обсуждается в следующем разделе.)

Элемент `<img>` позволяет изменять размеры изображения с помощью своих необязательных атрибутов `height` и `width`. Рассмотрим следующий элемент.

```
<img src="photo01.jpg" alt="An explicitly sized picture" width="100"
height="150" />
```

В этих строках кода задается ширина картинки — 100 пикселов и высота — 150 пикселов. Если это не соответствует реальным размерам вашей исходной картинки, браузеры растягивают или, наоборот, обрезают изображение до тех пор, пока оно не будет соответствовать заданному вами размеру (рис. 7.2).

> ***Примечание***
>
> Применяйте атрибуты `height` и `width` с осторожностью. Порой начинающие Web-разработчики используют их для создания миниатюр изображений (thumbnails), маленьких версий больших картинок. Но применение атрибутов `height` и `width` для уменьшения большой картинки приводит к снижению производительности — а именно, браузер вынужден загружать исходное большое изображение, хотя и выводит его на экран в уменьшенном размере. С другой стороны, если создать миниатюры в графическом редакторе, например Photoshop, можно будет сохранить их в файлах меньшего размера и повысить скорость загрузки ваших страниц.

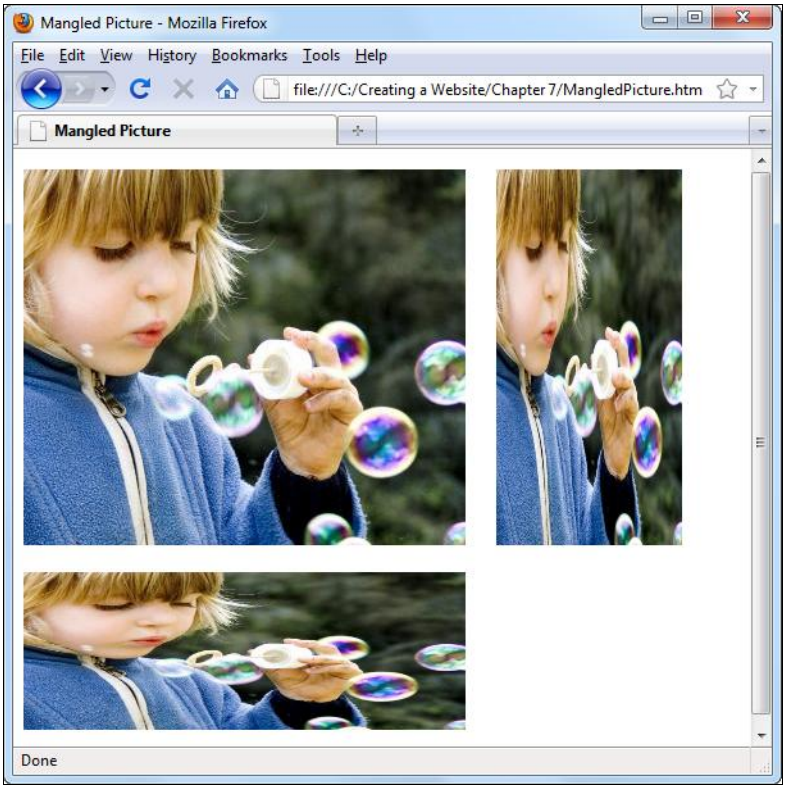


Рис. 7.2. Для изменения размера изображения никогда не применяйте атрибуты `height` и `width`, т. к. результаты почти всегда будут неудовлетворительными. Увеличенные картинки окажутся ступенчатыми, а сжатые — размытыми, если же вы измените отношение высоты к ширине (как на картинке справа сверху или на нижней картинке), браузеры нарушат нормальные пропорции изображения

Многие Web-разработчики не включают атрибуты изображения `height` и `width`. Но опытные дизайнеры иногда добавляют их, указывая те же размеры, что и у реального изображения. Как это ни странно звучит, у подобного приема есть несколько веских причин.

Во-первых, когда вы вставляете атрибуты размера изображения, браузеры знают величину изображения и могут начать выкладывать страницу, пока загружается графика (см. рис. 7.1, слева). С другой стороны, если атрибуты `height` и `width` не включены, браузер не будет знать о размере изображения, пока оно полностью не загрузится, и будет вынужден в этот момент реорганизовывать контент. Это может вывести из равновесия ваших посетителей, если у них медленные подключения, а страницу они уже начали читать.

Вторая причина применения атрибутов высоты и ширины — управление с их помощью полем вывода изображения в тех случаях, когда браузер не может загрузить изображение (см. рис. 7.1, в центре). Но не следует полагаться на этот прием, т. к. он не действует в Firefox. Браузер Firefox игнорирует атрибуты высоты и ширины у пропущенных изображений и не отображает поле для их вывода. (Для того чтобы помешать незагруженному изображению испортить макет вашей страницы, следует

разбивать страницы на отдельные секции с помощью элементов `<div>` и размещать их с помощью таблиц стилей, как описано в *главе 9*.)

Так нужно ли применять атрибуты `height` и `width`? Решать вам, но для среднестатистического Web-сайта они более опасны, чем полезны. Если уж вы их применяете, то обязательно обновляйте их при изменении размера вашего изображения, что очень скоро превратится в утомительное занятие. Упростите свою жизнь и сделайте все картинки одинакового размера, а затем напишите правило в таблице стилей, которое задает свойства высоты и ширины.

ПРИМЕЧАНИЕ

Некоторые HTML-редакторы, например Expression Web, автоматически добавляют атрибуты `height` и `width`, когда вы вставляете изображение.

Форматы графических файлов

Браузеры не могут выводить на экран любое изображение. В действительности они ограничены лишь несколькими графическими форматами, включая следующие.

- Формат JPEG (произносится "джей-пег") подходит для фотографий, допускающих некоторое снижение качества. (Как вы узнаете через несколько минут, формат JPEG сжимает файл с изображением, и в результате он загружается быстрее.) Формат JPEG не так хорош для картинок с текстом или графикой.
- Формат GIF (произносится "джиф" или "гиф") подходит для графики с малым количеством цветов (например, простые логотипы или аппликации). Если применить его для вывода фотографий, результаты будут ужасны.
- Формат PNG (произносится "пинг") подходит для всех видов изображений, но не всегда сжимает их так же хорошо, как JPEG. PNG особенно хорош для маленьких контрастных изображений (например, логотипов) и в настоящее время применяется как самая мощная замена стандарта GIF.
- Формат SVG (Scalable Vector Graphics, Масштабируемая векторная графика) — перспективный стандарт для векторных изображений (логотипов и рисунков, содержащих текст и контуры, но не фотографий). SVG обладает рядом преимуществ, включая маленький размер и гибкость, если применяется с подходящими типами изображений. Можно менять размер SVG-изображений без потери деталей и получения размытых изображений. Но Internet Explorer 8 и более ранние версии не поддерживают SVG, т. е. он все еще не вполне пригодный формат для Web-графики. Поэтому в данной главе этот формат не обсуждается.

ПРИМЕЧАНИЕ

Дополнительную информацию об SVG (и некоторых обходных путях, заставляющих его отчасти работать в Internet Explorer) см. на странице **http://tinyurl.com/2gy2vyg**.

К графике на Web-страницах применяют два вида сжатия: *с потерями* (lossy), сжимающее файлы в большей степени, чем второй вид, но при этом снижающее качество изображения, и *без потерь* (lossless), сохраняющее качество изображения, но сжимающее в меньшей степени. Подробное описание см. далее во *врезке "На про-*

фессиональном уровне. Как выполняется сжатие". В табл. 7.1 дается краткий обзор разных графических форматов, а на рис. 7.3 можно сравнить их на Web-странице.

Таблица 7.1. *Форматы файлов изображений для Web*

Формат	Тип сжатия	Максимальное количество цветов	Лучше всего подходит для
JPEG	С потерями	24-битный (16,7 млн цветов)	Фотографий
GIF	Без потерь	8-битный цвет (256 цветов)	Простых логотипов, графического текста и диаграмм с рисунками
PING-8	Без потерь	8-битный цвет (256 цветов)	Применяется нечасто, т. к. подобен GIF (но у него лучше поддержка прозрачности)
PING-24	Без потерь	24-битный (16,7 млн цветов)	Изображений, которые могли бы быть GIF-файлами, но им нужно больше цветов
SVG	Без потерь (но сжатие не обязательно, т. к. объем SVG-данных и так мал)	24-битный (16,7 млн цветов)	Изображений, сделанных в программе-иллюстраторе. Версии IE, предшествующие IE 9, не поддерживают его

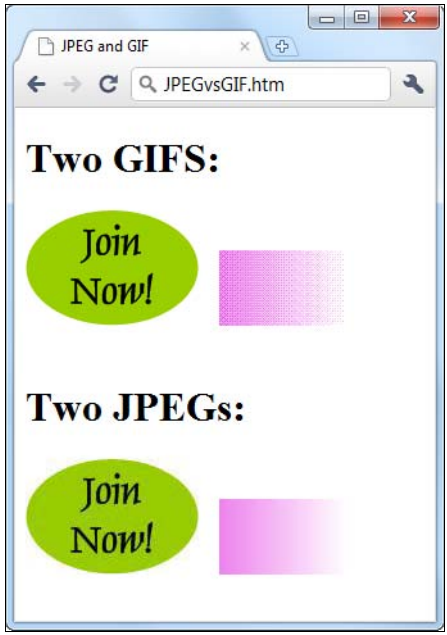


Рис. 7.3. JPEG и GIF — основные графические форматы для Web. Вы заметите, что GIF-форматы формируют более четкий текст, а JPEG гораздо лучше обрабатывают непрерывные цветные полосы. В GIF-формате дополнительные цвета имитируются с помощью сглаживания, процесса смешивания разных цветовых точек для имитации сплошного цвета. Результаты очевидно не профессиональны. (Вы, возможно, не увидите снижение качества текста на приведенном черно-белом скриншоте, но если посмотрите загружаемые из Интернета примеры к этой главе, сразу же заметите разницу.) Именно поэтому стандарт PNG по большей части вытеснил GIF

Выбор подходящего формата изображения

Очень важно уяснить, какой формат применить в конкретной задаче. Для того чтобы помочь вам принять решение, далее приведены наводящие вопросы.

- *Ваше изображение — крупная фотография и обладает хорошими градациями цвета?*

  ДА: формат JPEG лучше всего подходит для сокращения размеров крупных изображений с большим количеством деталей. Если используемая графическая программа позволяет, можно выбрать применяемую степень сжатия.

- *У вашего изображения резкие края и ему необходимо более 256 цветов?*

  ДА: в данном случае лучше всего формат PNG. Он поддерживает полный цвет, обеспечивает сжатие без потерь и ни одна деталь не будет потеряна. Если у вашего изображения ограниченное количество цветов (256 или меньше), можно вместо PNG использовать формат GIF, но веских причин отдавать предпочтение другому формату нет.

> ***Примечание***
>
> Некоторые браузеры предлагают дополнительные варианты, но лучше их избегать для обеспечения максимально широкой совместимости браузеров. Например, Internet Explorer поддерживает точечные или растровые изображения (графические файлы с расширением bmp). Не пользуйтесь ими, не только потому, что они вводят другие браузеры в замешательство, но и из-за их чрезвычайно большого размера: у этого стандарта нет сжатия.

- *Есть ли у вашего изображения прозрачные области?*

  ДА: применяйте PNG. Как вы узнаете из *разд. "Размещение картинок на цветном фоне" данной главы*, формат PNG обладает поддержкой частичной прозрачности, позволяющей создавать плавные переходы в прозрачных областях и делать изображение более естественным. У формата GIF есть более примитивное, двухуровневое (все или ничего) средство поддержки прозрачности, а формат JPEG вообще не предлагает никакой поддержки прозрачности.

> ***Примечание***
>
> Вы уже прочли о том, что HTML хранит элементы страницы как газетные заголовки в прямоугольных областях. Это справедливо и для изображений, даже картинка надувного пляжного мяча помещается в прямоугольную область. Вот почему так важна прозрачность в изображениях. Она позволяет помещать изображение на страницу так, что сквозь него просвечивает фон, и в результате получается страница с органично встроенным изображением.

ЧАСТО ЗАДАВАЕМЫЙ ВОПРОС

Типичные размеры файлов с изображениями

Сколько места на диске занимает обычное изображение?

Единого ответа нет, т. к. он зависит от нескольких факторов, включая размеры картинки, используемый графический формат, степень примененного сжатия и способность

изображения реагировать на метод сжатия. Тем не менее далее приведено несколько важных замечаний, о которых не стоит забывать.

Размер файла типичного логотипа Web-сайта очень мал. Например, фирменный логотип Google умещается практически в 20 Кбайт (файл в формате PNG).

Фотографии могут занимать гораздо больше места. Примером нижнего предела может служить изображение в статье на Web-сайте New York Times, редко занимающее более 70 Кбайт. А верхним пределом может быть изображение товара типичного участника аукциона eBay объемом 300 Кбайт. При таком размере изображение обычно занимает большую часть окна вашего браузера. Но это не идет ни в какое сравнение с размером изображения без сжатия. Например, даже древняя одномегапиксельная камера может получить непосредственное несжатое изображение объемом около 3 000 Кбайт. На Web-странице его можно сжать до 300 Кбайт и больше, используя файл в формате JPEG с качеством нижнего уровня.

Сжатие

Много лет тому назад Web-соединения были медленными, и Web-разработчики часами бились над размером каждого изображения на их Web-сайтах. Сегодня размер изображения не важен, и очень часто страницы засоряют рекламной анимацией или реальной видеосъемкой. Тем не менее профессиональные Web-сайты до сих пор тратят время на сокращение объема своей графики. Такой подход свидетельствует о внимании Web-сайта ко всем: даже к посетителям из тех стран, где широкополосные соединения трудно найти, или к завсегдатаям местной кофейни, страдающим от неустойчивого беспроводного соединения. Маленький объем также подходит для мобильных устройств, владельцы которых часто вынуждены платить ежемесячно за объем данных, загружаемых из Интернета.

Для того чтобы ваш сайт оставался максимально быстрым, легким и эффективным, следуйте приведенным далее рекомендациям.

- Применяйте картинки маленького размера. Если вам действительно нужно заполнить все окно браузера, создайте уменьшенную версию картинки, поместите ее на страницу и превратите ее в ссылку (см. *главу 8*). В этом случае, когда кто-нибудь щелкнет картинку кнопкой мыши, вы сможете открыть новую страницу с полноразмерным изображением.
- Используйте подходящий графический формат. Для больших фотографий — это JPEG.
- Для большего сжатия снижайте уровень качества ваших JPEG-файлов. Но сначала проверьте их и убедитесь, что вы готовы смириться с потерей некоторых деталей. При сжатии изображений в формате JPEG вы столкнетесь с различными проблемами, носящими общее название "артефакты сжатия". Самые распространенные артефакты — ступенчатые области на изображении, ореолы по краям и общая размытость. Одни изображения больше страдают от этих дефектов, чем другие, все зависит от количества мелких деталей на изображении.

ПРИМЕЧАНИЕ

Большинство графических программ позволяет выбрать степень сжатия изображения, а многие даже позволяют просмотреть результат, прежде чем сохранять что-либо.

НА ПРОФЕССИОНАЛЬНОМ УРОВНЕ

Как выполняется сжатие

Все три распространенных Web-формата изображений применяют сжатие для сокращения графической информации. Но типы сжатия, применяемые в каждом формате, существенно отличаются.

Форматы GIF и PNG поддерживают сжатие без потерь, т. е. ваша картинка не теряет никакой информации. При сжатии без потерь для свершения таинства сокращения объема применяются различные методы, например, в файле может быть найден повторяющийся фрагмент, и каждое его вхождение заменяется короткой аббревиатурой. Когда браузер распакует ваш файл, он получит все исходные данные изображения обратно.

В формате JPEG применяется сжатие с потерями, т. е. некоторая информация о вашей картинке отбрасывается (или теряется).

В результате качество картинки снижается, и нет способа вернуться к ее первоначальному превосходному виду. Но формат JPEG достаточно хитер и пытается обмануть ваш глаз, отбрасывая информацию, которая не слишком портит картинку. Например, он может превратить чуть-чуть отличающиеся цвета в один и тот же цвет или заменить мелкие детали размытыми пятнами, поскольку человеческий глаз не настолько чувствителен, чтобы заметить незначительные изменения цвета и формы. Обычный итог — картинка, которая выглядит менее ярко и более размыто. С другой стороны, уменьшение размера, получаемое при сжатии с потерями, гораздо существеннее того, что предлагает сжатие без потерь.

Размещение картинок на цветном фоне

Программы редактирования графики всегда сохраняют файлы изображений в виде прямоугольников, даже когда само изображение не прямоугольно. Например, если вы создадите картинку улыбающейся мордашки, программа-редактор сохранит круглую иллюстрацию на белом прямоугольном фоне.

Если у вашей страницы фон тоже белый, никаких проблем не возникнет, потому что фон изображения сольется с фоном вашей страницы. Но если у вашей страницы фон другого цвета (см. *разд. "Цвета" главы 6*), вы столкнетесь с графическим дефектом, показанным на рис. 7.4.

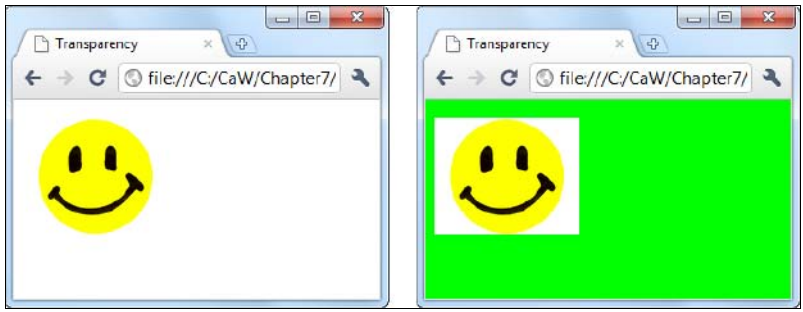


Рис. 7.4. *Слева*: когда картинка помещается на страницу с белым фоном, улыбающаяся физиономия становится частью страницы.
Справа: если фон не белый, белый прямоугольник вокруг вашей картинки становится виден невооруженным глазом

Web-разработчики нашли два решения этой проблемы. Первая остроумная идея — использовать прозрачность, характеристику, общую для графики в форматах PNG и GIF. Суть в том, что ваше изображение содержит *прозрачные пикселы*, пикселы вообще не имеющие цвета. Когда браузер сталкивается с ними, он не рисует ничего. Он позволяет отображаться фону страницы. Для того чтобы сделать часть вашего изображения видимой насквозь, вы определяете прозрачный цвет с помощью вашей графической программы. Например, в приведенном примере белый фон изображения улыбающейся физиономии задается как прозрачный цвет.

Несмотря на то, что прозрачность кажется удачным способом подбора нужного фона изображения, на практике это редко выглядит хорошо. Обычно при этом виден ступенчатый край там, где заканчиваются цветные пикселы вашей картинки и начинается фон Web-страницы (рис. 7.5).

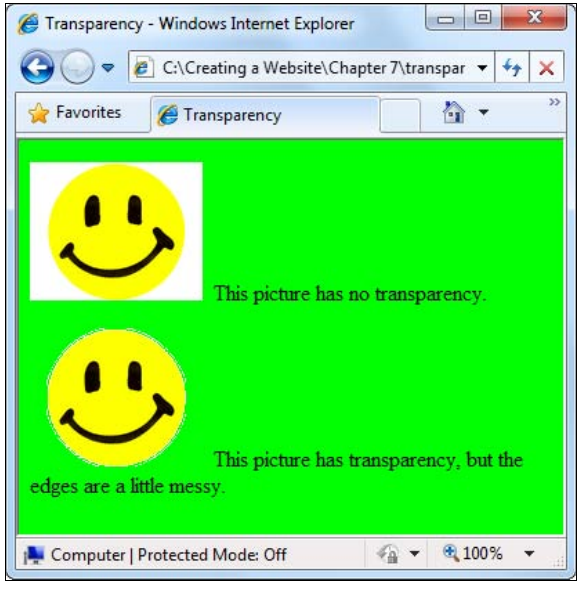


Рис. 7.5. На нижней картинке используется прозрачность, но результат — ступенчатый край вокруг улыбающейся физиономии — не назовешь превосходным. Для сглаживания этого края графические программы применяют сложный метод, который смешивает цвет картинки с цветом фона страницы. Браузеры не способны выполнить этот трюк, поэтому создаваемые ими края совсем не такие гладкие

Самый легкий путь решения этой проблемы — вместо прозрачности использовать подходящий цвет фона во время создания Web-графики. Другими словами, когда создаете улыбающуюся физиономию, задайте для нее тот же цвет фона, что и на вашей Web-странице. Ваша графическая программа далее сможет применить *сглаживание* (anti-aliasing), специальную технологию сглаживания ступенчатых краев, обеспечивающую им привлекательный внешний вид. В этом случае края хорошо сливаются с фоном и, когда вы выводите изображение на вашей Web-странице, оно отлично вписывается. У этого подхода есть лишь один недостаток — отсутствие гибкости. Если изменится цвет фона страницы, придется переделывать всю графику.

Другой метод — применение частичной прозрачности (который также называют *альфа-смешиванием*, где значение альфа — число, задающее прозрачность пиксела). Например, вместо создания изображения, которое резко переходит от цветных пикселов к прозрачным, используете картинку с плавным переходом на краях, созданным с помощью окантовки из полупрозрачных пикселов. Этот метод кажется замечательным и поддерживается в PNG-графике. Но для создания подобного эффекта вам понадобятся серьезные возможности программы Photoshop. (Если вас заинтересовал этот путь, начните с краткого руководства на странице **http://tinyurl.com/ypf78g**.) Как ни печально, но это плата за создание безупречной Web-графики.

НА ПРОФЕССИОНАЛЬНОМ УРОВНЕ

Графические программы

Выбор формата файла изображения — ваша задача. Большинство хороших графических программ (таких как Adobe Fireworks и Adobe Photoshop) сохраняют документы в файлах специального формата, который позволяет выполнять сложные процедуры редактирования. Photoshop, например, сохраняет файлы в формате psd. Когда ваша картинка готова к использованию на Web-странице, сохраните файл psd в *другом* формате, одном из предназначенных специально для Web (JPEG, GIF или PNG). Обычно это делается с помощью команд меню **File | Save As** (Файл | Сохранить как) (но иногда может быть и немного другая последовательность, например, **File | Export** (Файл | Экспортировать) или **File | Save For Web** (Файл | Сохранить для Web)).

Как показывает практика, всегда нужны как минимум две версии каждого созданного вами изображения: экземпляр в исходном формате, применяемом вашей графической программой, и копия в формате JPEG, GIF или PNG, который вы используете на своем Web-сайте. Исходный файл понадобится для внесения изменений по мере необходимости и гарантированного сохранения исходного качества картинки для будущих версий.

После выбора формата изображения графическая программа предложит ряд других параметров, позволяющих настроить детали, такие как степень сжатия, например. Чем больше степень сжатия, тем меньше станет ваш файл с изображением, но тем ниже будет качество картинки. Некоторые самые простые графические редакторы (например, программа Paint, входящая в состав ОС Windows) не позволяют настроить эти параметры, поэтому вы останетесь со стандартными настройками программы.

Графические редакторы, как правило, поставляются в двух базовых вариантах — как *редакторы изображений*, позволяющие ретушировать картинки и применять причудливые визуальные эффекты (например, Adobe Photoshop), и как *программы для рисования*, предназначенные для создания иллюстраций, включающих графические фигуры и текст (например, Adobe Illustrator). Если вы редактируете снимки офисной вечеринки, чтобы изъять неприличные моменты, нужен графический редактор. Если же вы разрабатываете логотип только что созданной компании по изготовлению печенья, потребуется программа для рисования.

Если у вас нет возможности приобрести профессиональную графическую программу, можно поискать какой-нибудь сайт с условно-бесплатным программным обеспечением, например **www.download.com**. Есть два популярных бесплатных графических редактора: GIMP (**www.gimp.org**), поддерживаемый всеми основными операционными системами, и Paint.NET (**www.getpaint.net**), работающий только под Windows. Можно также воспользоваться менее мощными, но вполне впечатляющими интерактивными графическими редакторами Phoenix и Vector на Web-сайте **www.aviary.com**.

Изображения и стили

У элемента `<img>` есть несколько необязательных атрибутов, которые можно использовать для управления выравниванием изображения и его границами. Но в современном мире эти атрибуты считаются устаревшими, и мы не будем применять их в этой книге. Вы узнаете более современный метод позиционирования изображений — с помощью таблиц стилей.

Встраивание изображений в текст

Если не предпринимать никаких дополнительных шагов, браузер вставляет любое изображение в поток HTML-текста. Он совмещает нижнюю сторону изображения с *базовой линией* окружающего его текста, как показано на рис. 7.6. (Базовая линия — это воображаемая линия, на которой располагается строка текста.)

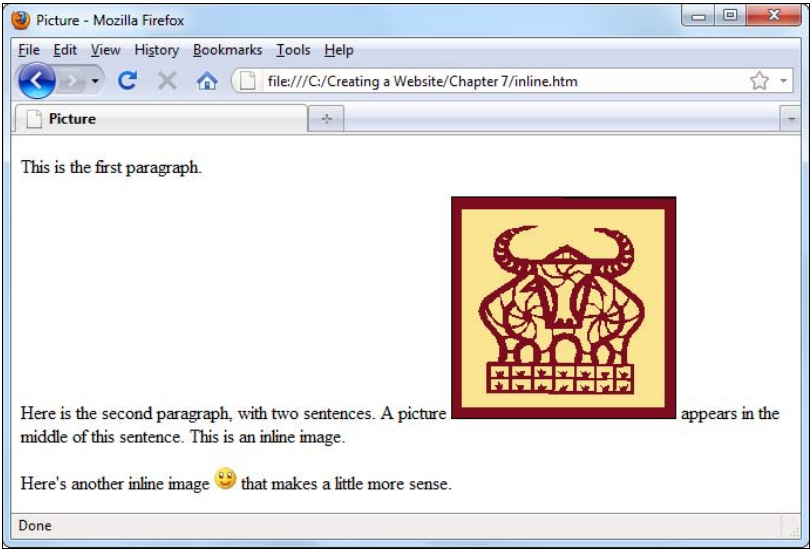


Рис. 7.6. Как правило, вам не нужна картинка в обычной строке текста (если это не маленький смайлик, как те, что встречаются в программах обмена мгновенными сообщениями). Для грамотного отделения изображений от текста можно воспользоваться абзацами, переносами строк или таблицами

Вертикальное выравнивание изображения можно изменить с помощью свойства `vertical-align`. Задайте значение `top`, `middle` или `bottom`, чтобы выровнять изображение по верху, середине или низу строки текста.

Далее приведен пример со встроенным стилем, использующим свойство `vertical-align` для выравнивания изображения по верхней границе строки текста.

```
<img src="happy.gif" alt="Happy Face" style="vertical-align: top" />
```

Этот прием подходит для выравнивания очень маленькой картинки, вроде графического маркера. Но он плохо работает в случае крупных изображений. Все потому, что независимо от выбранного значения свойства `vertical-align`, рядом с изобра-

жением может расположиться только одна строка текста (см. рис. 7.6). Если вы хотите создать плавающие изображения с обтеканием текста, см. *разд. "Обтекание текста вокруг изображения" данной главы*.

Границы

В *главе 6* вы знакомились со свойствами, позволяющими вставить или изменить границы вокруг текстовых полей. Вряд ли вас удивит, что такие же границы можно задать и вокруг изображений.

Например, далее приведено описание стиля, задающего тонкую вдавленную (в виде канавки) границу вокруг изображения.

```
img.BorderedImage {
  border-style: groove;
  border-width: 3px;
}
```

Это правило, как и любые другие правила таблиц стилей, необходимо поместить во внутреннюю таблицу стилей на текущей Web-странице или во внешнюю таблицу стилей, которую использует ваша страница (обсуждение разницы см. в *разд. "Три вида стилей" главы 6*).

Обратите внимание на то, что в данном примере стиль описан с именем класса (`BorderedImage`). Вы не хотите, чтобы браузер применял стиль ко всем изображе-

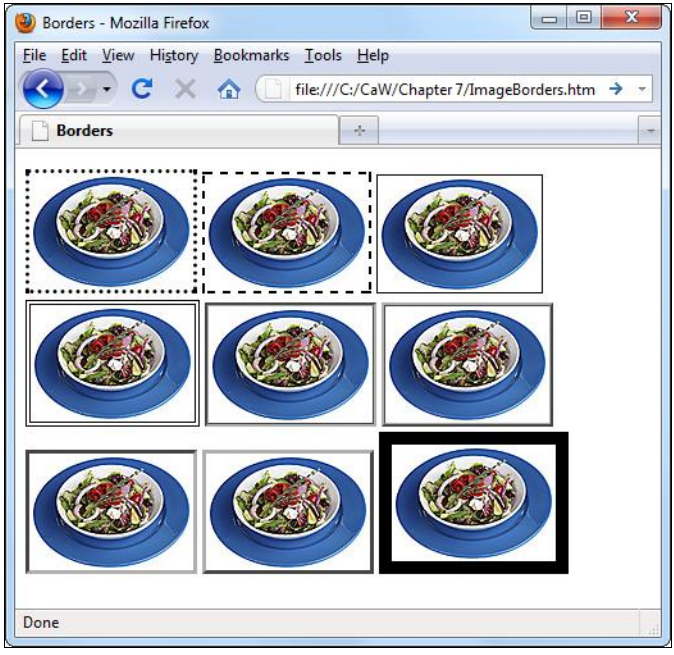


Рис. 7.7. В этом примере показано несколько строчных изображений, отделенных одно от другого одним пробелом. Все изображения в примере одинаковы, но у всех разные границы. Браузер размещает, пока может, все изображения в одной строке. Дойдя до правого края своего окна, он переходит на следующую строку. Если изменить размеры окна, расположение изображений тоже изменится

ниям. Вместо этого вы хотите с помощью атрибута `class` выбирать, когда его применять.

```
<img src="food.jpg" alt="A Meal" class="BorderedImage" />
```

На рис. 7.7 показаны основные стили границ. Напоминаю, что вы можете изменить толщину любой границы и ее внешний вид.

Обтекание текста вокруг изображения

Применение строчных изображений — самый легкий способ вставки изображений на Web-страницу, но у него есть недостаток: изображения выводятся внутри текста. Чтобы помешать этому, можно разделить изображения и текст с помощью элементов абзацев (`<p>`), переносов строк (`<br>`), горизонтальных линий (`<hr>`) и других разделителей. Можно, например, поместить картинку между двумя абзацами текста следующим образом.

```
<p>This paragraph is before the picture.</p>
<p><img src="food.jpg" alt="A Meal" /></p>
<p>This paragraph is after the picture.</p>
```

Строчные элементы всегда остаются на месте. Они никогда не перемещаются в неожиданное место.

Но порой нужен обратный эффект. Вместо разделения изображений и текста нужно поместить их рядом. Например, заставить текст обтекать изображение с одной стороны.

Изображения с текстом, обтекающим его с одной или другой стороны, называют *свободно перемещаемыми* или *плавающими*, поскольку они размещаются рядом с массивом текста (рис. 7.8). Создаются плавающие изображения с помощью свойства CSS `float`. Вы задаете значение свойства `left` или `right`, которое пристраивает изображение к левому или правому краю текста.

```
img.FloatLeft {
  float: left;
}
```

Обратите внимание, в примере использовано имя класса. Вы, возможно, не захотите, чтобы все изображения на вашей Web-странице были плавающими, в этом случае всегда неплохо применить имя класса. Далее приведен пример, в котором за изображением, использующим описанный класс, следует текст.

```
<p>
  <img src="food.jpg" alt="A Meal" class="FloatLeft" />
  If you place a floating image at the beginning of a paragraph,
  it floats in the top-left corner, with the text wrapped along
  the right edge.
</p>
```

Если задаете плавающее изображение, имеет смысл одновременно отрегулировать значения полей изображения, таким образом создается немного свободного пространства между вашим изображением и окружающим текстом.

```
img.FloatLeft {
  float: left;
  margin: 10px;
}
```

На рис. 7.8 показано несколько плавающих изображений.

Примечание

Если хотите добиться нужного обтекания текста, всегда помещайте элемент `<img>` сразу перед текстом, который должен обтекать изображение.

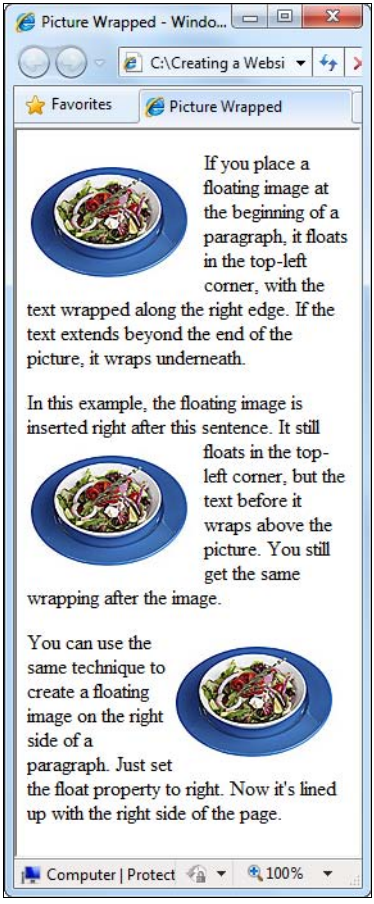


Рис. 7.8. Помните о том, что все файлы изображений содержат прямоугольные области с окружающим изображение пустым пространством (см. рис. 7.4). В результате браузер располагает текст вокруг границ этих невидимых областей

Обтекание текста может показаться немного замысловатым приемом, т. к. получаемый результат зависит от ширины окна браузера. Например, вы можете считать, что у вас достаточно текста для обтекания графики, но в широком окне текст может занять всего несколько коротких строчек, позволив оставшейся части контента налететь на вашу графику, чего вы вряд ли хотели (рис. 7.9).

Чтобы этого не случилось, можно поместить изображения в разные контейнеры, которые подобны разным ячейкам в таблице. Или же вы можете вручную прервать

в любой точке обтекание текста в вашем браузере с помощью свойства `clear` в элементе переноса строки (`<br>`).

```
<br style="clear: both;" />
```

Вставьте эту строку в конец абзаца, обтекающего изображение.

```
<p>
  <img src="food.jpg" alt="A Meal" class="FloatLeft" />
  Here is a paragraph with a floating image.
  <br style="clear: both;" />
</p>
<p>
  <img src="food.jpg" alt="A Meal" class="FloatLeft" />
  This should be a separate paragraph with another floating picture.
</p>
```

Свойство `clear` в элементе `<br>` сообщает браузеру о прекращении обтекания, гарантируя вывод нового абзаца после плавающего изображения (рис. 7.9).

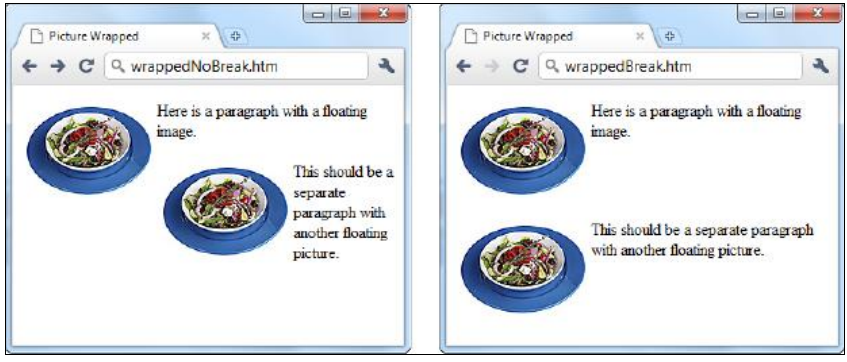


Рис. 7.9. *Слева*: без свойства `clear` есть опасность наложения плавающих изображений друг на друга, если окно браузера будет шире, чем вы рассчитываете.
Справа: свойство `clear` позволяет отключить обтекание в заданной точке вашего документа. В результате вы получите дополнительное свободное пространство

Глядя на приведенные примеры, можно решить, что свойство `float` помещает картинку на левую или правую сторону страницы, но это не совсем точно. В CSS HTML трактует каждый элемент на странице как контейнер. Когда вы создаете плавающее изображение, оно переходит на левую или правую сторону его контейнера. В предыдущих примерах это значит, что изображение перемещается на левую или правую сторону абзаца, т. к. абзац — это элемент, содержащий изображение.

В только что приведенном примере абзац занимает всю ширину страницы. Но это не единственно возможный вариант. Для того чтобы увидеть нечто другое, можно применить стилевые правила и поместить абзац в рамку для предупреждающих сообщений с увеличенными отступами.

Для того чтобы это проверить, нужно поместить изображение и абзац в следующий элемент `<div>`.

```
<div class="Box">
  <p>
    <img src="food.jpg" alt="A Meal" class="FloatLeft" />
    <b>But Wait!</b> A tip box can interrupt the discussion
    to let you know just how good mixed veggies can taste.
    Of course, this tip box is really just an ordinary paragraph with
    the right border and margin style properties.
  </p>
</div>
```

Затем с помощью стилевого правила можно задать изящную рамку для элемента .

```
div.Box {
  margin-top: 20px;
  margin-bottom: 10px;
  margin-left: 70px;
  margin-right: 70px;
  padding: 5px;
  border-style: dotted;
  border-width: 2px
}
```

На рис. 7.10 показан результат.

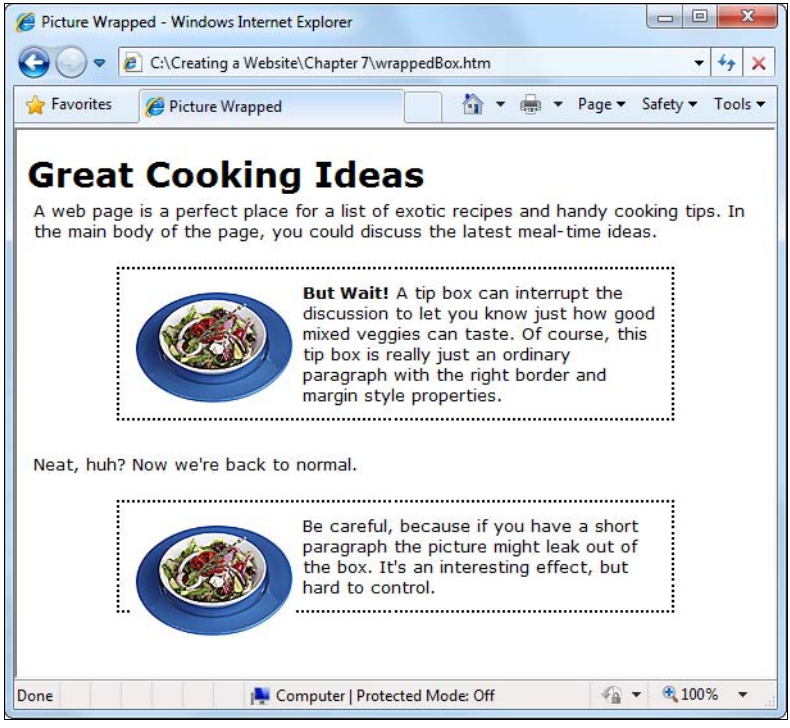


Рис. 7.10. Искусно применяя стили, можно размещать картинки на странице с той же ловкостью, какую обеспечивают стили при работе с текстом

Вставка подписей к изображениям

Еще один важный штрих — вставка пояснительных подписей над или под изображением. В случае строчных изображений это сделать очень легко: поместите строку текста сразу до или после изображения, отделенного переносом строки. В случае плавающего изображения это не так просто. В этом случае нужно иметь изображение *и* подпись, свободно перемещаемые в одном направлении.

Оказывается, решение очень простое. В стилевом правиле `FloatLeft`, приведенном ранее, измените имя `img.FloatLeft` на `.FloatLeft`, теперь вы сможете применять это правило к *любому* элементу.

```
.FloatLeft {
  float: left;
  margin: 10px;
}
```

Затем поместите элемент `<img>` и ваш текст в элемент `<span>`. Теперь можно весь элемент `<span>` сделать плавающим, применив правило `FloatLeft`.

```
<span class="FloatLeft">
   <img src="planetree.jpg" alt="Plane Tree" />
   <br />
   <i>The bark of a plane tree</i>
</span>
```

На рис. 7.11 показан результат.

Примечание

В данном примере вы использовали элемент `<span>` вместо элемента `<div>`, потому что его можно поместить внутрь других блочных элементов, таких как абзац. Другими словами, применяя элемент `<span>`, можно легко поместить плавающий контейнер с изображением и подписью внутрь одного из абзацев.

АНОНС HTML5

Лучший способ свободного перемещения изображения

На рис. 7.11 показано, как создать плавающий блок для хранения изображения и пояснительного текста к нему. Это популярная методика в мире Web-дизайна, но она не совершенна и затуманивает структуру Web-страницы. К примеру, непонятно, что элемент `<span>` предназначен для представления изображения и что текст как-то связан с изображением.

Возможно, вам не кажется это серьезной проблемой (так оно и есть), но высоколобые борцы за смысловую разметку мечтают о мире, в котором у Web-страниц более стройная структура, помогающая программам, будь то браузер, поисковая машина, специальное средство доступа или что-нибудь еще, лучше идентифицировать элементы страницы. (Дополнительную информацию об этом концептуальном сдвиге см. в *разд. "Логическая структура вместо физического форматирования" главы 5*.)

В HTML5 добавлены два новых элемента, позволяющих четко и строго структурированно определить изображение: `<figure>` и `<figcaption>`. Элемент `<figcaption>` содержит текст, а элемент `<figure>` — весь набор: изображение и все остальное. Далее показано, как следовало бы применить их для исправления предыдущего примера.

```
<figure class="FloatLeft">
  <img src="planetree.jpg"
    alt="Plane Tree" />
  <figcaption>The bark of a plane tree
  </figcaption>
</figure>
```

Ни один из новых элементов не содержит форматирования, поэтому вам придется применять правила таблиц стилей, чтобы добиться нужного внешнего вида. Но реальная проблема состоит в том, что, как и другие новые элементы HTML5, вы не можете форматировать элементы `<figure>` и `<figcaption>` с помощью CSS в версиях Internet Explorer, предшествующих IE 9. Так что если не хотите использовать обходной прием на JavaScript (описанный в *разд. "Семантические элементы HTML5" приложения 1*), фанатам семантического Web-пространства придется подождать.

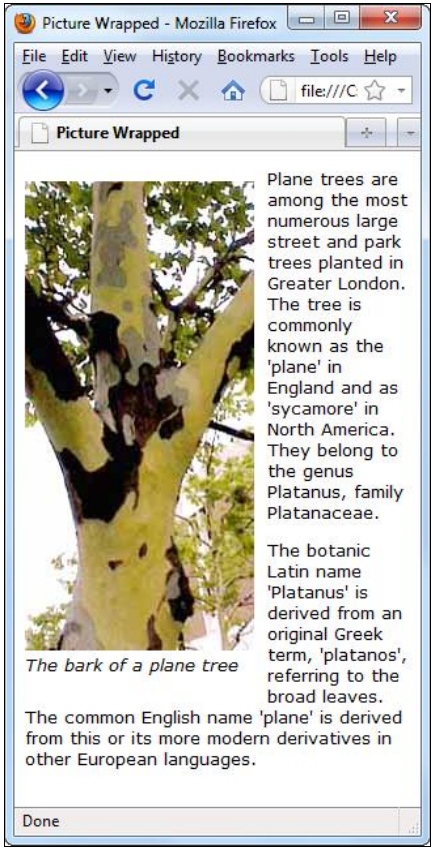


Рис. 7.11. Для создания подписей к плавающим изображениям применяйте стили

Фоновые изображения

CSS делает возможным применение изображения в качестве фона страницы, что очень удобно для создания "тематических" Web-сайтов. Например, можно использовать светлую пергаментную бумагу как фон для литературного сайта. На сайт

поклонников фильма "Сумерки" можно было бы для пользы дела поместить изображение темного кладбища. Некоторые сочтут такой прием отвлекающим, но его стоит принять во внимание, если хотите добавить истинный драматизм и не перегнуть палку.

Совет

Фоновые изображения легко могут сделать ваш Web-сайт безвкусным. Избегайте их использования на странице с резюме или на деловом сайте. Если же хотите добавить немного китча, дерзайте!

Web-дизайнеры почти всегда выбирают мозаичные фоновые изображения, т. е. браузер копирует маленькую картинку снова и снова, пока она не заполнит все его окно (рис. 7.12). Невозможно одним изображением заполнить окно браузера, т. к. нет способа узнать ширину и высоту окна, заданные пользователями. Если бы даже у вас были точные размеры экрана посетителя, не следовало создавать изображение такого большого размера, т. к. его загрузка длилась бы нереально долго.

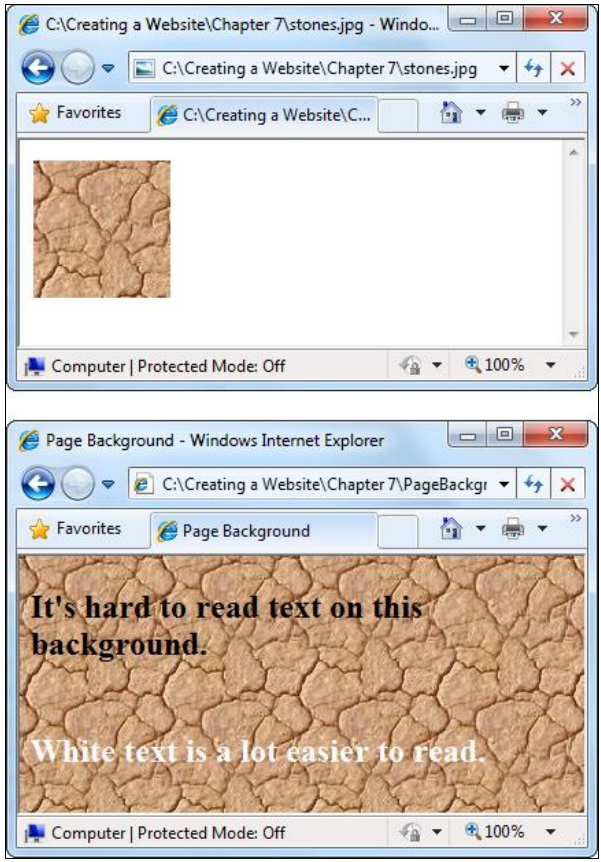


Рис. 7.12. *Вверху*: начинаем с маленькой картинки, элемента мозаики с каменистой текстурой. *Внизу*: с помощью таблиц стилей можно создать мозаичный фон из маленькой картинки, заполняющий всю страницу. В хорошем мозаичном изображении края подгоняются для создания иллюзии единой большой картины

Для создания мозаичного фона применяется стилевое свойство `background-image`. Первый шаг — применение этого свойства к элементу `<body>`, таким образом вы заполните мозаикой всю страницу. Далее необходимо задать имя файла с изображением в виде `url('filename')`, как показано ниже.

```
body {
  background-image: url('stones.jpg');
}
```

Это свойство формирует мозаику из файла stones.jpg на всей странице, создавая ее фон.

Работая над мозаичным фоном, имейте в виду следующее.

- Делайте ваш фон светлым, чтобы текст, располагающийся поверх фона, легко читался. (Если вам действительно нужен темный фон, можно применить белый текст с жирным начертанием, чтобы он выделялся на темном фоне. Но избегайте этого, если только не создаете Web-сайт для новой ультрамодной группы или не открываете интернет-магазин одежды в готическом стиле.)
- Задавайте цвет фона страницы такой же, как на изображении. Например, если у вас темная фоновая картинка с белым текстом на ней, сделайте цвет фона черным. В этом случае, если браузер не сможет загрузить фоновое изображение, посетители все же увидят текст.
- Применяйте маленькие элементы мозаики, чтобы сократить время, в течение которого посетители будут ждать вывода страницы.
- Если у изображения, формирующего мозаику, края сложной формы, добейтесь их выравнивания. Левый край должен быть продолжением правого, а верхний — продолжением нижнего. В противном случае, когда браузер создаст мозаику, вы увидите линии стыковки ее элементов.

> ***Совет***
>
> В Интернете множество стандартных фоновых изображений в виде звезд, голубых небес и облаков, матерчатых и каменистых текстур, огней, головокружительных геометрических орнаментов, рамок и т. д. Найти их можно, задав в строке поиска Google "фоновые изображения" или прямо отправившись на какой-либо из давно известных сайтов, специализирующихся на предоставлении загружаемых фоновых изображений, например, **www.grsites.com/textures** и **www.backgroundsarchive.com**.

Фоновые "водяные знаки"

Большинство Web-сайтов применяют мозаику для создания фоновых изображений, но это не единственная возможность. Можно взять одно изображение и поместить его в конкретное место на вашей странице. Представьте себе, например, шпионский сайт с фоном, на котором едва можно прочесть "Top Secret and Confidential" (Совершенно секретный и не подлежащий разглашению).

Бледный, состоящий из одного изображения фон называют "водяным знаком". (Имя связано с процессом нанесения просвечивающего знака на бумагу, пропитанную водой.) Для создания хорошего водяного знака используйте бледное и нена-

вязчивое изображение. Для вставки на страницу водяного знака применяется то же самое свойство `background-image`, о котором вы только что узнали. Но в набор придется добавить несколько дополнительных стилевых свойств (табл. 7.2). Сначала нужно применить свойство `background-repeat`, чтобы отказаться от формирования мозаики. В то же время имеет смысл использовать свойство `background-position`, чтобы выровнять вашу картинку относительно бокового края страницы или ее центра.

Таблица 7.2. *Свойства фонового изображения*

Свойство	Описание	Стандартные значения	Может наследоваться?
`background-image`	Файл с изображением, которое вы хотите использовать как фон вашей страницы	URL, указывающий на файл с изображением в виде `url('mypig.jpg')`	Нет[1]
`background-repeat`	Применяете вы или нет мозаичное расположение изображения для заполнения страницы, мозаику можно отключить полностью или в одном направлении (например, чтобы изображение повторялось по вертикали, но не по горизонтали)	`repeat`, `repeat-x`, `repeat-y`, `no-repeat`	Нет
`background-position`	Место, куда вы хотите поместить изображение. Применяйте, только если не используете мозаику	`top left`, `top center`, `top right`, `center left`, `center`, `center right`, `bottom left`, `bottom center`, `bottom right`	Нет
`background-attachment`	Если хотите зафиксировать изображение (или мозаику) на прежнем месте при прокрутке страницы посетителем	`scroll`, `fixed`	Нет

[1] Фоновые изображения не наследуются (см. *разд. "Наследование" главы 6*). Но если явно не задать для элемента цвет фона, у него будет прозрачный фон, т. е. будет виден фон содержащего изображение элемента.

Далее приведен пример, в котором изображение помещается в центр Web-страницы.

```
body {
  background-image: url('smiley.jpg');
  background-repeat: no-repeat;
  background-position: center;
}
```

Примечание

Центр вашего документа — не обязательно центр окна браузера. Если поместить изображение в центр длинной Web-страницы, вы не увидите его, пока не прокрутите страницу.

Вы также можете отключить способность изображения перемещаться вместе с остальной страницей, чтобы создать необычный эффект: изображение, застывшее на месте (рис. 7.13). Например, примените следующий стиль для создания фонового изображения, которое помещается точно в центре окна.

```
body {
  background-image: url('smiley.gif');
  background-repeat: no-repeat;
  background-position: center;
  background-attachment: fixed;
}
```

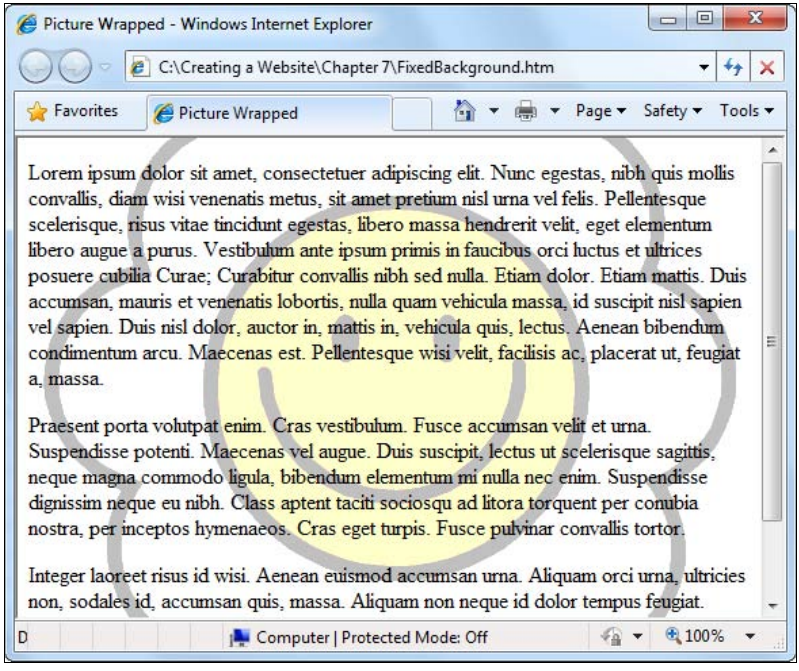


Рис. 7.13. Бросающая в глаза улыбающаяся физиономия всегда остается в центре окна, даже когда вы прокручиваете страницу вверх или вниз. Это слегка пугает

Приемы работы с графикой

Теперь, когда вы освоили элемент <img>, самое время узнать несколько профессиональных секретов. В следующих разделах вы познакомитесь с тремя популярными приемами, которые Web-разработчики используют везде для получения более совершенных изображений.

Графический текст

В *главе 6* вы узнали, что применение экзотических шрифтов на Web-страницах может создавать проблемы. Вы должны не только думать о лицензиях, но и создать несколько версий каждой гарнитуры, чтобы быть уверенным в корректном отображении необычного шрифта в разных браузерах.

С этой трудностью справиться невозможно, если у вас большие блоки текста. Но предприимчивые Web-художники иногда помещают небольшие фрагменты текста,

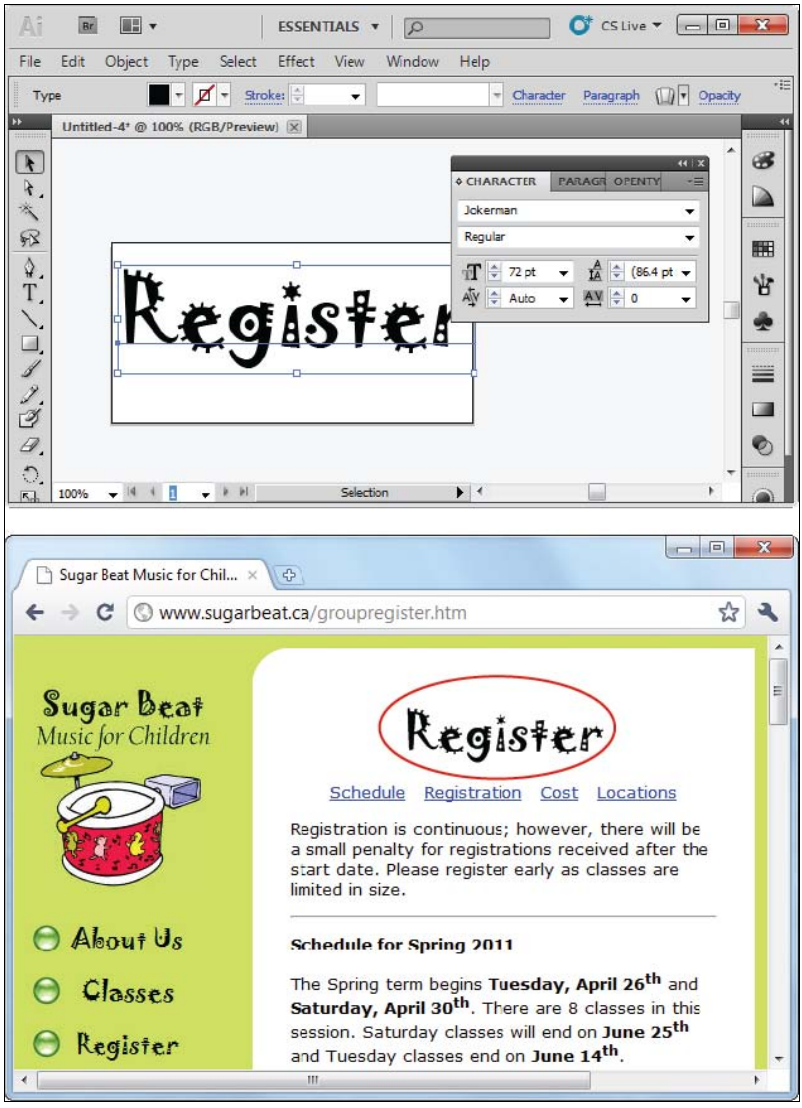


Рис. 7.14. *Вверху*: в Adobe Illustrator делается последний штрих в однословном заголовке. *Внизу*: заключительная картинка в том виде, в каком она отображается на Web-странице. Процесс создания графического текста может быть трудоемким, особенно если нужно создать много заголовков. Но это самый надежный способ включения причудливых шрифтов в Web-пространство

необходимые для заголовков, кнопок и логотипов в графические файлы. В этом случае можно полностью контролировать внешний вид текста и жизнь существенно упрощается.

Далее приводится несколько общих рекомендаций, касающихся создания небольших изображений, содержащих текст.

1. Запустите ваш любимый графический редактор или программу для рисования.

   На рис. 7.14 показан пример с использованием программы Adobe Illustrator.

2. Примените цвет фона, совпадающий с фоном вашей Web-страницы.

   В некоторых программах легче всего заполнить какую-нибудь область цветом, если нарисовать контур (например, прямоугольник) и задать для него нужный цвет заливки.

3. Выберите свой шрифт и введите текст поверх цвета фона.

4. Сократите размер изображения.

   В идеале вы должны получить изображение минимально возможного размера, но без обрезки текста.

5. Сохраните свое изображение.

   Подойдет формат GIF, но если в вашем изображении используется более 256 цветов, нужен PNG. Формат JPEG не используйте: у вашего текста будут размытые края.

Графический текст совершенно не подходит для больших объемов. Например, он не может выравниваться в соответствии с шириной окна браузера или учитывать стандартные настройки браузера у посетителя вашего сайта. Посетитель не сможет с помощью поиска найти в таком тексте конкретные слова, программа считывания — интерпретировать текст, а поисковая машина — определить, чему посвящен ваш сайт. Если решили использовать графический текст, обязательно повторите его в атрибуте `alt`, чтобы снять остроту описанных проблем.

Фон для других элементов

Не следует задавать фон для всей страницы. Можно привязать фон к одному абзацу или еще лучше к элементу `<div>`, создав подобие боковой полосы в журнале. Обычно хочется вставить рамку вокруг такого элемента, чтобы отделить его от остальной части Web-страницы. Также может появиться желание изменить текст на переднем плане, чтобы он стал отчетливее (например, белый лучше выделяется на темном фоне, чем черный).

Далее приведен пример фонового изображения, которое можно использовать с любым элементом-контейнером.

```
.pie {
  background-image: url('pie.jpg');
  margin-top: 20px;
  margin-bottom: 10px;
```

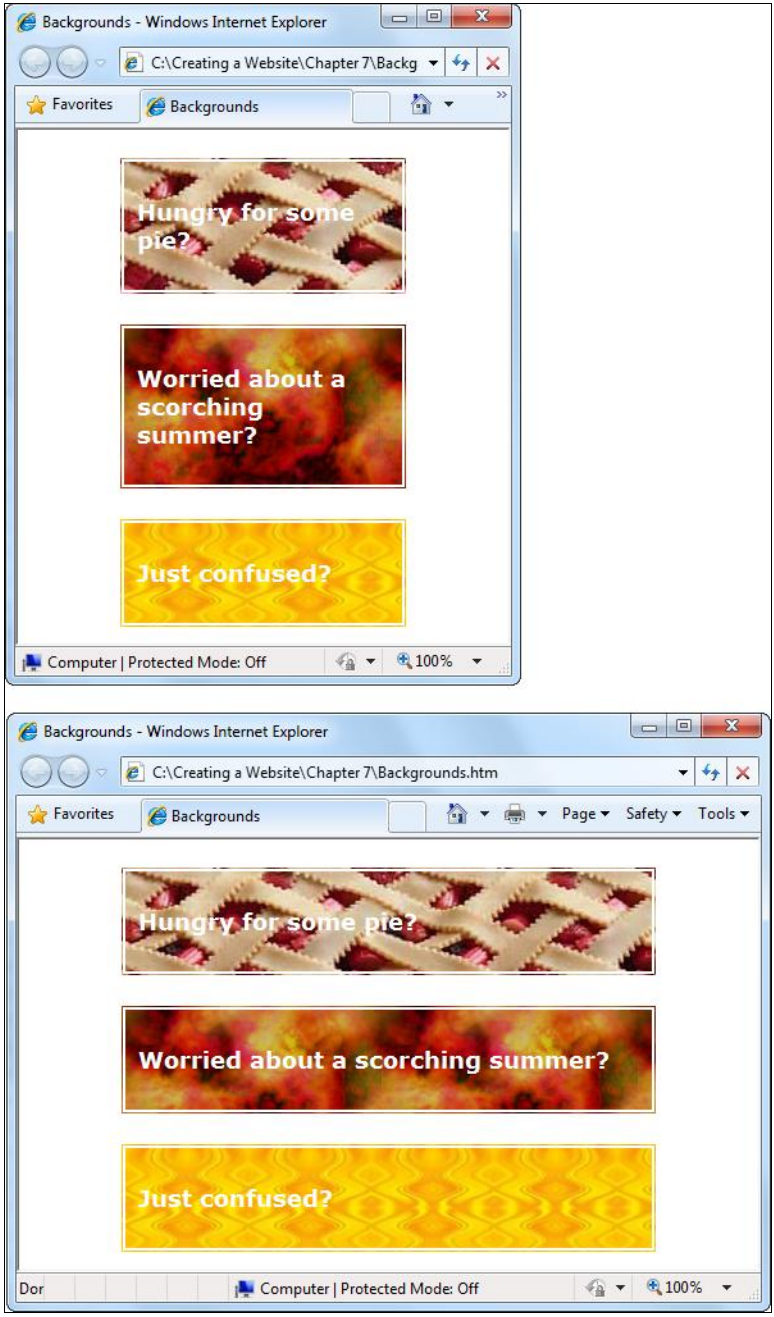


Рис. 7.15. *Вверху*: использование фоновых изображений в маленьких прямоугольниках на удивление превосходно.
Внизу: при изменении размера страницы особенно привлекательно увеличение фоновой картинки благодаря мозаичному отображению

```
  margin-left: 70px;
  margin-right: 70px;
  padding: 10px;
  border-style: double;
  border-width: 3px;
  color: white;
  background-color: black;
  font-size: large;
  font-weight: bold;
  font-family: Verdana,sans-serif;
}
```

В этом стиле задается фоновое изображение, устанавливаются поля и границы и выбираются согласующиеся цвета фона и переднего плана.

Далее приведен элемент `<div>`, использующий данный стиль.

```
<div class="pie">
  <p>Hungry for some pie?</p>
</div>
```

На рис. 7.15 показан результат.

Графические маркеры в списке

В *главе 5* вы узнали, как применять элемент `<ul>` для создания маркированного списка. Но вы были ограничены маленьким набором стандартных стилей маркеров. Если же заглянуть в Интернет, можно найти более интересные примеры маркированных списков, включая списки с пользовательскими маркерами в виде крошечных картинок.

С помощью элемента `<img>` вы можете вставить свои маркеры вручную, но есть более удобный вариант. Для установки изображения в качестве маркера можно применить свойство `list-style-image`. Далее приведен пример, в котором используется картинка, названная 3Dball.gif.

```
ul {
  list-style-image: url('3Dball.gif');
}
```

После того как вы создали это стилевое правило и поместили его в свою таблицу стилей, ваш браузер автоматически применит его к стандартному маркированному списку, например, следующему.

```
<ul>
  <li>Are hard to miss</li>
  <li>Help compensate for feelings of inadequacy</li>
  <li>Look so darned cool</li>
  <li>Remind people of boring PowerPoint presentations</li>
  </ul>
```

На рис. 7.16 показан результат.

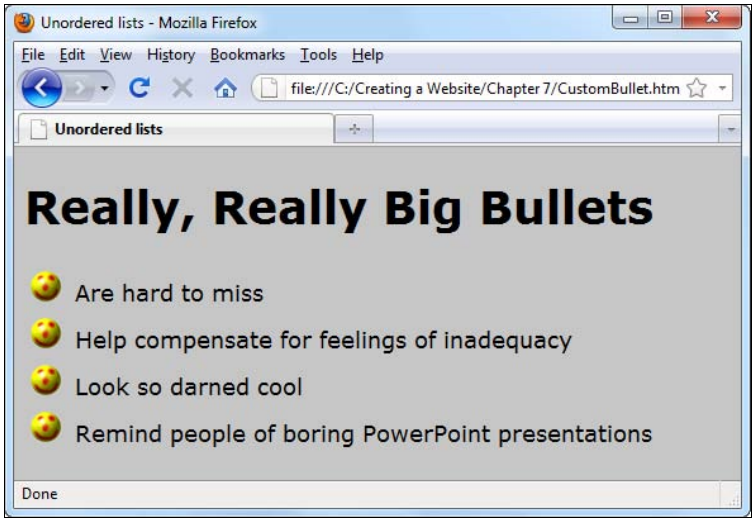


Рис. 7.16. Графическими маркерами могут быть как простые стрелки и флажки, так и экстравагантные трехмерные пятнистые мячи

Поиск бесплатной графики

Web-пространство завалено графикой. Действительно найти Web-страницу, не наполненную до краев изображениями, так же трудно, как увидеть Билла Гейтса в магазине "Все по $1". Но как сгенерировать все картинки, требующиеся для графически насыщенного сайта? Неужели придется сидеть часами в программе для рисования, вылизывая каждое нужное изображение? Ответ зависит от того, какие картинки вам нужны, но вас, конечно, обрадует сообщение о том, что Web — огромный источник уже готовых к использованию изображений.

Найти картинки в Web нетрудно. Можно даже применить удобное средство Google для поиска графики конкретной тематики (введите в вашем браузере **http://images.google.com** и отправляйтесь в путь). К сожалению, обнаружение картинки еще далеко не все. Если хотите ее применять и избежать гнусного юридического преследования, на использование картинки нужны *права*. Если вам повезет, владелец Web-сайта может дать разрешение на использование графики после того, как вы отправите короткое электронное письмо. Но это скорее исключение, чем правило.

К счастью, фотоэнтузиасты создали сайты, где они публикуют собственные картинки для всеобщего обозрения, и на некоторых из них можно найти и бесплатно использовать все, что захотите. Один из лучших — Stock.XCHNG (произносится как stock exchange (фондовая биржа), здесь обыгрывается словосочетание stock photography, название огромных каталогов фотографий для многократного использования, которые собирают графические дизайнеры). Для посещения Stock.XCHNG введите адрес **http://sxc.hu**. На рис. 7.17 показан процесс поиска на Stock.XCHNG.

Другое подходящее место для поиска бесплатных фотографий — Flickr (**www.flickr.com**). Но Flickr включает как доступные для повторного использования изо-

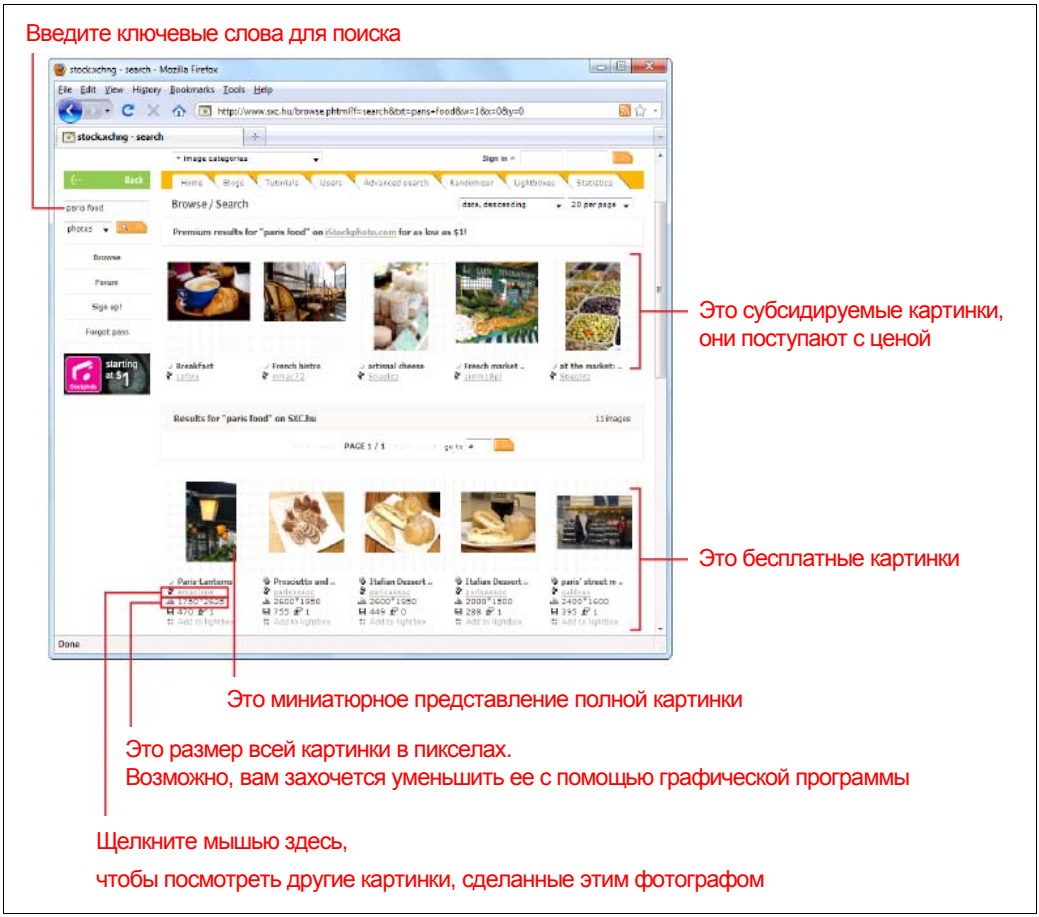


Рис. 7.17. Stock.XCHNG предлагает каталог с возможностью поиска, содержащий более 100 000 фотографий на любую тему. Ежедневно увлеченные фотоэнтузиасты выкладывают свою порой замечательную работу (включая некоторые рисунки из этой книги). На рисунке поиск "paris food" (парижская еда) привел к выводу на экран нескольких кулинарных шедевров

бражения, так и нет. Для поиска бесплатных картинок воспользуйтесь расширенным поиском на странице **www.flickr.com/search/advanced**. Введите ключевые слова поиска (как в обычном поиске), но установите флажок **Only search within Creative Commons-licensed content** (Поиск только в контенте с лицензиями Creative Commons) (рис. 7.18). В этом случае будут найдены только те файлы, которые можно стащить.

Имейте в виду, что контент с лицензией Creative Commons может применяться с одной или двумя оговорками. Например, может ставиться условие, не применять картинку на коммерческом Web-сайте (по крайней мере без получения разрешения от владельца изображения). Лицензия может также запрещать изменение или редактирование изображения. Когда найдете картинку, которая понравится, можно подробно ознакомиться с лицензией, переместившись с помощью прокрутки в поле **License** (Лицензионное соглашение) (которое находится в правом нижнем углу

страницы). Можно также выполнить поиск картинок, лишенных этих ограничений, установив дополнительные флажки, показанные на рис. 7.18.

Если вы не нашли нужного изображения на Stock.XCHNG или Flickr, возможно, вы не найдете его никогда — по крайней мере без похода на коммерческий сайт вроде iStockPhoto (**www.istockphoto.com**), Fotolia (**www.fotolia.com**) или Dreamstime (**www.dreamstime.com**). Все они просят по несколько долларов за изображения, не требующие авторских выплат. Если хотите еще поискать бесплатные ресурсы, посмотрите статью о поиске бесплатных фотографий по адресу **http://tinyurl.com/49yquv3**.

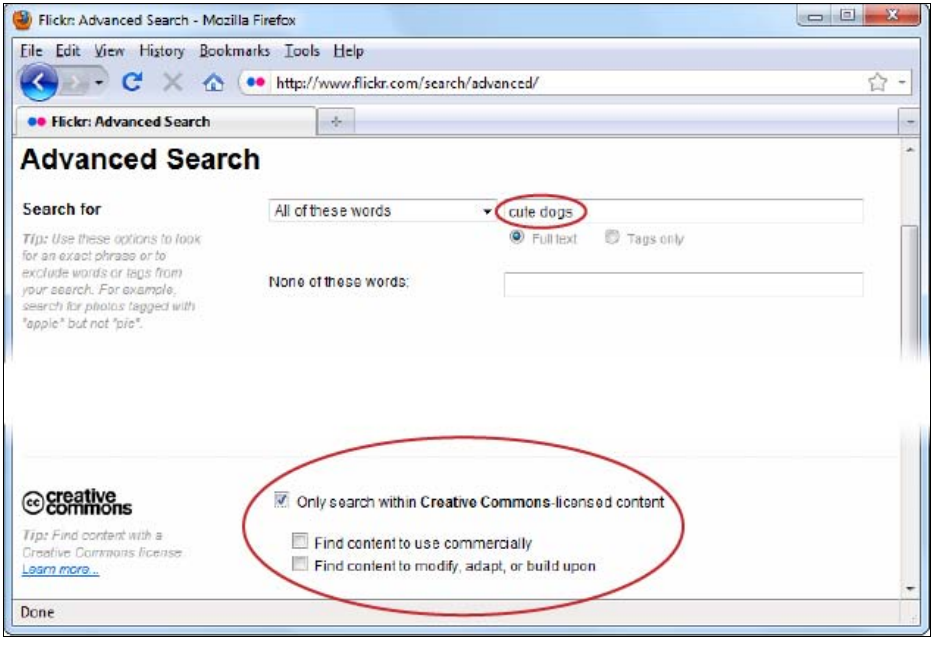


Рис. 7.18. Поиск на сайте Flickr ограничен картинками, которые вы можете использовать на собственном Web-сайте. Но у найденных картинок все равно могут остаться некоторые ограничения, запрещающие их использование на коммерческих сайтах или их модификацию

Совет

Огромное количество так называемых "бесплатных" сайтов с наборами графических или иллюстративных заготовок (clip art) забиты рекламой и навязываемыми подписками. Но если у вас есть пакет Microsoft Office, можно загрузить бесплатную библиотеку картинок с Web-сайта Office Online. Начните поиск со страницы **http://office.microsoft.com/images**. После выбора картинки, возможно, вам придется преобразовать ее в Web-формат (JPEG или PNG), но эта небольшая плата за доступ к большому и бесплатному источнику графики.

ГЛАВА 8

Связывание страниц

До сих пор вы работали с отдельными Web-страницами. Несмотря на то что создание единичной страницы — важный первый шаг в построении сайта, рано или поздно вы захотите связать вместе несколько страниц, чтобы Web-путешественник мог легко перепрыгивать с одной страницы на другую. Помимо всего прочего связывание — это и есть суть Web-пространства.

Устанавливать между страницами связи, формально называемые *гиперссылками*, поразительно легко. Действительно, для этого нужен всего лишь один новый элемент: *anchor* (точка привязки или якорь). Как только вы освоите эту частицу языка HTML, можно начинать собирать ваши страницы в отдельные папки и преобразовывать непритязательную коллекцию разрозненных документов в полноценный сайт.

Представление о якоре или точке привязки

В языке HTML для создания ссылки применяется элемент `<a>`, якорь или точка привязки. Когда посетитель щелкает кнопкой мыши эту ссылку, браузер загружает другую страницу.

Якорь — понятный и простой элемент-контейнер. Выглядит он следующим образом.

```
<a>...</a>
```

Текст, который пользователь щелкает кнопкой мыши, помещается внутрь элемента-якоря.

```
<a>Click Me</a>
```

Проблема в том, что приведенная ссылка не указывает никуда. Для превращения ее в полнофункциональную ссылку необходимо с помощью атрибута `href` (сокращение от *hypertext reference* (гипертекстовая ссылка)) задать URL страницы назначения. Например, если нужна ссылка, переносящая читателя на страницу LinkedPage.htm, создайте следующую ссылку.

```
<a href="LinkedPage.htm">Click Me</a>
```

Для того чтобы ссылка сработала, файл LinkedPage.htm следует поместить в ту же папку, что и страницу, содержащую ссылку **Click Me** (Щелкни меня). В *разд. "Относительные ссылки и папки" данной главы* вы прочтете, как лучше организовать ваш сайт, рассортировав страницы в разные вложенные папки или подпапки.

> ***Совет***
>
> Для создания ссылки на страницу в редакторе Expression Web выделите текст, который хотите сделать текстом ссылки, и затем нажмите комбинацию клавиш <Ctrl>+<K>. Укажите нужную страницу, и Expression Web создаст ссылку. Для повторения этого приема в редакторе Dreamweaver выделите текст и нажмите комбинацию клавиш <Ctrl>+<L>.

Тег якоря — строчный элемент (см. *разд. "HTML-элементы для основного текста" главы 5*), если он помещен внутрь другого блочного элемента. Это означает, что вполне допустимо создать ссылку всего из нескольких слов в обычном абзаце, например, таком.

```
<p>
  When you're alone and life is making you lonely<br />
  You can always go <a href="Downtown.htm">downtown</a>
</p>
```

На рис. 8.1 показана эта ссылка в действии.

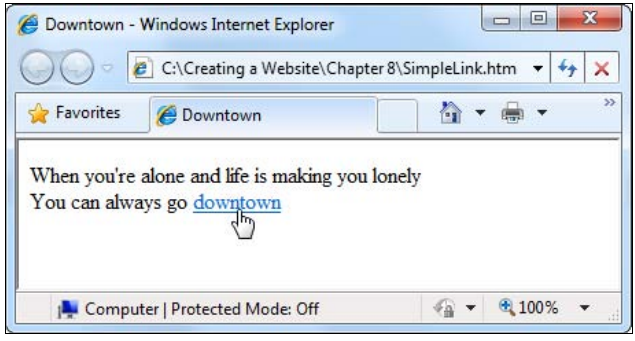


Рис. 8.1. Если больше не предпринимать никаких действий для настройки якоря, в браузере текст отобразится синими буквами с подчеркиванием. Когда вы наведете указатель мыши на гиперссылку, указатель превратится в кисть руки. Глядя на ссылку, нельзя сказать, действующая она или нет — если ссылка указывает на несуществующую страницу, сообщение об ошибке появится на экране только после того, как вы щелкните ссылку кнопкой мыши

Стоит отметить, что текст ссылки — текст внутри элемента <a> — важен. Как вы узнаете в *главе 11*, поисковые машины обращают на него особое внимание. Правильный выбор текста увеличит вероятность попадания вашего сайта в результаты поиска и привлечения новых посетителей. Далее приведено несколько примеров хорошего информативного текста ссылки: "Products" (Товары), "Register" (Регистрация), "Our Policies" (Наша политика) и "Contact Me" (Контакты). Бесполезный текст ссылки обычно состоит из одного-двух малозначащих слов в предложении, например, "click" (щелкни), "here" (здесь), "more" (еще) или "this link" (эта ссылка).

Внутренние и внешние ссылки

Ссылки могут переместить вас с одной страницы на другую в пределах одного Web-сайта или перенести на совершенно другой сайт на удаленном сервере. В каждом случае применяется определенный тип ссылки.

- *Внутренние ссылки* указывают на другие страницы вашего сайта. Как вы увидите позже, они также могут указывать на другие виды ресурсов на вашем сайте.
- *Внешние ссылки* указывают на страницы (или ресурсы) на других Web-сайтах.

Скажем, на вашем сайте есть два файла, страница с биографией и адресная страница. Если вы хотите отправить посетителей с вашей биографической страницы (MyBio.htm) на вашу адресную страницу ((ContactMe.htm), создайте внутреннюю ссылку. Независимо от того, находятся эти файлы в одной папке или в разных, они являются частью одного и того же Web-сайта на одном и том же Web-сервере, поэтому следует выбрать внутреннюю ссылку.

С другой стороны, если вы хотите посетителей вашей страницы Favorite Books (Любимые книги) (FavBooks.htm) отправить на страницу интернет-магазина Amazon.com (**www.amazon.com**), нужна внешняя ссылка. Щелчок внешней ссылки переносит читателя с вашего Web-сайта на новый сайт, расположенный где-то в Web-пространстве.

При создании внутренней ссылки всегда используйте *относительный URL*, сообщающий браузерам местонахождение нужной страницы *относительно текущей папки*. Другими словами, он инструктирует браузеры, как найти новую папку, задавая перемещение из текущей папки внутрь или наружу. (Перемещение *внутрь* означает переход из текущей папки во вложенную папку. Перемещение из текущей папки *наружу* — движение в обратном направлении, т. е. вы переходите из вложенной папки в родительскую папку, содержащую текущую папку как вложенную.)

Во всех приводимых ранее примерах использовались относительные URL. Допустим, вы отправляетесь на следующую страницу:

```
http://www.GothicGardenCenter.com/Sales/Products.htm
```

Текст на странице Products.htm включает предложение со следующей относительной ссылкой на Flowers.htm.

```
Would you like to learn more about our purple
<a href="Flowers.htm">hydrangeas</a>?
```

Если эту ссылку щелкнуть кнопкой мыши, браузер автоматически предположит, что файл Flowers.htm хранится там же, где файл Products.htm, и за кадром заполнит остальную часть URL. Это означает, что на самом деле браузер запросит следующую страницу:

```
http://www.GothicGardenCenter.com/Sales/Flowers.htm
```

В языке HTML есть и другой вариант связывания, именуемый *абсолютным URL*, задающий URL полностью, включая имя домена, папки и страницы. Если преобра-

зовать приведенный ранее URL в абсолютный, он будет выглядеть следующим образом.

```
Would you like to learn more about our purple <a href=
"http://www.GothicGardenCenter.com/Sales/Flowers.htm">hydrangeas</a>?
```

Какой же вариант выбрать? Решение принять просто. Существует только два правила, которые нужно запомнить.

- Если создается внешняя ссылка, вы *должны* использовать абсолютный URL. В этом случае относительный URL не сработает. Предположим, что вы хотите сослаться на страницу home.html на Web-сайте Amazon. Если создать относительную ссылку, браузер предположит, что вы ссылаетесь на файл с указанным именем, размещенный на вашем Web-сайте. Щелчок мышью ссылки не перенесет ваших посетителей туда, куда вы хотели их отправить (а может и вообще никуда не перенести, если на вашем сайте нет файла с именем home.html).
- Если создается внутренняя ссылка, вы всегда должны использовать относительный URL. Строго говоря, на внутренних страницах сработает любой тип ссылки. Но у относительных URL есть ряд преимуществ. Во-первых, они короче и делают ваш HTML-код более понятным. Еще важнее то, что относительные ссылки гибче. Вы можете реорганизовать ваш Web-сайт, поместить все ваши файлы в другую папку или даже изменить имя домена вашего сайта и при этом не испортить относительные ссылки.

Одна из самых чудесных особенностей относительных ссылок в том, что вы можете тестировать их на собственном компьютере, и они будут действовать так же, как в сети. Представьте себе, что вы разработали на своем компьютере сайт **www.GothicGardenCenter.com** и храните его в папке C:\MyWebsite (или папке HD/MyWebsite на компьютере Mac). Если щелкнуть кнопкой мыши относительную ссылку, направляющую со страницы Products.htm на страницу Flowers.htm, браузер будет искать заданную страницу в папке C:\MyWebsite (или HD/MyWebsite на компьютере Mac).

Когда доведете свою работу до совершенства, выгрузите сайт на ваш Web-сервер, у которого имя домена **www.GothicGardenCenter.com**. Поскольку применялись относительные ссылки, при переносе ваших страниц на сервер вам не придется изменять ни одну из них. При щелчке мышью ссылки браузер запросит соответствующую страницу на **www.GothicGardenCenter.com**. Если вы решите купить более короткое имя домена, например www.GGC.com, и перенести туда ваш сайт, ссылки все равно будут работать.

> ***ПРИМЕЧАНИЕ***
>
> У Internet Explorer есть одна особенность системы безопасности, которая проявится при тестировании страниц с внешними ссылками. Если вы загрузите страницу со своего жесткого диска и затем щелкните кнопкой мыши ссылку, указывающую куда-то в огромном Web-пространстве, для отображения указанной страницы Internet Explorer откроет совершенно новое окно. Объясняется это тем, что правила безопасности, управляющие Web-страницами на вашем жестком диске, мягче правил, налагающих ограничения на Web-страницы из Интернета, поэтому Internet Explorer не разрешит им быть рядом друг с другом. Эта проблема исчезнет после выгрузки ваших страниц в Web.

ЧАСТО ЗАДАВАЕМЫЙ ВОПРОС

Открытие нового окна

Как создать ссылку, открывающую страницу в новом окне браузера?

Естественно вам хочется помешать посетителям, щелкающим кнопкой мыши внешние ссылки, так легко покинуть ваш сайт. У Web-разработчиков есть часто используемый прием, позволяющий открывать внешние страницы в отдельном окне браузера (или на новой вкладке при соответствующих настройках браузера). В этом случае ваш сайт остается открытым в исходном окне посетителя и не дает ему забыть о вас.

Для реализации этого приема необходимо задать в элементе-якоре еще один атрибут, `target`, следующим образом.

```
<a href="LinkedPage.htm" target="_blank"> Click Me</a>
```

Атрибут `target` содержит имя фрейма, в котором браузер должен отобразить заданную страницу (дополнительную информацию о фреймах вы узнаете в *главе 10*). Значение `_blank` означает загрузку страницы из данной ссылки в новое пустое окно браузера.

Прежде чем вы начнете вставлять атрибут `target` во все якори, важно понять, что он не всегда срабатывает. Некоторые бдительные *блокировщики всплывающих окон* перехватывают ссылки этого типа и не дают вывести на экран новое окно. (Блокировщики всплывающих окон — это автономные программы или специально разработанные средства браузеров, препятствующие отображению на экране надоедливых всплывающих рекламных объявлений.) У Internet Explorer 6 (и последующих версий) есть собственный блокировщик всплывающих окон, но его стандартные настройки пропускают ссылки с атрибутом `target="_blank"`.

Одни пользователи любят средство отображения в отдельном окне, а другие считают его актом чрезвычайно назойливого и разрушительного вмешательства в Web-сайт. Если вы пользуетесь им, применяйте его экономно, лишь к некоторым ссылкам.

Относительные ссылки и папки

До этого момента в примерах с относительными ссылками, которые вы видели, предполагалось, что *исходная страница* (содержащая ссылку) и *конечная страница* (то место, куда вы попадаете после щелчка ссылки кнопкой мыши) хранятся в одной и той же папке. Нет особой причины строго соблюдать это ограничение. В действительности ваш Web-сайт будет в целом лучше организован, если хранить группы связанных страниц в отдельных папках.

Рассмотрим Web-сайт, показанный на рис. 8.2.

ПРИМЕЧАНИЕ

Корневая папка — отправная точка вашего Web-сайта, в ней содержатся все остальные файлы и папки сайта. У большинства сайтов в корневой папке есть страница с именем index.htm или index.html. Ее также называют *исходной* или *домашней страницей*. Если браузер отправляет запрос к домену с вашим Web-сайтом без указания имени файла, сервер посылает назад домашнюю страницу. Например, в ответ на запрос **www.TripToRemember.com** автоматически возвращается исходная страница **www.TripToRemember.com/index.htm**.

На показанном сайте используются самые разные относительные ссылки. Предположим, что вам нужно создать ссылку со страницы index.htm на страницу

contact.htm. Обе страницы хранятся в одной папке, поэтому нужна только относительная ссылка.

```
<a href="contact.htm">About Me</a>
```

Можно создавать и более интересные ссылки, переходящие из одной папки в другую. Об этом вы узнаете в следующих разделах.

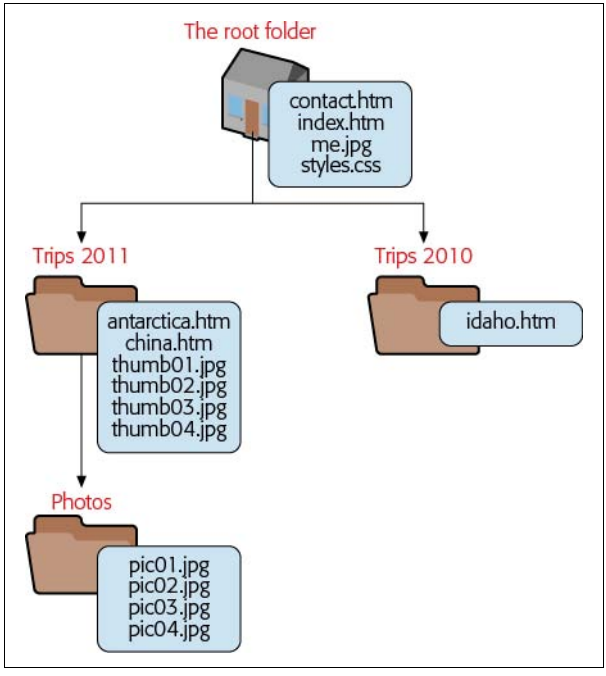


Рис. 8.2. На этой схеме показана структура очень маленького Web-сайта, отображающего на экране фотографии, сделанные в поездке. Корневая папка содержит таблицу стилей, применяемую ко всему сайту (styles.css), и две HTML-страницы. В двух вложенных папках, Trips2010 и Trips2011, содержатся дополнительные страницы. В папке Trips2011 хранятся миниатюры фотографий, сделанных в одной из поездок. Каждой миниатюре соответствует полноразмерная фотография во вложенной папке Photos

Совет

Если вы хотите опробовать этот пример, все файлы сайта можно найти на странице **Missing CD** (Отсутствующий CD) по адресу **www.missingmanuals.com/cds/caw3**. Благодаря магии относительных ссылок независимо от того, куда вы поместите файлы на вашем компьютере, все ссылки будут работать, но лишь до тех пор, пока сохранятся неизменными вложенные папки.

Переход вниз, в подпапку

Допустим, что вы хотите создать относительную ссылку, переходящую со страницы index.htm на страницу antarctica.htm, которая находится в папке с именем Trips2011. Когда пишется относительная ссылка, указывающая на antarctica.htm, необходимо в нее включить имя вложенной папки Trips2011 следующим образом.

```
See pictures from <a href="Trips2011/antarctica.htm">Antarctica</a>
```

Эта ссылка содержит две инструкции браузеру: сначала перейти в подпапку Trips2011, а затем получить страницу antarctica.htm. В ссылке имя папки (Trips2011) отделяется от имени файла (antarctica.htm) символом прямого слэша (/). На рис. 8.3 показаны обе участницы этого процесса.

Интересно, что относительные пути можно использовать и в других HTML-элементах, например `<style>` и `<img>`. Представим себе, что вы хотите отобразить

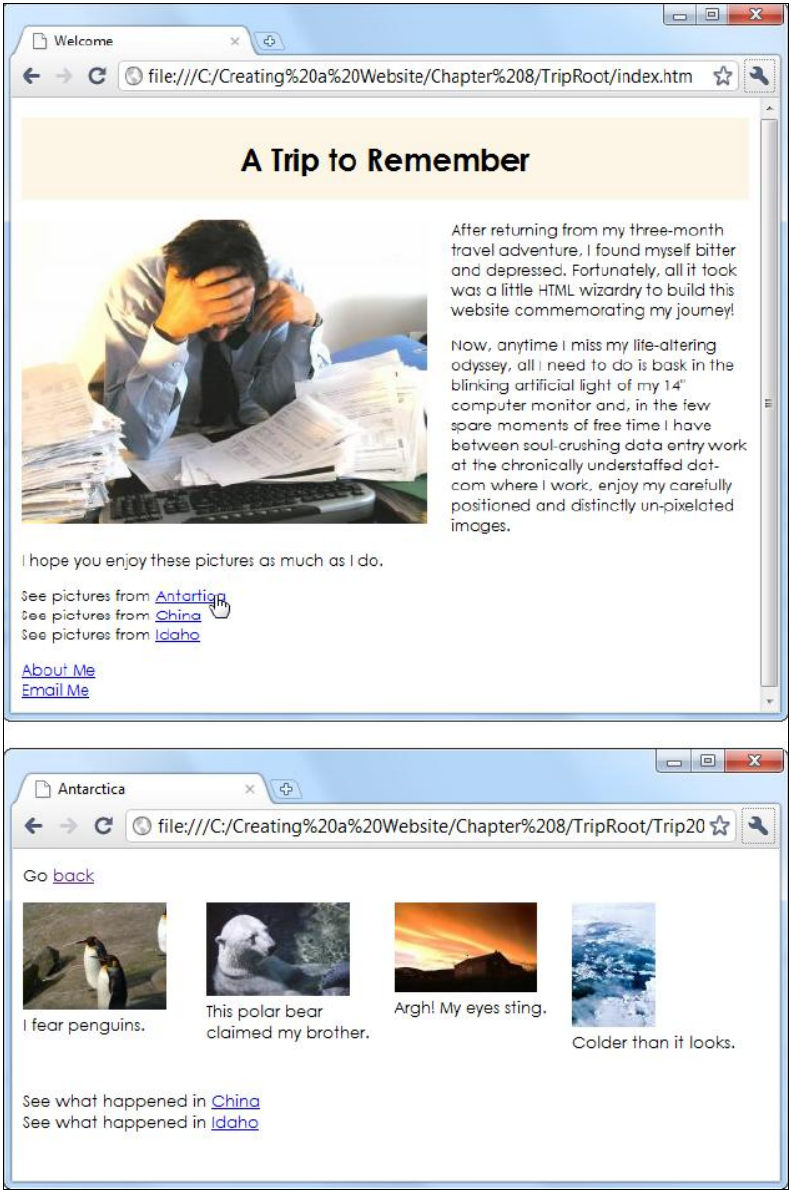


Рис. 8.3. С помощью относительной ссылки можно перейти с главной страницы index.htm (вверху) на страницу с миниатюрами картинок (внизу). Каждая фотография — это ссылка, которую для отображения увеличенной копии фотографии посетитель щелкает кнопкой мыши

фотографию photo01.jpg на странице index.htm. Эта картинка находится на расстоянии двух подпапок, в папке с именем Photos, хранящейся внутри папки Trips2011. Но это не помешает вам сослаться на нее в элементе <img>.

```
<img src="Trips2011/Photos/photo01.jpg" alt="A polar bear" />
```

С помощью этого приема можно забраться глубже в подпапки, находящиеся в подпапках, которые в свою очередь размещены в подпапках. Нужно всего лишь по очереди вставлять для каждой подпапки ее имя и символ слэша.

Только помните, что относительные ссылки всегда задаются *относительно текущей страницы*. Если вы хотите отобразить ту же фотографию photo01.jpg на странице antarctica.htm, приведенный элемент <img> не будет работать, потому что страница antarctica.htm в действительности находится в папке Trips2011. (Вернитесь к рис. 8.2, если хотите освежить в памяти структуру сайта.) Из папки Trips2011 нужен единственный переход на уровень вниз, поэтому нужна следующая ссылка.

```
<img src="Photos/photo01.jpg" alt="A polar bear" />
```

Теперь вы, наверное, поняли, что важно не количество папок на вашем сайте, а то, как организованы подпапки. Еще раз напоминаю: относительная ссылка всегда берет начало в текущей папке и затем переходит вверх или вниз к папке, содержащей заданную (или конечную) страницу.

> ***Совет***
>
> После того как подпапки появились в относительных ссылках, не следует менять их имена или перемещать их, поскольку вы можете испортить ссылку. Тем не менее многие Web-редакторы (например, Expression Web) достаточно сообразительны и могут исправить ваши относительные ссылки. Это еще один повод подумать о приобретении полнофункционального редактора Web-страниц.

Переход вверх, в родительскую папку

Следующая проблема, с которой вы столкнетесь, — переход на уровень вверх в дереве папок. Для этого используется символьная последовательность ../ (две точки и слэш). Например, для вставки на страницу antarctica.htm ссылки, возвращающей читателя к странице index.htm, пишется следующая ссылка.

```
Go <a href="../index.htm">back</a>
```

И как вы уже, наверное, догадались, для перехода на два уровня вверх можно применить эту команду дважды. Например, если у вас есть страница в папке Photos, которая отсылает на домашнюю страницу, для того чтобы туда вернуться, понадобится следующая ссылка.

```
Go <a href="../../index.htm">back</a>
```

В более сложных случаях можно комбинировать оба эти приема для создания относительной ссылки, которая переходит на один или несколько уровней вверх, а затем спускается вниз другим путем. Например, такой вид ссылки понадобится для перехода со страницы antarctica.htm, находящейся в папке Trips2011, на страницу idaho.htm из папки Trips2010.

```
See what happened in <a href="../Trips2010/idaho.htm">Idaho</a>
```

Эта ссылка переходит на один уровень вверх, в корневую папку, а затем спускается на один уровень в папку Trips2010. Когда вы просматриваете папки в поисках файлов, вы повторяете этот алгоритм.

Переход в корневую папку

Единственная проблема относительных ссылок, с которой вы пока столкнулись, — их трудно сохранить корректными при любой реорганизации вашего сайта. Представим себе, что у вас есть Web-страница в корневом каталоге вашего сайта. Вы хотите отобразить на этой странице картинку, хранящуюся в подпапке images. Вы применяете следующую ссылку.

```
<img src="images/flower.gif" alt="A flower" />
```

Но затем, спустя какое-то время, вы решили, что страница на самом деле относится к подпапке Plants, поэтому переместили ее туда. Но относительная ссылка теперь указывает на несуществующий файл plant/images/flower.gif, ведь папка images не подпапка в папке Plants, она вложена в корневую папку вашего сайта. В итоге браузер выведет пиктограмму испорченной или нерабочей ссылки.

Есть два возможных решения. В программах, подобных Expression Web, когда вы перетаскиваете с помощью мыши файл на новое место, HTML-редактор обновляет все относительные ссылки автоматически, оберегая вас от проблем. Второй вариант — попытаться сохранять связанные файлы в одной папке, так вы всегда будете перемещать их как единое целое. Но есть и третий подход — *ссылки относительно корня сайта*.

До сих пор вы видели *ссылки относительно документа*, поскольку задавали местоположение конечной страницы относительно текущего документа. *Ссылки относительно корня сайта* указывают местонахождение конечной страницы *относительно корневой папки вашего Web-сайта*.

Ссылки относительно корня сайта всегда начинаются с символа слэша (/) (который обозначает корневую папку). Далее приведен элемент `<img>` для файла flower.gif со ссылкой относительно корня сайта.

```
<img src="/images/flower.gif" alt="A flower" />
```

Самая замечательная особенность этой ссылки в том, что она работает независимо от местонахождения страницы, ее содержащей. Например, если скопировать эту страницу в подпапку Plants, ссылка останется рабочей, т. к. первый слэш заставляет браузер начать поиск с корневой папки сайта.

Единственная сложность применения ссылок относительно корня сайта — необходимость помнить реальный корневой каталог вашего Web-сайта. При использовании ссылки относительно корня сайта браузер следует простому алгоритму поиска пути. Сначала он из адреса текущей страницы изымает информацию о полном пути и имени файла, оставляя только имя домена. Затем вставляет ссылку относительно корня сайта в конец имени домена. Если ссылка на файл flower.gif появляется на следующей странице:

```
http://www.jumboplants.com/horticulture/plants/annuals.htm
```

браузер отбрасывает часть /horticulture/plants/annuals.htm, вставляет относительную ссылку, заданную в атрибуте `src` (/images/flower.gif), и ищет изображение в следующем месте:

```
http://www.jumboplants.com/images/flower.gif
```

Это совершенно логично. Но вот что происходит, когда у вас нет собственного имени домена. В такой ситуации ваши страницы, вероятно, собраны в подпапку на постороннем Web-сервере. Далее приведен пример.

```
http://www.superISP.com/~user9212/horticulture/plants/annuals.htm
```

Часть URL, включающая имя домена, — **www.superISP.com**, но фактически корень вашего Web-сайта — ваша персональная папка ~user9212. Это означает, что эту деталь следует включить во все ваши ссылки относительно корня сайта. В случае картинки flower.gif для получения желаемого результата вам нужно использовать более запутанную ссылку относительно корня сайта:

```
<img src="/~user9212/images/flower.gif" alt="A flower" />
```

Теперь браузер сохранит из URL только имя домена (**www.superISP.com**) и добавит относительную часть пути, начиная с вашей персональной папки (/~user9212).

НА ПРОФЕССИОНАЛЬНОМ УРОВНЕ

Правила для URL

Правила корректного написания URL в элементе-якоре очень строги и несколько распространенных ошибок проникает даже на самые лучшие Web-страницы. Далее приведены рекомендации, которые помогут избежать этих проблем.

- Когда создаете абсолютный URL, следует начинать с его протокола (обычно http://). Но это правило необязательно соблюдать при вводе URL в браузере. Например, если вы вводите **www.google.com**, большинство браузеров достаточно сообразительны для того, чтобы додумать http://. Но в HTML-документе это правило соблюдать обязательно.
- Не путайте обратный слэш (\) и обычный прямой слэш (/). В Windows обратный слэш используется в путях файлов (например, C:\Windows\win.ini). В Web-мире прямой слэш отделяет подпапки (как в **http://www.ebay.com/Help/index.html**). Многие браузеры, как уже говорилось, терпимо относятся к путанице слэшей, но такая ошибка в элементе-якоре делает ваши ссылки нерабочими.
- Никогда не используйте вместо URL файловые пути. С помощью протокола file можно создать URL, указывающий на файл на вашем компьютере (например, file:///C:/Temp/myPage.htm). Но подобная ссылка не будет работать на любом другом компьютере, потому что на его жестком диске нет этого файла. Порой программные средства редактирования, такие как Expression Web, могут вставить один из этих так называемых локальных URL (например, если вы перетащите файл с изображением на вашу страницу). Будьте бдительны — проверяйте все свои ссылки, чтобы убедиться, что этого не произошло.
- Не вставляйте в имена ваших файлов и папок пробелы или специальные символы, даже если это допустимые символы. Например, совершенно естественно вставить пробел в имя файла (такое как My Photos.htm), но для того чтобы запросить эту страницу, браузеру придется преобразовать пробел в специальный символьный код (My%20Photos.htm). Чтобы избежать подобной неприятности, не используйте ничего, кроме цифр, букв, дефисов (-) или подчеркиваний (_).

Изменения цвета ссылок и подчеркивания

В сущности все, рожденные после 1900 г., интуитивно понимают, что подчеркнутый текст синего цвета отображается на странице для того, чтобы его щелкнули кнопкой мыши. Но что если синие ссылки не вяжутся с общим внешним обликом вашего сайта? Благодаря таблицам стилей вам не придется играть по правилам, жестко определяющим цвет ссылок.

Основываясь на знаниях о CSS, полученных из *главы 6*, вы сможете быстро создать правило таблицы стилей, меняющее цвет текста всех формирующих ссылки тегов якорей на вашем сайте. Далее приведен пример.

```
a {
  color: fuchsia;
}
```

Но обратите внимание: пользовательские цвета ссылок меняют стандартное поведение ссылок. Обычно, когда вы щелкаете ссылку кнопкой мыши, она становится пурпурно-красной, чтобы показать, что вы посетили страницу. Настроенные пользователем ссылки никогда не меняют цвет — они сохраняют его даже после того, как вы щелкнули их мышью.

Лучше создавать полноцветные ссылки с помощью другого средства таблицы стилей: псевдоклассов. *Псевдоклассы* — специализированные версии CSS-классов, с которыми вы познакомились раньше (см. *разд. "Более мощные селекторы" главы 6*). Они основываются на деталях, которые браузер оставляет за кадром. Например, правила для обычных классов применяются без разбора к заданному элементу, такому как тег якоря. Правила для псевдоклассов применяются только к элементам, удовлетворяющим критериям, в данном случае к ссылкам, которые щелкали или не щелкали кнопкой мыши.

Форматировать ссылки помогают четыре псевдокласса. К ним относятся `:link`, для ссылок, указывающих на неизведанную землю; `:visited`, для ссылок, которые пользователь уже посетил; `:active`, для ссылок, изменяющих цвет при щелчке их мышью и до момента отпускания кнопки мыши, и `:hover`, для ссылок, изменяющих цвет при наведении на них указателя мыши. Как видите, имена псевдоклассов всегда начинаются с двоеточия (`:`).

Далее приведено стилевое правило с применением псевдоклассов для создания дезориентирующей страницы — на ней посещенные ссылки синие, а не посещенные — красные.

```
a:link {
  color: red;
}
a:visited {
  color: blue;
}
```

Если хотите применять это правило не ко всем, а лишь к некоторым ссылкам, можно добавить в ваше правило имя класса.

```
a.BackwardLink:link {
  color: red;
}
a.BackwardLink:visited {
  color: blue;
}
```

Теперь для отображения нового стиля в элементе-якоре потребуется задать имя класса, как показано далее.

```
<a class="BackwardLink" href="...">...</a>
```

И в заключение следует отметить, что аналогичное правило можно написать для свойства таблицы стилей `text-decoration`, изменяющего автоматическое подчеркивание ссылок браузером. Далее приведен пример, удаляющий стандартное подчеркивание.

```
a {
  text-decoration: none;
}
```

Это правило не подходит для ссылок, встроенных в основной контент страницы, т. к. затрудняет их обнаружение. Но оно полезно, если у вас есть панель, состоящая только из ссылок (например, боковое меню), и вы хотите сделать его внешний вид более четким.

Почтовые ссылки

Почтовая ссылка (mailto) — специальный вид ссылки, помогающий пользователям отправить сообщение для вас. При щелчке такой ссылки кнопкой мыши браузеры открывают вашу программу электронной почты и начинают создавать сообщение. Реальная отправка сообщения все равно зависит от вас, но почтовая ссылка запускает процесс создания сообщения со стандартными строкой темы письма и полем основного текста.

> ***Примечание***
>
> Почтовая ссылка не всегда срабатывает, если посетители применяют сервисы электронной почты на основе Web (как Hotmail и Gmail). Щелчок кнопкой мыши почтовой ссылки в сообщении может открыть настольную почтовую программу, которой они не пользуются, или даже привести к выводу сообщения об ошибке (в зависимости от применяемого браузера и настроек компьютера). Для решения этой проблемы или поиска обходного пути на вашем компьютере выполните поиск в Google, задав в строке поиска "ссылка mailto" и имя вашего браузера.

Для создания почтовой ссылки задайте путь, начинающийся со слова "mailto" с последующим двоеточием (:), и ваш адрес электронной почты. Далее приведен пример.

```
<a href="mailto:me@myplace.com">Email Me</a>
```

Большинство браузеров позволяют также задать текст в строке темы письма и в поле основного текста сообщения. Когда кто-нибудь щелкает почтовую ссылку

кнопкой мыши, эта информация загружается в новое письмо, готовое к отправке (или редактированию).

Для вставки текста в строку темы письма и в поле основного текста следует использовать странный синтаксис, отвечающий следующим правилам.

- Ставьте знак вопроса после адреса электронной почты.
- Для задания темы письма вставьте `subject=` и далее текст для строки темы.
- Если хотите указать какой-нибудь текст в основном поле письма, после текста в строке темы вставьте символьную последовательность `&`, а затем введите `body=` и нужный текст.
- Заменяйте символы, которые могут вызвать проблемы, специальными кодами. Буквы, цифры и точка годятся, а большая часть других знаков пунктуации — нет. Например, все пробелы в строке темы и в основном поле письма нужно заменить символьной последовательностью `%20`. Это довольно утомительно и затруднит чтение вашего письма в окончательном виде, но зато ваша почтовая ссылка будет работать во всех браузерах. Самый легкий способ подготовить текст письма — посетить страницу **http://meyerweb.com/eric/tools/dencoder**, на которой можно создать специальный код. Просто введите текст вашего письма в предлагаемое текстовое поле и затем щелкните мышью кнопку **Encode** (Кодировать), чтобы заменить потенциально проблемные коды подходящими.

Сбиты с толку? Легче всего усвоить эти правила, посмотрев пару примеров. Сначала приведена почтовая ссылка, содержащая в строке темы письма текст "Automatic Email".

```
<a href="mailto:me@myplace.com?subject=Automatic%20Email">
Email Me</a>
```

Далее приведена ссылка, содержащая текст и в строке темы, и в основном поле письма.

```
<a href=
"mailto:me@myplace.com?subject=Automatic%20Email&amp;body=
 I%20love%20your%20site.">Email Me</a>
```

После щелчка мышью такой ссылки может появиться предупреждающее сообщение, информирующее о том, что Web-страница собирается открыть вашу программу электронной почты, и запрашивающее у вас разрешение на это (точный текст сообщения зависит от вашего браузера и операционной системы). Если вы согласитесь, то увидите окно электронного письма, похожее на показанное на рис. 8.4.

НА ПРОФЕССИОНАЛЬНОМ УРОВНЕ

Связывание с другими видами контента

Большинство написанных вами ссылок будут указывать на настоящие HTML Web-страницы. Но это не единственная возможность. Вы можете также создавать ссылки на файлы других типов. Единственная сложность в том, что, если кто-то щелкнет мышью ссылку, указывающую на файл другого типа, как поступать, решает браузер.

Далее приведено несколько часто встречающихся примеров.

- Вы можете создать ссылку на файл изображения в формате JPEG, GIF или PNG. Когда посетители щелкают кнопкой мыши подобную ссылку, браузер отображает картинку в отдельном окне без какого-либо дополнительного контента. Web-сайты часто используют этот прием, чтобы дать возможность посетителям рассмотреть большое изображение в увеличенном масштабе. В примере из предыдущего раздела на Web-сайте Trips2011 есть страница с множеством изображений-миниатюр. Щелкните мышью любое из них, и на экране появится полноразмерное изображение.
- Вы можете создать ссылку на файл специального типа, такой как PDF-файл, документ Microsoft Office или аудиофайл в формате MP3. Вы рискуете, используя подобный прием. Такие ссылки полагаются на плагины браузера (маленькие программки, реализующие определенные задачи), распознающие тип файла, или на ваших посетителей, на компьютерах которых установлена подходящая программа. Если на машине нет подходящего программного обеспечения, ваши посетители смогут лишь загрузить такой файл (см. следующий пункт) на свой компьютер, где он будет лежать инертной двоичной глыбой. Но если у браузера есть нужный плагин, происходит маленькое волшебство, и файлы открываются прямо в окне браузера.
- Вы можете сослаться на файл, предназначенный для загрузки посетителями. Если ссылка указывает на файл специального типа, и у браузера нет подходящего плагина, у посетителей есть выбор: они могут игнорировать весь контент полностью, открыть его с помощью другой программы с их компьютера или сохранить его на своем компьютере. Это удобный способ распространения больших файлов (например, ZIP-файлов, описывающих ваше личное представление о планетарном движении).

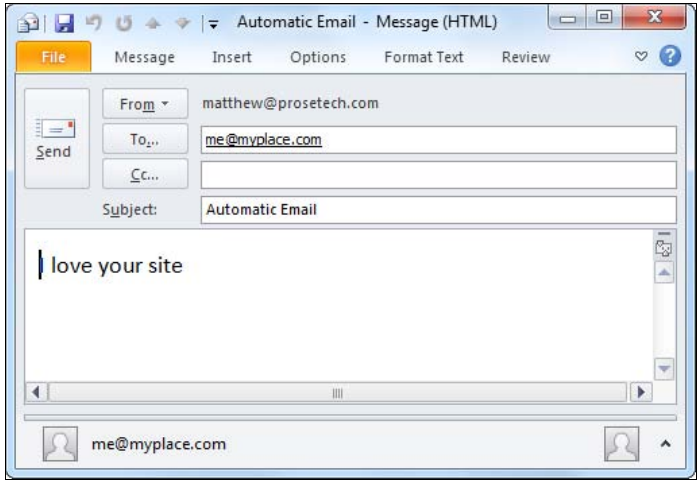


Рис. 8.4. Если вы щелкнете мышью почтовую ссылку, браузер создаст электронное письмо (как показано). В нем в соответствии с информацией, представленной в ссылке, заполнены поля получателя, темы и основного текста письма, но любой, щелкнувший мышью эту ссылку, может изменить эти детали (или закрыть окно, не нажимая кнопку **Send** (Отправить)). Реальное окно электронного письма может отличаться, т. к. его внешний вид зависит от программы электронной почты, установленной на компьютере посетителя Web-сайта. В данном примере показано окно письма из программы Microsoft Outlook

Изображения-ссылки и карты изображений

В ссылки можно вставлять и изображения. Нужно внутрь элемента `<a>` вставить элемент `<img>` следующим образом.

```
<a href="LinkedPage.htm"><img src="MyPic.gif" alt="My Picture" /></a>
```

HTML добавляет толстые синие границы вокруг изображений, показывая посетителям, что их можно щелкать кнопкой мыши. Как правило, вам захочется удалить эту уродливо выглядящую рамку с помощью свойств границ таблиц стилей, описанных в *разд. "Создание улучшенных границ" главы 6*. При наведении указателя мыши на картинку в ссылке указатель превращается в кисть руки.

> ***Примечание***
>
> В якорь можно вставить не только текст и изображения. В якорях можно разместить любой HTML-код, включая целые абзацы текста, маркированные списки и т. д. Если попробуете сделать это, то увидите, что весь текст внутри становится синим и подчеркнутым, а все изображения будут обведены синими рамками. Web-браузеры поддерживали это эксцентричное поведение годами, но только версия HTML5 превратила его в официальную часть стандарта.

Иногда необходимо создать *внутри* изображения конкретные, реагирующие на щелчок кнопки мыши области, называемые активными областями. Для примера рассмотрим рис. 8.5.

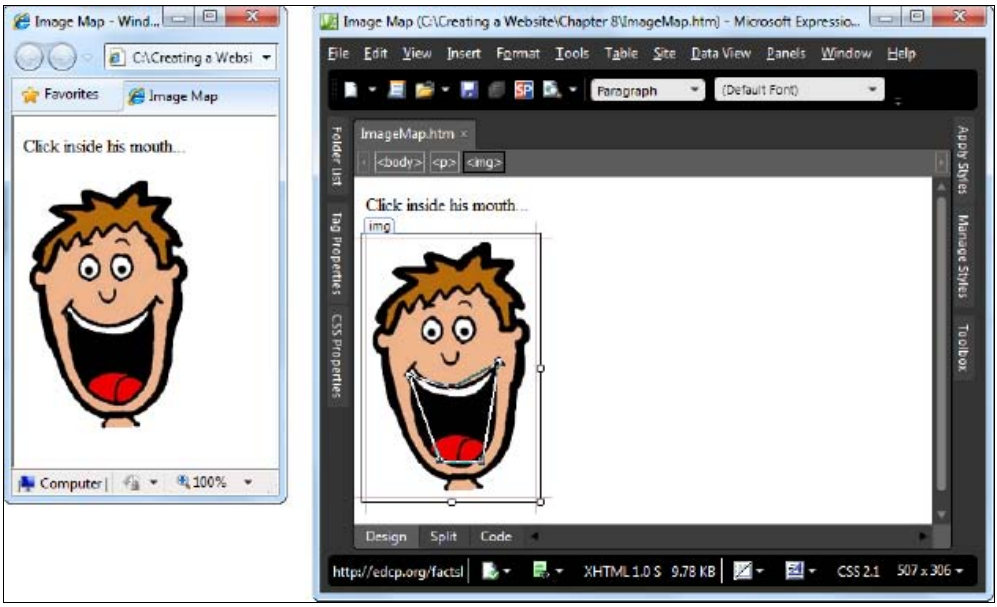


Рис. 8.5. *Слева*: обычная картинка, результат работы элемента `<img>`. *Справа*: область неправильной формы внутри открытого рта становится активной областью, реагирующей на щелчок кнопкой мыши и переносящей посетителей на другую страницу. В этом примере активная область видна, т. к. она редактируется в программе Expression Web. Обычно посетители при просмотре картинки в Web-браузере не видят контуры активных областей

Чтобы вставить в изображение активную область, необходимо создать *карту изображения* с помощью HTML-элемента <map>. Это сделать просто — нужно только выбрать уникальное имя для карты вашего изображения, чтобы вы впоследствии могли на нее ссылаться.

```
<map id="FaceMap" name="FaceMap">
</map>
```

> ***Примечание***
>
> Вы наверняка заметили, что элемент <map> использует атрибуты с одним и тем же значением (id и name). Хотя в теории должен работать только атрибут id, атрибут name тоже необходим, чтобы гарантировать совместимость с разными браузерами.

Далее нужно определить все активные области внутри элемента <map>, между его начальным и конечным тегами. Вы можете вставить сколько угодно активных областей, но они не должны накладываться друг на друга. (Если такое произойдет, у области, определенной первой, более высокий приоритет.)

Для определения активной области на изображении применяется элемент <area>. Этот элемент задает три важные параметра: конечную страницу, на которую переходит посетитель после щелчка мышью активной области (атрибут href), контур активной области (атрибут shape) и точные размеры контура (атрибут cords). Так же как и изображению, элементу <area> требуется атрибут alt с замещающим или альтернативным текстом, описывающим карту изображения для поисковых машин, программ считывания с экрана и древних текстовых браузеров.

Далее приведен пример элемента <area>.

```
<area href="MyPage.htm" shape="rect" coords="5,5,95,195"
alt="A clickable rectangle" />
```

Для этой активной области определен прямоугольный контур. Когда посетитель щелкает ее кнопкой мыши, он переходит на страницу MyPage.htm.

Атрибут shape поддерживает контуры трех видов, каждому из них соответствует свое значение атрибута. Вы можете применять окружности (circle), прямоугольники (rect) и многоугольники (poly). После выбора контура необходимо задать координаты, которые интерпретировать немного сложнее. Для того чтобы разобраться в координатах активной области, сначала нужно понять, как браузеры измеряют изображения (рис. 8.6).

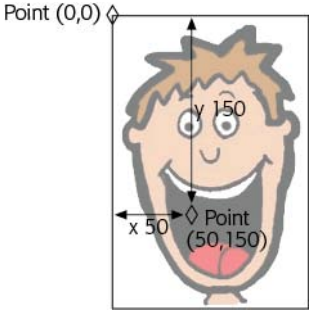


Рис. 8.6. Браузеры обозначают левый верхний угол изображения как точку (0, 0). При движении по изображению вниз растет координата *y* (второе число). Например, точка (0, 100) находится на левом краю изображения, на расстоянии 100 пикселов от верхнего края. При движении вдоль изображения вправо растет координата *x*. Следовательно, точка (100, 0) находится на верхнем краю изображения, на расстоянии 100 пикселов от левого края

Координаты карты изображения вводятся в виде списка чисел, отделенных запятыми. Для окружности координаты перечисляют в следующем порядке: точка центра (x-координата), точка центра (y-координата), радиус. Для любых других контуров указывайте по порядку углы как последовательность x-y координат, например, x1, y1, x2, y2 и т. д. В случае многоугольника указываются все углы. Для прямоугольника достаточно двух точек: левый верхний и правый нижний углы.

В приведенном примере прямоугольник задается следующими двумя точками: (5, 5) вверху слева и (95, 195) внизу справа. Многоугольник, более сложный, чем указанный на рис. 8.5, задается следующим образом.

```
<area href="MyPage.htm" shape="poly" title="Smiling Mouth" alt="Mouth"
coords="38, 122, 76, 132, 116, 110, 102, 198, 65, 197" />
```

Другими словами, ваш браузер создает этот контур, рисуя линии между следующими пятью точками: (38, 122), (76, 132), (116, 110), (102, 198) и (65, 197).

> ***ПРИМЕЧАНИЕ***
>
> Правильно задать координаты непросто. У многих редакторов Web-страниц, таких как Dreamweaver и Expression Web, есть встроенные редакторы активных областей, позволяющие создавать карту изображения, перетаскивая мышью контур по вашему изображению, что гораздо легче, чем стараться высчитать правильные координаты. Для применения этого инструмента в Dreamweaver выделите картинку и найдите на панели **Properties** (Свойства) три пиктограммы активных областей (круг, квадрат и многоугольник). Expression Web предлагает аналогичные пиктограммы на панели инструментов **Picture** (Изображение). (Если вы не видите панели **Picture** (Изображение), щелкните изображение правой кнопкой мыши и выберите команду **Show Pictures Toolbar** (Показать панель Изображение).)

После того как откорректированы все активные области, остается один шаг: применить активные области к изображению с помощью атрибута `usemap` в вашем элементе `<img>`. Используйте в этом атрибуте то же имя, что и в карте изображения, но введите перед ним символ решетки (#), чтобы сообщить браузеру об определении карты изображения для картинки.

```
<img src="face.gif" usemap="#FaceMap" alt="Smiling Face" />
```

Далее приведен полный HTML-код для примера с активной областью на месте открытого рта на картинке.

```
<!DOCTYPE html>

<html>

<head>
  <title>Image Map</title>
  <style type="text/css">
    img {
      border-style: none;
  }
  </style>
</head>
```

```
<body>
  <p>Click inside his mouth...</p>
  <p>
    <map id="FaceMap" name="FaceMap">
      <area href="http://edcp.org/factsheets/handfoot.html" shape="poly"
      coords="38, 122, 76, 132, 116, 110, 102, 198, 65, 197"
      alt="Smiling Mouth" />
    </map>
    <img src="face.gif" usemap="#FaceMap" alt="Smiling Face" />
  </p>
</body>

</html>
```

Активные области, созданные вами, невидимы (если вы не нарисуете линии на картинке, чтобы показать их местоположение). Когда посетитель перемещает указатель мыши по ним, указатель превращается в кисть руки. Щелчок мышью активной области аналогичен щелчку мышью обычной ссылки `<a>`: ваш посетитель попадает на другую страницу.

> ***Совет***
>
> Очень заманчиво применить карты изображений для создания ссылок в любых видах графики, включая кнопки, которые можно самостоятельно спроектировать в графическом редакторе. Отложите это ненадолго. Тщательно разработанные Web-сайты, подобные вашему, могут продвинуться далеко вперед с помощью меню и кнопок, но для реализации этих остроумных затей вам понадобится ноу-хау языка JavaScript, о котором вы узнаете в *главах 15, 16*.

Вставка закладок

Большинство ссылок приводят с одной страницы на другую. Когда вы перепрыгиваете на новую страницу, браузер плюхает вас в самое ее начало. Но можно создать ссылки на конкретные части страницы. Это особенно полезно, если вы создаете длинные прокручиваемые страницы и хотите обратить внимание ваших посетителей на конкретный фрагмент.

Вы можете формировать ссылки на другую часть *текущей* страницы (рис. 8.7) или на определенное место на *другой* Web-странице. Место, куда вы отправляете своего читателя, формально называется *фрагментом*.

Создание ссылки на фрагмент — двухшаговый процесс. Сначала нужно обозначить фрагмент. Делается это с помощью атрибута `id`, который присваивает уникальное имя любому HTML-элементу на странице.

Предположим, что вы хотите отправить посетителя к третьему по счету заголовку уровня 3 на Web-странице с именем sales.htm. Первоначально разметка выглядит следующим образом:

```
...
<h3>Pet Canaries</h3>
<p>Pet canary sales have plummeted in the developed world, due in large part to
currency fluctuations and other macroeconomic forces.</p>
...
```

Далее приведено изменение, присваивающее заголовку Pet Canaries (Домашние канарейки) уникальное имя (в данном случае имя `Canaries`).

```
...
<h3 id="Canaries">Pet Canaries</h3>
<p>Pet canary sales have plummeted in the developed world, due in large part to
currency fluctuations and other macroeconomic forces.</p>
...
```

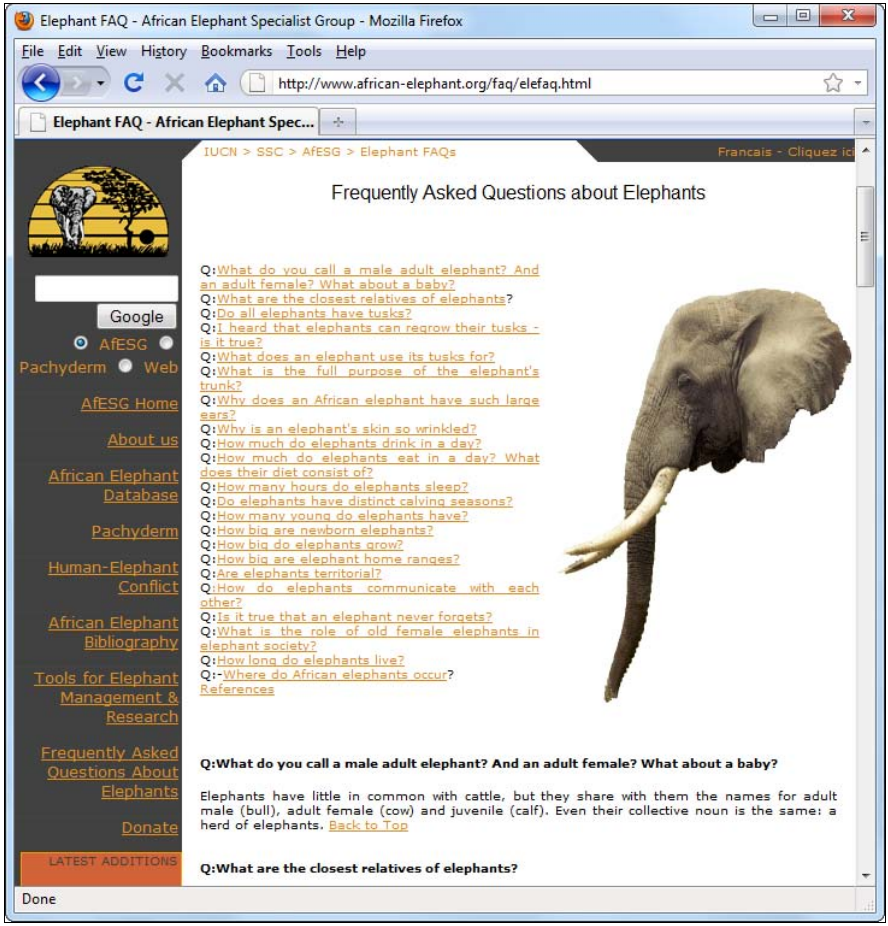


Рис. 8.7. Это страница с часто задаваемыми вопросами (FAQ) — пример рабочих закладок. Здесь весь раздел представляет собой одну длинную страницу с серией ссылок-закладок в начале, которые переносят вас сразу к интересующей теме. Можно разбить эту страницу на несколько, но тогда читатели не смогут просмотреть весь набор вопросов и не смогут сразу напечатать весь документ

Оно никак не повлияет на внешний вид страницы; посетители никогда не увидят атрибут `id`. Тем не менее он обеспечит удобный способ перемещения посетителей к заголовку Pet Canaries. По сути вы создали закладку, зафиксированную на этом заголовке.

После того как закладка создана, можно написать URL, указывающий на нее. Для этого данные о закладке вставляются в конец URL страницы. Сначала добавляется знак решетки (`#`), а затем имя закладки.

Далее приведена ссылка, отправляющая читателя к закладке с именем Canaries на странице sales.htm.

```
Learn about recent developments in <a href="sales.htm#Canaries">canary
sales</a>.
```

Когда вы щелкаете эту ссылку кнопкой мыши, браузер переводит вас на страницу sales.htm и прокручивает ее до тех пор, пока не обнаружит закладку `Canaries`. Затем браузер отображает в своем окне часть страницы, начинающуюся с заголовка Pet Canaries.

> ***Совет***
>
> Если закладка расположена близко к концу страницы, браузер не сможет в своем окне вывести страницу, начиная с этой закладки. Вместо этого помеченный закладкой раздел появится где-нибудь в середине окна браузера. Это происходит потому, что браузер наталкивается на конец страницы и не может прокрутить ее дальше. Если вы считаете, что такая ситуация может вызвать путаницу (возможно, у вас есть несколько помеченных закладками разделов, расположенных рядом друг с другом и близко к концу страницы), добавьте несколько элементов `<br>` в конец документа, что позволит браузеру прокрутить страницу чуть дальше.

Порой возникает желание создать ссылку, указывающую на закладку на вашей *текущей* странице. В этом случае не нужно задавать имя страницы. Просто вставьте знак решетки, а следом за ним имя закладки.

```
Jump to the <a href="#Canaries">canary</a> section.
```

Умелое применение закладок — это искусство. Не разрешайте себе набивать страницы ссылками, направляющими читателей к относительно мелким разделам контента. Закладки используйте только на монотонных и длинных страницах, занимающих при полной прокрутке несколько экранов.

Когда хорошие ссылки портятся

Теперь, когда вы узнали все способы построения ссылок, самое время узнать, что может пойти не так. Ссылки, ведущие к страницам на одном и том же сайте, могут испортиться, когда вы переименовываете или перемещаете файлы или папки. Ссылки на другие Web-сайты особенно недолговечны; они могут сломаться в любой момент и без предупреждения. Вы не сможете узнать, что что-то испортилось, пока не щелкните ссылку кнопкой мыши и не получите сообщение об ошибке: "Page Not Found" (Страница не найдена).

Испорченные ссылки так часто встречаются, что Web-разработчики придумали термин для описания постепенной потери Web-сайтами своих связующих способностей: потеря или разложение ссылок (link rot). К сожалению, вы можете выгрузить отлично работающий Web-сайт, и, вернувшись на него через несколько месяцев, обнаружить, что все его внешние ссылки потеряны. Они указывают на Web-сайты, которые больше не существуют, перенесены или реорганизованы.

Потеря ссылок — коварная проблема, поскольку снижает доверие посетителей к вашему сайту. Они видят страницу, обещающую привести их к другим интересным ресурсам, но когда они щелкают ссылку кнопкой мыши, их ждет разочарование. Опытные посетители не будут долго оставаться на сайте, страдающем от запущенной болезни, потери ссылок, они решат, что вы ни разу не обновляли свой сайт и уйдут на более крутой Web-сайт где-нибудь еще.

Как же снизить остроту проблемы потери ссылок? Во-первых, вы должны неукоснительно проверять все ваши внутренние ссылки, указывающие на страницы вашего Web-сайта. Проверяйте мельчайшие ошибки, способные испортить работающую ссылку, и проходите все пути, хотя бы по разу. В самых современных редакторах Web-страниц есть встроенные средства, автоматизирующие эту тяжелую и нудную процедуру.

Внешние ссылки создают другую проблему. Вы не можете написать железобетонную ссылку, поскольку конечные страницы вне зоны вашего контроля и могут измениться в любой момент. Можно, конечно, сократить количество внешних ссылок на вашем Web-сайте и тем самым минимизировать проблему, но вряд такое решение можно считать удовлетворительным. Важнейший компонент прелести Web — возможность одним щелчком мыши перейти от подробной дискографии рока к мемориальному сайту с канцелярскими принадлежностями, разрисованными рукой Элвиса. Если вы хотите связать ваш Web-сайт с остальным миром, без внешних ссылок не обойтись. Лучше всего регулярно тестировать ваш сайт с помощью валидатора ссылок, который проходит по всем страницам сайта и проверяет каждую ссылку, чтобы удостовериться, что она все еще куда-то ведет.

В следующих разделах дается краткое представление об управлении Web-сайтом и о работе валидаторов ссылок.

Управление Web-сайтом

Dreamweaver, Expression Web и многие другие редакторы Web-страниц содержат средства управления Web-сайтом, позволяющие окинуть взглядом весь Web-сайт целиком. В большинстве случаев, для того чтобы воспользоваться этими средствами, необходимо особым образом описать сайт (см. *разд. "Управление Web-сайтом" главы 4*). После этого вы сможете с высоты птичьего полета увидеть все, что включает в себя ваш Web-сайт (рис. 8.8).

Чаще всего просмотр содержимого папок ваших Web-сайтов не так интересен, как изучение паутины ссылок, связывающей ваши страницы в единое целое. Многие редакторы Web-страниц позволяют увидеть, куда ведут все ваши ссылки (рис. 8.9).

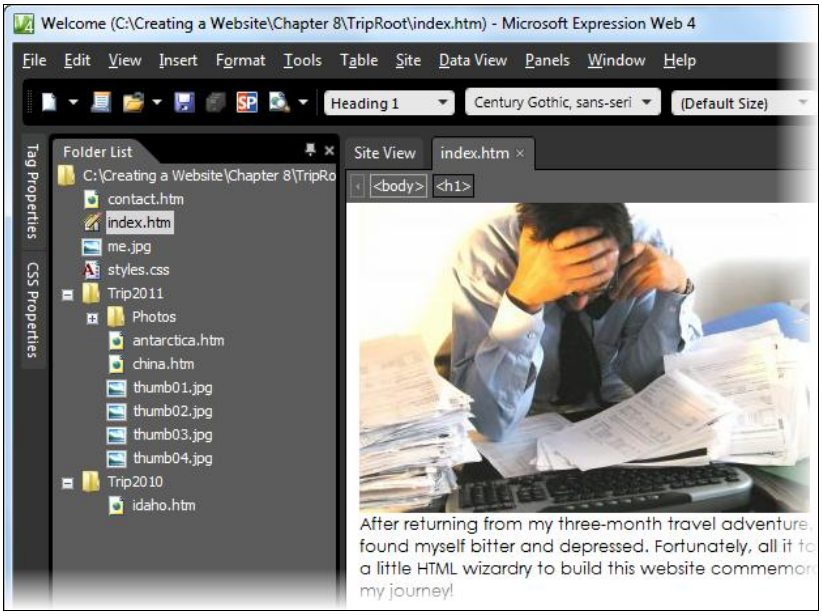


Рис. 8.8. В Expression Web вы можете открыть отдельную страницу для редактирования с помощью команд **File | Open Site** (Файл | Открыть сайт) (просто выберите папку верхнего уровня для вашего Web-сайта). В **Folder List** (Список папки) можно переупорядочить и переименовать файлы, а Expression Web автоматически обновит все связанные ссылки

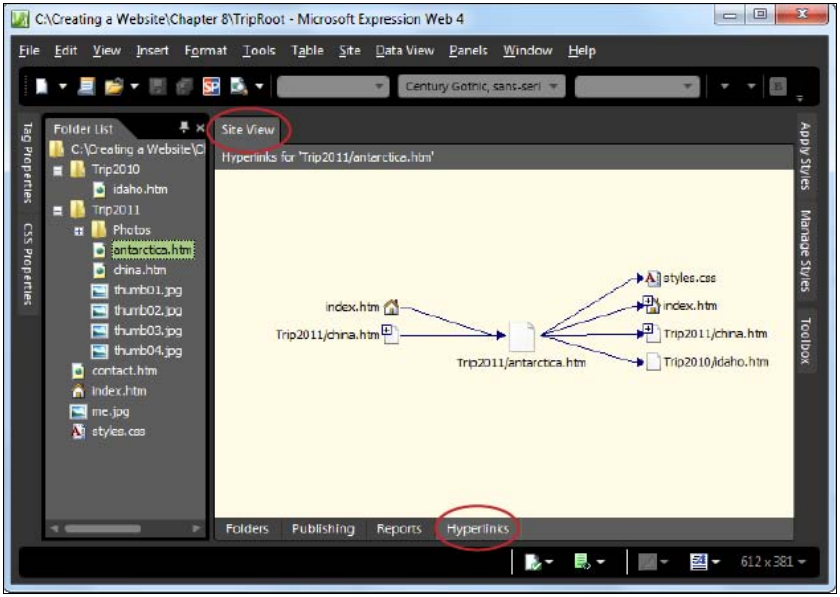


Рис. 8.9. В Expression Web перейдите на вкладку **Site View** (Вид сайта). Щелкнув мышью кнопку **Hyperlinks** (Гиперссылки), расположенную в нижней части, вы увидите, как выделенная в данный момент страница вписывается в Web-сайт. В данном примере текущая страница — antarctica.htm. Стрелки, направленные от страницы antarctica.htm, представляют ссылки на этой странице, ведущие на другие страницы. Стрелки, направленные к странице antarctica.htm, представляют ссылки на других страницах, ведущие на страницу antarctica.htm. Если щелкнуть кнопкой мыши поля со знаком +, расположенные рядом с другими страницами сайта, можно также увидеть все ссылки на этих страницах

ПРИМЕЧАНИЕ

Обозреватель гиперссылок в редакторе Expression Web — одно из средств, нуждающихся в скрытых папках метаданных (см. *разд. "Задание сайта в Expression Web" главы 4*). Если вы открыли обозреватель гиперссылок и выделили файл, но не увидели никаких ссылок, возможно, вы еще не добавили папки метаданных. Для их получения выберите команды **Site | Site Settings** (Сайт | Параметры сайта), установите флажок **Maintain the website using hidden metadata files** (Сопровождение Web-сайта с использованием скрытых файлов метаданных) и щелкните мышью кнопку **OK**.

Средства проверки ссылок

Средство проверки ссылок — инструмент автоматического просмотра одной или нескольких ваших Web-страниц. Оно тестирует каждую найденную ссылку, пытаясь получить конечную Web-страницу (страницу, на которую указывает ссылка). В зависимости от валидатора и вида выполняемой проверки средства проверки ссылок могут проверять только внутренние ссылки или следовать за каждой ссылкой на каждой странице до тех пор, пока не будут протестированы все ссылки на вашем сайте.

В состав таких программ, как Dreamweaver и Expression Web, входят сложные инструменты проверки ссылок, которые служат отличным подспорьем для прерывания всего сайта и обнаружения проблем. В Dreamweaver для проверки ссылок используйте команды **Site | Check Links Sitewide** (Сайт | Проверка ссылок по всему сайту). В Expression Web аналогичный инструмент применяется с помощью команд **Site | Reports | Problems | Hyperlinks** (Сайт | Отчеты | Проблемы | Гиперссылки).

Средства проверки ссылок, встроенные в эти два редактора Web-страниц, работают с копией вашего Web-сайта, хранящейся на вашем компьютере. Это лучший способ отслеживания ошибок во время разработки сайта, но они не помогут, когда ваш Web-сайт вырвется на волю. Например, они не смогут найти такие ошибки, как ссылка на файл на вашем жестком диске или на файл, который вы забыли выгрузить на Web-сервер.

Для того чтобы сделать окончательный вывод о ссылках вашего Web-сайта, возможно, придется воспользоваться бесплатной сетевой программой проверки ссылок. Организация World Wide Web Consortium предлагает отличный вариант проверки на странице **http://validator.w3.org/checklink**. Для запуска сетевой бесплатной программы проверки ссылок выполните следующие действия.

1. Перейдите на страницу **http://validator.w3.org/checklink**.

   Это приведет вас к утилите W3C Link Checker.

2. В текстовое поле введите полный URL страницы, которую хотите проверить.

   Если у вашего Web-сайта есть исходная страница, например index.htm, можно ввести только имя домена без явного указания имени файла.

3. Задайте нужные параметры (рис. 8.10).

   - Установите флажок **Summary only** (Только краткая сводка), если хотите пропустить подробный перечень действий валидатора во время проверки стра-

ниц. Лучше этот флажок не устанавливать, так вы легче поймете, какие именно страницы проверяет валидатор ссылок.

- Установите флажок **Hide redirects** (Скрыть перенаправления), если хотите, чтобы программа проверки игнорировала команды, перенаправляющие ее на Web-страницу, отличающуюся от заданной конечной страницы (дополнительную информацию см. в *разд. "Использование перенаправлений" данной главы*). Обычно перенаправления показывают, что ваша ссылка все еще работает, но вы должны обновить ее, указав новую конечную страницу.
- Флажок **Don't send the Accept-Language header** (Не отправлять заголовок о предпочитаемом языке) запрещает программе проверки сообщать Web-сайту его языковые настройки. Этот переключатель имеет значение, только если вы создаете многоязычный Web-сайт, но это тема не обсуждается в книге.

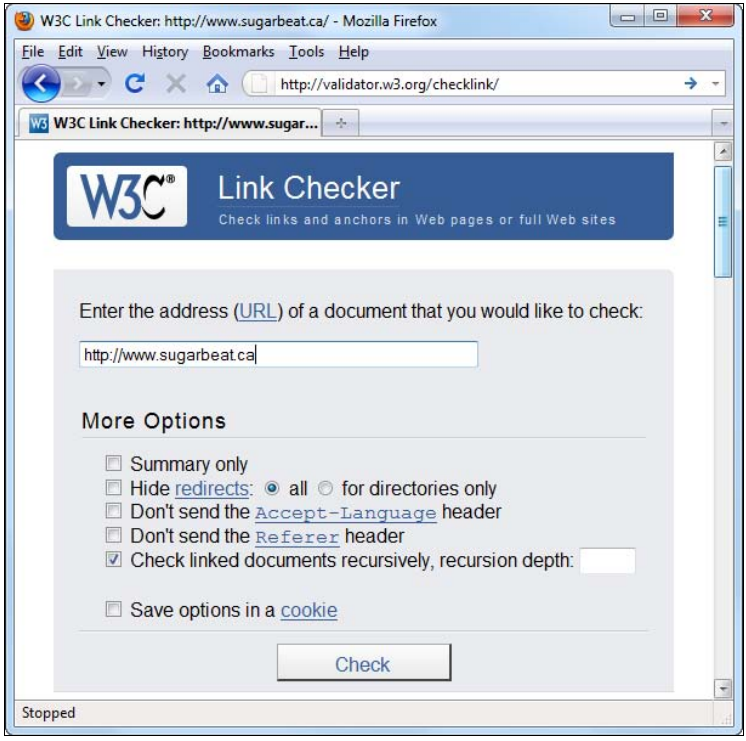


Рис. 8.10. При использовании программы проверки ссылок выберите проверяемую Web-страницу и вариант проверки с рекурсией, как в данном примере, или без. (Дополнительные сведения о рекурсивной проверке и принципах ее работы см. в п. 3 на предыдущей странице.) Далее для запуска проверки щелкните мышью кнопку **Check** (Проверить)

- Установка флажка **Check linked documents recursively** (Проверять связанные документы рекурсивно) позволяет проверять ссылки с помощью рекурсии. Если не устанавливать этот флажок, валидатор просто проверяет каждую ссылку на заданной вами странице и подтверждает, что она указывает на действующую Web-страницу. В случае рекурсивной проверки валидатор прове-

ряет все ссылки на текущей странице и затем каждую внутреннюю ссылку на вашем сайте. Например, если ссылка указывает на страницу с именем info.htm, валидатор сначала проверяет существование info.htm. Затем он ищет все внутренние ссылки на странице info.htm и начинает *их* тестирование. Собственно, если info.htm ссылается только на еще одну внутреннюю страницу (например, contact.htm), программа проверки ссылок переходит на эту страницу и начинает проверять также и *ее* ссылки. Программа проверки достаточно умна и не проверяет одну и ту же страницу дважды, таким образом не тратя время на повторную проверку уже проверенных ссылок.

Примечание

Программы проверки ссылок не применяют рекурсию к внешним ссылкам. Другими словами, если запустить программу проверки ваших ссылок на исходной странице вашего Web-сайта, она пройдет все ссылки на любую другую страницу вашего сайта, но не пойдет дальше. И все же рекурсия — прекрасный способ проверки всех ссылок на вашем сайте за один проход.

Если хотите ограничить рекурсию (допустим, у вас много страниц и нет нужды проверять все), можно задать глубину рекурсии, параметр, указывающий максимальное число уровней, на которые углубляется программа проверки. Например, если глубина рекурсии равна 1, программа проследит только первый набор встреченных ею ссылок. Если не задавать глубину рекурсии, программа проверки проверит все.

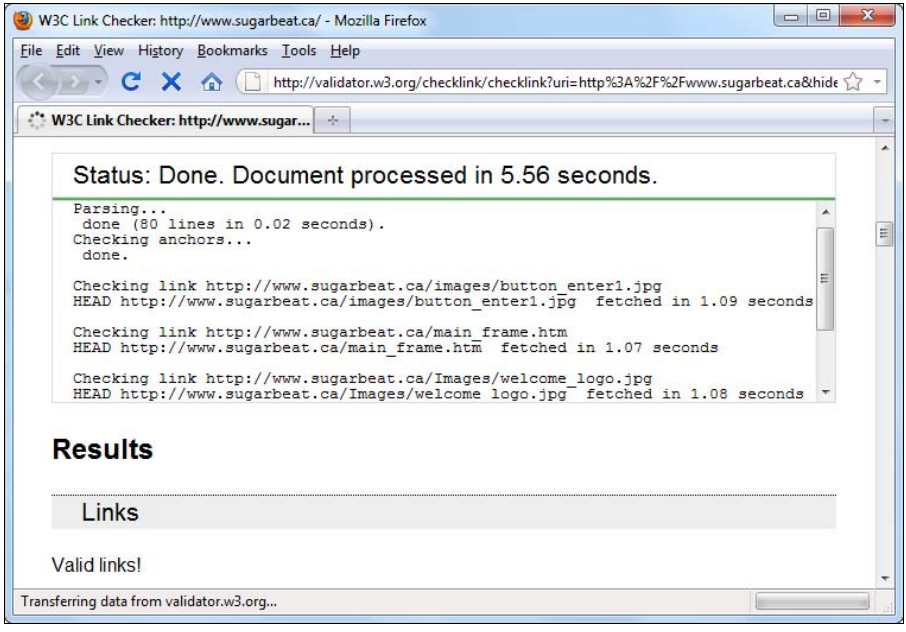


Рис. 8.11. В окончательном отчете программы проверки ссылок приводится перечень ссылок, найденных в элементах-якорях и изображениях. Программа проверки выделяет потерянные ссылки красным цветом, а требующие особого внимания — желтым. Перенаправленные ссылки — один из примеров ссылок, способных вызвать проблемы. Несмотря на текущую работоспособность, они могут устареть и вскоре выйти из строя

4. Установите флажок **Save options in a cookie** (Сохранить настройки в файле cookie), если хотите, чтобы ваш браузер запомнил заданные вами настройки программы проверки ссылок.

   Если вы воспользуетесь этой настройкой, в случае повторного применения программы проверки ссылок ваш браузер установит все флажки так же, как в предыдущий раз.

5. Для начала проверки ссылок щелкните мышью кнопку **Check** (Проверить).

   Программа проверки ссылок выводит отчет с перечислением всех проверенных ссылок (рис. 8.11). Отчет обновляется во время работы программы. Если применить рекурсию, можно увидеть, как программа проверки переходит с одной страницы на другую. Для каждой страницы в отчете создается отдельный раздел.

Использование перенаправлений

Как добропорядочный Web-обитатель вы должны уважать людей, связывающихся с вашим сайтом. Это означает, что после создания вашего сайта и приобретения им популярности следует избегать исправления имен страниц и папок. Даже мельчайшая корректировка может разорвать чью-либо ссылку и затруднить постоянным посетителям возврат на ваш сайт.

Некоторые специалисты справляются с этой проблемой с помощью *перенаправлений*. В процессе реорганизации своих сайтов они сохраняют все старые файлы, удаляя из них содержимое и заменяя старые страницы *перенаправлением*, специальной командой, автоматически направляющей браузер на новую страницу. Выгода от применения перенаправлений двойная: они препятствуют потере ссылок и не замыкают вас в старой структуре вашего сайта, когда возникает необходимость его изменения.

Для создания перенаправления нужно вставить в раздел `<head>` вашей Web-страницы особый элемент `<meta>`. Он указывает с помощью абсолютного URL новую конечную страницу и задает количество секунд, которое должен выдержать браузер перед выполнением перенаправления. Далее приведен пример.

```
<!DOCTYPE html>

<html>

<head>
  <meta http-equiv="REFRESH"
   content="10; URL=http://www.mysite.com/homepage.htm" />
  <title>Redirect</title>
</head>

<body>
  <h1>The page you want has moved</h1>
```

```
  <p>
    Please update your bookmarks. The new home page is
    <a href="http://www.mysite.com/homepage.htm">
    http://www.mysite.com/homepage.htm</a>.
  </p>
  <p>
    You should be redirected to the new site in 10 seconds. Click
    <a href="http://www.mysite.com/homepage.htm">
    here</a> to visit the new page immediately.
  </p>
</body>
</html>
```

Для того чтобы приспособить свою страницу для собственных нужд, измените количество секунд (в данный момент 10) и URL перенаправления. Когда браузер загружает эту страницу, он показывает временную страницу для отображения количества секунд и затем автоматически запрашивает новую страницу.

Перенаправленные страницы служат двум целям: они сохраняют работоспособность ваших страниц в процессе изменения структуры вашего сайта и информируют Web-посетителей о том, что ссылка устарела. Здесь оказывается полезным время задержки — посетителям несколько секунд показывается извещение о том, что они попадают на сайт неправильным путем. Многие сайты сохранят некоторое время (например, год) свои перенаправленные страницы, а затем удаляют их полностью.

ГЛАВА 9

Макет страницы

В *главе 6* вы узнали о замечательном стандарте CSS и о его способах превращения унылых страниц в настоящее сокровище. Но вы до сих пор не видели всего, что позволяют делать таблицы стилей. Вы можете применять их не только для игры с цветом, шрифтами, пробельными зонами и границами, но и для создания сложных макетов. С помощью небольшого планирования и таблиц стилей можно создать сверхгибкие страницы, которые вы сможете полностью реорганизовать простым переключением на новую таблицу стилей, не изменив ни строки HTML-кода.

В этой главе вы увидите, как именно работают свойства CSS-верстки. Вы научитесь применять современные инструменты верстки, такие как плавающие области, смежные колонки и перекрывающиеся слои с изображениями и текстом. Но сначала необходимо вернуться на шаг назад и рассмотреть проблемы верстки для Web и узнать, почему отображение в сети не так просто, как кажется.

Проблемы экранного пространства

Когда страница подготавливается к печати, учитывается реальный размер окончательного документа. Например, для плаката вы примените более крупный шрифт, чем для визитной карточки. Но в мире Web такой подход не годится, поскольку посетители вашего Web-сайта могут настроить экраны своих мониторов на разные разрешения и произвольно изменять окна браузеров. Эти настройки влияют на объем экранного пространства, с которым вынуждены работать ваши Web-страницы. Чем выше разрешение и больше окно браузера, тем больше содержимого страницы поместится на экране. Но возникает дилемма — как наилучшим образом отобразить ваши страницы, если не известны настройки экранов ваших посетителей?

Для разрешения этой проблемы Web-разработчики применяют две основные стратегии верстки.

- *Обеспечение гибкости с помощью страницы пропорционального размера.* Благодаря пропорциональной подгонке размеров ваш макет растягивается или сжи-

мается, заполняя доступное свободное пространство в окне браузера. Например, если вы создадите Web-страницу пропорционального размера со строкой меню фиксированного размера и областью контента переменного размера, строка меню всегда будет оставаться одинаковой ширины, а область контента будет расти или уменьшаться, заполняя окно браузера в любом случае, независимо от того, какого оно размера у вашего гостя. Если сомневаетесь, можно остановиться на пропорциональном размере, т. к. он гарантирует соответствие Web-страниц окну браузера любого размера. Но, возможно, придется задать макеты максимально и минимально допустимого размеров, чтобы ваши страницы не исказились до неузнаваемости. В этой главе вы узнаете, как это делается.

- *Выбор разумной фиксированной ширины.* Порой излишняя гибкость порождает собственные проблемы. Например, если сжать страницу пропорционального размера до крайности, некоторые ее элементы могут оказаться в неподходящих местах. Если у вас сложный макет с множеством графики и плавающих элементов, в результате может получиться кавардак. С другой стороны, окна очень большого размера также могут создать проблемы. Например, если растянуть страницу пропорционального размера на всю ширину широкоэкранного монитора, могут получиться слишком длинные строки текста, которые трудно читать. Подходящее решение — применение страниц фиксированной ширины, которые хорошо смотрятся в большинстве браузеров со стандартным набором настроек.

Фиксированный размер страницы — часто используемый подход. На рис. 9.1 показано, как он работает на примере сайта газеты New York Times (**www.nytimes.com**).

Если вы остановите свой выбор на варианте с фиксированным размером, необходимо выяснить, какой должна быть ширина страницы. Конечно, у вас нет возможности узнать, самый важный фактор, влияющий на ее выбор, — размер окна браузера у вашего посетителя. Помочь может свежая статистика браузеров; считается, что 20% пользователей применяют относительно небольшие дисплеи, шириной 1 024 пиксела. К ним относятся некоторые старомодные компьютерные мониторы, современные нетбуки и iPad. Фактически у всех остальных пользователей мониторы с бóльшим количеством пикселов. Но как Web-дизайнер вы должны учесть наименьший распространенный размер экрана, поэтому макет с фиксированной шириной должен отлично выглядеть при ширине в 1 000 пикселов. В этом случае Web-экстремалы смогут насладиться всем содержимым ваших страниц на устройствах с маленькими экранами, если раскроют окно своего браузера на весь экран.

> ***Совет***
>
> Некоторые редакторы Web-страниц позволяют открывать страницы в окнах браузера разного размера. Например, в Expression Web можно выбрать команды **File | Preview in Browser** (Файл | Предварительный просмотр в браузере), в которых есть возможность выбрать несколько стандартных, но устаревших размеров окон. У ряда Web-браузеров есть для этого специальные дополнительные модули. Например, в Firefox и Chrome можно применять расширение Web Developer (**http://chrispederick.com/work/web-developer**), вставляющее панель с множеством удобных инструментов Web-дизайна, включая средство быстрого изменения размера окна браузера.

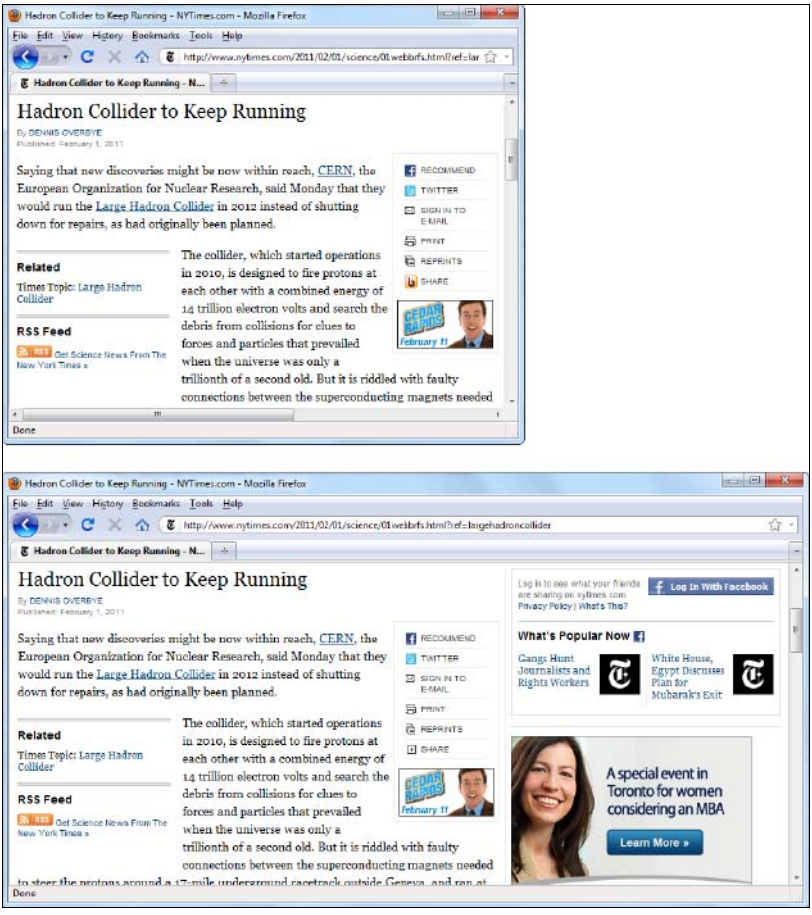


Рис. 9.1. *Вверху*: для чтения этой статьи без прокрутки необходимо окно шириной в 600 пикселов, что удобно для старых мониторов с разрешением 800×600 пикселов.
Внизу: увеличьте окно примерно до 1 000 пикселов, и вы увидите видео, рекламные объявления, поле поиска и список популярных статей. Если сделать окно еще шире, справа появится пустое пространство. Большинство новостных сайтов используют дизайн страниц с фиксированной шириной, чтобы справляться с окнами разных размеров

НА ПРОФЕССИОНАЛЬНОМ УРОВНЕ

Представление о разрешении

Пиксел — минимальная единица измерения на экране монитора, иначе называемая "точкой". Например, разрешение 1024×768 означает, что монитор отображает сетку пикселов, 1 024 пиксела в ширину и 768 пикселов в высоту (в сумме 786 432 пиксела). Самое низкое разрешение, которое сегодня можно встретить (за исключением мобильных телефонов) — 1024×600, что соответствует размерам стандартного экрана нетбука. Но имейте в виду, что люди с большими мониторами необязательно растягивают окно своего браузера на весь экран. (Помимо всего прочего по-настоящему длинные строки текста трудно читать.) По этой причине 1 000 пикселов — хорошее допущение для нижней границы ширины окна браузера.

Возможно, вы хотите для общего развития узнать, какое разрешение у вашего монитора или даже изменить его. Для этого на компьютерах Windows PC щелкните на Рабочем столе правой кнопкой мыши, выберите **Personalize** (Персонализация), щелкни-

те кнопкой мыши команду **Display Settings** (Параметры экрана) и с помощью удобного бегунка отрегулируйте разрешение. В более ранних версиях ОС Windows эти же параметры можно найти с помощью щелчка правой кнопкой мыши на Рабочем столе и выбора последовательности команд **Properties | Settings** (Свойства | Параметры). В Mac OS X щелкните кнопкой мыши команды **System Preferences | Displays** (Системные настройки | Мониторы) и затем выберите разрешение из списка.

На некоторых Web-сайтах пропорциональное изменение размеров действует лучше, чем макет с фиксированной шириной, т. к. позволяет Web-сайту наилучшим образом использовать любое имеющееся у посетителей экранное пространство. Например, выполните поиск картинок в Google и вы увидите, что графические миниатюры заполняют по ширине все окно браузера даже на широкоэкранном мониторе (рис. 9.2).

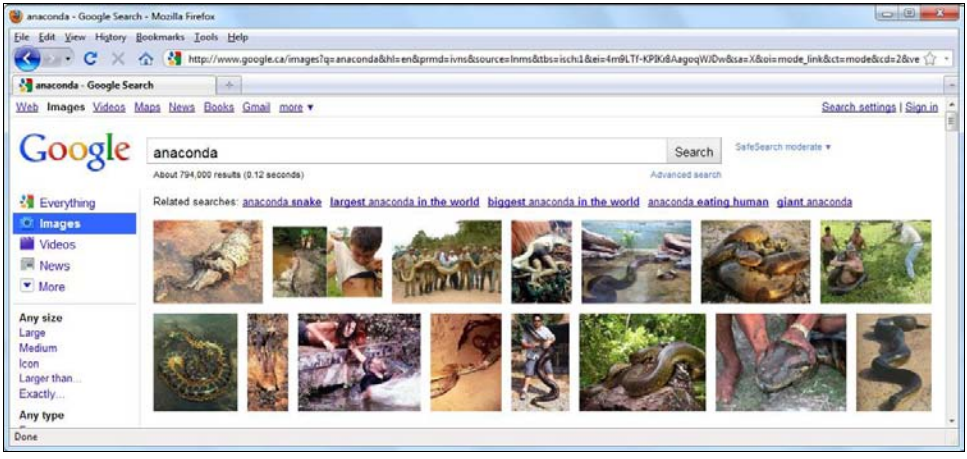


Рис. 9.2. Пропорциональная верстка способна удовлетворить потребности ваших Web-посетителей. Имеет смысл заполнять все окно браузера изображениями, а для текста такой подход не годится, т. к. глазу трудно перемещаться по слишком длинным текстовым строкам. По этой причине может возникнуть желание ограничить ширину страницы с пропорциональной версткой, как описано в *разд. "Максимальная ширина: подстраховка" данной главы*

Независимо от выбранной стратегии верстки следует тестировать ваши страницы в окнах браузеров различных размеров, чтобы убедиться в том, что посетители увидят вашу работу с самой лучшей стороны.

И в заключение следует отметить, что этот совет не применим для тех, кто хочет создать Web-сайт для мобильных телефонов и других миниатюрных мобильных устройств. Несмотря на то, что мобильные устройства могут отображать обычные Web-сайты, посетители будут вынуждены непрерывно прокручивать, уменьшать и увеличивать их. Для любителей мобильных устройств необходимо использовать таблицу стилей, предназначенную специально для мобильных устройств, и возможно даже придется переписать HTML-код ваших Web-страниц. Например, New York Times предоставляет отдельный мобильный Web-сайт ((**http://mobile.nytimes.com**). Такие сайты должны быть компактными, простыми и понятными, ведь помимо всего прочего, большинство пользователей платит за каждую порцию данных, загружаемых на их телефоны.

Несмотря на то, что огромное количество людей пользуются телефонами с выходом в Интернет, протяженность их Web-прогулок все же очень мала, и они сосредоточены в основном вокруг нескольких очень популярных Web-сайтов (Facebook и YouTube, я думаю) и нескольких простых задач (проверка прогноза погоды, новостей и результатов спортивных состязаний).

Большинство создателей Web-сайтов приходит к выводу, что нет смысла разрабатывать сайты для мобильных устройств, если они не предназначены для гуляющей публики (например, сайты с отзывами о ресторанах или с афишей кинотеатров). Если же вы создаете социальный сайт, который по замыслу должен стать блокбастером (скажем, еще одной сетью Facebook), стоит потратить силы на разработку полноценного сайта для мобильных устройств, а в случае создания личного блога, ваши усилия могут оказаться напрасными.

Макет на основе стилей

Во времена зарождения не подчиняющегося никаким законам Web-пространства дизайнеры должны были самостоятельно изобретать инструменты верстки. Одним из их любимых приемов были невидимые таблицы, использующие для размещения строк и столбцов контента элемент `<table>`, о котором вы узнали в *главе 5*. Несмотря на то, что невидимые таблицы неплохо справлялись с задачей, они оставляли после себя запутанную беспорядочную разметку. Сегодня Web-разработчики почти отказались от макетов на основе таблиц в пользу более ясного и эффективного макета на основе стилей.

Вы уже сделали первые робкие шаги к нирване, создаваемой таблицами стилей, познакомившись в *главе 7* с макетом на основе стилей для текста в специальном поле и изображений. Тем же подходом можно воспользоваться и для дизайна всей страницы. Но прежде чем двигаться дальше, неплохо рассмотреть базовые концепции, которые позволяют создавать макеты на основе стилей.

Структурирование страниц с помощью элемента * *

Перед тем как начать размещать элементы страницы, необходимо каким-то образом собрать связанный контент в единый хорошо организованный блок. В макетах на основе таблиц, описанных ранее, таким блоком была ячейка таблицы. Если вы применяете макет на основе стилей, таким блоком служит элемент `<div>`, универсальный контейнер, описанный в *разд. "Разделы и оболочки" главы 5*.

Предположим, вы хотите создать на вашей странице слева область с несколькими ссылками. Поместить каждую ссылку в эту колонку так же просто, как очистить кожицу с виноградин. С помощью элемента `<div>` можно сгруппировать их все вместе.

```
<div class="Menu">
  <a href="...">Home Page</a>
  <a href="...">Buy Our Products</a>
```

```
  <a href="...">File a Lawsuit</a>
   ...
</div>
```

Создавая любой элемент , следует выбрать имя класса, описывающего тип контента, который в нем содержится (например, Menu, Header, AdBar и т. д.). Позже вы сможете создать правило, которое определит местоположение этого элемента , а также его шрифт, цвета и границы.

Напоминаю, что у элемента нет никакого встроенного форматирования. В действительности сам по себе он ничего не делает. Волшебство возникает, когда вы соединяете свой элемент с правилом таблицы стилей.

Плавающие блоки

Обычно при верстке HTML-страницы используют потоковую модель; контент, предлагаемый вами, распространяется от верхней границы окна браузера до нижней. Браузер помещает элементы один за другим, поэтому они отображаются на выводимой странице в том же порядке, в каком они были в HTML-разметке. Когда применяются CSS-свойства верстки, вы извлекаете из потока выбранные элементы и реорганизуете их в соответствии с разными правилами.

Один из простейших приемов верстки — взять маленькую порцию контента и вывести ее из общего потока вашей страницы (рис. 9.3). В этом случае плавающий блок расположится там, куда вы его поместите, а остальной контент будет обтекать этот блок. На самом деле вы уже применяли эту методику в *главе 7*, во время формирования плавающих изображений с помощью свойства таблицы стилей с именем float. Плавающий макет обрабатывает элементы так же легко, как и элементы <img> за одним исключением: для элемента необходимо задать ширину.

> ***Примечание***
>
> Когда вы делаете изображение плавающим, браузеры автоматически формируют плавающий блок такой же ширины, что и изображение. Для плавающего элемента с текстом внутри можно выбрать подходящую ширину.

Далее приведен пример, определяющий класс, который располагается справа от какого-либо текста.

```
.FloatingBox {
  float: right;
  width: 150px;
  background-color: red;
  border-width: 2px;
  border-style: solid;
  border-color: black;
  padding: 10px;
  margin: 8px;
  font-weight: bold;
  color: white;
}
```

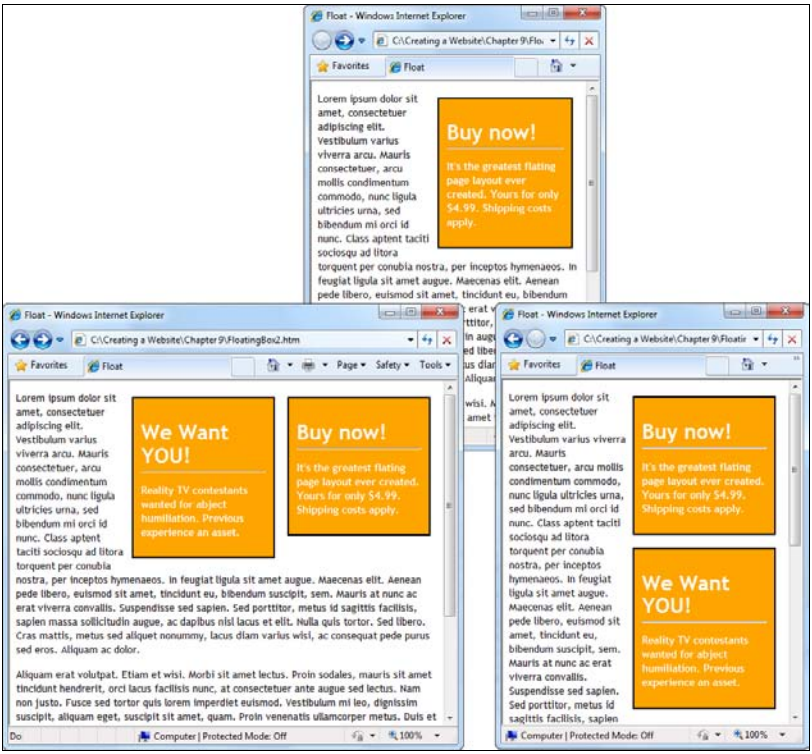


Рис. 9.3. Три примера плавающих макетов.
Вверху: стандартный плавающий блок.
Внизу слева: вы можете одновременно разместить несколько плавающих элементов. Ваш браузер вставит каждый новый блок слева от предыдущего плавающего блока.
Внизу справа: чтобы заставить второй плавающий блок расположиться под первым, добавьте свойство таблицы стилей `clear: both`

Далее приведен пример элемента `<div>`, использующего описанный класс.

```
<div class="FloatingBox">
  <h1>Buy now!</h1>
  <p>...</p>
</div>
```

Примечание

Класс `FloatingBox` задает ширину, но не высоту блока. В этом случае блок делается достаточно большим для того, чтобы вместить в себя контент. С помощью свойства `height` можно задать фиксированную высоту, но при этом есть риск обрезать окончание текста (если блок слишком мал) или отобразить слишком большую пробельную зону внизу (если блок слишком велик).

Фиксированные блоки

Примеры на рис. 9.3 называются плавающими блоками, потому что они могут "переплывать" по странице в разные места в зависимости от размера окна браузера и объема контента, окружающего их. Когда браузер обнаруживает элемент `<div>`,

применяющий свойство `float`, он располагает элемент с заданной вами стороны страницы (слева или справа). Верхнюю точку `<div>` он помещает в том месте страницы, где обнаружил этот элемент в HTML-разметке. Таким образом, если браузер находит `<div>` на середине создаваемой им страницы, он помещает плавающий блок на середину страницы.

Таблицы стилей предлагают другой вариант: можно поместить элемент в фиксированную неменяющуюся позицию. Для этого применяют свойства позиционирования `top`, `left`, `bottom`, `right` в сочетании со свойством `position` (в данном случае со значением `absolute`). Далее приведен пример.

```
.FixedBox {
  position: absolute;
  top: 20px;
  left: 0px;
  width: 150px;
  background-color: orange;
  border-width: 2px;
  border-style: solid;
  border-color: black;
  padding: 10px;
  margin: 8px;
  font-weight: bold;
  color: white;
}
```

В отличие от плавающих блоков (которые смещаются к боковым краям страницы) фиксированные блоки могут располагаться где угодно. Когда местоположение эле-

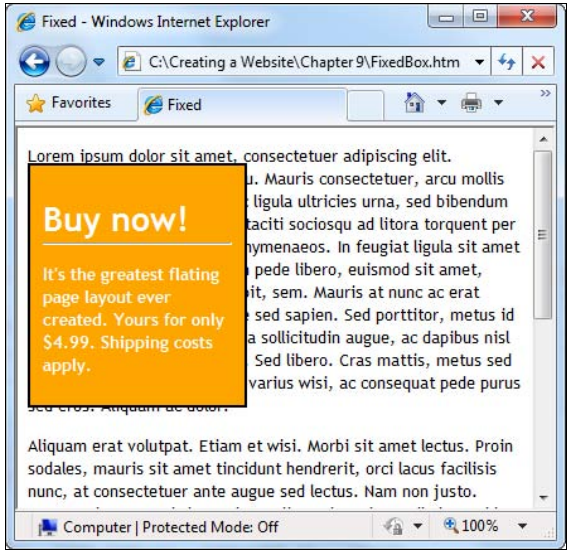


Рис. 9.4. HTML размещает фиксированный блок в соответствии с заданными вами координатами. Такой блок никогда не перемещается и не предоставляет место контенту, находящемуся под ним

мента задается с помощью абсолютного позиционирования, он удаляется из нормального потока страницы. В результате остальной контент не обтекает фиксированный блок. Вместо этого фиксированный блок помещается поверх контента, как показано на рис. 9.4.

На первый взгляд фиксированные блоки только создают проблемы. Кому захочется иметь дело с мешаниной из накладывающегося текста? Тем не менее фиксированный макет — ключевой компонент многоколоночных макетов, о которых вы узнаете в следующем разделе.

Создание макета с несколькими колонками

Как вы видели, таблицы стилей позволяют помещать элементы в определенное место на странице без какого-либо обтекания. Неосторожное использование этой возможности приводит к искажению страниц, а осторожное — позволяет создать несколько самых распространенных классических макетов Web-страниц. Хитрость заключается в создании *нескольких* блоков, одних для текста, других для изображений, и размещении их на странице рядом друг за другом (рис. 9.5). В этом случае текст и картинки будут оставаться тщательно и строго разделенными.

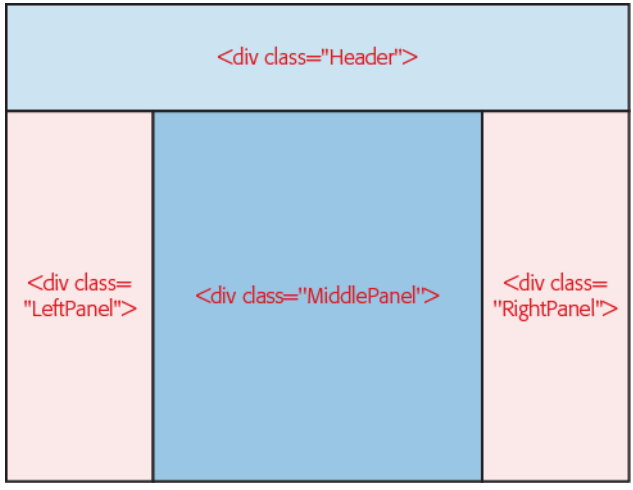


Рис. 9.5. Группировка из четырех элементов `<div>` создает классический макет страницы с тремя колонками и заголовком сверху. Вы можете загрузить эту страницу вместе со всеми примерами к данной главе со страницы **www.missingmanuals.com/cds/caw3**

Один из самых распространенных макетов Web-сайтов — страница, разделенная на две или три колонки. Колонка слева, как правило, содержит навигационные кнопки или другие ссылки. Средняя колонка занимает больше всего места и включает в себя основной контент страницы. В колонке справа, если она есть, отображается дополнительная информация, такая как реклама или еще один набор ссылок.

Сформировать страницу из данного примера удивительно просто. Но прежде чем размещать элементы `<div>`, компонующие страницу, необходимо решить, какой

макет вы выбираете: с фиксированной шириной или пропорционального размера. (Загляните в *разд. "Проблемы экранного пространства" данной главы* с кратким перечнем отличий.) В следующих разделах вы узнаете, как создавать оба варианта.

Построение макета с фиксированной шириной

Для создания макета с фиксированной шириной нужно просто определить ширину каждого элемента `<div>` в пикселах. Например, можно сделать левую и правую полосы шириной 100 пикселов, а среднюю полосу шириной 450 пикселов. В сумме (с учетом небольших полей между ними) это будет равно примерно 700 пикселам.

> ***Совет***
>
> Если вы остановились на макете с фиксированной шириной, придерживайтесь 1 000-пиксельной нормы и в пределах этой ширины удобно размещайте контент, чтобы не возникло необходимости в горизонтальной прокрутке страницы. Статья из New York Times (см. рис. 9.1) может служить хорошим примером. Пример шириной в 700 пикселов из текущего раздела соответствует этим рекомендациям, но немного маловат.

Теперь, когда пространство распределено, нужно просто создать три стилевых правила, по одному для каждой колонки. Как и в приведенных ранее примерах, вы управляете местоположением колонки с помощью задания ее левой координаты и позволяете контенту определить ее общую высоту.

Следующее правило определяет полосу шириной 100 пикселов, расположенную у левого края страницы.

```
.LeftPanel {
  position: absolute;
  left: 0px;
  width: 100px;
}
```

Средняя полоса начинается с отметки 120 пикселов и заполняет следующие 450 пикселов. Таким образом, между правым краем левой полосы и левым краем средней полосы остается пустое пространство шириной 20 пикселов. Кроме того, средняя полоса отделяется от боковых с помощью свойств, задающих границы (см. табл. 6.5).

```
.MiddlePanel {
  position: absolute;
  left: 120px;
  width: 450px;
  border-left-width: 1px;
  border-right-width: 1px;
  border-top-width: 0px;
  border-bottom-width: 0px;
  border-style: solid;
  border-color: blue;
  padding-top: 0px;
}
```

И наконец, полоса справа начинается на отметке 610 пикселов и занимает 100 пикселов.

```
.RightPanel {
  position: absolute;
  left: 610px;
  width: 100px;
font-weight: bold;
}
```

Хотя к элементу `<div>`, содержащему заголовок в верхней части страницы, также можно применить абсолютное позиционирование, этот шаг лишний. Просто убедитесь в том, что заголовок в разметке предшествует остальным элементам `<div>`.

```
<div class="Header">
  <h1>The Joy Of Styles</h1>
</div>

<div class="LeftPanel">
  <a href="...">Page 1</a><br />
  <a href="...">Page 2</a><br />
  ...
</div>
<div class="RightPanel">
<p>Donate to my untraceable Swiss Bank account ...</p>
</div>

<div class="MiddlePanel">
  <p>An expandable middle might not be ...</p>
</div>
```

В таблице стилей не задавайте верхнюю координату для всех элементов `<div>`. В этом случае браузер отобразит заголовок и затем создаст элементы `<div>` в текущей позиции, так что верх каждой колонки будет расположен сразу под заголовком. (Другой вариант — вручную задать свойство `width` для заголовка и свойство `top` для двух боковых полос, чтобы точно определить горизонтальное местоположение каждого раздела.)

Конечный результат показан на рис. 9.6.

Самое замечательное в этом примере то, что ваш HTML-документ свободен от многочисленных подробностей форматирования. Он являет собой чудесный образец четкости и ясности с контентом, разделенным на несколько понятных разделов. И если вы сохраните ваши стили во внешней таблице стилей (см. *разд. "Три вида стилей" главы 6*), то сможете начать создание второй страницы, используя тот же макет и не тратя время на разработку подходящего форматирования.

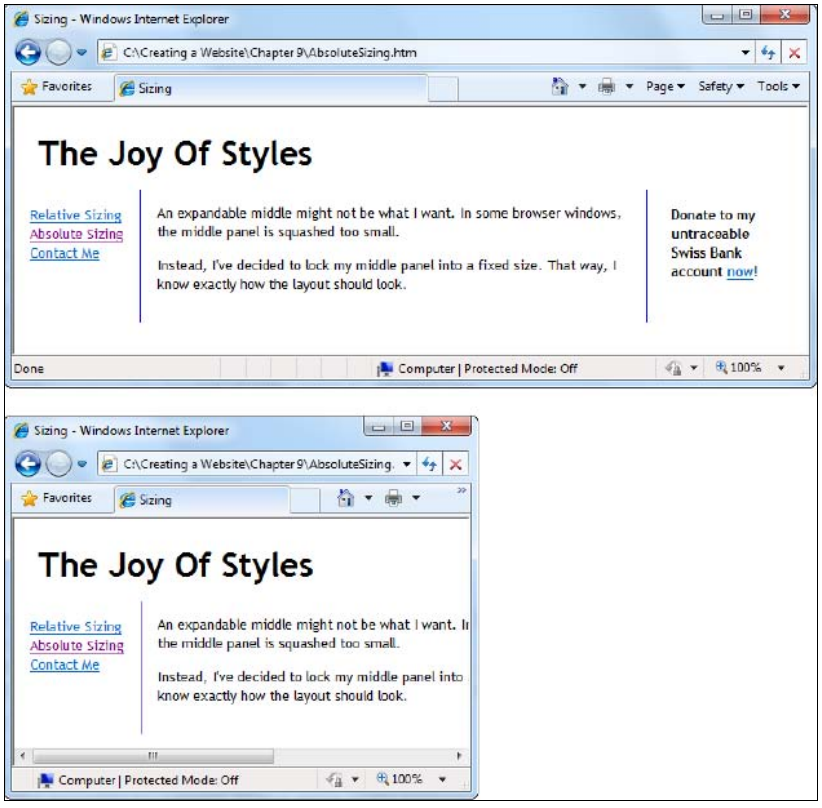


Рис. 9.6. Задание абсолютных размеров сохраняет целостность вашего макета при изменении размера окна браузера. Негативный побочный эффект заключается в том, что при слишком большой заданной ширине посетители вынуждены будут применять прокрутку, чтобы увидеть все (внизу), и это конечно же раздосадует их. Другой недостаток макета с фиксированной шириной проявится у посетителей с большими мониторами с высоким разрешением. Страница покажется им пустой из-за больших пробельных зон, которые они увидят по бокам страницы

УГОЛОК ДИЗАЙНЕРА

Центрирование макета фиксированной ширины

Порой Web-мастера центрируют горизонтально макеты с фиксированной шириной. В этом случае их контент всегда выводится в центре окна браузера (в отличие от выровненного по левому краю).

Реализация такого макета гораздо проще в теории, чем на практике. Во-первых, нужно заключить ваш макет целиком в еще один элемент `<div>`. В данном примере он назван `Body`.

```
<div class="Body">
  <div class="Header">...</div>
  <div class="Left">...</div>
  <div class="Middle">...</div>
  <div class="Right">...</div>
</div>
```

Затем нужно задать ширину этого элемента `<div>` равной общей ширине вашего макета, а также задать для правого и левого полей значение `auto`.

```
.Body {
  width: 750px;
  margin-left: auto;
  margin-right: auto;
}
```

Когда вы определите оба боковых поля как `auto`, браузер автоматически сделает их равными. В результате ваш элемент расположится посередине.

Но хитрости с оболочкой `<div>` недостаточно для решения задачи центрирования. Когда применяется этот прием, для всех элементов `<div>`, описывающих колонки, запрещено применять абсолютное позиционирование. Выход — задать все колонки `<div>` как плавающие слева, одну за другой. Если вы зададите для каждой из них фиксированную ширину и вставите их в нужном порядке, этот обходной прием будет работать чисто. Чтобы увидеть его на практике, посмотрите файл AbsoluteSizing_Centered.htm в примерах к этой главе на странице **www.missingmanuals.com/cds/caw3**.

Создание макета с изменяемым размером

В макете с изменяемым размером средняя колонка, содержащая основной контент вашей страницы, расширяется, используя все доступное пространство в окне браузера. На рис. 9.7 показана разница.

Вы можете переделать таблицу стилей, применявшуюся в предыдущем примере (для макета с фиксированной шириной) в таблицу, используемую для макета с изменяемым размером. Интересно, что стиль для левой полосы не придется менять. В нем для закрепления полосы сбоку может применяться абсолютное позиционирование.

```
.LeftPanel {
  position: absolute;
  left: 0px;
  width: 100px;
}
```

Колонка справа немного отличается. Поскольку вы не знаете, какой ширины будет средний раздел, левая координата правой полосы вам неизвестна. Обойти это затруднение можно очень тонким приемом, с помощью свойства позиционирования `right`. Оно располагает элемент `<div>` относительно правой стороны окна браузера, поэтому `0px` — это местоположение относительно правого края, `100px` — это 100 пикселов влево и т. д. В данном примере к полосе добавляется 10 пикселов дополнительного пространства, чтобы контент наверняка не попал на правую границу окна браузера.

```
.RightPanel {
  position: absolute;
  right: 10px;
  width: 100px;
  font-weight: bold;
}
```

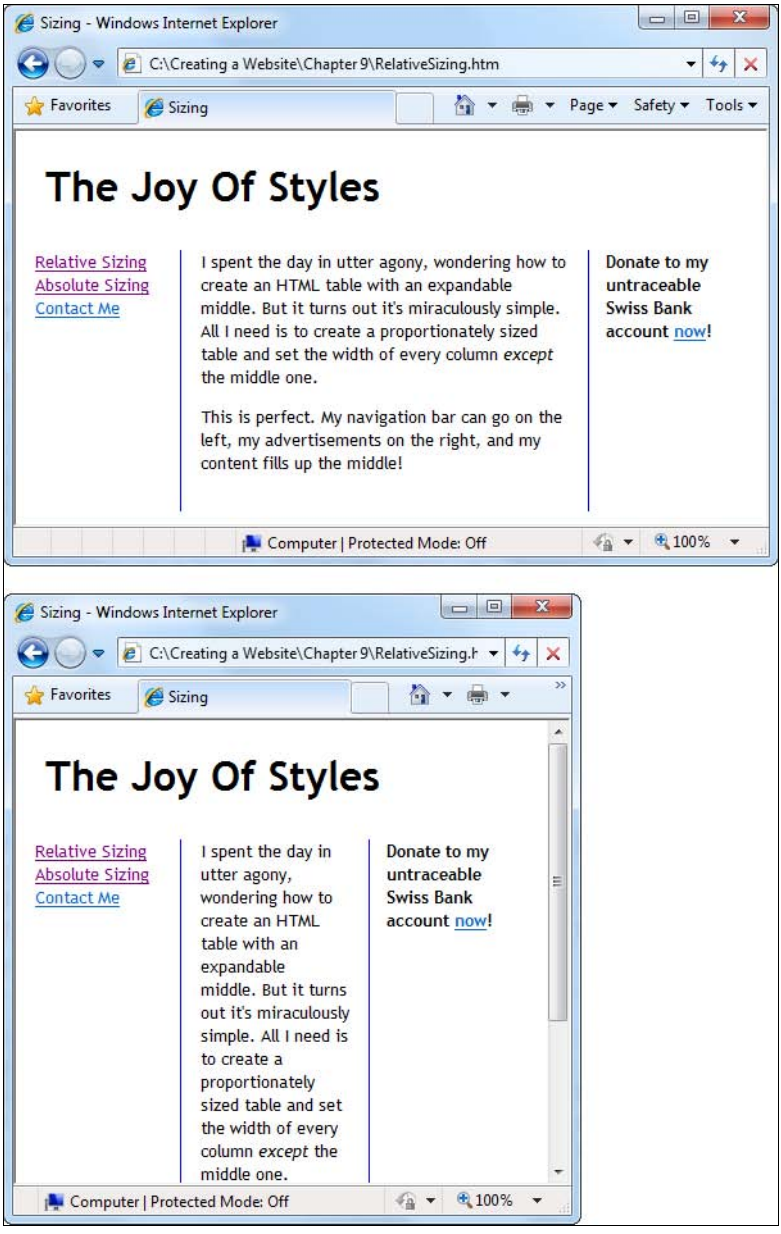


Рис. 9.7. *Вверху*: при малых размерах этот макет выглядит точно так же, как его двойник с фиксированной шириной.
Внизу: чуть-чуть сожмите или растяните окно браузера и вы увидите разницу. Боковые полосы сохраняют размер, а средняя полоса изменяет размер в зависимости от объема доступного пространства в окне браузера

Заключительный этап — задание раздела с основным контентом, расположенного между двумя описанными колонками. Для него нельзя воспользоваться абсолютным позиционированием, т. к. неизвестно, какого размера будет окно браузера. К счастью, есть еще один ловкий прием — задание величины бокового поля. Для

этого вы притворяетесь, что в среднюю полосу помещается весь поток из окна браузера, как любой стандартный фрагмент контента. Но при этом вы задаете как отступы левое и правое поля такой ширины, чтобы хватило места для размещения боковых полос.

В данном примере левая полоса равна 100 пикселам. Прибавьте 20 пикселов на промежуток между колонками и таким образом вашей средней полосе нужно левое поле шириной 120 пикселов. Правое поле — 130 пикселов, т. к. нужно компенсировать 10 пикселов, оставленных между правой полосой и правым краем окна браузера. Далее приведено окончательное правило таблицы стилей.

```
.MiddlePanel {
  margin-left: 120px;
  margin-right: 130px;
  border-left-width: 1px;
  border-right-width: 1px;
  border-top-width: 0px;
  border-bottom-width: 0px;
  border-style: solid;
  border-color: blue;
  padding-top: 0px;
}
```

В этом примере есть потенциальная каверза. В разметке теперь необходимо элементы `<div>` для левой и правой полос поместить перед элементом `<div>` для средней полосы. В противном случае браузер выведет боковые полосы после основного контента в нижней части страницы.

> ***Совет***
>
> Если размышления о горизонтальном позиционировании причиняют боль, есть простое решение. Для помещения каждого элемента `<div>` точно в нужное место используйте свойство позиционирования `top`. Это несколько удлинит таблицу стилей, но позволит вставлять в разметку элементы `<div>` в любом порядке, т. к. вы уже сообщили браузеру, где именно они должны располагаться.

Максимальная ширина: подстраховка

Когда создаются макеты с изменяемым размером, даже самые солидные страницы могут столкнуться с двумя типами проблем. Первая возникает, когда посетитель сжимает свое окно браузера до абсурдно малого размера. Пространства в нем становится так мало, что в средней колонке не помещается и одно слово, и разные разделы страницы начинают накладываться друг на друга.

Это не слишком серьезная проблема (в конце концов большинство людей не рассчитывают на привлекательный вид Web-сайта, если он сжат до толщины бумаги) — но все же небольшое упущение для профессиональной страницы. Вторая проблема встает во весь рост, если посетитель растягивает окно своего браузера до размера экрана монитора с высоким разрешением. В этом случае средняя полоса

становится такой широкой, что весть текст помещается в 2—3 строках. Помимо нелепого внешнего вида такой текст невероятно трудно читать.

Устранить обе эти проблемы можно с помощью двух дополнительных свойств CSS: `max-width` и `min-width`. Свойство `max-width` задает максимальную ширину, до которой может растягиваться элемент. Свойство `min-width` задает минимальную ширину, до которой элемент может сжиматься. По существу, когда вы предлагаете эти два ограничения, ваша страница преобразуется в макет с фиксированной шириной. Растяните страницу до ширины больше максимальной, и вы получите дополнительное свободное пространство. Сожмите ее до ширины у́же минимальной, и браузер предоставит полосы прокрутки.

Вы, наверное, думаете, что на практике этот метод можно реализовать, задав `max-width` и `min-width` только для средней полосы? Да конечно, вы можете, но результат будет неожиданный. Если вы ограничите рост средней полосы, правая полоса будет продолжать следовать за правым краем окна браузера и постепенно все дальше и дальше отодвигаться от контента. Для устранения проблемы нужно всю страницу со всеми ее колонками заключить в еще один контейнер `<div>`, как показано далее.

```
<!DOCTYPE html>
<html>

<head>
...
</head>

<body>
  <div class="BodyContainer">
    <div class="Header">...</div>
    <div class="LeftPanel">...</div>
    <div class="RightPanel">...</div>
    <div class="MiddlePanel">...</div>
  </div>
</body>
</html>
```

Затем можно задать правила для максимального и минимального размеров этого элемента `<div>`.

```
.BodyContainer {
position: absolute;
max-width: 1000px;
min-width: 100px;
}
```

Эта методика предлагает отличный компромисс между переменным размером и размером с фиксированной шириной. Теперь средняя колонка становится максимально возможной ширины в пределах заданных ограничений, и страница выглядит профессионально независимо от размера окна браузера.

СОВЕТ

Вы можете просмотреть этот пример в действии, загрузив сопроводительный материал к данной главе со страницы **www.missingmanuals.com/cds/caw3**.

Увеличение высоты колонки

До сих пор вы добивались нужной ширины колонок и не слишком заботились об их высоте. В конце концов об этом беспокоится браузер, делая элемент достаточного размера для размещения в нем контента и включая в него отступы, заданные вами в свойстве `padding` (см. табл. 6.2).

Однако есть несколько случаев, когда необходимо задавать размер колонки, исходя из размера окна браузера, а не из объема включенного в нее контента. Например, требуется применить цвет фона, заполняющий всю колонку, а не ее часть, содержащую текст. Или появляется желание применить границы, тянущиеся до нижнего края окна (в отличие от границ, виденных вами в предыдущих примерах, которые заканчивались вместе с текстом). На рис. 9.8 показано, как это выглядит на практике.

К счастью, задать высоту можно с помощью свойства `min-height`. Хитрость заключается в применении при задании размера процентов вместо пикселов. Например, если свойство `min-height` равно 100%, колонка будет растягиваться до нижнего края окна браузера, даже если в ней лишь небольшая порция контента.

Далее приведено правило форматирования для боковых полос на рис. 9.8.

```
.LeftPanel {
  position: fixed;
  top: 0px;
  left: 0px;
  left-padding: 10px;
  width: 150px;
  min-height: 100%;
  background-color:#eee;
  border-right-width: 1px;
  border-right-style: solid;
  border-right-color: black;
  padding-bottom: 8px;
}

.RightPanel {
  position: fixed;
  top: 0px;
  right: 0px;
  width: 150px;
  min-height: 100%;
  background-color:#eee;
```

```
  border-left-width: 1px;
  border-left-style: solid;
  border-left-color: black;
}
```

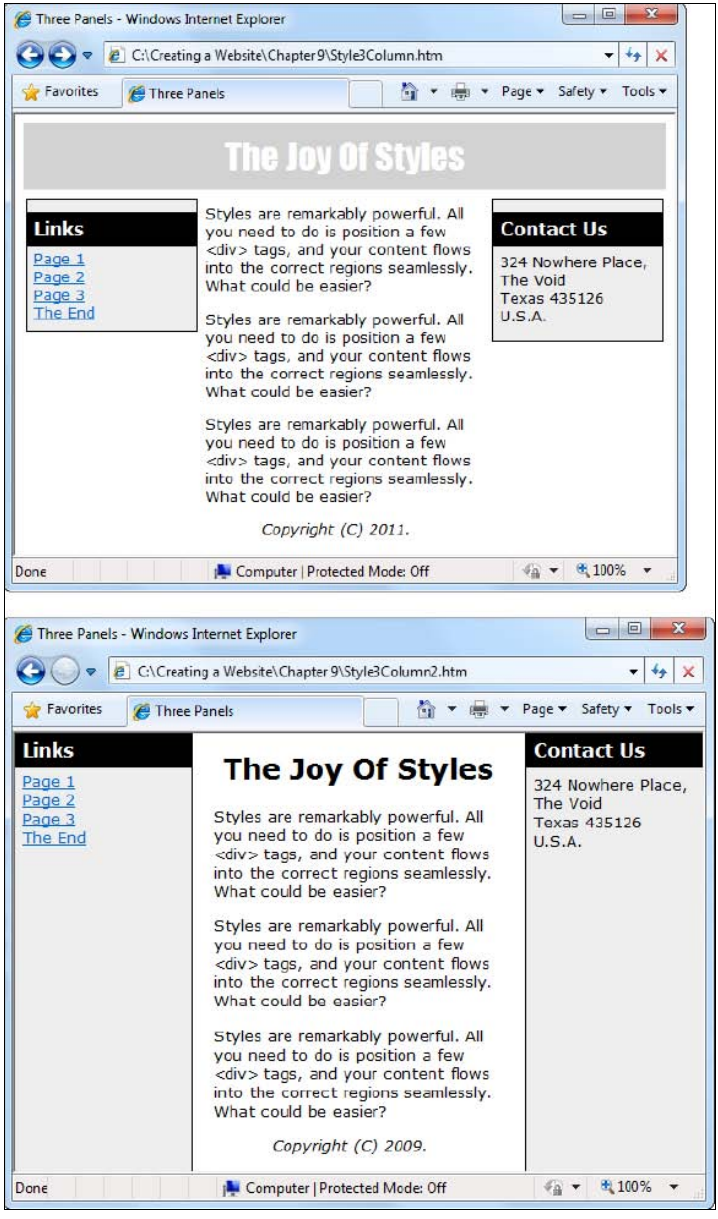


Рис. 9.8. *Вверху*: у этой страницы трехколоночный макет на основе таблицы стилей с несколькими дополнительными усовершенствованиями (например, точно настроенные границы, шрифты и фоновые цвета).
Внизу: в этом варианте того же самого дизайна высота боковых полос равна 100%, поэтому они всегда заполняют все окно браузера

Важно отметить, что в данном примере немного изменено свойство `position`. Прежде вы задавали ему значение `absolute`, которое помещало элемент на страницу в соответствии с заданными абсолютными координатами. Значение `fixed` помещает элемент в соответствии с абсолютными координатами, отсчитываемыми в окне просмотра, окне с содержимым, которое ваш браузер отображает на одном экране. В большинстве случаев `absolute` и `fixed` действуют одинаково, но в данной ситуации важна разница. Дело в том, что вам нужно определить размер полос по отношению к размеру окна браузера, а не по отношению к размеру всей Web-страницы. Если использовать абсолютное позиционирование, боковые полосы могут увеличиться больше, чем надо, и вынудить браузер отображать полосы прокрутки.

Сверхгибкие сайты: дзэн Web-дизайна

К настоящему моменту у вас, возможно, сложилось впечатление, что таблицы стилей, по крайней мере так же важны для современных Web-страниц, как HTML-разметка. В *главе 6* вы узнали, как пользоваться их возможностями форматирования. В этой главе вы увидели, как с помощью таблиц стилей управлять сложными макетами страниц. Комбинируйте вновь полученные навыки, и вы убедитесь, что таблицы стилей дают возможность полностью управлять отображением хорошо разработанной Web-страницы.

Таблицы стилей позволяют не только отформатировать и сверстать страницу, они также дают возможность изменить форматирование и верстку за один (весьма необременительный) шаг. Они — священный Грааль Web-дизайна, способ обновления, модернизации и настройки всего сайта при изменении ваших потребностей. На рис. 9.9 показан Web-сайт CSS Zen Garden, воплощающий эту мечту.

Далее приведен краткий перечень этапов разработки, которым вы должны следовать при работе над вашим Web-сайтом.

1. Прежде чем написать хотя бы один тег, спроектируйте ваши страницы.

   Начните с создания макета страницы, выделения отдельных областей.

2. Каждую область или участок страницы поместите в отдельный элемент `<div>`.
3. Задайте для каждого элемента `<div>` уникальное имя класса, в котором отражено назначение элемента, а не его формат.

   Сделайте это для всех областей страницы, даже если вы не намерены применять правила таблицы стилей прямо сейчас.

4. И в заключение напишите правила таблицы стилей, которые позиционируют и форматируют каждый элемент `<div>`.

   Это самый трудоемкий этап написания вашей разметки, но это время будет потрачено не зря. Вы сможете в любое время корректировать эти правила форматирования, не нарушая при этом вашего контента.

> ***Совет***
>
> Хотите поднять на новый уровень ваши макеты на основе таблиц стилей? Прочтите легендарные книги Эрика Мейера (Eric Meyer), например, "Smashing CSS: Professional Techniques for Modern Layout" (изд. Wiley).

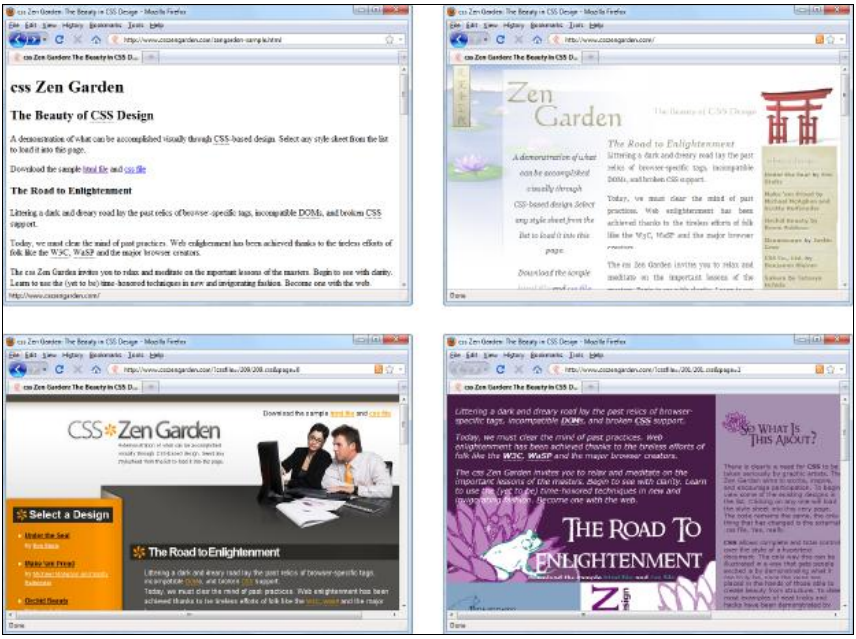


Рис. 9.9. Разные представления одной страницы. Web-сайт **www.csszengarden.com** показывает, как полностью переформатировать и реорганизовать обычную страницу (вверху слева), просто меняя используемую таблицу стилей. Самое замечательное в том, что вы можете загрузить с Web-сайта HTML-код и десятки примеров таблиц стилей для данной страницы и самостоятельно убедиться в мощи таблиц стилей

СЕМИНАР ДЛЯ ОПЫТНЫХ ПОЛЬЗОВАТЕЛЕЙ

Сложные макеты на основе таблиц стилей

Вы прошли еще совсем немного по пути, ведущему к построению многоколоночных макетов. Но Web-дизайнеры потратили многие годы, играя с CSS для того, чтобы создать более свежие, потрясающие варианты дизайна страниц. Если вы не хотите ограничиваться стандартными макетами с фиксированной шириной или с переменным размером, о которых вы уже узнали, далее перечислены три новых горизонта, к которым следует стремиться.

- **Клейкие нижние колонтитулы** (Sticky Footers). Что если у вас будет нижний колонтитул, который всегда виден и располагается не в конце вашей страницы, а в нижней части окна браузера, сразу под вашими колонками? Создание разметки для такого макета — непростая задача, но есть замечательное бесплатное решение на сайте **www.cssstickyfooter.com**. И как все стоящие разработки таблиц стилей оно поддерживается всеми браузерами.
- **Резиновые макеты** (Fluid Layouts). Резиновые макеты (также называемые текучими макетами) похожи на относительные макеты, которыми вы пользовались ранее в этой главе, за исключением того, что у них несколько колонок переменной ширины, а не одна средняя полоса переменного размера. Вы управляете тем, как браузер распределяет дополнительное пространство между колонками, определив для каждой колонки ширину в процентах. Например, если у одной колонки ширина 50%, а у двух других — по 25%, первая колонка всегда занимает половину окна браузера. Умелые руки могут с помощью резиновых макетов создавать разные динамические эффекты. (Более сложный пример см. на **www.dccdesign.co.uk**.) Тем не менее бывает трудно предусмотреть все побочные эффекты от одновременного

изменения размеров нескольких колонок и вам все же придется задать свойство `max-width` (см. *разд. "Максимальная ширина: подстраховка"*), чтобы помешать излишнему растяжению контента и затруднению его чтения.

- **Эластичные макеты** (Elastic Layouts). Эластичные макеты подобны резиновым, но ширина колонок в них задается не в процентах, а в пропорциональных единицах (em). Единица измерения em связана с размером текста, заданным в браузере. Это означает, что, если гость увеличил текущий размер текста (например, с помощью команд меню **Page | Text Size** (Страница | Размер шрифта) в Internet Explorer), ваш макет для компенсации растягивается, расширяя свои колонки для размещения увеличенного текста. Пример этой методики с сайта CSS Zen Garden можно найти по адресу **http://tinyurl.com/67v6hb6**.

Дополнительную информацию о методах резиновой и эластичной верстки можно найти в хорошей статье из журнала "Smashing Magazine" по адресу **http://tinyurl.com/q54tnl**. Если у вас есть Dreamweaver, вы можете опробовать ряд готовых резиновых и эластичных макетов. Просто создайте новую страницу и выберите один из множества шаблонов.

Несколько дополнительных приемов верстки

Теперь, когда сделаны первые шаги на пути постижения тайн верстки на основе стилей, пора рассмотреть несколько дополнительных приемов, которые могут вам пригодиться. Возможно, вы не будете применять их так часто, как многоколоночные макеты, но эти приемы неплохо взять на заметку.

Применение слоев

Помните, как тщательно приходилось размещать элементы при абсолютном позиционировании, чтобы не создать наложения одного элемента на другой? Что любопытно, самые современные Web-страницы порой *нарочно* накладывают элементы друг на друга для создания яркого визуального эффекта. Например, можно создать логотип с помощью наложения двух слов или заголовок, частично накладывающийся на изображение. В этих примерах используются перекрывающиеся *слои*.

Для применения перекрывающихся слоев необходимо сообщить браузеру, какой слой располагается сверху. Делается это с помощью простого числа, называемого *z-индекс*. Браузеры помещают элементы с высоким z-индексом поверх элементов с более низким.

Далее приведены два абсолютно позиционированных элемента, которые накладываются друг на друга.

```
.Back {
  z-index: 0;
  position: absolute;
  top: 10px;
  left: 10px;
  width: 150px;
  height: 100px;
  background-color: orange;
```

```
  border-style: dotted;
  border-width: 1px;
}

.Front {
  z-index: 1;
  position: absolute;
  top: 50px;
  left: 50px;
  width: 230px;
  height: 180px;
  font: xx-large;
  border-style: dotted;
  border-width: 1px;
}
```

Первый класс (`Back`) определяет оранжевый фоновый квадрат. Второй класс (`Front`) определяет шрифт большого размера для текста. Вы задаете `z-index` элементов так, что браузер накладывает блок `Front` (с `z-index`, равным 1) поверх блока `Back` (с `z-index`, равным 0). В приведенном примере HTML-код содержит точечные границы вокруг обоих элементов, чтобы показать, как блоки перекрываются на странице.

> ***Примечание***
>
> Реальное значение `z-index` не важно, важна только его сравнительная величина. Например, если у вас есть два элемента с `z-index`, равными 48 и 100, вы получите тот же эффект, что и для значений, равных 0 и 1: второй элемент накладывается на первый. Если у нескольких элементов одно и то же значение `z-index`, первый элемент в HTML-разметке прячется под элемент, встретившийся позже.

В вашем HTML-коде нужно с помощью элементов `<div>` создать два блока. Кроме того, имеет смысл вставить в блок `Front` какой-нибудь текст.

```
<div class="Back">
</div>

<div class="Front">
  This text is on top.
</div>
```

Загрузите эту страницу в браузер и вы увидите, что блок текста растягивается поверх части оранжевого прямоугольника и далее поверх пустого пространства (рис. 9.10, слева).

В приведенном примере можно поменять местами значения `z-index`.

```
.Back {
  z-index: 1;
  ...
}
```

```
.Front {
  z-index: 0;
  ...
}
```

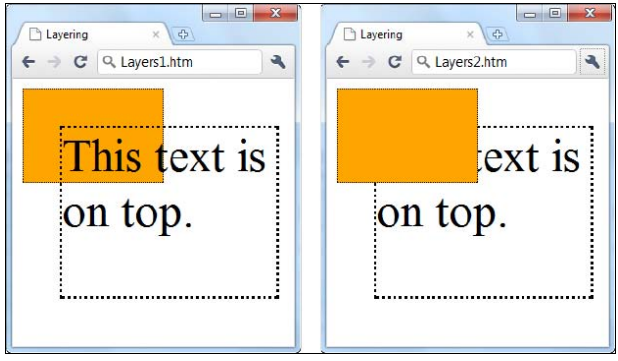


Рис. 9.10. *Слева*: у цветного прямоугольника более низкий `z-index`.
Справа: у цветного прямоугольника более высокий `z-index`, и он закрывает текст

Комбинирование абсолютного и относительного позиционирования

Специалисты по разработке таблиц стилей знают, что не надо придерживаться только абсолютного или относительного позиционирования. С помощью не громоздкого, но тщательного планирования они могут добиться наилучшего результата, применяя оба подхода.

Для того чтобы понять, как это работает, необходимо знать следующий секрет таблиц стилей: когда применяется абсолютное позиционирование, браузер интерпретирует координаты *относительно контейнера*. Как вы видели в нескольких ранее приведенных примерах, если поместить элемент `<div>` в раздел страницы `<body>`, браузер позиционирует элемент относительно страницы. Задайте левую координату элемента `<div>` равной 10 пикселам, и ваш браузер сдвинет элемент на 10 пикселов от левого края страницы. Но вот вам интересный эксперимент: попробуйте поместить тот же элемент `<div>` в другой элемент, например ячейку таблицы. Теперь браузер сдвинет элемент на 10 пикселов относительно левого края ячейки таблицы, *независимо от местоположения этой ячейки на странице*. Кажется, что элемент `<div>` существует в своем собственном мире, и этот мир — пространство контейнера, его содержащего, но не пространство основной страницы.

Как же применять этот секрет? Первый метод использует абсолютное позиционирование для создания специального эффекта, наложения текста на фотографию. Для его проверки создайте страницу с несколькими элементами `<div>`. Но не используйте абсолютное позиционирование — вместо него предоставьте элементам `<div>` возможность располагаться на странице один за другим, обычным образом. В первый и последний элементы `<div>` добавьте обычный контент (текст, картинки

и все, что захотите). А средний элемент отсоедините с помощью абсолютного позиционирования.

На рис. 9.11 показан результат. Первый элемент , как и третий, содержит обычный абзац. А средний элемент использует абсолютное позиционирование для вставки белого текста поверх картинки с надгробной плитой.



Рис. 9.11. На этой странице в среднем разделе для размещения текста поверх изображения применяется абсолютное позиционирование. Интересно то, что остальная часть страницы совершенно нормальна и, даже если вы сожмете окно браузера (тем самым сместив картинку на странице вниз), текст и картинка останутся сцеплены друг с другом

Далее приведен контент страницы.

```
<div>
  <p>Here is some ordinary content. Whatever you put here
  bumps the grave stone box further down the page.</p>
</div>

<div class="GraveContainer">
  <img alt="Tombstone" src="tombstone.jpg" />
  <p class="GraveText">Fatal error.<br />Please reboot.</p>
</div>

<div>
  <p>Here is some more ordinary content.</p>
</div>
```

К среднему элементу применяются два стилевых правила, содержащие все стилевые свойства, необходимые в данном примере. Правило GraveText задает абсолютное позиционирование.

```
p.GraveText {
  position: absolute;
  top: 60px;
  left: 115px;
  color: white;
  font-size: x-large;
  font-weight: bold;
  text-align: center;
}
```

В правиле `GraveContainer` указан размер элемента `<div>`. Обычно элемент `<div>` увеличивается до размера, вмещающего его содержимое. Но когда применяется абсолютное позиционирование, элемент `<div>` уже не знает, какого размера он должен быть, и сжимает себя до нуля. Далее приведено правило, задающее корректную высоту элемента `<div>` и гарантирующее, что последующий контент страницы (третий элемент `<div>` с заключительным абзацем) появится в нужном месте.

```
div.GraveContainer {
  position: relative;
  height: 250px;
}
```

Обратите внимание на то, что в `GraveContainer` применяется относительное позиционирование. Это позволяет элементу без каких-либо трудностей включиться в основной поток вашей Web-страницы.

Как правило, применяйте относительное позиционирование для обеспечения максимально возможной гибкости и модифицируемости макета.

Но как видно из данного примера, очень важно использовать элемент `<div>` для выключения небольших областей с применением абсолютного позиционирования — подобные действия позволяют вставлять некоторые привлекательные эффекты.

Задание размеров таблиц

Времена, когда таблицы применялись для формирования макета страницы, стали далекой историей. Но таблицы до сих пор играют определенную роль на Web-страницах, позволяя легко отобразить табличные данные (например, статистические показатели численности населения на рис. 9.12).

Обычно HTML формирует таблицу такой ширины, чтобы отобразились все ее столбцы, а каждый столбец расширяется настолько, чтобы поместилась самая длинная строка текста в нем (или поместился другой контент, например картинка). Но существует дополнительное правило: таблица не может быть шире окна браузера. Как только таблица достигает ширины, равной ширине текущего окна браузера, он начинает переносить текст внутри каждого столбца на следующую строку, и при вставке дополнительного контента таблица растет в высоту.

Конечно, возникают ситуации, когда требуется управлять размерами таблицы. Например, вы можете отвести больше места одному столбцу по сравнению с другим.

В таком случае вы можете рассчитывать на те же свойства `width` и `height`, которыми пользовались для задания размеров элементов `<div>`, когда формировали структуру страницы (см. *разд. "Построение макета с фиксированной шириной" данной главы*).

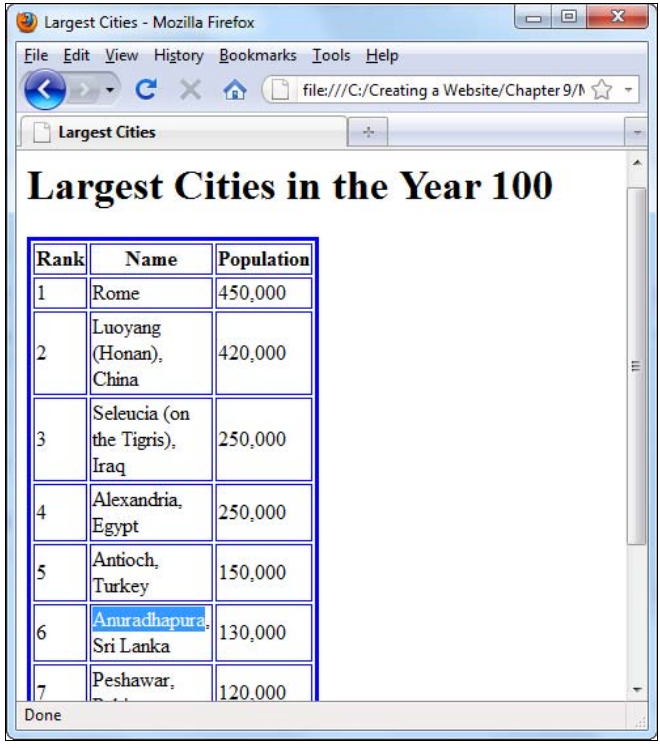


Рис. 9.12. В данном примере таблица стилей вызывается для таблицы шириной 1 пиксел. Но браузер не сжимает таблицу до такой степени и виной тому контент, влияющий на минимальный размер таблицы. В этой таблице название города Anuradhapura (Анурадхапура) — самое длинное неделимое значение, поэтому браузер использует это название для определения ширины столбца. Если вы хотите постепенно уменьшить размер столбца, попробуйте сжать текст, применив меньший размер шрифта

Задание размера таблицы

Чаще всего хочется явно задать ширину таблицы и отдельных столбцов. Если сделать это, браузер учтет заданные размеры и выполнит перенос текста в столбцах для соблюдения указанной ширины.

При задании размера таблицы можно указывать ширину в пикселах или процентах. Например, в следующем правиле ширина таблицы ограничена половиной ширины текущего контейнера (которая на стандартной странице равна половине ширины страницы).

```
table.Cities {
  width: 50%;
}
```

Для вывода на экран этой конкретной таблицы вы приводите имя класса таблицы в вашем HTML-документе.

```
<table class="Cities">
...
</table>
```

При изменении размера окна браузера таблица динамически изменяет свою ширину, сохраняя занятой половину ширины окна.

Если задать точные значения ширины в пикселах, таблица никогда не будет менять свои размеры. Например, следующее правило создает таблицу с приличной шириной 500 пикселов.

```
table.Cities {
  width: 500px;
}
```

Говоря о размере таблицы, следует сделать одну важную оговорку: вы можете сделать таблицу настолько большой, насколько захочется (даже если при этом она уйдет за границы окна браузера), но при сжатии таблицы у вас такой возможности нет. Если задать размер таблицы меньше минимального размера, необходимого для отображения ее контента, ваши настройки будут игнорироваться и на экране появится таблица этого минимального размера (см. рис. 9.12).

Задание размера столбца

Теперь, зная, как задать размер таблицы, вы, наверное, хотите выяснить, что сделает ваш браузер, если у таблицы есть больше места, чем необходимо для размещения контента. После достижения таблицей минимального размера (достаточного для размещения ее контента) браузер пропорционально распределит оставшееся свободное пространство, так что ширина всех столбцов увеличится на одинаковую величину.

Понятно, что это необязательно именно то, чего хотелось бы. Вам быть может захочется создать широкий столбец с описаниями и узкий столбец с плотно сжатым текстом. Или возникнет желание задать конкретный размер ширины столбцов, для того чтобы все ваши страницы выглядели одинаково, даже если их содержимое отличается.

Для задания ширины столбца применяют свойство `width` в сочетании с элементами `<td>` и `<th>`. И снова можно задать пропорциональные размеры в процентах или точные в пикселах. Но у пропорционального размера, когда он применяется к столбцам, проявляется небольшая особенность. Раньше, когда вы использовали значение в процентах для задания ширины таблицы, ширина таблицы определялась относительно ширины страницы. В предыдущем примере была таблица шириной 50%, т. е. она занимала 50% всей ширины страницы. Но когда значение в процентах используется для задания ширины *столбца*, вы определяете процент от ширины таблицы, в состав которой входит столбец. Таким образом, если задается ширина столбца 50%, столбец занимает 50% ширины таблицы.

Когда вы определяете размеры столбцов, необходимо создать стилевое правило для каждого из них, присвоив каждому столбцу уникальное имя класса (см. *разд. "Более мощные селекторы" главы 6*). Потенциально у столбцов может быть разная ширина: вы не можете написать одно стилевое правило, применяемое к каждому столбцу, если не хотите, чтобы у всех столбцов была одна и та же ширина.

В следующем наборе стилевых правил задается своя собственная ширина для каждого столбца таблицы, показанной на рис. 9.12.

```
th.Rank {
  width: 10%;
}
th.Name {
  width: 80%;
}
th.Population {
  width: 10%;
}
```

В данном примере имена классов совпадают с названиями столбцов, что позволяет легче понять, какое правило применяется к какому столбцу.

> ***Примечание***
>
> Когда для определения ширины столбцов применяются проценты, нет необходимости задавать значения для всех трех столбцов. Если вы один пропустите, браузер отведет этому столбцу все оставшееся пространство таблицы. Если же вы решите определить ширину всех столбцов (как в предыдущем примере), во избежание путаницы следите за тем, чтобы все значения в сумме равнялись 100%. В противном случае браузер переопределит одно из заданных вами значений, и вы не будете знать, как будет выглядеть ваша таблица на самом деле.

Для того чтобы приведенные правила оказали влияние, необходимо применить их к соответствующим ячейкам.

```
<table class="Cities">
  <tr>
    <th class="Rank">Rank</th>
    <th class="Name">Name</th>
    <th class="Population">Population</th>
  </tr>
  <tr>
    <td>1</td>
    <td>Rome</td>
    <td>450,000</td>
  </tr>
...
</table>
```

Обратите внимание, что ширина задана только для элементов столбцов в первой строке (в данном примере тех, что содержат ячейки с заголовками). Вы могли бы применить правило к каждой строке, но в этом нет смысла. Когда браузер форми-

рует таблицу, он просматривает структуру всей таблицы, чтобы определить требуемый размер, основываясь на содержимом ячеек и любых явно заданных значениях ширины. Если для нескольких ячеек в одном и том же столбце задать разную ширину, браузер просто применит наибольшее значение.

> **СОВЕТ**
>
> Неплохая идея задавать размеры таблицы, применив стилевые правила к первой строке. При таком подходе ваш HTML-код легче читается, так вы сразу можете четко видеть размеры вашей таблицы.

Задание размера строки

Задать размер строки так же легко, как размер столбца. Лучше всего применить свойство `height` в элементе `<tr>`, как показано далее.

```
tr.TallRow {
  height: 100px;
}
```

Когда вы меняете размер строки, меняются ячейки в каждом столбце, формирующие эту строку. Но при этом вы можете задать собственную высоту для каждой строки таблицы.

> **СОВЕТ**
>
> Нужно увеличить пробельные зоны внутри таблицы? Стилевые правила справятся с этим легко. Для увеличения свободного пространства между контентом ячейки и ее границами увеличьте значение свойства `padding` в элементах `<td>` и `<tr>`. Для увеличения пробельной зоны между границами ячейки и любыми соседними ячейками увеличьте ширину поля для элементов `<td>` и `<tr>`. В табл. 6.2 см. дополнительную информацию о настройке этих размеров.

10

ГЛАВА 10

Страницы с многократно используемыми элементами

Когда вы начнете создавать большие и более сложные Web-сайты, то наверняка столкнетесь с одной из главнейших проблем Web-дизайна: созданием общих компонентов, появляющихся на каждой странице.

Например, вы решите вставить меню из ссылок, позволяющих посетителям переходить из одного раздела вашего сайта в другой. Эти ссылки для отображения на странице можно поместить в таблицу или элемент `<div>` (оба приема описаны в *главе 9*), но в любом случае вам понадобится выполнить многократное копирование и вставку, чтобы вывести меню на каждой странице вашего сайта. Если вы не будете очень внимательны, на одной из страниц в результате может появиться немного отличающаяся версия одного и того же меню. Когда придет время вносить изменения в меню, вы столкнетесь с кошмарной необходимостью обновления всех ваших страниц. Web-создатели, выбирающие такой подход, вряд ли в выходные проводят много времени на свежем воздухе.

У описанной проблемы нет простого решения, но умелые Web-дизайнеры научились применять множество обходных приемов. В этой главе вы узнаете о двух самых доступных решениях: серверных включениях и шаблонах страниц в редакторах Dreamweaver и Expression Web.

Представление о страницах с многократно используемыми элементами

К настоящему моменту вы уже накопили солидный комплект тактических приемов и хитростей для создания Web-страниц. Вы научились отшлифовывать ваши страницы с помощью современных шрифтов и цветов, украшать их ультрамодными приемами верстки и включать в структуру изображения и ссылки. Но применение этих методов для создания Web-сайта порождает новые проблемы.

Первое затруднение, с которым вы столкнетесь при переходе от одной страницы к десятку или более страниц, — способы их унификации. Единообразное форматирование выполнить сравнительно легко. Если вы тщательно разработали структуру

вашего сайта и используете внешнюю таблицу стилей (см. *главу 6*), можно создать сколько угодно страниц с общим внешним оформлением.

Но у таблиц стилей есть собственные ограничения. Они не помогут, если вам нужно вставить одно и то же содержимое на несколько страниц. Это реальная проблема, т. к. на всех страницах современных Web-сайтов повторяются определенные элементы, такие как заголовки и наборы навигационных кнопок (рис. 10.1).

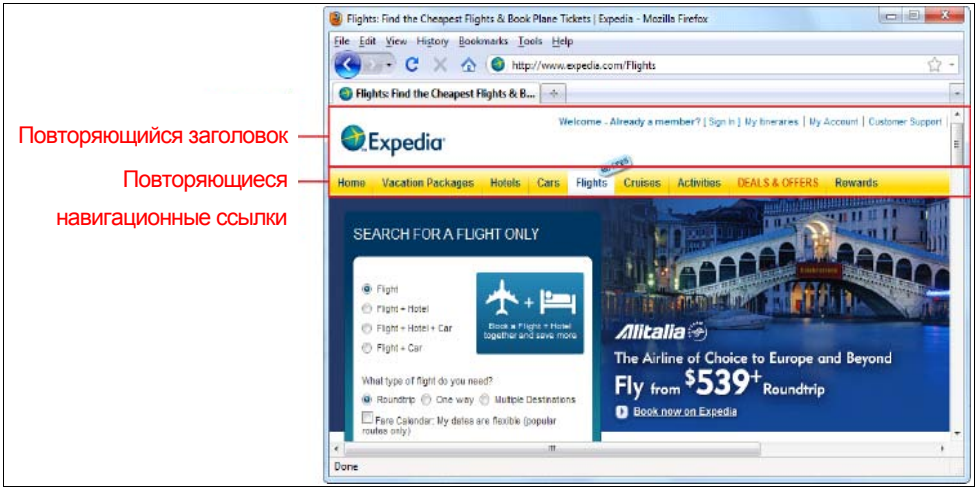


Рис. 10.1. Все страницы на **www.expedia.com** — это страницы с многократно используемыми элементами. На каждую из них вставлены вместе несколько компонентов, включая навигационные вкладки (одни и те же независимо от вашего местоположения на сайте) и контент (меняющийся от страницы к странице). Этот сорт дизайна встречается на сайтах в разных частях Web-пространства

Как же Web-дизайнеры создают страницы с повторяющимися элементами? Большие Web-сайты применяют системы управления контентом или настраиваемые Web-приложения (см. далее *врезку "Как это делается?"*). Обычные Web-разработчики предпочитают полагаться на нечто более простое. Далее перечислены три метода для решения этой проблемы.

- *Серверные включения.* Серверное включение — это команда, вставляющая один HTML-файл в другой. Она позволяет четко отделить блок HTML-контента (например, меню) и многократно использовать его на нескольких страницах. Но имейте в виду: HTML-стандарт не поддерживает серверные включения, поэтому применять их можно только если у вас есть подходящий тип сервера.
- *Бесшовные фреймы.* Новое средство HTML5, позволяющее применять элемент `<iframe>` для вставки контента одной страницы на другую. Оно похоже на серверное включение, но магическое действие совершает не сервер, а браузер. Бесшовные фреймы представляют собой замечательное решение, но в настоящее время мало пригодное, т. к. ни один браузер пока не поддерживает их полностью.
- *Шаблоны страниц.* У многофункциональных редакторов Web-страниц (а именно Dreamweaver и Expression Web) есть средство создания шаблонов страниц.

Вы начинаете с создания шаблона, определяющего структуру ваших Web-страниц и вставляющего повторяющийся контент, который должен появляться на каждой странице (например, меню или заголовок). Затем вы применяете этот шаблон для создания всех страниц вашего сайта. Замечательно то, что, когда шаблон обновляется, Web-редактор автоматически обновляет все страницы, использующие этот шаблон.

ПРИМЕЧАНИЕ

В недавнем прошлом для отображения страниц с многократно используемыми элементами в отдельных областях окна браузера Web-разработчики применяли вызывающее споры HTML-средство, именуемое фреймами. Но с фреймами связан целый ряд печально известных проблем. Они, например, сбивают с толку поисковые машины, создают проблемы для браузеров при отслеживании хронологии и не слишком приспособлены для разных размеров экранов и мобильных устройств. В наши дни фреймы считаются устаревшим средством.

КАК ЭТО ДЕЛАЕТСЯ?

Страницы с многократно используемыми элементами на больших Web-сайтах

У популярных Web-сайтов, похоже, нет проблем при обработке повторяющегося контента. На сайте Amazon, к примеру, вы всегда видите знакомую поисковую панель с вкладками независимо от того, какой товар просматриваете. На сайте Expedia сохраняется один и тот же набор навигационных вкладок независимо от того, какой отпуск вы планируете. Дело в том, что Amazon и Expedia, как почти все самые крупные и популярные Web-сайты, на самом деле — *Web-приложения*. Когда запрашивается страница с одного из таких сайтов, настраиваемый пользователем компонент программного обеспечения создает HTML-страницу на лету.

Например, при просмотре товара на Amazon Web-приложение считывает информацию о товаре из громадной базы данных, преобразует ее в HTML-страницу и дополняет ее самым свежим вариантом поисковой панели. Ваш браузер отображает окончательный результат как единую страницу. Этот метод позволяет Amazon собирать любой контент на внешней Web-странице, не заставляя разработчиков сайта поддерживать тысячи (или даже миллионы) разных HTML-файлов. Web-приложения во многом похожи на описанные ранее серверные включения (в обоих случаях процесс включает сборку страницы из фрагментов, имеющихся на Web-сервере), но они более тщательно разработаны.

Несмотря на то, что Web-приложения предлагают элегантное решение для создания страниц с многократно используемыми элементами, это тупиковый вариант для непрофессионала. Для того чтобы самостоятельно создать и сопровождать Web-приложение, нужно быть высококлассным программистом и иметь команду IT-специалистов для оказания помощи. Поэтому если вы еще не решили начать карьеру твердолобого программиста, лучше воспользоваться методами, описанными в этой главе.

Серверные включения

Несмотря на то, что вы не сможете написать Web-приложение самостоятельно, можно позаимствовать у модели Web-приложения несколько хитрых приемов, если ваш Web-хост их поддерживает. Простейший пример — технология, называемая

серверными включениями (Server-Side Includes, SSI), упрощенная версия методики HTML-сборки, применяемой на таких сайтах, как Amazon и Expedia.

По существу серверное включение — это команда, заставляющая Web-сервер вставить содержимое одного HTML-файла в другой. Предположим, что вы хотите использовать одно и то же меню на нескольких страницах. Можно было бы начать с сохранения меню в виде отдельного файла с именем menu.htm. Далее приведено его содержимое.

```
<h1>Menu</h1>
<a href="...">Page 1</a><br />
<a href="...">Page 2</a><br />
<a href="...">Page 3</a><br />
<a href="...">The End</a>
```

Обратите внимание на то, что menu.htm — не полный HTML-документ. В нем отсутствуют такие элементы, как `<html>`, `<head>` и `<body>`. Это всего лишь строительный блок, который вы поместите на другие полноценные HTML-страницы.

Теперь вы готовы к использованию меню на Web-странице. Для этого вставьте специальную команду `include` в то место страницы, где должно появиться меню. Далее показано, как это выглядит.

```
<!--#include file="menu.htm" -->
```

Команда `include` с помощью символов `<!--` в начале и символов `-->` в конце маскируется под HTML-комментарий (см. *врезку "Секреты ремесла" в разд. "Структурирование текста" главы 2*). Но ее середина содержит реальную историю. Знак решетки (`#`) означает, что это команда Web-серверу, а атрибут `file` указывает на файл, который вы хотите использовать.

Далее приведена команда `include` в действии на готовой Web-странице.

```
<!DOCTYPE html>

<html>
<head>
  <title>Server-Side Include Test</title>
  <link rel="stylesheet" href="styles.css" />
</head>

<body>
  <div class="Header">
    <h1>Templates Rule!</h1>
  </div>

  <div class="MenuPanel">
    <!--#include file="menu.htm" -->
  </div>

  <div class="Content">
    <p>This is the welcome page. Just above this text is the handy menu
    for this site.</p>
```

```
  </div>
</body>
</html>
```

Когда запрашивается эта страница, Web-сервер сканирует ее в поисках команд. Когда он находит команду `include`, то извлекает заданный файл и вставляет его содержимое в это место Web-страницы. Затем он отправляет окончательный обработанный файл вам. В данном примере это означает, что ваш Web-браузер получает Web-страницу, выглядящую следующим образом (рис. 10.2).

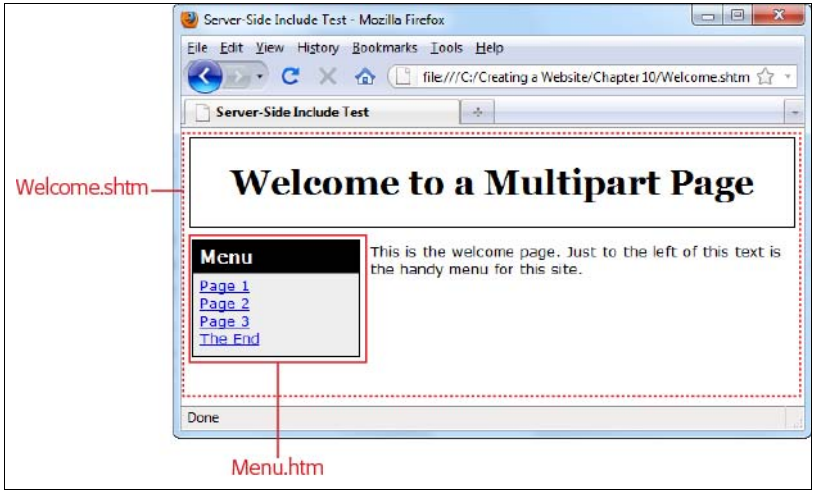


Рис. 10.2. Хоть эта страница и выглядит обычно, ее отображение не обошлось без магии. Перед тем как послать эту страницу вам, Web-сервер считывает из отдельного файла ссылки меню и вставляет их на страницу

```
<!DOCTYPE html>

<html>
<head>
  <title>Server-Side Include Test</title>
  <link rel="stylesheet" href="styles.css" />
</head>

<body>
  <div class="Header">
    <h1>Welcome to a Multipart Page</h1>
  </div>

  <div class="MenuPanel">
    <h1>Menu</h1>
    <a href="...">Page 1</a><br />
    <a href="...">Page 2</a><br />
    <a href="...">Page 3</a><br />
```

```
    <a href="...">The End</a>
  </div>

  <div class="ContentPanel">
    <p>This is the welcome page. Just to the left of this text is the
    handy menu for this site.</p>
  </div>
</body>
</html>
```

Достоинство этого метода очевидно. Вы можете вставить команду `include` на любое количество страниц и при этом сохранять единственную копию вашего меню. Это позволяет легко редактировать его и сохранять на всех страницах одну и ту же версию меню.

Если вам кажется, что приведенное рассуждение излишне оптимистично для того, чтобы быть правдой, что ж, так оно и есть. Вы можете столкнуться с рядом трудностей.

- *Поддержка Web-сервером.* Не все Web-серверы поддерживают серверные включения. Для получения достоверной информации свяжитесь с вашей компанией, предоставляющей Web-хостинг.
- *Типы страниц.* Для того чтобы серверное включение работало, Web-сервер должен обрабатывать вашу страницу и искать команды `include` при каждом запросе ее браузером. Делается это автоматически, но только если вы используете подходящий тип страницы. У ваших файлов не должно быть расширений htm или html, вместо них следует применять расширения shtm или shtml (на Web-сервере Apache) или asp или aspx (на Web-сервере Microsoft IIS). И снова детали выясняйте у компании, предоставляющей вам Web-хостинг.
- *Трудности разработки.* Серверные включения начинают действовать только когда есть работающий Web-сервер. Если открыть Web-страницу, хранящуюся на жестком диске, ваш браузер проигнорирует команду `include`, и вы вообще не увидите меню. Это затрудняет тестирование сайта без выгрузки его на реальный Web-сервер. Dreamweaver приносит лишь частичное облегчение — если вы откроете страницу, применяющую серверные включения, во время редактирования этой страницы в режиме дизайна в редакторе Dreamweaver вы увидите содержимое включаемых файлов.

Если вам известно, что Web-хост поддерживает серверные включения, и вас не беспокоят трудности разработки, почему не пустить их в ход? А если серверные включения вас все же не устраивают, можно рассмотреть другое решение: шаблоны страниц (в следующем разделе).

> ***Примечание***
>
> Поскольку команда `include` выглядит как HTML-комментарий, не возникнет проблем, даже если вы поместите страницы с серверными включениями на Web-сервер, их не поддерживающий. Он просто проигнорирует команды серверных включений.

Шаблоны страниц

Вы уже видели, как можно с помощью серверных включений вставлять один и тот же контент на ряд страниц. Это замечательный метод, если только ваша компания Web-хостинга поддерживает серверные включения, и вас не пугают трудности разработки. В конце концов альтернатива (создание на каждой странице отдельной копии повторяющегося контента) — это верный способ погубить последние оставшиеся нейроны вашего переутомленного мозга.

Но у Web-разработчиков, владеющих одним из двух ведущих программных средств Web-дизайна — Adobe Dreamweaver и Microsoft Expression Web — есть еще одна возможность. Они могут создать *шаблон страницы*, определяющий структуру страниц их сайта, и затем применять этот шаблон без устали. Это средство подобно серверным включениям, но вместо того, чтобы заставлять работать Web-сервер, вы ставите задачу перед вашей программой редактирования Web-страниц. Это значит, что вам не нужно беспокоиться об ошибках или совместимости Web-сервера. Таким образом, шаблоны страниц могут стать отличным компромиссом для маленьких или средних по размеру Web-сайтов.

Но прежде чем начать работать с шаблонами страниц, следует узнать о нескольких недостатках.

- *Дополнительное время.* Каждый раз, когда изменяется шаблон страницы, ваше средство Web-дизайна вынуждено обновлять все страницы, использующие шаблон. Поэтому шаблоны страниц не подходят для громадных сайтов, т. к. процесс обновления будет слишком долгим.
- *Больше уязвимостей.* Как вы увидите далее, технология применения шаблонов страниц основана на нескольких секретных комментариях, которые вы прячете на своих Web-страницах. К сожалению, очень легко случайно удалить или переместить один из этих комментариев и нарушить связь между страницей и ее шаблоном. При использовании шаблонов страниц редактировать страницы нужно с большой осторожностью.
- *Отсутствие стандарта.* Шаблоны страниц в Dreamweaver и Expression Web действуют по-разному. Если с помощью шаблонов страниц вы создали прекрасный сайт в Dreamweaver, перенастроить его для редактирования в Expression Web (или наоборот) не удастся, во всяком случае без мучительного процесса преобразования, который придется выполнять вручную.

Если вы согласны мириться с этими недостатками для того, чтобы создать настоящие страницы с многократно используемыми элементами, продолжайте чтение.

Общее представление о шаблонах страниц

Инструменты формирования шаблонов страниц в Dreamweaver и Expression Web удивительно похожи. Далее приведены основные правила.

- Вы начинаете с создания шаблона страницы. Как это ни странно, у файлов шаблонов в обеих программах одно и то же расширение dwt. В Dreamweaver это со-

кращение названия Dreamweaver Web Template, в Expression Web — названия Dynamic Web Template.

- Шаблон — это обычная HTML-страница, используемая как основа для всех остальных страниц вашего сайта. Контент, который вставляется с помощью шаблона, становится *фиксированным контентом*: ваш редактор Web-страниц переносит его на каждую страницу, создаваемую вами с помощью данного шаблона. Если вы помещаете в шаблон панель меню, эту панель получат все страницы. Изменять фиксированный контент можно только в самом шаблоне; на страницах, созданных с помощью этого шаблона, отредактировать фиксированный контент не удастся.
- Помимо фиксированного, неизменного контента шаблон страницы содержит *редактируемые области*, куда вставляется уникальный контент каждой страницы. Для создания такой области применяются особые HTML-комментарии. Хотя эти комментарии в Dreamweaver и Expression Web выглядят одинаково, они немного отличаются, поэтому шаблон страницы из одной программы нельзя использовать в другой.
- После того как шаблон создан, можно формировать отдельные страницы сайта, которые его используют. Эти страницы включают в себя весь фиксированный контент из шаблона и предоставляют новый контент для всех редактируемых областей страницы. Специальные комментарии всегда остаются на своем месте. Они позволяют редактору Web-страниц обновлять страницу при изменении шаблона.

Для того чтобы по-настоящему понять работу динамических Web-шаблонов, нужно увидеть их в содержимом законченной Web-страницы. На рис. 10.3 показан подходящий образец — простая страница с многократно используемыми элементами,

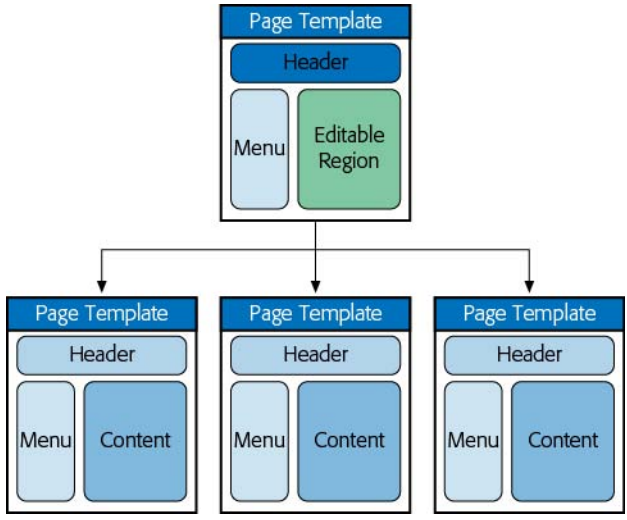


Рис. 10.3. С помощью шаблонов страниц ваш редактор Web-страниц соединяет вместе шаблон и уникальный контент каждой страницы

заголовком и меню. Она очень похожа на пример с серверными включениями, который вы видели в начале этой главы.

В следующих разделах будут созданы шаблон страницы, показанный на рис. 10.4, и четыре страницы, использующие его.

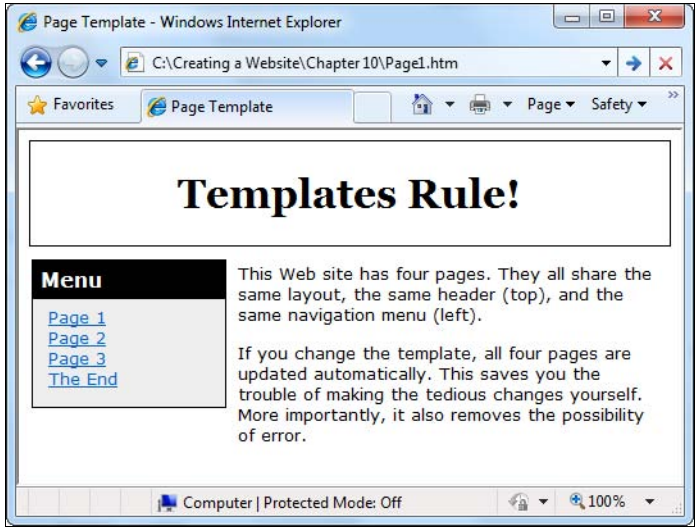


Рис. 10.4. Каждой странице на этом Web-сайте нужны одни и те же заголовок и меню

Создание нового шаблона страницы

В шаблон можно превратить любую имеющуюся страницу, но в большинстве случаев самый легкий способ — позволить вашему редактору Web-страниц показать пример создания шаблона.

Далее показано, как это делается в редакторе Dreamweaver.

1. Прежде чем начать, убедитесь, что вы описали ваш Web-сайт (см. *разд. "Задание сайта в Dreamweaver" главы 4*).

   Dreamweaver всегда помещает шаблоны страниц в подпапку Templates в папке вашего Web-сайта. Если вы не описали Web-сайт, все равно вы сможете создать шаблон страницы, но его нельзя будет использовать для создания других страниц, а это сделает ваш шаблон практически бесполезным.

2. Выберите команды **File | New** (Файл | Создать).

   Откроется окно **Dreamweaver's New Document** (Создать документ).

3. Слева выберите категорию **Blank Page** (Пустой шаблон). В списке **Page Type** (Тип шаблона) укажите **HTML Template** (HTML).

   В списке **Layout** (Макет) оставьте выделенным значение **<none>** (Нет). Вы сможете применять другие макеты, но в этом примере предполагается, что вы создаете весь шаблон с нуля.

Сейчас самое время выбрать из списка **DocType** (Тип документа) нужный тип, тогда не придется изменять его вручную после создания шаблона.

4. Щелкните мышью кнопку **Create** (Создать).

   Будет создан новый шаблон страницы с минимальной разметкой. В следующих разделах вы узнаете, как настроить шаблон.

После того как создание шаблона завершено, выберите команды **File | Save** (Файл | Сохранить) для того, чтобы открыть диалоговое окно **Save As Template** (Сохранить как). Из списка **Site** (Web-сайт) выберите описанный вами сайт, в папке которого вы хотите сохранить шаблон. Первоначально для шаблона предлагается имя Untitled_1.dwt, но его можно заменить более информативным и затем нажать кнопку **Save** (Сохранить), чтобы узаконить внесенные изменения. Редактор Dreamweaver сохраняет шаблон в папке Templates, вложенной в папку вашего Web-сайта.

Далее описано, как делается то же самое в редакторе Expression Web.

1. Перед тем как начать, убедитесь, что созданы скрытые папки метаданных (см. *разд. "Задание сайта в Expression Web" главы 4*).

   Expression Web использует эти папки для хранения подробных сведений о Web-страницах вашего сайта. Например, он отслеживает все страницы, использующие данный шаблон, что позволяет редактору обновлять их при изменении вами шаблона.

2. Выберите команды **File | New | Page** (Файл | Создать | Страница).

   Редактор Expression Web откроет окно **New** (Создать).

3. В левом списке выберите строку **General** (Общие). В списке справа выберите **Dynamic Web Template** (Динамический Web-шаблон).

   В этот момент можно щелкнуть кнопкой мыши ссылку **Page Editor Options** (Параметры редактора страниц), чтобы установить дополнительные параметры, включая автоматический выбор типа документа.

4. Щелкните мышью кнопку **OK**.

   Первоначально у вашего шаблона будет имя Untitled_1.dwt. Когда будете сохранять шаблон (выбрав команды **File | Save** (Файл | Сохранить)), можно заменить его более подходящим именем.

Теперь у вас есть совершенно новый шаблон страницы. Пока он не более чем базовая HTML-заготовка. Для того чтобы превратить его во что-то более полезное, необходимо понять, как действуют шаблоны страниц.

Анатомия шаблона страницы

Шаблоны страниц — абсолютно обычные HTML-страницы. Магия возникает благодаря специальным комментариям. Хотя они выглядят как обычные HTML-комментарии, на самом деле они разбивают страницу на несколько отдельных редактируемых областей (рис. 10.5).

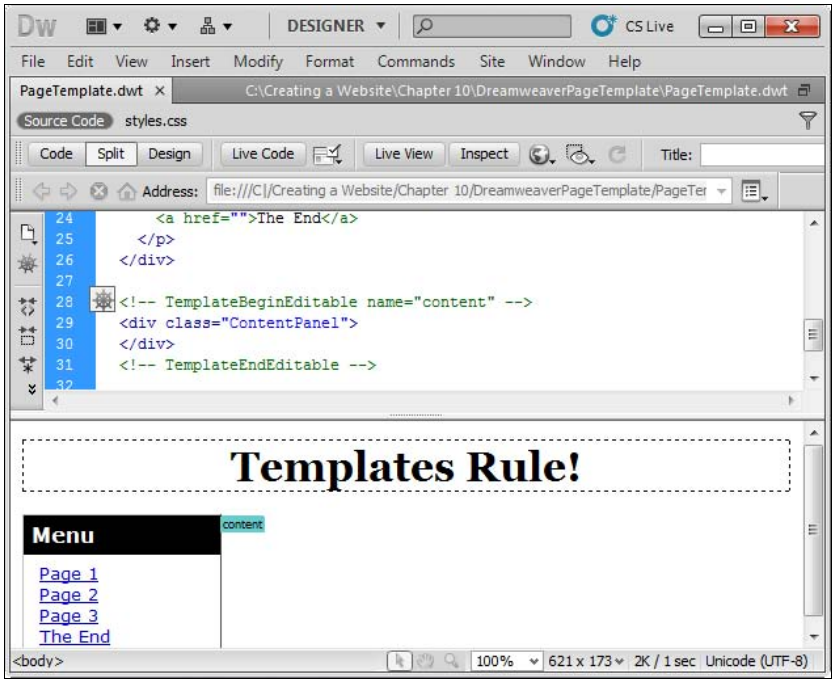


Рис. 10.5. Редактирование шаблона подобно редактированию обычной Web-страницы. Можно применять то же самое представление **Design** и те же самые команды. Но если просмотреть исходный код страницы, вы обнаружите специальные комментарии, обозначающие редактируемые области страницы

Как вы уже знаете, в шаблон включен фиксированный контент. Когда вы создаете новую страницу, использующую данный шаблон, фиксированный контент вы изменить не можете. Комментарии в шаблоне обозначают редактируемые области, в которые можно вставлять новый контент. Комментарии эти парные, первый из них указывает на начало редактируемой области, а второй обозначает ее конец. Далее приведена пара комментариев в шаблоне редактора Dreamweaver.

```
<!-- TemplateBeginEditable name="body" -->
...
<!-- TemplateEndEditable -->
```

Далее показана аналогичная пара в шаблоне редактора Expression Web.

```
<!-- #BeginEditable "body" -->
...
<!-- #EndEditable -->
```

Следует обратить внимание на две особенности. Во-первых, комментарии начинаются со стандартного индикатора комментариев `<!--`, за которым следует специальная команда (например, `TemplateBeginEditable` или `#BeginEditable`). Именно по ним ваш редактор Web-страниц определяет, что комментарий — на самом деле команда шаблона. Во-вторых, вы видите, что комментарии присваивают имя вашей редактируемой области. В обоих примерах область называется `body`.

Примечание

Поскольку шаблоны используют комментарии, они становятся более уязвимы. Казалось бы, незначительные изменения, например удаление одного из пары комментариев, изменение имени области или размещение комментариев в неправильном порядке могут вызвать проблемы. В худшем случае ваш редактор может так запутаться, что обновление шаблона приведет к стиранию части страницы. Для того чтобы избежать подобных проблем, всегда создавайте резервную копию вашего Web-сайта перед его редактированием, особенно когда применяются шаблоны.

Теперь, когда структура комментариев понятна, можно создавать шаблон для страницы, показанной на рис. 10.4. В этом примере заголовок и панель меню — фиксированные, неизменяемые элементы. Область контента — это часть страницы под заголовком и справа от панели меню.

Самый простой способ создания редактируемой области — собственноручная вставка волшебных комментариев в режиме представления **Code** (Код). В Dreamweaver можно выбрать команды **Insert | Template Objects | Editable Region** (Вставка | Шаблоны | Редактируемая область). Web-разработчики, находящиеся в цейтноте, могут нажать комбинацию клавиш <Ctrl>+<Alt>+<V>.

В Expression Web у вас будет чуть больше работы. Сначала поставьте ваш курсор туда, где хотите расположить редактируемую область, и затем выберите последовательность команд **Format | Dynamic Web Template | Manage Editable Regions** (Формат | Динамический Web-шаблон | Управление редактируемыми областями), чтобы открыть окно **Editable Regions** (Редактируемые области). В нем введите имя новой области и щелкните мышью кнопку **Add** (Добавить), чтобы создать ее.

Далее приведена окончательная версия шаблона, созданного в редакторе Dreamweaver.

```
<!DOCTYPE html>

<html>
<head>

  <!-- TemplateBeginEditable name="title" -->
  <title></title>
  <!-- TemplateEndEditable -->

  <link rel="stylesheet" href="styles.css" />
</head>
<body>
  <div class="Header">
    <h1>Templates Rule!</h1>
  </div>
  <div class="MenuPanel">
    <h1>Menu</h1>
    <p>
      <a href="">Page 1</a><br />
      <a href="">Page 2</a><br />
```

```
      <a href="">Page 3</a><br />
      <a href="">The End</a>
    </p>
  </div>

  <!-- TemplateBeginEditable name="content" -->
  <div class="ContentPanel">
  </div>
  <!-- TemplateEndEditable -->

</body>
</html>
```

А теперь его эквивалент в редакторе Expression Web.

```
<!DOCTYPE html>

<html>
<head>

  <!-- #BeginEditable "title" -->
  <title></title>
  <!-- #EndEditable -->

  <link rel="stylesheet" href="styles.css" />
</head>

<body>
  <div class="Header">
    <h1>Templates Rule!</h1>
  </div>

  <div class="MenuPanel">
    <h1>Menu</h1>

    <p>
      <a href="">Page 1</a><br />
      <a href="">Page 2</a><br />
      <a href="">Page 3</a><br />
      <a href="">The End</a>
    </p>
  </div>

  <!-- #BeginEditable "content" -->
  <div class="ContentPanel">
  </div>
  <!-- #EndEditable -->

</body>
</html>
```

Обратите внимание на то, что в данном примере на самом деле создаются две редактируемые области. Одна для контента, отображаемого справа от панели меню, а другая — названия сверху, в окне браузера. Благодаря этой последней детали вы сможете дать всем вашим страницам уникальные названия.

Кроме того, обе редактируемые области содержат некоторый контент (теги элементов `<title>` или `<div>`). Когда вы будете создавать страницу из этого шаблона, редактируемые области всегда будут содержать эти элементы. Но вы можете удалить их или заменить чем-то совершенно другим.

Применение шаблона страницы

После окончания работы над шаблоном вы можете начать его использовать. Этот процесс аналогичен в обеих программах. Вот как он выполняется в редакторе Dreamweaver.

1. Выберите команды **File | New** (Файл | Создать).

   Редактор Dreamweaver откроет окно **New Document** (Создать документ).

2. В левой части окна **New Document** (Создать документ) выберите строку **Page From Template** (Страница из шаблона). Затем в списке **Site** (Веб-сайт) выберите ваш Web-сайт.

   Программа отобразит список шаблонов в папке Templates вашего Web-сайта.

3. Выберите нужный шаблон и щелкните мышью кнопку **Create** (Создать).

   Вы увидите разметку для фиксированного и редактируемого контента на вашей новой странице. Как уже отмечалось, вы не можете редактировать фиксированный контент; чтобы напомнить вам об этом Dreamweaver отображает его светло-серым цветом.

   Убедитесь, что установлен флажок **Update page when template changes** (Обновлять страницу при изменении шаблона). В этом случае при изменении вашего шаблона Dreamweaver обновит все страницы, использующие данный шаблон.

Далее описана та же самая задача в редакторе Expression Web.

1. Выберите команды **File | New** (Файл | Создать).

   Редактор Expression Web откроет окно **New** (Создать).

2. В первом списке выберите строку **General** (Общие). Во втором — **Create from Dynamic Web Template** (Создать из динамического Web-шаблона). Затем щелкните мышью кнопку **OK**.

   Программа откроет окно **Attach Dynamic Web Template** (Присоединить динамический Web-шаблон).

3. Найдите файл с расширением dwt, созданный в предыдущем разделе, выделите его и затем щелкните мышью кнопку **Open** (Открыть).

   Редактор Expression Web отображает разметку для фиксированного и редактируемого контентов на вашей новой странице. Он выделяет фиксированный кон-

тент желтым цветом фона. (И Dreamweaver, и Expression Web делают фиксированный контент еще более отличимым в режиме представления **Design** (Дизайн), как показано на рис. 10.6.)

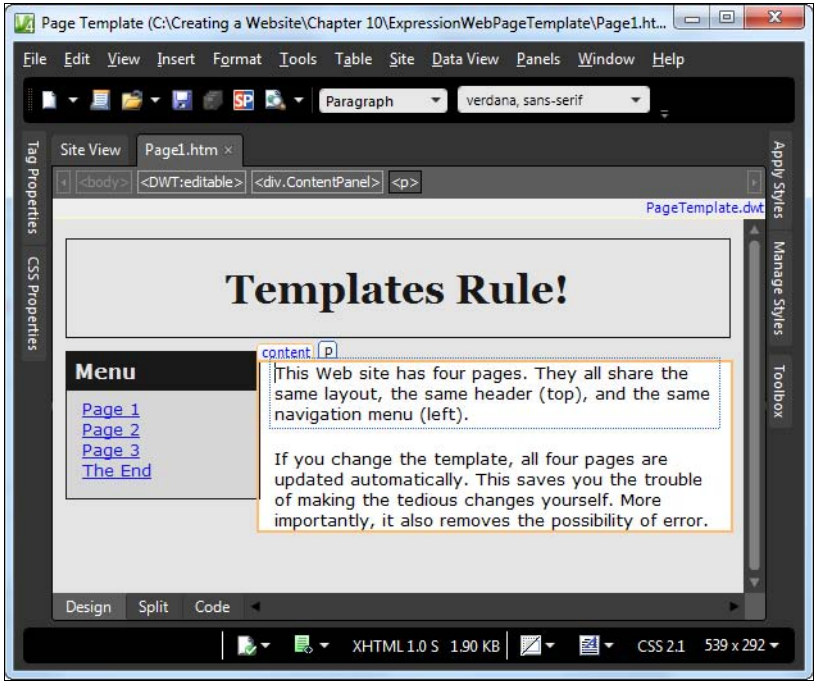


Рис. 10.6. В режиме представления **Design** обе программы обозначают редактируемые области рамкой и крошечной меткой в верхней части страницы с именем области (в данном случае "content"). Для вставки контента вы можете щелкнуть кнопкой мыши внутри редактируемой области, но в любой другой части страницы щелчок кнопкой мыши ни к чему не приведет

Для создания страницы, показанной на рис. 10.4, вы просто вставляете в область редактируемого контента заголовок и пару абзацев текста. Далее приведена разметка законченной страницы в редакторе Dreamweaver. Вставленный контент выделен жирным шрифтом.

```
<!DOCTYPE html>

<html>
<!-- InstanceBegin template="/Templates/PageTemplate.dwt"
codeOutsideHTMLIsLocked="false" -->
<head>
  <!-- InstanceBeginEditable name="title" -->
  <title>Page Templates</title>
  <!-- InstanceEndEditable -->

  <link rel="stylesheet" href="styles.css" />
</head>
```

```
<body>
  <div class="Header">
    <h1>Templates Rule!</h1>
  </div>

  <div class="MenuPanel">
    <h1>Menu</h1>
    <p>
      <a href="">Page 1</a><br />
      <a href="">Page 2</a><br />
      <a href="">Page 3</a><br />
      <a href="">The End</a>
    </p>
  </div>

  <!-- InstanceBeginEditable name="content" -->
  <div class="ContentPanel">
    <p> This website has four pages. They all share the same layout,
    the same header (top), and the same navigation menu (left).</p>
...
  </div>
  <!-- InstanceEndEditable -->

</body>

<!-- InstanceEnd --></html>
```

Когда в редакторе Dreamweaver вы применяете шаблон для создания отдельных страниц, комментарии на новых страницах немного отличаются от комментариев на первоначальной странице. Например, Dreamweaver заменяет команду `TemplateBeginEditable` командой `InstanceBeginEditable`.

В редакторе Expression Web, как показано далее, новая страница получает те же самые комментарии, что и исходный шаблон.

```
<!DOCTYPE html>

<html>

<!-- #BeginTemplate "PageTemplate.dwt" -->
<head>
  <!-- #BeginEditable "title" -->
  <title>Page Templates</title>
  <!-- #EndEditable -->

  <link rel="stylesheet" href="styles.css" />
</head>
```

```
<body>
  <div class="Header">
    <h1>Templates Rule!</h1>
  </div>

  <div class="MenuPanel">
    <h1>Menu</h1>
    <p>
      <a href="">Page 1</a><br />
      <a href="">Page 2</a><br />
      <a href="">Page 3</a><br />
      <a href="">The End</a>
    </p>
  </div>

  <!-- #BeginEditable "content" -->
  <div class="ContentPanel">
    <p> This website has four pages. They all share the same layout,
    the same header (top), and the same navigation menu (left).</p>
    ...
  </div>
  <!-- #EndEditable -->

</body>
<!-- #EndTemplate -->
</html>
```

> ***Примечание***
>
> Когда вы будете готовы выгрузить ваш Web-сайт на сервер, не забудьте, что вы должны выгрузить ваши *Web-страницы*, а не шаблоны. Шаблоны создаются для удобства проектирования, и вы применяете их только на своем компьютере.

В приведенном примере показана только одна страница. Вы поймете реальную пользу шаблона, когда создадите десятки основанных на нем страниц. При создании новой страницы вам всегда будет нужно задать название и вставить немного контента.

Но самое главное достоинство шаблона вы оцените, когда понадобится корректировать исходный шаблон. Представьте себе, что вы модифицируете шаблон, вставляя стильную современную графику в его заголовок.

```
<div class="Header">
  <img src="TemplatesRule.jpeg" alt="Templates Rule!" />
</div>
```

После того как вы сохраните изменения, ваш редактор Web-страниц поинтересуется, хотите ли вы применить эти изменения ко всем связанным страницам текущего Web-сайта. Ответьте утвердительно, и редактор быстро и бесшумно откроет все страницы, использующие этот шаблон, и обновит их, добавив новый контент. Результат — моментальное внешнее обновление вашего Web-сайта (рис. 10.7).

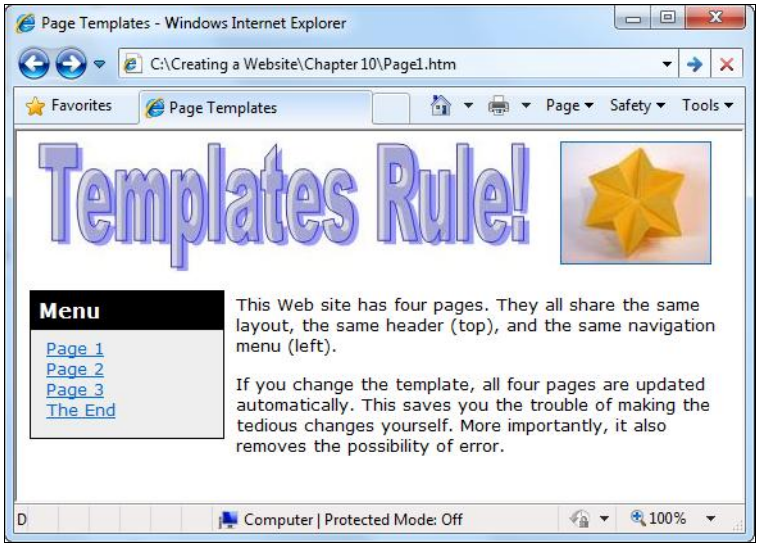


Рис. 10.7. Графика в заголовке делает ваши Web-страницы выгодно отличающимися от остальных. Но лучше всего то, что вам не нужно было даже заглядывать в файл Page1.htm для вставки графики. Ваш редактор Web-страниц сделал обновление за вас

Примечание

Несмотря на то что в данной главе дан подробный обзор технологии применения шаблонов, вам могут понадобиться дополнительные сведения. Например, изменения шаблона могут оказаться так велики, что отредактированный шаблон станет несовместим со страницами, использующими его. Или у вас может возникнуть желание связать существующую страницу с другим шаблоном. Для того чтобы узнать о нюансах применения шаблонов страниц, обращайтесь к электронному ресурсу, посвященному вашему редактору Web-страниц. Поклонники Dreamweaver могут приобрести книгу "Dreamweaver CS5: The Missing Manual". Пользователи Expression Web могут познакомиться с бесплатной электронной книгой, посвященной основам работы с шаблонами на Web-сайте **http://tinyurl.com/9whbc7**.

11

ГЛАВА 11

Обнародование вашего сайта

Вы уже отшлифовали ваш дизайнерский стиль и узнали, как создавать элегантные и тщательно проработанные страницы. Пришло время перейти к совершенно иной роли, промоутера Web-сайта.

Самый лучший способ привлечь посетителей на ваш сайт — оказаться в списке результатов Web-поиска. Чтобы это произошло, ведущие поисковые машины (такие как Google, Yahoo и Bing) должны узнать о вашем сайте и считать его достаточно значимым для включения в результаты поиска. Например, если вы на сайте **www.sweetsaltysweets.com** торгуете жареными деликатесами, хотелось бы, чтобы ваш Web-сайт оказался в результатах поиска "chocolate-covered potato chips" (покрытые шоколадом картофельные чипсы). Идеальным было бы его появление в одной из первых строк результатов поиска или по крайней мере на первой странице списка, что существенно увеличит ваши шансы быть замеченным.

В этой главе вы поймете, как работают поисковые машины, и узнаете, как гарантировать регулярное попадание вашего сайта в результаты поиска, захват нужной информации и демонстрацию вашего совершенства всему миру. (Позже, в *главе 12*, это путешествие будет продолжено, и вы узнаете о дополнительных стратегиях продвижения сайта, которые могут помочь вам пробиться в верхние строчки результатов поиска.)

И наконец, вы научитесь измерять успешность вашего сайта с помощью *учета* или *отслеживания посетителей*. Вы примените бесплатный сервис Google Analytics, чтобы узнать некоторые сокровенные тайны ваших посетителей, например, где они живут, какие браузеры используют и какие из ваших Web-страниц считают совершенно невыносимыми. Благодаря этой информации у вас появятся все необходимые средства для улучшения вашего контента и тонкой настройки структуры Web-сайта. Вы не успеете оглянуться, как станете популярнее шоколадного мороженого.

План продвижения вашего Web-сайта

Прежде чем вплотную заняться продвижением Web-сайта, необходимо выработать план. Хватайте карандаш и кипу бумаги и приготовьтесь записать ваши сообра-

жения о глобальном доминировании Web-сайта (зловещий хохот здесь вполне уместен).

Хотя у всех Web-специалистов своя тактика, но Web-знатоки обычно соглашаются с тем, что для лучшего представления сайта на рынке необходимо выполнить следующее.

1. Создать действительно выдающийся сайт.

   Если вы начнете продвигать ваш сайт, когда на нем еще нечего будет смотреть, вы потратите усилия зря (а возможно, и сожжете несколько мостов). Для того чтобы посетители никогда больше не вернулись к вам, нет ничего лучше, чем Web-сайт, состоящий из сообщения "Under Construction" (В разработке).

2. См. п. 1.

   Если сомневаетесь, продолжайте шлифовать и совершенствовать ваш сайт. Затейливая графика — не самое главное. Гораздо важнее, есть ли у вас действительно ценный контент. Спросите себя: "Путешествуя по Web, я задержал бы свой взгляд на этом сайте?" Обязательно потратьте время на вставку инструментов, заставляющих посетителей вернуться назад. Одна из прекрасных возможностей — организация дискуссионного форума (подробности об этом см. в *следующей главе*).

3. Сделайте ваш сайт удобным для поисковых машин.

   Существует ряд способов настройки и оптимизации сайта, позволяющих поисковым машинам легче понять ваш контент. Мелкие детали вроде названий страниц, альтернативного текста и метаэлементов легко просмотреть на стадии разработки сайта, но их важность возрастает, когда необходимо сделать его популярным.

4. Предложите ваш Web-сайт интернет-каталогам.

   Как и поисковые машины, интернет-каталоги помогают посетителям находить Web-сайты. Разница между каталогами и поисковыми машинами состоит в том, что каталоги, как правило, меньше и формируются людьми, в отличие от громадных беспорядочно развивающихся текстовых индексов, собранных компьютерами. Поисковые машины могут предоставить привилегированный режим Web-сайтам, появившимся в списках каталогов.

5. Представьте ваш сайт поисковым машинам в Интернете.

   Теперь вы на пути к успеху. Как только вы представили ваш сайт поисковым тяжеловесам, таким как Google и Yahoo, он официально становится предметом общественного интереса. Но ему потребуется время, чтобы повысить свое положение в иерархии и привлечь внимание.

6. Настройте публичный профиль вашего Web-сайта с помощью средств Google Webmaster Tools.

   Эти удобные средства позволят вам понять, как Google видит ваш сайт и выявляет некоторую важную информацию о нем. Например, с их помощью можно задать географический регион, с которым вы хотите связать ваш сайт, или обна-

ружить, есть ли у Google проблемы при индексировании некоторых ваших страниц.

7. Выясните, что происходит.

   Для оценки достоинств и недостатков вашей стратегии продвижения необходимо измерить некоторые важные статистические параметры — сколько людей посещает ваш сайт, как долго они остаются на нем и сколько посетителей возвращается еще раз. Для оценки результатов вам придется поработать с такими инструментами, как счетчики посещений и серверные журналы.

Перечисленные шаги будут подробно разбираться в данной главе.

Сделайте ваш сайт удобным для поисковых машин

"Удобный для поисковых машин" Web-сайт — это сайт, понятный поисковым машинам. Как объясняется в приведенной далее врезке, поисковой машине необходимо иметь возможность добыть некоторую существенную информацию с ваших Web-страниц. Затем она анализирует полученные данные и решает, насколько ваш сайт ценен для миллионов ежеминутных Web-искателей.

Современные поисковые машины решают эту задачу быстро и незаметно, не показывая, насколько она трудна. Но за кадром для анализа исходных данных Web-сайта они применяют множество запутанной логики и числовую обработку больших объемов информации. Если вы сможете упростить работу поисковых машин и сделать контент, качество и актуальность вашего сайта исключительными, вы тут же можете быть вознаграждены более высоким местом в результатах поиска.

СЕКРЕТЫ ПРОФЕССИОНАЛА

Как работают поисковые машины

У поисковой Web-машины, такой как Google, есть три компонента. Первый — автоматическая программа, блуждающая по Web-пространству и загружающая все, что найдет. Эта программа (часто называемая более красочными именами, например паук, робот, бот или краулер) однажды натыкается на ваш сайт и копирует его содержимое.

Второй компонент — индексатор, пережевывающий Web-страницы и извлекающий порцию важной информации, включая название страницы, описание, ключевые слова. Кроме того, индексатор записывает уйму более заумных данных. Например, поисковая машина Google отслеживает слова, чаще других встречающиеся на странице, какие еще сайты связаны с вашей страницей и т. д. Всю эту обработанную информацию индексатор включает в гигантский каталог (точнее, базу данных).

Финальная задача поисковой машины, которая известна вам больше других, — внешний интерфейс или исходная поисковая страница. Вы вводите ключевые слова, за которыми охотитесь, а поисковая машина просматривает свой каталог и ищет подходящие страницы. У разных машин различные способы выбора страниц, но основная идея — гарантировать появление самых лучших и самых подходящих страниц в первых строках результатов поиска. (*Лучшие* — это страницы, которые поисковая машина оценивает, как очень популярные и тесно связанные. *Самые подходящие* — это страницы, которые наилучшим образом соответствуют ключевым словам.) Из-за сложных

алгоритмов, применяемых поисковыми машинами, немного отличающийся поиск (скажем, "целебная сила зеленого чая" и просто "зеленый чай") может привести к совершенно разным спискам результатов поиска.

Выбирайте информативные названия страниц

Помните этот короткий фрагмент контента, который помещается в элемент `<title>` на вашей Web-странице? Его легко пропустить, но этот элемент жизненно важен, потому что текст названия становится ключевым фрагментом идентификационной информации, который люди видят на странице с результатами поиска (рис. 11.1).

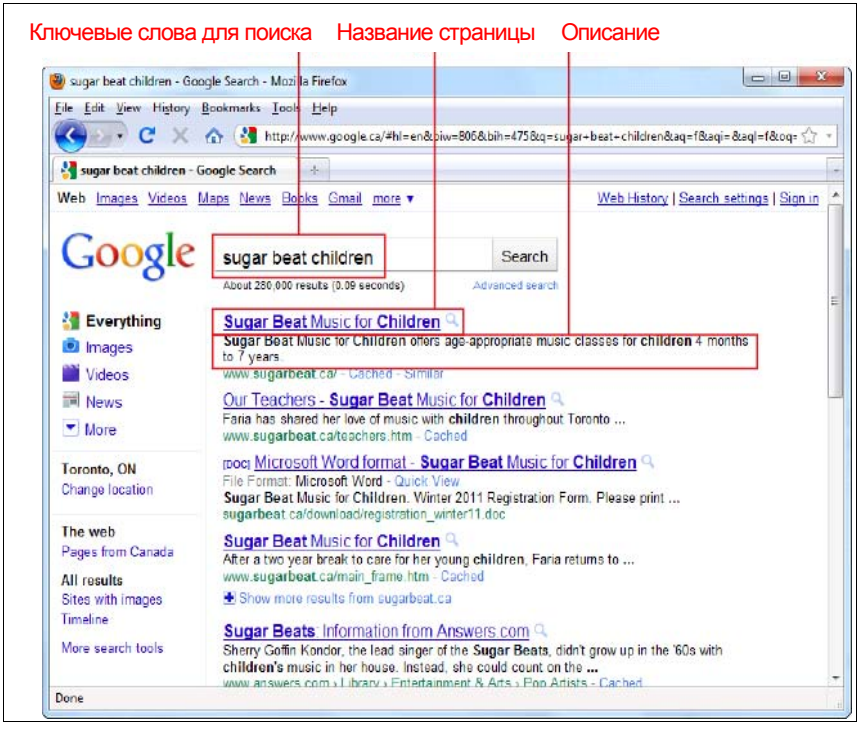


Рис. 11.1. Интересовались когда-нибудь, откуда берется информация, которую вы видите в списках результатов поиска? Подчеркнутая ссылка в данном примере (Sugar Beat Music for Children) — это название Web-страницы, найденной поисковой машиной. Она извлекла описание сайта (показанное под названием) непосредственно из одного из элементов страницы, содержащего скрытые метаданные (см. *разд. "Вставляйте описание страницы" данной главы*)

Давать всем страницам одинаковые названия, например, имя компании, — это распространенная ошибка новичков. Такое название подходит для исходной или начальной страницы, но не для остальных страниц вашего сайта. Хорошее название всегда описывает назначение страницы. Например, у музыкальной школы Cacophony Studios могут быть страницы с такими названиями: Our Teachers (Наши учителя), The Cacophony Studios Difference (Преимущество Cacophony Studios), Signing Up for Music Class (Запись в музыкальный класс), Payments and Policies (Платежи и обязательные процедуры) и т. д. (Если вам действительно нужно вклю-

чить название компании в заглавие, помещайте его после описательного текста, например, Our Teachers — Cacophony Studios.)

> ***Совет***
>
> Если посетитель создает закладку на вашей странице, текст в меню закладок передается из элемента `<title>`, расположенного на вашей странице. Помните об этом и воздерживайтесь от длинных названий. "Ketchup Crusaders — Because ketchup isn't just for making food tasty" (Борцы за кетчуп, т. к. кетчуп не только придает вкус еде) — одно из самых длинных названий, которое можно себе позволить, но и это рискованно. С другой стороны, в названия следует включать важную информацию; названия Welcome (Добро пожаловать) или Untitled 1 (Безымянный 1) (любимый вариант имени в Expression Web) не слишком полезны.

Текст названия важен еще и потому, что его ищут поисковые машины. В действительности они придают ему больший вес, чем тексту на странице. Поэтому если кто-то выполнит поиск с ключевыми словами, совпадающими с вашим названием, с вероятностью пятьдесят процентов ваша страница получит лучшее место в результатах поиска. (Но не пытайтесь играть с системой, задавая слишком длинные названия, т. к. поисковые машины могут начать игнорировать их.)

Вставляйте описание страницы

Наряду с названием у каждой Web-страницы должна быть еще одна базовая порция данных: описание или краткое резюме страницы. И опять у текста описания два назначения. Во-первых, поисковые машины придают ему дополнительный вес при поиске. Во-вторых, они часто отображают описание на странице с результатами поиска, как показано на рис. 11.2.

Для вставки описания нужно применить нечто, именуемое метаэлементом (также известное как метатег). Фактически метаэлементы содержат скрытую информацию, которая не отображается в браузере посетителя, и могут передать важные сведения о контенте вашего сайта кому-либо еще (например, автоматическому поисковому боту).

> ***Примечание***
>
> Забавное замечание для любителей этимологии и компьютерных фанатов: термин "метаэлемент" означает "элемент *о*" т. е. "элементы, предоставляющие информацию *о* вашей Web-странице".

Все метаэлементы помещаются в раздел страницы `<head>`. Далее приведен пример метаэлемента, присваивающего описание или аннотацию Web-страницы.

```
<!DOCTYPE html>

<html>
<head>
  <meta name="description"
    content="Noodletastic offers custom noodle dishes made to order." />
```

```
  <title>Noodletastic</title>
</head>

<body>...</body>

</html>
```

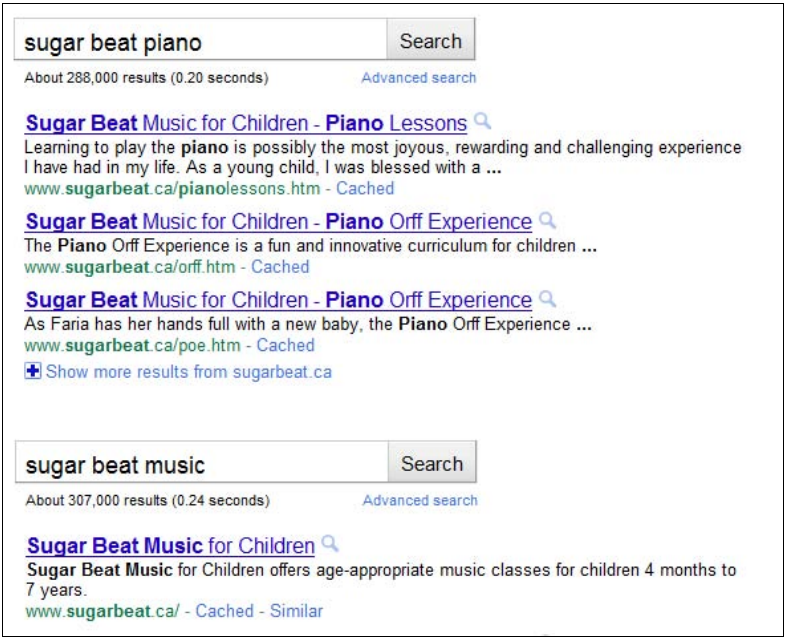


Рис. 11.2. Если поисковая машина находит соответствие ключевым словам во фрагменте вашей страницы, она отображает этот фрагмент (вверху). Но если вы введете в поле поиска часть имени домена или поисковая машина не найдет контента, который кажется ей подходящим, она в этом случае отобразит описание страницы (внизу)

Все метаэлементы выглядят более или менее одинаково. Имя элемента — `<meta>`, атрибут `name` обозначает тип метаэлемента, а атрибут `content` предоставляет значимую информацию. Теоретически вы можете помещать в метаэлемент любую информацию. Например, некоторые программы редактирования Web-страниц вставляют метаэлементы, сообщающие о том, что именно эти программы создали ваши страницы (не беспокойтесь, когда вы познакомитесь с метаэлементами, вы сможете узнавать эти безвредные пометки и легко удалять их). Страница может использовать метаэлемент для записи имен своих создателей или дату последнего изменения страницы.

Совет

Несмотря на то что вы можете втиснуть большой объем информации в ваше описание, имеет смысл ограничить его парой четких предложений длиной не более 50 слов. Хотя ваше описание и появится на странице с результатами поиска, читатели увидят лишь его начало, за которым в месте обрыва следует многоточие (...).

УГОЛОК ВОСПОМИНАНИЙ

Метаэлемент с ключевыми словами

Давным-давно, когда жизнь была проще, и нигерийским джентльменам никогда не требовалось переводить большие суммы денег, у поисковых машин была простая задача. Многие из них игнорировали большую часть Web-страницы и уделяли особое внимание метаэлементу `keywords`. Затем, когда кто-нибудь выполнял поиск, поисковая машина просто искала страницы, у которых были искомые термины в списке ключевых слов, и помещала их на страницу результатов поиска.

Далее приведен пример типичного списка ключевых слов. Он содержит около 25 слов или фраз, лучше всего представляющих Web-сайт и отделенных друг от друга запятыми.

```
<meta name="keywords"
content="Noodletastic, noodles, noodle, pasta,
delicious, Italian food, fantastic noodles,
ramen, custom, made-to-order, dishes, organic,
whole-wheat, spelt" />
```

В наши дни большинство поисковых машин игнорирует списки ключевых слов, т. к. они печально известны злоупотреблениями (многие Web-разработчики набивают свои списки ключевых слов сотнями слов, часть которых лишь косвенно связана с контентом сайта). Поисковые машины вроде Google применяют более очевидный подход — они просматривают все слова на Web-странице и обращают особое внимание на слова, встречающиеся чаще других, слова в заголовках и т. д. Большинство Web-специалистов утверждает, что списки ключевых слов изжили себя, и многие не тратят время на включение их в свои страницы.

Предоставляйте замещающий текст ко всем своим изображениям

Поисковая машина извлекает информацию из разных частей вашей страницы. Одна из часто пропускаемых деталей — альтернативный или замещающий текст, отображаемый браузером при невозможности извлечь изображение. Как вы узнали из *разд. "Альтернативный текст" главы 7*, этот текст задается в атрибуте `alt` элемента `<img>`.

```
<img src="fish0123.jpg" alt="rainbow trout" />
```

Поисковые машины обращают внимание на замещающий текст. Если Web-поиск использует ключевые слова, которые есть в вашем замещающем тексте, у вашей страницы увеличиваются шансы оказаться выше в результатах поиска. Шансы еще возрастают, если поисковик ищет картинки. Например, Google использует замещающий текст как основу для своего средства поиска изображений на сайте **http://images.google.com**. Если у вас нет замещающего текста, Google вынужден догадываться о чем картинка, исследуя близлежащий текст, что менее надежно.

Применяйте информативный текст для ссылок

Известный тег `anchor` может включать в себя любой фрагмент текста. Поисковые машины, например, Google придают этому фрагменту, подчеркнутому и выделен-

ному синим цветом, особый вес. Но это мало поможет, если вы все испортите, воспользовавшись следующей ссылкой.

```
To learn more about elephants, <a href="ElephantStories.htm">
click here</a>.
```

Приведенная далее лучше спроектированная ссылка расскажет поисковым машинам гораздо больше о вашем сайте.

```
To learn more about elephants, visit our
<a href="ElephantStories.htm">Elephant Stories</a> page.
```

Стоит также убедиться, что у Web-страниц, на которые вы ссылаетесь (в данном случае, ElephantStories.htm), хорошие, удобочитаемые названия.

Не пытайтесь жульничать

Есть немало вредных трюков, которые умелые Web-разработчики обычно применяют для того, чтобы обхитрить поисковые машины (или хотя бы попытаться). Например, они могут вставить огромное количество ключевых слов, но скрыть текст так, что он будет невидим на странице (белый шрифт на фоне белого цвета — один из возможных вариантов, но есть и другие хитрости на основе таблиц стилей). Другой прием — создание страниц, не являющихся на самом деле частью вашего Web-сайта, но хранящихся на вашем сервере. Вы можете заполнить их повторяющимися ключевыми словами. Для реализации этого трюка применяется короткий код на языке JavaScript, гарантирующий перенаправление на входную точку вашего сайта людей, случайно набредших на такую страницу, а поисковые машины при этом начинают обрабатывать ключевые слова.

Некоторые из этих приемов могут показаться соблазнительными уединенным сайтам (и их владельцам), но лучше всего отказаться от них совсем. Прежде всего они порождают ряд новых трудностей и технических проблем, которые могут отнять у вас много времени. Но еще более существенно то, что поисковые машины узнают об этих хитростях почти сразу после того, как Web-разработчики их придумали. Если поисковая машина уличит вас в их применении, она накажет ваш сайт (его ранг в результатах поиска снизится) или может полностью отлучить его, отнеся к разряду мусорных контейнеров в Web. (Такое случается даже с тяжеловесами. Например, сайт сети универмагов J. C. Penney получил существенное снижение ранга из-за применения нескольких нечистоплотных приемов оптимизации поиска в сезон Рождества в 2010 г.)

Если соблазн все еще остался, помните: многие из этих трюков просто не срабатывают. Во времена зарождения Web-пространства примитивные поисковые машины присваивали сайту более высокий ранг в зависимости от количества встретившихся ключевых слов, но современные поисковые машины, такие как Google, используют гораздо более сложные системы ранжирования страниц. Громадная куча скрытых ключевых слов ни на йоту не переместит вас вверх в списке результатов поиска.

Регистрация в каталогах

Каталоги — это реестры сайтов с возможностями поиска, отличающиеся лишь тем, что создают их не программы, а люди. Это значит, что небольшая армия сотрудников скрупулезно собирает коллекцию сайтов, аккуратно рассортированных по категориям. Достоинство этих каталогов в хорошей организации. Пара щелчков кнопкой мыши может предоставить вам, например, полный перечень местных калифорнийских газет. Бесспорный недостаток их в том, что каталоги эти гораздо меньше каталогов полнотекстового поиска. Это означает, что каталоги не слишком полезны для тех, кто ищет трудно формулируемую информацию, с трудом попадающую в определенную категорию, например, список слов английского языка, в которых чаще всего допускаются орфографические ошибки. Спустя годы, когда Web разросся до значительного размера, каталоги начали игнорировать, и люди, охотящиеся за информацией, чаще всего стали применять средства полнотекстового поиска, такие как Google и Yahoo.

Итак, с учетом того, что каталоги — просто бедные родственники полнотекстовых поисковых машин, почему вы должны вообще думать о них? По двум причинам. Во-первых, некоторые Web-посетители все еще пользуются каталогами, хоть и не так часто, как машинами полнотекстового поиска. И во-вторых, некоторые поисковые машины (включая Google) обращают внимание на реестры каталогов и стараются повысить ранг сайтов, встреченных в верхних строках определенных каталогов. Попадание в нужные каталоги может помочь в продвижении к начальным строкам списка результатов полнотекстового поиска. И так же, как при приеме в колледж, для включения в каталог нужно подать заявление, о котором вы узнаете в следующем разделе.

Каталог Open Directory Project

Самый важный каталог для регистрации вашего сайта — Open Directory Project (ODP) на сайте **http://dmoz.org**. ODP — это громадный многолетний каталог Web-сайтов со штатом из тысяч добровольных редакторов, просматривающих заявления в бесчисленных категориях. ODP — не самый популярный Web-каталог (в настоящее время первенство принадлежит Yahoo), но за кадром другие поисковые системы им пользуются. Между прочим, компания Google основала на ODP свою службу каталогов (**http://directory.google.com**).

> ***Примечание***
>
> В последние годы ODP завалили запросами на регистрацию. В некоторых категориях до рассмотрения сайта могут пройти месяцы и даже годы. И существует не одна отвратительная история о редакторах ODP, требовавших взятки за включение Web-сайта в каталог. Лучший совет таков: когда ваш Web-сайт будет готов, выполните описанные в этом разделе действия, необходимые для подачи заявления в ODP. Не паникуйте, если ваш сайт не будет зарегистрирован, т. к. у поисковых машин есть множество других способов, позволяющих найти и ранжировать ваши страницы.

Прежде чем подавать заявку в ODP, потратьте время и убедитесь, что она подготовлена правильно. Некорректная заявка может привести к тому, что ваш сайт во-

обще не будет зарегистрирован. Полное описание правил ее заполнения можно найти на странице **http://dmoz.org/add.html**, а далее перечислены основные рекомендации.

- Не представляйте ваш сайт более одного раза.
- Не представляйте ваш сайт в несколько категорий.
- Не представляйте более одной страницы или раздела вашего сайта (если только у вас нет веских причин, таких как наличие очевидно отличающихся разделов).
- Не представляйте сайты, содержащие "противозаконный" контент. По определению ODP оскорбительным контентом называется, например, порнография, клеветнический контент или материал, защищающий противозаконную деятельность — вы знаете, кого я имею в виду.
- Вычистите все потерянные ссылки, устаревшие данные и другие рискованные приметы, которые могут заставить редактора предположить, что ваш сайт не задержится здесь надолго.
- Представляя заявку на ваш сайт, опишите его тщательно и точно. Не рекламируйте его. Другими словами, фраза "Ketchup Masters — производитель деликатесного кетчупа" приемлема. А фраза "Ketchup Masters — лучший в Web-мире сайт, посвященный пищевым продуктам, — газета "Louisville Times" утверждает, что его нельзя не посетить" — нет.
- Не представляйте незавершенный сайт. Ваша страница "В разработке" не будет зарегистрирована.

Следующий ваш шаг — провести некоторое время на сайте ODP и найти единственную самую подходящую категорию для вашего Web-сайта (рис. 11.3).

После того как лучшая категория найдена, щелкните кнопкой мыши ссылку **Suggest URL** (Предложить URL) в верхней части страницы и заполните регистрационную форму. В ней запрашивается ваш URL, название вашего сайта, краткое его описание и адрес вашей электронной почты.

> ***Совет***
>
> Если у вас есть немного свободного времени, можете предложить свою помощь в редактировании категорий сайта; просто щелкните кнопкой мыши ссылку **Become an Editor** (Стать редактором). И даже если у вас нет редакторских амбиций, почему бы не просмотреть пособие по редактированию по адресу **http://dmoz.org/guidelines/**, чтобы лучше представлять себе, что творится в головах редакторов ODP, и как они оценивают вашу заявку.

После того как заявка подана, ничего не остается, кроме как ждать (и попробовать зарегистрировать ваш сайт в других каталогах и поисковых машинах, обсуждавшихся в этой главе). Если спустя месяц ваш сайт не оказывается в списке, и вы не получали электронное письмо с описанием возникших с сайтом проблем, пора связаться с редактором категории. По электронной почте вежливо задайте вопрос: почему ваш сайт не был включен в список каталога, укажите дату подачи заявки, ваш URL и описание своего сайта. Электронный адрес редактора категории можно найти в самом низу страницы категорий (рис. 11.4).

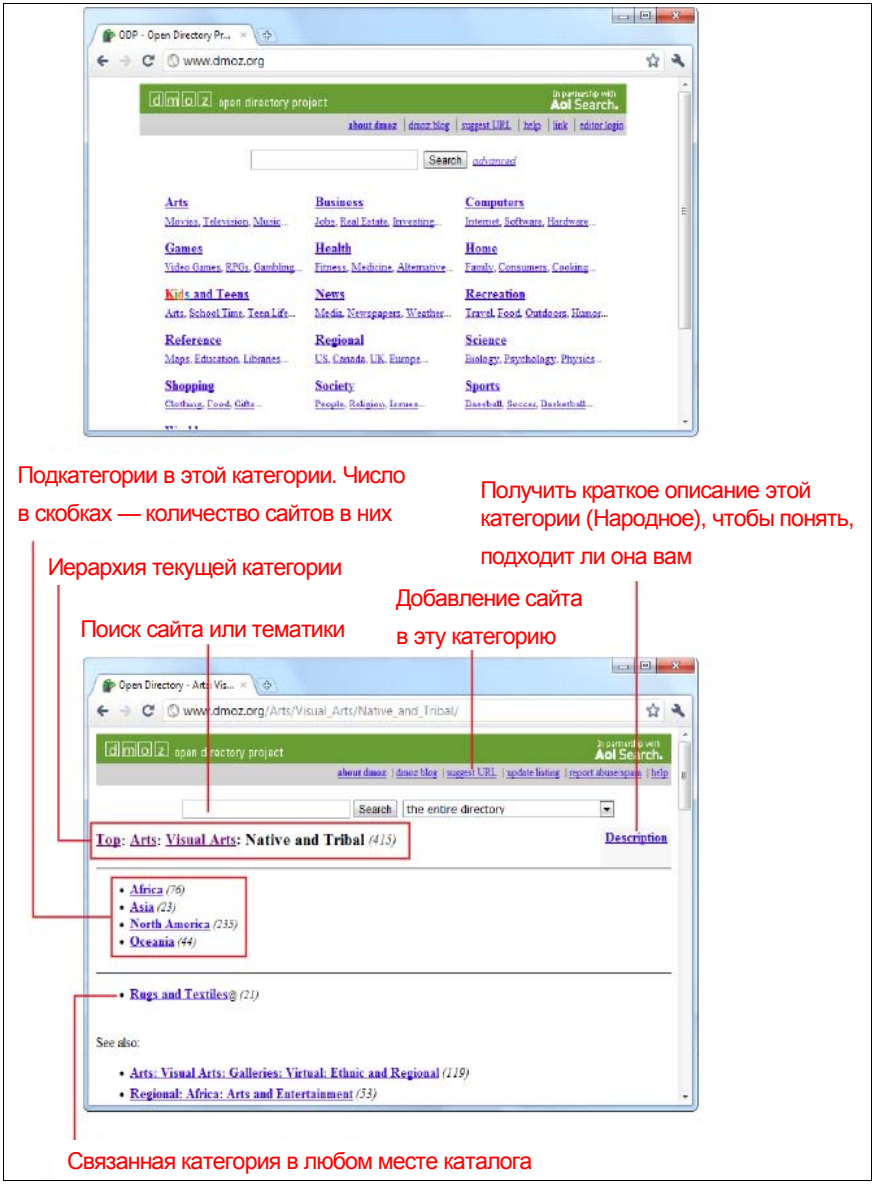


Рис. 11.3. *Вверху*: когда вы попадете на сайт ODP, то увидите группу основных категорий верхнего уровня.
Внизу: если щелкнуть мышью одну из категорий верхнего уровня вы, возможно, найдете конкретную подкатегорию, которая подойдет для хранения сведений о вашем сайте, например категорию **Arts→Visual Arts→Native and Tribal** (Искусство→Изобразительное искусство→Народное). В ней есть несколько подкатегорий (например, **Asia** (Азия) с 22 сайтами). Категории со знаком @ после их названий ссылаются на связанные категории в других местах каталога

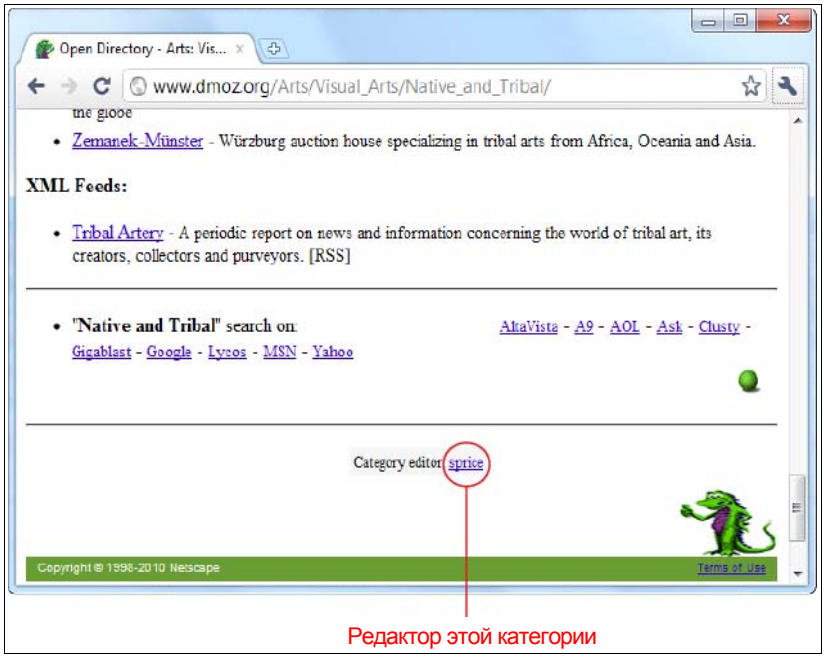


Рис. 11.4. Щелкните кнопкой мыши имя редактора (sprice), чтобы выяснить, кто он такой, какими категориями занимается и как можно связаться с ним по электронной почте. Если это не поможет, пора идти дальше. К сожалению, ODP не всегда надежен, но есть еще десятки других способов представить ваш Web-сайт всему миру

Каталог Yahoo

ODP — отличная отправная точка, но это не единственный каталог такого назначения. Другой тяжеловес — каталог Yahoo (**http://dir.yahoo.com**). К сожалению, для включения вашего сайта в каталог Yahoo потребуется гораздо больше работы.

Во-первых, возникает вопрос денег. Если вы создаете некоммерческий сайт, вы, возможно, сможете зарегистрировать его бесплатно, но для этого может потребоваться настойчивость, электронная переписка, многочисленные заявки и немного везения. Если создан коммерческий сайт (одна из основных задач которого — зарабатывание денег), и вы хотите зарегистрировать его в каталоге U.S. Yahoo, придется ежегодно платить абонентскую плату в несколько сотен долларов. И последнее, что касается удара по вашему бюджету: если Yahoo отклонит ваш сайт, свои деньги назад вы не получите.

Для начала можно просмотреть официальное руководство Yahoo по оформлению заявки на странице **http://tinyurl.com/47quxhk**. Но гораздо больше вам понравится неофициальное описание на странице **www.apromotionguide.com/yahoo.html**, в котором обсуждаются бесплатные и платные варианты и объясняется, что на самом деле означают зашифрованные электронные послания Yahoo с отклонением вашего сайта. Если у вас коммерческий Web-сайт или вы просто не хотите принимать участие в медленном и ненадежном процессе бесплатной регистрации, следу-

ет воспользоваться сервисом Yahoo Directory Submit (раньше называвшемся Yahoo Express), который представлен на странице **https://ecom.yahoo.com/dir/submit/intro**.

После того как с каталогами покончено (или вы готовы двигаться дальше), пора обратиться к машинам полнотекстового поиска.

Совет

Мнения Web-экспертов по поводу того, стоит ли включение сайта в каталог Yahoo взимаемой абонентской платы, разделились. Лучший совет таков: список каталога Yahoo улучшит ваше положение при Web-поиске и принесет ощутимое увеличение трафика, только если вы создали высококлассный Web-сайт. Если вы попытаетесь зарегистрировать посредственный сайт, вы скорее всего получите отказ и потеряете вашу регистрационную плату.

Регистрация с помощью поисковых машин

Для большинства людей поисковые машины — единственное средство поиска информации в Web-пространстве. Если вы хотите, чтобы среднестатистический посетитель нашел ваш сайт, нужно обеспечить его присутствие в каталогах самых популярных поисковых машин и в списке результатов релевантных поисков. Эта задача труднее, чем кажется, т. к. Web-пространство заполнено миллионами сайтов, правдами и неправдами отстаивающих свое место. Чтобы быть замеченным, нужно потратить время на разработку вашего сайта и улучшение его внешнего вида. Вы также должны понять, как поисковые машины ранжируют страницы (дополнительную информацию см. в *следующей врезке*).

Неоспоримый лидер среди поисковых машин — Google (**www.google.com**). Поисковая машина Google — самое популярное средство поиска с главенствующей долей всего поискового трафика, более 70%.

Второе место сейчас принадлежит Yahoo и третье — поисковой машине Bing корпорации Microsoft. Но Yahoo для усиления своих результатов поиска негласно, за кадром, использует Bing. Это означает, что на Западе только две поисковые машины (Google и Bing) отвечают за 95% всех Web-поисков. В других странах, особенно с Web-цензурой, например, в Китае есть собственные поисковые машины.

Заставить Google и другие поисковые машины увидеть ваш сайт совсем нетрудно. За месяц с момента появления вашего сайта Google, вероятно, натолкнется на него хотя бы однажды, как правило, следуя по ссылке с другого сайта или из каталога ODP. Google принимает во внимание внешние ссылки, когда рассчитывает ранг сайта, поэтому, чем больше сайтов ссылается на вас, тем вероятнее ваше появление в чьих-то результатах поиска.

Если не терпится, или вы решили, что поисковая машина Google не обращает на вас внимания, можно представить себя явно с помощью регистрационной формы на странице **www.google.com/addurl** (рис. 11.5). Для того чтобы завладеть вниманием Bing, отправляйтесь на страницу **www.bing.com/webmaster/SubmitSitePage.aspx**. Большинство поисковых машин предлагают похожую форму для регистрации.

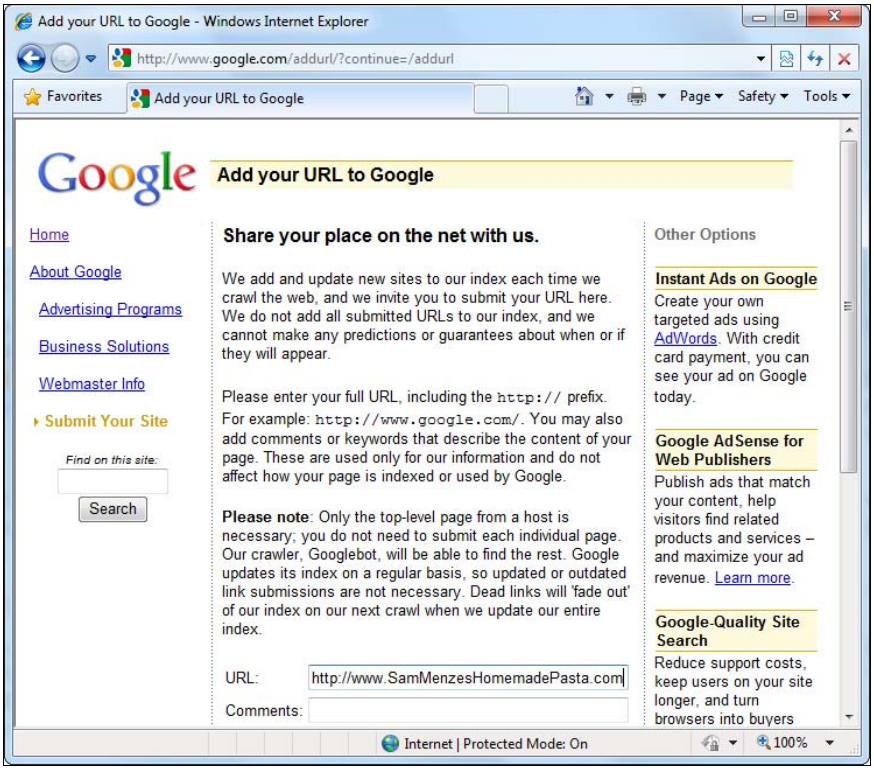


Рис. 11.5. Раздел комментариев на этой странице можно спокойно пропустить, но убедитесь, что в начало URL вашей Web-страницы вы включили префикс http://

Только проверьте, где именно вы регистрируетесь, чтобы случайно не представить заявку на регистрацию сайта одной и той же поисковой машине несколько раз.

НА ПРОФЕССИОНАЛЬНОМ УРОВНЕ

Как работает система PageRank в Google

Google использует для ранжирования разных Web-страниц рейтинговую систему PageRank. Она применяется не для *формирования* результатов поиска, а для их *упорядочивания*. Когда вы выполняете поиск с помощью Google, поисковая машина извлекает все сайты, соответствующие ключевым словам вашей строки поиска. Затем она выполняет упорядочивание результатов в соответствии с рангом (PageRank) каждой страницы.

Основная идея системы ранжирования страниц состоит в том, что ценность вашего сайта определяется группой других Web-сайтов, ссылающихся на него. Есть несколько золотых правил.

- Чем больше сайтов ссылается на вас, тем лучше.
- Ссылка с более популярного сайта (сайта с высоким рангом) более ценна, чем ссылка с менее популярного сайта.
- Чем больше ссылок есть у сайта, тем меньше ценность каждой из них. Другими словами, если кто-то ссылается на ваш сайт и лишь на небольшое количество других сайтов, такая ссылка значима. Если же кто-то ссылается на ваш сайт и на сотни других сайтов, ценность такой ссылки очень низка.

Несмотря на то, что Google постоянно подстраивает свое тайное средство ранжирования страниц, Web-специалисты тратят часы, стараясь проанализировать его. Узнать больше о том, как работает PageRank (в общих чертах), можно на странице **www.markhorrell.com/seo/pagerank.html**. Более строгое представление о математическом аппарате, используемом в алгоритме PageRank, можно найти на странице **http://en.wikipedia.org/wiki/PageRank**.

Повышение ранга в результатах поиска

Вы скоро поймете, что попасть в индекс Google нетрудно, а вот оказаться замеченным исключительно тяжело. Например, вы представляете сайт **www.SamMenzesHomemadePasta.com**. Чтобы проверить, есть ли вы в Google, попробуйте в высшей степени конкретный поиск, указывающий прямо на ваш сайт, такой как "Sam Menzes Homemade Pasta". Он определенно приведет к вашему порогу. Теперь поищите просто "Homemade Pasta". С вероятностью 50% вы не появитесь в верхних 10 и даже 100 строчках результатов.

Как же создать сайт, который с большой вероятностью найдет случайный искатель? Короткого ответа нет. Просто помните, что секрет получения высокого ранга при поиске кроется в присвоении высокого ранга странице (PageRank), а получение высокого PageRank целиком зависит от связей. Для того чтобы выделиться, вашему Web-сайту нужно использовать ссылки совместно с лидирующими в вашей категории сайтами. Это поможет приобрести людей, обсуждающих ваш сайт в сети, будь то Twitter, Facebook или персональный блог. В следующей главе вы получите множество советов по продвижению Web-сайта.

Google не сообщит вам ранг страницы вашего Web-сайта. (Они предлагают вам сосредоточиться на создании фантастического сайта, а не зацикливаться на номере.) Однако большинство Web-дизайнеров хотят знать, какое место они занимают, ведь только так они могут следить за тем, как меняется положение сайта со временем. Хотя вы не сможете получить точное значение PageRank, которое Google использует в своих алгоритмах поиска, вы *можете* получить приблизительное значение ранга страницы, которое даст вам общее представление об эффективности вашего Web-сайта. Это приблизительное значение PageRank основано на реальном значении, но обновляется только дважды в год и представляет собой значение в диапазоне от 1 до 10. (При прочих равных условиях Web-сайт с рангом 10 появится гораздо раньше в чьих-либо результатах поиска, чем страница с рангом 1.)

Есть два способа получения приближенного PageRank. Во-первых, можно воспользоваться бесплатным приложением Google Toolbar (**www.google.com/toolbar**), которое вставляется в окно браузера и содержит кнопку **PageRank**. Но вы должны явно включить это средство, как описано на странице **http://tinyurl.com/64bjmtd**. Более простой способ — использовать неофициальный Web-сайт проверки ранга страницы, например, тот, что показан на рис. 11.6. Для того чтобы получить приближенное значение PageRank, вы просто вводите свой URL и щелкаете мышью кнопку.

Если хотите углубиться в изучение практически важных подробностей поисковой оптимизации (Search Engine Optimization, SEO), советую стать постоянным читате-

лем Web-сайтов **www.webmasterworld.com** и **www.searchengineland.com**. Там вы найдете статьи и форумы профессионалов с обсуждением хороших, плохих и самых сомнительных приемов, которые вы сможете опробовать для привлечения к себе внимания.

Совет

Вполне вероятно стать одержимым ранжированием поисковых машин, но есть хорошее правило, проверенное временем: тратьте на попытки повысить ранг, присваиваемый поисковыми машинами, не больше времени, чем на улучшение своего Web-сайта. В долгосрочной перспективе единственный способ увеличить реальную популярность — превратить свой сайт в один из лучших в своей категории.

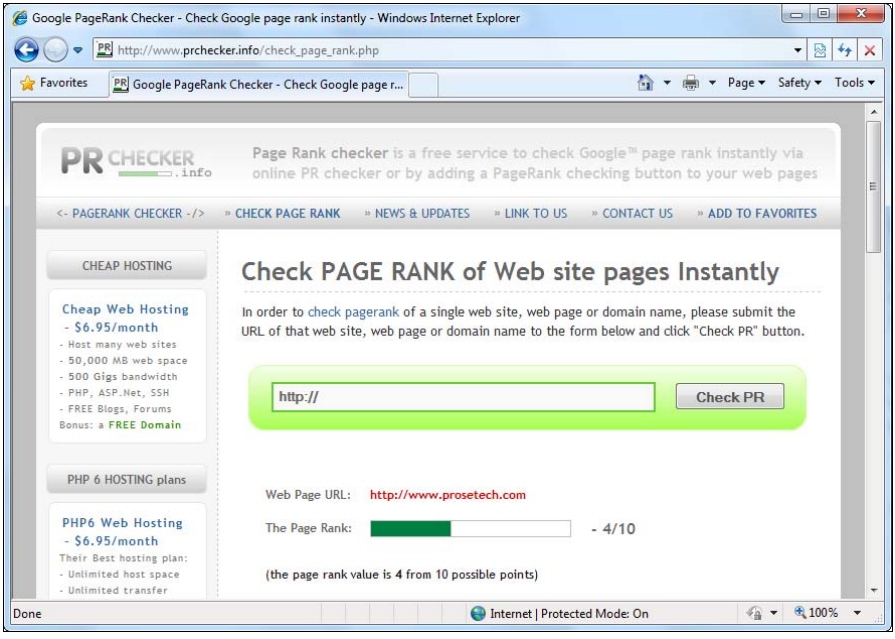


Рис. 11.6. Для быстрого получения ранга любого Web-сайта воспользуйтесь **www.prchecker.info** (или с помощью поиска Google найдите другое средство подсчета PageRank). В данном случае сообщается о том, что сайт-середнячок **www.prosetech.com** набрал 4 очка по 10-балльной системе

Как спрятаться от поисковых машин

В редких случаях создаются страницы, появление которых в результатах поиска нежелательно. Самая частая причина — включение информации, которой вы хотите поделиться только с узким кругом друзей, например последними электронными купонами компании Amazon. Если Google проиндексирует ваш сайт, объявятся тысячи посетителей, рассчитывающих найти купон со скидкой, и до конца месяца забивающие всю вашу полосу пропускания. Другой причиной может стать полусекретная информация на странице, с которой вы не хотите знакомить других людей, например рассказ о том, как вы украли у шефа десяток степлеров. Если вы попадаете в последнюю категорию, будьте очень осторожны. Скрыться от поисковых

машин — это самая простая из ваших проблем, т. к. сайт, появившийся в Web, *будет* обнаружен. А после того как он будет обнаружен, исчезнуть ему уже не удастся (см. далее *врезку "Секреты профессионала. Web-долговечность"*).

Для минимизации видимости вашего сайта, а возможно, и полного его сокрытия от поисковых машин вы можете выполнить, по крайней мере, одну процедуру. Чтобы понять, как она работает, вспомните, что поисковые машины выполняют поиск в три этапа. На первом программа-робот ползает по Web-просторам, загружая сайты. Вы можете несколькими способами заставить этого робота не индексировать ваш сайт или его часть.

Для того чтобы спрятать от робота одну страницу, вставьте на нее метаэлемент `robots`. Укажите для атрибута `content` значение `noindex`, как показано далее.

```
<meta name="robots" content="noindex" />
```

Напоминаю, что, как и все метаэлементы, вы помещаете этот элемент в раздел `<head>` вашего HTML-документа.

В качестве альтернативного варианта, для того чтобы заставить роботов индексировать текущую страницу, но не переходить ни по каким ее ссылкам, можно использовать значение `nofollow`.

```
<meta name="robots" content="nofollow" />
```

Если хотите закрыть большие порции вашего сайта, лучше создать специальный файл с именем robots.txt и поместить его в папку верхнего уровня для вашего сайта. Программа-робот проверяет этот файл перед тем, как идти куда-либо еще. В файле robots.txt задаются правила.

Если вы хотите помешать роботу индексировать любую часть вашего сайта, вставьте в файл robots.txt следующие строки.

```
User-Agent: *
Disallow: /
```

Строка `User-Agent` идентифицирует тип программы-робота, к которой вы обращаетесь, а звездочка представляет все типы роботов. В строке `Disallow` указывается, какая часть сайта закрыта для посещений; единичный прямой слэш обозначает весь сайт целиком.

Для того чтобы отгородить на вашем сайте только подпапку Photos, используйте следующее правило (убедитесь в том, что строчные и заглавные буквы в имени папки воспроизведены точно).

```
User-Agent: *
Disallow: /Photos
```

Можно заставить робота не индексировать контент определенного типа (например, изображения), применив следующие правила.

```
User-Agent: *
Disallow: /*.gif
Disallow: /*.jpeg
```

Как показывает данный пример, в файл robots.txt можно вставить одно за другим столько правил `Disallow`, сколько захотите.

Но помните, что файл robots.txt — это только набор инструкций для роботов поисковых машин, а не способ управления доступом. Другими словами, он подобен надписи "Никакой рекламы" на вашем почтовом ящике: она будет действовать, только если рекламные агенты обращают на нее внимание.

Совет

Вы можете узнать больше о программах-роботах, включая способы общения с ними при посещении вашего сайта и задание ограничений для роботов определенных поисковых машин, на Web-сайте **www.robotstxt.org**.

НА ПРОФЕССИОНАЛЬНОМ УРОВНЕ

Web-долговечность

Вы наверняка слышали разговоры о постоянно меняющейся природе Web. Возможно, вы боитесь, что ссылки, созданные вами сегодня, завтра уже будут указывать на заброшенные сайты или исчезнувшие страницы. Но на самом деле сейчас назревает совершенно иная проблема — копии старых сайтов, которые просто так не исчезают.

После того как вы поместили свою работу в Web, вы навсегда теряете над ней контроль. Эта простая мысль приводит к следующему совету: всегда убеждайтесь в том, что на вашем сайте нет ничего противозаконного, нарушающего авторские права, ставящего кого-либо в неловкое положение или способного сделать вас изгоем. Материал, единожды выпущенный вами в Web-пространство, может остаться там навсегда.

Представьте себе, что вы случайно открыли торговый секрет вашей компании, касающийся жевательной резинки со вкусом и ароматом моркови. Через несколько недель любознательный покупатель попадает по ссылке на ваш сайт. Вы осознали свою ошибку и удалили страницы с Web-сервера. Но действительно ли вы устранили проблему?

Допустим, что робот Google недавно посетил ваш сайт (что более чем вероятно), теперь у Google есть копия вашего старого сайта. Хуже того, пользователи могут получить эту *кэшированную* (сохраненную) копию от Google, если им знакомо ключевое слово *кэш*. Например, если URL создавшей проблему страницы — **www.GumLover.com/newProduct.htm**, ушлый пользователь Google может получить старую копию вашей страницы с помощью строки поиска "cache:www.GumLover.com/newProduct.htm". (Менее осведомленные посетители также могут наткнуться на кэшированную страницу, щелкнув мышью ссылку **Cached** (Сохраненная копия), которая появляется в списке результатов поиска Google после каждой выведенной страницы.) Хотите верьте, хотите нет, но именно этот прием использовался раньше для получения случайно просочившейся информации, начиная со сплетен и заканчивая лицензионными ключами программного обеспечения.

Вы можете попытаться как можно быстрее удалить вашу страницу из кэша Google, используя средство удаления URL на странице **www.google.com/webmasters/tools/removals**. Но даже если оно сработает, у вас останется вопрос: как узнать, сколько поисковых машин успели сделать копию вашей работы? Любопытные пользователи, заметившие, что вы удалили информацию с вашего сайта, посетят эти поисковые машины и скопируют подробности на свои собственные сайты, сделав практически невозможным удаление многочисленных следов вашей ошибки. Существуют даже каталоги, предназначенные для сохранения старых сайтов для потомков (см. подборку Wayback Machine (Машина в прошлое) на сайте **www.archive.org**).

Google Webmaster Tools

Если у вас лишь смутные представления о взаимоотношениях вашего Web-сайта с Google, вас порадует сервис Google, способный помочь. Называется он Google Webmaster Tools (Инструменты для Web-мастеров) и служит двум целям. Во-первых, он предоставляет информацию, которая помогает понять, как Google индексирует ваши страницы. Во-вторых, он позволяет настроить несколько параметров, управляющих тем, как Google трактует ваш сайт.

Для применения Google Webmaster Tools выполните следующие действия.

1. Перейдите по адресу **www.google.com/webmasters/tools** и зарегистрируйтесь с вашей учетной записью Google. Если у вас ее нет, щелкните мышью кнопку **Create an account now** (Зарегистрироваться).

   Учетная запись в Google предоставит вам доступ к ряду незаменимых сервисов Google, таких как Google Analytics (Средство Web-аналитики) (см. *разд. "Описание Google Analytics" данной главы*), Gmail (сервис электронной почты Google, основанный на Web), Google Groups (Группы Google) (см. *главу 12*), Google AdSense (Сервис контекстной рекламы) (см. *главу 14*) и Blogger (сервис для ведения блогов) (см. *главу 13*). У всех Web-разработчиков есть хотя бы одна.

2. После того как войдете в систему, зарегистрируйте ваш Web-сайт. Щелкните мышью кнопку **Add a site** (Добавить сайт), введите полный URL вашего сайта (например, **www.supermagicalpotatoes.co.uk**) и затем щелкните мышью кнопку **Continue** (Продолжить).

   Ваш Web-сайт будет вставлен в вашу учетную запись. Но прежде чем вы сможете по-настоящему просматривать данные о вашем сайте и управлять его параметрами поиска в Google, необходимо доказать, что вы — законный владелец сайта.

3. Выберите способ подтверждения вашего права собственности на Web-сайт (рис. 11.7).

   Google позволяет подтвердить ваше право собственности несколькими способами; вы можете выбрать любой из них. Как правило, самый легкий вариант — вставка предлагаемого Google метаэлемента на начальную страницу вашего Web-сайта или же выгрузка в папку вашего Web-сайта файла, предлагаемого Google. В некоторых случаях может понадобиться и более трудный способ. Например, если применяется переадресация домена (см. *разд. "Переадресация домена" главы 3*), Google будет проверять выгруженный файл и метаэлемент на Web-сайте, выполняющем переадресацию, который не является сайтом с вашим контентом. В такой ситуации надо связаться с провайдером вашего доменного имени и попросить выгрузить контрольный файл в папку вашего сайта. Или же вы можете выполнить верификацию вашего Web-сайта, вставив DNS-запись.

> ***ПРИМЕЧАНИЕ***
>
> Смею вас заверить, что неважно, какой способ верификации вы выбрали, он не повлияет на работу вашего Web-сайта. Кстати, Google советует не удалять вставленный метаэлемент (или верификационный файл, выгруженный вами). В этом случае в Google не придется снова проверять ваше право собственности.

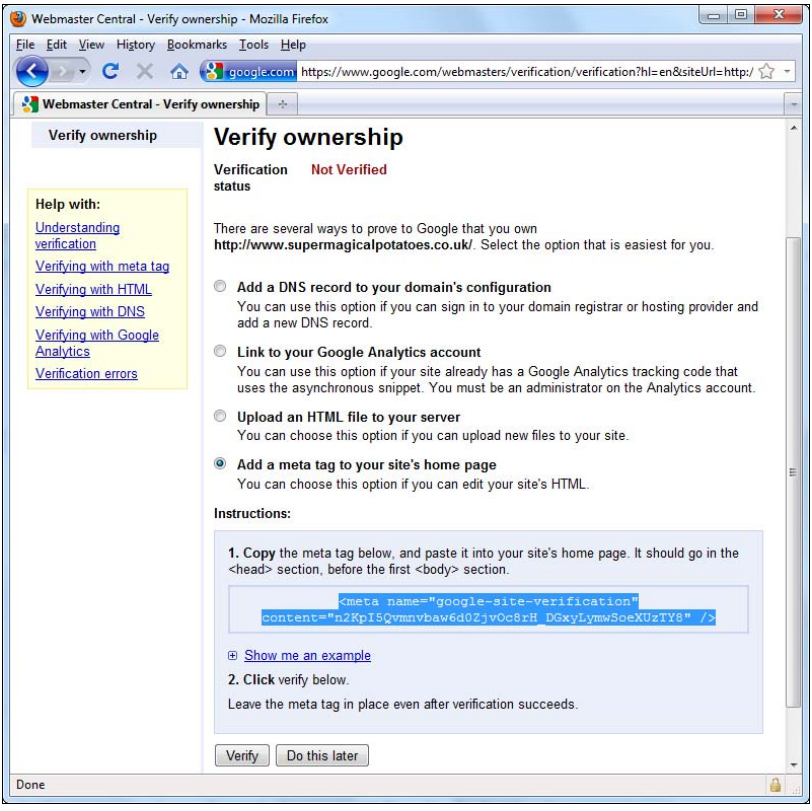


Рис. 11.7. Для большинства пользователей метод с применением метатега — самый быстрый способ встроить Google Webmaster Tools в ваш Web-сайт. Google предлагает вам фрагмент HTML, а вы просто копируете его в раздел `<body>` вашей исходной страницы (это страница index.html или index.htm, с которой Web-посетители начинают знакомство с вашим сайтом)

4. После того как вы удостоверили право собственности на сайт, щелкните мышью кнопку **Verify** (Подтвердить).

   Обычно Google успешно завершает верификацию вашего сайта через несколько секунд после того, как вы внесли соответствующее изменение (либо вставили метаэлемент, либо выгрузили верификационный файл). После того как ваше право собственности подтверждено, Google обновляет страницу и вставляет набор ссылок управления на левую половину страницы (рис. 11.8). С этого момента, когда бы вы ни зарегистрировались на странице Google Webmaster Tools, вам будут доступны все инструменты Google для просмотра вашего сайта и управления им.

Google Webmaster Tools позволяют посмотреть на ваш Web-сайт глазами Google. Вы выбираете нужное действие, щелкнув мышью одну из ссылок в разделе **Dashboard** (Инструментальная панель) на левой половине страницы. Далее перечислены некоторые возможности.

- **Site configuration | Sitemaps** (Конфигурация сайта | Карты сайта). Этот раздел поможет вам сформировать *карту сайта* — специальный файл, описывающий

структуру вашего сайта и файлы в нем. Вы можете предоставить карту сайта Google и другим поисковым машинам, чтобы они знали, что индексировать. Это особенно полезно, если у вас есть страницы, которые Google в обычной ситуации может пропустить, например, автономные страницы (не ссылающиеся на другие страницы).

- **Site configuration | Crawler access** (Конфигурация сайта | Доступ для робота). Этот раздел поможет вам создавать файлы robots.txt, которые скрывают фрагменты вашего сайта от назойливых поисковых машин, как объяснялось ранее.
- **Site configuration | Sitelinks** (Конфигурация сайта | Ссылки сайта). Ссылки сайта — это ссылки быстрого доступа, которые появляются под некоторыми строками в результатах поиска и позволяют посетителям прямо переходить на конкретную страницу на Web-сайте. К сожалению, только Google может решить, наградить ли ваш Web-сайт ссылками быстрого доступа (основываясь на популярности и структуре вашего сайта). Если Google все-таки предоставляет вам ссылки сайта, вы можете воспользоваться этим разделом, чтобы удалить любые неподходящие.
- **Site configuration | Change of address** (Конфигурация сайта | Изменение адреса). Переходите на новый URL? Этот раздел поможет переадресовать Google на ваш новый адрес, так что вы сможете забрать с собой весь ваш Web-трафик.
- **Site configuration | Settings** (Конфигурация сайта | Настройки). Этот раздел предоставляет возможность тонкой настройки нескольких разнородных параметров, определяющих, как Google трактует ваш сайт. Самый важный из них — **Geographic target** (Географический регион). Убедитесь, что вы указали страну проживания, чтобы ваш сайт появлялся в подходящих списках результатов регионального поиска.
- **Your site on the web | Search queries** (Ваш сайт в Интернете | Поисковые запросы). Здесь вы получите информацию о поисках, которые приводят пользователей Google из поисковых машин на ваш Web-сайт (рис. 11.8).
- **Your site on the web | Links to your site** (Ваш сайт в Интернете | Ссылки на ваш сайт). Этот вариант показывает, какие Web-сайты ссылаются на ваш. Это удобный способ узнать, кто обращает внимание на ваш контент.
- **Diagnostics** (Диагностика). В этом разделе вы получаете предупреждения о проблемах, обнаруженных Google во время индексирования вашего сайта, например, некорректных метаданных (см. *разд. "Вставляйте описание страницы" данной главы*), страницах с отсутствием доступа (а следовательно, с невозможностью индексации) или отвратительных вредоносных программах, притаившихся на вашем сервере.

Самые ответственные Web-разработчики рано или поздно проверяют свои сайты средствами Google Webmaster Tools. С их помощью вы всегда можете убедиться в том, что все идет гладко, т. е. что Google может получить ваши страницы, что его автоматический поисковый робот часто возвращается для поиска нового контента и что робот просматривает все страницы, предлагаемые вами.

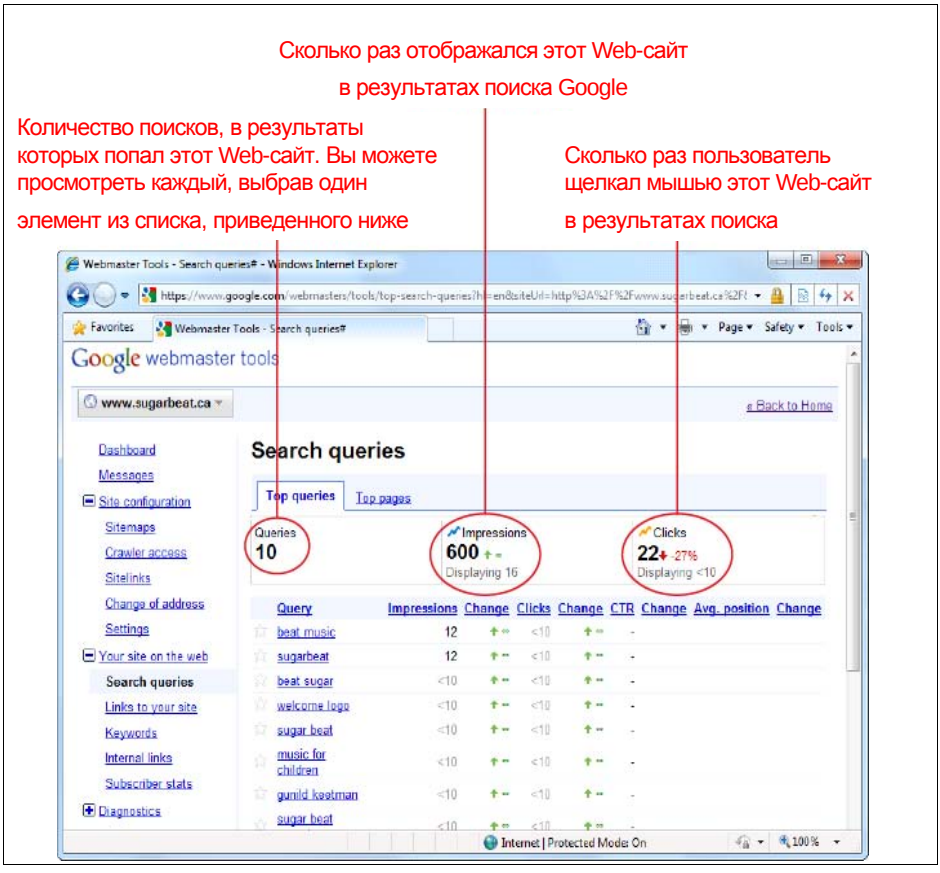


Рис. 11.8. На этом недавно запущенном сайте есть трафик. Ключевые слова поиска показывают, что несколько посетителей случайно нашли сайт (например, в результате поиска "beat music" (музыкальный бит) или "music for children" (детская музыка)), в то время как другие искали конкретное фирменное название (с помощью запросов "sugarbeat" и "sugar beat"). С помощью сервиса отслеживания Google Analytics, описанного далее, вы сможете получить более детальные статистические характеристики

Отслеживание посетителей

Как владелец Web-сайта вы примените разнообразную тактику для его продвижения. Естественно некоторые приемы будут действовать, другие — нет, а вам нужно сохранить удачные методы и отказываться от несработавших приемов. Для того чтобы сделать это грамотно, необходим способ оценки эффективности вашего сайта.

Почти все компании Web-хостинга (за исключением бесплатных Web-хостов) предоставляют средства отслеживания посетителей вашего сайта (рис. 11.9). Спросите вашего провайдера, как применять эти средства. Как правило, необходимо зарегистрироваться на сайте вашего хоста в разделе **Control Panel** (Панель управления) или **My Account** (Моя учетная запись). Вы увидите множество инструментов, найдите пиктограмму **Site Counters** (Счетчики сайта) или **Web Traffic** (Web-трафик).

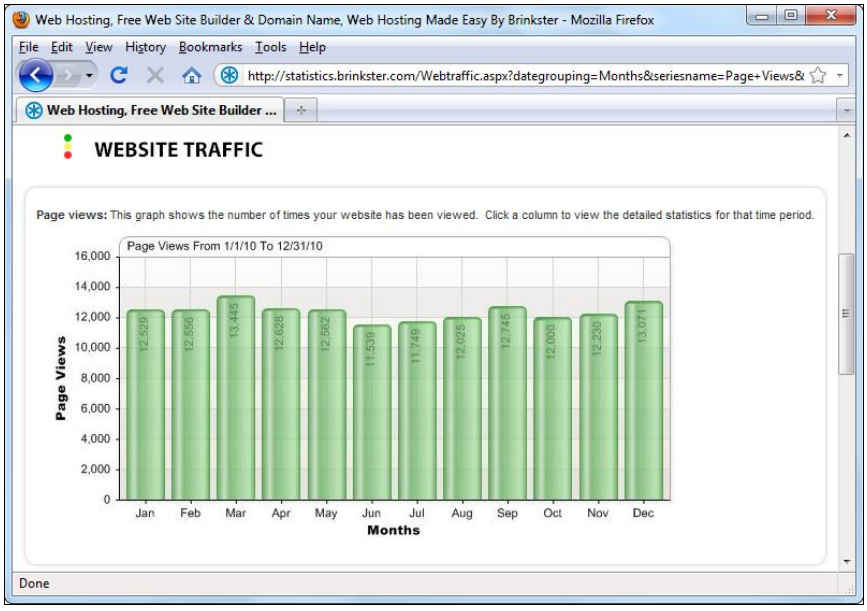


Рис. 11.9. На странице **Brinkster Page View Summary** (Сводка результатов просмотра страницы от компании Brinkster) отображено количество результатов поиска (запросов страницы) в течение данного дня. Диаграмма под этой сводкой (на рисунке не показана) дает детальное представление об общем количестве байтов данных, загруженных с вашего сайта. Важно понять, что "результат поиска" определяется как запрос любой одиночной страницы. Если один и тот же посетитель путешествует по вашему Web-сайту, запрашивая несколько страниц, будет сгенерировано несколько результатов поиска. Для выяснения количества разных посетителей вашего сайта придется применить отдельную программу анализа серверных журналов, описанную далее

У служб хостинга более высокого класса часто можно найти дополнительные средства для просмотра статистических показателей трафика вашего сайта. Некоторые хосты предлагают исходные журналы Web-серверов, хранящие подробную информацию о каждом из ваших посетителей. Она включает в себя время прихода посетителей на сайт, их IP-адреса (см. *разд. "Как браузеры анализируют URL" главы 3*), типы их браузеров, сайт, приведший их к вам, возможные ошибки, игнорируемые ими страницы и их любимые страницы и т. д. Для того чтобы разумно распорядиться этой информацией, ее придется передать в многофункциональную программу, выполняющую анализ журналов. Эти программы часто сложны и дороги. Столь же мощный, но более удобный способ — применение сервиса Web-аналитики Google Analytics, рассмотренного в следующем разделе.

Описание Google Analytics

В 2005 г. компания Google купила Urchin, одну из ведущих компаний в сфере Web-аналитики. Они преобразовали ее программы в Google Analytics и отменили внушительную ежемесячную плату 500 долларов, сделав доступ к сервису бесплатным. Сегодня сервис Google Analytics — быть может, самый лучший способ узнать, что творится на любом Web-сайте, трехстраничном сайте о танцующих хомячках или массивном каталоге рецензий на кинофильмы.

Сервис Google Analytics удивительно прост (рис. 11.10). В отличие от других инструментов анализа регистрационных журналов, он не запрашивает у вас серверные журналы или другую низкоуровневую информацию. Вместо этого сервис отслеживает всю необходимую информацию *самостоятельно*. Он хранит эту информацию бесконечно и дает возможность в любое время анализировать ее с помощью ряда классных Web-отчетов.

Для пользования Google Analytics следует на каждую страницу, которую хотите отслеживать (как правило, это все страницы вашего сайта), вставить фрагмент кода на языке JavaScript. После вставки программного кода все работает без сучка без задоринки. Когда посетитель идет на страницу вашего сайта, браузер посылает запись с запросом на множество исполинских Web-серверов Google, которые сохраняют ее для последующего анализа. Посетитель не видит этой начинки Google Analytics.

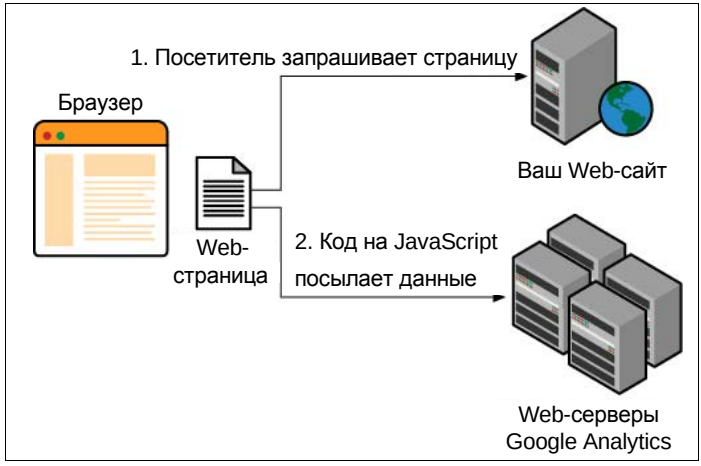


Рис. 11.10. Самое замечательное в Google Analytics то, что вам не нужно отслеживать информацию самостоятельно

Примечание

Имейте в виду, что код JavaScript — это разновидность мини-программы, которая выполняется внутри практически любого существующего браузера. В *главе 14* предлагается введение в язык JavaScript.

С помощью этой системы Google Analytics собирает данные двух видов.

- Сведения о браузерах и компьютерах ваших посетителей. Когда браузер запрашивает страницу, он предоставляет базовый набор данных. В него входит тип браузера, поддерживаемые им средства и IP-адрес компьютера, с помощью которого он подключается (IP-адрес — это цифровой код, однозначно определяющий компьютер в сети Интернет). Эти детали не содержат информации, которая вам действительно нужна: например, нет возможности определить ваши персональные характеристики, такие как имена и адреса. Но Google использует эти сведения браузера для того, чтобы выяснить дополнительные подробности. На-

пример, на основании IP-адреса он делает обоснованное предположение о географическом местоположении вашего посетителя.

- Отслеживание посетителей. Благодаря сложной системе трассировки Google Analytics может получить более интересные данные о типах посетителей. Обычно, если посетитель запрашивает две разные страницы с одного и того же сайта, невозможно установить, что эти запросы пришли от одного и того же пользователя. Но Google для однозначной идентификации каждого посетителя использует файлы *cookie* (небольшой пакет данных, хранящийся на компьютере посетителя). В результате, когда посетители щелкают ссылки кнопкой мыши и переходят со страницы на страницу, сервис Google может определить их путь следования, количество времени, проведенного ими на каждой странице, и вероятность их возврата в будущем.

Сервис Google Analytics не был бы столь полезным инструментом, если вам самим пришлось разбираться во всей этой информации. Но как вы увидите, Google не только отслеживает подробности, но и предлагает отчеты, которые помогают понять, что на самом деле означают полученные данные. Отчеты формируются с помощью удобного экранного меню, их также можно напечатать или загрузить для обработки в другую программу, например Excel, чтобы продолжить анализ.

Регистрация в Google Analytics

Зарегистрироваться в Google Analytics легко.

1. Перейдите на страницу **www.google.com/analytics** и щелкните кнопкой мыши ссылку **Sign Up Now** (Зарегистрироваться сейчас).

   Google Analytics — это всего лишь один из множества сервисов, к которым вы можете получить доступ с помощью единственной учетной записи (аккаунта) Google. Если у вас ее нет, вам предложат создать ее, указав адрес электронной почты и пароль.

2. Введите информацию о вашем Web-сайте.

   Вы должны предоставить следующие данные.

   - URL сайта, который вы хотите отслеживать (например, **www.supermagicalpotatoes.co.uk**). С помощью учетной записи Google Analytics можно отслеживать столько сайтов, сколько захотите, но пока ограничимся только одним.
   - Ваш часовой пояс. Эти данные позволят Google Analytics синхронизировать свои часы с вашими.

3. Щелкните мышью кнопку **Finish** (Готово).

   После завершения регистрации Google предложит вам поле с кодом на JavaScript, который нужно вставить на ваши страницы для запуска процесса отслеживания посетителей (рис. 11.11). В следующем пункте описывается, как это сделать.

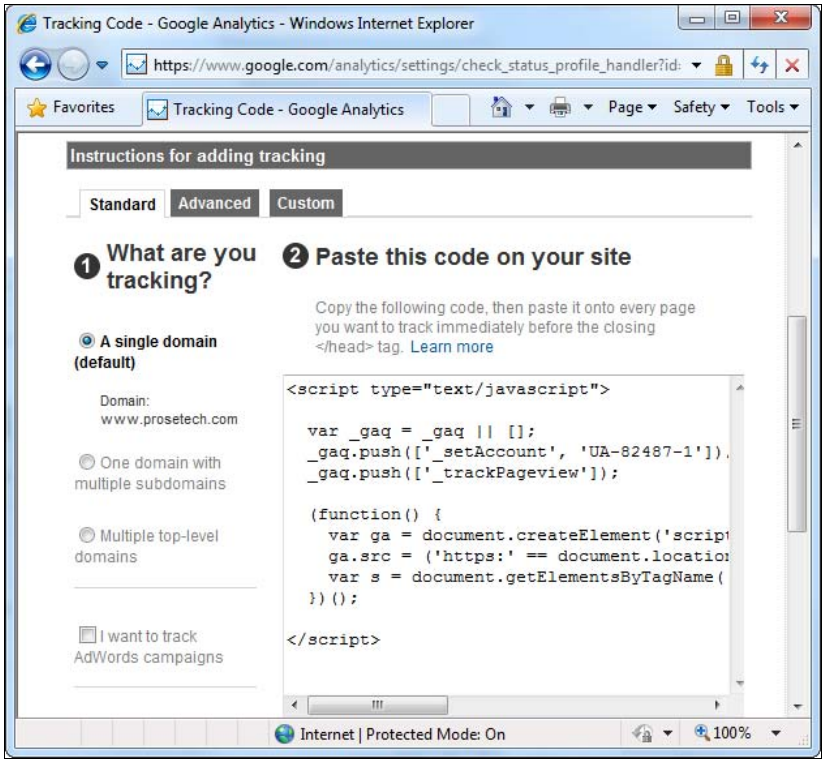


Рис. 11.11. Код Google Analytics компактный и немногословный, требующий всего несколько строк. Для получения реальных данных отслеживания он связывается с файлом на Web-серверах Google. Выделите весь код, отображенный на экране, и затем скопируйте его в ваш буфер (в большинстве браузеров для этого нужно щелкнуть выделенный текст правой кнопкой мыши и выбрать команду **Copy** (Копировать))

4. Добавьте код отслеживания на ваши Web-страницы.

   Поместите код отслеживания, предлагаемый Google, в самый конец раздела `<head>`, прямо перед закрывающим тегом `</head>`. Далее приведен пример с указанием места вставки кода на типовую Web-страницу.

```
<!DOCTYPE>

<html>
<head>
  <title>Welcome</title>
  <!-- Put the analytics code here. -->
</head>
<body>
  ...
</body>
</html>
```

   Имейте в виду, в течение нескольких лет сервис Google Analytics применял другой подход, требовавший вставки немного отличающегося кода отслеживания после тега `</body>`. Поэтому если вы посмотрите на какую-нибудь другую стра-

ницу, применяющую Google Analytics, то можете обнаружить код отслеживания в другом месте.

> ***Совет***
>
> Для достижения наилучших результатов скопируйте код отслеживания на каждую страницу вашего сайта.

5. Выгрузите новые версии ваших Web-страниц.

   После изменения всех ваших страниц выгрузите их на ваш сервер. Только после этого Google Analytics начнет отслеживание посетителей. Средства отслеживания не будут работать при запуске страниц с жесткого диска вашего компьютера.

Теперь остается ждать. В течение 24 часов Google Analytics получит достаточно информации о ваших самых последних посетителях и сможет предоставить подробные отчеты о них.

Просмотр вашего Web-трафика

Как зарегистрированный пользователь Google Analytics вы сможете зарегистрироваться и прочитать отчеты, чтобы убедиться, что все идет гладко. Если вы еще не сделали этого, самое время перейти на страницу **www.google.com/analytics**.

При первой регистрации вы увидите страницу **Analytics Settings** (Настройки Analytics), показанную на рис. 11.12. С ее помощью вы сможете добавить новые Web-сайты для отслеживания или настроить уже имеющиеся.

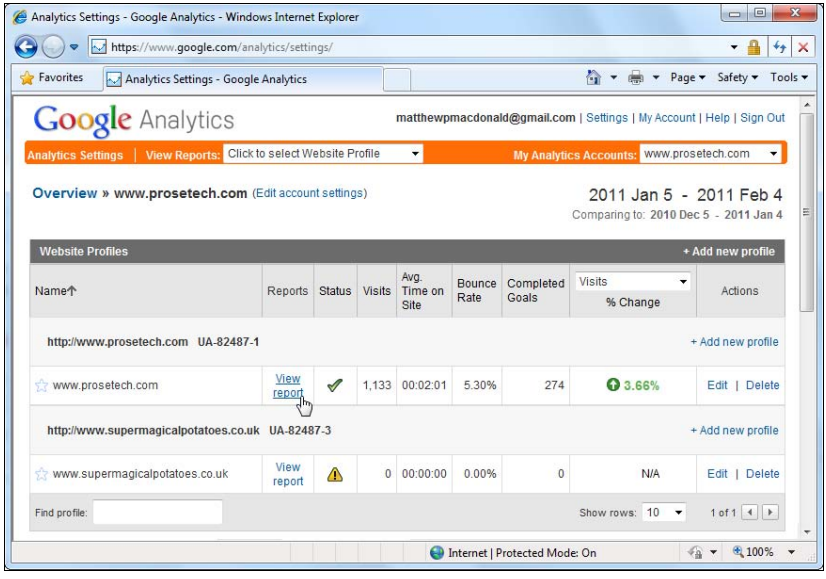


Рис. 11.12. Пример учетной записи, отслеживающей два Web-сайта, но только один из них успешно собирает данные (как показывает галочка в столбце статуса). Для просмотра отчетов можно использовать ссылку **View Report** (Просмотреть отчет), для изменения данных о вашем Web-сайте, предоставленных вами, — ссылку **Edit** (Редактировать), для удаления профиля полностью — ссылку **Delete** (Удалить)

Для того чтобы выяснить, работает ли код отслеживания Web-сайта, наведите курсор на пиктограмму в столбце **Status** (Статус). Далее описывается, как интерпретировать текст, который вы увидите.

- **Receiving Data** (Сбор данных) означает, что все в порядке. Ваши посетители переходят со страницы на страницу под недремлющим оком Google Analytics.
- **Waiting For Data** (Данные еще не поступили) означает, что код на JavaScript, предложенный Google, выполняется на ваших страницах, но информация, необходимая для составления отчетов, все еще недоступна. Обычно это сообщение вы будете видеть в первое время (от 6 до 12 часов) после регистрации нового сайта.

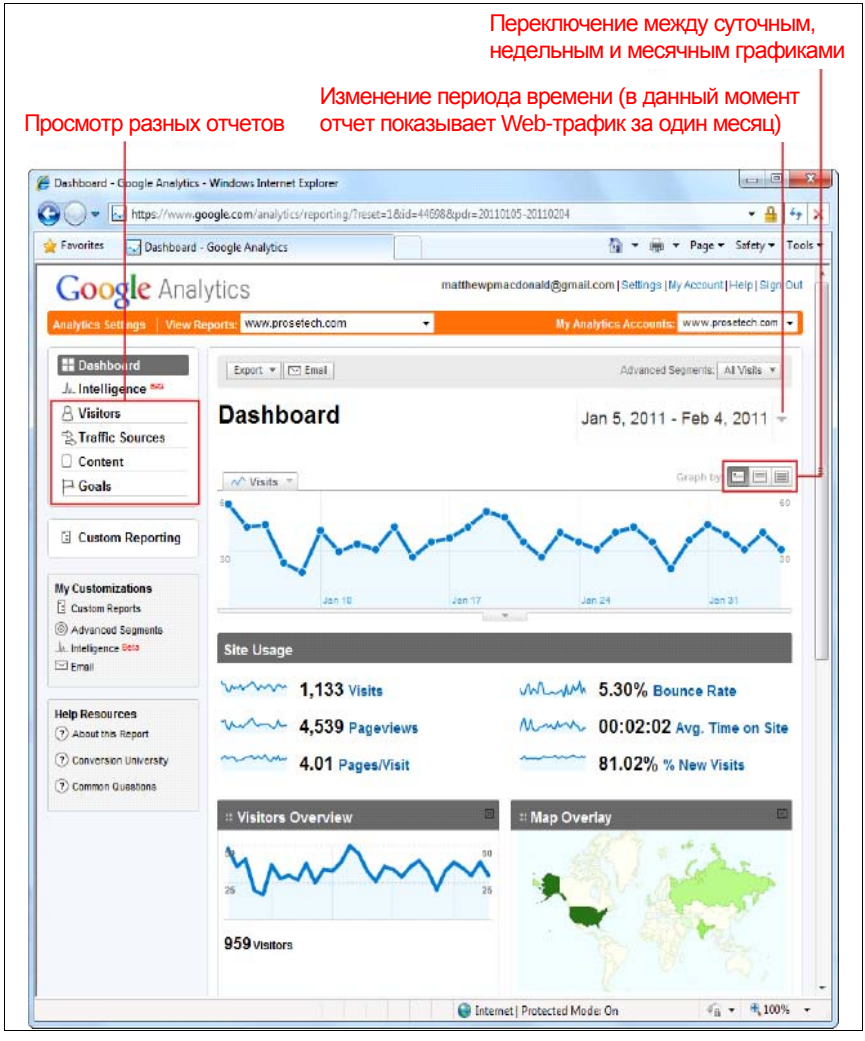


Рис. 11.13. На **Dashboard** выводится ряд базовых графиков, демонстрирующих основные показатели жизнедеятельности вашего Web-сайта. Меню слева позволяет просмотреть набор более специализированных отчетов

- **Tracking Data Not Installed** (Отслеживание не установлено), т. е. Google не собирает никакие данные. Возможно, нужно дождаться посетителей, попавших на ваш сайт, или же вы не вставили корректный код отслеживания на JavaScript.

Если вы видите текст **Receiving Data**, поздравляю! Ваши отчеты готовы и ждут вас. Щелкните кнопкой мыши ссылку **View Report** (Просмотреть отчет), находящуюся рядом с URL вашего Web-сайта, чтобы перейти на панель **Dashboard** (Инструментальная панель) (рис. 11.13).

Объем информации, отображенной на **Dashboard**, может слегка ошеломить. Для того чтобы вы лучше могли понять, что происходит, в следующих разделах анализируются ключевые точки поочередно на каждом графике.

График посещений

В верхней части страницы график посещений (рис. 11.14) показывает ежедневную популярность вашего сайта за последний месяц. Он ничего не говорит о том, сколько страниц просмотрел среднестатистический посетитель или насколько он задержался. Он просто показывает, сколько людей посетили ваш сайт. С помощью этого графика вы можете составить беглое представление об общей тенденции роста или падения вашего Web-трафика и увидеть, растет ли он в определенные дни недели или вблизи конкретных дат.



Рис. 11.14. Для получения значения в определенной точке укажите на нее. Например, на этом графике четко видно скромное число визитов — 51 за 19 января

С помощью нескольких щелчков кнопкой мыши можно получить график просмотров страниц, подсчет количества страниц, просмотренных всеми посетителями. Например, если энергичный покупатель посещает ваш магазин электронной коммерции Banana Painting, отбирает несколько соблазнительных товаров и завершает покупку, Google Analytics может зафиксировать около дюжины просмотров страниц, но только один визит. Количество просмотров страниц всегда не меньше числа визитов; в конце концов каждый визит включает как минимум просмотр одной страницы. Для отображения просмотров страниц щелкните кнопкой мыши стрелку вниз, расположенную рядом со словом **Visits** (Посещения) в левом верхнем углу графика, и затем выберите из списка вариант **Pageviews** (Просмотры страниц).

Совет

Помните о том, что с помощью поля дат в правом верхнем углу **Dashboard** (Инструментальная панель) можно просмотреть данные, собранные Google за разные периоды времени (см. рис. 11.13).

Использование сайта

Раздел **Site Usage** (Использование сайта) заполнен важными статистическими показателями (рис. 11.15). К ним относятся следующие.

- Общее количество визитов за выбранный период времени (первоначально это последний месяц).
- Общее число просмотров страниц.
- Среднее количество страниц, прочитанное посетителем до ухода с вашего сайта.
- Показатель отказов (bounce rate). В случае отказа посетитель отображает только одну страницу — иначе говоря, он попадает на ваш сайт с определенной страницы и затем покидает сайт без каких-либо дополнительных просмотров. Показатель отказов, равный 5%, говорит о том, что 5% посетителей, оказавшись на вашем сайте, тут же покидают его. Отказы от просмотра крайне важны для Web-разработчиков, т. к. они указывают на потерянных потенциальных посетителей. Если вы сможете определить, что вызвало большой процент отказов, вам, возможно, удастся привлечь несколько дополнительных посетителей.
- Среднее время, проведенное посетителем на вашем сайте до просмотра чего-либо другого.
- Процент новых посещений. Например, число 81% означает, что 19% вашего трафика — вернувшиеся посетители, а 81% — новые. Для нормальной жизнедеятельности вашего Web-сайта важны посетители обоих типов.

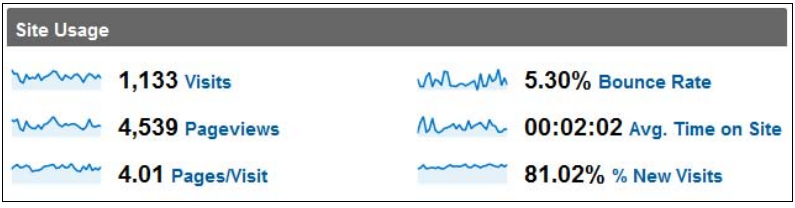


Рис. 11.15. Числа в разделе **Site Usage** — самый главный индикатор Web-состояния вашего сайта. Можно щелкнуть кнопкой мыши любой из этих показателей и увидеть отдельный отчет с множеством подробностей

Примечание

Существует несколько видов вернувшихся посетителей, которых Google не может корректно идентифицировать. Например, если вернувшийся посетитель пользуется другим компьютером, другим браузером или зарегистрировался на своем компьютере с другим именем пользователя, во всех этих случаях он будет учитываться как новый посетитель. По этой причине количество вернувшихся посетителей в отчетах может быть слегка занижено.

Наложение на географическую карту

Раздел **Map Overlay** (Наложение на карту) дает наглядное представление о том, где на земном шаре находятся ваши посетители. Сервис Google Analytics делит карту на разные страны и окрашивает их разными оттенками зеленого цвета — чем темнее цвет страны, тем популярнее в ней ваш Web-сайт.

Map Overlay (Наложение на карту) станет еще интереснее, если использовать уровень **City** (Город), показанный на рис. 11.16. Для его отображения щелкните кнопкой мыши на панели **Dashboard** ссылку **View Report** (Просмотреть отчет), расположенную под строкой **Map Overlay**. На экране появится новая страница с увеличенной картой и подробной информацией под ней. Найдите под картой заголовок **Detail Level** (Уровень детализации) и щелкните кнопкой мыши слово **City**.

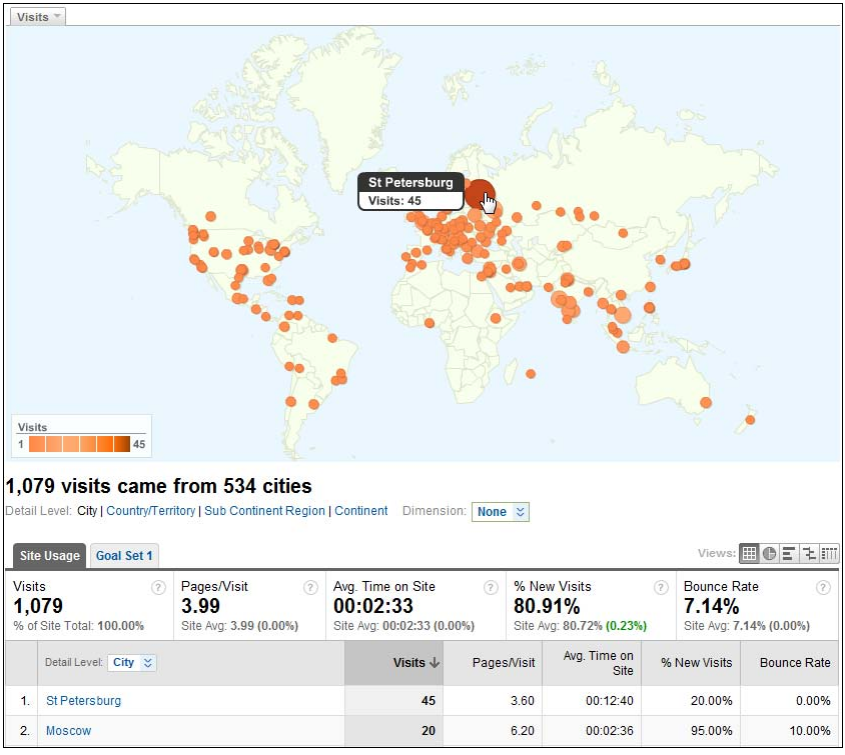


Рис. 11.16. Представление **City** отображает небольшой кружок рядом с местом нахождения каждого посетителя, применяя кружки большего размера для областей, в которых сосредоточен особенно большой объем вашего трафика. Для просмотра показателей по каждому городу указывайте на отдельные точки данных. Здесь показано, что 45 посещений в месяц берут свое начало в браузерах из Санкт-Петербурга в России. В таблице под картой содержатся дополнительные подробности — например, посетители из Санкт-Петербурга просматривают в среднем 3.6 страниц за одно посещение (немного меньше чем обычно), проводят на сайте 13 (в среднем) минут, но насчитывают только 20% новых посетителей

ЧАСТО ЗАДАВАЕМЫЙ ВОПРОС

Неужели Google действительно знает, где я живу?

Насколько точны данные о географическом местоположении, предоставляемые Google?

Умение определять географическое местонахождение посетителей — это мощное средство. В результате, если вы знаете, что ваш сайт Graceland Vacation популярен в Японии, можно подумать о приеме платежей в иенах, переводе нескольких страниц или вставке новых картинок. Но сервис Google, определяющий географическое местоположение, не совершенен. У него есть два недостатка.

- Сервис определения местоположения основан на сведениях о провайдере интернет-услуг (Internet Service Provider, ISP) посетителя, а не на данных о самом посетителе. Во многих случаях провайдер находится не там, где собственный компьютер посетителя.
- Провайдеры экономят, объединяя свой трафик и выводя его в Интернет в одном месте. Это означает, что даже если ваш посетитель и его Web-сервер находятся в определенном городе, компьютер, подключающий этих посетителей к Интернету, может находиться где-нибудь еще.

На практике географическая информация, используемая Google, с большой вероятностью близка к реальной, но не точна. Скорее всего страна определена верно, а конкретный город может и не быть местом жительства вашего посетителя.

Обзор источников трафика

Диаграмма **Traffic Sources Overview** (Обзор источников трафика) показывает, как посетители попадают на ваш сайт. У нее есть три сектора.

- **Referring Sites** (Исходные сайты). Этот сектор включает посетителей, пришедших с Web-сайтов, ссылающихся на ваш.
- **Direct Traffic** (Прямой трафик). В этом секторе учтены посетители, которые ввели ваш URL непосредственно в строку адреса в браузере или использовали закладку для повторного посещения.
- **Search Engines** (Поисковые системы). Этот сектор включает посетителей, пришедших на ваш сайт из поиска в Google или другой поисковой машине.

На рис. 11.17 показано подробное представление диаграммы.

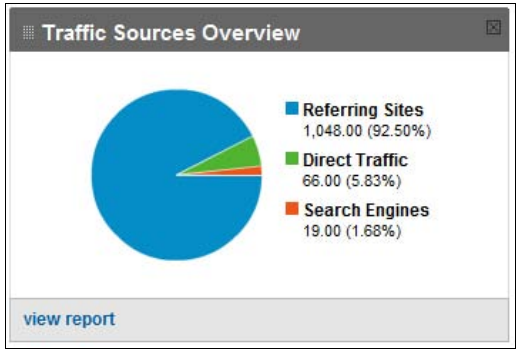


Рис. 11.17. За прошлый месяц 92% посетителей вашего сайта пришли на него с исходных сайтов, 6% — непосредственно на сайт (набрав его URL). Менее 2% нашли дорогу с помощью поисковой машины, что указывает на необходимость дополнительных затрат времени на поисковую оптимизацию сайта (см. *разд. "Регистрация с помощью поисковых машин" данной главы*)

Диаграмма **Traffic Sources Overview** (Обзор источников трафика) дает четкое общее представление о том, как посетители попадают на ваш сайт. Для того чтобы увидеть дополнительную информацию, щелкните кнопкой мыши ссылку **View Report** (Просмотреть отчет), расположенную ниже, или используйте меню **Traffic Sources** (Источники трафика), находящееся слева. Вы можете изучить таблицы, в которых анализируются секторы с исходными сайтами и поисковыми системами.

В таблице с исходными сайтами вы увидите, какие именно Web-сайты направляют к вам трафик, что служит очень удобным инструментом для быстрого определения своих самых удачных Web-связей. В таблице с анализом сектора поисковых систем вы сможете просмотреть ключевые слова, приведшие на ваш сайт и позволяющие понять, что именно ищут ваши посетители (вы, возможно, установите, почему они ушли неудовлетворенными).

Обзор контента

Все Web-разработчики знают, что страницы получаются разные. Одни могут вызывать громадный интерес, а другие, слабые, — игнорироваться. В разделе **Content Overview** (Обзор) перечислены самые эффективные страницы (рис. 11.18).

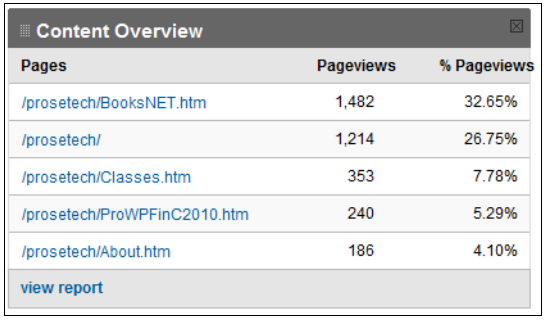


Рис. 11.18. На этом сайте страница BooksNET.htm — вне всякого сомнения обладатель награды "Самая популярная страница", обеспечившая 32% общего трафика Web-сайта

Однако простого выявления самых часто просматриваемых вашими посетителями страниц недостаточно. Некоторые страницы становятся излишне важными благодаря своей способности *заманить* посетителей. Например, страница Member Photos (Фотографии участников) на вашем международном нудистском сайте International Nudist может привлечь огромные массы посетителей, которые затем останутся на близлежащих страницах сайта и будут разглядывать сделанные на заказ специально для вас кофейные чашки, одежду и памятные вещи. Хорошее средство Web-аналитики покажет, где именно посетители *заходят* на ваш сайт, чтобы вы знали, какие страницы привлекают внимание и трафик. Почти так же важны страницы, на которых посещение заканчивается. Они могут указывать на проблему, например, медленную загрузку страницы, некорректное ее отображение в некоторых браузерах или раздражающее людей содержимое. Для получения такого сорта подробностей в Google Analytics щелкните мышью ссылку **View Report** (Просмотреть отчет) в нижней части раздела **Content Overview** (Обзор) или используйте меню **Content** (Содержание), расположенное на странице слева.

Примечание

В Google Analytics есть еще огромное множество отчетов, которые вы можете изучить, и внушительное количество способов компоновать их результаты, чтобы сделать выводы о вашем Web-трафике. На самом деле о тонком искусстве анализа эффективности Web-сайтов написаны целые книги. Но пять отчетов, описанных в книге, помогут вам начать, уже имея достаточное представление о том, как функционирует ваш сайт.

12

ГЛАВА 12

Продвижение Web-сайта

Самый лучший в мире Web-сайт не принесет никакой пользы, если будет оставаться в гордом одиночестве. Для процветания вашего сайта необходимо привлекать новых посетителей и заставить их возвращаться еще и еще.

В предыдущей главе вы вступили на этот путь: узнали, как помочь поисковым машинам заметить ваш сайт и как отслеживать количество посетителей, заглянувших к вам. Но списка с результатами поиска недостаточно для превращения нового сайта в процветающий. Для этого потребуется ряд стимулирующих тактических действий, начиная с публикации ссылок и заканчивая покупкой мест для размещения рекламы. Вопреки вашим ожиданиям такого рода примитивное продвижение может принести вашему сайту больше трафика, чем действия сверхмощных поисковых машин, например Google.

В этой главе вы узнаете о нескольких самых лучших способах продвижения Web-сайтов, увидите, как создавать прилипчивый сайт — тот, который не только привлекает новые лица, но и заставляет посетителей вернуться. Для того чтобы преуспеть в этом, нужно превратить ваш Web-сайт в Web-*сообщество*. Вашим посетителям должен потребоваться способ взаимодействия с вашим контентом и общение с вами и друг с другом, для этого вы можете использовать Facebook, Twitter, блоги и группы.

Примечание

Задачи, обсуждаемые в большей части книги, — создание Web-страниц, форматирование их с помощью таблиц стилей, выгрузка вашего завершенного сайта — относительно понятны и просты. Они могут потребовать какого-то времени и усилий, но когда они выполнены, вы *знаете*, что они выполнены. Задачи, с которыми вы столкнетесь в этой главе, — продвижение вашего Web-сайта и формирование на его основе сообщества — не так четко определяются. Они требуют непрерывной работы и возможно огромного неблагодарного, изнурительного труда, прежде чем трафик вашего сайта начнет расти.

Рассказать всем

Для успешного продвижения вашего Web-сайта вам придется овладеть многими навыками. Умелое продвижение Web-сайтов включает в себя как традиционную рекламу, так и магию поисковых систем.

В следующих разделах вы научитесь нескольким фундаментальным методикам. Вы начнете задумываться о том, как разрекламировать себя в нужных местах, познакомитесь с сервисами Google Places и Google AdWords и разработаете стратегию, которая будет способствовать повторным посещениям.

Беззастенчивая самореклама

Для того чтобы ваш Web-сайт был упомянут на многих самых популярных Web-сайтах, придется выложить живые деньги. Но самая лучшая реклама ничего не стоит: поищите сайты, на которых вы сможете одновременно рекламировать себя и вносить вклад.

Например, если вы создали Web-сайт **www.HotComputerTricks.com**, почему бы не ответить на несколько вопросов, заданных в группе новостей по вычислительным наукам, форуме или на странице Facebook? Явное рекламирование своего сайта считается бестактным, но нет ничего плохого в том, чтобы отправить полезный совет и завершить его подписью, в которую входит ваш URL.

Далее приводится пример ответа на вопрос участника форума и заодно появляется возможность замолвить доброе слово о себе:

> Джим,
>
> Проблема в том, что большинство жестких дисков ломаются при погружении в воду. Следовательно, твоя идея о создании компьютера-рыболова не сработает.
>
> Саша Медник
>
> **www.HotComputerTricks.com**

Публикация ответа гораздо лучше отправки по электронной почте непосредственно участнику форума, потому что на популярном сайте ваше сообщение прочтут сотни компьютерных фанатов, у которых возник тот же вопрос. Даже если лишь некоторые решат проверить ваш сайт, это уже будет значительным прогрессом.

Если вы очень старательны, то можете вставить и нечто более недвусмысленное.

> Джим,
>
> Проблема в том, что большинство жестких дисков ломаются при погружении в воду. Следовательно, твоя идея о создании компьютера-рыболова не сработает. Но может быть ты хочешь проверить мою вакуумную камеру для жесткого диска (**www.HotComputerTricks.com**), которую я разработал дома для решения именно этой проблемы.
>
> Саша Медник
>
> **www.HotComputerTricks.com**

ПРЕДУПРЕЖДЕНИЕ

Этот прием требует очень легкого прикосновения. Как показывает практика, ваше сообщение должно быть целенаправленно. Посылайте посетителей на ваш сайт, только если там есть что-то конкретное, отвечающее на вопрос.

На некоторых сайтах можно публиковать советы, обзоры или статьи. В этом случае вы можете применять варианты описанного ранее метода. Напоминаю, выдайте полезный совет и затем поставьте подпись в конце сообщения. Например, если вы предлагаете бесплатную статью, описывающую, как создать вашу потрясающую вакуумную камеру, завершите ее следующей строкой.

> Саша Медник — компьютерный гений, запустивший перворазрядный сайт о компьютерах **www.HotComputerTricks.com**.

Продвижение всегда идет легче, если вы верите в свое творение. Поэтому прежде чем хвастаться, убедитесь в том, что у вас на сайте соответствующий высококачественный контент. Никого не посылайте на свой сайт, если вы только планируете вставить на него контент (когда-нибудь).

> ***Совет***
>
> Если вы — бизнесмен, пытающийся рекламировать свой товар, вы добьетесь большего, если подыщите людей, которые помогут вам рассказать всем о вашем товаре. Отличная мысль найти влиятельных блогеров, людей создавших Web-сайты с форматом интернет-дневника или личного журнала, о котором вы узнаете в *главе 13*. Например, если вы пытаетесь продавать пушистые махровые пижамы для малышей, подыщите популярных людей, ведущих блоги, посвященные родительским обязанностям. Предложите им несколько бесплатных пижам, если они поделятся своими мыслями о них в блоговом обзоре. Такого сорта продвижение из уст в уста может быть гораздо успешнее в обширных Web-сообществах, чем в обычном реальном мире вне сети.

Google Places

В предыдущей главе вы узнали, как ваш сайт может оказаться результатом чьего-то Web-поиска. Это незаменимый способ привлечения новых людей, и многие Web-разработчики проводят бессонные ночи, мысленно возвращаясь вновь и вновь к ранжированию страниц и ключевым словам.

Но если вы управляете коммерческим предприятием или организацией, существующими в реальном мире (а не только в виртуальном), есть очень простой инструмент, который поможет вашему сайту выделиться из толпы. Скорее всего вы видели его раньше, если когда-либо использовали Google для поиска местных предприятий (рис. 12.1).

Ясно, что такая страница места весомее, чем стандартное упоминание в результатах поиска. Она занимает больше места, предоставляет дополнительную информацию, и предприятия на ней выглядят более профессионально. Когда посетители видят подобную страницу, им кажется, что они нашли реальное сложившееся коммерческое предприятие.

Сервис, создающий ее, называется Google Places (Google Адреса). К счастью, вы можете зарегистрироваться в Google Places и иметь бесплатно собственную усиленную таким образом запись для результатов поиска. Нужно только помнить следующее.

- *Вы должны предоставить свой местный почтовый адрес.* Для того чтобы убедиться в существовании вашей фирмы по указанному вами адресу, Google по-

шлет по почте код подтверждения. После того как вы введете этот код, ваша запись Google Places станет активной. Если у вас нет реального почтового адреса (например, как домашний), или вы не хотите предоставлять его Google, сервис Google Places не для вас.

- *Расширенная запись появляется только в определенных поисковых списках.* Задача Google — показать улучшенную запись, когда Web-искатель ищет именно вашу фирму. Если кто-то вводит название фирмы или адрес в строку поиска Google, он вполне вероятно получит расширенную страницу. Но если некто просто вводит несколько ключевых слов, соответствующих контенту вашего Web-сайта, Google отображает в результатах поиска обычную строку.

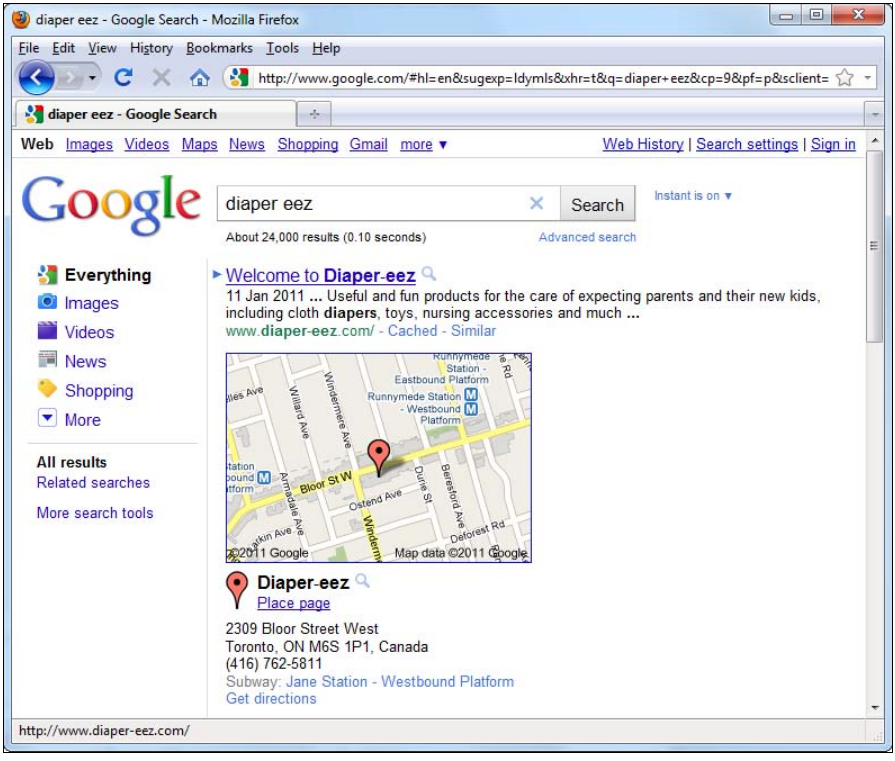


Рис. 12.1. Первый результат в поисковом списке — расширенная запись, включающая карту, адрес и ссылку для определения местонахождения. Щелкните мышью ссылку **Place page** (Страница места), и вы получите полную с картинками страницу места для предприятия

Для создания собственной расширенной записи в Google Places перейдите на страницу **http://places.google.com/business**. Если у вас есть учетная запись (аккаунт) Google (используемый в других сервисах, таких как Gmail и Google Webmaster Tools), входите прямо сейчас. В противном случае для ее создания щелкните мышью кнопку **Sign up now** (Зарегистрироваться сейчас).

После регистрации вы можете добавить новую запись с данными о вашей компании. Вам необходимо заполнить одностраничную форму с информацией о вашем предприятии, включающую ваш адрес, номер телефона, адрес Web-сайта, короткое

описание и категорию. При желании вы можете вставить картинки (что всегда неплохо, т. к. посетители увидят их на вашей странице в Google Places), видео с YouTube и указать часы работы. Когда закончите, передайте форму на рассмотрение в Google. Помните, что ваша запись не появится автоматически, т. к. компания Google должна отправить вам по почте код подтверждения. Для того чтобы сделать активной вашу страницу места, введите полученный от Google код. И после этого, через какое-то время, ваша расширенная запись будет отображаться в списках результатов Web-поиска.

Google AdWords

Как человек, увлеченный Web, вы, конечно же, видели более чем достаточно мигающих сообщений, безвкусных баннеров и навязчивых всплывающих окон, и все они стараются продать вам совершенно отвратительные товары. Нетрудно догадаться, что реклама такого сорта не поможет продвижению вашего сайта — она скорее отвращает людей, чем привлекает их. Но есть вполне приличные платные места, способные представить ваш сайт нужным читателям в подходящее время и с достаточным тактом. Одно из лучших — AdWords (**http://adwords.google.com**), невероятно гибкая система размещения рекламы, принадлежащая Google.

Принцип действия AdWords заключается в отображении созданных вами рекламных текстов среди обычных элементов списка результатов поиска (рис. 12.2). Причем Google отображает их небеспорядочно. Вы сами выбираете ключевые слова, с которыми хотите связать ваши рекламные объявления.

Самая щекотливая (и слегка смущающая) особенность AdWords — оплата вами ключевых слов, которые хотите использовать. Например, вы можете сообщить Google, что собираетесь заплатить 25 центов за ключевое слово "пища". Google принимает во внимание ваше предложение наравне с любыми другими и отображает чаще рекламные объявления с более высокой оплатой, но не старается на ком-то нажиться и требует с вас только текущую ставку за ваше ключевое слово независимо от заявленной вами цены за него. Кроме того, Google не берет ничего за обычное отображение вашего рекламного объявления на странице результатов. Он потребует с вас плату, только если кто-то щелкнет кнопкой мыши ваше рекламное объявление, чтобы попасть на ваш сайт.

К этому моменту вы, возможно, слегка занервничали. Учитывая, что Google обрабатывает сотни миллионов поисков в день, даже мизерная цена в один цент быстро приведет к банкротству вас и ваш сайт? Но у Google и для этого есть решение. Вы просто сообщаете Google, сколько готовы платить в день. Как только ваш лимит исчерпан, Google перестает отображать ваши рекламные объявления.

Интересно, что частота вывода ваших объявлений определяется не только величиной заявленной вами цены. Очень важна частота обращений. Если Google показывает ваше объявление снова и снова и по нему никто никогда не щелкает мышью, Google понимает, что оно просто не работает и дает вам знать об этом с помощью автоматически создаваемого сообщения электронной почты. И до тех пор, пока вы не улучшите вашу рекламу, Google будет показывать ее реже и реже или вообще перестанет отображать.

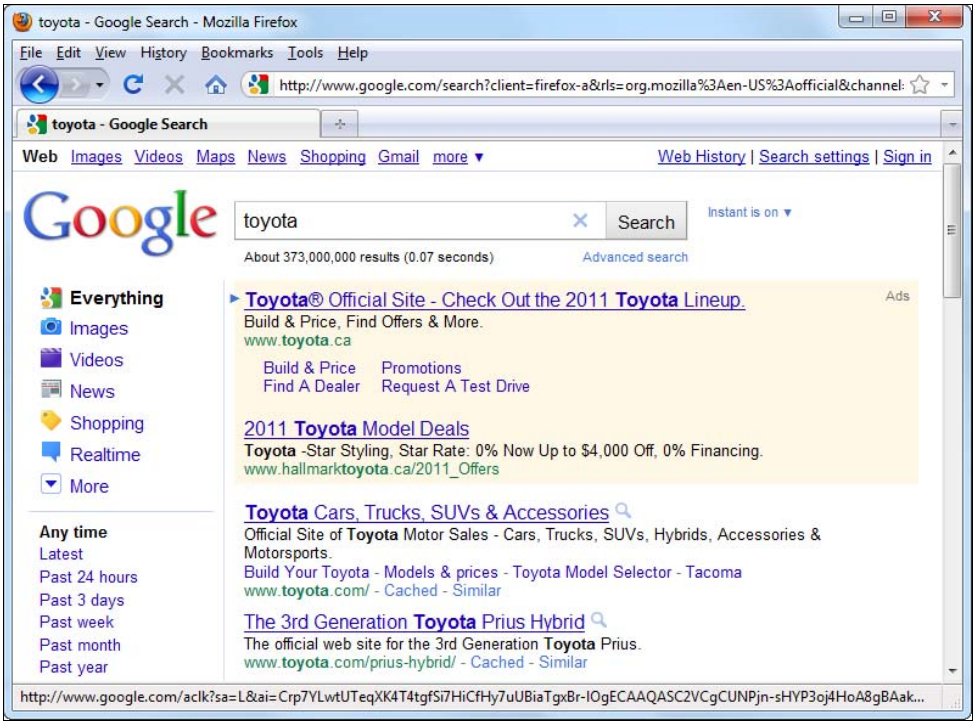


Рис. 12.2. Для того чтобы увидеть AdWords в действии, попробуйте найти название такой известной марки, как Toyota. Справа от результатов поиска или прямо над ними в области, окрашенной светло-желтым цветом, вы увидите раздел с четкой пометкой **Ads** (Объявления)

Вы также можете ограничить круг людей, указав определенные географические регионы (например, конкретную страну или город). В этом случае вы не будете попусту тратить деньги, показывая рекламу вашего салона-парикмахерской для малышей людям из Туниса. Такого рода географическая избирательность делает сервис AdWords одинаково эффективным как для местного, так и для глобального продвижения.

В AdWords могут быть конкуренты. Для того чтобы иметь шансы в борьбе со всеми акулами AdWords, вам нужно знать, сколько для вашего сайта стоит щелчок кнопкой мыши. Например, если вы продаете носки с монограммой, следует выяснить, какой процент посетителей действительно *покупает* что-нибудь (показатель эффективности рекламы или коэффициент конверсии), и сколько они могут потратить. Типичные затраты-на-щелчок колеблются около 75 центов, но существует большой разброс. По последним данным слово *юрист* стоило 8.53 доллара, комбинация ключевых слов *уход за ламой* была в наличии за бесценок, жалкие 5 центов. (В недавнем прошлом ставка за слово "мезотелиома" — разновидность рака, связанная с асбестом, которая могла бы стать основой для коллективного иска, — доходила почти до 100 долларов.) Прежде чем регистрироваться в AdWords, неплохо провести серьезное исследование, чтобы выяснить свежие цены на ключевые слова, которые хотите использовать.

ПРИМЕЧАНИЕ

Дополнительную информацию о сервисе AdWords см. на его сайте (**http://adwords.google.com**). Для разнообразия загляните на страницу **www.iterature.com/adwords**, прочтите историю о попытке одного художника использовать AdWords для распространения стихов и почему она потерпела неудачу.

Вернувшиеся посетители

Привлечение свежих посетителей — очень важный компонент продвижения Web-сайта, но новоиспеченные Web-мастера часто забывают нечто столь же важное — вернувшихся посетителей. Для того чтобы Web-сайт стал по-настоящему популярным, ему нужно привлекать посетителей, которые будут возвращаться снова и снова. Web-разработчику гораздо полезнее тратить меньше времени на попытки заманить новых посетителей и больше времени на удержание уже имеющихся.

Если вы занимаетесь маркетингом, то знаете, что клиент, пришедший в один и тот же магазин в третий или четвертый раз, гораздо скорее сделает покупку, чем человек, оказавшийся там впервые. Постоянные покупатели с большей вероятностью могут войти в азарт и уговорить своих друзей прийти и посмотреть. Подобный заражающий энтузиазм может привести к виртуальному порогу вашего Web-сайта все больше и больше посетителей. Явление это настолько распространено, что получило специальное название: вирус посещаемости (traffic virus).

ПРИМЕЧАНИЕ

Вернувшиеся посетители — это идеальное мерило успешности Web-сайта. Если вы не заинтересовали кого-нибудь настолько, чтобы он вернулся к вам снова, ваш Web-сайт просто не соответствует своему назначению.

Как же вашему Web-сайту стать любимым конечным пунктом для Web-путешественников? Все уже сказано в старой интернет-поговорке: контент — всему голова. Ваш сайт должен быть до отказа набит интереснейшей информацией, требующей непременного прочтения. Также важно, чтобы эта информация обновлялась регулярно и ощутимо. Если вы обновляете ее один раз в месяц, вряд ли у вашего сайта будет пульс. Но если данные меняются два раза в неделю и даже чаще, ваш сайт будет жить.

Никогда не умаляйте важность регулярных обновлений. Для приобретения вернувшегося посетителя нужны недели и месяцы обновления информации. А один пропуск — скажем, три месяца без каких-либо изменений за исключением цвета ваших кнопок — не просто перестанет привлекать новичков, но и ликвидирует текущий реестр вернувшихся посетителей. Сообразительные посетители сразу поймут, что Web-сайт пришел в упадок. Их ощущения похожи на ваши в тот момент, когда вы открываете холодильник и достаете когда-то ароматную выпечку и обнаруживаете, что она тверда, как вулканическая порода. Вы знаете, что за этим последует: выпечку выбросить, все закладки Web-сайта стереть и двигаться дальше.

Примечание

Признаками заброшенного сайта служат старомодное форматирование, потерянные ссылки и ссылки на устаревшие новости (например, вечеринка по случаю выпуска компакт-диска Spice Girls или технический анализ причин нерушимости капиталовложений в кооперативные квартиры во Флориде).

Другой способ — вовлечь вернувшихся посетителей в создание сообщества. Связующей нитью могут стать дискуссионные форумы, рекламные акции и информационные рассылки. Они помогают посетителям ощущать себя участниками вашего сайта, совместно использующими ваше Web-пространство. Если вы правильно поймете это, множество посетителей придет к вам и никогда не захочет уходить.

СЕКРЕТЫ РЕМЕСЛА

Любимые пиктограммы

Одна из первостепенных задач при продвижении сайта — заставить пользователей добавить ваш сайт в закладки их браузеров. Но создание закладки еще не гарантирует повторного посещения. Ваш Web-сайт должен быть достаточно интересным, чтобы привлекать к себе внимание в списке закладок и заставлять посетителей возвращаться. Если вы обычный Web-путешественник, вы регулярно посещаете около 5% сайтов из списка закладок.

Один из способов гарантированного выделения сайта из множества других — замена пиктограммы, которая появляется в закладках посетителя или меню **Избранное** (такую пиктограмму называют *иконкой сайта* (favicon)). Этот прием действует в любом современном браузере (рис. 12.3).

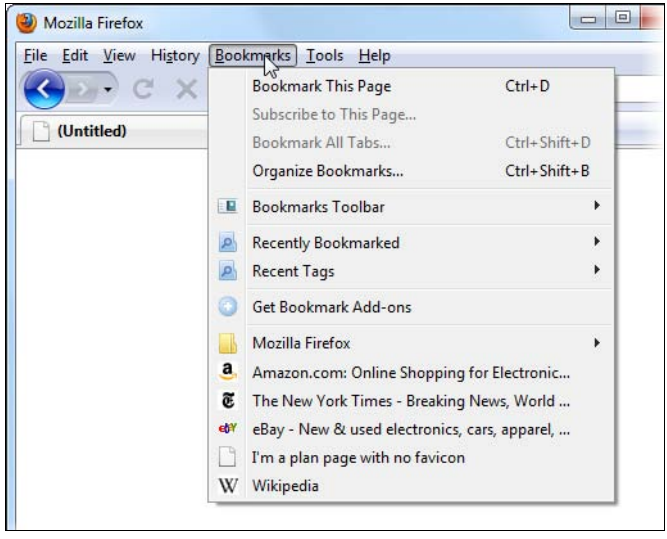


Рис. 12.3. В меню **Bookmarks** (Закладки) показаны пользовательские иконки сайтов книжного магазина Amazon, газеты The New York Times, центра интернет-торговли eBay и интернет-энциклопедии Wikipedia

Для создания иконки сайта вставьте файл иконки в папку верхнего уровня для вашего Web-сайта, убедитесь, что его имя — favicon.ico. Лучше всего воспользоваться специализированным редактором пиктограмм, поскольку он позволит создать в одном файле иконку размером 16×16 пикселов и иконку размером 32×32 пиксела. Иконку

меньшего размера браузеры применяют в своем меню закладок, а когда посетители с помощью мыши перетаскивают иконку сайта на свой рабочий стол, компьютеры выводят версию большего размера. Если у вас нет редактора пиктограмм, просто создайте bitmap-файл (с расширением bmp) шириной и высотой точно 16 пикселов. Найти редактор пиктограмм можно на сайте с условно-бесплатным программным обеспечением, например, **www.download.com**.

Превращение сайта в сообщество

Web — это густо населенный дом для нескольких миллионов человек, поэтому, когда вы публикуете свой сайт, он попадает не в вакуум. Он оказывается в центре внимания аудитории, всегда заинтересованной и часто излишне категоричной.

Для того чтобы действительно понравиться остальному Web-миру, необходимо взаимодействовать с вашими посетителями. Идея диалога — прямых и обратных контактов с узлами одного уровня — аппаратно реализована в сетевой архитектуре Интернета. Обеспечьте такое взаимодействие, и люди будут не просто говорить о вас, они разрекламируют вас своим друзьям, познакомятся с вами поближе и привлекут к вам общественное внимание.

Но для воплощения этой магии в жизнь необходимо некоторое планирование, желание разрекламировать себя (почти до стойкого смущения) и общее представление о принципах работы социальных сетей.

Взращивание Web-сообщества

Как начать превращать ваш Web-сайт в Web-сообщество? Первая хитрость — изменить ваш взгляд на сайт и планировать его как место встречи, а не просто место, дающее выход вашим (бесспорно гениальным) мыслям. Далее приведено несколько советов, которые помогут вам настроиться на нужный образ мыслей.

- *Четко определите назначение вашего сайта.* Например, описание "**www.BronteRevival.com** предназначен для объединения поклонников Шарлотты Бронте, обсуждения и рекламирования ее творчества" более ориентировано на сообщество, чем "**www.BronteRevival.com** содержит сведения и критические статьи о творчестве Шарлотты Бронте". Первое предложение описывает, что сайт намерен делать, а второе — что он содержит, ограничивая тем самым его поле деятельности. После того как короткое описание сформулировано, его можно применить в метаэлементе описания (см. *разд. "Вставляйте описание страницы" главы 11*) или в программном заявлении на вашей исходной странице.
- *Создайте места сбора.* Никому не захочется слоняться вокруг коллекции ссылок и неподвижного текста. Придайте живость вашему сайту с помощью дискуссионных форумов или досок интерактивной переписки (chatboard), где ваши посетители смогут вытянуть усталые ноги. Позже в этой главе вы узнаете, как добавить эти компоненты.
- *Предложите вашим пользователям разные роли.* Успешные сайты-сообщества распознают участие разного уровня. На одном краю подходящие пользователи

могут дорасти до роли лидера и даже координировать события, информационные рассылки и дискуссионные группы или части вашего сайта. На другом краю посетители, предпочитающие притаиться на заднем плане, наблюдая за тем, что делают другие. Существуют разные способы распознавания индивидуального уровня участия — некоторые сайты применяют систему ранжирования персональной обратной связи, которая вставляет золотые звезды (или другие пиктограммы) рядом с именем посетителя. Другой метод — предоставление отдельным посетителям дополнительных прав, например, возможность управлять членами группы Google (см. *разд. "Группы" данной главы*).

- *Объявляйте о новом контенте до и после его вставки.* Для того чтобы заставить посетителей возвращаться снова и снова, вам понадобится огромный объем нового контента. Но самого по себе нового содержимого недостаточно — нужно подогревать интерес посетителей к новому контенту, чтобы у них возникло желание вернуться, и следует четко выделять новый материал, чтобы гости могли его найти, когда вернутся. Для того чтобы все работало четко, попробуйте на первую страницу вставить ссылки на новый контент вместе с короткой строкой или двумя о ближайшем планируемом контенте и точные сведения о том, когда он появится.
- *Оповещайте о регулярных событиях.* Очень трудно заставить себя регулярно обновлять свой сайт. Но даже если это делается, посетителям неоткуда узнать, когда появляется что-либо заслуживающее их повторного посещения. Почему бы с помощью рекламирования регулярных событий (например, раздела новостей, обновляемого еженедельно, или поощрительной лотереи, проводимой в назначенный день) не помочь всем следить за тем, что происходит.
- *Создавайте циклы обратной связи.* Закон Web — хорошие сайты становятся лучше, а плохие — множат свои ошибки. Чтобы помочь вашему сайту выйти на верную дорогу, убедитесь, что у ваших посетителей есть способ рассказать вам о своих предпочтениях. После этого тратьте уйму времени на усиление того, что работает, и отбрасывание того, что не работает.

Теперь, когда вы одолели принципы солидаризации для вашего Web-сайта, самое время узнать, как создать ингредиенты, необходимые любому Web-сообществу.

Инструменты Web-сообществ

Сам по себе обычный Web-сайт — это одноканальная среда, как кабельное телевидение или газеты. Web-посетитель попадает на ваш сайт, читает несколько страниц и покидает его. В этот момент общение заканчивается.

По мере того как Web-сайты становились более ориентированными на сообщества, они углубляли и расширяли это общение. Главная цель — напомнить случайным посетителям о своем существовании. Если вы представляете крупный бизнес, такой подход — часть стратегии продвижения торговой марки, которая убеждает потенциальных покупателей в ценности ваших товаров. Если вы — представитель мало-

го бизнеса, она может стать частью длительного процесса обхаживания, который ведет к продаже. Но если вы даже ничего не продаете и не рекламируете, сохранение посетителей, поддерживающих связь с вашим Web-сайтом, — лучший способ увеличить трафик, приобрести новые контакты и остаться в центре внимания.

Web-сайты отличаются степенью ориентированности на Web-сообщество. Одни сайты включают только легкое соприкосновение с сообществом (например, система отзывов о товарах на сайте электронной коммерции), другие неразрывно связаны с Fan-страницами Facebook и дискуссионными форумами. В табл. 12.1 перечислено несколько способов построения сообщества на Web-сайте. Остальная часть данной главы поясняет, как практически применять их на вашем собственном сайте.

Таблица 12.1. *Разные способы охвата ваших посетителей*

Способ	**Описание**	**Степень объединения**	**Дополнительная информация**
Информационные рассылки по электронной почте	Читатели подписываются, и вы в любое время посылаете им последние новости. Это хороший способ поддерживать информированность сообщества, не заставляя людей повторно посещать ваш сайт	Низкая. Рассылки — это все еще монологи, у читателей нет возможности ответить	*Разд. "Электронные информационные рассылки" данной главы*
Блоги	Вы пишете регулярные неформальные заметки. Читатели могут постоянно следить за ними с помощью клиентского ПО чтения рассылок (feed reader) и реагировать, вставляя комментарии	Средняя. Блоги кажутся более удобными для диалога, и популярные блоги насчитывают массу комментариев	*Глава 13*
Twitter	Вы посылаете миру короткие фрагменты текста с новостями, комментариями и случайными мыслями. Другие пользователи могут подхватить вашу тему и разослать сообщение своим подписчикам	Высокая. На первый взгляд сообщения в Twitter выглядят как монолог. Но в действительности Twitter — часть обширной перекрывающейся беседы, которая никогда не кончается	*Разд. "Twitter" данной главы*
Группы	Читатели присоединяются к форуму, где они могут оставлять сообщения и беседовать друг с другом. Единственный недостаток — обсуждение ограничено совокупностью зарегистрированных участников	Высокая. Не только вы можете отвечать своим посетителям, но и они могут общаться друг с другом	*Разд. "Группы" данной главы*

Таблица 12.1 *(окончание)*

Способ	Описание	Степень объединения	Дополнительная информация
Facebook	Вы создаете свою страницу, на которую можете выкладывать новости, начинать обсуждения и предоставить возможность другим людям болтать о вас. В некотором смысле ваша страница — гибрид информационной рассылки и группы, которые Facebook соединяет вместе	Высокая. Похожа на группы, но выкладывать информацию может любой из сотен миллионов зарегистрированных пользователей Facebook	*Разд. "Facebook" данной главы*

Электронные информационные рассылки

Идея рассылок по электронной почте проста. Когда посетители задерживаются на вашем Web-сайте, вы предлагаете им возможность подписаться на информационную рассылку вашего Web-сайта. Если немного повезет, посетители введут свои адреса электронной почты, щелкнут мышью кнопку и предоставят вам открытый доступ к своим почтовым ящикам.

После этого вы периодически посылаете посетителям электронные письма с новостями и специальными предложениями. Обычно это делается не чаще одного раза в месяц. В конце концов вы ведь не хотите, чтобы рассерженный адресат заблокировал ваш адрес электронной почты и отправлял ваши сообщения прямо в корзину для информационного мусора.

Даже в современном Web-мире с блогами, Facebook и социальными сетями рассылки по электронной почте все еще широко распространены. Но они создают несколько новых проблем.

Во-первых, вам нужен небольшой ящик где-нибудь на вашем Web-сайте, куда посетители смогут положить свои адреса электронной почты и подписку на информационные рассылки (рис. 12.4). Создать такой код на HTML несложно. Строго говоря, он называется *контактной формой* и использует набор элементов, называемых HTML-формами (см. *разд. "Интерактивные формы" главы 15*). Самое интересное происходит, когда кто-нибудь щелкает мышью кнопку регистрации.

Если вы — Web-программист, работающий с многофункциональной платформой, контактные формы вам сделать легко. Когда посетитель щелкает кнопку мышью, браузер посылает его зарегистрированный адрес электронной почты Web-приложению, выполняющемуся на Web-сервере, и эта программа сохраняет адрес в базе данных. Но простым смертным нужен другой подход. По существу у вас два варианта: позаимствовать чей-нибудь серверный сценарий (крошечный кусочек кода, выполняющийся на Web-сервере) или воспользоваться сервисом предоставления форм.

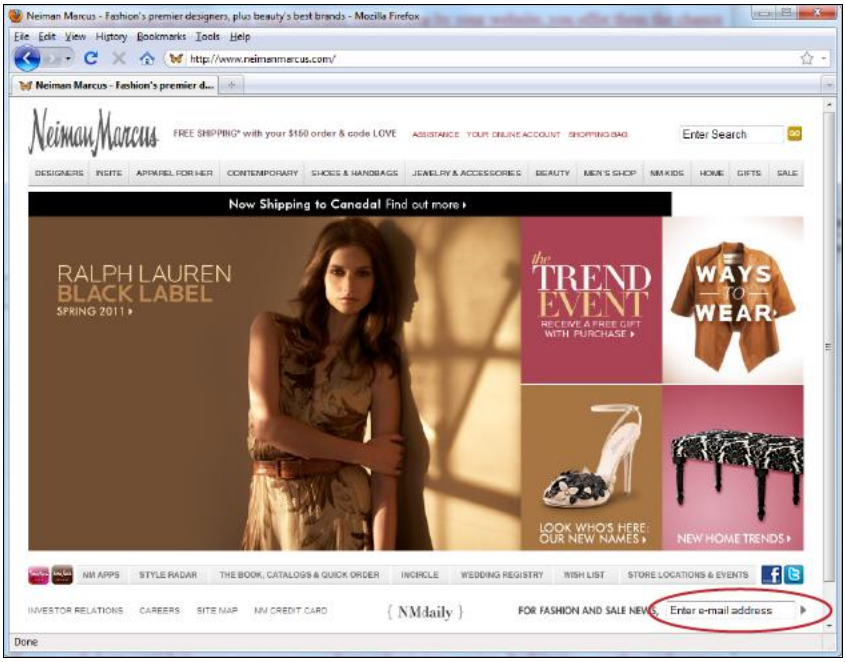


Рис. 12.4. Продавцы модной одежды обычно помещают ссылку на регистрацию, позволяющую подписаться на информационные рассылки, на главные страницы своих сайтов. На этом рисунке пиктограмма обычной стрелки играет роль кнопки **Sign Up** (Зарегистрироваться) или **Subscribe** (Подписаться) и способствует спонтанным регистрациям

В случае сценария трудность в том, что вам нужен сценарий, работающий на серверах вашей компании Web-хостинга. Код сценария зависит от серверных программных средств, поддерживаемых ими. Решить эту проблему можно тремя способами.

- *Запросите ваш Web-хост.* Проверьте, нет ли на вашем Web-хосте каких-нибудь готовых стандартных сценариев, которые вы сможете применить для создания базовой контактной формы. Как правило, вы воспользуетесь небольшим фрагментом заранее написанного кода на языке PHP или ASP, извлекающим адрес электронной почты посетителя и отправляющим его вам по электронной почте. Вам просто нужно вставить этот код на свою Web-страницу и настроить некоторые мелкие детали.
- *Поищите в Web.* Если вы не сможете найти готовый сценарий, снова обратитесь к вашему Web-хосту — на этот раз выясните, поддерживают ли они язык серверных сценариев и какой. Затем вы можете обратиться к Google, чтобы найти подходящий сценарий. Например, на странице **www.freecontactform.com/free.php** есть заготовки HTML и сценарий, который можно использовать для сбора адресов электронной почты на Web-сервере, поддерживающем PHP.
- *Сервисы создания форм.* Наконец, если ваш Web-хост не поддерживает серверные сценарии, или вы просто не хотите решать задачи, с которыми обычно имеют дело программисты, можно использовать бесплатный *сервис создания форм*. По сути, этот сервис обрабатывает вашу форму на своих Web-серверах, а дан-

ные посылает вам по электронной почте. Для создания формы вы просто выбираете данные, которые хотите собирать. Компания, занимающаяся созданием и обработкой форм, предоставляет вам блок HTML-кода, который вы помещаете на одну из ваших Web-страниц. Бесплатные сервисы создания форм можно найти на сайтах **www.emailmeform.com** и **www.123contactform.com**. Но бесплатные сервисы могут вставлять крошечную пометку внизу на форму (что-нибудь вроде "this form powered by TheSuper-CoolFormCreationCompany" (эта форма поддерживается TheSuper-CoolFormCreationCompany), если вам это не нужно, придется платить.

Примечание

Контактные формы — это простейший пример форм для сбора данных. С помощью сервиса создания форм вы можете создать форму, собирающую массу информации. Например, вы можете получить почтовый адрес для рассылки образцов товаров. Но знайте меру: чем длиннее форма, тем меньше вероятность того, что ее заполнят и отправят.

Наличие рабочего ящика для регистраций на вашем Web-сайте вовсе не означает, что его будет использовать любой. Люди не без основания боятся спама и предоставят вам свои электронные адреса, только если будут считать, что выиграют от этого. Сделайте процесс регистрации легким, т. е. одно поле для адреса электронной почты и кнопка с названием **Sign Up** (Зарегистрироваться) или **Subscribe** (Подписаться). Вам не нужны имена, номера телефонов, даты рождения или любая другая информация, которая может прервать потенциальную регистрацию. Затем опишите преимущества регистрации в одном коротком предложении. Они могут включать скидки, купоны, оповещение о специальных событиях или возможностях, которые не доступны на Web-сайте.

После получения электронных адресов гостей используйте их. Исследование Web-рынка показало, что среднестатистический пользователь просматривает рассылки в электронной почте только 51 сек., поэтому следует включать в них только что-то явно привлекательное, способное заинтересовать читателя за это короткое время. Большинство информационных рассылок форматируются в виде HTML-файлов, но лучше всего создавать их в программе электронной почты, таким образом у вас не будет возможности использовать HTML-средства, не поддерживаемые некоторыми почтовыми клиентами (например, внешние или встроенные таблицы стилей). Картинки применяйте экономно и всегда вставляйте замещающий текст (см. *разд. "Альтернативный текст" главы 7*), т. к. многие почтовые программы блокируют изображения, если не знают ничего об отправителе. Не забывайте вставлять ссылки, заставляющие читателей перейти из электронной почты на соответствующую страницу вашего Web-сайта, чтобы они могли продолжать следить за новостями или новыми выгодными предложениями.

Twitter

Twitter иногда называют системой микроблогов. Она позволяет любому человеку общаться с миром, отправляя текстовые сообщения длиной не более 140 символов

(отсюда префикс "микро"). Конечно, успех ваших Twitter-сообщений — называемых *твитами* — зависит от внимания окружающих к вашим посланиям. У самых популярных пользователей Twitter огромная толпа подписчиков или читателей, которые ловят каждое слово. Часто читатели могут привлечь внимание к вам, передавая ваши сообщения своим читателям.

Следить за понравившимся вам пользователем Twitter можно по-разному. Можно просто просматривать его сообщения на обычной Web-странице (рис. 12.5), но более опытные пользователи поддерживают контакты с множеством людей с помощью сторонних программных средств или доставляя свои твиты на смартфоны подписчиков. (Точно так же пользователи Twitter могут отправлять твиты разными способами, начиная с набора на Web-странице Twitter и заканчивая применением приложений для смартфонов.)

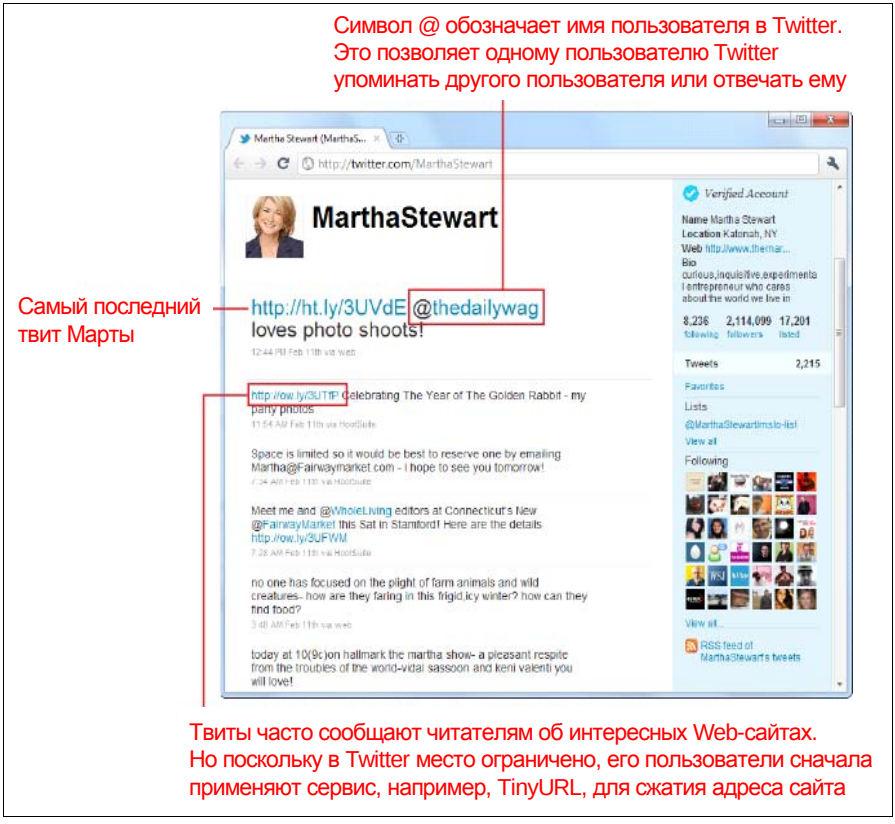


Рис. 12.5. Это поток Twitter-сообщений Марты Стюарт (Martha Stewart) прямо с Web-сайта Twitter

Поразительно, но главная сила Twitter в его жестких ограничениях: длина твита должна быть 140 символов (одно-два предложения). Это заставляет пользователей выражать свои мысли четко и кратко. Например, одно Twitter-сообщение может анонсировать что-то новое, ссылаться на полезный Web-сайт, комментировать текущие события или отвечать на другой твит. Еще одна замечательная особенность

Twitter — огромное количество пользователей. Вы можете стать чьим угодно подписчиком, начиная с далай-ламы и заканчивая Линдси Лохан (Lindsay Lohan). Самых популярных пользователей Twitter можно найти на странице **http://twitaholic.com/top100/followers**. Для регистрации в Twitter перейдите на Web-сайт **http://twitter.com**.

Twitter — коварная среда. Для усвоения ритма и потока сообщений потребуется время, возможно несколько недель. Самая большая ошибка новоиспеченных пользователей Twitter — сразу же начинать рассылку, использовать его как место для публикации случайных и малосодержательных мыслей (примерами могут быть "Obama sucks" (Обама — отстой), "I've lost my socks" (Я потерял свои носки) и "Am eating a piece of gristly chicken, rightnow!" (Прямо сейчас я жую кусочек курицы с хрящиком)), а потом удивляться, почему никто не обращает на них внимание. Но если вы будете публиковать полезные интересные сообщения, то сможете постепенно привлечь больше внимания к себе, к вашей фирме и вашему Web-сайту.

Далее приведено несколько полезных советов начинающему пользователю Twitter.

- *Не спешите.* Не начинайте бездумно отправлять сообщения, пока не разобрались как следует, что работает, а что нет. Для того чтобы войти в курс дела, присмотритесь к другим пользователям. Используйте поисковые средства Twitter (**http://twitter.com**) для поиска твитов, касающихся вас (название вашей компании, ваша сфера деятельности, тематика вашего Web-сайта), и посмотрите, что пишут другие люди.
- *Публикуйте полезную информацию.* Указывайте высококлассные средства, рекомендуйте Web-ссылки и раздавайте полезные советы. Не используйте свою учетную запись в Twitter для беспрестанных продаж и саморекламы, т. к. это быстро наскучит всем.
- *Интересуйтесь мнением других.* Обратитесь к другим пользователям с просьбой об общении. Далее перечислены примеры того, как самые популярные в Twitter компании привлекают к себе внимание: предлагая сигнальные экземпляры книги для просмотра, выясняя проблемы, возникающие у находящихся в употреблении товаров, запуская рекламу там, где люди ведут общий разговор, обмениваясь историями и предложениями. Главное — привлечь Twitter-сообщество *до того*, как у вас появится что-нибудь на продажу. Они оценят это, обеспечат стойкую обратную связь и, возможно, помогут расширить круг осведомленных. Но если вы начнете слишком поздно, все поймут, что вы не воспринимаете Twitter-сообщество всерьез.
- *Отвечайте другим пользователям.* Комментируйте чужие твиты или пересылайте своим подписчикам чужие сообщения, схожие с тематикой вашего сайта. В этом случае вы не просто будете кричать как глашатай, а установите реальные контакты с другими людьми.
- *Продемонстрируйте свои личные качества.* Ваши интересы не ограничиваются бизнесом или профессиональными амбициями. Когда люди становятся подписчиками или читателями того или иного пользователя, они рассчитывают на не-

формальное общение. Поэтому используйте Twitter для того, чтобы поделиться комментариями, затрагивающими жизненные аспекты вашей деятельности.

> ***Примечание***
>
> Есть два способа добиться успеха в Twitter. Во-первых, вы можете начать общаться, будучи таким важным, замечательным и известным, что никто не захочет пропустить ваше сообщение с мобильного в 3 часа ночи. Но если предположить, что вы не знаменитость, следует применить второй вариант: сообщить что-нибудь, касающееся других пользователей Twitter. В этом случае другие могут заметить ваши комментарии, начать внимательно следить за ними и помогут донести ваши слова до более широкой аудитории.

В задачу этой книги не входит научить вас в совершенстве овладеть приемами общения в Twitter. Если вы только начинаете, посмотрите книгу The Twitter Book (изд. O'Reilly), которая может помочь вам начать мыслить как истинный пользователь Twitter и предложит важные рекомендации по рекламированию с помощью Twitter.

Группы

На заре Интернета Web-сайты не были центрами активности. Самое интересное и живое общение происходило на гигантских коллекциях досок объявлений, названных *Usenet*. К несчастью, сеть Usenet пришла в упадок по мере роста Web, спам стал всеобъемлющим, и привлекательные графические сайты превратились в норму. Гораздо позже компания Google купила коллекцию групп Usenet, которая переживает малый ренессанс как часть сервиса Google Groups (см. **http://groups.google.com**).

Прежде чем создавать группу, необходимо понять, где она будет действовать, а где нет. Google Groups не помогут завязать беседу с незнакомыми вам людьми и теми, у кого нет личного интереса к вашему Web-сайту. Потенциальные покупатели, любопытные Web-туристы и пользователи, случайно натолкнувшиеся на ваш сайт, не готовы преодолевать трудности присоединения к группе ради беседы с вами. Лучше привлечь их через Facebook (см. *соответствующий раздел далее в этой главе*).

С другой стороны, возникают ситуации, в которых группы действуют лучше свободно объединяющихся социальных сайтов вроде Facebook. Далее приведено несколько примеров.

- *У вас есть узкоспециальная тематика.* Это тематика, привлекающая предметно-ориентированных фанатов, и пока в других местах не обсуждаемая. Это маловероятно, потому что в Web уже есть форумы для поющих а капелла, для людей, жалующихся на своих начальников, форумы фанатов вампиров и т. д.
- *Вы обеспечиваете поддержку товара.* Если у вас бизнес, имеет смысл отвечать на общие вопросы на форуме, так чтобы каждый нашел ответ на интересующий его вопрос, и вам не пришлось бы самому отвечать одно и то же снова и снова в сообщениях электронной почты.
- *Ваша группа продолжение реального мира.* В данном случае люди в группе уже знают друг друга и у них есть причина для объединения. Например, группа может быть удобным для вас способом общения с членами кружка по вязанию.

- *Вашей группе нужна конфиденциальность.* Например, если вы создали группу поддержки бывших наркоманов и хотите не допустить в нее интернет-шпану. В этом случае можно создать группу Google и ограничить участие в ней.

Самые лучшие форумы самоуправляемы. Если вы собрали вместе нужные ингредиенты, форум может успешно развиваться без вашего вмешательства. Воспринимайте форумы, как праздничный обед, который вы устраиваете, и нужно только начать беседу, а затем вежливо уйти в тень. Если вы используете форум для ответов на технические вопросы, вы можете существенно снизить свою рабочую нагрузку. Например, на многих форумах особое значение придается клиентам или экспертам, помогающим друг другу. Это означает, что участники форума обмениваются информацией, советами и ответами, а ваше вмешательство необходимо только для разрешения долгих споров.

Введение в Google Groups

Самый легкий способ создать свою группу — использовать бесплатный сервис, такой как Google Groups. Далее приведены некоторые важные подробности.

- Когда создается группа, вы получаете уникальный легко запоминающийся URL. Это адрес группы, который никогда не меняется (если вы его не измените сами).
- Создатель группы (вы) разрешает и запрещает публиковать сообщения. Когда группа начнет активно действовать, вы сможете передать другим членам группы некоторые или все свои полномочия по управлению.
- Процессом регистрации Google управляет самостоятельно. Это означает, что вам не нужно вручную добавлять и удалять членов группы (хотя вы можете, если захотите).
- Каждый участник группы может выбрать чтение сообщений в сети или регулярное получение их по электронной почте, которую Google отправляет автоматически.
- Макет страницы Google — мечта усталого Web-путешественника. На ней легко искать сообщения, просматривать сразу все ответы на сообщение, делать закладки на сообщениях, которые хочется отслеживать, и т. д.
- Хотя Google и отображает рекламные объявления в углах окон вашей группы, выбираются только относящиеся к контенту. Например, если самая популярная тематика в группе — любимые DVD-диски, вы скорее всего увидите рекламу Blu-ray DVD-плейеров и клубов кинофильмов, заказываемых по почте.

Дополнительную информацию о Google Groups можно найти на странице **http://groups.google.com/googlegroups/overview.html**.

Создание группы

Для создания новой группы в сервисе Google Groups выполните следующее.

1. Перейдите на Web-сайт **http://groups.google.com.** Для начала щелкните мышью кнопку **Create a group** (Создать группу).

Для создания группы понадобится учетная запись Google. Если она у вас есть, входите прямо сейчас. В противном случае необходимо зарегистрироваться, щелкнув мышью ссылку **Create an account** (Зарегистрироваться). После завершения процесса и активации учетной записи вы можете войти в Google Groups и продолжить работу.

Теперь появится страница **Create a group** (Создание группы), показанная на рис. 12.6.

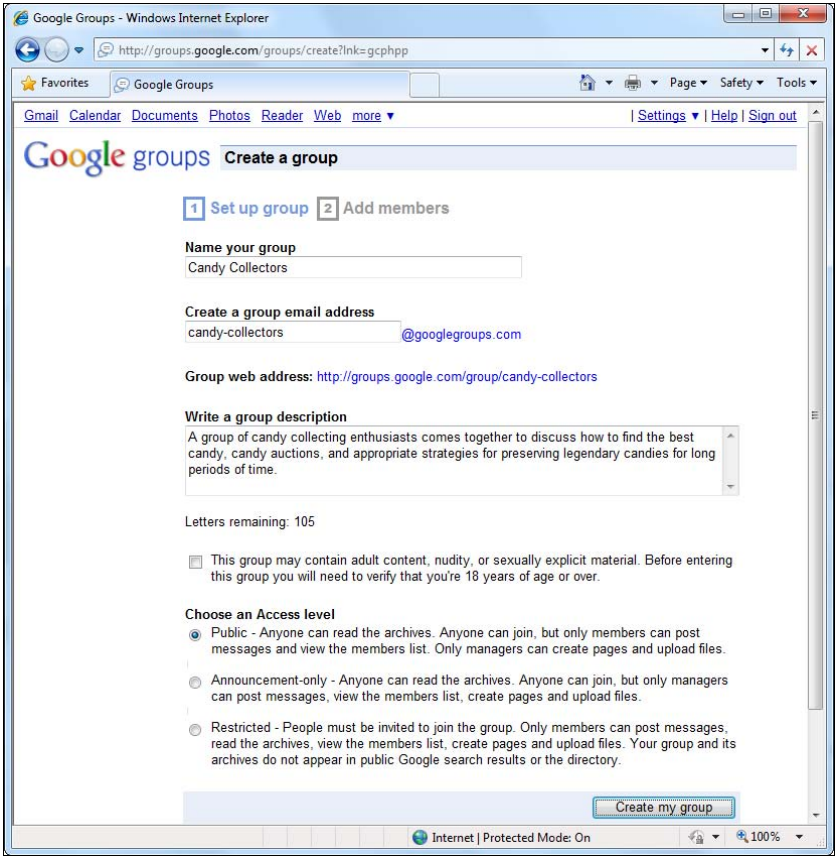


Рис. 12.6. Создание группы Google требует двух шагов. На первом вы определяете базовые сведения о вашей группе, включая ее название и адрес электронной почты

2. Введите информацию о вашей группе.

   Название группы идентифицирует вашу группу, например, Candy Collectors (Коллекционеры леденцов).

   Адрес электронной почты группы — это вариант имени, и он действует как адрес или как URL. Пробелы не допускаются, но в крайнем случае можно использовать дефисы. Подходят оба варианта Candy-Collectors и CandyCollectors. Адрес электронной почты также становится частью URL группы, поэтому убедитесь, что его легко запомнить.

В описании группы с помощью двух-трех предложений поясняется ее назначение.

Уровень доступа определяет, кому разрешено публиковать сообщения. Если вы хотите создать полностью открытую группу с доступом для всех пришедших, выберите вариант **Public** (Общий доступ), подходящий для большинства Web-групп. Любой, кто натолкнется на вашу группу, сможет присоединиться к ней по своему усмотрению без вашего вмешательства. Если вы хотите использовать группу исключительно как место публикации собственных размышлений, выберите вариант **Announcement-only** (Только для объявлений). Но в таком случае лучше поместить ваши сообщения прямо на ваш Web-сайт, а не в группу. И наконец, если вы хотите собственноручно отбирать участников, выберите **Restricted** (Доступ ограничен). В этом случае к группе смогут присоединиться только те пользователи, которых вы пригласили.

На странице есть еще флажок, разрешающий контент, предназначенный для взрослых. Если оставить его не установленным, Google автоматически будет блокировать пикантные сообщения, щадя вашу стеснительность.

3. Щелкните мышью кнопку **Create my group** (Создать группу).

   Прежде чем продолжить, вам нужно будет скопировать несколько букв с картинки, чтобы доказать, что вы реальный человек, а не разгулявшаяся программа генерации группы Google.

4. Решите приглашать или просто добавлять первоначальных членов вашей группы.

   Обычно новые члены группы приглашаются по электронной почте (рис. 12.7). Для того чтобы по-настоящему стать членом группы, приглашенный должен посетить сайт и выразить согласие. Но если вы предпочитаете сделать определенных людей членами группы автоматически, не требуя подтверждения с их стороны, можно щелкнуть мышью вариант **Add members directly** (Добавить участников). В любом случае участники должны создать учетную запись Google (если у них до сих пор ее нет), прежде чем смогут публиковать сообщения.

5. Заполните начальный список членов группы.

   Создайте список адресов электронной почты, отделенных друг от друга запятыми. Когда создание группы будет завершено, Google сообщит этим людям по электронной почте о возможности присоединения к группе (или о том, что они стали участниками группы, если вы выбрали вариант **Add members directly**).

   Примечание

   Не беспокойтесь, если у вас под рукой нет адресов электронной почты. Вы не должны кого-либо приглашать именно сейчас. На эту страницу можно вернуться позже и пригласить людей после создания группы.

6. Если вы выбрали вариант **Add members directly** (Добавить участников), задайте параметры подписки.

   Если добавляете, а не приглашаете участников группы, на экране появится раздел **Email subscription options** (Установить тип доставки). В нем можно выбрать

способ взаимодействия новых членов с группой с помощью электронной почты (У вас не будет такой возможности, если вы просто приглашаете новых членов. В этом случае они самостоятельно выберут параметры подписки, когда присоединятся к группе.)

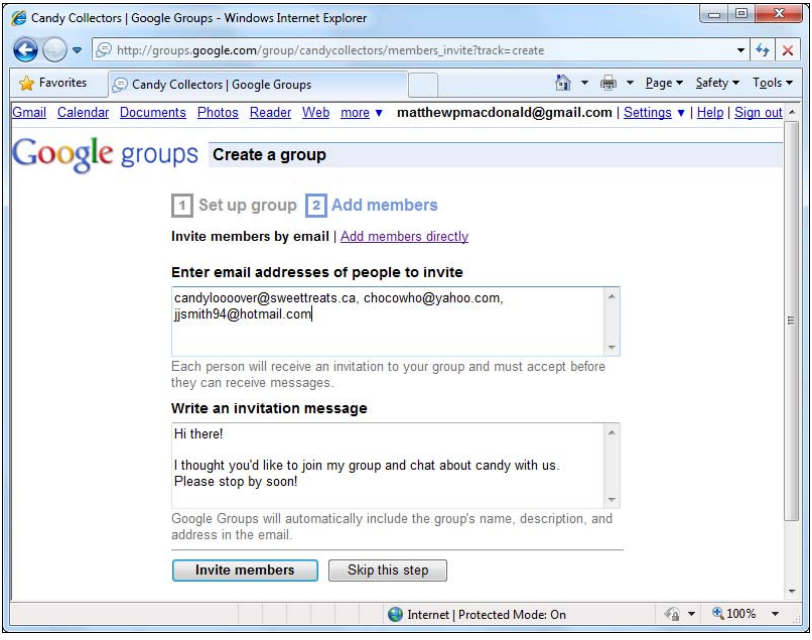


Рис. 12.7. Второй шаг — выбор ваших первых членов группы. Помните, что если создаете публичную группу, новые участники смогут через Google присоединиться к ней в любое время

Возможны следующие типы доставки.

- **No email** (Без получения сообщений электронной почты). Члены группы должны будут читать групповые сообщения в браузере, посещая сайт группы.
- **Send email for each message** (Электронная почта). Google отправляет по электронной почте каждое сообщение каждому участнику группы. Это неудачный вариант, если только у вас не очень спокойная дискуссионная группа.
- **One summary email a day** (Краткий обзор сообщений). Члены группы получают по электронной почте одно письмо в день (если за день появилось хотя бы одно новое сообщение). Это письмо содержит список заголовков новых сообщений с расположенными рядом ссылками на полный текст сообщения. Этот вариант позволяет оставаться на пике групповой активности и следить за интересными сообщениями.
- **One email with all activity** (Дайджест сообщений). Участники группы получают весь новый контент один раз в день в едином гигантском письме. Этот вариант не приведет к такому загромождению почтового ящика электронной почты, как вариант с получением каждого сообщения в отдельном письме, но

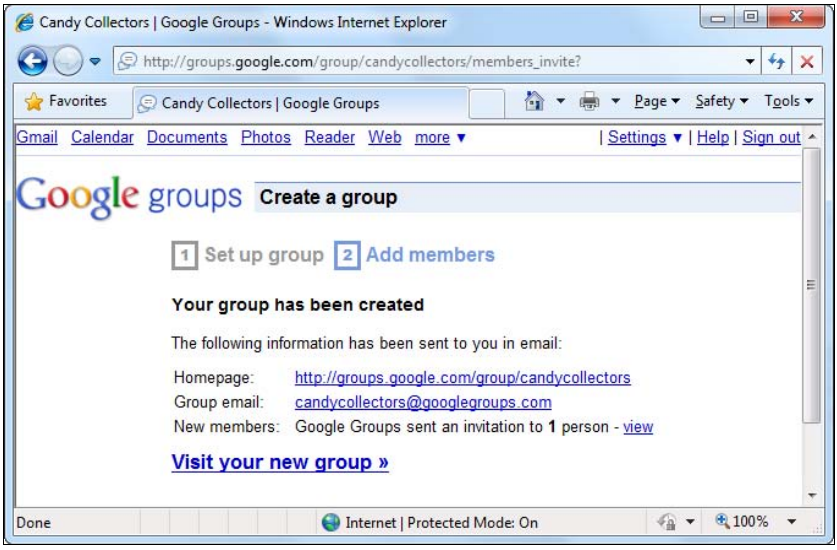


Рис. 12.8. Сводная страница Google Groups содержит важную информацию, включая URL главной страницы группы (на которую вы сможете вставить ссылку со своего Web-сайта) и адрес электронной почты (который позволит вам отправлять сообщения по электронной почте)

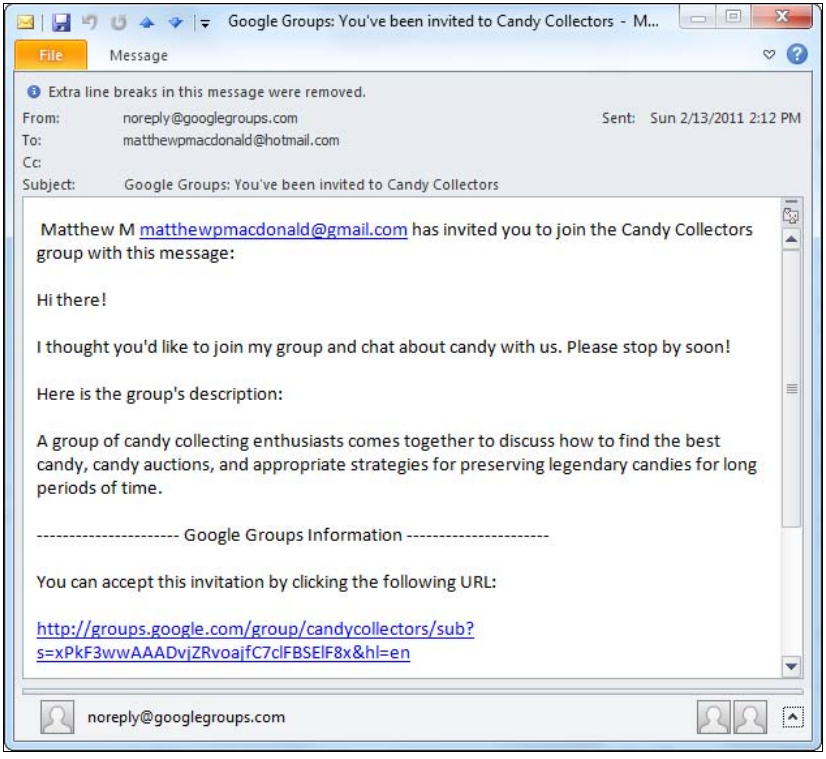


Рис. 12.9. Добро пожаловать на борт!
Вы приглашаетесь принять участие в новой группе Google

в активно общающейся группе он приведет в результате к рассылке длинных электронных писем, которые станет неудобно читать.

Независимо от выбранного вами варианта доставки члены группы позже смогут изменить эту настройку в соответствии со своими личными вкусами.

7. Введите приветственное сообщение и щелкните мышью кнопку **Invite members** (Пригласить участников) или **Add members** (Добавить участников), если вы выбрали непосредственное добавление членов группы.

   Google создаст группу и покажет вам сводную информацию (рис. 12.8).

Спустя короткое время приветственные письма будут разосланы всем участникам группы из первоначального списка (рис. 12.9).

Участие в группе

Когда вы впервые начнете руководить своей группой, то увидите, что это очень скучное занятие. Почему бы не опубликовать первую тему, чтобы начать обсуждение?

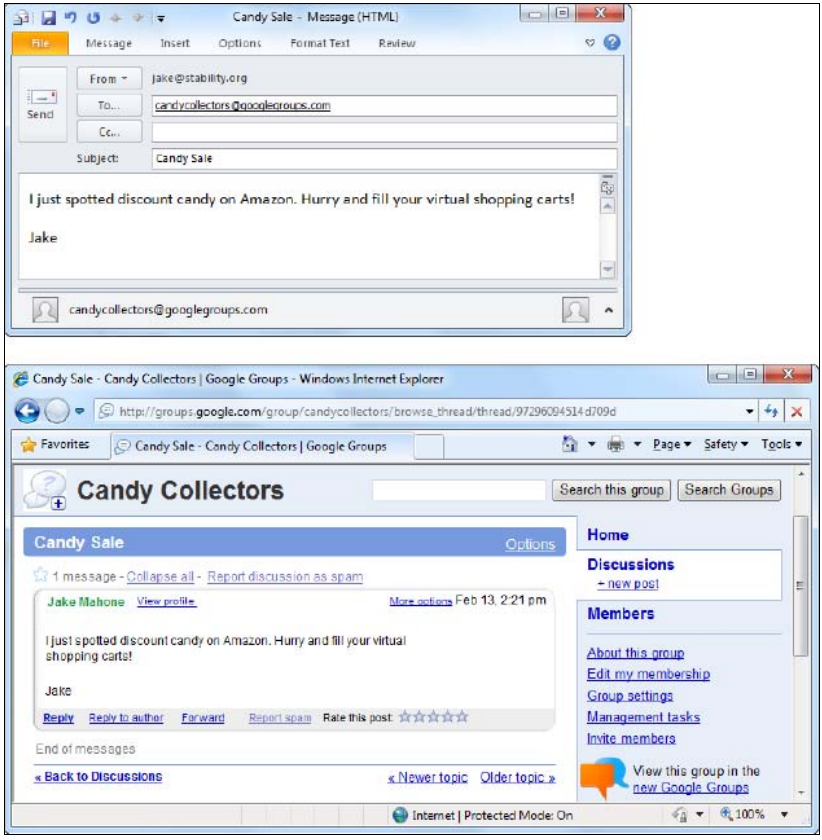


Рис. 12.10. *Вверху*: программа электронной почты собирается отправить сообщение в группу Google. *Внизу*: после получения сообщения электронной почты сервисом Google Groups оно становится обычным сообщением группы (показанным здесь). Вы можете щелкнуть мышью ссылку **View profile** (Просмотреть профиль), чтобы узнать больше об авторе сообщения, и **More options** (Дополнительные параметры), чтобы ответить, переслать или удалить сообщение (при условии, что вы владелец группы)

Для этого Google предоставляет две возможности. Можно вставить сообщение с главной страницы группы, щелкнув кнопкой мыши ссылку **Start a new topic** (Начните обсуждение новой темы). Или, если вы действительно спешите, можно просто послать сообщение по электронной почте на адрес группы. Google преобразует его в тему обсуждения группы (рис. 12.10).

Конечно, дискуссии целиком состоят из прямых и обратных обменов информацией. После того как кто-то опубликовал сообщение, вы можете прочитать его, а затем щелкнуть кнопкой ссылку **Reply** (Ответ) и опубликовать ответ. Google *нанизывает на нитку* сообщения и ответы на них, т. е. группирует их вместе, поэтому вы легко можете видеть, какое сообщение связано с какой темой (рис. 12.11).

Примечание

Если добраться до правого нижнего угла дискуссионной страницы вашей группы, вы увидите небольшую область с рекламными объявлениями от Google. Google выбирает их, основываясь на контенте вашей группы, и все время совершенствует отбор, определяя, какие рекламные объявления получили от участников группы больше всего щелчков кнопкой мыши.

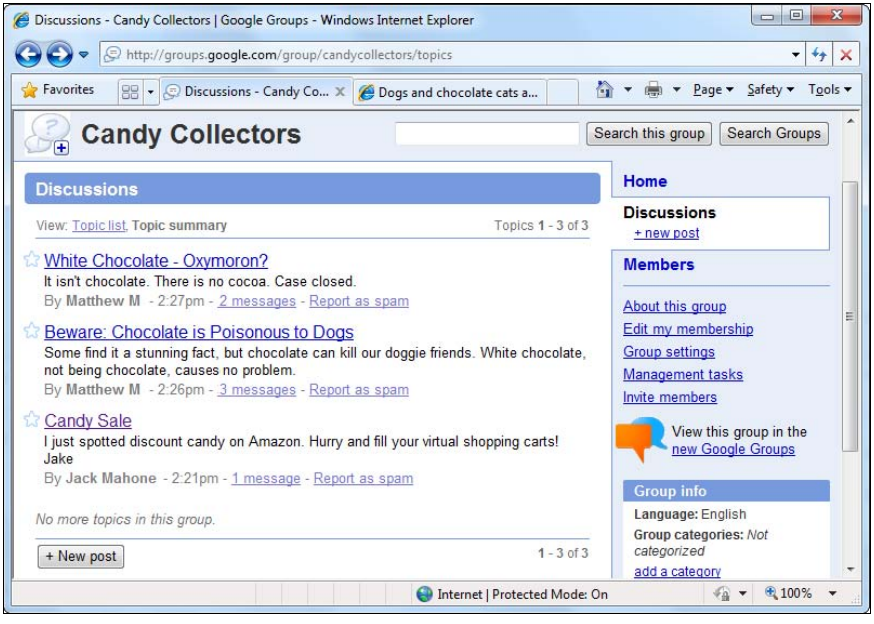


Рис. 12.11. При просмотре группы вы видите только список тематических сообщений, а не ответов. Но в каждый элемент списка включена информация о количестве ответов. В данном примере есть три сообщения и в целом 6 ответов. У первого сообщения один ответ, а у второго — два. (Щелкните заголовок тематического сообщения и увидите его полный текст и все ответы)

Теперь у вас есть полностью рабочая группа. С этого момента основная трудность не в использовании группы, а в привлечении достаточного количества интересных людей, чтобы превратить ее в живое сообщество.

Управление вашей группой

Когда группу посещают обычные участники, у них есть возможность публиковать новые сообщения, отвечать на уже имеющиеся и изменять параметры своей доставки (щелкнув кнопкой мыши ссылку **Edit my membership** (Изменить настройки участия)). Участники группы могут использовать эту последнюю возможность для получения сообщений группы, автоматически отправляемых им по электронной почте или для просмотра сводной картины активности группы.

Если же группу посещает ее *создатель*, появляются дополнительные ссылки. На странице вашей группы вы увидите ссылку **Invite members** (Пригласить или добавить участников), которая позволит отправить приветственные письма новой группе участников. Ссылки **Group settings** (Настройки группы) и **Management tasks** (Административные задачи) позволят вам управлять гораздо более широким набором параметров.

В раздел **Group settings** включено множество параметров, собранных в несколько подкатегорий. Далее приведено краткое описание того, что они предлагают.

- **General** (Общие). На этой странице приводится основная информация о группе, название группы, описание, URL и адрес электронной почты. Для редактирования этих параметров щелкните мышью кнопку **Edit** (Изменить).
- **Access** (Доступ). Параметры этой категории позволяют определить, кто может выполнять различные задачи. Например, можно выбирать, кому разрешить чтение и создание тематических сообщений (всем или только участникам группы), кому разрешить приглашать новых участников (только вам или любому участнику группы), и кому разрешается присоединиться к группе. Последний параметр самый интересный. Вы можете разрешить присоединиться любому пользователю, ограничить группу только приглашенными вами людьми или заставлять незнакомцев становиться участниками группы. Если вы остановитесь на последнем варианте, любой может попытаться присоединиться, но у вас есть возможность просмотреть его заявление и вынести окончательное решение о приеме или отказе. Вы даже можете попросить Google задать многообещающим кандидатам конкретный вопрос. Затем, после просмотра ответов вы сможете определить их ценность для группы. И наконец, вы можете в параметрах доступа включить предварительный просмотр сообщений ведущим (модератором). В этом случае Google отправляет каждое новое сообщение вам перед тем, как опубликовать его. Сообщения не появятся до тех пор, пока вы не дадите им зеленый свет (а если не дадите, то и никогда не достигнут группы). Применяйте этот параметр, только если у вас много лишнего времени.

ЧАСТО ЗАДАВАЕМЫЙ ВОПРОС

Групповые ограничения

Должен ли я ограничивать права на присоединение к моей группе и публикацию сообщений?

Очень хочется способствовать участию людей в вашей группе, но при этом препятствовать проявлениям эгоизма. Web-пользователи нетерпеливы и легко выходят из себя. Если вы поставите барьер на пути потенциальных участников группы, они могут просто уйти.

С другой стороны, бывают ситуации, когда ограниченное участие в группе очень оправдано. К ним относятся следующие две: когда вы хотите обсуждать полусекретную информацию, например стратегию компании, или если вы боитесь, что ваша тема привлечет множество неподходящих пользователей. Предположим, что вы создали группу Software-Piracy (Компьютерное пиратство), чтобы обсудить социальные последствия компьютерного пиратства, вас могут завалить запросами на последние версии украденного программного обеспечения. Как правило, ограничения имеют смысл, только если вы применяете их для поддержания контроля качества вашей группы.

То же самое справедливо и для сообщений, контролируемых ведущими или модераторами. Самые здоровые сетевые сообщества — саморегулируемые системы. Если участник нечаянно обидит основное сообщество, другие его или ее поправят; если обида нанесена преднамеренно, большинство в итоге проигнорирует провокацию. Возможно, вам придется вмешиваться иногда для лишения участника прав, но просматривать каждое сообщение — это перебор. Это добавит вам огромный объем дополнительной работы и существенно снизит динамизм вашей группы, т. к. новое сообщение сможет появиться только после того, как вы его просмотрите, что, как правило, занимает несколько часов после написания сообщения. Для фанатов Web, ждущих незамедлительного вознаграждения, это плохие новости.

Параметры для ограничения прав участников и управления просмотром сообщений ведущими вы найдете в разделе **Access** (Доступ) на странице **Group settings** (Настройки группы).

- **Appearance** (Внешний вид). Если вы хотите, чтобы ваша группа выделялась, можно применить эти параметры для задания на сайте группы разных шрифтов и привлекательной цветовой схемы.
- **Navigation** (Перемещение). Этот раздел позволяет скрыть некоторые ссылки, отображаемые на главной странице группы. Например, если вы не хотите разрешать участникам группы выгружать и совместно использовать файлы, вы можете удалить ссылку **Files** (Файлы).
- **Email delivery** (Доставка сообщений по электронной почте). Если некоторые участники вашей группы получают сообщения по электронной почте, вы можете использовать эту группу параметров для настройки текста нижнего колонтитула и управления ответами на сообщения, посланные по электронной почте. Обычно ответ публикуется рядом с исходным сообщением. Чтобы вся группа могла его видеть, но вы можете отправить ответы только автору исходного сообщения или владельцу группы (вам).
- **Categories** (Категории). Здесь вы можете выбрать категорию для вашей группы (например, People | Relationships or Health | Addictions (Люди | Отношения или Здоровье | Пристрастия)). После этого пользователям Google будет легче случайно натолкнуться на вашу группу.
- **Advanced** (Дополнительные). В этом разделе вы сможете выполнить несколько низкоуровневых административных задач, например, задать группу, которая хранится на другом сервере, или удалить вашу группу целиком.
- **Spam** (Защита от спама). В этом разделе можно управлять тем, как Google обращается с потенциальным спамом. Например, можно удалять такие сообщения немедленно или помещать их в раздел сообщений, управляемых ведущими или модераторами (это вариант по умолчанию).

Ссылка **Group settings** (Настройки группы) предлагает вам большой набор параметров, а ссылка **Management tasks** (Административные задачи) сосредоточена на одном аспекте — работе с участниками группы. С помощью страницы **Management tasks** вы можете просмотреть полный список участников группы, увидеть, кто не ответил на приглашение в группу, исключить участников, нарушающих нормальную работу группы, и предоставить другим членам управленческие полномочия (рис. 12.12).

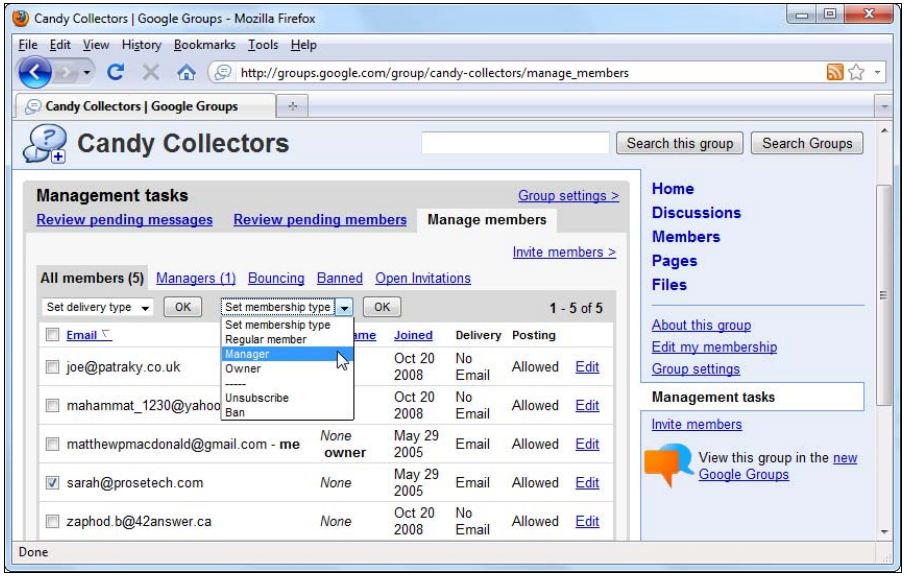


Рис. 12.12. Выделена участник группы Сара (Sarah), обратите внимание на флажок слева от ее имени. С помощью раскрывающихся списков в верхней части страницы вы можете быстро изменить ей тип доставки сообщений или назначить другой уровень доступа. В данном примере ее собираются сделать модератором, предоставив ей право удалять сообщения, приглашать новых участников и изменять настройки группы (но не удалять группу целиком). К другим вариантам относятся **Unsubscribe** (Отменить подписку), удаляющий участника из группы, и **Ban** (Исключить), удаляющий со строжайшим предупреждением, в этом случае участнику группы никогда не будет разрешено вновь присоединиться к группе

Facebook

Создание группы с нуля может оказаться не простым делом. Если новая группа быстро не приобретет популярность, она превратится в пустынное место, где никто не захочет задерживаться. Вот почему все больше Web-обитателей не пытается сделать это в одиночку. Вместо этого они приводят свою публику в существующее сообщество — в то, что организовано на базе сайта социальных сетей. А в социальных сетях нет более известной компании, чем Facebook.

Facebook начиналась как сеть для тесного общения студентов колледжа. Всего за несколько лет она выросла как на дрожжах в социальный сайт, где сотни миллионов обычных людей находят кого угодно, от давно потерянных любимых до почти забытых школьных приятелей. Благодаря Fan-страницам (средство, о котором вы

узнаете в этом разделе) Facebook даже превратилась в инструмент для коммерческих и некоммерческих организаций, обладающий достоинствами и недостатками.

Создание Fan-страницы

Fan-страница — это страница Facebook, которая создается для продвижения чего угодно, включая компании, идеи, торговые марки, книги, телевизионные шоу и т. д. Если вы пытаетесь продвигать себя, Fan-страница — отличный выбор. Это замечательный инструмент для музыкантов, актеров, журналистов и для всех, кто хочет обратиться к широкой аудитории. Например, **www.facebook.com/kristof** — Fan-страница Николаса Д. Кристофа (Nicholas D. Kristof), ведущего колонку в газете New York Times. Он использует свою страницу для комментирования дел и обсуждения с читателями злободневных вопросов. Совсем иная, хотя и более аппетитная **www.facebook.com/benjerry**, Fan-страница мороженого компании Ben & Jerry. Они используют более непринужденную и менее формальную версию своего Web-сайта, полную самых причудливых обсуждений забытых сегодня вкусов мороженого.

Fan-страница похожа на персональную страницу Facebook, но она больше подходит для рекламы. Ее может посетить любой человек и прочесть контент, даже если у него нет собственной учетной записи в Facebook. Те же, у кого есть учетные записи, могут применять обычные средства Facebook, например, щелкнуть мышью кнопку **Like** (Мне нравится), чтобы стать поклонником (follower) Fan-страницы, публиковаться на ее стене и присоединиться к любым дискуссиям, ведущимся на ней.

> ***Примечание***
>
> Личная страница Facebook (также называемая страницей профиля) обладает бóльшими ограничениями, чем Fan-страница. Она лучше подходит для связи с вашими друзьями или использования сетевых деловых контактов. А с помощью Fan-страницы гораздо легче представить себя, свою компанию или идеи множеству незнакомых вам людей.

Далее описано, как создать Fan-страницу Facebook[1].

1. Для создания Fan-страницы перейдите на Web-страницу **www.facebook.com/pages/create.php**.

   Facebook делит эту страницу на две области (рис. 12.13). Слева вы можете создать страницу сообщества или публичную страницу, которая используется совместно, и если достаточно популярна, то будет поддерживаться сообществом Facebook. Справа можно создать "официальную" страницу по вашему желанию.

   Если у вас нет учетной записи Facebook, не беспокойтесь, ее можно создать попутно.

[1] На момент выхода русского перевода книги интерфейс Facebook изменился, но общие принципы создания Fan-страницы остались прежними. — *Прим. ред.*

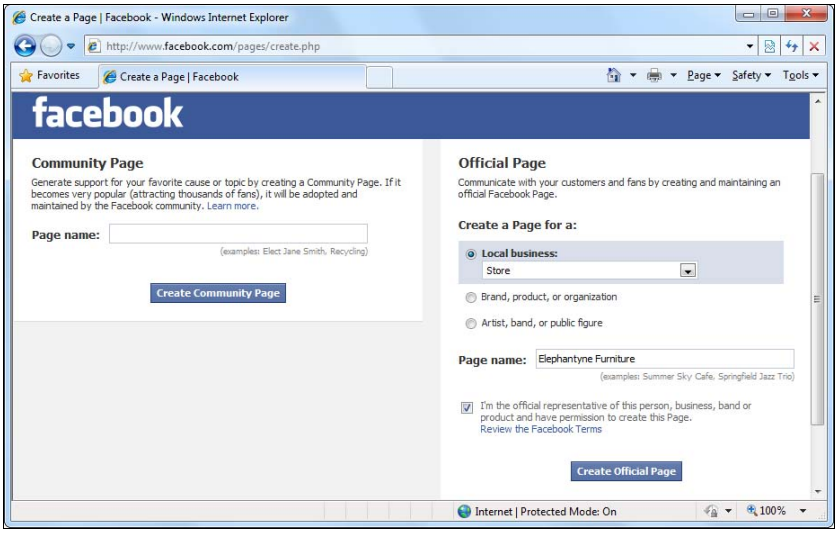


Рис. 12.13. Это Fan-страница Facebook, которая будет создана для магазина Elephantyne Furniture

2. Выберите для кого создается страница.

   Возможные варианты — **Local business** (Местный бизнес или место), **Brand, product, or organization** (Бренд, продукт или организация) или **Artist, band, or public figure** (Исполнитель, музыкальная группа или известная личность). Если вам кажется, что страницы разных видов объединены в одну категорию, не волнуйтесь. После того как выберете вариант, под категорией появится более подробный список, позволяющий точнее задать вид бизнеса, бренда, организации или персону (на рис. 12.13 из списка выбран **Store** (Магазин)).

3. Введите название страницы.

   К удачным примерам, зависящим от того, что вы собираетесь продвигать, можно отнести Larry S. Tindleman или Larry's Polka Band или Tindleman World-o-Shoes. В название страницы можно включать пробелы. Название страницы всегда отображается на Fan-странице, и Facebook применяет его в URL Fan-страницы.

4. Щелкните мышью переключатель **I'm the official representative** (Я официальный представитель), если вы им действительно являетесь.

   В некоторых случаях пользователи Facebook создают Fan-страницы для своих любимых торговых марок. Часто компания, которой принадлежит торговая марка, позже наследует эту страницу, но порой она позволяет поклонникам торговой марки продолжать управлять страницей.

5. Щелкните мышью **Create Official Page** (Создать официальную страницу).

   Теперь вам нужно войти в систему с вашей учетной записью Facebook или создать новую.

   Для создания новой учетной записи Facebook нужно представить всего несколько подробностей: ваш адрес электронной почты, пароль, который хотите ис-

пользовать, и вашу дату рождения. Для подтверждения получения вашей учетной записи щелкните кнопкой мыши ссылку в сообщении, которое вам пришлет Facebook.

Когда вы завершите вход или регистрацию, Facebook создаст вашу Fan-страницу.

Любая Fan-страница начинается с двух общедоступных вкладок: **Wall** (Стена) (на которой вы и ваши поклонники могут публиковать краткие сообщения) и **Info** (Информация) (на которой вы можете распространять основные данные о вас или вашей компании). У вашей Fan-страницы также есть вкладка **Get Started** (Начало работы), предназначенная только для ваших глаз. Сначала Facebook переносит вас именно туда (рис. 12.14).

6. Щелкните кнопкой мыши ссылку **Edit Info** (Редактировать) в верхней части страницы прямо под названием.

   Теперь самое время добавить основные подробности о вашей странице и картинку. Вы увидите новую страницу с множеством вкладок, на которых вы можете разместить кучу информации для вашей новой Fan-страницы. Вы выбираете категорию слева, на боковой панели, и вводите информацию в правой части страницы.

   Для начала выберите строку **Basic Information** (Основная информация). Вы можете ввести адрес и описание (что очень важно, т. к. данная информация появится на вкладке **Info** (Информация) вашей Fan-страницы). Когда закончите, щелкните мышью кнопку **Save Settings** (Сохранить настройки).

   Затем щелкните мышью строку **Profile Picture** (Фотография профиля). Здесь вы сможете выгрузить изображение, которое будет располагаться в верхнем правом углу вашей Fan-страницы. Эта деталь крайне важна, потому что ее первую видят посетители, и именно она идентифицирует вас.

7. Для того чтобы вернуться на Fan-страницу, щелкните мышью кнопку **View Page** (Просмотреть страницу) (в правом верхнем углу).

   Несмотря на то, что у вас нет ощущения, что сделано много, ваша Fan-страница Facebook теперь жива и доступна любому пользователю социальной сети. Люди за пределами Facebook тоже могут ее видеть, но для того, чтобы она появилась в поисковой системе Google, потребуется какое-то время. В данный момент вы можете связаться с ней напрямую с помощью URL Fan-страницы.

8. Запишите URL вашей Fan-страницы в Facebook.

   Быстрее всего получить URL страницы можно из адресной строки вашего браузера. К вашему URL относится все до символа вопросительного знака. В него входит сайт Facebook, имя страницы и последовательность цифр.

   Например, если в адресной строке вы видите следующие данные:

   http://www.facebook.com/pages/Elephantyne-Furniture/129277497141285?sk=wall#!/pages/Elephantyne-Furniture/129277497141285

   У вашей страницы следующий URL:

   http://www.facebook.com/pages/Elephantyne-Furniture/129277497141285

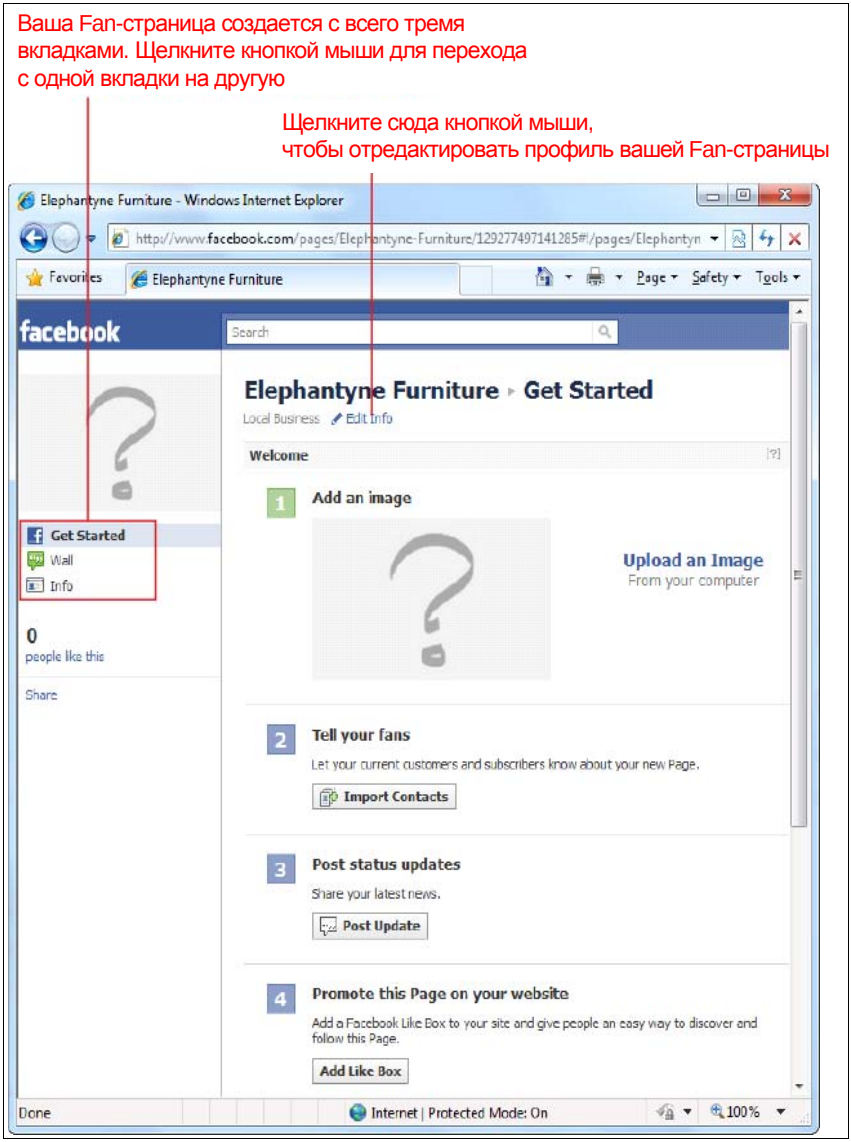


Рис. 12.14. В данный момент Fan-страница Elephantyne Furniture представляет собой пустую заготовку без картинки или данных. На вкладке **Get Started** приводятся ссылки на некоторые самые типовые установочные задачи, такие как выгрузка картинки профиля, импорт списка контактов электронной почты, публикация обновлений статуса и генерация HTML-кода для ссылки **Like** (Мне нравится), которую можно поместить на страницу вашего Web-сайта. (С помощью этой кнопки читатели из Facebook премируют людей и вещи, которые им нравятся. Чем больше людей щелкнет мышью ссылку **Like** вашего Web-сайта, тем лучше ваша репутация в Facebook.)

Совет

URL Fan-страниц трудно запоминаются из-за длинных цифровых последовательностей на конце. Но вы, возможно, заметили, что у крупных игроков более привлекательные URL без каких-либо цифровых окончаний. Вы тоже можете подать заявку на компактный, без цифр URL в Facebook, но сначала нужно приобрести не менее 25 поклонников вашей страницы. (Поклонники — это пользователи, нажавшие кнопку **Like** (Мне нравится) на вашей странице.) Дополнительную информацию см. на странице **http://tinyurl.com/4z9wzlz**.

СЕКРЕТЫ РЕМЕСЛА

Под именем вашей Fan-страницы

В прошлом создатели Fan-страниц вынуждены были использовать свою персональную учетную запись Facebook для выполнения целого ряда задач. Это было не лучшим решением. Например, это неудобно, если вы не хотите смешивать свои личные дела и бизнес или не хотите брать на себя роль официального пресс-секретаря вашей организации (возможно, Fan-страница на самом деле принадлежит группе людей, или вы не хотите подвергать сомнению торговую марку из-за собственного присутствия).

Но совсем недавно создатели Facebook ввели новое средство, позволяющее делать множество вещей, которые раньше выполнялись с помощью персональной учетной записи Facebook — связываться со своими друзьями, другими понравившимися вам Fan-страницами и т. д. — применяя учетную запись Fan-страницы.

Для переключения на учетную запись вашей Fan-страницы щелкните мышью кнопку **Account** (Аккаунт) в правом верхнем углу и выберите команду **Use Facebook As Page** (Использовать Facebook как). На экране появится список всех ваших Fan-страниц. Щелкните мышью кнопку **Switch** (Переключиться), расположенную рядом с Fan-страницей, под именем которой вы хотите войти в Facebook.

После того как вы отождествили себя с вашей страницей, можно оставлять сообщения на стенах других пользователей, отслеживать активность на вашей Fan-странице, вставлять комментарии в продолжающиеся обсуждения и нажимать кнопку **Like** (Мне нравится), оценивая другие компании (что открывает широкую дорогу для перекрестного продвижения). Например, Elephantyne Furniture может выбрать щелчок кнопки **Like** у оптовика, продавца или поставщика, с которым работает. В Facebook тут же будет показано, что магазину Elephantyne Furniture нравится личность или фирма без отображения каких-либо личных характеристик пользователя, управляющего Fan-страницей.

Когда будете готовы снова вернуться в свой личный профиль, опять щелкните мышью кнопку **Account** (Аккаунт) и выберите команду **Switch back ...** (Вернуться назад ...). Точный текст зависит от вашего имени — например, если вы зарегистрировались в Facebook как Rob Sampson, вы увидите команду **Switch back to Rob** (Вернуться к Rob).

Добавление вкладок на вашу Fan-страницу

Ваша страница создается с двумя общедоступными вкладками: **Wall** (Стена) и **Info** (Информация).

Wall (Стена) — это центральное место для ваших публикаций и публикаций ваших поклонников. Для того чтобы следить за вашей страницей, люди читают сообщения на вашей стене. Если она полна оживленной болтовни, это верный признак того, что страница процветает. Но самая важная характеристика стены — способ доставки ваших сообщений вашим поклонникам. Вы публикуете обновление статуса, фотографию, ссылку на Web-сайт или видеоролик и ваши поклонники немедленно получат уведомления об этом (рис. 12.15).

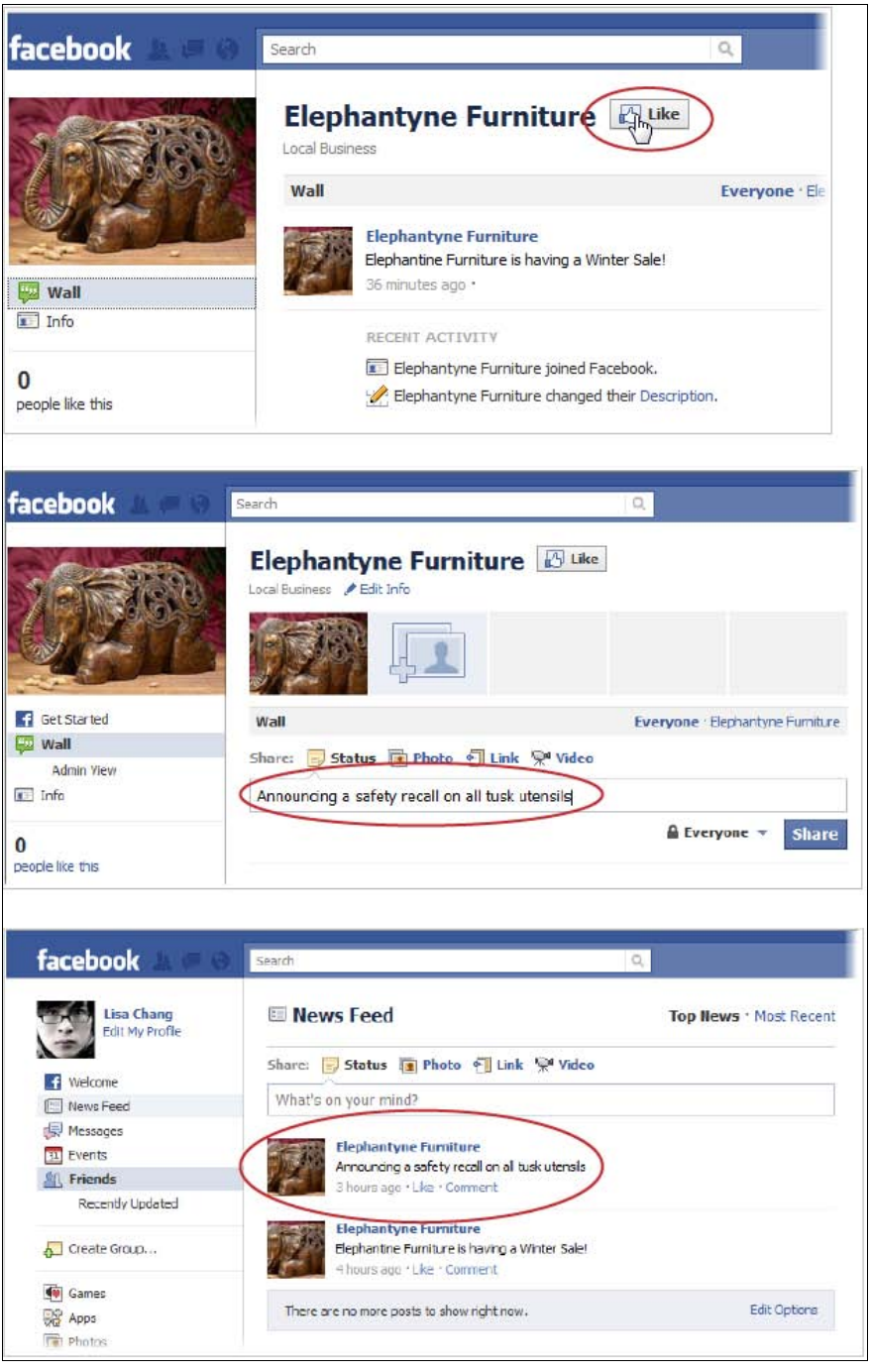


Рис. 12.15. Все заработает, когда кто-нибудь щелкнет мышью кнопку **Like** (вверху). Теперь, когда вы публикуете сообщение на своей стене (в середине), информация приходит в ленту новостей вашего поклонника (внизу)

Совет

Помните о том, что Facebook целиком ориентируется на сообщество. Вам не нужно отвечать на каждый комментарий, но вы должны откликаться часто. Благодарите за похвалу и просите прощения, если порицают. См. Fan-страницу Facebook (**http://tinyurl.com/6amf47k**) небольшой компании по продаже ювелирных изделий, которая все делает правильно.

Вкладка **Info** (Информация) также важна, но далеко не так увлекательна. На ней отображаются подробности, приведенные вами в разделе **Basic Information** (Основная информация) вашего профиля, например описание.

Это вполне приемлемое начало, но солидной Fan-странице нужны дополнительные вкладки. Строго говоря, Facebook называет их apps (сокращение от *application* — приложение). Далее перечислены вкладки-приложения, которые часто включаются в Fan-страницы Facebook.

- **Events** (Мероприятия). Отличная возможность объявить о предстоящем событии, будь то концерт в местном пабе или однодневная распродажа непарных носков.
- **Photos** (Фотографии). Пользователи Facebook надеются *увидеть* вас и фотографии этому способствуют. Фотографии Facebook неформальные и на них в центре внимания люди (например, компания может выгрузить фотографии своих сотрудников или общественное мероприятие). Вы самостоятельно решаете, одному заняться выгрузкой фотографий или разрешить вашим поклонникам принять участие в этом действе.
- **Notes** (Заметки). Заметка, на языке Facebook, — это короткое сообщение, похожее на те, что вы видите в блогах (см. *главу 13*). Но по-настоящему замечателен способ переноса заметок к вашим поклонникам и далее к другим заинтересовавшимся пользователям, похожий на перенос сообщений со стены, но более мощный. Facebook дает своим пользователям дополнительную возможность делиться заметками (по существу повторно публиковать их), как показано на рис. 12.16.
- **Discussion Boards** (Форумы). Сообщения на стене совершенно беспорядочны. А форумы группируют тематические сообщения и ответы на них в единую целостную нить. Вы можете начать собственное обсуждение (например, попросить совета или откликов) или разрешить другим пользователям Facebook продолжить его.
- **Reviews** (Отзывы). Если предоставить другим людям возможность высказаться о вас и вашей компании, это поможет завоевать их расположение и установить конструктивную обратную связь.
- **Static FBML** (статический FBML). Используйте язык FBML (Facebook Markup Language, Язык разметки Facebook) для создания собственных вкладок. В состав FBML входит уже знакомые вам стандарты HTML и CSS, поэтому вы можете копировать разметку непосредственно из программ Web-дизайна (например, Dreamweaver или Expression Web) на вкладку **Static FBML**. Разработчики Fan-страниц часто применяют это средство для создания искусно отформатированных страничек приветствия.

Наряду с перечисленными стандартными приложениями Facebook можно найти массу инструментов сторонних организаций, которые вы сможете внедрить на свои страницы. Введите в строку поиска "Facebook apps" и вы найдете программы, проводящие голосование, запускающие игры и т. д.

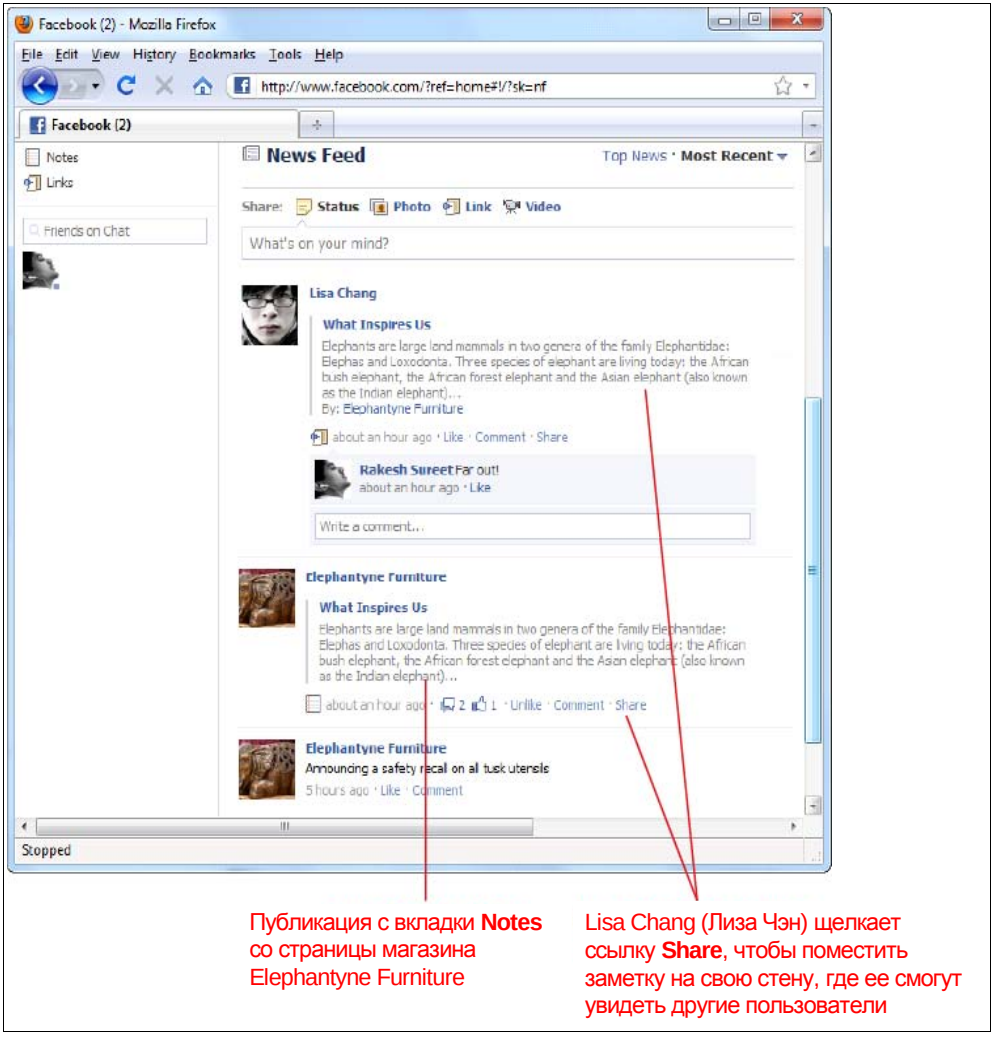


Рис. 12.16. Lisa Chang (Лиза Чэн) — поклонник. Когда она видит на стене сообщение от Elephantyne Furniture, она может щелкнуть мышью его ссылку **Like** или прокомментировать его. Но когда она видит заметку, у нее появляется дополнительная возможность: опубликовать ее на своей стене для совместного прочтения, щелкнув мышью ссылку **Share** (Поделиться). Теперь ее друзья могут видеть заметку и прокомментировать ее, усиливая таким образом влияние вашей Fan-страницы

Совет

В этот момент вы можете решить, что ваши посетители должны начинать знакомство с вашей страницей не со стены. Такое желание часто возникает, если вы добавляете приложение Static FBML и применяете его для создания страницы приветствия. Для

смены начальной вкладки, редактирования профиля щелкните мышью вкладку **Manage Permissions** (Управление правами доступа) и выберите нужную вкладку в списке **Default Landing Tab** (Страница по умолчанию для всех остальных).

Существует два способа вставки новой вкладки на вашу страницу. Лучше всего выполнить следующие действия.

1. Отредактируйте свой профиль (найдите вкладку **Edit Info** (Редактировать)).
2. Щелкните кнопкой мыши вкладку **Apps** (Приложения).

   Появится список приложений, которые у вас на самом деле есть, но не отображаются на вашей Fan-странице. Тут же есть раздел с приложениями, которые могут вам понравиться.
3. Щелкните кнопкой мыши ссылку **Edit Settings** (Изменить настройки) рядом с нужным вам приложением.

   На экране появится всплывающее окно.
4. Для вставки приложения на вашу Fan-страницу Facebook щелкните мышью кнопку **Add** (Добавить), а затем кнопку **OK**.

Этот алгоритм годится для большинства приложений, упомянутых ранее, но не для всех. (Например, здесь вы не найдете приложение Reviews.) Если вы выполнили все перечисленные действия и не нашли приложения, которое хотите вставить, перейдите к следующему способу.

1. Введите название нужного вам приложения в строку поиска в верхней части страницы Facebook.

   В ответ сайт выведет список всех элементов Facebook, соответствующих на данный момент критериям вашего поиска, включая приложения.
2. Когда искомое приложение появится в списке, щелкните его кнопкой мыши.

   Facebook перенесет вас на страницу этого приложения.
3. Щелкните кнопкой мыши ссылку **Add to my page** (Добавить на страницу).

   Вы увидите, что приложение появилось на левой боковой панели.

Предупреждаю, этот подход не годится для особых приложений Events (Мероприятия) и Notes (Заметки), т. к. они всегда прикреплены к вашей Fan-странице (несмотря на то, что они не всегда видны). Для включения этих приложений выполните первую последовательность действий.

Рекламирование вашей страницы Facebook на вашем Web-сайте

Теперь, когда вы собственноручно сделали замечательную Fan-страницу, пора использовать ее на вашем Web-сайте. У вас есть несколько вариантов рекламирования, включая следующие.

- **Facebook badge** (визитки Facebook). Это маленькое поле, рекламирующее вашу Fan-страницу Facebook на Web-странице. Когда кто-нибудь щелкает по нему мышью, он попадает с вашего Web-сайта на вашу Fan-страницу в Facebook. Для

создания визитки перейдите на страницу **www.facebook.com/badges** и щелкните мышью ссылку **Profile Badge** (Визитка профиля). При желании щелкните кнопкой мыши ссылку **Edit this badge** (Редактировать визитку) для настройки ее внешнего вида. Затем щелкните мышью кнопку **Other** (Другое) и получите HTML-код, который можно скопировать на любую Web-страницу.

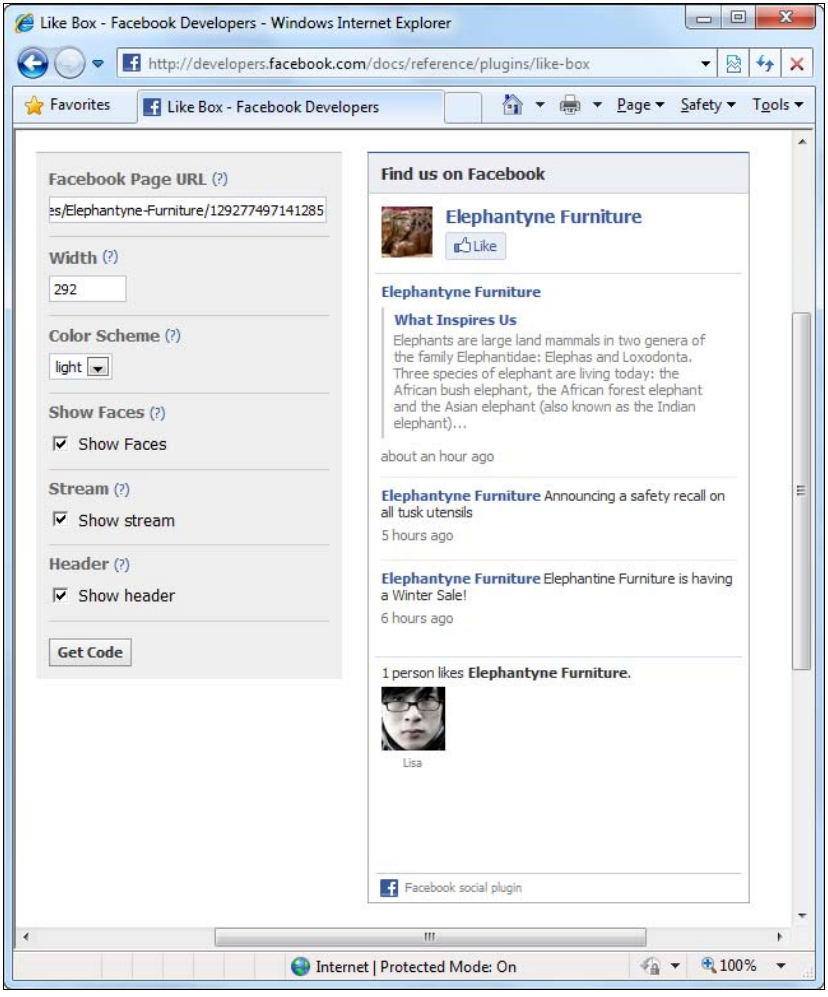


Рис. 12.17. Чтобы создать панель **Like**, заполните настройки слева. Внешний вид получившейся панели можно оценить справа. Панель **Like** показывает заголовок **Find us on Facebook** (настройка **Show header**), ваших поклонников (**Show Faces**), ваши последние новости (**Show stream**)

- **Like button** (кнопка Мне нравится). Это кнопка **Like** в стиле Facebook, которую можно поместить на любой странице. Если пользователь Facebook посетит вашу страницу и щелкнет мышью эту кнопку, произойдет то же самое, что и на реальной Fan-странице при щелчке мышью **Like**: устанавливается связь, повышающая ваш рейтинг и позволяющая информации перетекать с вашей страницы на вашу Fan-страницу в Facebook. Для получения кнопки **Like** (Мне нравится) пе-

рейдите по адресу **http://developers.facebook.com/docs/plugin** и щелкните мышью ссылку **Like button** (Кнопка Like). Введите URL вашей Fan-страницы в Facebook, настройте другие параметры, если хотите изменить внешний вид кнопки, и затем щелкните мышью кнопку **Get Code** (получить код).

- **Like box** (панель Мне нравится). Это панель, включающая сводку происходящего на вашей Fan-странице — например, последние сообщения на стене и заметки. На ней также есть важная всегда кнопка **Like** (Мне нравится). Для получения панели **Like** перейдите на страницу **http://developers.facebook.com/docs/plugin** и щелкните кнопкой мыши ссылку **Like box**. Введите URL вашей Fan-страницы в Facebook, настройте другие параметры, если хотите изменить внешний вид панели, и затем щелкните мышью кнопку **Get Code** (получить код) (рис. 12.17).

Обычно благодаря свойствам позиционирования CSS, о которых вы узнали в *главе 9*, панель **Like** или визитка Facebook размещаются на вашей Fan-странице в отдельной колонке.

Совет

Для того чтобы приобрести навыки работы в Facebook и отточить вашу стратегию продвижения, прочтите книгу "Facebook: The Missing Manual".

13

ГЛАВА 13

Блоги

Традиционный Web-сайт — золотой стандарт Web-пространства. Он бесконечно гибок — позволяет вести хронологические записи о личной поездке в крепость Мачу-Пикчу и поддерживать предприятие электронной коммерции.

Но природе Web-сайта не свойственна спонтанность. Представим, что вы хотите опубликовать кое-какие слухи о знаменитости, замеченной в вашем родном городке. Прежде чем делиться своими байками с остальным миром, придется выбрать имя файла с Web-страницей, решить, какую HTML-разметку использовать, определить, как связать свою страницу с другими страницами (и отредактировать их, если нужно) и т. д. Все эти задачи не очень трудны, но собранные вместе они способны отвратить случайных Web-авторов от написания чего-либо, что по их мнению не стоит этих хлопот.

И здесь блоги оказываются очень уместны. Это своеобразный самиздат, переносящий в сеть ваши мысли легко и быстро без заморочек, связанных с управлением Web-сайтом. Блоги — это новый понятный и слегка беспорядочный способ общения в Web. Ведение блога заключается в публикации коротких сообщений в любое время, когда возник порыв. Высокопроизводительное программное обеспечение для сопровождения блогов собирает, организует в хронологическом порядке и представляет записи вашего блога на Web-страницах. Это означает, что если вы не хотите чрезмерно заботиться о тонкостях управления Web-сайтом, в этом нет необходимости. Вам нужно только побеспокоиться об отправке сообщений для публикации с помощью программного обеспечения для ведения блогов, что так же просто, как отправка электронной почты.

В этой главе вы узнаете, как работают блоги, и увидите, как создать блог с помощью Blogger, одного из ведущих бесплатных Web-сервисов ведения блогов.

Общее представление о блогах

Слово "blog" — это сокращение от web log (Web-журнал), что неслучайно, т. к. блоги — это своего рода журналы, регулярные датированные записи, нечто среднее между дневниковыми записями и сообщениями в дискуссионных формах. В анг-

лийском языке слово "blog" может использоваться и как глагол, например, "I just ate at a terrible restaurant; when I get home I'm going to blog about it". (Я слегка перекусил в ужасном ресторане и, придя домой, собрался написать о нем в блоге). На рис. 13.1 представлена структура типового блога.

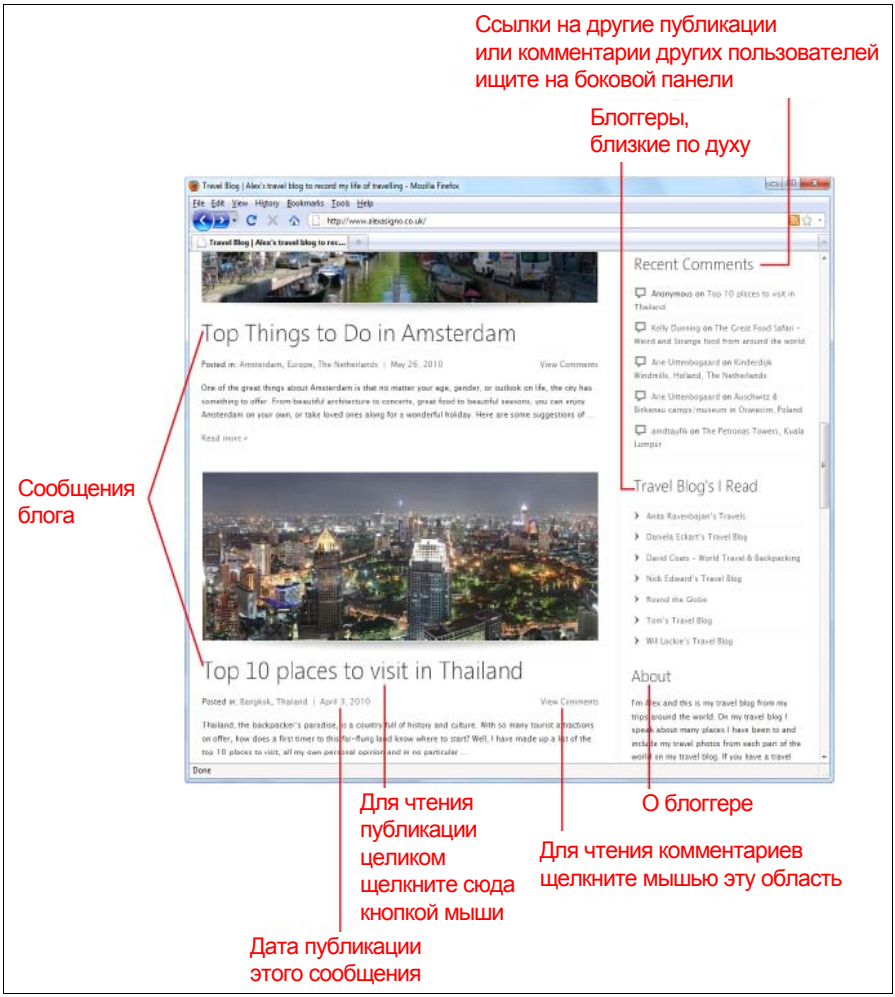


Рис. 13.1. В стандартном блоге публикации отображаются в обратном хронологическом порядке, самые свежие из них помещаются в верхней части страницы. На исходной странице блога может быть показано несколько последних записей или только одна последняя. Если записи блога очень длинные, на исходной странице может отображаться только несколько первых абзацев публикации со ссылкой на остальной текст (как в данном примере). Сбоку вы увидите вспомогательные разделы, календарь или список ссылок, позволяющих увидеть более старые записи, фотографию или статью об авторе или список рекомендуемых блогов (по-английски blogroll)

Несмотря на то что блоги упрощают Web-публикации, нельзя сказать, что они предоставляют упрощенный способ работы в Web. Блоги — это совершенно иная форма сетевых связей. И хотя нет строгого теста для определения, что является блогом, а что нет, у большинства блогов есть общие характеристики.

- Чаще всего блоги бывают персональными. Можно найти тематические, профориентированные, политические и массу других блогов, наполненных случайными нетривиальными размышлениями. Независимо от назначения блоги обычно подчеркивают авторскую точку зрения. Они редко стараются быть объективными — напротив, они выражают откровенно субъективные взгляды.
- Блоги организованы в хронологическом порядке. При проектировании Web-сайта вы тратите много времени на поиск наилучшего способа организации материала, часто применяя меню или ссылки для того, чтобы провести посетителей по всем тематическим разделам. В блогах совсем иной подход. Они не применяют ничего другого, кроме хронологического упорядочивания публикаций. Все остальное будет только отнимать лишнее время у нетерпеливых блоггеров (читателей и авторов других блогов).
- Блоги регулярно обновляются. Блоги реализуют быстрое непринужденное общение в отличие от скрупулезно разработанных Web-страниц. Блоггеры, как известно, вставляют контент, когда захочется, часто несколько раз в неделю. Поскольку записи в блогах датированы, сразу видно, поддерживаете ли вы актуальность блога. Если вы не готовы писать в блоге регулярно, не заводите блог, а создайте вместо него простую Web-страницу.
- Блоги очень гибки. Известно мудрое изречение о том, что нет размышлений слишком мелких для публикации в блоге. И это правда — публикуете вы подробное обсуждение полезности арахисового масла Oreos или краткий отчет о небогатом событиями дне, в обоих случаях публикация в блоге проходит без сбоев.
- Блоги обеспечивают более широкое общение. Блоги легче, чем обычные Web-сайты формируют сообщества. Не только потому что они более диалоговые по своей природе, но и потому что поддерживают комментарии и ссылки, способные привлечь к беседе разные блоги. Если кто-то публикует в блоге интересное сообщение, легион сотоварищей блоггеров включается в обсуждение с помощью ответов, публикаций в блогах, сообщений в Twitter и нажатий кнопки **Like** (Мне нравится) в Facebook. Скандальный слух из блога может облететь мир со скоростью ракеты.

Блоги занимают особую нишу, резко отличаясь от других виденных вами типов сайтов. Например, в блоге невозможно удачно продать пальто-тренчи для собак. Но многие люди создают блоги *в дополнение* к обычным Web-сайтам. Это отличное сочетание. Посетители любят блоги, потому что им хочется заглянуть за кулисы. Они наверняка будут возвращаться снова и снова, если рассчитывают на регулярно обновляемый блог, предлагающий устойчивый поток новостей, сплетен и откровений.

Примечание

Значительное количество крупных компаний поняли, что они могут стать более дружелюбными, более открытыми и более доступными, добавив блог с регулярными публикациями ведущих менеджеров, программистов, дизайнеров или даже исполнительного директора.

Популярные блоги

Текущий контент блога не постоянен, он может меняться от политических комментариев до личных путевых заметок. Легче всего составить представление о блогосфере, если познакомиться с несколькими популярными примерами.

Отличная отправная точка — Technorati (**http://technorati.com**), поисковая машина, индексирующая блоги. Перейдите туда и вы увидите списки самых популярных блогов и оперативных данных, показывающих популярность каких блогов растет (или падает).

Важно понять, что блоги — "длиннохвостая" технология, другими словами, существует ряд сверхпопулярных блогов, но во много-много раз больше блогов, охватывающих малого и среднего размера аудиторию. Для того чтобы осознать, что предлагают блоги, оставим пока крупнейших игроков с сайта Technorati (где преобладают медиа, новостные и политические блоги) и познакомимся с примерами помельче. Далее приведена случайная выборка только нескольких из них.

- **http://stefaniewildertaylor.com**. Родительские блоги — блоги, в которых родитель подробно описывает тяготы и заботы, связанные с воспитанием детей, — широко распространены. Они позволяют блоггерам обмениваться советами (и делиться болью) с огромным виртуальным сообществом родителей. Тематика очень разнообразна, начиная с обсуждения родителями серьезных медицинских проблем, возникших у их детей, и заканчивая заметками о милых забавных пустяках. Стефани Уайлдер-Тейлор (Stefanie Wilder-Taylor), актриса и писатель, привлекла внимание блогом, который житейски комичен и серьезен (среди других тем ее блог фиксирует проблемы матерей, борющихся с алкоголизмом).
- **www.rottengods.com**. С момента возникновения блоги предоставляли возможность людям, участвующим в военных действиях, переживающим голод и гражданскую войну, рассказать свои истории. В своем блоге Rotten Gods (Дрянные Боги) Фариборз Шамшири (Fariborz Shamshiri) публикует размышления о жизни и религии в Иране. Английские читатели слетались в его блог за самой свежей информацией об оспариваемых иранских выборах 2009 г.
- **www.schneier.com**. Как вы, вероятно, догадались, блоги — самое подходящее место для компьютерных фанатов всех мастей. Виртуоз компьютерной защиты Брюс Шнейер (Bruce Schneier) анализирует всякого рода проблемы безопасности, от копирования данных с кредитной карты до похищения голосов на выборах и террористических атак. Если вы ищете для знакомства менее искушенного компьютерного гения, посетите необычайно популярный блог Уила Уитона (Wil Wheaton) (**http://wilwheaton.typepad.com**), актера, сыгравшего компьютерного гения-выскочку Уэсли Крашера (Wesley Crusher) в фильме "Звездный путь" (Star Trek).
- **http://thesartorialist.blogspot.com**. Скотт Шуман (Scott Schuman) создал формат блога о моде во время отпуска в связи с рождением его дочери. На улицах Нью-Йорка он снимал прохожих в интересной одежде. Популярность его блога, содержащего авторские фотографии и нерегулярные комментарии, взлетела до небес. Приобретенная известность превратила его в фотографа, автора и популяр-

ную легенду в мире моды, что, конечно же, свидетельствует об огромном влиянии блогов.

Список можно продолжить, начиная с журналистов и заканчивая любителями, героями спорта, порнозвездами, кажется, что все захотели подвергнуть свою жизнь психоанализу или обсудить животрепещущие темы с многомиллионной аудиторией в блогах.

ДЛЯ ТЕХ, КТО ПОНИМАЕТ

Опасности, подстерегающие блоггера

Есть нечто располагающее к откровенности в природе блога и вынуждающее людей сообщать гораздо больше, чем следовало бы. Благодаря таким неосторожным публикациям возлюбленные узнали об обманах своей второй половины, бабушки прочли незабываемые отчеты о сексуальных победах своих внучек, а благонамеренные сотрудники потеряли свою работу.

Опасности импульсивных высказываний в блогах особенно велики в трудовой сфере. В большинстве стран компании имеют обыкновение увольнять сотрудников, сделавших заявления, наносящие ущерб бизнесу (даже если они справедливы). Даже известная своими широкими взглядами компания Google бросила на произвол судьбы Марка Джена (Mark Jen) (**http://roseandsnail.com**) после того, как он написал в блоге несколько резких слов о собрании отдела продаж Google, напоминавшем по его словам студенческую попойку. Важно подчеркнуть, что он не хотел навредить Google или разрекламировать свой блог. В действительности о существовании блога знали только его близкие друзья и члены семьи. К сожалению, несколько сайтов, следящих за Google, обратили внимание на публикацию и распространили ссылку на нее в Интернете.

Историй, похожих на эту, великое множество. Сотрудники теряют работу после публикации коммерческих тайн, свидетельств недостойного поведения на работе (например, публикации пикантных фотографий на рабочем месте или демонстрации напрасной траты времени за раскладыванием пасьянса на рабочем компьютере) или просто жалоб на шефа.

Для того чтобы при ведении блога защитить себя от опасностей, помните следующие правила.

1. Пользователей "Anonymous" (Аноним) не бывает.
2. Если вы решили скрывать свое имя, подберите псевдоним или спрячьте личную информацию, помните о правиле 1.
3. Смешное субъективно. Ваши юмористические истории, связанные с работой, будут восприняты в ином свете прочитавшим их руководством, которое лишено вашего тонкого чувства юмора.
4. Сначала подумайте, потом пишите. Между секретами компании и информацией в открытом источнике тонкая грань.
5. Обратной дорого нет. Хотя множество инструментов для блогов позволяют редактировать и удалять старые публикации, исходные версии могут сохраниться в кэшах поисковых машин навеки (см. врезку в *разд. "Как спрятаться от поисковых машин" главы 11*).

Агрегирование

Одно из замечательных средств блогов — *агрегирование* или *синдикация*, позволяющая заядлым читателям блогов просматривать их любимые блоги с помощью

программы, называемой программой чтения RSS-лент или агрегатором новостных лент. Для применения программы чтения RSS-лент вы вводите ссылки на все свои любимые блоги, а затем следите за обновлениями в них. Программа чтения RSS-лент периодически проверяет эти блоги и уведомляет вас о новых публикациях, избавляя от необходимости проверять каждый блог 94 раза в сутки для просмотра свежего контента. Если вы читаете блоги регулярно, такие программы — самый удобный способ оставаться на связи со всеми вашими друзьями из блогосферы.

Совет

Программы чтения RSS-лент немного похожи на программы электронной почты, регулярно проверяющие наличие новых писем от ваших друзей. Это гораздо эффективнее, чем связываться с каждым из них и интересоваться, не хотят ли они сообщить что-нибудь новенькое. Аналогичным образом программы чтения RSS-лент можно применять для отслеживания активности блогов ваших друзей. Если ничего нового нет, вы узнаете об этом немедленно.

Несмотря на то что большинство блогов работают с программами чтения RSS-лент, некоторые этого не делают. Для работы с этой программой блоги должны предоставить ленту (рис. 13.2), последние сообщения из блогов в удобном для компьютеров формате. Программы чтения RSS-лент анализируют ленты и извлекают из них важную информацию, такую как заголовки сообщений, описание, дату и текст. Эту информацию они выводят на экран для чтения, не заставляя вас отправляться для этого на Web-сайт.



Рис. 13.2. У большинства блогов где-нибудь на исходной странице есть ссылка на ленту. Ищите слово "feed" (лента или канал) , "syndication" (агрегирование), "subscribe" (подписаться) или "RSS" (Really Simple Syndication, Действительно простая синдикация) (формат, который используют ленты). Как правило, ссылка обозначается оранжевой пиктограммой радара



Рис. 13.3. *Вверху*: когда Firefox находит на текущей странице ссылку на RSS-ленту, он отображает в адресной строке особую оранжевую пиктограмму. Щелкните ее кнопкой мыши и выберите один из вариантов подписки, чтобы добавить живую закладку.
В центре: Firefox спрашивает, как вы хотите хранить вашу закладку. Можно применить инструмент Firefox Live Bookmarks (Закладки онлайн) или отдельный сервис чтения лент, например Google Reader (просто выберите его из списка). Если не хотите, чтобы эта страница выводилась снова, установите флажок **Always use ...** (Всегда применять ...). Далее щелкните мышью кнопку **Subscribe Now** (Подписаться).
Внизу: инструмент Live Bookmarks (Закладки онлайн) предлагает подменю текущих сообщений из блогов, которые Firefox обновляет автоматически. Вам все равно придется проверить закладку, чтобы увидеть, есть ли новое сообщение (что требует дополнительных усилий по сравнению с программой чтения RSS-лент, например, FeedDemon), но зато исходный сайт посещать не придется

Если хотите опробовать программу чтения RSS-лент, у вас есть множество вариантов.

- Сетевые программы чтения требуют, чтобы вы зарегистрировались и создали бесплатную учетную запись, при этом на свой компьютер не надо ничего устанавливать. Вы просто переходите на Web-сайт программы чтения RSS-лент и читаете ленты прямо в окне вашего браузера. Популярные примеры — Google Reader (**www.google.com/reader**) и NewsGator (**www.newsgator.com**).
- Настольные программы чтения RSS-лент устанавливаются на ваш компьютер. Вы можете проверить и загрузить многие из них с популярных сайтов условно-бесплатного программного обеспечения, например, **www.download.com**. Если вы пользуетесь компьютером с ОС Windows, можно загрузить отличную программу FeedDemon с сайта **www.feeddemon.com**. Любителям Mac может понравиться широко разрекламированная NetNewsWire с сайта **http://netnewswireapp.com**. Обе программы бесплатны.
- Web-браузеры все чаще включают в состав такие инструменты, как чтение RSS-лент. Если вы пользуетесь Firefox, его инструмент Live Bookmarks (Закладки онлайн) отслеживает ленты (рис. 13.3). У Internet Explorer 7 (и в более поздних версиях) и Safari также есть встроенные программы чтения. (Дополнительную информацию о программе чтения RSS-лент в IE см. на странице **www.microsoft.com/windows/rss**. О Safari читайте обзор на странице **www.apple.com/safari/features.html**.) Но ни один браузер не предлагает инструмент для чтения лент, столь же функциональный и удобный, как специализированные программы, например, FeedDemon.

> ***Совет***
>
> Большинство блогов разрешают подписку по электронной почте, т. е. вы получаете новые сообщения в вашу папку входящих писем. Если вы регулярно читаете несколько блогов, и они не очень загружены, подписки по электронной почте помогут избежать трудностей, связанных с установкой программы чтения RSS-лент.

Хостинг блога и программное обеспечение

Прежде чем создавать собственный блог, было бы неплохо понять, какие есть возможности создания блога. На самом деле существует два вида блогов.

- Блоги на отдельном или самостоятельном хостинге. Если вы энтузиаст с высоким профессиональным уровнем и амбициями, вас, возможно, заинтересует самостоятельный хостинг для блога. Для этого вам нужно выбрать подходящее программное обеспечение, найти Web-хост, поддерживающий его, и затем настроить все. Такой подход обеспечит неограниченную гибкость (и, возможно, более высокую производительность). Если вам нравится решать проблемы, возникающие при самостоятельной работе, например организации переговоров вашего ноутбука и кофеварки, блог с самостоятельным хостингом — ваш выбор.

  Примерами программного обеспечения для ведения блога могут служить программы WordPress (**http://wordpress.org**) и Movable Type (**www.movabletype.org**).

- Блоги на базе сервиса. В этом случае вы просто регистрируетесь у провайдера блогов и начинаете вести блог немного в стороне. Вставка нового блога столь же проста, как заполнение формы в вашем браузере. Вам никогда не придется прибегать к отдельной программе или выяснять, как выгрузить содержимое файлов, т. к. провайдер блогов сохранит для вас все ваши файлы. Вам даже необязательно иметь Web-сайт. Блоги на базе сервиса — лучший выбор для новичков, поскольку при этом они не причиняют себе вреда и получают гибкое управление.

  К провайдерам блогов на основе сервиса относятся Blogger (**www.blogger.com**), Word-Press (**http://wordpress.com**, но не **http://wordpress.org**, где вы найдете бесплатную блог-платформу) и TypePad (**www.typepad.com**).

В этой главе вы будете тратить время на одно средство ведения блогов, называемое Blogger. Сервис Blogger чрезвычайно популярен, входит в 50 самых посещаемых Web-сайтов (**http://mostpopularwebsites.net/1-50**). Преимущества Blogger в его простоте наряду с многофункциональностью, включая поддержку групп в блогах, имен персональных доменов, настраиваемых шаблонов и множества дополнительных инструментов. Эти функции делают Blogger лучшим кандидатом для бесспорного чемпионства в ведении блогов.

Примечание

Ряд инструментов создания Web-сайтов, названных *системами управления контентом* (Content Management System, CMS), позволяют создавать обычные блоги и Web-сайты других типов. Два популярных примера — Drupal (**http://drupal.org**) и Joomla (**www.joomla.org**). Как правило, эти продукты подходят для деловых людей, нуждающихся в установке сложных Web-приложений — скажем, просторного торгового предприятия электронной коммерции или Web-журнала с командой распространителей — без необходимости создания всех компонентов с нуля.

СЕМИНАР ДЛЯ ОПЫТНЫХ ПОЛЬЗОВАТЕЛЕЙ

Почему вы можете отказаться от Blogger

Некоторые несгибаемые фанатики ведения блогов предпочитают блоги с самостоятельным хостингом, которые предлагают несколько уникальных преимуществ, отсутствующих у сервиса Blogger. Далее приведено несколько примеров.

- *Цензура.* Если вы пользуетесь сервисом ведения блогов, например, Blogger, Google контролирует ваш контент. Если другие пользователи сообщат, что в вашем блоге есть оскорбительный контент или порнография, Google может заставить будущих читателей щелкать мышью предупреждающую страницу до того, как они доберутся до ваших публикаций. Если же Google решит, что вы нарушаете законы об авторском праве или поощряете преступную деятельность, они могут удалить ваш блог без предупреждения. Более того, некоторые помешанные на цензуре страны, как Китай, блокируют сайты сервиса Blogger, что означает невозможность видеть ваш блог ни в одной из этих стран (если вы не заплатите за имя личного домена, как описано в *разд. "Присвоение вашему блогу имени пользовательского домена" данной главы*).
- *Связанные Web-сайты.* Благодаря ссылочной мощи языка HTML достаточно легко отправить читателей с вашего Web-сайта в ваш блог и обратно. Им даже не нужно знать, что меняются Web-серверы. Но если вы размещаете и Web-сайт, и блог на одном и том же сервере, у вас появляются уникальные преимущества. И сайт, и блог могут использовать одно и то же имя домена, совместно использовать кон-

тент (например, фотографии) и помогать друг другу в ранжировании, выполняемом поисковыми машинами, т. к. Google рассматривает их как части одного сайта.

- *Полный контроль.* У некоторых блогов много авторов и огромный трафик. Их сфера влияния и популярность конкурируют с традиционными газетами. (См. Huffington Post, на сайте **www.thehuffingtonpost.com** как один из самых популярных примеров.) Создателям сайта такого сорта необходим полный контроль над любой мелкой деталью, чтобы вручную регулировать кампании поисковых машин и рекламные стратегии. Ясно, что на этом уровне бесплатный сервис блогов действовать не сможет.

Но за эти преимущества приходится платить. Блоги с самостоятельным хостингом сложнее установить и ими сложнее управлять, и кроме того, вам придется платить ежемесячную плату за хостинг. Но если вам все еще интересно, начните с WordPress (**http://wordpress.com**), бесспорного фаворита у профи. Если же вы еще не уверены, не бойтесь использовать Blogger или любой другой сервис для ведения блогов. Ваш выбор не повлияет на успех, т. к. существуют чрезвычайно популярные блоги на всех блог-платформах, упомянутых в этой главе.

Начало работы с Blogger

Blogger — один из самых широко применяемых сервисов ведения блогов. Он позволяет легко начать вести блог и полон отличных элементов управления. Когда-то первоклассным средствам Blogger требовался небольшой ежегодный взнос. Но все изменилось, когда Blogger купила компания Google. Теперь все средства Blogger — это часть одного бесплатного пакета.

Организация блога в Blogger до смешного легка. В следующих разделах вы узнаете, как создать блог, добавить в него сообщения и распорядиться некоторыми замечательными инструментами.

Совет

Вы также можете проверить официальный каталог файлов помощи Blogger (**http://help.blogger.com**) и дискуссионные форумы (**www.bloggerforum.com**), на которых блоггеры делятся советами, задают вопросы и пишут о своих печалях.

Создание блога

Перед созданием блога неплохо было бы определить ваши цели и точно решить, какого типа контент вы намерены представить. Хотя вы можете создать блог со случайными мыслями или хроникой повседневной жизни, у самых успешных блогов есть свой голос и четкая цель. Они привлекают постоянную аудиторию целенаправленными тематически выверенными сообщениями.

После того как вы определили, каким хотите видеть свой блог, можете выбрать привлекательное имя и подходящий URL. Начнем со следующих шагов[1].

1. Перейдите на сайт **www.blogger.com**.

   Это исходная страница сервиса Blogger.

[1] На момент выхода русского перевода книги интерфейс Blogger изменился, но общие принципы создания блога остались прежними. — *Прим. ред.*

2. Если у вас есть учетная запись Google, введите регистрационные данные и щелкните мышью кнопку **Sign In** (Войти), если нет — кнопку **Get started** (Начать).

   Прежде чем создавать блог, нужно завести учетную запись Google (рис. 13.4). Если вы пользуетесь другими сервисами Google (например, Gmail, Google AdSense или Google Analytics), для создания блога можно воспользоваться уже имеющейся учетной записью.

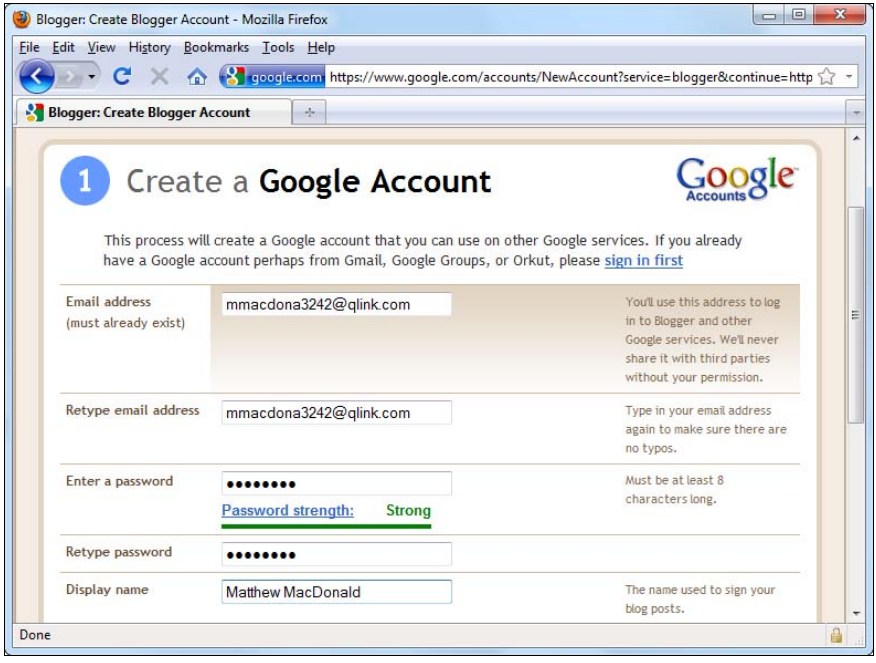


Рис. 13.4. Первый шаг в создании блога на сервисе Blogger — регистрация (или создание) учетной записи Google

Примечание

Учетную запись нужно создать только один раз. Но с одной учетной записью можно создать несколько блогов.

3. Введите данные вашей учетной записи.

   Если у вас уже есть учетная запись Google, введите отображаемое название (которое Blogger будет использовать для подписи ваших сообщений в блоге) и пароль. Если у вас еще нет учетной записи, необходимо ввести действующий адрес вашей электронной почты, пароль, который хотите использовать, и отображаемое название (имя).

   В любом случае для продолжения вам придется ввести символьную строку, чтобы доказать, что вы не компьютерная программа, и установить флажок в нижней части страницы для официального принятия правил сервиса Blogger.

Примечание

В процессе создания блога Google может попросить вас подтвердить вашу личность. Для этого он предложит ввести код подтверждения или позвонить вам. Пройти этот тест легко. Сначала сообщите Google номер вашего телефона. Затем, когда через несколько секунд Google позвонит вам, введите код подтверждения на странице регистрации (Blogger будет терпеливо ждать) и щелкните мышью кнопку **Verify** (Проверить).

4. Щелкните мышью кнопку **Continue** (Продолжить), чтобы перейти на **Dashboard** (Панель инструментов) сервиса Blogger.

   На **Dashboard** (Панель инструментов) перечисляются все ваши блоги и есть возможность управлять ими.

5. Щелкните мышью кнопку **Create Your Blog Now** (Создать блог), чтобы перейти к следующему этапу создания блога.

   Blogger отобразит новую страницу (рис. 13.5).

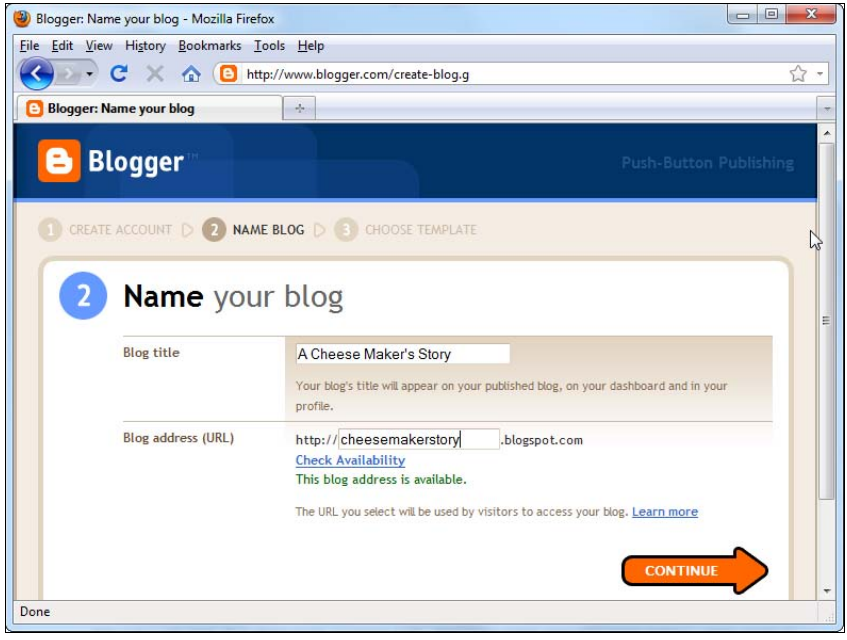


Рис. 13.5. На втором шаге вы создаете важную часть — блог, как таковой. Для того чтобы убедиться, что вы точно на месте, щелкните мышью кнопку **Check Availability** (Проверить доступность), чтобы проверить доступность URL вашего блога. (Если он не доступен, нужно попробовать еще раз.) К счастью, найти удачный Blogger URL (который всегда заканчивается **.blogspot.com**) гораздо легче, чем хороший URL для Web-сайта

6. Введите название и URL вашего блога.

   Название блога — то же, что название Web-страницы. Это информативный фрагмент текста, появляющийся в строке заголовка окна браузера.

   URL — это действительно важный элемент, поскольку вы не захотите менять его позже и рисковать потерей постоянных читателей. Это адрес, который заинтересовавшийся пользователь применяет для поиска вашего блога. Blogger уди-

вительно щедр на URL-адреса. В отличие от провайдеров бесплатного Web-хостинга Blogger позволяет использовать почти любой URL, если он заканчивается добавкой **.blogspot.com**. Несмотря на то, что другие блоггеры уже расхватали самые удачные имена, до сих пор еще довольно легко создать короткое и привлекательное имя, например, **http://secretideas.blogspot.com** или **http://richwildman.blogspot.com**.

Если хотите использовать имя пользовательского домена для вашего блога, вам придется добавить его *после* создания блога с обычным адресом **.blogspot**. Как это сделать, поясняется в *разд. "Присвоение вашему блогу имени пользовательского домена" данной главы*.

7. Щелкните мышью кнопку **Continue** (Продолжить) и выберите шаблон для вашего блога.

   Сервис Blogger для начала предлагает вам несколько шаблонов (рис. 13.6). Но после создания блога вы безбоязненно сможете изменять форматирование шаблона или даже перейти на совершенно другой шаблон (см. *разд. "Настройка вашего шаблона" данной главы*).

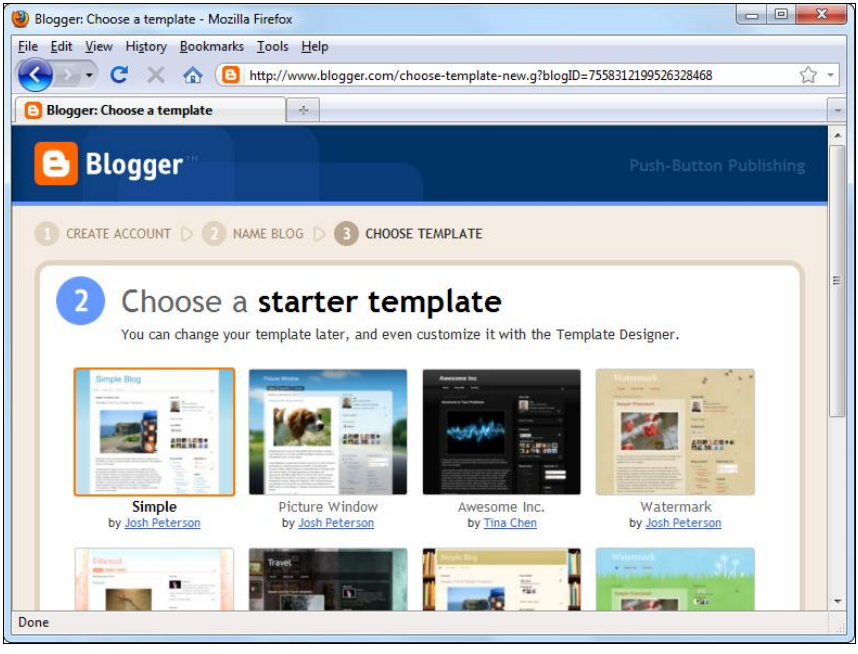


Рис. 13.6. Когда создается блог, вы выбираете готовый шаблон, и Blogger форматирует ваши сообщения, учитывая цвет шаблона, графику и макет. Если позже вы измените ваш шаблон, Blogger перенастроит ваши сообщения, чтобы они соответствовали новому визуальному стилю

8. Выберите понравившийся шаблон и щелкните мышью кнопку **Continue** (Продолжить), чтобы завершить создание блога.

   Blogger отобразит приветственное сообщение о создании фирменного нового блога.

9. Для создания первого сообщения в блоге щелкните мышью кнопку **Start Blogging** (Написать первое сообщение в блог).

   К управлению блогом вы сможете вернуться в любое время, перейдя на сайт **www.blogger.com**. Сейчас продолжим и создадим ваше первое сообщение в блоге.

10. Введите название вашего сообщения и затем наберите текст сообщения в большом текстовом поле, действующем как миниатюрный текстовый процессор (рис. 13.7).

    Пока не отвлекайтесь на всякие затейливые штучки в окне редактирования, вы узнаете о них в следующем разделе.

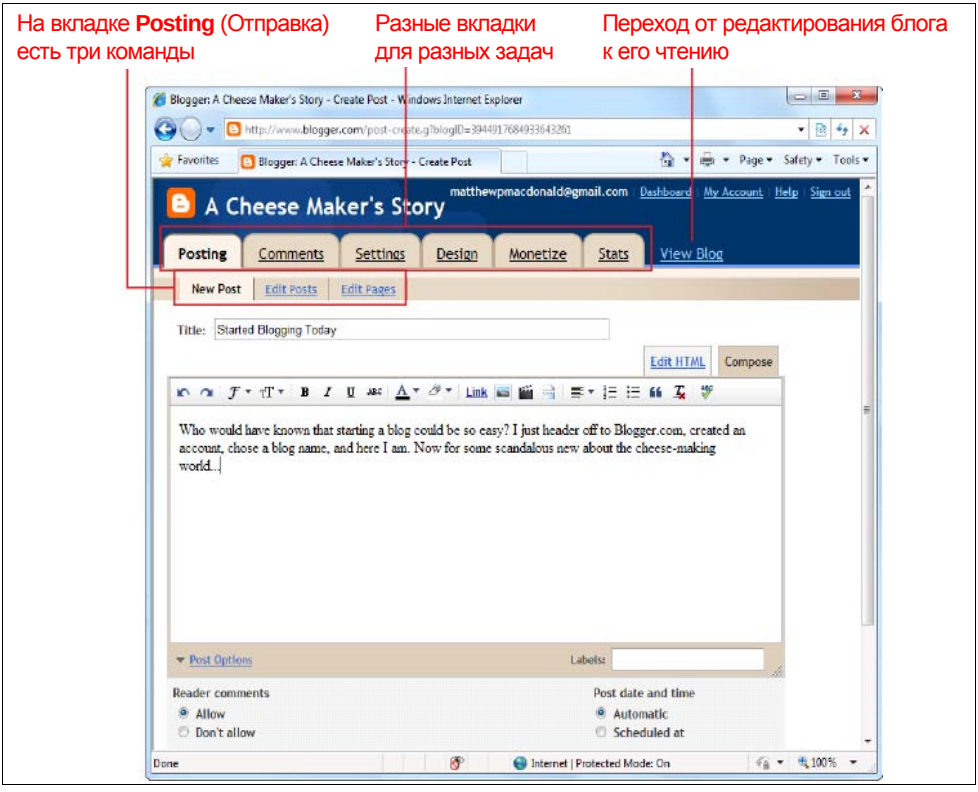


Рис. 13.7. Blogger применяет макет страницы с вкладками, предназначенный для решения нескольких задач. Когда создается сообщение, вы пользуетесь вкладкой **Posting** (Отправка) с тремя кнопками: одна для создания нового сообщения, вторая для редактирования уже имеющегося и третья для редактирования обычных Web-страниц. (Ваш блог может включать до 10 обычных HTML-страниц, на которые можно ссылаться в ваших сообщениях в блоге. Например, можно добавить страницу **About Me** (Обо мне).)

Примечание

Длина сообщения в блоге зависит только от вашего желания. Одни люди помещают в блоги длинные истории, а другие пишут одно-два предложения, содержащие только ссылку на интересную новость (или, что гораздо чаще, на сообщение другого блоггера).

11. Щелкните мышью ссылку **Post Options** (Настройки сообщения) в нижней части страницы, чтобы просмотреть несколько дополнительных параметров.

    Вы можете задать категорию для вашего сообщения, указав ключевое слово (или список ключевых слов, разделенных запятыми) в поле **Labels for this post** (Ярлыки). Когда пользователи запускают поиск в вашем блоге, они могут искать сообщения с конкретными ключевыми словами. Для получения лучших результатов всегда применяйте одни и те же ключевые слова для одинаковых тем. Например, каждый раз, когда вы пишите о домашних свинках, вставляйте ярлык "поросенок".

    Вы можете разрешить пользователям комментировать ваши сообщения (выберите **Allow** (Разрешить), настройку по умолчанию) или запретить делать это (выберите вариант **Don't allow** (Запретить)).

    Можно задать дату и время публикации сообщения. Выберите запланированную дату, и Blogger подождет ее наступления, а затем поместит ваше сообщение в блог.

12. Для создания публикации в блоге щелкните кнопку **Publish Post** (Опубликовать сообщение).



Рис. 13.8. В блоге показаны два последних сообщения (в обратном хронологическом порядке, так что самое последнее сообщение показано первым). На правой боковой панели расположены разделы с информацией об авторе и других Web-сайтах, представляющих интерес (пока ни одного такого не указано), а также ссылки на самые последние публикации в них

Blogger отображает подтверждение, информирующее о публикации нового сообщения.

Если вместо немедленной публикации вы хотите подождать какое-то время и подумать над вашим сообщением, щелкните мышью **Save Now** (Сохранить). В этом случае Blogger сохраняет только что введенный текст и хранит его, дожидаясь вашего следующего возвращения к блогу. (На рис. 13.11 поясняется, как можно найти неопубликованное сообщение и отредактировать его.)

Совет

Blogger сохраняет черновик вашего сообщения во время его ввода просто на случай проблем с интернет-подключением (или случайного закрытия окна браузера). Но он выполняет автосохранение только каждые несколько минут. Щелчок кнопки **Save Now** (Сохранить) сохраняет текущее сообщение немедленно.

Теперь очень подходящий момент взглянуть на то, как выглядит ваша публикация. Щелкните кнопкой мыши ссылку **View Blog** (Просмотреть блог) или введите URL вашего блога в окне браузера вручную. На рис. 13.8 показан пример того, как выглядит обычный блог.

Создание форматированных сообщений

До сих пор вы видели, как помещать текстовые сообщения в блог. Но Blogger проявляет удивительную гибкость, когда приходит время настраивать ваш блог. Вы можете применять все виды искусных средств дизайна, начиная с выделения текста и заканчивая вставкой графики. Самое замечательное то, что Blogger дает возможность хозяйничать в HTML-разметке. Вам просто нужно ориентироваться в редакторе сервиса Blogger.

Для выполнения некоторого форматирования начните создавать новое сообщение, щелкнув мышью вкладку **Posting** (Отправка). Введите какой-нибудь текст в поле **Compose** (Создать), расположенном в середине страницы. Далее выделите фрагмент текста и опробуйте некоторые кнопки на панели инструментов, находящейся над текстовым полем (рис. 13.9). За кулисами Blogger применяет встроенные стили (см. *разд. "Три вида стилей" главы 6*) для форматирования вашего сообщения.

Этот редактор, называемый *визуальным компоновщиком* (visual composer), разработан по образу и подобию текстового процессора. Но если очень нужно какое-либо HTML-воздействие, щелкните мышью кнопку **Edit HTML** (Изменить HTML) в верхнем правом углу окна редактирования. Теперь можно непосредственно вставлять элементы и другие HTML-примочки. Для возврата в режим WYSIWYG щелкните мышью кнопку **Compose** (Создать).

Помните, что так выполняется единичное форматирование. Если вы хотите согласованно изменить внешний вид часто встречающихся элементов в ваших сообщениях (например, названия сообщений, текст сообщений, подпись автора и т. д.), необходимо изменять шаблон вашего блога (см. *разд. "Настройка вашего шаблона" данной главы*). Такой подход сродни применению таблицы стилей на большом Web-сайте: редактируя свой шаблон, вы создаете набор правил форматирования, настраивающих внешний вид вашего блога.

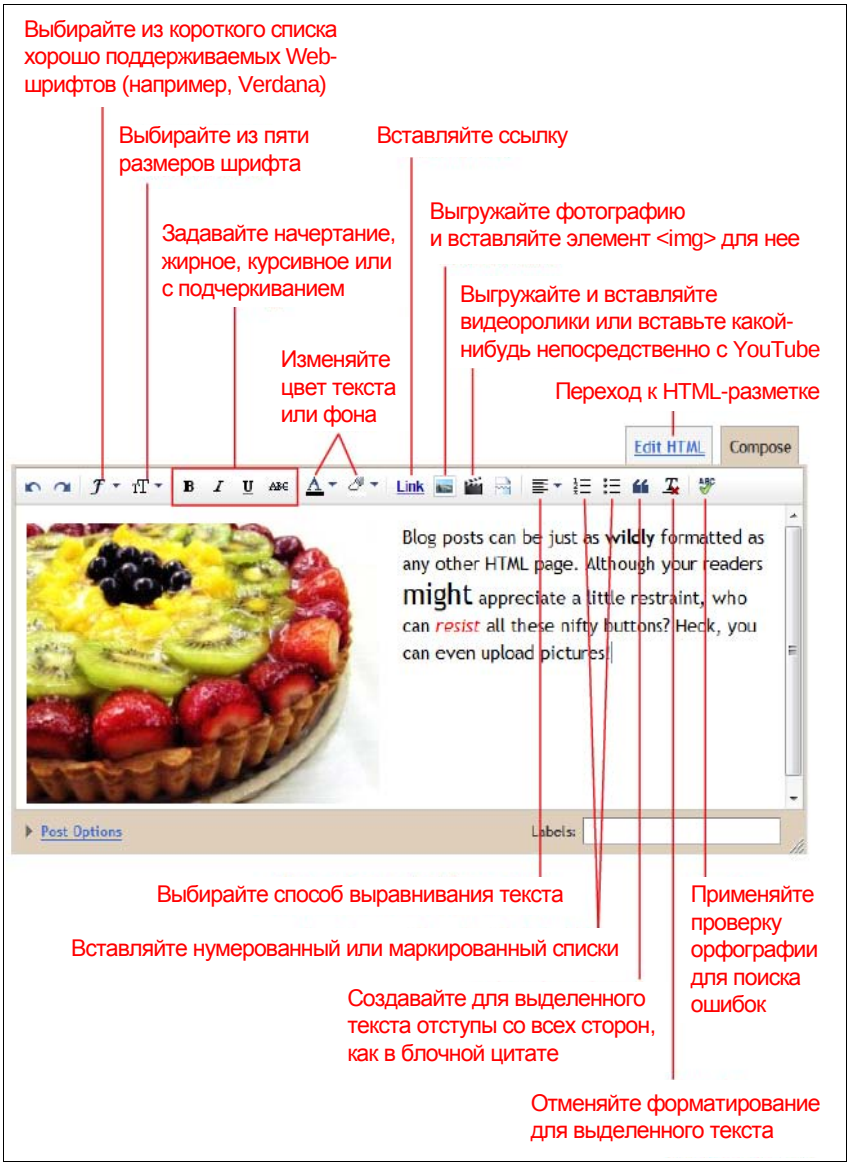


Рис. 13.9. Кнопки на панели инструментов в режиме **Compose** (Создать) предоставляют несколько базовых вариантов форматирования. Можно изменить шрифт сообщения, размер шрифта, добавить жирное или курсивное начертание, создать простые списки и вставить картинки

СЕКРЕТЫ РЕМЕСЛА

Отправка в блог сообщения по электронной почте

Для использования в полной мере всех средств сервиса Blogger имеет смысл применять редактор на Web-сайте Blogger. Но если вы заняты и свободного времени в обрез, может пригодиться способность Blogger превращать сообщения электронной почты в публикации в блоге. Это становится действительно полезным, если нужно переслать сообщение с мобильного телефона или если у вас временное интернет-под-

ключение. В последнем случае вы можете подготовить публикацию в программе электронной почты и затем подключиться к Интернету только на время отправки сообщения.

Для применения инструментов публикации с помощью электронной почты в сервисе Blogger вам сначала нужно включить их. Щелкните кнопкой мыши вкладку **Settings** (Настройки), затем выберите ссылку **Email & Mobile** (Электронная почта и мобильные устройства) и перейдите с помощью прокрутки к разделу **Posting Options** (Параметры отправки сообщений). В поле адреса **Email Posting Address** (Адрес отправки электронной почты) введите секретное слово. Blogger встраивает это слово в адрес электронной почты, предоставляемый для удаленной публикации. Это помешает другим пользователям подсунуть их сообщения в ваш блог, поскольку они не будут знать точного адреса. Например, если ваше имя пользователя — lisajones и ваше секретное слово — antelope12, для вставки публикации в ваш блог вы должны отправить сообщение на адрес **lisajones.antelope12@blogger.com**.

Ниже выберите переключатель **Publish emails immediately** (Публиковать письма сразу) (Blogger опубликует сообщения, как только получит их) или **Save emails as draft posts** (Сохранять письма как черновики сообщений) (Blogger преобразует присланные по электронной почте письма в черновики, и вам придется зарегистрироваться и просмотреть их до того, как они обретут жизнь). Если же позже вы решите отключить публикацию по электронной почте, выберите переключатель **Disabled** (Отключено). Затем, чтобы подтвердить ваши изменения, щелкните мышью кнопку **Save Settings** (Сохранить настройки) в нижней части страницы.

Когда вы посылаете по электронной почте сообщение в Blogger, строка темы письма немедленно превращается в название публикации, а его текст становится телом сообщения. И самое замечательное то, что вы можете воспользоваться форматированием вашей программы электронной почты, т. к. Blogger понимает все стандартные HTML-теги. И наконец, если вы используете учетную запись электронной почты, которая добавляет некоторый текст в конец письма (например, отказ от ответственности за соблюдение конфиденциальности), вы можете помешать его появлению в тексте сообщения в блоге. Просто поместите в конец письма текст `#end`, и Blogger проигнорирует весь последующий текст.

Управление блогом

После создания блога вы можете выполнять любую из следующих задач.

- ❒ Добавлять новые сообщения.
- ❒ Редактировать существующие публикации.
- ❒ Просматривать комментарии, оставленные другими людьми.
- ❒ Изменять настройки вашего блога.

Центр управления блогом — страница, именуемая **Dashboard** (Панель инструментов) (рис. 13.10). Именно на ней вы оказываетесь после каждой регистрации. На панели инструментов перечислены все ваши блоги и представлен набор управляющих ссылок для каждого из них. Если прокрутить страницу до конца, можно увидеть раздел **Reading List** (Список чтения), в который включены ссылки на новости блогосферы, злободневные блоги и все блоги, на которые вы решили подписаться.

Хотя отправная точка для управления блогом — **Dashboard** (Панель инструментов), как только вы начнете работать, вы вернетесь к странице Blogger с вкладками,

которая использовалась, когда вы создавали свое первое сообщение. Эта страница — основной компонент сервиса Blogger; в зависимости от выбранной вкладки вы можете выполнять самые разные задачи, начиная с добавления и редактирования сообщений и заканчивая изменением способа функционирования блога.

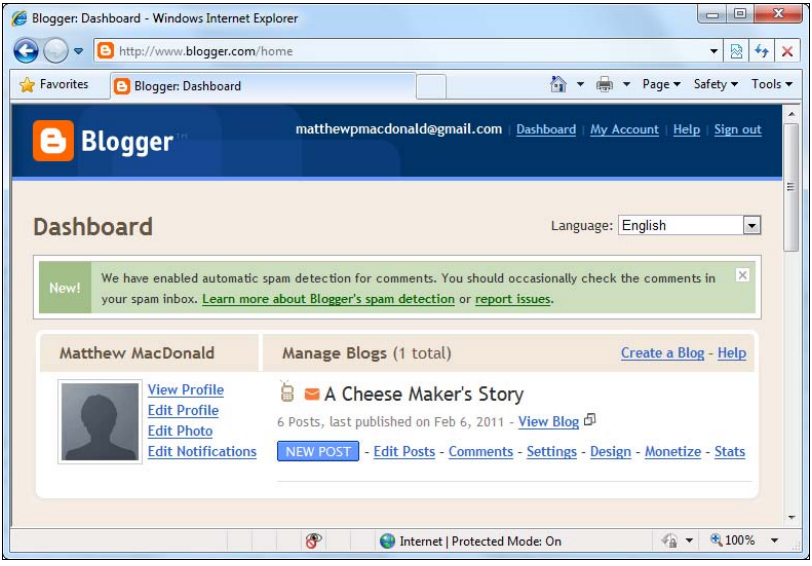


Рис. 13.10. Когда вы регистрируетесь в Blogger, то попадаете на страницу, называемую **Dashboard** (Панель инструментов) и отображающую список всех ваших блогов и количество публикаций в них. Для добавления публикации щелкните мышью кнопку **New Post** (Новое сообщение) (см. рис. 13.7), а для редактирования и пересмотра более ранних публикаций — кнопку **Edit Posts** (Изменить сообщения) (см. рис. 13.11)

Например, если щелкнуть мышью кнопку **Edit Posts** (Изменить сообщения), вы окажетесь снова на странице управления Blogger, но выбрана будет вкладка **Edit Posts** (рис. 13.11). Данная вкладка позволяет пересматривать и редактировать уже опубликованные сообщения. Это прекрасное средство для исправления грубой оплошности — если вы будете достаточно расторопны и выловите ее прежде, чем это сделает кто-то другой.

> ***Примечание***
>
> Если вы еще не опубликовали ваше сообщение, вы увидите его с пометкой **Draft** (Черновик), расположенной рядом с ним. Это удобное свойство позволяет прерваться и прочесть ваше сообщение на следующий день (или на следующей неделе) прямо там, где вы его оставили.

В страницу управления упаковано множество средств и параметров. Далее приведен краткий обзор ее многочисленных вкладок.

- **Posting** (Отправка). Используйте эту вкладку для создания нового сообщения, редактирования существующего или создания нескольких разнообразных HTML-страниц (не более 10), которые могут сосуществовать с вашим блогом на одном и том же сайте.

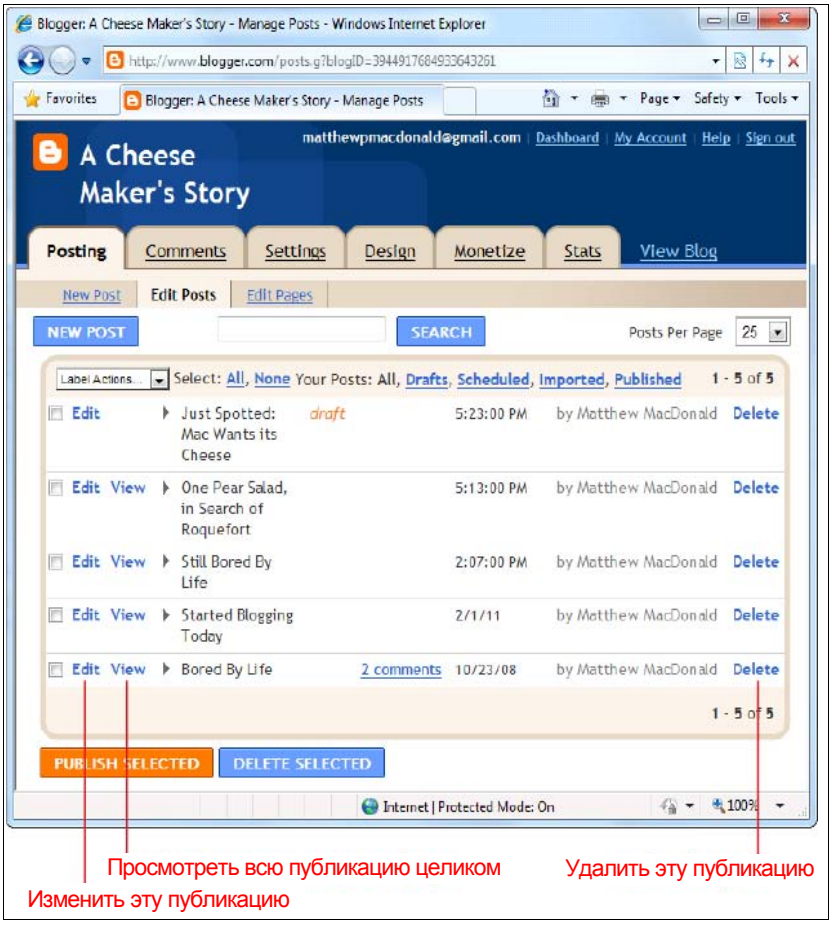


Рис. 13.11. Страница **Edit Posts** (Изменить сообщения) позволит вам просмотреть, что вы написали. Вы сможете просмотреть весь список сообщений, найти конкретное содержимое (введя что-либо в строку поиска и затем щелкнув мышью кнопку **Search**) или просмотреть только сообщения с определенными ярлыками (щелкните кнопкой мыши ярлык в поле **Labels** (Действия с ярлыком...), расположенном левее). Когда сообщение, которое хотите редактировать, найдено, щелкните мышью ссылку **Edit** (Изменить). Если у вас есть сомнения по поводу уже опубликованного сообщения, щелкните кнопкой мыши ссылку **Delete** (Удалить) и удалите сомнительную публикацию целиком

- **Comments** (Комментарии). Это вкладка позволяет просматривать комментарии других читателей, которые они присоединили к вашим сообщениям (и те, которые Blogger определил как просто спам и заблокировал). Как вы узнаете из *разд. "Управление комментариями" данной главы*, вы можете решать, какие комментарии оставить, а какие удалить.
- **Settings** (Настройки). На этой вкладке сгруппировано довольно много параметров, разбитых на подгруппы. Вы можете задать все, начиная с базовой информации о вас и вашем блоге и заканчивая параметрами управления комментариями читателей и пользовательскими именами доменов.
- **Design** (Дизайн). На этой вкладке вы можете задать реорганизацию содержимого вашей страницы или выбрать другой шаблон. Если вас не радует внешний вид

блога, переходите на эту вкладку и без усилий создайте новый имидж вашего блога.

- **Monetize** (Коммерция). После того как блог создан, вы можете заработать на ваших читателях, отобразив для них рекламные объявления средствами программы Google's AdSense (более подробно обсуждаемой в *главе 14*). Укажите путь к этой вкладке, выберите, где должны располагаться рекламные объявления, и зарегистрируйтесь в AdSense (см. *разд. "Google AdSense" главы 14*).
- **Stats** (Статистика). На этой вкладке вы сможете проанализировать трафик вашего блога. Для этого применяется вариант сервиса Google Analytics, о котором вы узнали в *главе 11*, а значит, вы легко сможете выбрать свои популярные сообщения и любимые вами области местонахождения.
- **View Blog** (Просмотреть блог). Эта ссылка позволяет увидеть ваш блог глазами читателей. После внесения изменений в блог можно воспользоваться данной ссылкой, чтобы просмотреть результаты.

Настройка общих параметров

Теперь, когда ваш блог создан и действует, потратьте немного времени на тонкую настройку нескольких параметров. Далее вы добавите описание блога, выберете, сколько сообщений отображать на вашей исходной странице, и зададите часовой пояс для того, чтобы у ваших публикаций была правильная датировка. Попутно вы познакомитесь с многими настройками Blogger, которыми сможете управлять.

1. Если вы на **Dashboard** (Панель инструментов), щелкните кнопкой мыши ссылку **Settings** (Настройки) рядом с вашим блогом. Если вы уже работаете на странице управления, щелкните мышью вкладку **Settings** (Настройки).

   Вы увидите несколько ссылок под заголовком **Settings** (Настройки), в каждой из которых собрана своя группа параметров. Первоначально Blogger отображает группу **Basic** (Основные сведения), включающую некоторые из чаще всего регулируемых настроек.

2. Добавьте описание вашего блога.

   Этот текст появится на исходной странице обычно сразу под названием блога, хотя точное местоположение зависит от выбранного шаблона. Постарайтесь оставить в описании одно-два предложения, подчеркивающие изюминку вашего блога. Два хороших примера — "Трезвые откровения мясника без лицензии" и "Моя полосатая жизнь и приключения".

3. Прокрутите страницу и щелкните мышью кнопку **Save Settings** (Сохранить настройки).

   После сохранения настроек изменения вступят в силу немедленно. Но до проверки блога вам еще многое нужно сделать.

4. Под заголовком **Settings** (Настройки) щелкните мышью ссылку **Archiving** (Архивирование).

Архивирование — это процесс, применяемый Blogger для группировки старых публикаций и перемещения их за пределы видимости. У каждого архива есть ссылка на вашу исходную страницу. Например, если Blogger по вашему требованию создает архивы ежемесячно, он добавляет ссылки January 2011 (январь 2011), February 2011 (февраль 2011) и т. д. Если посетитель щелкнет кнопкой мыши одну из таких ссылок, Blogger отобразит сообщения, начиная с этого периода.

5. Задайте **Archive Frequency** (Частота обновления архива) и решите, хотите ли вы создать для каждого сообщения собственную страницу.

   Вы можете задать ежемесячное, еженедельное или ежедневное архивирование сообщений или совсем отказаться от него. Легкомысленные блоггеры считают, что лучший выбор — ежемесячное архивирование. Если вы публикуете сообщения ежедневно, можно разделить их на недельные группы, но со временем вы заполните вашу индексную страницу множеством дополнительных ссылок (по одной на каждую неделю ваших публикаций).

6. Щелкните мышью кнопку **Save Settings** (Сохранить настройки).

   И это еще не конец.

7. Под заголовком **Settings** (Настройки) щелкните кнопкой мыши ссылку **Formatting** (Форматирование).

   На этой вкладке вы можете выбрать, сколько публикаций Blogger будет отображать на вашей исходной странице и как он будет форматировать даты (рис. 13.12).

8. Выберите количество публикаций, выводимых на вашу первую страницу.

   Вы можете попросить Blogger отображать конкретное количество сообщений или определенное число дней. Например, вы можете попросить Blogger показывать все сообщения за последние 14 дней или отображать только последние три сообщения, независимо от даты их публикации. Для достижения наилучших результатов не загромождайте вашу исходную страницу большим количеством сообщений. Если вы публикуетесь ежедневно, придерживайтесь небольшого количества сообщений или ограничьтесь темами за текущую неделю.

9. Выберите формат даты и задайте часовой пояс.

   Blogger вставляет дату в каждое сообщение блога, в начало или в конец — в зависимости от выбранного шаблона. Если указать часовой пояс, не придется вручную вставлять дату в каждое создаваемое сообщение.

10. Щелкните мышью кнопку **Save Settings** (Сохранить настройки).

    Чтобы проверить, как выглядят эти изменения, щелкните кнопкой мыши ссылку **View Blog** (Просмотреть блог).

СЕКРЕТЫ РЕМЕСЛА

Групповое ведение блога

Не успеваете постоянно поддерживать блог в актуальном состоянии? Если хотите быть частью блогосферы, но не можете публиковаться чаще одного раза в месяц, по-

пробуйте разделить трудности с друзьями. Найдите естественную причину для объединения — например, коллеги могут создать блоги для обсуждения конкретных рабочих проектов, а семьи могут использовать их для постоянного общения (если они уже не пристрастились к Facebook). В более крупном масштабе групповое ведение блога позволит единомышленникам создать блог, который будет интереснее, чем сумма его составляющих. Два широко известных формирующих тенденцию групповых блога — Lifehacker (**http://lifehacker.com**), рассказывающий об эффективных трюках в хакерском стиле, применяемых в реальной жизни, и Daily Kos (**www.dailykos.com**), предлагающий политические новости и аналитику левого толка.

Создать групповой блог в Blogger легко. Выберите ваш обычный блог, перейдите на вкладку **Settings** (Настройки), щелкните кнопкой мыши ссылку **Permissions** (Разрешения), а затем кнопку **Add Authors** (Добавить авторов), чтобы вставить ваших последователей-блоггеров.

Вы должны предоставить только адрес электронной почты блоггера, которого хотите включить в список. Blogger отправит приглашения всем потенциальным участникам. Для выражения согласия блоггер должен щелкнуть кнопкой мыши ссылку в письме электронной почты (и создать учетную запись Google, если есть блоггер, у которого ее нет).

У всех блоггеров есть право публиковать сообщения. Кроме того, вы можете придать некоторым из них административный статус, что позволит им добавлять новых блоггеров самостоятельно (и удалять уже имеющихся).

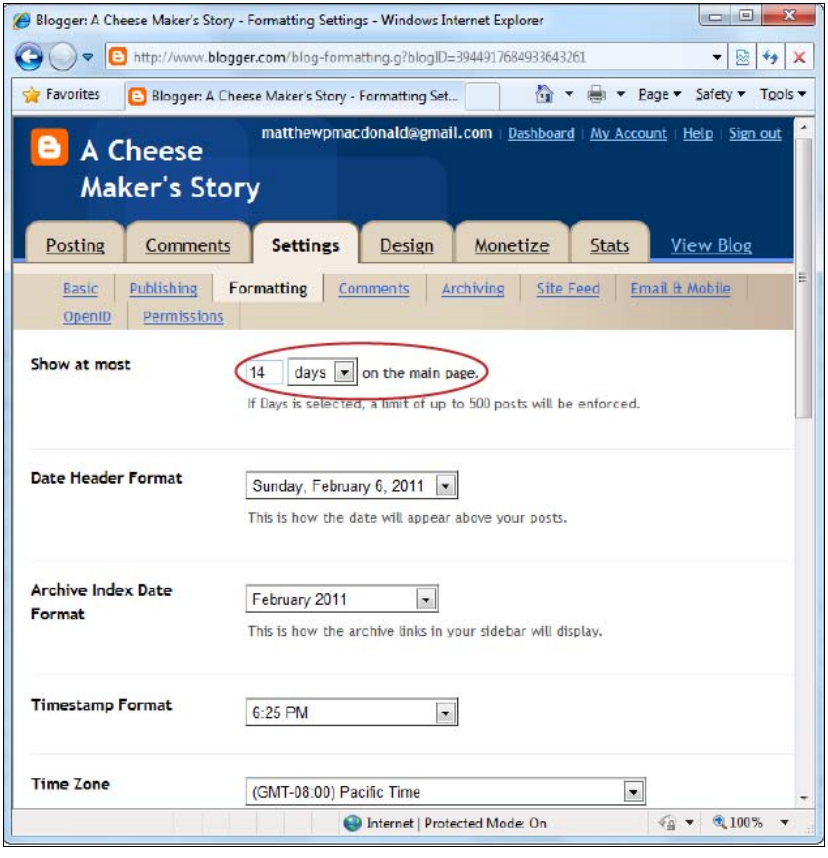


Рис. 13.12. Пример настройки вашего блога на отображение двухнедельного набора сообщений

Настройка вашего профиля в Blogger

Интересуетесь настройкой данных, которые появляются на вашей исходной странице помимо ваших публикаций? Blogger извлекает их из вашего профиля пользователя и их легко изменить. Выполните следующее.

1. Перейдите на **Dashboard** (Панель инструментов) на главной странице Blogger.

   Если вы на странице с вкладками, щелкните кнопкой мыши ссылку **Dashboard** (Панель инструментов) в правом верхнем углу страницы. В противном случае перейдите на сайт **www.blogger.com** и зарегистрируйтесь.

2. Щелкните кнопкой мыши ссылку **Edit Profile** (Изменить профиль), которая отображается под картинкой с силуэтом головы и плеч.

   На экране появится страница вашего профиля.

3. Отредактируйте данные профиля (рис. 13.13).

   На этой странице вы можете ввести некоторую информацию о себе. На исходной странице отображается только часть этих данных, включая ваше имя (**Display Name** (Отображаемое имя)), необязательную фотографию, ваше местожительство (**City, State, Country** (Город, Регион\область, Страна)) и описание в поле **About Me** (Обо мне).

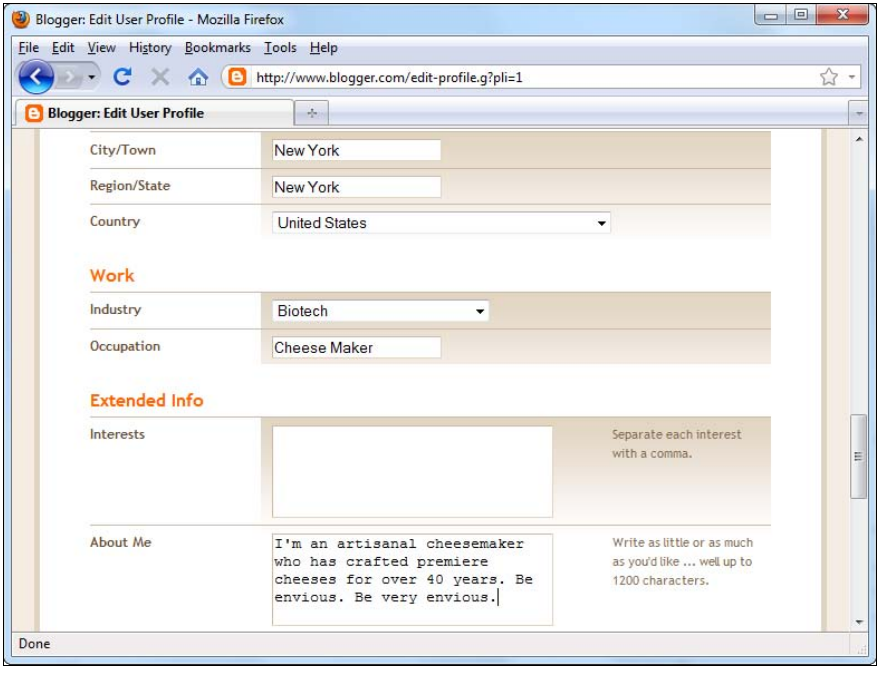


Рис. 13.13. Поле **About Me** (Обо мне) — одна из самых важных частей вашего профиля, т. к. оно бросается в глаза на исходной странице вашего блога. Другие поля, такие как **Interests** (Интересы), **Occupation** (Род занятий) и **Favorite Movies** (Любимые фильмы), не отображаются на вашей исходной странице, но читатели могут найти их, если щелкнут мышью ссылку **View my complete profile** (Весь профиль) под ссылкой **About Me** (Обо мне)

4. После введения всей необходимой информации профиля щелкните мышью кнопку **Save Profile** (Сохранить профиль).

   Вернитесь на главную страницу Blogger (щелкните мышью ссылку **Dashboard** (Панель инструментов)) и затем ссылку **View Blog** (Просмотреть блог), чтобы увидеть результаты внесенных изменений. На рис. 13.14 показан пример.

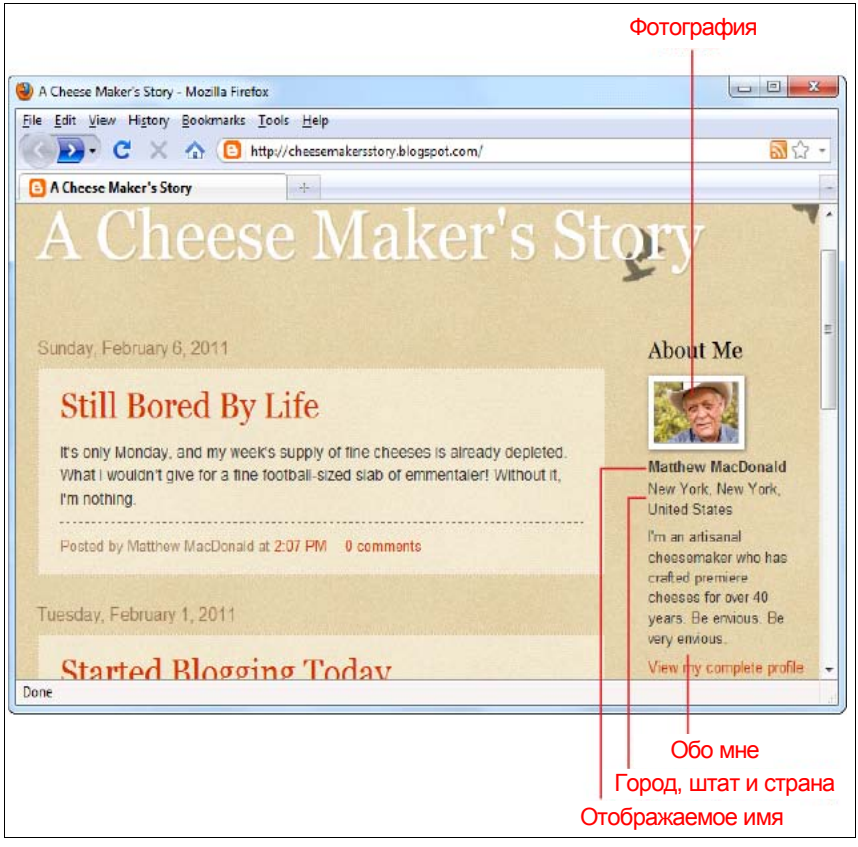


Рис. 13.14. Пример хорошо настроенной исходной страницы, на которой отображается описание пользователя и текст из поля **About Me** (Обо мне)

Присвоение вашему блогу имени пользовательского домена

Несмотря на то что нет ничего плохого в URL с окончанием .blogspot.com, есть все основания присвоить вашему блогу имя пользовательского домена. Независимо от того, как сильно вы любите сервис Blogger сейчас, придет время, когда вы сможете перейти на другой сервис. Такое преобразование пройдет легче, если не придется сообщать всем вашим читателям о необходимости обновить их закладки и направлять их на совершенно новый Web-адрес.

Есть три способа получить пользовательский домен для вашего блога.

- Купить домен у сервиса Blogger. Это самый простой вариант.
- Купить домен у другого продавца доменов. Или же использовать имя домена, которое вы уже купили, но еще не применяли. В любом случае ваша цель — настроить имя вашего домена так, чтобы оно указывало на Web-серверы Blogger (см. п. 2 в списке далее). Такой подход дает больше гибкости, но требует некоторой настройки, нуждающейся в согласовании с вашим продавцом имени домена.
- Использовать домен совместно с Web-сайтом, который у вас уже есть. Например, если у вас есть Web-сайт **www.deviousweevils.org**, вы можете разместить ваш блог на поддомене **blog.deviousweevils.org**. Не все Web-хосты предоставляют такую возможность. К сожалению, нельзя подключить подпапку вашего Web-сайта к сервису Blogger (поэтому оставьте надежду на получение вашим блогом адреса **www.deviousweevils.org/blog**). Для этого вам потребуется блог с самостоятельным хостингом (см. *разд. "Управление блогом" данной главы*).

> ***Примечание***
>
> Вплоть до середины 2010 г. Blogger предоставлял средства FTP-хостинга, позволяющие разработчикам хранить файлы Blogger на их собственных Web-серверах. Хотя это средство облегчало объединение самостоятельного Web-сайта с блогом, оно никогда не пользовалось популярностью у обычных блоггеров, и компания Google упразднила его.

Независимо от выбранного вами варианта адрес вашего блога с окончанием .blogspot.com будет обслуживаться и дальше, и все ваши файлы останутся на Web-серверах Blogger. Вы просто создадите для своих почитателей новый путь к вашему блогу.

Вот что для этого следует сделать.

1. Если вы используете домен от другого Web-хоста, убедитесь, что вы купили нужный вам домен. Если вы собираетесь купить имя домена у сервиса Blogger, переходите сразу к пункту 3.

   В *главе 3* написано гораздо больше об искусстве покупки имени домена. Только напомню о том, что не нужно платить за место на Web-хосте, потому что оно вам не нужно. Если у вас уже есть Web-хост, вы можете просто купить дополнительный домен.

2. Если вы пользуетесь доменом от другого Web-хоста, настройте его так, чтобы он указывал на ваш блог.

   Для этого следуйте инструкциям Blogger со страницы **http://tinyurl.com/n2zsbl**. Для внесения изменений вам потребуется помощь вашего Web-хоста, но в инструкциях вы найдете все нужные вам технические подробности.

3. Сейчас самое время изменить конфигурационные настройки Blogger. Если вы на странице **Dashboard** (Панель инструментов), щелкните кнопкой мыши ссылку

Settings (Настройки) рядом с вашим блогом. Если вы уже работаете на странице управления, щелкните мышью вкладку **Settings** (Настройки).

В любом случае вы перейдете на уже знакомую страницу **Settings**.

4. Щелкните кнопкой мыши ссылку **Publishing** (Публикация) под заголовком вкладки **Settings**.

   На этой странице можно выбрать новый адрес с окончанием .blogspot.com или указать пользовательский домен.

5. Щелкните кнопкой мыши ссылку **Custom Domain** (Персональный домен).

   Появится текстовое поле, в котором вы можете подобрать URL, если найдете подходящий, купите его у Blogger (обычно он стоит 10 долларов в год). Если вас все устраивает, введите имя домена, проверьте его доступность (щелкните мышью кнопку **Check Availability** (Проверить доступность)) и следуйте инструкциям, касающимся его оплаты с помощью сервиса Google Checkout.

   Если вам хотелось бы использовать свой собственный домен, переходите к следующему пункту.

6. Если у вас уже есть домен, щелкните кнопкой мыши ссылку **Switch to advanced settings** (Перейти к расширенным настройкам).

   Blogger откроет текстовое поле, в которое вы сможете ввести имя вашего домена.

   Введите имя домена, наберите проверочное слово (доказывающее, что вы не разновидность программы автоматизированной регистрации доменов) и щелкните мышью кнопку **Save Settings** (Сохранить настройки), чтобы узаконить изменения.

   Теперь у вас есть доступ к блогу через новый домен, а старый домен .blogspot.com будет продолжать исправно работать.

Настройка вашего шаблона

Шаблоны в сервисе Blogger очень важны. Они не только отражают стиль вашего блога (эпатажный, серьезный, технический, легкий и т. д.), но и определяют его компоненты и способ их отображения на странице. К счастью, Blogger позволяет настраивать многие компоненты ваших шаблонов. Например, вы можете перенести поле **About Me** (Обо мне) на новое место, изменить его внешний вид или просто удалить его.

Вы также можете добавлять новые области, например набор ссылок, указывающих на ваших любимых читателей-блоггеров, или боковую панель контекстных рекламных объявлений Google (см. *разд. "Google AdSense" главы 14*). Вы можете поступить и более радикально и полностью заменить шаблон, даже если уже публикуетесь годами. Blogger подгонит все ваши старые публикации под новый шаблон, так что ваши размышления старые и новые будут по-прежнему доступны пытливым читателям.

Для управления описанными деталями перейдите на вкладку **Design** (Дизайн). Вы увидите три способа настройки вашего шаблона.

- **Template Designer** (Дизайнер шаблонов). Применяйте этот вариант для выбора нового шаблона или изменения форматирования текущего шаблона. Возможности форматирования так разнообразны, что, вероятно, вам никогда не придется обращаться к редактированию непосредственно HTML-разметки.
- **Page Elements** (Элементы страницы). Используйте этот вариант для реорганизации макета вашего блога, удаления непонравившихся вам компонентов или добавления новых типов контента. Например, можно быстро вставить перечень ссылок, список ваших читателей, слайдшоу или совместно используемую ссылку из Facebook.
- **Edit HTML** (Изменить HTML). Применяйте этот вариант для настройки непосредственно HTML-разметки в вашем шаблоне.

В следующих разделах вы увидите, как пользоваться этими средствами.

Переформатирование шаблона

Впервые создавая блог, вы выбираете шаблон. Спустя какое-то время (или сразу) вы можете решить, что шаблон больше не соответствует вашему контенту. К счастью, вы можете довести блог до совершенства, изменяя в дизайнере шаблонов шрифты, цвета и параметры верстки. Вот как это делается.

1. На вкладке **Design** (Дизайн) щелкните кнопкой мыши ссылку **Template Designer** (Дизайнер шаблонов).

   Загрузится страница с набором инструментов форматирования вверху и окном предварительного просмотра результатов внизу (рис. 13.15).

   Blogger делит инструменты форматирования на несколько категорий.

   - **Templates** (Шаблоны) позволяет выбрать полностью новый шаблон. Ограничивает вас теми шаблонами, которые вы видели во время создания блога, поэтому нет большого смысла использовать эту категорию.
   - **Background** (Фон) позволяет выбрать набор тематических фоновых цветов и фоновую картинку. Для фонового изображения можно выгрузить и свое собственное творение или выбрать из большого набора готовых картинок, собранных в категории, например, **Abstract** (Абстрактные), **Food & Drink** (Еда и напитки) и **Travel** (Путешествие).
   - **Layout** (Дизайн) позволяет переключаться между несколькими базовыми макетами. Можно иметь одну простую колонку, можно добавить боковую панель с любой стороны (или с обеих сторон) или поместить все ваши гаджеты в нижнюю область, нижний колонтитул или подвал вашего блога. Для перемещения отдельных фрагментов контента по странице применяйте средство **Page Elements** (Элементы страницы), описанное в следующем разделе.
   - **Adjust Widths** (Регулировка ширины) позволяет задать максимальную ширину (в пикселах) основной колонки вашего блога и боковых панелей. Блоги

пользуются эластичными макетами, которые растягиваются в соответствии с размерами окна браузера, наряду с ограничением максимальной ширины, которое защищает компоненты страницы от чрезмерного увеличения. Этот вариант макета описан в *разд. "Максимальная ширина: подстраховка" главы 9.*



Рис. 13.15. Готовы к изменениям? У блога появляется новый фон, более широкий одноколоночный макет, причудливые шрифты (обратите особое внимание на заголовок) и соответствующая цветовая схема

- **Advanced** (Дополнительно) содержит самые интересные элементы управления форматированием. В списке выберите деталь, которую собираетесь форматировать (**Page Text** (Текст страницы), **Links** (Ссылки), **Blog Title** (Название блога), **Post** (Заголовок сообщения) и т. д.), и затем используйте элементы управления, расположенные справа, чтобы изменить ее параметры (шрифт и цвет). Параметры шрифта особенно впечатляют. Они не только поддерживают стандартные Web-шрифты, такие как Arial, Verdana и Georgia, но включают библиотеку бесплатных шрифтов Google, реализованных с помощью

нового современного CSS-средства внедрения шрифтов (см. *разд. "Встроенные шрифты" главы 6*).

2. Щелкните мышью кнопку **Apply to Blog** (Применить к блогу), чтобы запомнить ваши изменения (по крайней мере до тех пор, пока не потребуется новое переформатирование).

   Или, если вам не понравились сделанные изменения, щелкните кнопкой мыши ссылку **Back to Blogger** (Назад к Blogger), чтобы отказаться от них.

Изменение (или реорганизация) гаджетов в вашем шаблоне

Главная часть вашего блога — ряд публикаций, занимающий середину страницы. Но в каждом блоге есть дополнительные блоки контента, расположенные на боковой панели рядом с вашими публикациями или в нижнем колонтитуле, в нижней части страницы. Примерами могут служить область **About Me** (Обо мне), **Blog Archive** (Архив блога) (в котором перечисляются ваши самые последние публикации, сгруппированные в виде дерева по месяцам) и список **Google Friends** (Постоянные читатели) (список людей, регулярно читающих ваш блог). В Blogger эти компоненты называют *гаджетами*, и сервис предоставляет вам полный контроль над ними, включая перемещение, добавление новых и удаление имеющихся. Вот как это делается.

1. На вкладке **Design** (Дизайн) щелкните кнопкой мыши ссылку **Page Elements** (Элементы страницы).

   Blogger отображает текущую структуру вашего блога, обозначая местоположение ваших текущих гаджетов (рис. 13.16).

2. Переместите ваши гаджеты.

   Первоначально блоки контента отображаются на странице в соответствии с определением выбранного вами шаблона. Но у вас есть масса возможностей все изменить. Самое простое изменение, которое вы можете сделать, — перенос гаджета. Просто перетащите его мышью на новое место (см. рис. 13.16).

3. Если нужны новые гаджеты, добавьте их.

   Для вставки нового гаджета щелкните кнопкой мыши одну из ссылок **Add a Gadget** (Добавить гаджет) на схеме, отображаемой на вкладке **Page Elements** (Элементы страницы). Вы увидите раскрывающийся список с большим набором удобных готовых дополнений, предлагаемых Blogger. Щелкните кнопкой мыши одно из них, и вы увидите в окне предварительного просмотра его внешний вид. Если он вам понравится, щелкните мышью кнопку плюс (+) и добавьте гаджет на вашу страницу, а затем перетащите его мышью в любое понравившееся вам место.

   Некоторые гаджеты очень просты — блоки обычного текста, списки, изображения и ссылки (рис. 13.17). Но есть и очень интересные. Далее перечислено несколько примеров.

   - **Blog List** (Список блогов) позволяет вставить дополненный ссылками каталог блогов, которые вам интересны или которыми вы восхищаетесь. Гаджет **Link List** (Список ссылок) делает то же самое, но для обычных Web-сайтов.

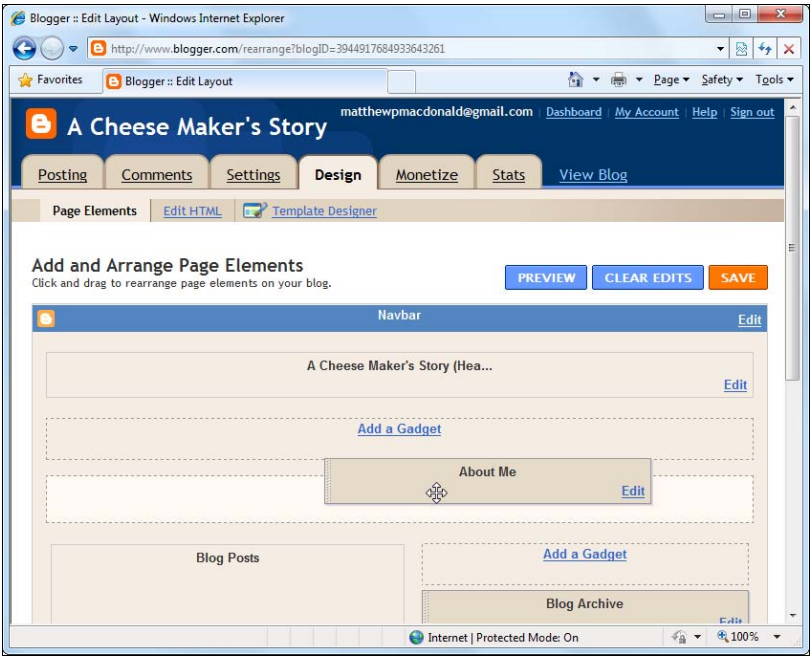


Рис. 13.16. Вы можете перетащить мышью любой гаджет в одну из выделенных на странице областей. Ваши возможности зависят от макета, заданного вами в дизайнере шаблонов (см. предыдущий раздел). Можно выбрать левую или правую боковые панели, область верхнего заголовка и область нижнего колонтитула. В данном примере раздел **About Me** (Обо мне) переносится на более заметное место



Рис. 13.17. У этого блога три новых гаджета справа: поле рекламы, список ссылок и список популярных сообщений

- **Search Box** (Окно поиска) вставляет строку поиска Google, которую посетители смогут использовать для поиска сообщений в вашем блоге.
- **Popular Posts** (Популярные сообщения) отображает список самых читаемых публикаций в вашем блоге, таким образом новые читатели легко смогут найти их и прочесть. Этот гаджет очень популярен на новостных сайтах. **Recent Comments** (Последние комментарии) аналогичен, но отображает самые недавние комментарии к любому вашему сообщению. Это позволяет читателям следить за мнением других.
- **AdSense** отображает рекламные объявления Google, описанные в *разд. "Google AdSense" главы 14*. Прежде чем вы сможете воспользоваться этим гаджетом, необходимо подключиться к сервису AdSense, ответив на пару вопросов на вкладке **Monetize** (Коммерция).
- **Twitter Updates** (Обновления Twitter) отображает последние твиты, если вы пользуетесь Twitter (см. *разд. "Twitter" главы 12*).
- **Flickr Photostream and Picasa Photostream** отображает фотографии, выгруженные вами в последнее время на сайты Flickr или Picasa.
- **Poll** (Опрос) позволяет проводить опрос ваших читателей и подсчитывать их голоса.

4. Настройте или удалите уже имеющиеся гаджеты.

   Для изменения параметров гаджета щелкните кнопкой мыши ссылку **Edit** (Изменить) в поле гаджета. Варианты настройки зависят от конкретного гаджета. Например, гаджет **Blog Archive** (Архив блога) отображает календарь прошлых сообщений блога. Вы можете изменить порядок отображения, способ группирования сообщений сервисом Blogger или выбранный им формат дат. Если вы редактируете гаджет **Blog List** (Список блогов), вы можете предоставить список блогов, на которые хотите ссылаться.

   Для удаления гаджета щелкните кнопкой мыши ссылку **Edit** (Изменить), а затем кнопку **Remove** (Удалить).

   Вы не сможете удалить только **NavBar** (Панель навигации), тонкую полоску в верхней части вашего блога. Посетители, просматривающие ваш блог, могут применять эту панель для перехода от одного блога к другому, регистрации в их собственном блоге или (что полезнее всего) искать ваш блог по ключевым словам. Но несмотря на то, что вы не можете удалить навигационную панель, можно задать ее цвет так, чтобы она соответствовала стилю вашего шаблона и сочеталась с окружающей обстановкой.

   Совет

   Если очень-очень хочется удалить навигационную панель, можно отредактировать HTML-разметку вашего шаблона и изменить ее стиль, сделав ее невидимой. Описание этого приема см. на странице **http://tinyurl.com/4tbxvl8**.

5. Когда закончите вставку и реорганизацию гаджетов, щелкните мышью кнопку **Save** (Сохранить).

Вы также можете щелкнуть мышью кнопку **Preview** (Просмотр), чтобы увидеть, как будет выглядеть ваш блог, когда вы примите изменения, или щелкнуть мышью кнопку **Clear Edits** (Убрать изменения), чтобы отменить ваши изменения и вернуться к прежнему положению вещей.

Редактирование HTML-разметки в шаблоне

Инструменты, которыми вы пользовались до сих пор, предоставляли широкие возможности для изменения внешнего вида вашего блога. Но существует еще один инструмент предельного уровня для блоггеров, желающих неограниченно контролировать свои страницы, — HTML-разметка в вашем шаблоне. Благодаря навыкам, полученным вами на протяжении всей книги, вы практически можете изменить в ней что угодно.

Конечно, это потребует гораздо больше работы, чем применение дизайнера шаблонов сервиса Blogger. Тогда зачем нужны эти дополнительные действия? Есть несколько веских причин.

- *Вы хотите применить совершенно новый шаблон.* У Blogger есть для выбора лишь небольшой набор шаблонов. В сети можно найти дополнительные (задайте в строке поиска "blogger template" (шаблон blogger)).
- *Вы хотите применить сложное CSS-форматирование.* У Blogger нет многочисленных инструментов для управления дизайном помимо шрифтов и цветов. Если вы хотите использовать что-то иное — тонкую настройку полей и отступов, например, или границы и фоновые изображения — придется покопаться в шаблоне и изменить его таблицу стилей.
- *Вам нужна настройка без ограничений.* Добавление гаджетов — модное средство, но оно не дает полного и скрупулезного управления, обеспечиваемого редактированием HTML. Конечно, временами слишком тщательный контроль сильно усложняет жизнь.

Шаблоны сервиса Blogger — это просто HTML-документы, описывающие страницы вашего блога. На первый взгляд кажется странным, что у современного блога с множеством страниц есть только относительно простой шаблон. Хитрость в том, что он определяет специальные заменяемые области. Когда посетитель запрашивает страницу в вашем блоге, Blogger в первую очередь обращается к вашему шаблону и вставляет соответствующий контент всюду, где находит специальные коды.

Например, если Blogger находит странно выглядящий код

```
<title><data:blog.pageTitle /></title>
```

Он заменяет выделенный жирным элемент названием вашего блога. Окончательный HTML-файл, создаваемый им для вашей исходной страницы, на самом деле содержит следующий текст.

```
<title>A Cheese Maker's Story</title>
```

Для изменения HTML-кода в вашем шаблоне сделайте следующее.

1. На вкладке **Design** (Дизайн) щелкните кнопкой мыши ссылку **Edit HTML** (Изменить HTML).

Вы увидите текстовое поле с полным HTML-файлом вашего шаблона, код и все остальное. Вы можете прямо здесь изменить некоторые детали, например, правила форматирования во встроенной таблице стилей. Но если вы хотите добавить новый элемент или перекомпоновать страницу, необходимо разобраться в коде шаблонов Blogger. Дополнительную информацию см. на странице **http://tinyurl.com/295vg5**.

> ***Совет***
>
> После того как шаблон отшлифован, сделайте его резервную копию, прежде чем вносить дальнейшие изменения. В противном случае вы можете повредить его и не сможете вернуться к корректной версии. Для создания резервной копии щелкните кнопкой мыши ссылку **Download Full Template** (Загрузить весь шаблон) и затем выберите место хранения на вашем компьютере. Для восстановления шаблона щелкните мышью кнопку **Browse** (Обзор), чтобы найти резервную копию на вашем компьютере, и затем кнопку **Upload** (Загрузить) для передачи его в Blogger.

2. Когда все изменения будут внесены, щелкните мышью кнопку **Save Template** (Сохранить шаблон).

   Blogger обновит ваш блог, чтобы немедленно начать использовать новый шаблон.

Управление комментариями

Обычно Blogger позволяет посетителям оставлять комментарии в вашем блоге. Это означает, что ваши читатели могут публиковать собственные мысли и дополнения, и Blogger отображает их наряду с вашими сообщениями.

Для того чтобы оставить комментарий, посетители должны иметь учетную запись Google или учетную запись OpenID (стандарт, который применяют многие сервисы ведения блогов и Web-сайты). Blogger настаивает на этом ограничении, чтобы уменьшить спам в комментариях — раздражающие сообщения, обычно публикуемые автоматическими программами, которые рекламируют товары. К счастью, у Blogger есть ваша поддержка. Он перехватывает сообщения, выглядящие как спам, и переносит их в специальную зону хранения. Затем ваша задача просмотреть эти комментарии и решить, пропускать ли их. Для этого щелкните кнопкой мыши вкладку **Comments** (Комментарии) и затем ссылку **Spam** (Спам). Blogger отобразит список помещенных на карантин комментариев ко всем вашим сообщениям в блоге. Если находите корректный комментарий, выделите его и щелкните мышью кнопку **Not Spam** (Не спам). Если тоните в ненужных комментариях, щелкните кнопкой мыши ссылку **Select All** (Выбрать: Все) и затем кнопку **Delete** (Удалить), чтобы очистить блог от мусора.

Это средство борьбы со спамом не перехватывает оскорбительные комментарии, оставляемые реальными людьми. Если в вашем блоге обсуждаются острые темы, возможно, потребуется более строгое регулирование, чтобы уберечь возбужденных комментаторов от нападок друг на друга. Контролировать комментарии можно двумя способами: удалять их или контролировать их.

Удаление комментариев

Первый вариант — удалять оскорбительные комментарии задним числом. Для этого просто щелкните кнопкой мыши вкладку **Comments** (Комментарии). Blogger покажет список всех комментариев, оставленных в вашем блоге в обратном хронологическом порядке (новые комментарии перечисляются в верхней части списка независимо от даты публикации, к которой они относятся). Если вы увидите непонравившийся вам комментарий, у вас есть три варианта обращения с ними.

- **Delete** (Удалить) удаляет комментарий немедленно.
- **Spam** (Спам) удаляет комментарий немедленно и предупреждает Blogger о том, что действует спаммер (что может привести к занесению ID комментатора в черный список).
- **Remove content** (Удалить содержание) удаляет текст комментария, но оставляет запись комментария в блоге с ID пользователя и пометкой "This post has been removed by the author" (Этот текст был удален автором). Порой такой подход может заставить несдержанного комментатора вести себя лучше и помешать другим критиковать вас за скрытую цензуру.

Единственная проблема при удалении комментариев — необходимость регулярной регистрации и запоздалая, задним числом реакция. Если у вас нет времени на постоянный контроль блога, или вы хотите быть уверены в том, что ни у кого нет шанса опубликовать провокационные заметки (даже краткие), вам нужен другой подход.

Модерация комментариев

Еще одна возможность — контроль или модерация комментариев, позволяющая им попасть в ваш блог только после того, как вы лично дадите "добро". Модерация комментариев потребует дополнительных усилий, но в некоторых ситуациях это единственный способ заблокировать нежелательные комментарии, особенно если у вас популярный блог или острая тема. Вот как следует контролировать комментарии.

1. Если вы на **Dashboard** (Панель инструментов), щелкните кнопкой мыши ссылку **Settings** (Настройки), расположенную рядом с вашим блогом. Если вы уже работаете на странице управления, щелкните мышью вкладку **Settings** (Настройки).
2. Под заголовком вкладки **Settings** щелкните кнопкой мыши ссылку **Comments** (Комментарии).

   Blogger предоставляет удивительно подробный набор параметров для управления комментариями. Например, вы можете разрешить анонимным читателям публиковать комментарии (обычно у них нет на это права) или отключить их комментарии целиком. Но самый интересный параметр управляет модерацией.

ПРИМЕЧАНИЕ

Разрешили вы публиковаться анонимным читателям или применяете модерацию комментариев, всегда следует оставлять включенной настройку **Show word verification**

(Показывать проверку на основе графического кода для комментариев?). Она заставляет читателей вводить проверочное слово с картинки прежде, чем опубликовать свои комментарии. Несмотря на свою назойливость, эта процедура выводит из строя автоматизированные программы публикации комментариев, оставляющие наихудший вид спама в комментариях.

3. Измените выбор переключателя **Comment moderation** (Ввод комментариев) со значения **Never** (Никогда) на **Always** (Всегда) или **Only on posts older than 14 days** (Только для сообщений старше 14 дн.).

   Вариант **Only on posts older than 14 days** — интересный компромисс. Он позволяет читателям помещать комментарии для новых сообщений в блоге, в то время когда приходит большинство комментариев, но переключает их на модерацию по истечении определенного числа дней. Эта модель хорошо работает, потому что старые публикации с гораздо большей вероятностью прокомментируют спаммеры, а не ваши читатели. Хотя Blogger предлагает 14-дневный период, вы можете его сократить для более непримиримой борьбы со спамом или наоборот увеличить, чтобы дать читателям высказаться без помех.

   При желании укажите ваш адрес электронной почты, чтобы Blogger мог оповещать вас о попытках опубликовать комментарии. Но это действие необязательно, т. к. вы можете контролировать комментарии с вкладки **Posting** (Отправка).

4. Щелкните мышью кнопку **Save Settings** (Сохранить настройки), чтобы сохранить новые настройки для комментариев.

   Теперь вы готовы к испытанию системы. Сначала выйдите из сервиса Blogger. Для тестирования системы модерации зарегистрируйтесь в Google с другой учетной записью или как анонимный посетитель (если ваш блог разрешает анонимным читателям оставлять комментарии). Вы могли бы оставить комментарий, пока были зарегистрированы как владелец блога, но в этом случае Blogger полагает, что вам не нужно перепроверять себя, и пропускает этап модерации.

5. Опубликуйте новый комментарий.

   Чтобы оставить комментарий, щелкните кнопкой мыши ссылку **Comments** (Комментарии) в конце сообщения блога. Вставьте комментарий и информацию о вашем ID, если потребуется, и затем щелкните мышью **Publish Your Comment** (Отправить комментарий). Blogger достаточно сообразителен, чтобы понять, что блог применяет модерацию, поэтому он выводит сообщение, объясняющее, что комментарий не появится до тех пор, пока владелец блога (т. е. вы) не одобрит его.

6. Просмотрите ваши комментарии.

   Чтобы увидеть новый комментарий, зарегистрируйтесь снова как владелец блога, щелкните кнопкой мыши ссылку **Comments** (Комментарии) и затем ссылку **Awaiting Moderation** (Ожидающие проверки) под заголовком вкладки. Вы увидите комментарии, ожидающие вашего одобрения (рис. 13.18).

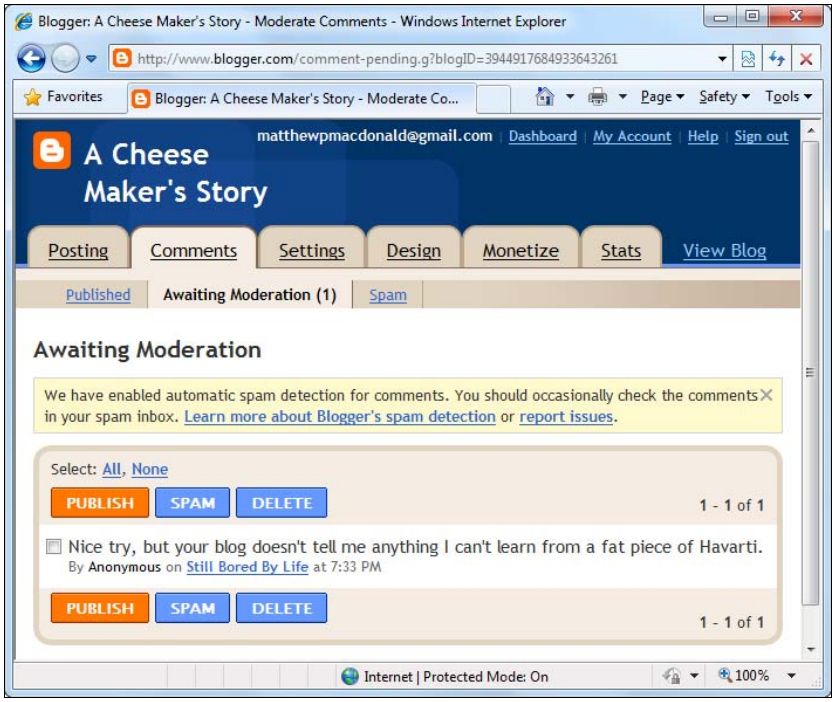


Рис. 13.18. У этого блога только один непроверенный комментарий. Щелкните мышью кнопку **Publish** (Опубликовать) для публикации его рядом с соответствующим сообщением или кнопку **Delete** (Удалить) для его уничтожения. Если вы хотите одобрить или удалить несколько комментариев одновременно, просто установите флажки рядом с каждым из них, а затем щелкните мышью кнопку **Publish** или **Delete**

СЕКРЕТЫ РЕМЕСЛА

Продвижение вашего блога

Вам нужно рекламировать ваш блог так же, как и любой другой Web-сайт. Несмотря на то, что можно применять любые методы, описанные в *главе 11*, для блогосферы существует еще несколько характерных приемов. Далее приводятся важные рекомендации, которые помогут приступить к делу.

- Поместите на ваш сайт список любимых блогов. Этот список представляет собой перечень ссылок на блоги, которые вам нравятся. Но такой список, кроме всего прочего, неявно делает заявление: "Существуют люди, которые мне нравятся" или "Я хочу, чтобы меня ассоциировали с этой группой людей". Другими словами, подобный список представляет лучшую часть социальной сети. Для создания списка любимых блогов добавьте гаджет **Blog List**.
- Принимайте участие в обсуждениях других блогов. Блоггеры — открытое сообщество. Если вы оставите остроумный комментарий на сообщение в чьем-то блоге, вполне вероятно, что хотя бы несколько читателей отправится в ваш блог, чтобы посмотреть, что еще вы можете сказать.
- Облегчайте другим доступ к вашим публикациям. Вам следует подогревать энтузиазм посетителей. Если вы сообщаете действительно интересные байки или новости, у читателей может возникнуть желание рассказать об этом всем своим друзьям, если вы облегчите им это занятие. Для поддержки читателей вставьте ссылку, позволяющую пересылать ваши сообщения по электронной почте. Выбе-

рите вкладку **Settings** (Настройки), щелкните кнопкой мыши ссылку **Basic** (Основные сведения) и затем задайте для режима **Show Email Post links** (Показывать ссылки на отправку электронной почты?) значение **Yes** (Да). Можно также добавить гаджет **Share It** (Поделиться), вставляющий ссылки, которые пользователи могут применять для того, чтобы быстро сослаться на ваше сообщение в Facebook или Twitter.

- Рекламируйте вашу RSS-ленту. RSS-ленты, обсуждавшиеся в *разд. "Агрегирование" данной главы*, работают с программами чтения RSS-лент. Приверженцы блогов любят их, т. к. они могут отслеживать десятки и даже сотни блогов одновременно. Программное обеспечение сервиса Blogger позволяет создать RSS-ленту, и ее следует рекламировать среди своих регулярных читателей. Для просмотра вашей ленты щелкните кнопкой мыши ссылку **Subscribe to Posts (Atom)** (Подписаться на: Сообщения (Atom)) на вашей исходной странице. Для вставки знакомой пиктограммы с оранжевым радаром, делающей это средство более наглядным, добавьте в блог гаджет **Feed** (Канал).
- Применяйте средство **BlogThis** (Поделиться). Огромное количество сообщений в блогах просто привлекают внимание к интересным новостям, скандальным историям или смешным фотографиям, появившимся в сети. Если вы не часто пишете в блог, ссылки на подобные материалы — отличный способ наполнить ваш блог. С помощью остроумного средства **BlogThis** вы можете одним щелчком кнопки мыши создать новое сообщение в блоге, ссылающееся на существующую Web-страницу. Есть два варианта применения **BlogThis** — можно вставить в Internet Explorer или Firefox инструментальную панель **Google Toolbar**, на которой есть для этого специальная кнопка, или добавить ссылку на пункты вашего меню **Bookmark** (Закладки) или **Favorites** (Избранное), делающую то же самое. Более подробную информацию см. на странице **http://tinyurl.com/3n4hvu**.

14

ГЛАВА 14

Как зарабатывать деньги на своем сайте

Если что-либо не предназначено для продажи в Интернете, возможно оно вообще не предназначено для продажи. Не секрет, что Интернет — это всемирный базар с бóльшим ассортиментом товаров, чем в "гаражных" распродажах за десятилетие. Web-путешественники генерируют огромный объем трафика, охотясь за скидками на туристические поездки, обсуждая выгодные предложения и прочесывая eBay в поисках дешевых товаров. Итак, как же получить свою долю в Web-капиталах?

Первый очевидный ответ — продавать реальные материальные товары. Интернет полон специализированных магазинов, торгующих предметами изобразительного искусства, ювелирными изделиями и товарами ручной работы. Но даже если у вас уже есть готовый для продажи товар, вам все равно понадобится несколько специализированных инструментов для превращения вашего угла в Web-пространстве в оживленный электронный магазин. Например, может потребоваться виртуальная корзина для откладывания товаров, которые посетители хотят купить, безопасный способ получения денег от покупателей, как правило, с помощью транзакций по кредитной карте. В этой главе вы узнаете, как реализовать оба компонента на вашем сайте, применяя торговые инструменты PayPal.

Даже если вы не ищете место для выгрузки ваших самодельных карандашей из рыбных костей, ваш Web-сайт все равно может пополнить ваш кошелек. На самом деле практически любой Web-сайт может стать прибыльным, продавая место для рекламных объявлений или рекомендуя товары других компаний. В этой главе вы увидите, как применять две самые популярные в Web *партнерские программы* (affiliate programs) — Google AdSense и Amazon Associates — для получения дополнительных денежных средств.

Примечание

Вы не гражданин США? Не беспокойтесь — в этой главе все соображения по поводу заработка касаются компаний, предоставляющих услуги по всему миру. Google, Amazon и PayPal позволят сгребать наличные лопаткой независимо от места вашего проживания.

Зарабатывание денег в Интернете

Web предлагает множество способов получения доходов, основанных на доверии. Далее перечислены самые популярные.

- *Добровольные пожертвования.* Это звучит дико, но некоторые Web-сайты просят разменную монету у посетителей. Пожертвования или взносы могут появиться, если ваш сайт предлагает по-настоящему значимый уникальный контент (рис. 14.1). В противном случае избавьте себя от напрасных хлопот. Расчет на то, что каждый из 1 000 посетителей, пришедших на ваш сайт, заплатит хотя бы по 10 центов, успеха не принесет. Они не заплатят.

Совет

Если вы считаете, что ваш Web-сайт предлагает уникальную и полезную информацию, и у ваших посетителей может возникнуть желание выложить несколько центов в знак признательности, можно использовать PayPal и добавить кнопку **Donate** (Пожертвования) (см. рис. 14.20). Подумайте о сообщении, например, "Купи мне пива/чашку кофе", размещенном над кнопкой **Donate**, чтобы виртуальная транзакция выглядела как обычные чаевые (и все понимали, что вы надеетесь собрать немного мелочи, а не сумму автокредита).

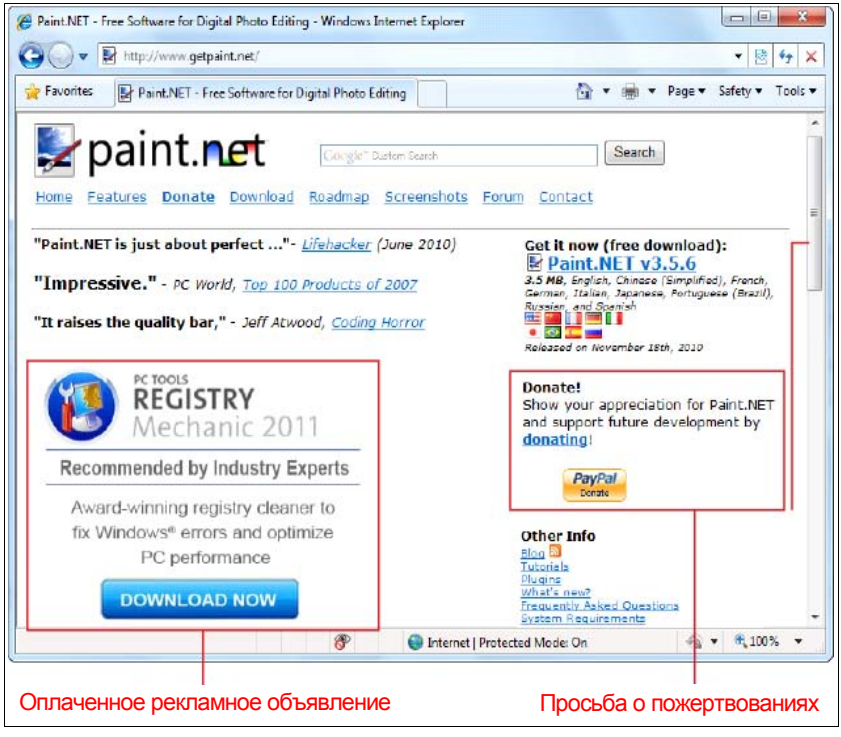


Рис. 14.1. Сайты, предлагающие свободные программы, чаще всего просят пожертвования. В данном случае Paint.net просят о небольших добровольных взносах. Но при таком способе зарабатывания денег даже у самых лучших Web-сайтов возникают проблемы. Paint.net подстраховывается, продавая на своем сайте место для рекламных объявлений

- *Рекламные объявления.* Самый популярный способ зарабатывания денег в Web — продажа небольших участков полезной площади Web-страниц. К сожалению, это отличный способ рассердить ваших посетителей, особенно если объявления отвлекают, не связаны с вашим сайтом или просто занимают слишком много места. Еще недавно рекламные объявления были наибольшим злом для Web-страниц. К счастью, в XXI веке мониторы стали больше, а компании, такие как Google, предоставляют адресную, ненавязчивую рекламу, в точности соответствующую остальному содержимому вашего сайта.
- *Партнерские программы.* Не лучше ли вместо аляповатых рекламных объявлений поместить несколько добрых слов о компании, которой вы действительно доверяете? Многие партнерские программы предлагают комиссионные за направление покупателей на их сайты. Например, если вы приводите обзор кулинарных книг, посвященных деликатесам, почему бы не включить ссылки на подобные книги на Web-сайте Amazon? Если заинтересовавшийся читатель купит книгу, программа-партнер Amazon выложит несколько долларов.
- *Торговля вещами.* Если у вас есть собственные товары для продажи, Web — отличное место, т. к. стоимость создания магазина в сети гораздо ниже стоимости аналогичного магазина в реальном мире. Вы можете, приложив совсем немного усилий, организовать прекрасный магазин, дополнив его фотографиями товаров и корзиной для купленных вещей. (А если у вас нет своих товаров для продажи, можно самостоятельно изготовить простые товары на сайте CafePress, как описано во *врезке "Уголок дизайнера. Продавайте созданный вами дизайн" разд. "Расширенные ссылки" данной главы*.)
- *Плата-за-контент.* Если у вас действительно интересное содержимое, вы можете попросить деньги у ваших посетителей, *прежде* чем пустить их на свой сайт. Предупреждение: добиться успеха будет еще труднее, чем в случае пожертвований, т. к. посетителям придется резко изменить свои убеждения. Подобной методикой пользуются авторитетные медиакомпании, например журнал Wall Street Journal, и нечистые на руку торговцы, обещающие тайные способы завоевания рынка недвижимости или получения бесплатных записывающих видеокамер.

> ***ПРИМЕЧАНИЕ***
>
> *Плата-за-контент* — это единственный вариант зарабатывания денег, об успешном применении которого вы не узнаете в этой главе. Все дело в его принципе действия, нуждающемся в аутентификации посетителей — другими словами, вам нужно уметь идентифицировать посетителей, чтобы узнать, заплатили они вам или нет. А для этого потребуется сложное программирование (или платный сервис сторонней компании).

Google AdSense

Даже если у вас нет товара для продажи, все равно остается один значимый актив: интерес ваших посетителей. Существует огромное множество компаний, готовых платить за это пристальное внимание.

Некоторые из этих компаний платят мизерную плату каждый раз, когда кто-то посещает страницу с их рекламой, другие платят только, когда читатель щелкает рекламное объявление кнопкой мыши или когда посетитель щелкает мышью рекламу *и* покупает что-нибудь. К счастью, вам не придется тратить часы на проверку всех этих вариантов, поскольку у Google есть программа размещения рекламы Google AdSense, которая с успехом победит практически любую похожую систему.

AdSense нуждается в выводе небольших текстовых, графических или видео рекламных объявлений на ваших страницах. Вы регистрируетесь, выделяете некоторое пространство на одной или нескольких страницах и помещаете поставляемый Google HTML-код (рис. 14.2). Об остальном позаботится Google, заполняя это пространство одним или несколькими рекламными объявлениями каждый раз, когда кто-либо запрашивает вашу страницу.

Простое отображение рекламных объявлений Google AdSense не даст вам ничего, но если посетитель щелкнет кнопкой мыши (кликнет) одно из них (и отправится на рекламируемый Web-сайт), вы заработаете несколько центов. Когда ваш доход достигнет 100 долларов, Google отправит вам по электронной почте чек или переведет деньги прямо на ваш счет в банке.

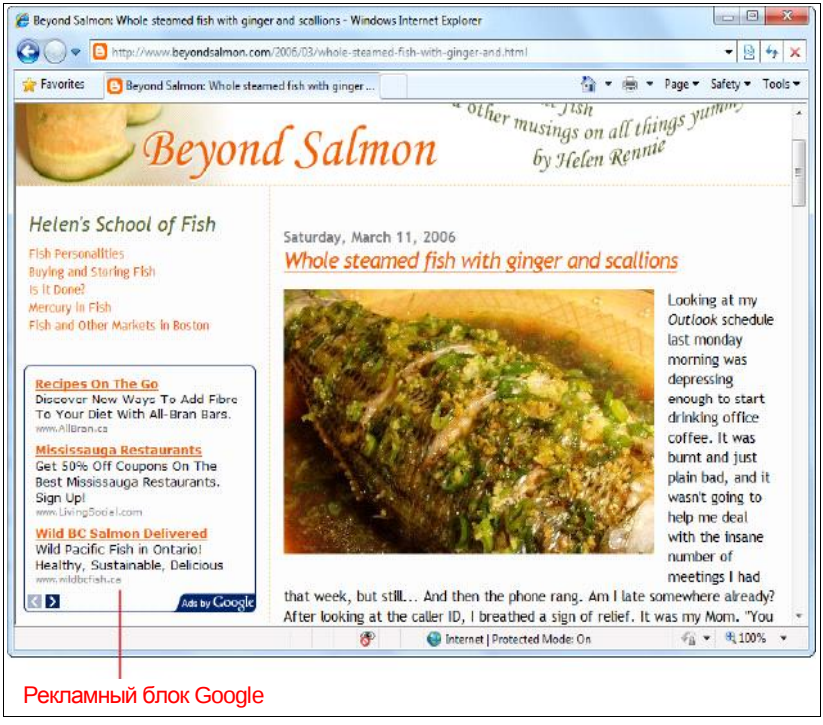


Рис. 14.2. Web-сайт содержит область из трех рекламных объявлений Google AdSense рядом с превосходным кулинарным рецептом. Реклама прекрасно сочетается с остальным контентом, т. к. использует тот же визуальный стиль (тот же цвет фона и шрифт), и ее содержание соответствует помещенной на странице статье. Google называет эту группировку рекламных объявлений "рекламным блоком". Вы выбираете макет рекламных объявлений и их количество на странице, поэтому только от вас зависит выбор едва заметного размещения рекламы или ее явное доминирование на вашей странице

Невозможно с уверенностью сказать, сколько стоит отдельный щелчок кнопкой мыши объявления AdSense. Дело в том, что рекламодатели Google конкурируют, набавляя плату за ключевые слова (см. *разд. "Google AdWords" главы 12*), и цены ключевых слов со временем могут меняться. Когда Google заплатит вам, вы сможете узнать, сколько стоили ваши щелчки кнопкой мыши (в сумме). Обычный щелчок мышью приносит около 10 центов, но цены за каждый щелчок часто колеблются от нескольких центов до нескольких долларов.

Прежде чем вы станете ярым приверженцем AdSense, следует узнать, что отличает AdSense от других программ, предлагающих рекламу. Далее перечислены самые важные ее достоинства.

- *Рекламные объявления AdSense контекстно-зависимы.* Google автоматически просматривает ваш сайт и выбирает рекламные объявления, тематически связанные с вашим контентом. Таким образом, если у вас сайт, посвященный SpongeBob SquarePants (мультсериал "Губка Боб Квадратные штаны)", Google предоставит рекламные объявления, предлагающие SpongeBob DVD-диски, надувные игрушки и аксессуары к дню рождения. Применение контекстно-зависимой рекламы во много раз лучше раздражения ваших посетителей неуместными предложениями совершенно не относящихся к делу товаров, например, суперсовременных камер наблюдения. Такие "целевые" рекламные объявления существенно повышают вероятность того, что посетитель щелкнет мышью объявление, заработав для вас плату-за-щелчок. А если вы боитесь, что в рекламе окажется сайт вашего конкурента, можно попросить Google отсеять его (см. далее *разд. "Интерфейс AdSense"*).
- *Рекламные объявления AdSense хорошо сочетаются с внешним представлением страницы.* Google предлагает для своих реклам ряд параметров верстки и цветовых схем, поэтому вы сможете подобрать дизайн и привлекательную цветовую схему вашего сайта.
- *Google обеспечивает справедливую оплату.* Как вы узнали в *разд. "Google AdWords" главы 12*, Google платит лицам, помещающим рекламу, за разные ключевые слова разные денежные суммы. Некоторые рекламодатели платят своим участникам одну и ту же сумму за любой переход по рекламной ссылке и проглатывают дополнительные деньги. С Google все иначе. Вы получаете деньги в соответствии с текущей ценой ключевого слова, что гарантирует постоянное получение выгодной ставки.
- *Первоначальные взносы отсутствуют.* Программа AdSense бесплатная.

> ***СОВЕТ***
>
> Не пытайтесь обманывать AdSense. Хитрые Web-разработчики пытались обыграть систему, многократно щелкая мышью собственные рекламные объявления или даже запуская автоматические программы, делавшие это за них. Проблема в том, что Google применяет различные методики для определения подозрительных примеров использования. Если за короткий промежуток времени отмечается неправдоподобное количество щелчков мыши с одного и того же компьютера, это вполне может быть расценено как жульничество и на ваш сайт будет навсегда наложен запрет.

Регистрация в AdSense

Дополнительную информацию о программе размещения рекламы компании Google см. на странице **www.google.com/adsense**. Кроме того, на странице **www.google.com/services/adsense_tour** представлен отличный обзор без излишних подробностей.

Когда вы будете готовы начать работу с программой AdSense, выполните следующие действия.

1. На исходной странице AdSense (**www.google.com/adsense**) щелкните мышью кнопку **Sign Up Now** (Зарегистрироваться).

   Google начнет со сбора данных учетной записи (рис. 14.3). Сначала введите URL вашего сайта и укажите его язык. Далее выберите, как вы действуете, от имени частного лица или в качестве зарегистрированной компании. Эти сведения определяют вид налоговой информации, которую следует собрать Google. Зарегистрированным коммерческим фирмам, находящимся в США, необходимо указать EIN (Employer Identification Number, Идентификационный номер фирмы), а гражданам США, действующим как частное лицо, нужно предоставить Google SSN (Social Security Number, Номер социального страхования). Гражданам дру-

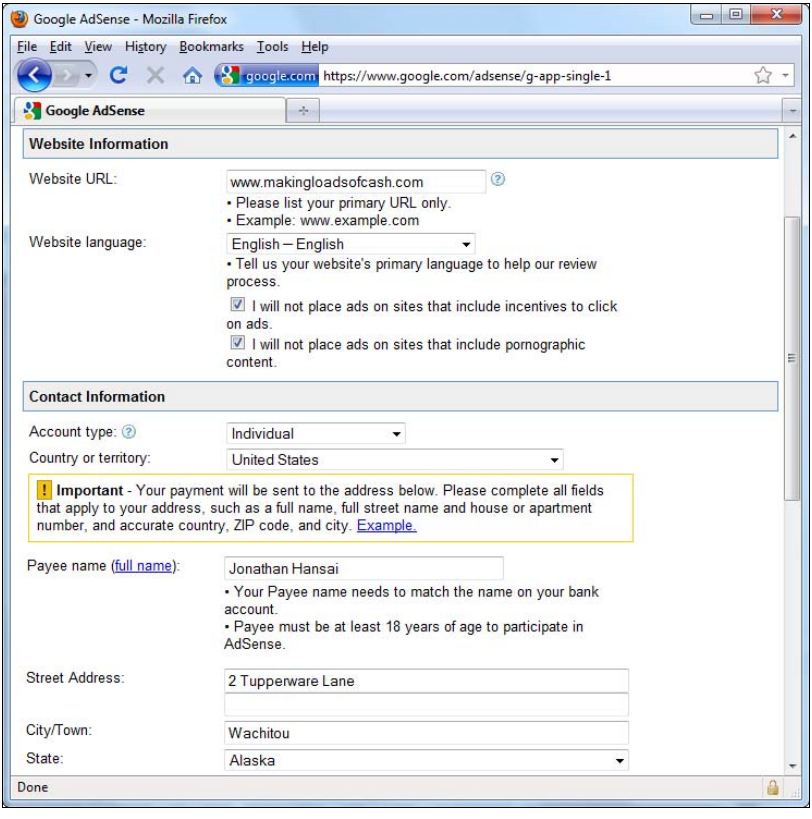


Рис. 14.3. Google AdSense собирает на одной странице всю нужную ей информацию, начиная с деталей, касающихся вашего Web-сайта, и заканчивая вашим адресом

гих государств, возможно, потребуется подать заявку на получение U.S. TIN (Taxpayer Identification Number, Индивидуальный номер налогоплательщика). Дополнительные подробности см. на странице **www.google.com/adsense/taxinfo**.

Примечание

Google не заплатит вам до тех пор, пока не получит всех налоговых данных. Для облегчения процесса вам предлагается корректная форма, которую можно представить на рассмотрение в Интернете. Google не будет напоминать вам о налоговой информации до тех пор, пока не соберет доход от размещения рекламы не менее 10 долларов.

В конце введите ваше имя, адрес и номер телефона.

Попутно вам придется установить несколько флажков, расположенных рядом с несколькими отрицаниями, подтверждающими, что вы не будете щелкать кнопкой мыши свои собственные рекламные объявления, помещать рекламу на порнографических сайтах и т. д.

2. По завершении щелкните мышью кнопку **Submit Information** (Отправить информацию).

   На следующей странице суммируется вся предоставленная вами информация.

3. Если у вас уже есть учетная запись Google, выберите вариант **I have an email and password I already use with Google services** (У меня есть адрес электронной почты и пароль (аккаунт Google), которые я уже использую для входа в службы Google). В противном случае выберите **I do not use these other services** (Я не использую другие службы...).

   Google предлагает невообразимое количество сетевых сервисов. Для того чтобы избавить себя от необходимости отслеживания десятков паролей, лучше сгруппировать их все под одной учетной записью Google. Если у вас уже есть такая запись, введите адрес электронной почты и выберите пароль.

4. Для завершения процесса щелкните мышью кнопку **Continue** (Продолжить).

   Теперь нужно подождать, пока Google свяжется с вами по электронной почте, чтобы подтвердить вашу учетную запись. Сначала, почти немедленно, Google отправляет вам электронное письмо с подтверждением. В нем есть ссылка, которую следует щелкнуть кнопкой мыши для подтверждения собственного адреса электронной почты.

   Далее некто в компании Google должен бегло просмотреть ваш сайт, чтобы подтвердить его реальное существование и отсутствие на нем рекламы незаконной деятельности (например, предложений пиратских копий Windiws 7). После того как дело сделано, что обычно занимает пару дней, вы получите второе сообщение, подтверждающее активацию вашей учетной записи.

НА ПРОФЕССИОНАЛЬНОМ УРОВНЕ

Правила AdSense

Google требует соблюдения группы правил, которым ваш Web-сайт должен следовать. Многие просто отвечают здравому смыслу, но все же стоит бросить беглый взгляд на них.

- Вы не можете помещать рекламные объявления Google в сообщения электронной почты или всплывающие окна; слишком велико желание спаммеров неправомочно использовать систему.
- Вы не можете помещать рекламу на страницы, не содержащие реального контента. К ним относятся страницы ошибок, входа в систему, регистрации, приветствия и страницы, находящиеся в разработке. Вы, конечно же, не можете создавать страницы, содержащие только рекламу и ничего больше.
- Нельзя заслонять части рекламного объявления (например, помещая поверх них другие элементы с помощью таблицы стилей). Видимым должно быть все содержимое рекламного объявления.
- Нельзя щелкать кнопкой мыши собственные рекламные объявления. Запрещено применять автоматические программы, выполняющие щелчок кнопкой мыши вместо вас. И наконец, нельзя с помощью угроз и вознаграждений заставлять посетителей щелкать мышью ваши ссылки.
- Ваш Web-сайт не должен содержать ненормативную или оскорбительную лексику, материалы, охраняемые авторским правом, порнографию, сведения о способах взлома современных систем безопасности, пропаганду наркотиков, агрессивные речи или что-либо, относящееся к азартным играм.

Полное описание политики AdSense см. на странице **www.google.com/adsense/policies**.

Интерфейс AdSense

Теперь, когда у вас есть учетная запись AdSense, вы готовы к формированию нескольких рекламных объявлений и размещению их на своем сайте. Перейдите на страницу **www.google.com/adsense** и введите свой адрес электронной почты и пароль. На экране появится страница AdSense, разделенная с помощью вкладок в верхней части на четыре области (рис. 14.4).

Далее перечислены эти области.

- **Home** (Главная страница). На этой вкладке представлена важная сводная информация, включая сумму денег, заработанную вами за текущий день, и сообщения о любых проблемах, мешающих отображению ваших рекламных объявлений (называемые оповещениями — alerts).

  На панели, расположенной слева, вы увидите дополнительные ссылки. Например, можно просмотреть историю платежей, в которой перечислены все чеки, отправленные вам Google по электронной почте (щелкните кнопкой мыши ссылку **Payments** (Платежи)), или изменить информацию, предоставленную вами в момент регистрации, такую как ваш почтовый адрес и налоговые сведения (щелкните мышью ссылку **Account settings** (Настройки аккаунта)). Или щелкните кнопкой мыши ссылку **Resources** (Ресурсы), чтобы просмотреть блог AdSense, пообщаться с другими пользователями на справочном форуме программы AdSense и посмотреть видеоролики с вдохновляющими историями успеха.
- **My ads** (Мои объявления). Это отправная точка для генерации рекламных объявлений AdSense, именно здесь вы задаете формат отображения объявлений и получаете код для вставки в ваши Web-страницы.

- **Allow & block ads** (Разрешить/блокировать объявления). На этой вкладке вы управляете одним из самых замечательных инструментов программы AdSense — отфильтровыванием нежелательных рекламных объявлений. Например, можно попросить Google воздерживаться от показа рекламных объявлений определенных категорий или конкретных Web-сайтов. При этом преследуется двоякая цель: помешать вашим конкурентам и гарантировать вашим посетителям отображение только значимых для них объявлений. (Если рекламные объявления неуместны, меньше вероятность того, что посетители будут щелкать их кнопкой мыши, и ваши доходы резко упадут.)

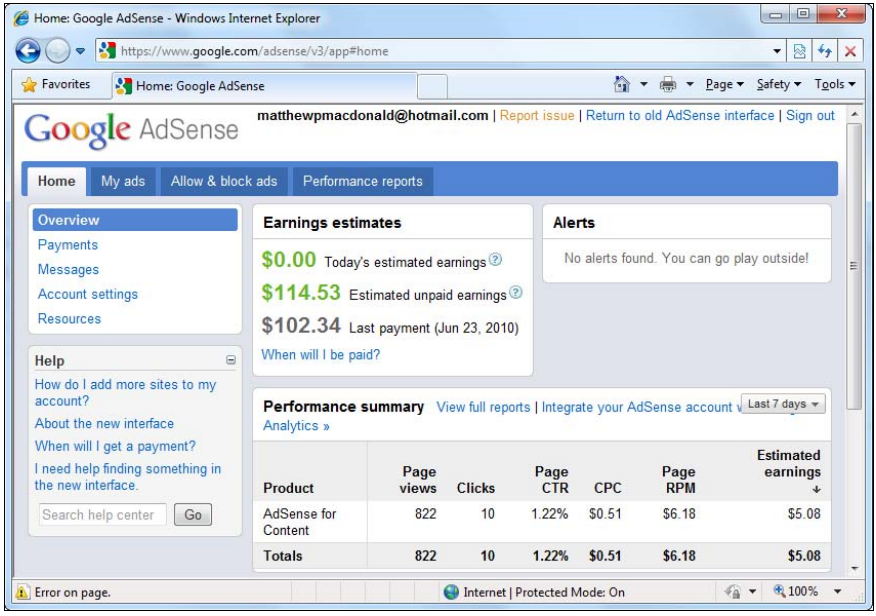


Рис. 14.4. Google делит страницу программы AdSense на ряд вкладок. Вы начинаете с вкладки **Home** (Главная страница), на которой настраиваете свою учетную запись AdSense и проверяете деньги, заработанные вами к данному моменту

- **Performance reports** (Отчеты по эффективности). Эти отчеты помогут вам оценить успех ваших рекламных объявлений. Вы начинаете с графика, на котором показано, сколько денег вы заработали за последнюю неделю. Можно настраивать этот график, меняя параметры и играя с фильтрами для получения именно тех данных, которые вам нужны. Для более серьезного анализа можно загрузить одним щелчком кнопки мыши числовые данные в формате CSV (Comma Separated Value, Значения, разделенные запятыми), подобном формату Excel.

Примечание

Ни один вид отчета не даст вам информации о том, сколько стоит отдельный щелчок кнопкой мыши или какое конкретное рекламное объявление привлекло внимание читателей. Вместо этого Google предоставляет оценку стоимости щелчка в текущий день (то, что называется CPC (cost-per-click, *цена за клик*)). Google также отображает и другую полезную информацию, например, процент щелчков ваших посетителей по

рекламным объявлениям на вашем сайте (называемый CTR (clickthrough rate)). Например, CTR страницы 2% означает, что если вашу страницу запрашивали 100 раз, рекламу на ней щелкнули мышью только дважды.

Создание рекламного объявления

Google предлагает четыре основных типа рекламных объявлений.

- *Текстовые рекламные объявления.* Это короткие модули, содержащие только текст, как представленные на рис. 14.2. Они начинаются ссылкой, за которой следует фрагмент текста и иногда имя Web-сайта рекламодателя, расположенное в нижней части модуля. В каждом рекламном объявлении в сумме всего 100 символов. Текстовые объявления, как правило, отображаются группами от одного до четырех в каждой.

> ***Примечание***
>
> Текстовые рекламные объявления до сих пор самый популярный вид рекламы Google. Они отлично гармонируют с фоном, что приводит к меньшему раздражению посетителей (и это плюс) и меньшему числу щелчков мыши (и это, скорее, минус).

- *Графические рекламные объявления.* Такие объявления на самом деле — изображения разных размеров. Графическая реклама более навязчива, чем текстовая, но популярность ее непрерывно растет. Если на вашей Web-странице уже есть множество картинок, графические рекламные объявления смотрятся особенно привлекательно — например, их часто можно видеть рядом с новостной статьей.

- *Мультимедийные рекламные объявления.* Такие объявления показывают небольшие флэш-анимации или видеоролики. Они используют вежливую модель, именуемую click-to-play (нажми-для-воспроизведения), т. е. аудио- и видеоданные не начнут воспроизводиться, пока посетитель не щелкнет мышью крошечную кнопку воспроизведения (он также может управлять громкостью с помощью расположенного рядом бегунка). Если ваш гость не нажмет кнопку **Play** (Воспроизвести) для запуска видеоролика, мультимедийная реклама будет выглядеть как обычное графическое объявление.

- *Блок ссылок.* Это сжатый набор ссылок без описания. У этого блока есть заголовок "Ads by Google" и ссылки, состоящие из одно-, двухсловных элементов, например, "Digital Cameras" (Цифровые фотоаппараты) или "Consumer Electronics" (Бытовая электроника). Если посетитель щелкает ссылку кнопкой мыши, Google отображает новую страницу, заполненную текстовыми рекламными объявлениями о товарах этой категории, и вы получаете деньги.

Прежде чем создать корректный рекламный блок, необходимо придумать, где разместить рекламу. Вы хотите расположить вертикальный или горизонтальный баннеры рекламных объявлений, и какой ширины и длины они должны быть? На рис. 14.5 показано лишь несколько возможных вариантов макета.

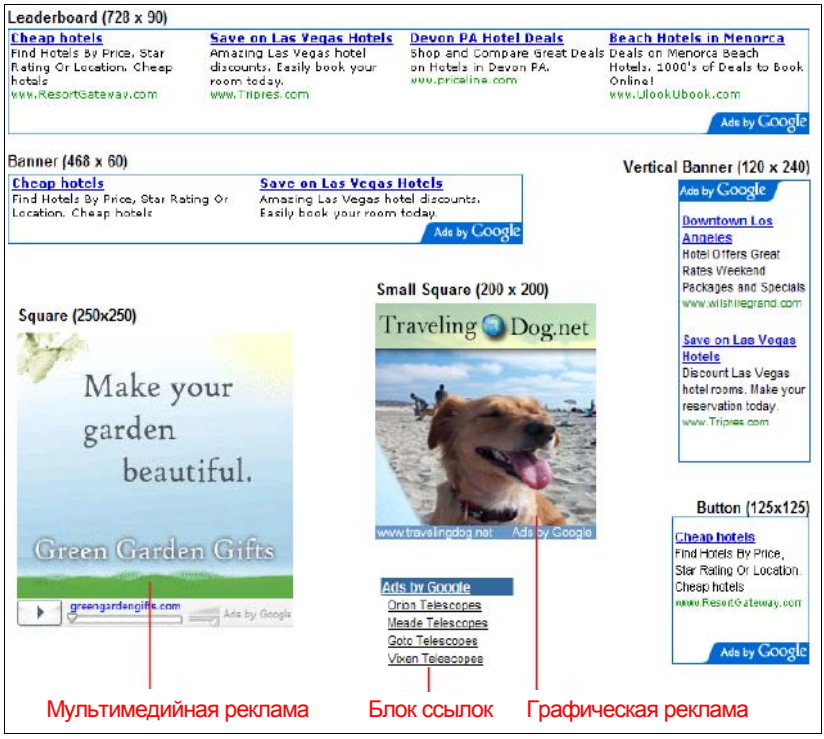


Рис. 14.5. Рекламные объявления всех видов представлены с множеством контуров и размеров, начиная с квадратов и заканчивая вертикальными и горизонтальными баннерами. На рисунке показаны одно мультимедийное рекламное объявление, одно графическое, один блок ссылок, а все остальные объявления — текстовые

Когда будете готовы "нырнуть" и создать свой первый рекламный блок, сделайте следующее.

1. Щелкните кнопкой мыши вкладку **My ads** (Мои объявления) в верхней части страницы.

   На экране появится область **My ads**, в которой перечислены все созданные вами объявления. В данный момент этот список пуст.

2. Щелкните мышью кнопку **New ad unit** (Новый рекламный блок), которая выводится под заголовком **Content > Ad units** (Контент > Рекламные блоки).

   Пришло время ввести данные вашего рекламного объявления.

3. Введите название вашего рекламного блока.

   Это название не будет выводиться в рекламе, но оно существенно облегчит вашу жизнь в процессе управления рекламными объявлениями. После создания рекламного блока вы сможете обращаться к нему по имени (например, Travel Page Ads) и изменять его. Это избавит вас от необходимости повторного построения с нуля вашего рекламного блока.

4. Выберите из списка **Size** (Размер объявления) размер вашего рекламного блока (рис. 14.6).

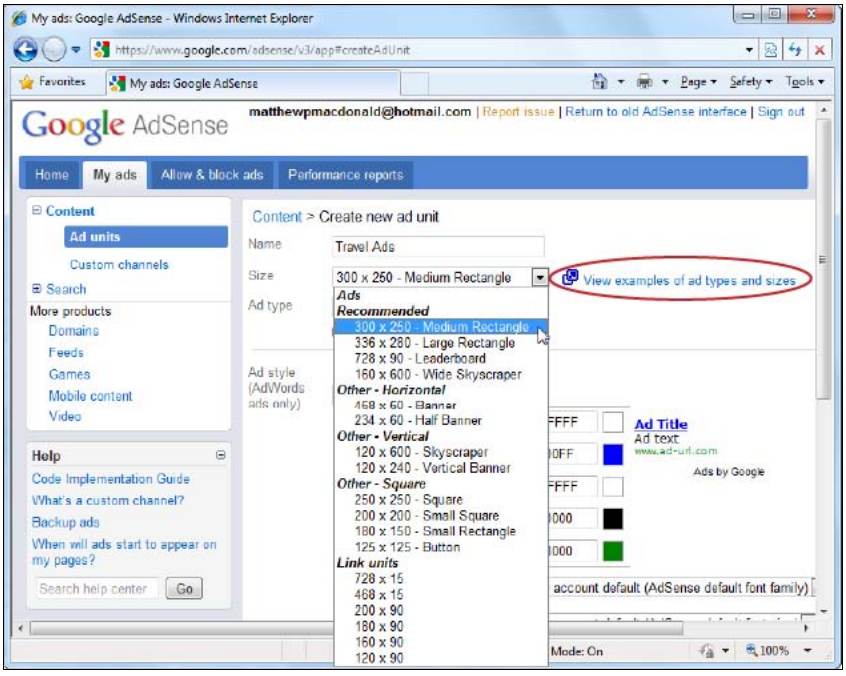


Рис. 14.6. У программы Google AdSense есть варианты макетов практически для любого Web-сайта. В данном примере используется прямоугольник размером 300×250 пикселов, содержащий четыре рекламных объявления, одно над другим. Конечно, единственный способ узнать, сколько объявлений поместится в макете, — щелкнуть кнопкой мыши команду **View examples of ad types and sizes** (Просмотреть примеры типов и размеров объявлений) (обведена) и увидеть примеры Google

Google всегда помещает пакеты рекламных объявлений AdSense в блоки. Блок может включать от одной до четырех реклам. Выбранный вами формат определяет, получите вы вертикальный столбец объявлений или горизонтальный баннер. Он также задает количество одновременно отображаемых объявлений (от одного до пяти).

Совет

Трудно представить себе, как будут выглядеть разные макеты рекламных объявлений, поэтому Google приходит на помощь. Щелкните кнопкой мыши ссылку **View examples of ad types and sizes** (Просмотреть примеры типов и размеров объявлений) и откроется окно, отображающее пример каждого типа рекламного блока. (Некоторые из них показаны на рис. 14.5.) С помощью этой страницы вы сможете подобрать формат, наиболее подходящий для вашего сайта.

5. В списке **Ad type** (Тип объявления) выберите тип рекламного объявления.

   Именно здесь вы решаете, остановиться только на текстовых объявлениях, разрешить графические и мультимедийные объявления или настаивать только на графических и мультимедийных. Обычно графические и мультимедийные объявления выразительней текстовых. Но необходимо уравновесить две цели: желание заработать деньги, увеличивая количество щелчков мыши за счет более привлекательных рекламных объявлений, и стремление минимизировать рас-

сеивание внимания посетителей с помощью менее назойливых рекламных объявлений.

6. Настройте (при желании) цвета, рамку и шрифты ваших рекламных объявлений.

   Цветовая палитра определяет цвета, применяемые Google для текста рекламного блока, фона и рамки (рис. 14.7). В списке **Color palette** (Палитры) можно выбрать одну из готовых палитр Google, например Open Air и Blue Mix. Если же вы хотите быть уверены в том, что цвета наилучшим образом соответствуют цветам вашего сайта, подберите их самостоятельно. Например, для того чтобы рамка и фон рекламного блока сочетались с фоном вашей страницы, измените эти цвета, откорректировав их числовые значения в полях, расположенных под списком цветовых палитр.

   Аналогичным образом можно выбрать гарнитуру шрифта (из очень ограниченного списка) и размер шрифта. Есть возможность задать прямые или скругленные углы рамки вокруг рекламного блока. (Если хотите отказаться от рамки, выберите белый цвет в качестве цвета рамки.)

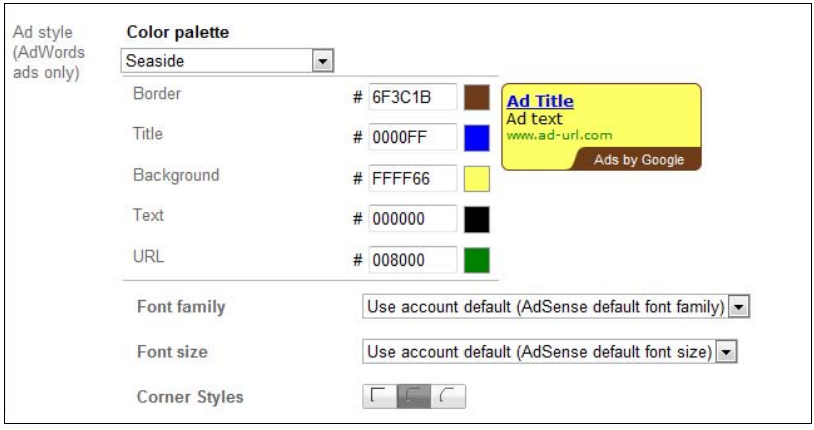


Рис. 14.7. Во время изменения цветов, шрифта и рамки Google отображает полученные результаты в формате единичного рекламного объявления

> ***Совет***
>
> Обычно цвета фона и текста выбираются соответствующими цветам, которые уже использованы на Web-странице. Если вы не меняете стандартные цвета, у вашего рекламного блока будет тот же цвет фона, что и на вашей Web-странице (поскольку он прозрачный) и не будет рамки. Способы подбора пользовательских HTML-цветов см. в *разд. "Задание цвета" главы 6*.

7. Выберите канал (необязательно).

   Если вы генерируете полдюжины рекламных объявлений и затем разбрасываете их на разные страницы вашего сайта, вы не сможете узнать, какие из них приносят вам деньги. Отчет о продаже рекламы, формируемый Google, покажет только общее число щелчков мышью на всех страницах сайта. Многим владельцам

сайтов хочется иметь более подробные данные о том, как работают конкретные рекламные объявления. Воспользуйтесь *каналами* Google.

Для отслеживания эффективности разных рекламных объявлений вы помещаете каждое из них в отдельный виртуальный "канал". Google позволяет создавать отчеты, сравнивающие все каналы, поэтому вы сможете узнать, какие рекламные объявления самые эффективные.

Для создания нового канала щелкните кнопкой мыши ссылку **Create new custom channel** (Создать новый клиентский канал), которая раскроет окно, запрашивающее название канала.

После того как канал создан, его можно использовать для рекламного объявления, щелкнув мышью крошечную ссылку **add** (Добавить) рядом с названием канала (рис. 14.8). (Для прекращения отслеживания эффективности объявления щелкните кнопкой мыши ссылку **remove** (Отключить) рядом с тем же названием канала.)

Совет

Каналы — чудесный инструмент для опробования различных рекламных стратегий и возможность увидеть, какие формат и местоположение рекламного объявления приносят наибольший успех, выжимая из ваших посетителей щелчки мышью. В один и тот же канал можно вставить несколько объявлений и отслеживать их как группу или создать отдельные каналы для каждого из них.

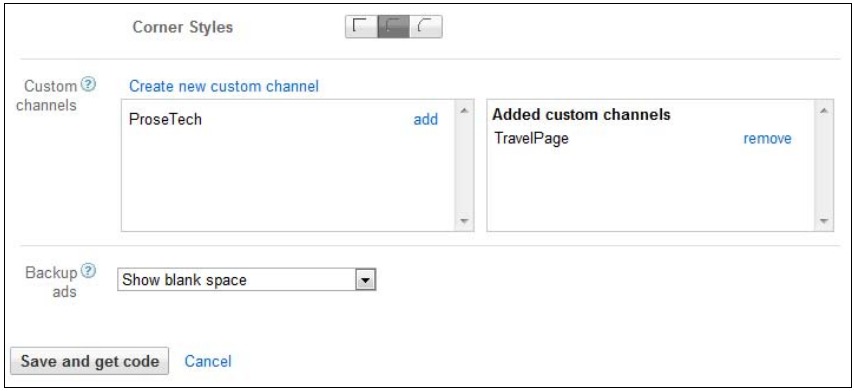


Рис. 14.8. У этой учетной записи AdSense есть два канала. Текущий блок рекламных объявлений (одно из них как раз создается) использует канал TravelPage

8. Выберите резервное рекламное объявление.

Когда вы впервые помещаете на страницу рекламный блок, Google еще не знает, какие объявления соответствуют вашему контенту, поэтому он временно использует замещающий контент. Обычно Google использует рекламные объявления общественных служб — сообщения некоммерческих организаций, например, ЮНИСЕФ. Спустя короткое время на вашу страницу заглядывает проверяющее текст программное обеспечение Google, и в течение пары дней реальные рекламные объявления материализуются.

Если вы не хотите давать приют типовым рекламным объявлениям, можно отобразить замещающий контент.

Существует две возможности: можно выбрать сплошной цвет, и в этом случае Google заполнит рекламный блок только этим цветом, не включая никакого рекламного содержимого. Суть в том, что вы применяете цвет, совпадающий с цветом фона вашей страницы, и таким образом "реклама" полностью скрывается. Вторая возможность — задать URL страницы, которую хотите использовать. До тех пор пока не будут готовы настоящие рекламные объявления, на вашей странице будет отображаться этот контент.

> ***Совет***
>
> Быть может, замещающие рекламные объявления не стоят хлопот. Лучше применить типовые объявления, т. к. макет у них такой же, как у псевдорекламы из сплошного цвета, что облегчает выбор подходящего местоположения и дает возможность увидеть, как рекламное объявление смотрится вместе с остальным контентом вашей страницы.

9. Щелкните мышью кнопку **Save and get code** (Сохранить и получить код).

После этого на экране появится окно с кодом на языке JavaScript для вашей рекламы (рис. 14.9).

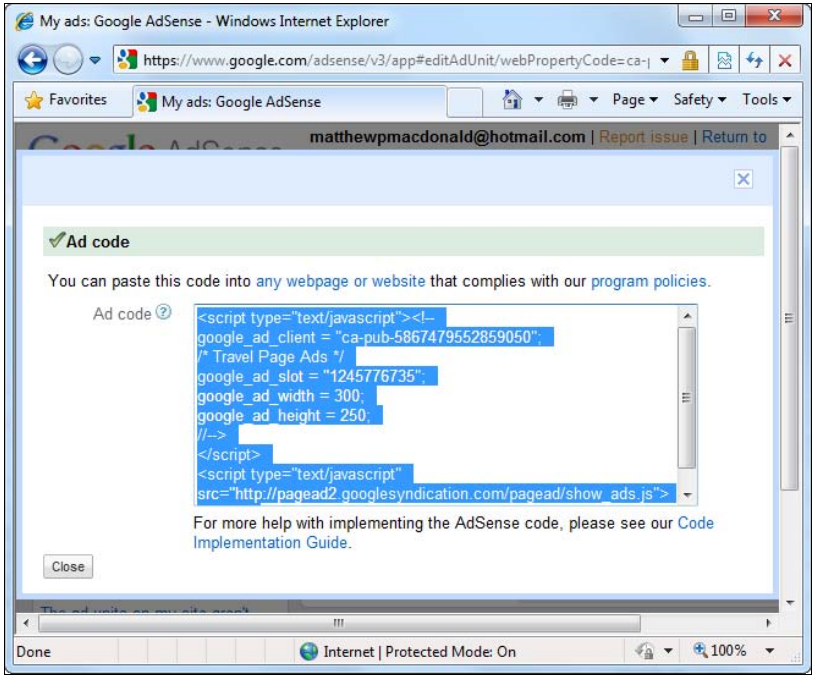


Рис. 14.9. Рекламное объявление AdSense представляет собой не набор HTML-элементов, а код на языке JavaScript (о котором вы узнаете в *главе 15*). Google применяет сценарий, т. к. рекламные блоки должны создаваться динамически, в соответствии с выбранными вами предпочтениями. Когда посетитель запрашивает вашу страницу, выполняется сценарий рекламного блока, он связывается с Web-серверами Google и запрашивает набор рекламных объявлений. Сервер ищет подходящие объявления, применяет ваш макет и цветовые параметры и отправляет окончательный HTML-блок назад в поле сценария, так что его можно вставить на вашу страницу

10. Скопируйте код вашего рекламного блока и щелкните мышью **Close** (Закрыть).
11. Щелкните кнопкой мыши в текстовом поле для выделения кода вашего рекламного блока, щелкните выделенный текст правой кнопкой мыши и выберите команду **Copy** (Копировать). Теперь можно вставить код на одну или несколько Web-страниц, как описано в следующем разделе.

Примечание

Если позже вам понадобится изменить ваш рекламный блок, снова зарегистрируйтесь в Google AdSense, щелкните кнопкой мыши вкладку **My ads** (Мои объявления) и найдите ваш блок в списке. Когда вы щелкните его кнопкой мыши, Google отобразит подробные сведения о нем и позволит их отредактировать. Большинство изменений — скажем, другая цветовая схема рекламного блока — отображаются немедленно. Но если вы захотите изменить формат вашего рекламного блока, необходимо получить новый программный код и вставить его на ваши страницы, т. к. размер вашего блока изменится.

Размещение рекламных объявлений на ваших Web-страницах

После того как Google создаст сценарий для вашего рекламного блока, его можно вставлять на вашу Web-страницу. Легче всего разместить горизонтальные баннеры: вы просто вставляете весь сценарий целиком в то место, где он должен отображаться.

Далее приведен пример, в котором рекламные объявления помещены внизу страницы.

```
<!DOCTYPE html>

<html>
<head>...</head>
<body>
  <h1>A Trip to Remember</h1>
  <p><img src="me.jpg" alt="Me" class="floatLeft" />
  After returning from my three-month travel adventure ...</p>
  <p>I hope you enjoy these pictures as much as I do.</p>
  <p>See pictures from ...</p>

  <script type="text/javascript"><!--
  google_ad_client = "ca-pub-5867479552859050";
  google_ad_slot = "1245776735";
  google_ad_width = 300;
  google_ad_height = 250;
  //-->
  </script>
  <script type="text/javascript"
  src="http://pagead2.googlesyndication.com/pagead/show_ads.js">
/script>
```

```
</body>
</html>
```

На рис. 14.10 показан результат.

Для размещения вертикальных баннеров объявлений потребуется немного больше работы, но сделать это будет легко, когда узнаете одну хитрость. Проблема заключается в выводе остального контента вашей страницы *рядом* с вертикальным рекламным блоком. Как вы узнали в *главе 9*, для смещения объявлений к боковому краю страницы можно применить правила таблицы стилей.

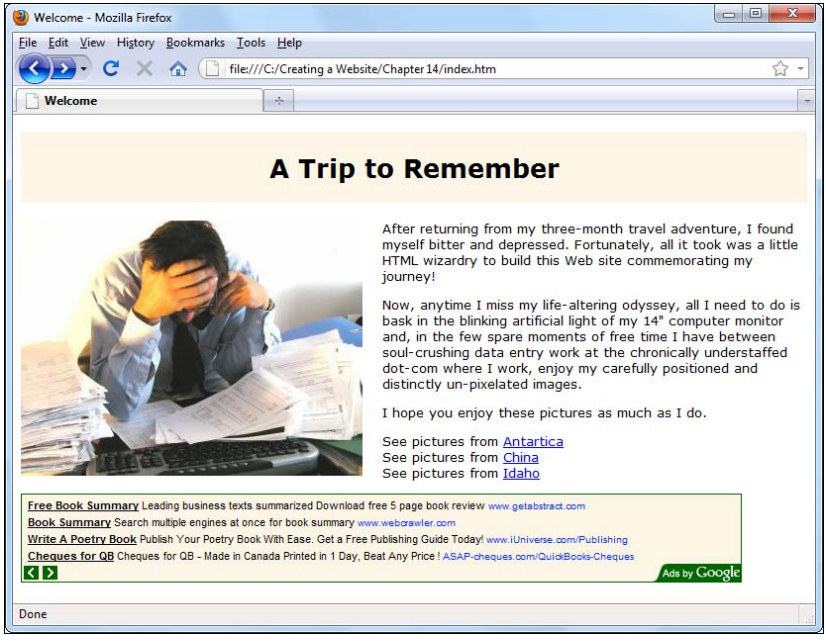


Рис. 14.10. На этой странице в горизонтальном баннере размером 728×90 пикселов отображается четыре рекламных объявления. Обратите внимание на то, что фоновый цвет рекламного блока совпадает с фоном заголовка и все благодаря выбору пользовательских цветов (см. рис. 14.7)

Для применения технологии таблиц стилей начните с заключения сценария вашего рекламного блока в элемент `<div>`. Далее приведен пример, содержащий контент, который показан на рис. 14.10.

```
<!DOCTYPE html>

<html>
<head>...</head>
<body>
  <div class="floatRight">
    <h1>A Trip to Remember</h1>

    <script type="text/javascript"><!--
     google_ad_client = "pub-5876479552859050";
```

```
     google_ad_slot = "4493177655";
     google_ad_width = 120;
     google_ad_height = 240;
     //-->
    </script>
    <script type="text/javascript"
     src="http://pagead2.googlesyndication.com/pagead/show_ads.js">
    </script>
  </div>
    <p><img src="me.jpg" alt="Me" class="floatLeft" />
     After returning from my three-month travel adventure ...</p>
    <p>I hope you enjoy these pictures as much as I do.</p>
    <p>See pictures from ...</p>
</body>
</html>
```

В элементе `<div>`, не имеющем собственного форматирования, используется класс таблицы стилей `floatRight`. В вашей таблице стилей применяется правило, приведенное ниже, чтобы заставить раздел `<div>` смещаться с помощью атрибута `float` (см. *разд. "Обтекание текста вокруг изображения" главы 7*).

```
.floatRight {
  float: right;
  margin-left: 20px;
}
```

На рис. 14.11 показан результат.

НА ПРОФЕССИОНАЛЬНОМ УРОВНЕ

Как AdSense создает контекстно-зависимые рекламные объявления

Каждый раз, когда открывается Web-страница, содержащая рекламу Google, сценарий AdSense посылает сообщение Web-серверу Google, запрашивая рекламные объявления. В сообщение включается информация о ваших предпочтениях, касающихся рекламы, и ваш уникальный идентификатор клиента (Ваш клиентский ID, который выглядит примерно так: pub-5876479552359052, можно найти в коде сценария.)

Когда сервис Google впервые получает такой запрос, он понимает, что еще не просматривал вашу страницу и не знает, какие типы объявлений больше всего для нее подходят. Поэтому он отправляет вам блок типовых рекламных объявлений (или ваш замещающий контент, если вы его задали, как описано ранее в *разд. "Создание рекламного объявления"*). Сервис Google также добавляет вашу страницу в список страниц, которые должен посетить. Иногда в течение следующих двух дней программа-робот Google направляется на ваш сайт и анализирует его контент. Начиная с этого момента, вы видите рекламные объявления, контекстно-связанные с содержимым вашей страницы.

Если прошло 48 часов, а вы все еще не видите интересных для вашей аудитории рекламных объявлений, возможно, есть проблемы. Одна из самых распространенных ошибок — размещение объявлений на страницах со слабым текстовым наполнением, в этом случае Google не может выяснить, чему на самом деле посвящен ваш сайт. Помните, что Google AdSense, когда проверяет ваш сайт, рассматривает единственную страницу, ту на которой рекламный блок. Вы можете столкнуться и с другой потенциальной проблемой, если поместите ваше рекламное объявление на недоступ-

ную страницу. Например, программа-робот Google не может попасть ни на одну страницу, которая не выгружена в Интернет, страницы на вашем персональном компьютере или в локальной сети недоступны для нее, точно так же, как и страницы, защищенные паролем. Некоторые Web-сайты блокируют программы-роботы с помощью правил исключений (см. *разд. "Как спрятаться от поисковых машин" главы 11*), что полностью останавливает робота Google.

В заключение напомню о том, что сервис Google может создать рекламные объявления, не подходящие для вашего сайта. Например, вы можете обсуждать достоинства и недостатки языка программирования Python (т. е. питон) и Google может откликнуться рекламным объявлением из зоомагазина. Часто такого сорта проблемы Google обнаруживает самостоятельно, анализируя ваши страницы и убирая объявления, не порождающие большое количество щелчков мышью. Но вы можете помочь сервису отказаться от неподходящих вариантов с помощью средства блокирования объявлений. Для того чтобы начать, зарегистрируйтесь и щелкните кнопкой мыши вкладку **Allow & block ads** (Разрешить/блокировать объявления).

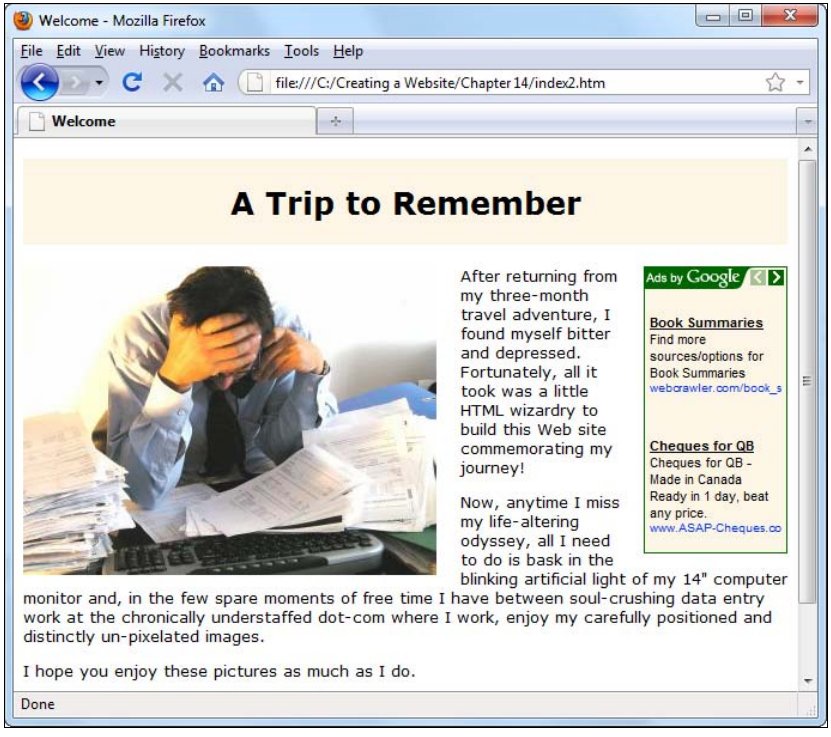


Рис. 14.11. Вертикальный баннер размером 120×240 пикселов вмещает два рекламных объявления на этой странице. Вертикально расположенные объявления — самый популярный формат отображения рекламы Google. Не только, потому что они аккуратно ложатся рядом с контентом Web-страницы, но и потому, что при достаточной протяженности остаются видимыми даже, если посетители прокручивают страницу

Улучшенный поиск Google

Google предоставляет еще одну возможность порадовать ваших посетителей (и попутно заработать вам немного денег). Вы можете добавить поле поиска на ваши

страницы, позволив посетителям запускать поисковые запросы Google с вашего сайта. Более того, вы получите деньги за любые объявления, которые посетители щелкнули мышью в результатах поиска — средство, которое в Google назвали довольно прозаично *AdSense for Search* (AdSense для поиска).

Из вашей учетной записи AdSense вы легко сможете вставить поле или окно поиска Google на ваш сайт.

1. Войдите в систему с вашей учетной записью AdSense и щелкните кнопкой мыши вкладку **My ads** (Мои объявления).

   Это то же самое место, где вы создаете рекламные блоки, но для создания окна поиска следует перейти в другую область вкладки (рис. 14.12).

2. На левой панели щелкните кнопкой мыши ссылку **Search** (Поиск), чтобы увидеть подкатегории. Щелкните мышью ссылку **Custom search engines** (Системы пользовательского поиска), а затем кнопку **New custom search engine** (Создание новой системы пользовательского поиска).

   Теперь можно ввести данные для вашего окна поиска.

3. Назовите вашу поисковую систему.

   Название позволит вам выводить свое окно пользовательского поиска, настраивать параметры и получить новый программный код, не начиная все с нуля.

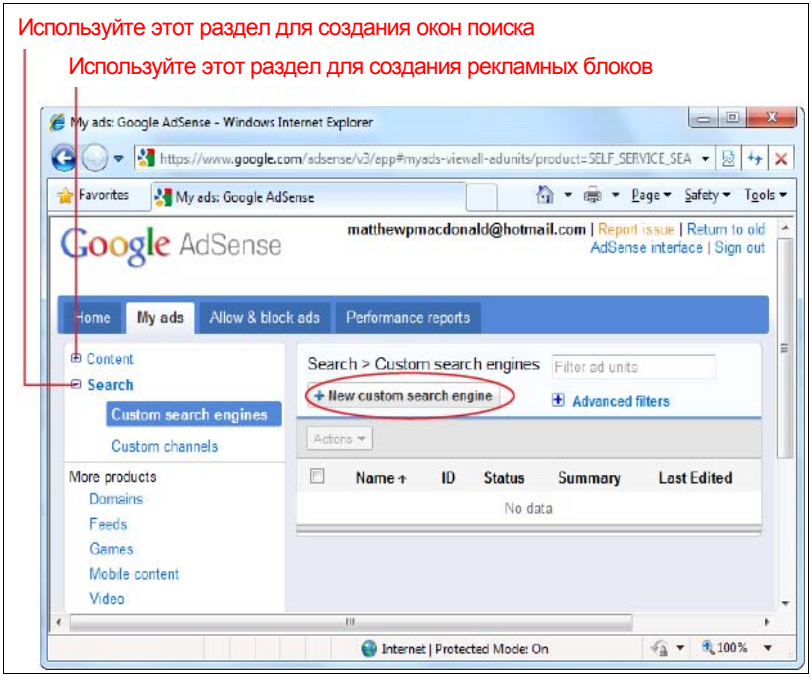


Рис. 14.12. Пока данный участник AdSense не создал никаких окон поиска. Щелчок мышью кнопки **New custom search engine** (Создание новой системы пользовательского поиска) создаст первое окно поиска

4. Выберите область поиска.

   Для формирования окна поиска, использующего хорошо знакомую и любимую нами поисковую машину Google, выберите вариант **The entire Web** (Вся сеть).

   Для ограничения области поиска определенным набором сайтов выберите **Only sites I select** (Только выбранные мной сайты).

5. Если вы выбрали ограниченный поиск, введите в поле **Selected sites** (Выбранные сайты) список сайтов, участвующих в поиске (рис. 14.13).

   Вы можете вводить отдельные страницы (например, **http://www.ChocolateMysteries.com/recipe01.html**) или использовать символ звездочки для задания всех страниц в папке или на сайте (например, **http://www.ChocolateMysteries.com/***), как более распространенный вариант. Если нужно ввести несколько URL, помещайте их в поле на отдельные строки.

Примечание

Даже если вы ограничиваете поиск пределами вашего Web-сайта, Google все равно запрашивает свой стандарт, централизованный каталог Web-страниц, просто отображаемые результаты поиска он ограничивает страницами только вашего сайта. Если в каталоге Google нет ваших страниц (потому что вы только что создали сайт или потому что Google не знает о существовании вашего сайта), эти страницы никогда не попадут в результаты поиска, независимо от настроек вашего окна поиска. Информацию о том, как заставить Google узнать о вас, см. в *разд. "Регистрация с помощью поисковых машин" главы 11*.

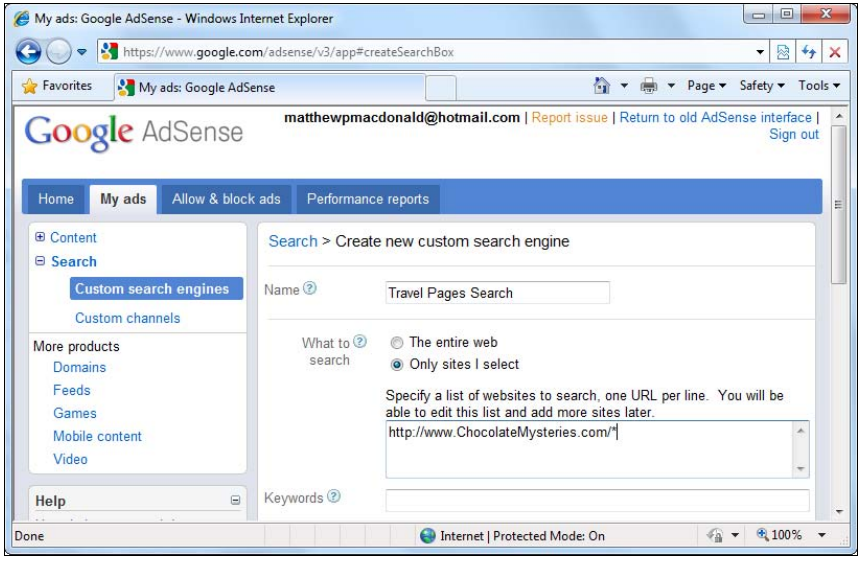


Рис. 14.13. Обычно, когда выбирается вариант **Only sites I select** (Только выбранные мной сайты), вы вводите в поле **Selected sites** (Выбранные сайты) единственный сайт — ваш. В этом случае посетители могут искать одну из ваших страниц без соблазнов со стороны других сайтов в Web. Например, если у вас десятки страниц с путевыми заметками, посетители смогут обосноваться на странице, которую они выбрали благодаря поиску "смешная история о резиновой курочке в Перу"

6. Заполните поле **Keywords** (Ключевые слова) (необязательно).

   Google автоматически добавляет в строку поиска ваших посетителей любые ключевые слова, которые вы вставляете сюда. Это дает вам возможность создать окно поиска, нацеленное на контент определенного типа. Например, если ваш сайт весь посвящен гольфу, можно вставить ключевое слово гольф. В этом случае, если пользователь ищет "тигр", поиск вернет страницы о Тайгере Вудсе (Tiger Woods)[1], а не об африканской саванне.

7. Установите флажок **SafeSearch** (Использовать безопасный поиск), если хотите.

   SafeSearch отфильтровывает сквернословие и содержимое сексуального характера из результатов поиска. Этот фильтр может оказаться полезным в двух ситуациях. Во-первых, он строго обязателен для сайтов, предоставляющих информацию для детей. Во-вторых, он удобен, когда ваш Web-сайт вращается в тематике, использующей некоторые ключевые слова, характерные для Web-сайтов только для взрослых. Например, страница о раке груди — вы не захотите, чтобы кто-то ввел в ваше окно поиска "обследование груди" и получил неподходящие материалы.

8. Отрегулируйте настройки страны и языка, если есть подходящие.

   К ним относятся язык вашего Web-сайта и ваше географическое местоположение. Как вы, возможно, знаете, у Google есть страницы, ориентированные на конкретную страну, что позволяет настраивать результаты поиска, предоставляя их на разных языках или с предоставлением приоритета местным сайтам.

9. При желании щелкните значок (+) рядом со списком **Custom channels** (Клиентские каналы) и выберите клиентский канал.

   Вы можете поместить свое окно поиска в конкретный канал, так же как в случае рекламного блока, что может помочь отслеживать доллары, заработанные от рекламы из окна поиска. Дополнительную информацию о каналах см. ранее в *разд. "Создание рекламного объявления" данной главы*.

10. Щелкните кнопкой мыши значок (+) рядом с разделом **Search box look and feel** (Внешний вид окна поиска), так вы сможете настроить внешний вид окна поиска Google (необязательно).

    Здесь всего несколько настроек. Можно изменить размер окна поиска и местоположение логотипа Google. Вы также сможете выбрать белый, черный или темно-серый фон.

11. При желании щелкните кнопкой мыши значок (+) рядом с разделом **Search results and ad location** (Стиль объявлений и результатов поиска), чтобы настроить внешний вид страницы результатов поиска, отображаемой Google.

    С помощью этих настроек можно задать цвета, используемые Google при выводе результатов поиска. В этом случае можно добиться их соответствия осталь-

[1] Тайгер Вудс — американский гольфист.

ным страницам вашего сайта. Предлагаемый инструмент очень похож на цветовые палитры для рекламных блоков AdSense (см. рис. 14.7).

Вы также можете выбрать, где отображать результаты поиска.

- Выберите вариант **on a Google page in the same window** (на странице Google в том же окне), чтобы заменить текущую страницу стандартной страницей Google с результатами поиска.
- Выберите **on a Google page in a new window** (на странице Google в новом окне), чтобы автоматически открывалось новое окно браузера с результатами поиска. Автоматически открывающиеся окна обычно раздражают Web-посетителей, но этот прием удобен, если вы хотите, чтобы ваш Web-сайт остался на Рабочем столе вашего посетителя.
- Выберите **on my website using an iframe** (на моем Web-сайте с использованием фреймов), чтобы оставить посетителей на своем Web-сайте и показать результаты поиска рядом с вашим контентом. Этот вариант нравится всем, но он потребует немного больше усилий, т. к. вам придется создать две страницы: одну с окном поиска, а другую с результатами поиска. Google предоставит разметку для размещения на каждой из этих страниц.

  Если вы хотите применить последний вариант, необходимо предоставить URL страницы результатов поиска (например, **http://www.Chocolate Mysteries.com/searchresults.html**). Не беспокойтесь, если эта страница еще не создана — вы сможете сформировать ее и выгрузить, когда закончите создавать окно поиска. Вы также должны сообщить Google, какой ширины будут результаты поиска. Стандартный вариант — 800 пикселов — хорошо подходит, если вы не планируете занимать боковые области страницы дополнительным контентом (например, рекламными объявлениями или полосой меню). Если у вас есть другой контент, который вы хотите поместить на этой странице, можно расположить его над результатами поиска.

12. Установите флажок **I have read and agree to the Google Custom Search Terms of Service** (Я прочитал(а) и принимаю Условия использования Системы пользовательского поиска Google) и щелкните мышью кнопку **Save and get code** (Сохранить и получить код).

    На последней странице установки появится разметка для вашего законченного окна пользовательского поиска (рис. 14.14) и элемент `<form>` (дополнительную информацию о формах в языке HTML см. в *разд. "Интерактивные формы" главы 15*). Как и программный код AdSense, вы можете вставить этот HTML-код на любую Web-страницу.

    Если вы решили создать страницу результатов поиска на вашем собственном сайте, вы получите второе поле с разметкой, необходимой для создания этой страницы. Она включает элемент `<div>` и код на языке JavaScript. Он действует во многом так же, как код, полученный в AdSense. Вы просто вставляете элемент `<div>` туда, где хотите разместить результаты поиска.

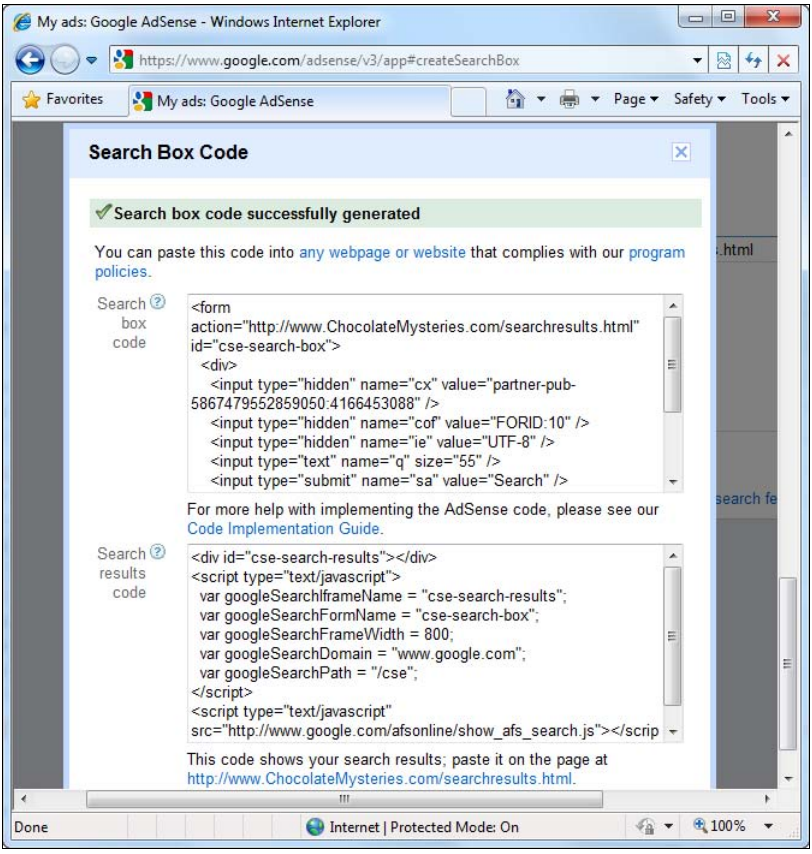


Рис. 14.14. Этот пример содержит разметку для окна пользовательского поиска (вверху) и программный код для страницы результатов поиска (внизу). Как бы ни было велико желание, не меняйте разметку окна поиска, т. к. Google строго пресекает подобную практику

Amazon Associates

Как бы ни были популярны среди владельцев Web-сайтов рекламные объявления, у них есть один серьезный недостаток: они загромождают ваши страницы. После того, как вы отшлифовали дизайн с помощью тщательно подобранных изображений и таблиц стилей, возможно, вам не захочется вставлять чье-то рекламное объявление. И хотя реклама от Google визуально не так назойлива, как любая другая, например, анимационные баннеры или всплывающие окна, она все равно проглатывает значительную часть экранного пространства. Если вы не в силах нанести вред своим гениальным Web-страницам, может вас заинтересует более мягкий вариант.

Amazon Associates — партнерская программа, Web-долгожитель. Если у вас есть персональный сайт со страницей "любимые книги" или вы предпочитаете "стрелять" книги там и сям, вы сможете заработать немного денег, зарегистрировавшись как партнер.

Ключевая идея программы Amazon Associates заключается в том, что вы предоставляете ссылки на страницы с книгой или другим товаром, предлагаемым на Web-сайте Amazon. Например, если вы пишете аннотацию на величайший рецепт, можно вставить ссылку на страницу Amazon, на которой продается цитируемая вами кулинарная книга. Ссылка сама по себе — замечательный инструмент для вашего сайта, т. к. она снабжает посетителей дополнительной полезной информацией. Но самое замечательное происходит, когда посетитель решает купить эту книгу. Вы в итоге получаете твердые комиссионные в размере 4% от продажной цены книги.

Примечание

Amazon платит комиссионные не только за книги. Вы можете предоставлять ссылки на огромное множество товаров, продающихся на Amazon (за исключением товаров, которые продают другие розничные продавцы, например, Target и Office Depot). Правда, программа Amazon ограничивает размер доходов, которые вы можете получить от продажи не книг. Например, для персональных компьютеров максимальные комиссионные — 25 долларов за штуку. Вы также можете заработать более 4% на товарах некоторых категорий (например, MP3-скачиваниях) или, если продадите определенное количество товаров за месяц. Эти правила меняются время от времени, поэтому периодически тщательно прочесывайте Web-сайт Amazon Associates, чтобы получить актуальную информацию.

Регистрация в качестве партнера

Зарегистрироваться в программе Amazon Associates даже легче, чем присоединиться к AdSense.

Выполните следующие действия.

1. Перейдите на Web-сайт **http://affiliate-program.amazon.com**, щелкните мышью кнопку **Join now** (Присоединиться сейчас) и зарегистрируйтесь, указав ваш адрес электронной почты и пароль в Amazon.

   Для присоединения к партнерской программе Amazon вам нужна учетная запись Amazon. Если у вас ее нет, для создания учетной записи щелкните мышью кнопку **I am a new customer** (Я новый клиент).

2. Введите личные данные и сведения о своем Web-сайте.

   В личных данных надо указать имя, адрес и номер телефона. Для Web-сайта следует сообщить его имя, URL и краткое описание (рис. 14.15).

3. Щелкните мышью кнопку **Finish** (Готово) для отправки вашей заявки.

   Очень скоро вы получите подтверждающее сообщение о том, что вы приняты с испытательным сроком. В это письмо включен ваш уникальный идентификатор партнера. Этот номер важен, т. к. это единственная нужная вам информация для того, чтобы начать добавлять ссылки Amazon и зарабатывать комиссионные. Теперь вы можете применять инструменты для партнеров на сайте **http://associates.amazon.com** (см. *следующий раздел*).

   Через пару дней, когда Amazon проверит ваш сайт и подтвердит, что он не находится в конфликте с законом, вы получите второе сообщение, подтверждающее, что вы приняты окончательно.

4. Если вы хотите сообщить Amazon о способе вашей оплаты прямо сейчас, щелкните мышью ссылку **Specify Payment Method Now** (Указать способ оплаты сейчас).

   Вы можете выбрать предпочтительную форму оплаты до того, как Amazon официально примет вас в свою программу. Возможные варианты включают чек, подарочную карту Amazon или прямой перевод на счет в банке США. Amazon не отправляет чек, пока вы не заработаете как минимум 100 долларов, и берет с вас 15 долларов за оформление выплаты. Другие виды оплаты начинают действовать после того, как вы заработали 10 долларов, и не включают никаких сборов.

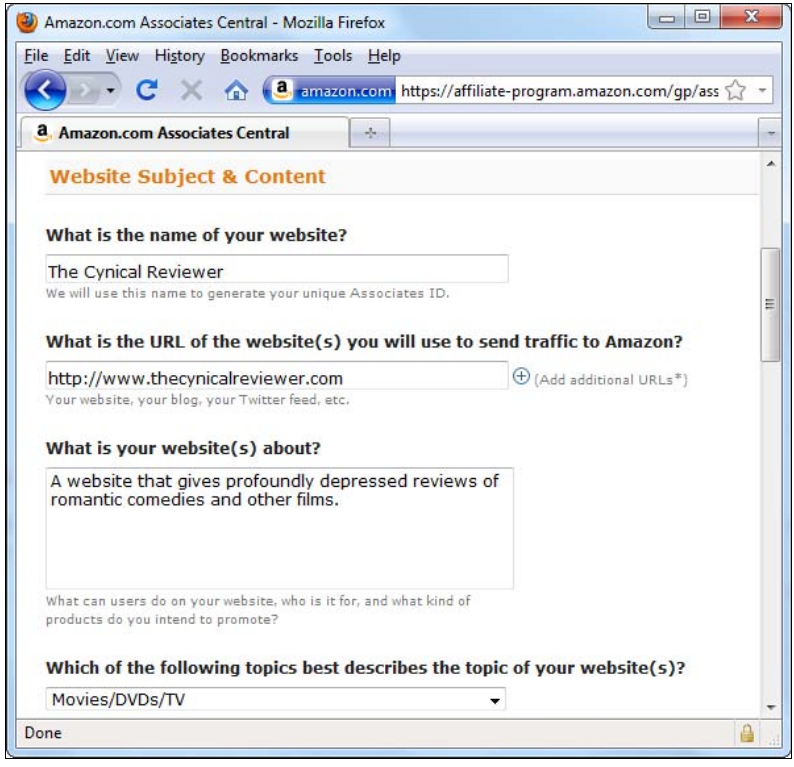


Рис. 14.15. Для того чтобы стать партнером Amazon, необходимо предоставить некоторую базовую информацию о вашем сайте. Не пропускайте этот шаг, т. к. представитель Amazon обязательно посмотрит ваш сайт, прежде чем одобрит ваше участие в программе

Генерация партнерских ссылок

После получения вашего партнерского идентификатора, вы можете создавать *партнерские ссылки*, гиперссылки, отправляющие ваших посетителей на Amazon. Хитрость заключается в особом форматировании URL.

Вы добавляете в самый конец партнерской ссылки ваш идентификатор партнера. Например, в первое письмо, присланное Amazon, вставлен образец партнерской ссылки на его исходную страницу.

Он выглядит следующим образом.

http://www.amazon.com?tag=prosetech-22

В этом примере идентификатор партнера — `prosetech-22`. (Замените его своим идентификатором, чтобы создать ссылку для своего Web-сайта.) Если кто-нибудь последует за этой ссылкой и купит что-то, вы заработаете 4% комиссионных.

Далее показано, как сослаться на страницу Amazon с помощью элемента-якоря.

```
Visit <a href="http://www.amazon.com?tag=prosetech-22">Amazon</a> and
help me save up to buy a Ferrari.
```

> ***СОВЕТ***
>
> Amazon настоятельно советует объявить, что вы являетесь его партнером. Если вам хочется покрасоваться, компания предоставляет коллекцию готовых логотипов и баннеров на странице **http://tinyurl.com/46cqc84**. Вы можете вставить их на свой сайт и даже поместить их в элементы-якори для преобразования в партнерские ссылки (чтобы увидеть страницу с баннерами, нужно быть партнером Amazon).

Ссылки на товары

Вы получите бóльшие комиссионные с помощью ссылок, указывающих на конкретный товар. Amazon предлагает ряд форматов партнерских ссылок и далее приводит пример простейшего из них.

```
http://www.amazon.com/dp/ASIN/?tag=AssociateID
```

Далее показан пример конкретной ссылки.

```
http://www.amazon.com/dp/0141181265/?tag=prosetech-22
```

В этой ссылке вы задаете две детали: ASIN (Amazon Standard Item Number, Стандартный номер товара Amazon) и партнерский идентификатор. ASIN — `0141181265` (указывает на книгу "Поминки по Финнегану"), а идентификатор партнера — `prosetech-22`. На рис. 14.16 показано, где найти ASIN.

Далее приведен пример полной партнерской ссылки.

```
The development of the modern personal computer was first presaged in Joyce's
<a href="http://www.amazon.com/dp/0141181265/?tag=prosetech-22">
Finnegans Wake</a>.
```

Это все, что нужно.

> ***ПРИМЕЧАНИЕ***
>
> Если посетитель Web-сайта перейдет по ссылке к конкретному товару Amazon, но в результате купит что-то совсем другое, все в порядке — вы все равно получите те же 4% комиссионных.

Расширенные ссылки

У Amazon есть специализированные средства, которые помогут вам генерировать ссылки. С их помощью вы можете создать ряд эффектных ссылок.

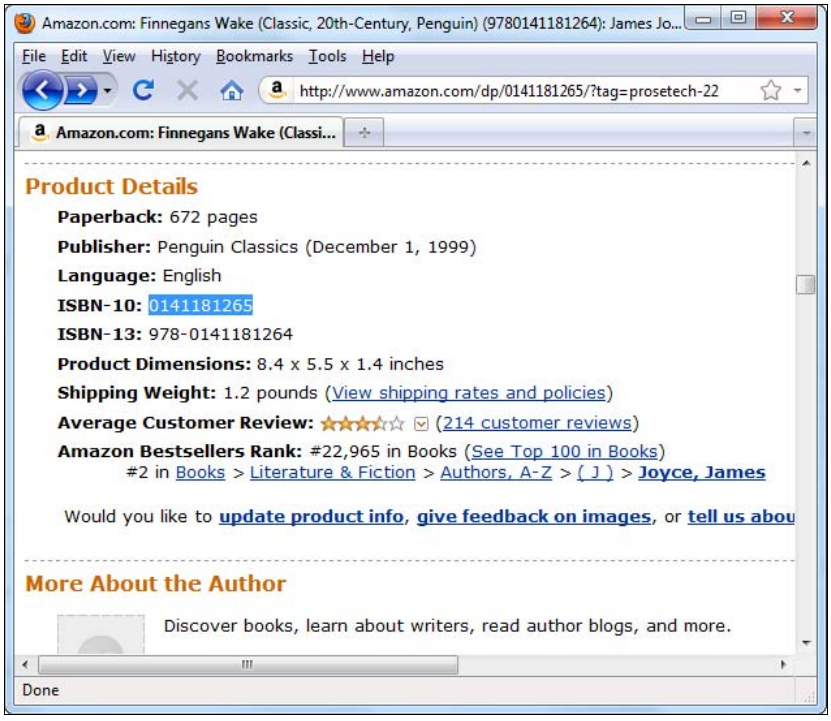


Рис. 14.16. В каталоге Amazon у каждого товара есть уникальный ASIN, который можно найти в разделе **Product Details** (Подробное описание) на странице, посвященной этому товару. У книг ASIN совпадает с номером ISBN из 10 цифр (выделен), стандартным отраслевым идентификационным номером книги

К возможным вариантам относятся:

- ссылки с мини-картинками;
- ссылки на категории товаров (например, журналы о конном спорте или самые ходовые кухонные устройства);
- рекламные баннеры, посвященные конкретным отделам Amazon;
- окна поиска Amazon, позволяющие посетителям выполнять собственные запросы.

Даже если вам не нужны эти улучшенные ссылки (и если ваша жизнь вовсе не посвящена продаже книг), все равно стоит сформировать ссылки средствами, предлагаемыми Amazon, т. к. в их ссылках есть встроенное отслеживание, и вы сможете определить, сколько людей просмотрело каждую ссылку.

Примечание

Отслеживание ссылок Amazon очень умное. По существу оно встраивает крошечное однопиксельное изображение рядом с каждой ссылкой. Если кто-то запрашивает страницу, содержащую одну из таких ссылок, браузер автоматически выводит невидимое изображение из Amazon. Когда Amazon получает запрос на невидимое изображение, он знает, что кто-то щелкнул ссылку, и записывает один *показ* (просмотр страницы) в свою базу отслеживания.

Далее описано, как применять средства построения ссылок от Amazon.

1. Перейдите на сайт **http://affiliate-program.amazon.com** и зарегистрируйтесь.

   Вы попадете на исходную страницу Associates Central (Центр управления партнерами), которая предлагает множество отчетов для проверки процесса продаж за текущий день, а также инструменты для построения ссылок.

   Совет

   Чтобы узнать подробную информацию о самых амбициозных вещах, которые можно делать с помощью Amazon Associates, щелкните мышью кнопку **Get Started Now** (Начать сейчас). Вы также можете получить неоценимый совет от других партнеров, посетив дискуссионные форумы — ссылка **Discussion Boards** находится в нижней части панели меню слева.

2. Щелкните кнопкой мыши вкладку **Links & Banners** (Ссылки и баннеры). Далее в меню слева выберите **Product Links** (Ссылки на товары).

   Amazon позволяет создавать ссылки различных типов. Ссылки на товары указывают на конкретные изделия на сайте Amazon. Они, как правило, самые полезные. Но если вы хотите углубиться в изучение сайта Amazon, исследуйте ссылки всех типов.

3. В поле поиска введите ASIN вашего товара и щелкните мышью кнопку **Go** (Поиск).

   Если не знаете ASIN, выберите самую подходящую по вашему мнению категорию и введите название товара. Когда вы нажмете кнопку **Go**, Amazon выполнит поиск товара и выведет на экран список результатов поиска (рис. 14.17).

4. Щелкните мышью кнопку **Get Link** (Получить ссылку) рядом с названием товара.

   Вы увидите страницу, отображающую товар и примерный вид ссылки, которую вы создаете.

5. Выберите тип ссылки и настройте ее внешний вид.

   Вы можете настроить ссылку на товар, сделав ее настолько привлекательной или простой, насколько захотите (рис. 14.18). Для создания текстовой ссылки выберите тип **Text Only** (Текстовая). В противном случае будет выведено более подробное окно, включающее изображение товара и его цену.

   Можно задать открытие страницы товара, на которую указывает ссылка, в новом окне браузера определить размер изображения, сведения о цене товара и цвета, используемые на странице товара.

   В заключение скопируйте HTML-код из текстового поля Amazon и вставьте его на любую Web-страницу вашего сайта.

После создания текстовой ссылки Amazon сгенерирует элемент-якорь, который кажется довольно сложным. (Как писалось ранее, элемент-якорь содержит невидимый элемент `<img>`, позволяющий Amazon отслеживать количество показов ссылки.)

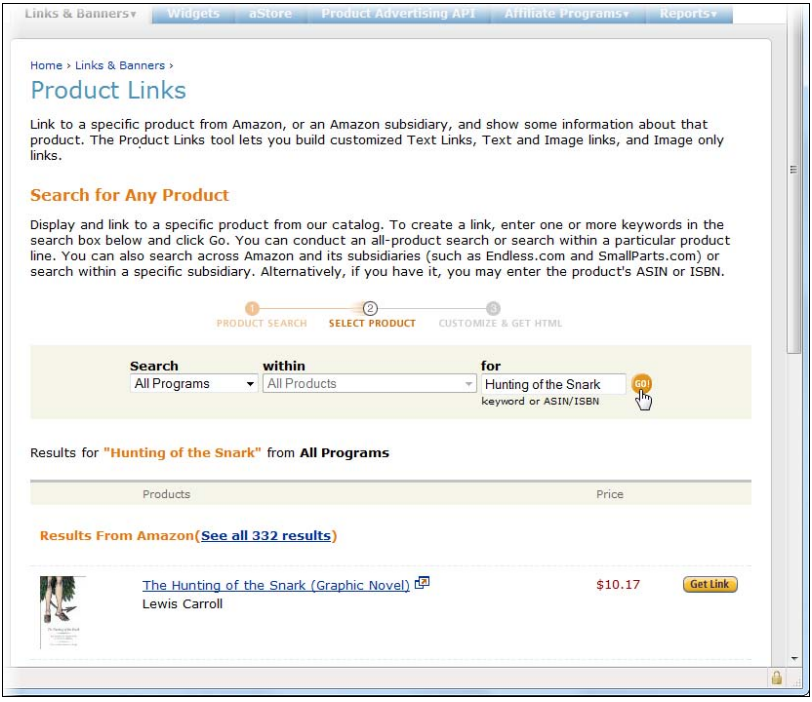


Рис. 14.17. Во время создания ссылки можно выполнить поиск конкретного товара. Этот поиск работает более или менее так же, как поиск Amazon на исходной странице

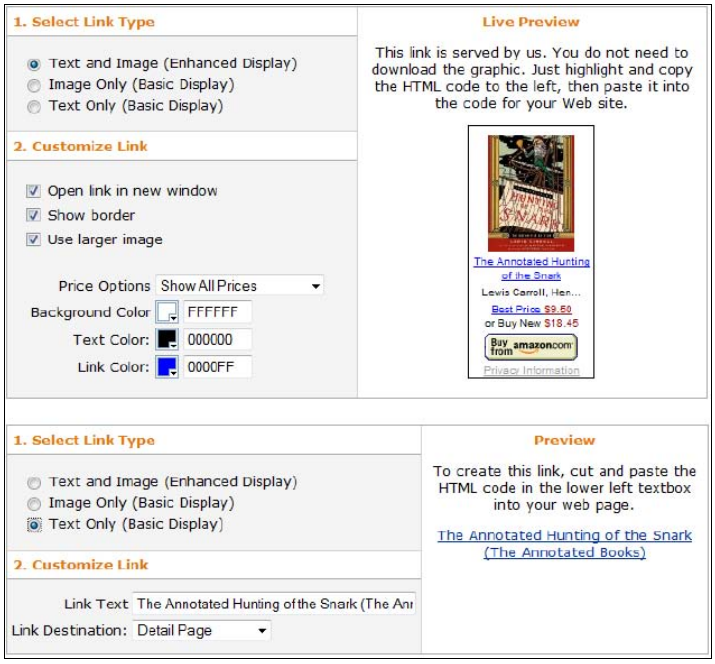


Рис. 14.18. Когда параметры ссылки заданы, Amazon отображает в окне предварительного просмотра полученный результат. Вы можете выбрать ссылку с текстом и изображением (вверху) или более простую, легче встраиваемую текстовую ссылку (внизу)

Но, как и все элементы-якори, этот довольно легко поместить в нужное место разметки. Просто вставьте его в существующий абзац, например, так:

```
<p>Lewis Carroll's work as a mathematician may have driven him insane,
as his famous book
<a href="http://www.amazon.com/exec/obidos/...">The Hunting of the Snark</a>
<img src="http://www.assoc-amazon.com/..." width="1" height="1" />
attests.</p>
```

Примечание

Amazon помещает в элемент-якорь полное название книги. Это сделает вашу ссылку немного длиннее, чем вы ожидали, т. к. в нее может быть включена информация об издании и авторе книги. В этом случае можете сократить ее, оставив только заголовок.

Amazon ежемесячно рассылает электронные письма, сообщая о размерах вашего заработка, но если не хотите ждать, можно зарегистрироваться на сайте Amazon Associates в любое время. Щелкните кнопкой мыши вкладку **Reports** (Отчеты) (рис. 14.19), чтобы получить подробную информацию о ваших доходах за день, неделю, месяц или квартал.

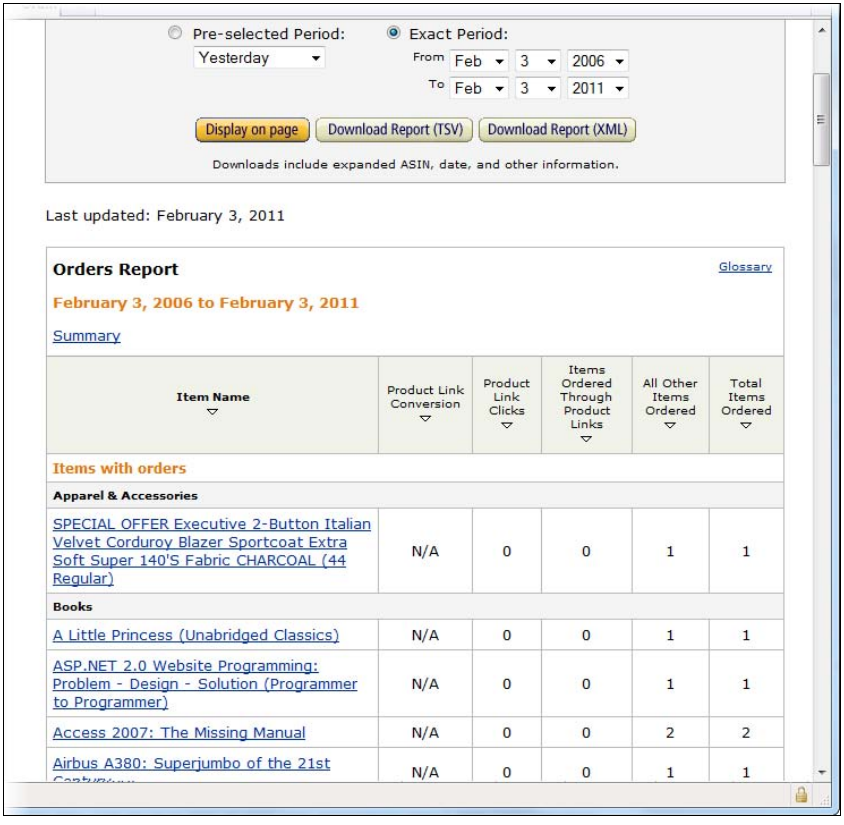


Рис. 14.19. Amazon предлагает много отчетов разных типов. Чтобы увидеть на экране занимательное зрелище с товарами, которые покупают ваши посетители, щелкните кнопкой мыши ссылку **Orders Report** (Отчет о заказах) в меню слева

УГОЛОК ДИЗАЙНЕРА

Продавайте созданный вами дизайн

Не секрет, что Web-пространство заполнено малым бизнесом, продающим товары ручной работы и сделанные на заказ. Вы можете расширить ассортимент своих товаров, не ограничиваясь предложениями от Amazon, но это потребует некоторых усилий. С учетом того, что вы хотите продавать, придется поискать оптовиков или народных умельцев, либо просто создать столярную мастерскую в подвале. Но если у вас есть что-то попроще, например, индивидуального изготовления одежда, кружки, шляпы, магниты, плакаты, пуговицы, чехлы для iPhone, маты для йоги, CafePress (**www.cafepress.com**) и Zazzle (**www.zazzle.com**) — ваши лучшие новые друзья.

Оба сайта позволяют моделировать свои собственные товары и предлагать их для продажи в именной секции Web-сайта (никакой HTML-разметки не требуется). Гениальность CafePress и Zazzle в том, что они разрешают вам использовать собственное художественное оформление. Например, если вы хотите создать уникальную пивную кружку, вы не ограничены отпечатыванием сомнительной надписи и выбором предустановленного шрифта. Вы можете выгрузить собственный графический файл (будь то цифровое изображение или шедевр из Photoshop) и поместить его на боковой поверхности кружки. Конечно, не все так прекрасно.

Для многих товаров CafePress и Zazzle ограничивают место для размещения ваших картинок. Например, вы можете поместить графику на грудь и на спину большинства моделей одежды, но нельзя заполнять всю поверхность или окутывать бока (туфли Zazzle — эффектное исключение. Они разрешают помещать графику на всю внешнюю поверхность).

После того как художественное оформление создано, можно заказать одно изделие для себя или предложить его миллионной аудитории, отправив своих посетителей на автоматически генерируемую страницу магазина на сайтах CafePress и Zazzle. Если кто-нибудь купит этот товар, вы получите небольшую часть от цены, а CafePress и Zazzle обработают платеж, нанесут ваше оформление и доставят его вашему покупателю.

Если вас заинтересовала идея, но ждете вдохновения, пойдите на Zazzle и щелкните мышью кнопку **Search** (Поиск), без ввода названия товара. Вы увидите перечень всего, что они предлагают, причем самые ходовые товары появятся первыми. Регулярно выпускаемые футболки со слащавой графикой, нахальными девизами или ссылкой на политиков и текущие события — всегда особенно популярны.

Услуги для торговцев системы PayPal

Если ваш Web-сайт не слишком популярен, рекламные объявления и другие партнерские программы принесут вам лишь немного разменных денег. Если же у вас всепоглощающая мечта разбогатеть в Web, необходимо действительно чем-то торговать.

Однако не стоит идти далеко и превращаться в добившихся всего своими силами коммерческих лидеров Интернета. На удивление много людей устроили свою жизнь благодаря креативным товарам: футболкам с крылатыми фразами политиков, пустым винным бутылкам с наклейками "R.M.S. Titanic" и коллекционным игрушкам из подвала родственников. Ваш путь к процветанию в электронной коммерции может состоять не только из покупки оловянных ложек у Honest Ed’s и декорирования их макраме.

Независимо от качества ваших товаров вам необходим легкий и удобный способ продажи их в огромных количествах. Очень немного людей преодолеет барьер и отправит вам по электронной почте персональный чек. Но если у них будет возможность совершить импульсивную покупку с помощью кредитной карты, ваши шансы продать существенно возрастут.

Прием кредитных карт — не самая простая задача в мире. Сделать это можно двумя способами.

- Открыть коммерческий счет или транзитный счет продавца (merchant account) в банке. Это стандартный коммерческий способ приема кредитных карт. Требования в разных странах разные, вам может понадобиться бизнес-план, бухгалтер и некоторый начальный капитал.
- Использовать сторонний сервис. Ряд компаний принимают от вашего имени платежи по кредитным картам в обмен на фиксированный сбор или процент от продажи. В этой главе вы узнаете, как пользоваться лучшей из них, PayPal.

Если у вас не крупный бизнес, второй вариант всегда лучше из-за дополнительных рисков, сопутствующих сетевым продажам.

Прежде всего, Интернет — это открытое место. Даже если у вас есть коммерческий счет, необходим безопасный способ приема информации кредитной карты от ваших покупателей. Это означает необходимость *кодирования* номера кредитной карты (шифрования с помощью секретного ключа), чтобы интернет-перехватчики не смогли добраться до него. У большинства Web-разработчиков, сидящих в своем подвале, нет защищенного сервера, и многие Web-хосты начисляют дополнительную плату за пользование их серверами.

Другая проблема заключается в невозможности получить подпись электронного покупателя, когда операция продажи проводится в сети. Это делает вас незащищенными от *возвратных платежей* или *возмещений* (см. врезку далее).

ПРИМЕЧАНИЕ

PayPal — невероятно крупная интернет-компания, предлагающая платежные услуги в более чем 100 странах. Если бы вы решили ранжировать банки по количеству открытых у них счетов, PayPal (с более чем 200 миллионами владельцев счетов) оказалась бы крупнейшим банком в мире. Компания была основана в 1998 г. и в 2002 г. куплена компанией eBay.

ЧАСТО ЗАДАВАЕМЫЙ ВОПРОС

Представление о возвратных платежах

Что такое возвратный платеж?

Компании, обслуживающие кредитные карты, выставляют возвратный платеж, когда держатель карты просит их удалить платеж с его счета. Покупатель может заявить, что продавец не выполнил до конца взятые на себя обязательства или что он вообще не делал никакой покупки. Возмещение может появиться спустя недели или месяцы после того, как кто-то купил что-то.

Покупателю оформить возмещение относительно просто. Он только звонит в компанию, обслуживающую его кредитную карту, и аннулирует сделку. Заработанные вами деньги списываются с вашего счета, несмотря на то, что вы уже доставили товар. Если

вы захотите оспорить заявление покупателя, то окажетесь в незавидном положении, пытаясь убедить монополистическую компанию по выпуску кредитных карт принять вашу сторону. Многие мелкие предприниматели вообще не оспаривают возвратные платежи, потому что этот процесс очень труден, дорог и безуспешен.

Но если вы пользуетесь сторонним сервисом, шансы склонятся в вашу сторону. Если покупатель требует возмещение, оно направлено против сторонней компании, принимавшей платеж (например, PayPal), а не против вас. И несмотря на то, что PayPal не такая крупная компания, как средний международный банк, она все же основной клиент большинства компаний по обслуживанию кредитных карт и, следовательно, обладает значительным весом в борьбе с возвратными платежами.

В итоге у покупателей гораздо меньше шансов получить обратно деньги, если они платят через PayPal. И если они делают это, у вас есть надежда оспорить возмещение. Это даже позволит вам связаться с покупателем и выяснить, не простое ли это недоразумение (например, выяснить, не отправили ли вы товар по неверному адресу). А уж если вы настоящий параноик, можно воспользоваться PayPal's Seller Protection Policy (Программа защиты продавца PayPal), которая, если предпринять несколько дополнительных действий (например, сохранить квитанцию о доставке), застрахует вас от потерь размером до 500 долларов. Дополнительную информацию о том, как PayPal обрабатывает возвратные платежи см. на странице **www.paypal.com/chargeback**. Узнать больше о программе защиты продавца PayPal можно по адресу **www.paypal.com/SellerProtection**.

Регистрация в PayPal

После регистрации в компании PayPal вы сможете принимать платежи от покупателей со всего света. Вот как следует это сделать.

1. Перейдите на Web-сайт PayPal (**www.paypal.com**) и на исходной странице щелкните мышью ссылку **Sign Up** (Зарегистрироваться).

   Вы попадете на страницу **PayPal's Sign Up** (Зарегистрироваться в системе PayPal).

2. Выберите страну и язык.

3. Выберите тип счета, который хотите создать (**Personal** (Личный), **Premier** (для индивидуальных предпринимателей) или **Business** (для представителей компаний)), и затем щелкните мышью кнопку **Get Started** (Открыть счет) в соответствующей области страницы.

   Личный счет — идеален, если вы хотите использовать PayPal для покупки товаров на сайтах, таких как eBay, пользуясь своей кредитной картой или деньгами с вашего банковского счета. Вы сможете также получать без дополнительной оплаты деньги, переведенные другими участниками системы PayPal. Но есть важное ограничение: на личные счета нельзя перевести платежи с кредитной карты. В результате покупатели, желающие установить деловые отношения с вами, должны иметь деньги на счетах PayPal (которые они получают, продавая что-либо и получая платеж PayPal или переводя деньги на свой счет со счета в связанном с PayPal банке).

   Счет premier — лучший способ вести малый бизнес. Вы можете отправлять деньги (чудесно, если вы страстно мечтаете о редком фильме на eBay) и полу-

чать любой тип принимаемого PayPal платежа, включая кредитные и платежные банковские карты. Вы также сможете воспользоваться инструментами электронной коммерции PayPal. Но PayPal будет брать с вас сбор за каждый полученный вами платеж, и размер сбора зависит от объемов продаж, но не выходит за пределы 1,9—2,9% от общей суммы платежа (минимальный сбор равен 30 центам). Это означает, что с 25-долларовой продажи PayPal возьмет примерно 1 доллар сверху. Если вы принимаете платежи в иностранной валюте, вы отдадите дополнительные 2,5%. Получить полное представление о сборах и увидеть самые свежие процентные ставки можно на странице **www.paypal.com/fees**.

Счет business почти такой же, как счет premier за исключением того, что он доступен разным участникам. Это наилучший выбор для представителей крупного бизнеса с сотрудниками, которым предоставляется доступ к счету для совместного управления вашим сайтом.

4. Введите ваш адрес электронной почты и выберите пароль, а также укажите ваше имя, адрес и номер телефона.

   Выбирайте пароль посложнее, если не хотите, чтобы злонамеренный хакер раскрыл его и использовал ваш счет PayPal, чтобы учинить марафон электронных покупок.

   Совет

   Возьмите за правило беречь информацию вашего счета PayPal так же, как свой банковский PIN-код. Если вы параноик, не пользуйтесь счетом PayPal для покупки товаров на других Web-сайтах. Не применяйте также для этого и свою кредитную карту — электронные перехватчики могут украсть ваши данные и затем отправиться первым классом в круиз на Ибицу.

5. В заключение щелкните мышью кнопку **Agree and Create Account** (Согласен и открываю счет) для завершения процесса.

   PayPal пришлет вам подтверждение немедленно. Щелкните кнопкой ссылку в письме для активации счета и затем для сбора платежей можно создавать кнопки PayPal и корзины для покупок (подробности см. далее).

Прием платежей

PayPal делает создание Web-страниц электронной коммерции до смешного легким. Один из способов — добавить кнопку **Buy Now** (Купить сейчас) на любую страницу вашего сайта.

1. Перейдите на сайт **www.paypal.com** и зарегистрируйтесь.

   - После регистрации вы получите доступ к страницам, заполненным товарами.
   - Используйте поле **My Account** (Мой счет) для обновления информации о состоянии вашего счета, просмотрите последние выполненные транзакции, проверьте ваш баланс и запросите снятие денег со счета.

- Щелкните мышью ссылку **Send Money** (Отправка средств), чтобы отправить по электронной почте кому-либо чек (который вы обеспечиваете средствами с вашего счета PayPal, вашего счета в реальном банке или кредитной карты).
- Щелкните мышью вкладку **Request Money** (Запрос о получении средств), чтобы отправить покупателю по электронной почте сообщение с запросом об оплате.
- Воспользуйтесь вкладкой **Create an Invoice** (Создать счет-фактуру), чтобы послать по электронной почте платежное требование через PayPal.
- Щелкните кнопкой мыши вкладку **Merchant Services** (Услуги для торговцев) для создания кнопок, которые можно добавить на ваши Web-страницы.
- Используйте **Auction Tools** (Инструменты аукциона), чтобы задать PayPal, как форму платежа за товары, которые вы продаете на eBay. (eBay — только одно из самых популярных мест для организации электронной коммерции.)
- Щелкните кнопкой мыши вкладку **Products & Services** (Продукты и услуги), чтобы узнать больше о различных сервисах PayPal.

2. Щелкните кнопкой мыши вкладку **Merchant Services** (Услуги для торговцев).

   Прокрутите страницу и увидите множество инструментов для сбора денег (рис. 14.20).

3. Щелкните кнопкой мыши ссылку **Buy Now** (Купить сейчас) в списке **Create Buttons** (Создание кнопок).

   PayPal отобразит страницу, на которой вы настроите внешний вид вашей кнопки **Buy Now** и установите цену вашего товара (рис. 14.21).

4. Присвойте название вашему товару и, если хотите отслеживать его, код товара. Затем задайте цену и валюту.

   При выборе валюты не беспокойтесь о блокировке иностранных посетителей. Компании, обслуживающие кредитные карты, будут рады одолжить канадским покупателям доллары США, покупателям из Соединенных штатов — евро и европейским клиентам — рупии. Просто выберите валюту, на которую по вашему мнению рассчитывают покупатели.

5. Если хотите предоставить покупателям разные варианты, введите информацию в поле **Customize button** (Кнопка "Настроить").

   Вы можете собирать дополнительную информацию от своих покупателей тремя остроумными способами.

   - Выберите вариант **Add drop-down menu with price/option** (Добавить раскрывающееся меню с ценой/вариантом). Он позволит предоставить покупателям список разновидностей вашего товара, отличающихся ценой (рис. 14.22). Например, можно предложить выбрать между банкой со стандартным, высококачественным или органическим кормом для хомяков.

     Выберите **Add drop-down menu** (Добавить раскрывающееся меню). Этот вариант также предоставляет покупателям список разновидностей товара, но

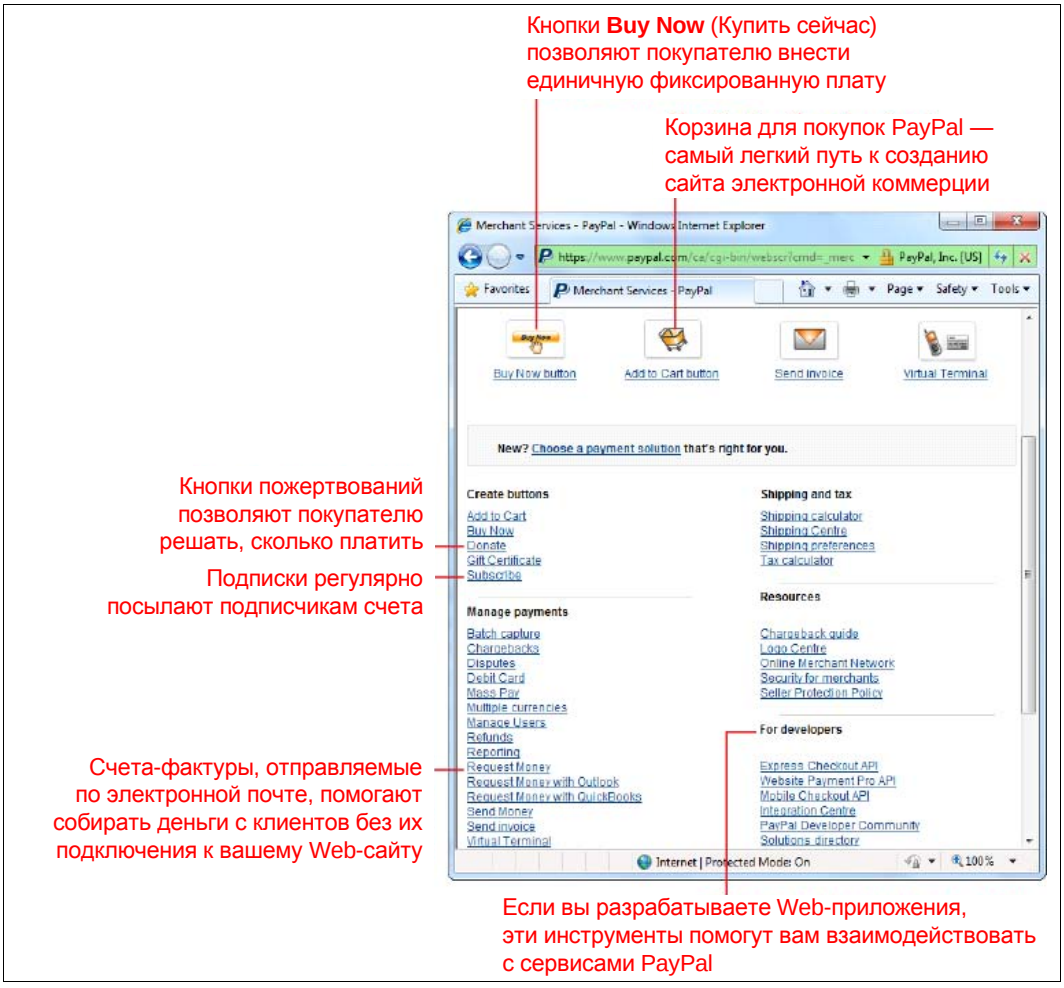


Рис. 14.20. PayPal предлагает ряд способов сбора денег по электронной почте или с помощью кнопок, установленных на ваших Web-страницах

без изменения цены. Например, можно использовать список, позволяющий покупателям выбрать цвет предметов вышитого нижнего белья, которое они собираются купить.

- Выберите вариант **Add text field** (Добавить текстовое поле). Он вставляет текстовое поле, в которое покупатели могут вводить текст. Применяйте его, если хотите собирать меняющуюся информацию, например имя, которое ваш покупатель хочет, чтобы было выгравировано на магнитной отвертке.

6. Если хотите изменить внешний вид кнопки, щелкните мышью ссылку **Customize appearance** (Настроить текст или внешний вид).

   PayPal предложит несколько вариантов размеров кнопки и текста на ней.

   Стандартная кнопка **Buy Now** (Купить сейчас) очень удобна, но простовата. Если вы создали лучшее изображение кнопки и выгрузили его на свой сайт, здесь

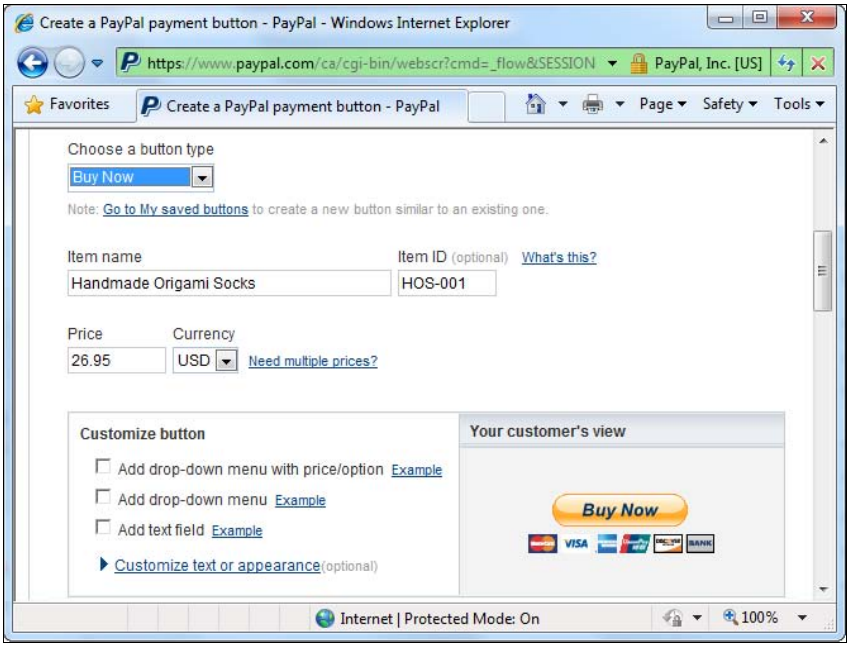


Рис. 14.21. В этом окне вы задаете основные параметры кнопки **Buy Now** (Купить сейчас)

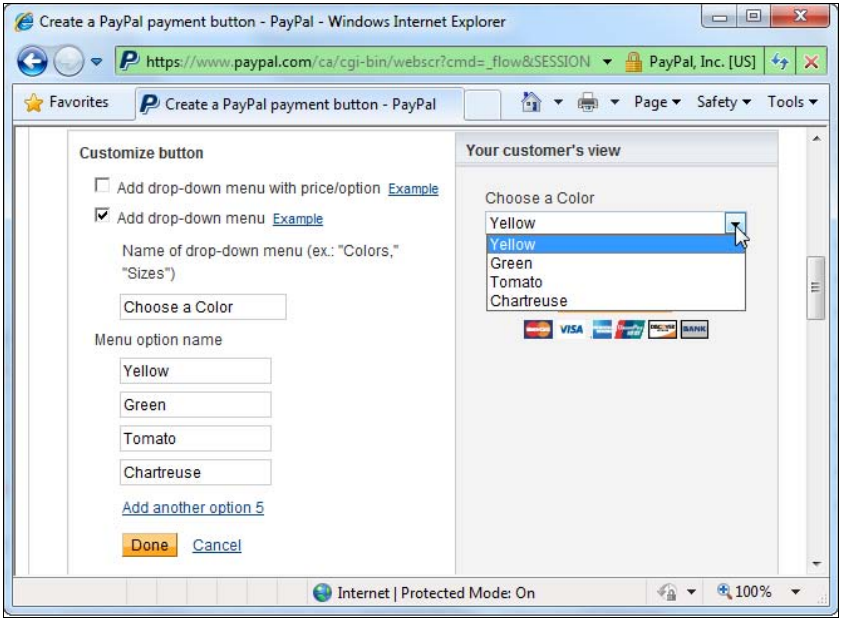


Рис. 14.22. С помощью раскрывающегося списка вариантов можно собрать дополнительную информацию о типе товара, который нужен покупателю. Это полезно, если вы предлагаете один и тот же товар, разных размеров или цветов (как показано в этом примере). Для добавления в список нового цвета щелкните мышью ссылку **Add another option** (Добавить вариант 4) под текущим набором вариантов. После щелчка мышью кнопки **Done** (Готово) PayPal обновляет предварительное отображение кнопки, расположенное справа, чтобы показать вам, как этот список будет выглядеть

можно указать URL изображения. (Если позже захотите применить другую кнопку, вы всегда можете изменить HTML-код, сгенерированный PayPal.)

7. Прокрутите страницу и при желании введите дополнительные параметры.

   Например, нужен ли вам адрес покупателя (PayPal полагает, что нужен), хотите ли вы, чтобы покупатели добавили свои комментарии к платежу (обычно они не могут), и должен ли PayPal направлять посетителей в определенное место после завершения или отмены платежа (вы можете отправить покупателей на определенный URL на вашем сайте вместо стандартных страниц PayPal).

8. Щелкните мышью кнопку **Create Button** (Создать кнопку).

   PayPal отображает текстовое поле с разметкой для вашей настроенной кнопки **Buy Now** (Купить сейчас). Скопируйте и вставьте разметку на вашу Web-страницу.

Когда создается кнопка **Buy Now** (Купить сейчас), PayPal помещает все внутрь элемента `<form>` (см. *разд. "Элементы формы" главы 15*). Далее приведен пример кнопки для пары самодельных носков оригами, основанной на последних нескольких пунктах, приведенных ранее.

```
<form action="https://www.paypal.com/cgi-bin/webscr" method="post">
  <input type="hidden" name="cmd" value="_s-xclick" />
  <input type="hidden" name="hosted_button_id" value="633788" />
  <table>
    <tr><td>
      <input type="hidden" name="on0" value="Choose a Color" />
       Choose a Color</td></tr><tr><td>
      <select name="os0">
        <option value="Yellow">Yellow</option>
        <option value="Green">Green</option>
        <option value="Tomato">Tomato</option>
        <option value="Chartreuse">Chartreuse</option>
      </select>
    </td></tr>
  </table>
  <input type="image" border="0" name="submit" alt=""
   src="https://www.paypal.com/en_US/i/btn/btn_buynowCC_LG.gif" />
  <img src="https://www.paypal.com/en_US/i/scr/pixel.gif" alt=""
   width="1" height="1" />
</form>
```

Если вы добавили варианты, то увидите в HTML-коде элементы `<select>` и `<option>`, описывающие соответствующие поля списка (см. табл. 15.3). Вы можете изменить у кнопки атрибут `src` (выделенный жирным шрифтом в только что приведенном листинге) и указать на другой файл изображения. PayPal вставляет после кода для кнопки **Buy Now** (Купить сейчас) невидимое изображение (pixel.gif в приведенном примере), которое отслеживает просмотры страницы. (Это тот же метод отслеживания, который использует Amazon.)

СОВЕТ

Не меняя поля <input> и сохраняя весь код в элементах <form>, вы можете настраивать разметку, созданную для вас PayPal. Например, можно вставить в форму другие элементы или приукрасить ее средствами таблицы стилей. Вы также можете изменить макет, удалив невидимую таблицу (представленную элементами <table>, <tr> и <td>), которую PayPal использует для формирования кнопки и ваших вариантов.

Что же происходит, когда покупатель щелкает мышью кнопку **Buy Now** (Купить сейчас) и отправляет эту форму? Атрибут action в самой первой строке приведенного ранее кода сообщает, что браузер посылает информацию покупателя в систему PayPal с помощью URL, указанного в action (в данном случае **https://www.paypal.com/cgi-bin/webscr**). В действительности для этого используется защищенный канал, мешающий интернет-перехватчикам, вот почему URL начинается с "https" вместо "http".

Обратите внимание на то, что в этой форме нет ключевых фрагментов данных, например, названия товара или его цены. Это мера безопасности, разработанная для защиты от злоумышленников, которые могут повредить разметку на вашей Web-странице и заплатить вам меньше, чем стоят ваши товары. Когда PayPal получает данные формы, извлекается скрытое значение ID (в данном примере 633788) и в гигантской секретной базе данных кнопок ищется соответствующий товар, цена и продавец (т. е. вы).

Фактически разметка PayPal не содержит никакой информации о товаре, который вы продаете. Вы помещаете название товара, его изображение, описание и цену на свою Web-страницу (возможно, перед кнопкой **Buy Now** (Купить сейчас)). Далее приведен пример.

```
<!DOCTYPE html>
<html>
<head>...</head>

<body>
  <h1>Handmade Origami Socks</h1>
  <p><img src="origami.jpg" alt="Handmade Origami Socks" class="float">
  You've waited and they're finally here. Order your own
  pair of origami socks for only $26.95 and get them in time
  for the holidays. What better way to show your loved ones how
  poor your gift giving judgement really is?</p>
  <form action="https://www.paypal.com/cgi-bin/webscr" method="post">
      ...
  </form>
</body>
</html>
```

На рис. 14.23 показан результат. Этот пример отображает стандартную страницу заказа PayPal, но вы можете настроить ее, добавив свой собственный логотип (см. *следующий раздел*).

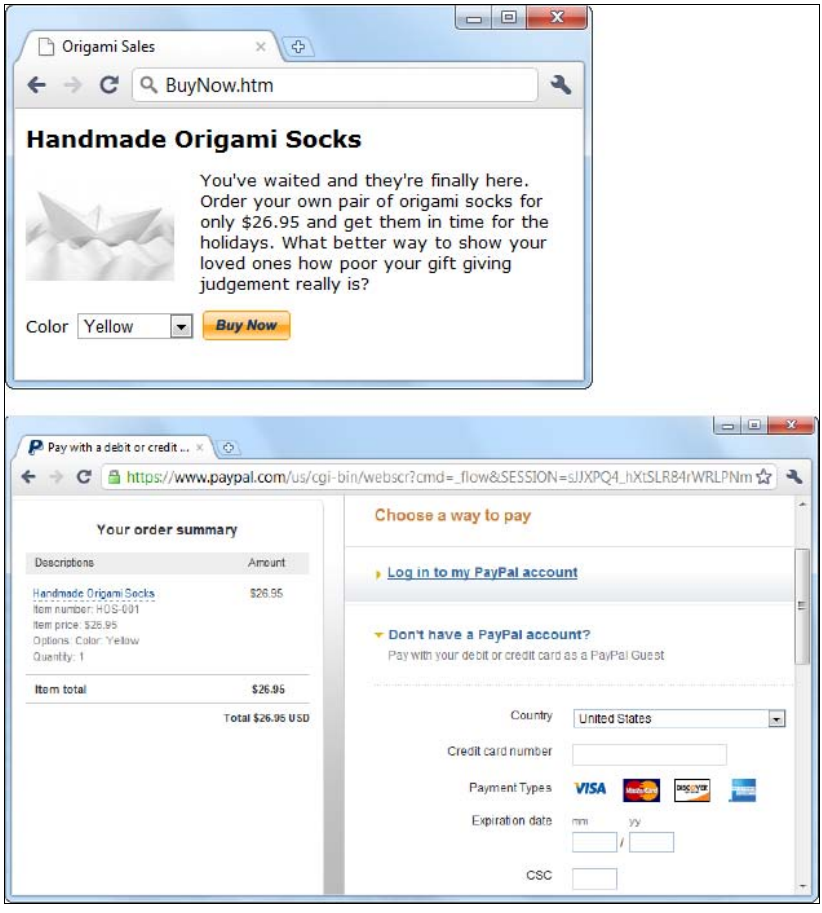


Рис. 14.23. *Вверху*: кнопка **Buy Now** (Купить сейчас) терпеливо ждет на вашей странице. *Внизу*: щелчок кнопки **Buy Now** (Купить сейчас) начинает безопасный процесс проверки с помощью PayPal. Ваш посетитель может заплатить за товар кредитной картой, и вы оба получите по электронной почте подтверждение транзакции. Теперь выбор способа завершения сделки только за вами

Создание корзины для покупок

Кнопка **Buy Now** (Купить сейчас) PayPal предоставляет вам замечательный способ быстрых продаж. Но если вы мечтаете об империи электронной коммерции, необходимо создать хранилище, где посетители могут отложить сразу несколько товаров, а потом одновременно оплатить их все.

Для того чтобы предоставить посетителям такого рода удобство, вам понадобится корзина для покупок, основной элемент Web-сайтов электронной коммерции. К счастью, вам не нужно программировать собственную корзину, вместо этого можно использовать сервис готовых корзин для покупок PayPal, которые очень легко встраиваются в ваш Web-сайт.

Создание корзины PayPal удивительно похоже на создание кнопки **Buy Now** (Купить сейчас) (поэтому, если вы не опробовали кнопку, можно поиграть с корзиной,

прежде чем идти дальше). Суть в том, что для каждого продаваемого товара создается отдельная кнопка **Add to Cart** (Добавить в корзину). Вам будет предложено множество тех же параметров, которые вы видели во время создания кнопки **Buy Now** (Купить сейчас). Например, вы можете задать цену, код товара, сборы за доставку и т. д. Разница заключается в том, что при щелчке мышью кнопки **Add to Cart** (Добавить в корзину) PayPal не направляет посетителей прямо на страницу оформления заказа. Вместо этого он отображает в новом окне страницу корзины для покупок. Посетителям разрешается продолжать делать покупки до тех пор, пока есть то, что им хочется купить. Для завершения сеанса покупок они щелкают мышью кнопку **Checkout** (Оформить заказ).

Для демонстрации этого сервиса в действии в следующем примере как отправная точка используется страница, изображенная на рис. 14.24. Данный пример также показывает огромную пользу верстки на основе стилей. Проверьте загружаемые примеры, доступные на странице Missing CD (Отсутствующий CD) по адресу **www.missingmanuals.com/cds/caw3**, чтобы опробовать их самостоятельно.

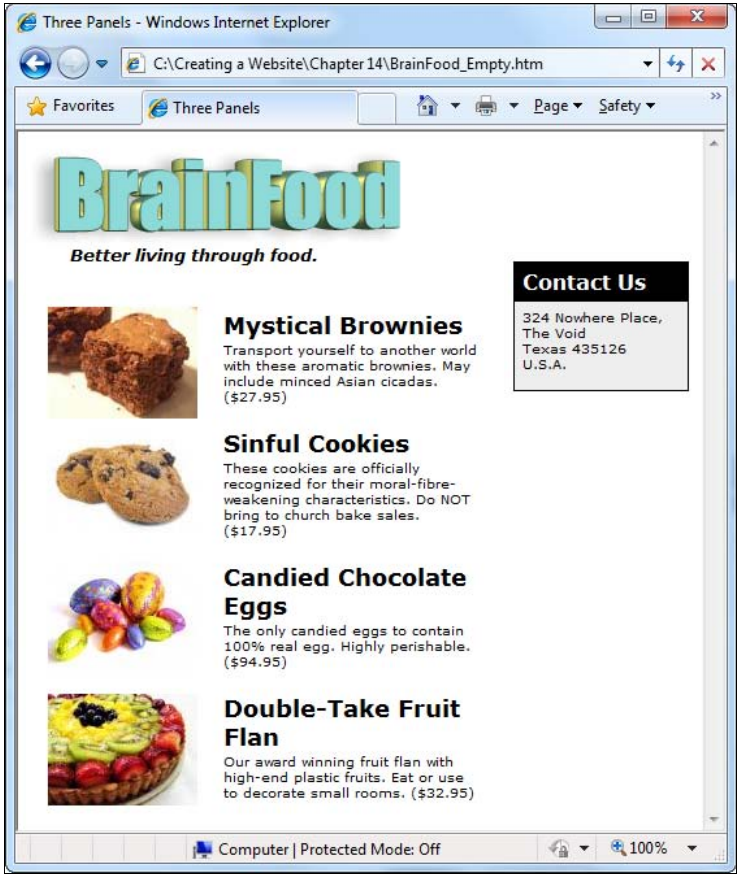


Рис. 14.24. В данный момент на странице **BrainFood** (Пища для мозга) есть огромный перечень описаний продуктов, но у ваших посетителей нет никакой возможности сделать спонтанную покупку. Ситуацию можно изменить, добавив кнопку корзины покупок, предлагаемую PayPal

Создание пользовательского стиля страницы

Прежде чем создавать корзину для покупок, следует предпринять одно дополнительное действие, чтобы по-настоящему персонифицировать свою страницу платежей. Если вас устраивает стандарт PayPal, пропустите этот раздел и переходите к следующему. Но если вы хотите, чтобы на страницах корзины для покупок отображался логотип вашей компании, продолжайте чтение.

1. Если вы еще не на сайте PayPal, перейдите туда (**www.paypal.com**) и зарегистрируйтесь.
2. Выберите вкладку **My Account** (Мой счет) и щелкните кнопкой мыши ссылку **Profile** (Профиль), расположенную под заголовком.

   На странице **Profile** предоставляется масса информации о ваших предпочтениях, сгруппированная в три основные категории **Account Information** (Личные данные) (кто вы и где живете), **Financial Information** (Финансовые данные) (ваш банк, кредитная карта и хронология платежей) и **Selling Preferences** (Настройки продаж) (дополнительные возможности, которые можно использовать с помощью услуг для торговцев системы PayPal). В данном случае вас интересует раздел **Selling Preferences**.
3. В разделе **Selling Preferences** щелкните кнопкой мыши ссылку **Custom Payment Pages** (Стили пользовательской страницы платежа).

   Вы попадете на страницу **Custom Payment Page Styles** (Стили пользовательской страницы платежа), где можно задать новые стили страницы или отредактировать существующие.
4. Щелкните мышью кнопку **Add** (Добавить), чтобы добавить новый стиль страницы.

   Вы начнете со стиля единственной страницы — стандарта PayPal, на которой щеголяет стандартный логотип PayPal.
5. Задайте информацию для стиля вашей страницы.

   Введите информативный заголовок в поле **Page Style Name** (Название стиля страницы), который поможет запомнить, как выглядят разные стили.

   Для ссылки на изображение на вашем Web-сайте задайте **Header Image URL** (URL изображения заголовка). Это изображение — логотип, который вы хотите отображать в левом верхнем углу страницы корзины для покупок PayPal. Максимально возможная ширина изображения 750 пикселов, а высота — 90 пикселов.

> ***Примечание***
>
> Поскольку страница корзины PayPal — защищенная страница, когда применяется пользовательский логотип, покупатель может получить сообщение, информирующее о наличии некоторых небезопасных элементов на странице (а именно вашей картинки). Если вы хотите спрятать от своих гостей это сообщение, попросите вашу компанию, предоставляющую Web-хостинг, о помещении вашего изображения на защищенный (HTTPS) сервер.

Вы можете с помощью числовых кодов задать цвета **Header Background Color** (Цвет фона заголовка), **Header Border Color** (Цвет рамки заголовка), **Background Color** (Цвет фона) вашей страницы. Это часть не обязательна, если вас устраивают стандартный белый цвет, оставьте ее без изменений.

6. Щелкните мышью кнопку **Save** (Сохранить), чтобы сохранить ваш стиль страницы.

   Перед подтверждением внесенных изменений можно щелкнуть мышью **Preview** (Предварительный просмотр), чтобы посмотреть предварительно на то, как будет выглядеть страница платежей PayPal.

7. Выберите ваш новый стиль страницы и затем щелкните мышью **Make Primary** (Сделать основным).

   Теперь все ваши посетители увидят пользовательскую страницу оформления заказа.

Генерация кнопок корзины

Вы готовы создавать кнопки, добавляющие товары в корзину вашего электронного магазина. Вот как это делается.

1. Если вы еще не на сайте PayPal, перейдите туда (**www.paypal.com**) и зарегистрируйтесь.
2. Щелкните кнопкой мыши ссылку **Merchant Services** (Услуги для торговцев), а затем ссылку **Add to Cart button** (кнопка Добавить в корзину).

   PayPal отобразит страницу, на которой вы настроите кнопку **Add to Cart** (Добавить в корзину), рассчитанную на один товар.

3. Присвойте вашему товару название и, если хотите его отслеживать, код товара. Затем укажите цену, валюту и другую сопутствующую информацию.

   Эти параметры точно такие же, как у кнопки **Buy Now** (Купить сейчас).

4. Щелкните мышью **Create Button** (Создать кнопку).

   Вы увидите текстовое поле с разметкой для вашей настроенной кнопки **Add to Cart** (Добавить в корзину). Скопируйте текст и вставьте его на вашу Web-страницу. Но помните, что этот код применяется к одному конкретному товару. Если у вас их несколько на странице (как в примере **BrainFood** (Пища для мозга)), необходимо сгенерировать несколько кнопок. Для этого щелкните кнопкой мыши **Create similar button** (Создать такую же кнопку) и вернитесь к пункту 3. Когда завершите генерацию всех необходимых кнопок и скопируете каждую из них на свою страницу, перейдите к следующему пункту.

5. Создайте кнопку **View Cart** (Просмотреть корзину).

   Ваше решение для корзины покупок будет неполным без кнопки, которая позволяет покупателям увидеть содержимое их корзин (и затем переходить к виртуальной кассе). Для создания такой кнопки щелкните кнопкой мыши ссылку **Create a View Cart button** (Создать кнопку Просмотреть корзину).

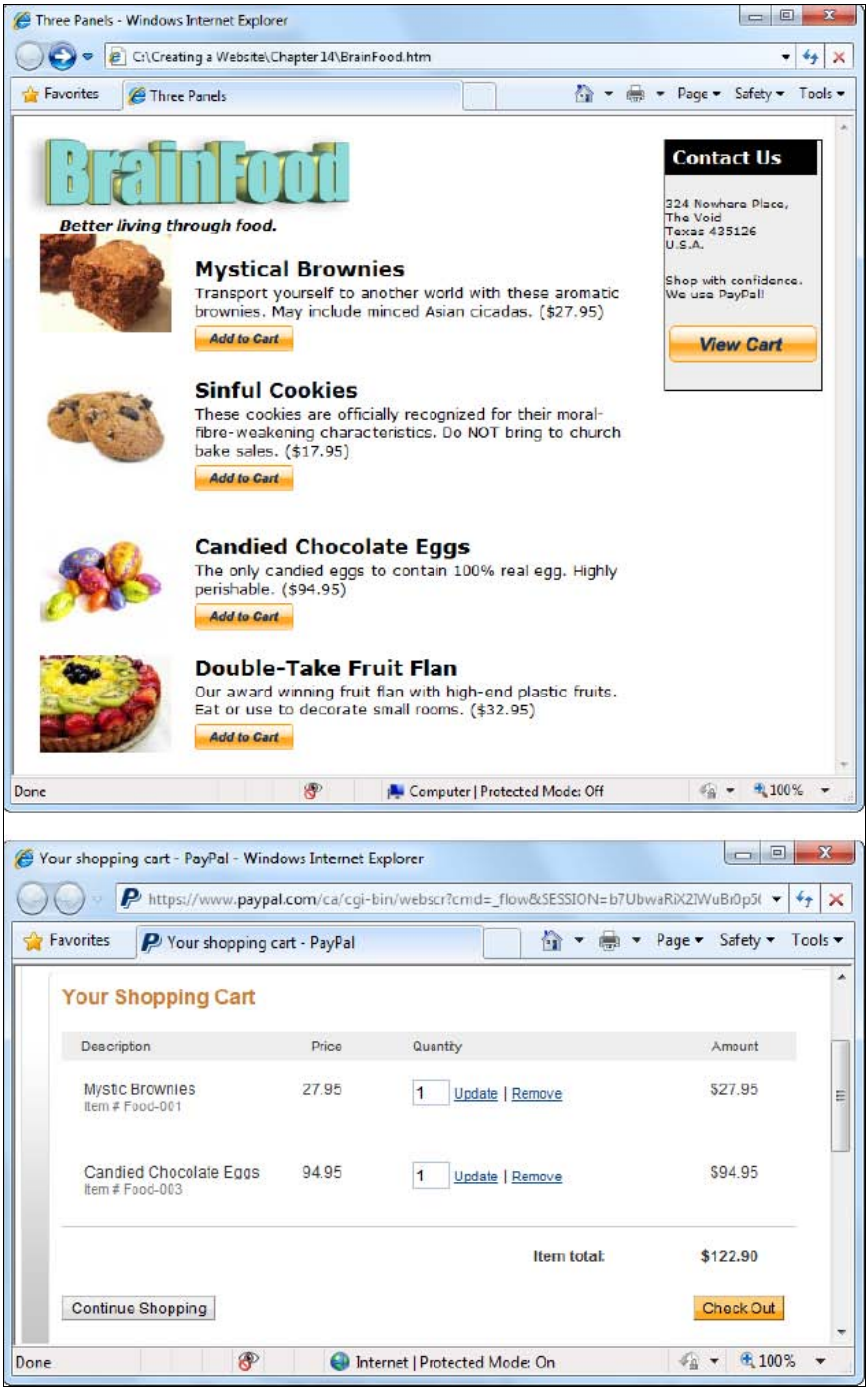


Рис. 14.25. *Вверху*: показана обновленная страница **BrainFood** (Пища для мозга) с кнопками корзины для товаров.
Внизу: после щелчков мышью нескольких кнопок **Add to Cart** (Добавить в корзину) выводится страница корзины с покупками, которую увидят ваши посетители (отдельное окно). Для совершения покупки им нужно только щелкнуть мышью кнопку **Check Out** (Оформить заказ)

У кнопки **View Cart** (Просмотреть корзину) практически нет настраиваемых параметров, т. к. ее задача очень проста. Вы можете использовать стандартную кнопку **View Cart** (Просмотреть корзину) или задать URL, указывающий на кнопку с вашим собственным дизайном. После того как выбор сделан, щелкните мышью кнопку **Create Button** (Создать кнопку) и затем скопируйте разметку на вашу страницу с другими кнопками. На рис. 14.25 показан окончательный результат.

Вывод ваших денег

PayPal надежно прячет все ваши платежи на вашем счете (который подобен счету в виртуальном банке). Вы можете в любое время просмотреть свой баланс. Просто зарегистрируйтесь и щелкните кнопкой мыши вкладку **My Account** (Мой счет).

Если вы заработали немного денег, возможно, вы с радостью оставите их на PayPal, чтобы можно было покупать товары на Web-сайтах, например, eBay и **www.buy.com**. Но если вы загребли внушительную кучку, возможно, появится желание перевести часть ее в реальный мир.

Самый распространенный вариант — отправить деньги на ваш счет в банке. Для этого необходимо предоставить PayPal информацию о банке. PayPal не берет плату за перевод денежных средств до тех пор, пока выводимая сумма меньше определенного минимума (например, 150 долларов). Но ваш банк может запросить плату за электронную транзакцию. Людям, проживающим в разных странах, PayPal предлагает разные варианты вывода средств. Например, система может разрешить перевести деньги на дебетовую (расходную) или кредитную карту.

Для того чтобы воспользоваться любым предложенным вариантом, зарегистрируйтесь, щелкните кнопкой мыши вкладку **My Account** (Мой счет), затем ссылку **Withdraw** (Вывести средства), расположенную прямо под заголовком, и потом следуйте инструкциям.

15

ГЛАВА 15

JavaScript: добавка интерактивности

JavaScript — облегченный язык программирования, разработанный для наполнения ваших Web-страниц интерактивными средствами. Его возможностей хватает только для вставки кое-каких причудливых эффектов, при этом он не сможет сильно повредить ваш сайт, если программный код окажется ненадежным. JavaScript отлично подходит для создания всплывающих окон, встроенных анимационных эффектов или модификации контента, выводимого на вашу Web-страницу. С другой стороны, он не сможет помочь в создании электронного магазина, для этого вам потребуются инструменты PayPal, описанные в *главе 14*, или платформа для серверного программирования, рассмотренная далее.

Цель этой главы — не научить вас всем тонкостям программирования на JavaScript, а дать достаточно знаний, чтобы вы могли находить в сети отличный программный код на JavaScript, разбираться в нем настолько, чтобы вносить элементарные изменения и затем помещать его на свои страницы и получать желаемые результаты. Поскольку бесплатный код на JavaScript можно найти на сотнях Web-сайтов, эти навыки могут оказаться очень полезными.

Основные сведения о JavaScript

У языка JavaScript длинная история: впервые он появился на сцене вместе с браузером Netscape Navigator 2 в 1995 г. Internet Explorer присоединился к ним с версией 3. Сейчас все современные браузеры запускают сценарии на JavaScript, и в результате этот язык стал очень популярен.

> ***Примечание***
>
> JavaScript сильно отличается от языка Java (хотя код иногда выглядит очень похоже, поскольку в обоих языках одни и те же правила записи кода). Java — полнофункциональный язык программирования, во всех отношениях столь же мощный и сложный, как C# и Visual Basic.

JavaScript может следующее:

- динамически вставлять новое содержимое на Web-страницу или изменять имеющийся HTML-элемент. Например, вы можете отображать персонифициро-

ванное приветствие ваших посетителей ("Примет, Джо!") или заставить заголовки то и дело увеличиваться и уменьшаться (см. *разд. "Использование HTML-объектов в сценарии" данной главы*);

- собирать сведения о текущей дате, браузере вашего посетителя или содержимое, которое ваш посетитель вводит в форму. Всю эту информацию можно отображать на странице или использовать для принятия решения о дальнейших действиях вашей страницы. Вы могли бы, например, не разрешить посетителям перемещаться по вашему сайту, пока они не введут свой адрес электронной почты;
- реагировать на события, происходящие в браузере. Например, можно вставить код на JavaScript, который запускается, когда страница завершает загрузку или когда посетитель щелкает кнопкой мыши изображение;
- общаться с программой, выполняющейся на Web-сервере. JavaScript может запросить у сервера самые свежие котировки акций, например, или пакет записей из базы данных компании. Решение именно этой задачи не будет рассматриваться в данной главе, поскольку для нее потребуется серьезный программист, который должен написать программу не на JavaScript, выполняемую на Web-сервере.

Так же важно понять, чего не может сделать JavaScript. Код на JavaScript помещается в "песочницу", т. е. изолируется. Браузер запирает вашу страницу с кодом на JavaScript в тщательно контролируемой области памяти, называемой *песочницей*. В результате код не может совершать никаких рискованных действий на компьютере вашего посетителя, например, посылать запросы на принтер, получать доступ к файлам, запускать другие программы, форматировать жесткий диск и т. д. Такой подход гарантирует надежную защиту.

Серверное и клиентское программирование

Для того чтобы понять, как JavaScript встраивается в Web-вселенную, важно разобраться с двумя видами программирования в Web-пространстве.

Когда вы используете поисковую машину, например Google, или идете на сайт электронной коммерции, такой как Amazon, вы фактически подключаетесь к мощному компоненту программного обеспечения, называемому *серверным приложением* и выполняющемуся на Web-сервере. Когда вы посещаете один из таких сайтов, вы посылаете информацию программе на сервере, например, ключевые слова для поиска или выбранную для покупки книгу. Программа в свою очередь справляется в массивной базе данных и выдает HTML-код, создающий страницу, которую вы видите в своем браузере.

Web-миром правят серверные приложения, т. к. практически нет ничего такого, чего они не могут делать. Но их очень трудно программировать. Разработчики должны не только обеспечивать выполнение программой любого сложного программного кода и обработку гигантских баз данных, но и добиваться, чтобы она делала это без нарушения нормальной работы сайта во время просмотра его миллионами людей, т. к. будто на сайте единственный посетитель. Это тяжелая работа и она в надежных руках несчастных созданий, которых мы называем программистами.

Клиентские приложения, напротив, применяют совсем иную модель. Они встраивают небольшие легковесные программки в обычную HTML-страницу. Когда браузер загружает ее, он сам запускает программу (если только ваши параметры безопасности или проблемы совместимости не сделают ее неактивной). Клиентские программы далеко не так сложны, как серверные — они, например, не могут надежно опрашивать громадные базы данных, хранящиеся на Web-серверах, и по соображениям безопасности непосредственно менять множество компонентов программного обеспечения на вашем компьютере. Но зато их гораздо проще создать.

ПРИМЕЧАНИЕ

Разница между серверными и клиентскими программами порой сглаживается из-за того, что высококлассные Web-сайты применяют и те, и другие. Например, Amazon использует точно настроенный код на JavaScript для своих раскрывающихся меню и средства **Look Inside**, позволяющего заглянуть в книгу. Но все серьезные задачи, такие как отслеживание заказов, обработка кредитных карт и хранение клиентских отзывов, выполняются на Web-сервере.

Языки сценариев

Мир клиентского программирования делится на две категории. С одной стороны, вы найдете *многофункциональные среды программирования* (rich programming environments). Эти технологии нуждаются в дополнительных модулях браузеров (плагинах) и предоставляют расширенные средства, подобные аналогичным инструментам для настольных приложений. Самый популярный пример — Adobe Flash, но Microsoft Silverlight также достойный конкурент. Если вы когда-либо видели видео или играли в бесплатную сетевую игру в вашем браузере (рис. 15.1), вы почти наверняка пользовались одной из двух. И если вы когда-нибудь во весь голос ругали анимационное рекламное объявление, которое вдруг занимало все окно вашего браузера, благодарить также надо одну из этих платформ.

К другой категории относятся *языки сценариев*, такие как JavaScript. Язык JavaScript не так развит, как Flash и Silverlight, но его простота идет ему на пользу.

- Вам не нужно писать целую программу для применения языка сценариев. Вместо этого вы создаете *сценарий* (скрипт) — короткий клочок текста, который можно поместить непосредственно на Web-страницу. Этот сценарий передает браузеру ряд команд, например "прокрути заголовок слева направо" или "отобрази рекламу зубочисток со вкусом сливочной помадки в новом окне".
- Даже если вы не знаете всех тонкостей языка JavaScript, вы часто можете скопировать и вставить отличный скрипт с бесплатного Web-сайта и получить моментальное вознаграждение.
- Поскольку коду на JavaScript не нужен плагин и поскольку он надежно работает во всех браузерах, крупные компании могут полагаться на него, делая свои страницы немного живее и привлекательнее, при этом не оставляя никого за бортом (iPhone и iPad компании Apple не поддерживают такие плагины, как Flash и Silverlight).

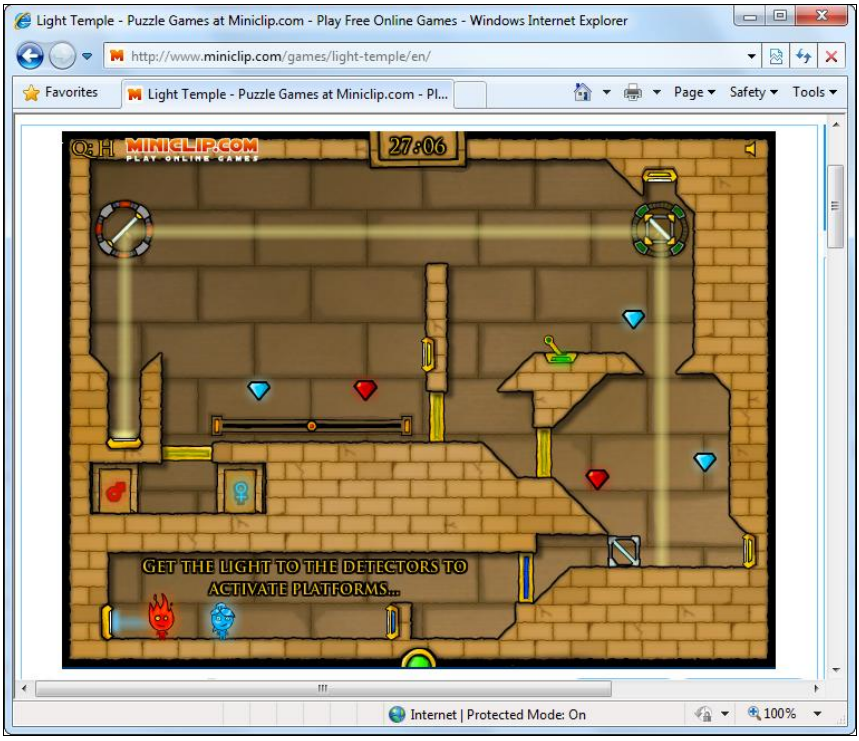


Рис. 15.1. Эта обычная Web-страница содержит встроенное Flash-приложение, которое управляет тщательно прорисованной диалоговой компьютерной игрой

Сегодня JavaScript — единственный язык сценариев с полной поддержкой любыми браузерами, поэтому данный язык все применяют в Web-пространстве.

АНОНС HTML5

Не тот JavaScript, которым пользовались ваши отцы

JavaScript начал свое существование как замысловатая игрушка. Даже когда начиненные программным кодом Web-сайты, например Amazon и eBay, начали набирать обороты, роль JavaScript была незначительной.

Сегодня же сайты электронной коммерции используют JavaScript все время. Когда крупные компании создают Web-приложения, они применяют на своем сервере одну из множества платформ программирования (включая ASP.NET, PHP, JSP, Perl, Ruby и др.), но при этом сохраняют JavaScript для управления клиентскими программами. А сверхгениальные разработчики для оживления Web-страниц постоянно расширяют границы возможного, комбинируя JavaScript с серверным программным кодом. Google сейчас применяет подобную смесь — а не Flash — в таких передовых Web-приложениях, как Google Maps и Gmail.

Эта тенденция усиливается благодаря HTML5. Новая версия языка HTML связана с множеством стандартов следующего поколения, расширяющих JavaScript. В действительности этим во многом объясняется то, что версия HTML5 привлекла столько внимания и в рекордное время перешла из статуса аутсайдера на позицию принятого стандарта (см. *разд. "HTML, XHTML и HTML5" главы 2*). Когда работа над HTML5 завершится и эта версия языка будет реализована во всех современных браузерах, разработчики смогут применять JavaScript для определения географического место-

положения Web-путешественника, отображать динамическую графику и даже управлять Web-камерами и микрофонами. Так что считайте время, потраченное сегодня на изучение JavaScript, инвестицией в будущие Web-разработки.

JavaScript 101

Теперь, когда вы узнали кое-что о JavaScript и причинах его существования, пора погрузиться в него и приступить к созданию вашего первого реального сценария.

Элемент <*script*>

Любой сценарий начинается с блока `<script>`, который вы вставляете в HTML-документ. На самом деле у вас есть два варианта.

- Раздел `<body>`. Браузер запустит ваш сценарий, как только доберется до элемента `<script>`. Если поместить элемент `<script>` в начало раздела `<body>`, браузер выполнит сценарий до отображения страницы.

  Совет

  Обычно любители JavaScript помещают свои сценарии в конец раздела `<body>`. В этом случае вы избегаете ошибок, которые могут появиться, если вы применяете сценарий, базирующийся на другой части страницы, а браузер еще не считал этот раздел. Поскольку браузеры быстро считывают всю страницу, такие сценарии выполняются почти мгновенно.

- Раздел `<head>`. Если поместить стандартный сценарий в раздел `<head>` вашего HTML-документа, он запускается немедленно, прежде чем браузер обработает какую-либо часть разметки. Но гораздо чаще раздел `<head>` используется для сценариев, содержащих функции (см. *разд. "Функции" данной главы*). Функции не выполняются немедленно, вместо этого вы вызываете их, когда ваш посетитель совершает определенное действие на странице, например перемещает мышь.

  Примечание

  При желании можно поместить на Web-страницу сколько угодно блоков `<script>`.

Типичный блок сценария состоит из последовательности программных инструкций. Чтобы составить представление о том, как эти инструкции работают, рассмотрим следующий пример, который отображает на вашей странице окно предупреждения на JavaScript.

```
<!DOCTYPE html>

<html>

<head>

  <title>JavaScript Test</title>
</head>
```

```
<body>
  <h1>You Will Be Wowed</h1>
  <p>This page uses JavaScript.</p>
  <script type="text/javascript">
    alert("Welcome, JavaScript coder.")
  </script>
</body>

</html
```

Этот сценарий раскрывает окно с сообщением, как показано на рис. 15.2. Когда вы щелкните мышью кнопку **OK**, сообщение исчезнет и на странице все вернется к привычной жизни.

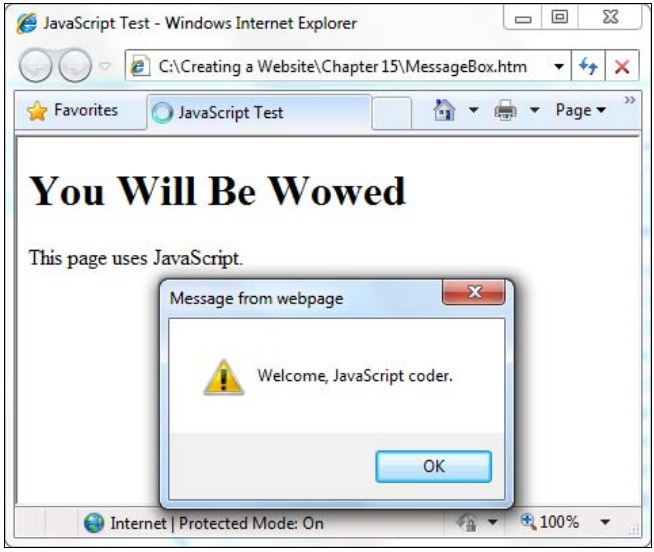


Рис. 15.2. Поскольку вы поместили элемент <script> для этой страницы в конец HTML-разметки, браузер сначала отобразит весь HTML-код, а затем раскроет окно предупреждения. Если поместить элемент <script> в начало раздела <body> (или в раздел <head>), окно предупреждения появится раньше, пока страница еще пуста. Далее браузер будет ждать, пока вы нажмете кнопку **OK**, и лишь потом загрузит остальную HTML-страницу и отобразит ее содержимое

Как все сценарии, сценарий этого примера помещается в элемент <script>.

```
<script>
   ...
</script>
```

Когда браузер обрабатывает сценарий, он запускает весь код строка за строкой. В данном примере только одна строка:

```
alert("Welcome, JavaScript coder.")
```

В ней применяется встроенная функция JavaScript, называемая `alert`. *Функция* — это программная процедура, состоящая из одной или нескольких строк кода и вы-

полняющая определенную задачу. В языке JavaScript много встроенных функций, но вы сможете создавать и свои собственные.

Функции JavaScript `alert()` требуется некоторая информация, именуемая на языке программистов *аргументом*. В данном случае эта информация представляет собой фрагмент текста, который вы хотите отобразить в окне предупреждения. Если хотите видеть обычное число, например 6, можете ввести его как есть, т. е. не заключая в кавычки. В случае текста нет другого способа сообщить браузеру, где текст начинается и где заканчивается. Поэтому в JavaScript вы помещаете текст в апострофы (') или кавычки ("), как в приведенном примере.

> ***ПРИМЕЧАНИЕ***
>
> На языке программистов отдельные фрагменты текста, используемые в программе, называют *строками* (string). "The friendly fox", "a" и "Rumpelstiltskin" — все это примеры строк.

Итак, рассматриваемый нами простой сценарий выполняет единственное действие — он вызывает функцию JavaScript `alert()`. (Покрутитесь подольше около программистов и скоро узнаете, что "вызов" — предпочитаемый способ обозначения действия, которое запускает функцию.) Функция `alert()` делает все остальное: раскрывает окно подходящего размера и ждет, пока посетитель щелкнет мышью кнопку **OK**.

> ***ПРИМЕЧАНИЕ***
>
> Для написания этого сценария вы должны знать, что существует готовая функция `alert()`, подходящая для вас, сведения об этом можно найти на одном из многочисленных сайтов, обучающих программированию на JavaScript.

На основе уже полученных знаний вы должны уметь изменять данный сценарий:

- для вывода на экран другого сообщения (с помощью изменения аргумента);
- для отображения нескольких окон с предупреждениями, одного за другим (с помощью дополнительных строк кода в вашем сценарии);
- для отображения окна предупреждения до вывода браузером Web-страницы (с помощью изменения размещения блока `<script>`).

Этих знаний недостаточно, чтобы всерьез занять вас, но функция `alert()` показывает, как легко можно начать применять и изменять простой сценарий.

СЕКРЕТЫ РЕМЕСЛА

Борьба с паранойей Internet Explorer

Если вы запустите описанный пример с предупреждением в браузере Firefox, то увидите, что все работает отлично. Если запустить его в Internet Explorer, удовлетворения вы не получите. Вы увидите предупреждение системы безопасности в желтой полосе в верхней части страницы. До тех пор пока вы не щелкните кнопкой мыши эту полосу и не выберите команду **Allow Blocked Content** (Разрешить заблокированный контент), ваш код на JavaScript не запустится.

На первый взгляд предупреждение системы безопасности IE кажется верным способом спугнуть самого смелого Web-посетителя. Но не стоит беспокоиться: это сообщение — всего лишь часть замысловатого метода обработки браузером Internet Explorer

Web-страниц, которые вы храните на своем жестком диске. Если вы обратитесь к той же странице через Web, Internet Explorer не выразит ни малейшего протеста.

Тем не менее предупреждение системы безопасности создает неудобства во время тестирования вашей Web-страницы, т. к. оно вынуждает вас явно сообщать браузеру о том, что запуск JavaScript на странице разрешен. Для того чтобы убрать его совсем, можно предложить Internet Explorer сделать вид, что ваша страница загружена с Web-сервера. Сделать это можно, добавив специальный комментарий, называемый Mark of the Web (Web-метка). Вы помещаете этот комментарий сразу после элемента `<html>`, начинающего вашу страницу.

```
<html>
<!-- saved from url=(0014)about:internet -->
```

Когда IE видит Mark of the Web, он воспринимает страницу, как пришедшую с Web-сервера, пропуская предупреждение системы безопасности и запуская ваш сценарий на JavaScript без колебаний. Для всех остальных браузеров Mark of the Web выглядит как обычный HTML-комментарий.

Сценарии и HTML

Как вы уже видели, для хранения кода на JavaScript в языке HTML применяется блок `<script>`. Но этот блок не что иное, как еще один HTML-элемент, и как все остальные элементы, он должен соблюдать правила языка HTML. Одно из таких правил ограничивает набор разрешенных символов, помещаемых внутрь блока `<script>`. Язык HTML запрещает использование некоторых символов, например печально известные угловые скобки, поскольку в языке HTML у них особое назначение.

К несчастью, специальные символы порой все-таки проникают в код. Их можно заменить символьными последовательностями (см. *разд. "Специальные символы" главы 5*), но это существенно затрудняет чтение, понимание и редактирование вашей программы другими людьми. Гораздо лучше заключить весь сценарий в так называемую *секцию CDATA*, специальную область вашего HTML-документа, которая временно отменяет действие обычных HTML-правил. Далее приведен пример.

```
<script>
//<![CDATA[
  alert("Welcome, JavaScript coder.")
//]]>
</script>
```

Эта методика немного уродлива, но она решает проблему. Лучше всего не слишком задумываться об этом, а просто взять за правило любой длинный блок кода сценария заключать в секцию `CDATA`. Единственное исключение — код на JavaScript, помещенный в отдельный файл (см. *разд. "Внешние файлы сценариев" данной главы*) — ваш браузер не будет интерпретировать его как HTML-код, поэтому его не нужно включать в секцию `CDATA`.

Примечание

Для простоты в этой главе секция `CDATA` применяется, только когда она, безусловно, необходима, другими словами, когда сценарий содержит специальные символы, запрещенные в обычном языке HTML.

НА ПРОФЕССИОНАЛЬНОМ УРОВНЕ

Пробелы и переносы строк в JavaScript

Код на языке JavaScript невероятно терпим к лишним пробелам. В этой главе в большинстве примеров применяются отступы в абзацах, чтобы помочь вам увидеть структуру документа. Но, как и в языке HTML, вам совсем не обязательно вставлять их.

Единственное правило в JavaScript — каждый оператор языка должен находиться в отдельной строке. Это ограничение можно обойти с помощью символа окончания строки, точки с запятой (`;`). Например, посмотрите, как можно сжать три оператора программного кода в одну строку:

```
alert("Hi"); alert("There"); alert("Dude");
```

Каждая точка с запятой обозначает завершение оператора кода. (Это странное соглашение пришло из мира Bizarro с языками C и Java.)

Если вы не хотите оставлять несколько операторов на одной строке, то точки с запятой не нужны. Но вам никто не запрещает вставлять их.

Браузеры, не поддерживающие JavaScript

Очень редко некоторые браузеры распознают элемент `<script>`, но отказываются выполнять код, находящийся в нем. Происходит это, когда браузеры не поддерживают JavaScript (например, малоинтересный текстовый браузер Lynx) или если JavaScript был отключен (что встречается в сверхзащищенных корпоративных средах, но все же очень редко).

Для преодоления подобной ситуации можно применить элемент `<noscript>`, позволяющий предложить замещающий HTML-контент. Вы помещаете элемент `<noscript>` сразу за элементом `<script>`. Далее приведен пример, отображающий абзац текста в браузерах, не имеющих поддержки JavaScript.

```
<script>
  alert("Welcome, JavaScript coder.")
</script>
<noscript>
  <p>Welcome, non-JavaScript-enabled browser.</p>
</noscript>
```

Переменные

В любом языке программирования есть *переменные* — временные контейнеры для хранения важных данных. В переменных могут храниться числа, объекты или фрагменты текста. Прочитав этот раздел, вы поймете, что переменные играют ключевую роль во многих сценариях и являются мощным средством в арсенале любого программиста.

Объявление переменных

Для создания переменной в JavaScript применяется ключевое слово `var`, за которым следует имя переменной. Вы можете выбирать любое имя, имеющее для вас смысл,

если вы последовательны (и избегаете пробелов и специальных символов). В приведенном примере создается переменная myMessage:

```
var myMessage
```

Для запоминания информации в переменной используется знак равенства (=), копирующий данные, находящиеся справа, в переменную, расположенную слева. Далее приведен пример, помещающий в переменную myMessage фрагмент текста.

```
myMessage = "Everybody loves variables"
```

Помните, что строки всегда следует заключать в кавычки. Если же вы хотите скопировать в переменную *число*, кавычки не нужны.

```
myNumber = 27.3
```

> ***ПРИМЕЧАНИЕ***
>
> Имена переменных JavaScript чувствительны к регистру, т. е. переменная myMessage отличается от переменной MyMessage. Если вы будете использовать их как взаимозаменяемые, то получите в результате сообщение об ошибке (если у вас хороший браузер) или аномалию на странице (что обычно и происходит).

Зачастую возникает желание создать переменную и запомнить в ней полезные данные в одном операторе. JavaScript позволяет это сделать, если вы во время объявления переменной сразу же за ее именем поставите знак равенства.

```
var myMessage = "Everybody loves variables"
```

Чтобы внести немного путаницы, JavaScript разрешает ссылаться на переменные, которые еще не объявлены. Использование этой возможности — крайне плохой стиль, который с большой вероятностью приводит к всевозможным проблемам. Но знать о нем следует, поскольку такие необъявленные переменные — источник множества неожиданных ошибок.

Изменение переменных

Одна из самых полезных возможностей, предоставляемых числовыми переменными, — выполнение разных операций над ними для изменения ваших данных. Например, для математических вычислений можно применять арифметические операции:

```
var myNumber = (10 + 5) * 2 / 5
```

Эти вычисления выполняются с соблюдением стандартного порядка операций (операции в скобках первые, затем умножение и деление, потом сложение и вычитание). Результат приведенного выражения равен 6.

Вы также можете использовать операции для объединения фрагментов текста в одну длинную строку. В этом случае применяется знак плюс (+).

```
var firstName = "Sarah"
var lastName = "Smithers"
var fullName = firstName + " " + lastName
```

Теперь переменная fullName содержит текст "Sarah Smithers". (Текст " " в примере заставляет JavaScript вставить пробел между именем и фамилией.)

Пример с использованием переменных

Для того чтобы узнать все о переменных, вам придется прочесть толстый фолиант, но многое из того, что следует знать, можно извлечь из простого примера. В следующем сценарии на Web-страницу вставляется текущая дата. В примере пронумерованы все строки сценария, чтобы легче было на них ссылаться.

```
<!DOCTYPE html>

<html>

<head>
  <title>JavaScript Test</title>
</head>

<body>
  <h1>What Day Is It?</h1>
  <p>This page uses JavaScript.</p>
  <p>
  <script>
1       var currentDate = new Date()
2       var message = "The current date is: "
3       message = message + currentDate.toDateString()
4       document.write(message)
  </script>
  </p>
</body>
</html>
```

Вот что происходит строка за строкой.

1. Эта строка создает новую переменную с именем `currentDate`. Она заполняет переменную `currentDate` объектом `new Date` (см. строку 3 ниже). Вы узнаете, что JavaScript создает объект, когда встречается ключевое слово `new`. (Вы узнаете больше об объектах в *разд. "HTML-объекты" данной главы*; сейчас достаточно знать, что объекты включают в себя встроенные функции, которые действуют более или менее так же, как функции, с которыми вы познакомились раньше.)
2. Эта строка создает новую переменную с именем `message` и заполняет ее началом предложения, объявляющего дату.
3. Эта строка добавляет новый текст в конец сообщения, созданного вами в строке 2. Текст берется из объекта `currentDate`. Хитрость в том, что объект `currentDate` включает в себя встроенную функцию `toDateString()`, которая преобразует информацию о дате, полученную с вашего компьютера, во фрагмент текста, подходящий для отображения в браузере (рис. 15.3). Подробную информацию об этом можно получить, прочитав хорошее справочное руководство по JavaScript.

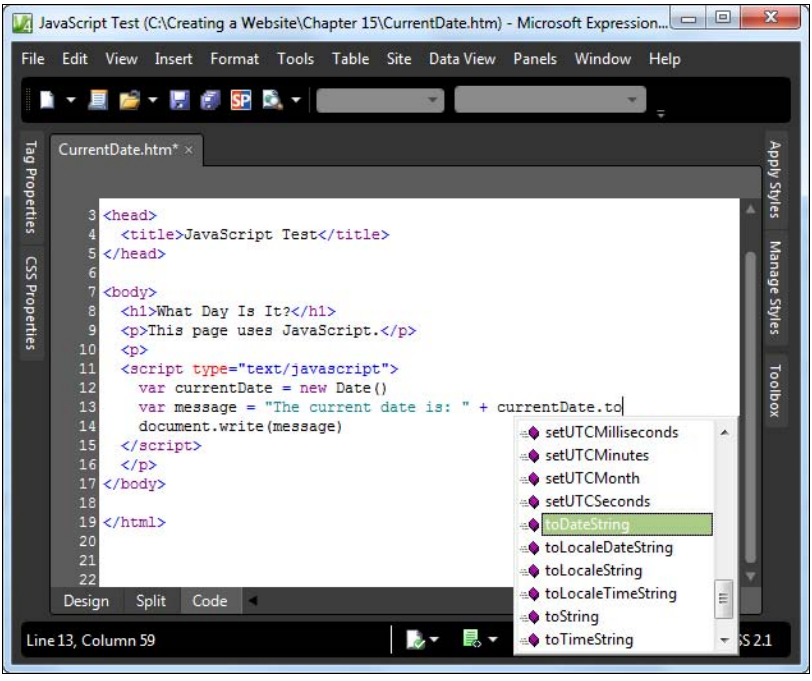


Рис. 15.3. Некоторые редакторы Web-страниц могут помочь вам во время написания кода на JavaScript. Например, Expression Web выводит на экран раскрывающееся меню с полным набором функций, предоставляемых объектом. При первом обращении к объекту `Date`, возможно, этого перечня вам будет недостаточно для выбора способа работы с объектом, но впоследствии он очень поможет освежить память

4. Эта строка использует объект JavaScript `document`, у которого есть функция `write()`. Функция `write()` отображает на Web-странице в текущем месте фрагмент текста. Конечный результат — страница отображающая ваш вежливый ответ (рис. 15.4).



Рис. 15.4. Команда `document.write()` вставляет текст прямо на страницу, в то место, где вы поместили блок сценария. В данном случае команда отображает текущую дату

Сценарии могут выполнять гораздо более сложную работу, чем эта. Например, они могут применять циклы для многократного повторения одного и того же действия или принимать решения, используя условную логику. Вы увидите примеры использования этих методов позже в этой главе, но не получите полного описания языка JavaScript — для этого требуется отдельная книга, посвященная языку. Если хотите узнать больше, просмотрите, например, книгу "JavaScript: The Missing Manual".

Функции

До сих пор вы видели сценарии, состоящие всего из нескольких строк кода. Более реальные сценарии на JavaScript могут достигать десятков строк и, если вы не будете внимательны, могут превратиться в причудливый клубок, который затруднит редактирование остальной части вашей страницы. Для контроля над хаосом умные программисты почти всегда используют *пользовательские функции*.

Функция — это набор инструкций кода, которые вы собираете вместе и наделяете именем. Отчасти функции похожи на маленькие программки, т. к. могут выполнять ряд операций. Основное достоинство функций в том, что вы создаете их только один раз, а применять можете снова и снова.

Объявление функции

Создавая функцию на JavaScript, сначала решите, что ваша функция будет делать (например, отображать предупреждающее сообщение), и затем выберите для нее подходящее имя (например, `ShowAlertBox`). Как и большинство компонентов в мире программирования, имена функций не могут содержать пробелы или специальные символы.

Снабженные этой информацией, вы можете поместить блок `<script>` на вашу страницу, в раздел `<head>`. Но этот блок `<script>` немного отличается от блоков сценариев, которые вы видели до сих пор. В нем есть полноценная функция, отображающая окно с заранее заданным сообщением.

```
<script>
  function ShowAlertBox() {
  alert("I'm a function.")
  }
</script>
```

Разобьем сценарий на фрагменты, чтобы легче понять, что в нем происходит, и рассмотрим их один за другим.

Любое объявление функции начинается со слова `function`, которое сообщает JavaScript, что вы намерены сделать.

```
function
```

Затем вы присваиваете имя функции и добавляете две скобки. Скобки применяются для передачи вашей функции дополнительной информации, как вы скоро увидите.

```
function ShowAlertBox()
```

В этот момент *объявление* функции заканчивается. Остается только вставить в функцию код, который заставит ее работать. Для этого необходимы забавные фигурные скобки, показанные в функции создания окна предупреждения, приведенной ранее. Скобка `{` обозначает начало кода вашей функции, а скобка `}` — его конец. Между ними можно поместить сколько угодно строк кода.

Одна из хитростей в написании функций заключается в том, что JavaScript устанавливает нечеткие правила для переносов строк. Это означает, что вы можете создать эквивалентную функцию на JavaScript и поместить фигурные скобки в отдельных строках, например, так:

```
<script>
  function ShowAlertBox()
  {
    alert("I'm a function.")
  }
</script>
```

Но не волнуйтесь — обе функции будут выполняться одинаково.

> ***Совет***
>
> В один блок `<script>` можно поместить столько функций, сколько захотите. Просто добавляйте их одну за другой.

Вызов функции

Создание функции — это полдела. Сами по себе функции ничего не делают. Вы должны *вызвать* функцию где-нибудь на своей странице, чтобы выполнить ее код. Для вызова функции используется ее имя, за которым следуют скобки.

```
ShowAlertBox()
```

> ***Примечание***
>
> Не забывайте вставлять скобки после имени функции. Иначе браузеры решат, что вы пытаетесь использовать переменную, а не вызывать функцию.

Вы можете вызвать функцию в любом месте обычного кода на JavaScript. Например, далее приведен сценарий, отображающий предупреждение три раза в одной строке, чтобы по-настоящему надоесть вашим посетителям.

```
<script>
  ShowAlert()
  ShowAlert()
  ShowAlert()
</script>
```

Таким же способом вы раньше вызывали функцию `alert()`. Разница в том, что `alert()` встроена в JavaScript, а `ShowAlertBox()` — нечто, созданное вами самостоятельно. Кроме того, функции `alert()` требуется один аргумент, а функции `ShowAlertBox()` — ни одного.

Функции, получающие данные

Функция `ShowAlertBox()` совершенна в своей простоте. Вы просто вызываете ее, и она отображает предупреждающее сообщение, которое вы задали. Большинство функций выполняет не столь легкую работу. Поэтому во многих случаях требуется передавать конкретную информацию в функцию или принимать от нее результаты и использовать их в другой операции.

Представьте, что вы хотите вывести на экран приветствие с некоторой стандартной информацией, например текущей датой. Но вы также хотите получить возможность при каждом вызове функции изменять часть сообщения, подставляя собственные остроумные слова. Для этого необходимо найти способ вызывать функцию *и* передавать ей текстовую строку с вашим сообщением.

Для решения проблемы можно создать функцию `ShowAlertBox()`, которая принимает один *параметр*. Этот параметр задает текст, который вы хотите встроить в ваше приветствие. Для вставки параметра сначала нужно дать ему имя, например, так:

```
function ShowAlertBox(customMessage) {
   ...
}
```

> ***ПРИМЕЧАНИЕ***
>
> Строго говоря, порции информации, получаемые функцией (в данном случае `customMessage`), называются *параметрами*. Когда вы вызываете функцию, имеющую параметры, вы передаете ей по одному значению для каждого параметра. Значение, предоставляемое вами, называется *аргументом*.
>
> Другими словами, одна и та же порция информации называется параметром с точки зрения функции и аргументом с точки зрения кода, вызывающего функцию. Порой вы встретите взаимозаменяемое употребление терминов "параметр" и "аргумент", но теперь вы знаете официальную разницу между ними.

На количество параметров функции нет ограничений. Вы только должны отделять каждый параметр запятой. Далее приведен пример функции `ShowAlertBox()` с тремя параметрами, названными `messageLine1`, `messageLine2` и `messageLine3`.

```
function ShowAlertBox(messageLine1, messageLine2, messageLine3) {
     ...
}
```

Далее приведен еще один пример полноценной функции `ShowAlertBox()`. Она принимает единственный параметр, названный `customMessage`, и использует его для формирования текста, который отображает в окне предупреждения.

```
<script>
1  function ShowAlertBox(customMessage)
2  {
3     // Получить дату
4     var currentDate = new Date()
5
6     // Сформировать полное сообщение
7     var fullMessage = "** IMPORTANT BULLETIN **\n\n"
```

```
8     fullMessage += customMessage + "\n\n"
9     fullMessage += "Generated at: " + currentDate.toTimeString() + "\n"
10    fullMessage += "This message courtesy of MagicMedia Inc."
11
12    // Отобразить сообщение
13    alert(fullMessage)
14  }
</script>
```

Далее приведено несколько замечаний, которые помогут вам одолеть код.

- Любая строка, начинающаяся с символов `//`, — это комментарий (см. строки 3 и 6). Хорошие программисты вставляют много комментариев, чтобы помочь другим понять, как работает функция (и помочь себе вспомнить, что они делали во время ночного написания программы). Браузер игнорирует эти строки.
- Для вставки переносов строк в окно предупреждения используются символы `\n` (строки 7, 8 и 9). Каждая последовательность `\n` эквивалентна одному переносу строки. (Это правило действует только в окнах сообщений. Если хотите вставить перенос строки в HTML-код, используйте уже знакомый элемент `<br />`.)
- Для формирования текста в переменной `fullMessage` (строки 7 и 10) используется сокращенная форма в виде операции `+=`. Эта операция берет то, что располагается справа от знака равенства, и вставляет его в конец переменной, находящейся слева. Другими словами, строка

  ```
  8     fullMessage += customMessage + "\n\n"
  ```

  эквивалентна следующей более длинной строке:

  ```
  8     fullMessage = fullMessage + customMessage + "\n\n"
  ```

Применять эту функцию просто. Только помните, что, когда вы вызываете ее, необходимо задать аргументы для каждого параметра, разделяя их запятыми. В случае функции `ShowAlertBox()` вам следует указать единственное значение для переменной `customMessage`. Далее приведен пример.

```
<script>
ShowAlertBox("This web page includes JavaScript functions.")
</script>
```

На рис. 15.5 показан результат этого сценария.

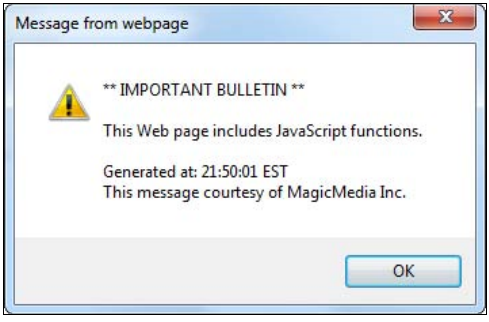


Рис. 15.5. Это сообщение сформировано из нескольких фрагментов текста, один из которых вы предоставили функции `ShowAlertBox()` как аргумент

Функции, возвращающие информацию

Параметры позволяют передавать информацию в функцию. Но вы также можете создавать функции, которые передают информацию обратно в код сценария, вызвавшего их. Ключевой компонент в этом случае — команда `return`, которую вы помещаете в конце вашей функции. Команда `return` немедленно завершает функцию и отделяет любую информацию, сгенерированную вашей функцией.

Конечно, сложные функции способны и принимать и возвращать информацию. Например, далее приведена функция, перемножающая два числа (`numberA` и `numberB` — параметры) и возвращающая результат всем, кого это интересует.

```
<script>
  function MultiplyNumbers(numberA, numberB)
  {
    return numberA * numberB
  }
</script>
```

Далее показано, как применять эту функцию в любом месте вашей Web-страницы.

```
<p>The product of 3202 and 23405 is
<script>
  var product = MultiplyNumbers(3202, 23405)
  document.write(product)
</script>
</p>
```

Этот HTML-код содержит единственную строку текста, за которой следует код сценария. В нем вызывается функция `MultiplyNumbers()`, получается результат (число 74942810) и переносится в переменную `product` для последующего использования. Затем код применяет команду `document.write()` для отображения на странице содержимого переменной `product`. Конечный результат — следующий текстовый абзац.

```
The product of 3202 and 23405 is 74942810
```

Для использования типового сценария, полученного из Web-пространства, вам придется скопировать на вашу страницу одну или несколько функций. Эти функции, скорее всего, будут выглядеть гораздо более сложными, чем те, что вы видели до сих пор. Но теперь, понимая базовую структуру функции, вы сможете разобраться в коде, чтобы составить общее представление о том, что в нем делается (или по крайней мере точно определить, где сценарий дает сбой).

Внешние файлы сценариев

Многократное применение сценариев на странице удобно, но знаете ли вы, что можно совместно использовать сценарии на разных страницах и даже на разных Web-сайтах. Вы просто помещаете ваш сценарий во внешний файл и затем ссылаетесь на него. Эта процедура аналогична заданию ссылки на внешние таблицы стилей, о котором вы узнали в *главе 6*.

Представьте себе, что вы довели до совершенства функцию `ShowAlertBox()`, так что она выполняет сложную задачу именно тем способом, каким вы хотите, но требует пары десятков строк кода. Для облегчения жизни и вашего HTML-документа вы создаете новый файл для хранения сценария.

Файлы сценариев — всегда обычные текстовые файлы, как правило, у них расширение js (сокращение от JavaScript). Вы помещаете весь ваш код внутрь файла сценария, но не вставляете элемент `<script>`. Например, вы можете создать файл на JavaScript с именем ShowAlert.js.

```
function ShowAlertBox()
{
  alert("This function is in an external file.")
}
```

Теперь сохраните файл и поместите его в ту же папку, что и вашу Web-страницу. На Web-странице определите блок сценария, но не вставляйте его код. Вместо этого добавьте атрибут `src` и задайте файл сценария, на который хотите сослаться.

```
<script src="ShowAlert.js">
</script>
```

Когда браузер дойдет до блока сценария, он запросит файл ShowAlert.js и будет обрабатывать его так, как будто код находится прямо на странице. Далее приведена полная тестовая страница, использующая файл ShowAlert.js. Сценарий в теле страницы вызывает функцию `ShowAlertBox()`.

```
<!DOCTYPE html>

<html>

<head>
  <title>Show Alert</title>
  <!-- Make all the functions in the ShowAlert.js file
    available in this page. Notice there's no actual content here. -->
  <script src="ShowAlert.js">
  </script>
</head>

<body>
  <!-- Test out one of the functions. -->
  <script>
    ShowAlertBox()
  </script>
</body>

</html>
```

Встроенный и внешний сценарии действуют одинаково. Но хранение ваших сценариев в отдельных файлах помогает поддерживать организацию вашего Web-сайта и облегчает повторное применение сценариев на нескольких страницах. На самом

деле вы даже можете ссылаться на функции JavaScript, размещенные на другом Web-сайте, — только помните о том, что атрибут `src` в блоке `<script>` должен указывать на реальный URL (например, **http://SuperScriptSite.com/ShowAlert.js**), а не просто на имя файла. Конечно, это рискованный вариант, потому что владелец Web-сайта мог переименовать, переместить или удалить файл на JavaScript. Если вы действительно хотите использовать код, гораздо лучше скопировать его на свой сервер, чтобы избежать подобных проблем.

ПРИМЕЧАНИЕ

Применение отдельных файлов сценариев не повышает вашу безопасность ни на йоту. Поскольку любой может запросить ваш файл сценария, сообразительный Web-посетитель может выяснить, какие сценарии применяются на вашей странице, и просмотреть их. Поэтому никогда не вставляйте в сценарий никакого кода или секретных деталей, которые вы хотите скрыть от мирового сообщества.

Dynamic HTML

Язык JavaScript претерпел небольшую революцию в конце 1990 гг., включив в себя поддержку ряда средств, именуемых Dynamic HTML (динамический HTML или сокращенно DHTML). Dynamic HTML — это не новая технология, а слияние следующих трех разных компонентов:

- языков сценариев, таких как JavaScript, позволяющих писать код;
- стандарта CSS (Cascading Style Sheet, Каскадные таблицы стилей), позволяющего управлять положением и внешним видом HTML-элемента;
- и объектной модели HTML-документа (Document Object Model или DOM), позволяющей трактовать HTML-страницу как набор объектов.

Последний пункт особенно важен. Динамический HTML представляет страницу как набор *объектов*. Это радикальный поворот в Web-программировании. DHTML трактует каждый HTML-элемент, включая изображения, ссылки и даже непритязательный абзац, как отдельный программный компонент, с которым умеет работать код на JavaScript. С помощью этих объектов вы можете менять внешний вид каждого элемента и даже его положение на странице при отображении в браузере.

HTML-объекты

Ясно, что динамический HTML требует полностью нового способа представления о дизайне Web-страницы. Ваши сценарии уже не смотрят на Web-страницу как на статичный блок HTML-кода. Они видят комбинацию объектов.

НА ПРОФЕССИОНАЛЬНОМ УРОВНЕ

Представление об объектах

Во многих языках программирования, включая JavaScript, все крутится вокруг объектов. Что же такое объект на самом деле?

В мире программирования объект — это всего лишь удобный способ группирования нескольких связанных свойств или данных. Например, вы хотите изменять изображе-

ние на Web-странице, заданное в элементе <img> (что полезно, когда пишется сценарий, передающий серию изображений). Самый легкий способ взаимодействия с элементом <img> в языке JavaScript — применение соответствующего объекта image. Фактически объект image — это контейнер, содержащий всякого рода потенциально полезную информацию о том, что происходит в элементе <img> (включая размеры изображения, его положение, имя файла, связанного с ним, и т. д.). Объект image также предоставляет метод манипулирования элементом <img>, т. е. изменения некоторых или всех его характеристик.

Например, для получения информации об изображении вы можете использовать объект image следующим образом:

```
document.write("The tooltip says" + image.title)
```

Вы даже можете изменить одну из его характеристик. Например, можно изменить действительное изображение, которое показывает элемент <img>, с помощью следующего кода:

```
image.src = "newpic.jpg"
```

О том, что обрабатывается объект, можно узнать по наличию точки (.) в строке кода. Точка отделяет имя переменной (первая часть) от одной из встроенных функций, предоставляемых объектом (они называются *методами*), или от связанных переменных (называемых *свойствами*). Методы и свойства всегда помещают после точки.

В предыдущих примерах src и title — два свойства объекта image. Другими словами, написать код image.src = "newpic.jpg" все равно, что сказать: "Привет, мистер Объект по имени Image! У меня есть для Вас новое изображение. Измените свое свойство src, так чтобы оно указывало на файл newpic.jpg".

Программисты пользуются объектами очень давно, т. к. это прекрасный способ концептуальной организации программного кода (не говоря уж об удобствах совместного и многократного применения). Вы, возможно, поймете это не сразу, но работать с объектом image намного легче, чем запоминать несколько десятков разных команд, обрабатывающих изображение как таковое.

Прежде чем начать манипулировать объектом на Web-странице, нужно найти способ его однозначно идентифицировать. Как только вы это сделаете, ваш код сможет найти нужный ему объект. Наилучший метод идентификации — атрибут id. Вставьте этот атрибут в начальный тег элемента, которым хотите управлять, и подберите уникальное имя, как показано далее.

```
<h1 id="PageTitle">Welcome to My Page</h1>
```

После того как объекту присвоен уникальный ID, можно легко найти объект в вашем программном коде и заставить JavaScript работать с ним.

У JavaScript есть хитрый прием для обнаружения объекта: метод document.getElementById(). Собственно говоря, document — это объект, представляющий весь HTML-документ. Он всегда есть в наличии, и вы можете применять его, когда захотите. Этот объект, как любой объект достойный своего имени, предоставляет ряд полезных свойств и методов. Метод getElementById() — один из самых крутых: он сканирует страницу в поисках конкретного HTML-элемента.

Примечание

В *разд. "Пример с использованием переменных"* вы видели объект document, работающий над другой задачей, — отображением информации на Web-странице. Для ее исполнения сценарий использовал метод write() объекта document.

При вызове метода `document.getElementById()` вы задаете ID HTML-элемента, который ищите. Далее приведен пример, в котором разыскивается объект для HTML-элемента с идентификатором `PageTitle`.

```
var titleObject = document.getElementById("PageTitle")
```

В этой строке кода находится объект для элемента `<h1>`, показанного ранее, и сохраняется в переменной `titleObject`. Сохранив объект в переменной, вы сможете выполнить серию операций над ним, без повторного поиска объекта.

Итак, что же именно вы можете делать с HTML-объектами? До известной степени ответ зависит от типа элемента, с которым вы работаете. Например, если у вас гиперссылка, можно изменить ее URL. Если вы работаете с изображением, можно изменить его источник. Но есть ряд действий, которые можно совершать почти со всеми HTML-элементами, например, изменять их стиль или модифицировать текст, находящийся между начальным и конечным тегами. Как вы увидите, эти приемы очень полезны и помогают сделать вашу страницу динамичнее. Например, можно изменить страницу, когда посетитель совершит какое-то действие, щелкнет кнопкой мыши ссылку. Такое взаимодействие заставляет посетителей думать, что они имеют дело с интеллектуальной, способной реагировать программой, а не с простой инертной Web-страницей.

Далее показано, как изменить текст внутри уже упоминавшегося элемента `<h1>`.

```
titleObject.innerHTML = "This Page Is Dynamic"
```

Если вы вставите эту строку в сценарий, текст заголовка изменится, как только браузер запустит сценарий.

В приведенном примере используется свойство, названное `innerHTML` и задающее контент, вложенный внутрь элемента (в данном случае элемента `<title>`). Как и все свойства, `innerHTML` — только одна характеристика HTML-объекта, которую можно изменять. Для записи операторов кода, подобных приведенному в примере, необходимо знать, с какими свойствами JavaScript разрешает вам работать. Понятно, что некоторые свойства применимы только к определенным HTML-элементам, как, например, атрибут `src` изображения. Современные браузеры хвалятся гигантскими каталогами свойств DOM, которые можно применять практически к любому HTML-элементу. В табл. 15.1 перечислены некоторые самые полезные из них.

> ***Совет***
>
> Для получения списка свойств, поддерживаемых конкретным HTML-элементом, обратитесь к справочному руководству на странице **www.w3schools.com/htmldom/dom_reference.asp**.

В настоящий момент приведенный ранее пример выполняется в два шага (получается объект, а затем он изменяется). Несмотря на то что этот двухшаговый подход, вероятно, самый понятный, есть возможность соединить оба шага в одной строке, что сценарии часто и делают. Далее приведен пример.

```
document.getElementById("PageTitle").innerHTML = "This Page Is Dynamic"
```

Имейте в виду, что преимущество первоначального получения объекта заключается в том, что вы можете изменить несколько свойств по очереди, и при этом не требуется каждый раз искать HTML-объект с помощью метода `getElementById()`.

Таблица 15.1. *Общие свойства HTML-объектов*

Свойство	Описание
`className`	Позволяет извлечь или задать атрибут `class` (см. *разд. "Селекторы классов" главы 6*). Другими словами, это свойство определяет, какой стиль (если есть) использует этот элемент. Конечно, определить стиль вы должны во встроенной или связанной таблице стилей или все закончится уродливым стандартным форматированием
`innerHTML`	Позволяет считать или изменить HTML-код внутри элемента. Это свойство безумно полезно, но у него есть две хитрости. Во-первых, его можно применять к любому HTML-содержимому, включая текст и теги. Поэтому, если вы хотите поместить внутрь абзаца жирный текст, можно задать `innerHTML` для `<b>Hi</b>`. Специальные символы не приветствуются — их нужно заменить символьными последовательностями, описанными в *разд. "Специальные символы" главы 5*. Во-вторых, когда вы задаете `innerHTML`, в указанном элементе заменяется все содержимое, включая другие HTML-элементы. Поэтому, если вы задаете `innerHTML` для элемента `<div>`, содержащего несколько абзацев и изображений, все эти элементы заменяются новым содержимым. Если вы хотите изменить фрагмент абзаца, заключите его в элемент `<span>`
`parentElement`	Предоставляет HTML-объект для элемента, содержащего данный элемент. Например, если текущий элемент в абзаце — элемент `<b>`, это свойство возвращает объект для элемента `<p>`. Когда вы получите этот объект, можно модифицировать абзац. С помощью данного приема (и аналогичных приемов в Dynamic HTML) можно переходить от одного элемента к другому
`style`	Собирает вместе все атрибуты CSS, определяющие внешний вид HTML-элемента. Строго говоря, свойство `style` возвращает объект с полноценным стилем, и вам нужно вставить еще одну точку (.) и имя атрибута стиля, который хотите изменить, например, `myObject.style.fontSize`. Вы можете применять объект стиля для задания цветов, границ, шрифтов и даже позиционирования
`tagName`	Предоставляет имя HTML-элемента для данного объекта, без угловых скобок. Например, если текущий объект представляет элемент `<img>`, это свойство вернет текст `img`

Использование HTML-объектов в сценарии

Самый легкий способ поближе познакомиться с работой HTML-объектов — рассмотреть пример. Web-страницы, показанные на рис. 15.6, содержат абзац, который постоянно увеличивается и уменьшается, когда ваш код периодически изменяет размер шрифта.

Алгоритм работы этого примера очень интересен. Прежде всего, вы определяете две переменные в разделе `<head>` исходного HTML-документа. Переменная `size` отслеживает текущий размер шрифта текста (начальный размер 10 пикселов). Пе-

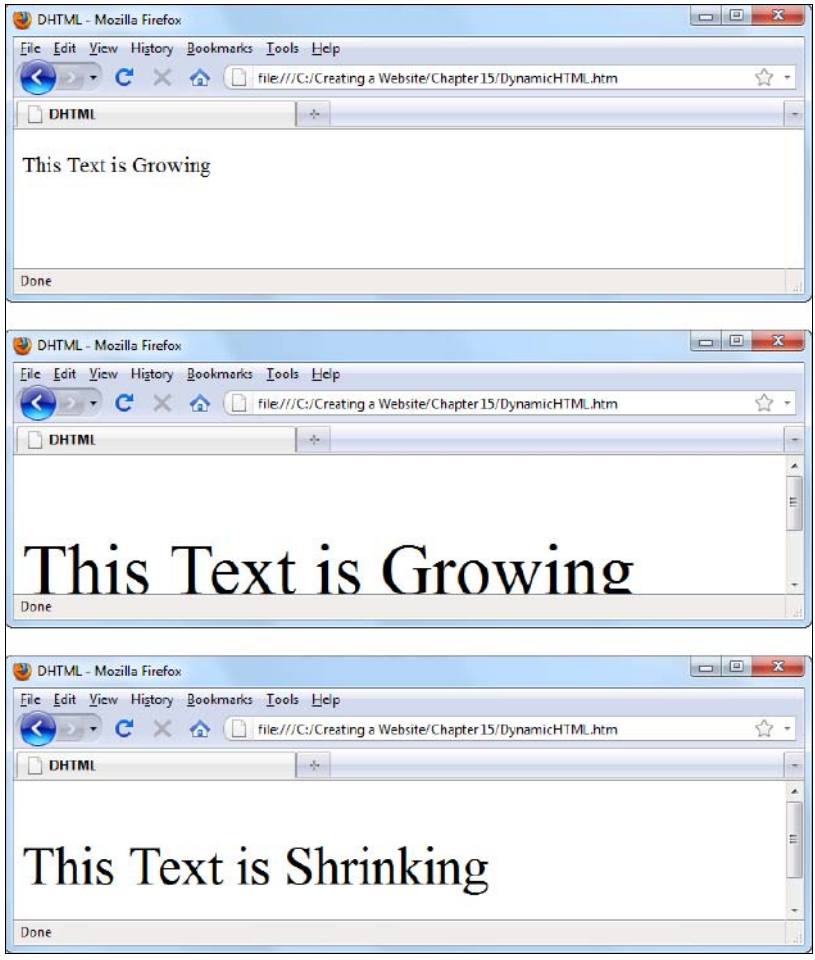


Рис. 15.6. Если вы посмотрите на этот заголовок в реальном Web-браузере, вы увидите, что текст все время меняет размер, привлекая к себе внимание

ременная `growIncrement` определяет, насколько меняется размер шрифта при каждом выполнении кода браузером (первоначально он увеличивается каждый раз на 2 пиксела).

```
<!DOCTYPE html>

<html>
<head>
  <title>Dynamic HTML</title>
  <script>
//<![CDATA[
    // Текущий размер шрифта
    var size = 10
    // Величина, на которую изменяется размер
    var growIncrement = 2
```

Примечание

Этот пример сценария заключен в блок CDATA (см. *разд. "Сценарии и HTML" данной главы*), потому что использует угловые скобки для выполнения сравнения чисел.

Затем в сценарии описывается функция ChangeFont(). Эта функция извлекает HTML-объект, в данном случае элемент <p>, содержащий текст, который будет увеличиваться и уменьшаться. И снова работу выполняет функция getElementById().

```
function ChangeFont() {
  // Находит объект, представляющий абзац,
  // размер шрифта которого хотите изменить
  var paragraph = document.getElementById("animatedParagraph")
```

Теперь, используя переменные size и growIncrement, вы определяете переменную, выполняющую вычисления для определения нового размера текста абзаца:

```
size = size + growIncrement
```

В данном примере знак + выполняет арифметическое сложение, поскольку обе переменные, size и growIncrement, содержат числа.

С помощью свойства paragraph.style.fontSize очень легко установить новый размер шрифта. Только присоедините в конец буквы px, чтобы показать, что ваша характеристика стиля измеряется в пикселах:

```
paragraph.style.fontSize = size + "px"
```

Если этот код будет выполняться постоянно, текст вырастет до немыслимых размеров, и вы не увидите ни одного символа на странице. Для того чтобы помешать этому, вставьте в код предохранительный клапан.

Допустим, вы решили, что, когда размер шрифта достигнет 100 пикселов, нужно прекратить увеличение текста и перейти к его уменьшению. Для этого вы в сценарии делаете переменную growIncrement, равной –2, когда размер шрифта текста достигает 100. С этого момента при каждом выполнении сценария текст начнет уменьшаться на 2 пиксела. Для того чтобы определить, когда сообщение станет слишком большим, воспользуйтесь предупредительностью условной логики оператора if. Вот как это выглядит:

```
// Решает, изменять ли направление
// с увеличения на уменьшение (или наоборот)
if (size > 100) {
  paragraph.innerHTML = "This Text is Shrinking"
  growIncrement = -2
}
```

Конечно, вы не хотите, чтобы уменьшение тоже продолжалось вечно. Поэтому имеет смысл добавить еще одну последнюю проверку, определяющую, не стал ли размер шрифта текста меньше 10 пикселов. Если это так, сценарий возвращается к увеличению размера шрифта, установив снова переменную growIncrement, равной 2.

```
if (size < 10) {
  paragraph.innerHTML = "This Text is Growing"
  growIncrement = 2
}
```

Теперь пришло время по-настоящему хитроумной штучки. У JavaScript есть функция `setTimeout()`, которая позволяет проинструктировать браузер: "Вызови эту функцию, но перед этим подожди немного". В данном примере функция `setTimeout()` просит браузер вызывать метод `ChangeFont()` снова через 100 миллисекунд (одну десятую секунды).

```
    setTimeout("ChangeFont()", 100)

//]]>
  </script>
</head>
```

Поскольку функция `ChangeFont()` всегда использует `setTimeout()` для повторного вызова себя самой, уменьшение и увеличение никогда не заканчивается. Но вы можете изменить это поведение. Например, добавить условную логику, так что JavaScript вызовет метод `setTimeout()` только определенное число раз.

Последняя деталь — раздел `<body>`, который содержит реальный абзац, размер которого вы меняете, и сценарий, который вызывает функцию `ChangeFont()` в первый раз, запуская весь процесс.

```
<body>
  <p id="animatedParagraph">This Text is Growing</p>
  <script>
    ChangeFont()
  </script>
</body>

</html>
```

Хотя пример с изменением размера шрифта абсолютно бесполезен, алгоритм может служить основой для гораздо более впечатляющих сценариев (для получения целого сценария и работы с ним загрузите его со страницы **Missing CD** по адресу **www.missingmanuals.com/cds/caw3**). Легко можно найти сценарии, создающие различные анимационные эффекты для текста, например, заставляющие его сверкать, выплывать из-за бокового края страницы или отображать поочередно по одной букве в стиле машинописного набора.

События

Самые впечатляющие страницы, усиленные JavaScript, — динамические, т. е. они выполняют разные действия в процессе взаимодействия с ними вашего посетителя (перемещающего мышь, набирающего текст, щелкающего мышью различные элементы и т. д.). Динамическая страница гораздо эффектнее обычной HTML-страницы, которая появляется в браузере за один цикл и сидит в нем неподвижно.

Для создания динамических страниц вы программируете их реакцию на *события* JavaScript. События — это уведомления, которые генерирует HTML-элемент при возникновении определенных ситуаций.

Например, JavaScript приписывает каждому элементу гиперссылки `<a>` событие, названное `onmouseover` (сжатая форма словосочетания "on mouse over"). Как видно из названия, это событие происходит (*возбуждается*, используя программистский жаргон), когда посетитель наводит указатель мыши на HTML-элемент, такой как абзац, ссылка, изображение, ячейка таблицы или текстовое поле. Это действие запускает событие `onmouseover`, и ваш код начинает действовать.

Далее приведен пример, в котором выводится на экран предупреждающее сообщение, если посетитель наведет указатель мыши на ссылку.

```
<!DOCTYPE html>

<html>

<head>
  <title>JavaScript Test</title>
  <meta http-equiv="Content-Script-Type" content="text/javascript" />
</head>

<body>
  <h1>You Will Be Wowed (Again)</h1>
  <p>When you hover over <a href="SomePage.htm"
  onmouseover="alert('Colorless green ideas sleep furiously.')">this
     link</a>
  you'll see a secret message.
  </p>
</body>
</html>
```

Когда вы пишете код, описывающий реакцию на событие, совсем необязательно вставлять блок `<script>` (хотя неплохо применять его в любом случае, как будет показано в следующем разделе). Вам достаточно заключить код в кавычки и поместить его следом за атрибутом события:

```
<a onmouseover="[Код располагается здесь]">...</a>
```

Следует иметь в виду одну деталь. В данном примере текстовое значение (`'Colorless green ideas sleep furiously.'`) заключено в апострофы вместо кавычек. Дело в том, что в самом атрибуте события используются кавычки, а одновременное применение кавычек для разных целей сильно озадачит ваш браузер.

Когда вы присоединяете код к событию, ваш браузер полагает, что вы используете JavaScript, вне всяких сомнений самый популярный язык сценариев. Но если следовать самым строгим правилам HTML, необходимо явно указать выбранный вами язык, добавив в раздел `<head>` вашей страницы следующий элемент `<meta>`:

```
<meta http-equiv="Content-Script-Type" content="text/javascript" />
```

На рис. 15.7 показан результат выполнения этого сценария и наведение указателя мыши на ссылку.

Для эффективного применения событий нужно знать, какие события поддерживает JavaScript. Кроме того, вы должны выяснить, какие события порождаются какими HTML-элементами. В табл. 15.2 приведен перечень часто используемых событий и HTML-элементов, к которым они применимы (более полное справочное руководство можно найти по адресу **www.w3schools.com/jsref**).

В следующих разделах вы познакомитесь с двумя самыми популярными сценариями, использующими некоторые из этих событий.

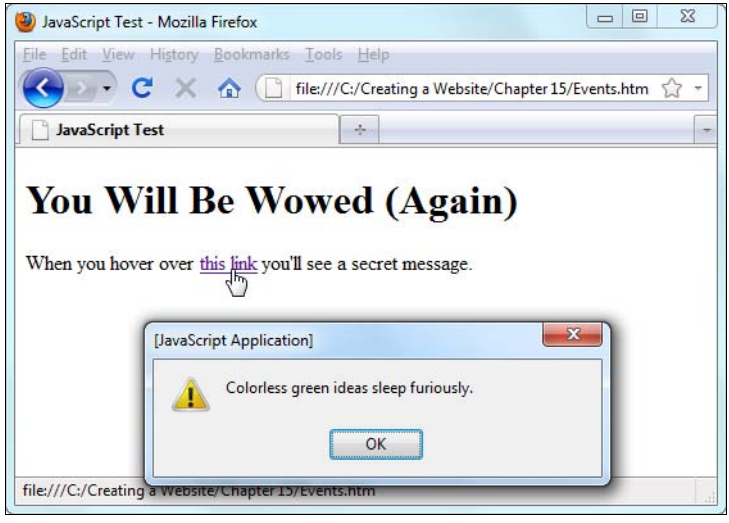


Рис. 15.7. В данном примере окно предупреждения не появится до тех пор, пока вы не укажите мышью на ссылку

Таблица 15.2. *Часто используемые события HTML-объектов*

Имя события (С заглавными буквами)	Описание	К каким HTML-документам применяется
`onClick`	Запускается, когда элемент щелкают кнопкой мыши	Почти ко всем
`onMouseOver`	Запускается, когда указатель мыши наводится на элемент	Почти ко всем
`onMouseOut`	Запускается, когда указатель мыши отводится в сторону от элемента	Почти ко всем
`onKeyDown`	Запускается, когда нажимают клавишу	`<select>`, `<input>`, `<textarea>`, `<a>`, `<button>`
`onKeyUp`	Запускается, когда отпускают нажатую клавишу	`<select>`, `<input>`, `<textarea>`, `<a>`, `<button>`

Таблица 15.2 *(окончание)*

Имя события (С заглавными буквами)	Описание	К каким HTML-документам применяется
onFocus	Запускается, когда элемент управления получает фокус (другими словами, когда курсор устанавливается в элементе управления так, что можно вводить в него данные). К элементам управления относятся текстовые поля, переключатели-флажки и т. п. — дополнительную информацию см. в табл. 15.3	<select>, <input>, <textarea>, <a>, <button>
onBlur	Запускается, когда элемент управления теряет фокус	<select>, <input>, <textarea>, <a>, <button>
onChange	Запускается, когда в элементе управления меняется вводимое значение. В текстовом поле это событие не происходит до тех пор, пока вы не перейдете к другому элементу управления	<select>, <input type="text">, <textarea>
onSelect	Запускается, когда выделена порция текста в поле ввода элемента управления	<input type="text">, <textarea>
onError	Запускается, когда браузер не может загрузить изображение (обычно из-за некорректного URL)	<img>
onLoad	Запускается, когда браузер завершает загрузку новой страницы или заканчивает загрузку такого объекта, как изображение	<img>, <body>, <frame>,<frameset>
onUnload	Запускается, когда браузер выгружает страницу. (Обычно событие происходит после того, как вы ввели новый URL или когда вы щелкнули ссылку кнопкой мыши. Оно возбуждается *до того*, как браузер загрузит новую страницу)	<body>, <frameset>

Графические ролловеры

Одно из самых популярных событий, связанных с мышью, — *графический ролловер* (с *англ.* буквально "переворачиватель"). Для его формирования начните с создания элемента <img>, отображающего картинку. После этого, когда посетитель наведет указатель мыши на нее, браузер отобразит новое изображение благодаря событию onmouseover. Создать графический ролловер очень легко. Нужно только получить HTML-объект для элемента <img> и затем изменить свойство src.

В данном случае не удастся сделать все в одной строке кода. Можно засунуть весь сценарий в атрибут события (отделяя строки кода с помощью точек с запятыми), но разметка будет выглядеть громоздкой. Лучше записать код как функцию. Далее с помощью атрибута event можно будет подключить элемент к функции.

Далее приведена функция для смены изображения. В данном сценарии функция написана очень распространенным способом, с использованием параметров, поэтому ее можно будет применять снова и снова, как вы скоро увидите. При каждом

вызове функции вы указываете, какое изображение хотите заменить (задавая соответствующий ID) и какой файл с новым изображением хотите использовать. Поскольку функция использует параметры, ее можно вызывать для замены любого изображения в ролловере для мыши в любом месте вашей страницы.

```
<script>
  function ChangeImage(imageName, newImageFile) {
    // Находит объект, представляющий элемент img
    var image = document.getElementById(imageName)

    // Изменяет изображение
    image.src = newImageFile
  }
</script>
```

При создании графического ролловера вам понадобятся два события. Событие `onmouseover` переходит к изображению ролловера, а событие `onmouseout` (запускающееся, когда посетитель отводит указатель мыши в сторону от HTML-элемента) возвращает назад исходное изображение. На рис. 15.8 показан результат.

```
<img id="SwappableImage" src="pic1.gif" alt=""
onmouseover="ChangeImage('SwappableImage', 'LostInterestMessage.gif')"
onmouseout="ChangeImage('SwappableImage', 'ClickMe.gif')" />
```

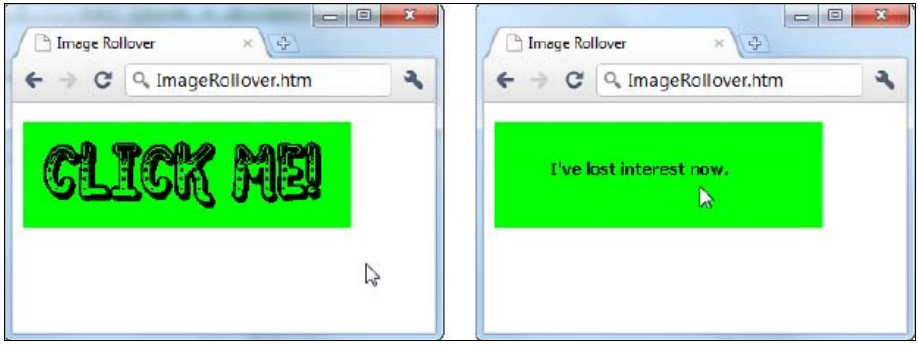


Рис. 15.8. Графический ролловер в действии

Если хотите вставить дополнительные изображения в ролловер, просто добавьте новый элемент `<img>` с другим именем. Следующий элемент использует то же самое исходное изображение, но отображает другое изображение при каждом наведении посетителем указателя мыши на картинку и отведении указателя мыши в сторону.

```
<img id="SwappableImage2" src="pic1.gif" alt=""
onmouseover="ChangeImage('SwappableImage2', 'MouseOverPicture.gif')"
onmouseout="ChangeImage('SwappableImage2', 'InitialPicture.gif')" />
```

Если хотите создать причудливый эффект, можно использовать даже событие `onclick` (которое запускается при щелчке элемента кнопкой мыши) и добавить еще одно изображение в набор.

Примечание

В *главе 16* вы увидите другой способ создания графических ролловеров, с помощью CSS. Хотя технология CSS не всегда срабатывает, это замечательное средство для построения базовых кнопок-ролловеров.

Сворачивающийся текст

Еще один остроумный прием, использующий события, — создание *сворачивающихся страниц*. Суть сворачивающейся страницы такова: если у вас есть большой объем информации, приготовленной для посетителей, но вы не хотите утомлять их длинной страницей, вы скрываете (сворачиваете) фрагменты текста, пряча их за заголовками, которые гости могут щелкнуть кнопкой мыши, когда захотят увидеть свернутый текст (рис. 15.9).

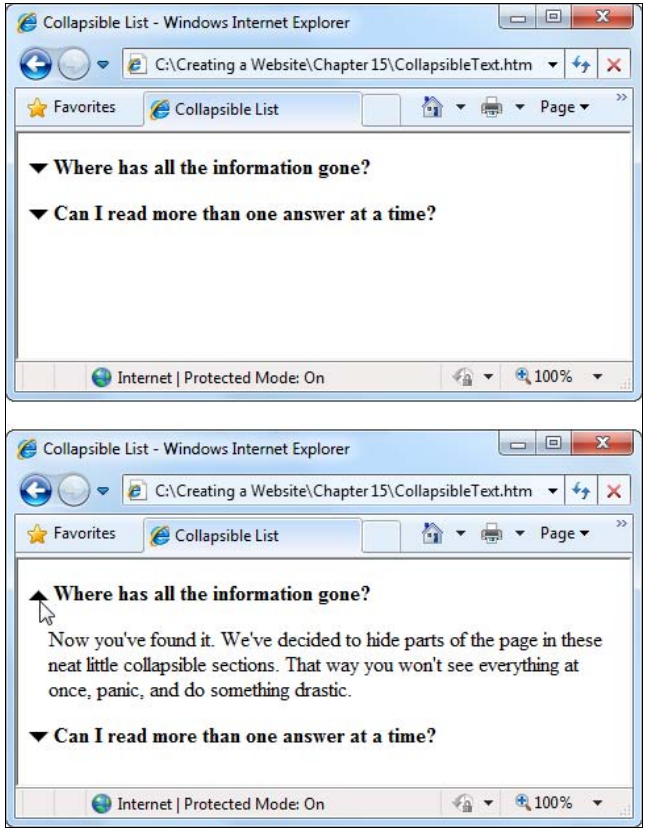


Рис. 15.9. Первоначально браузер скрывает весь основной текст (вверху), но когда вы щелкаете мышью изображение стрелки, направленной вниз, браузер отображает содержимое данного раздела (внизу). Вы можете одновременно раскрыть столько разделов, сколько захотите

Динамический HTML предоставляет много способов, позволяющих обманом заставить браузеры скрывать текст, и в следующем примере показан один из самых лучших. Метод тесно связан со свойством CSS `display`. Если значение этого свой-

ства — block, элемент отображается на HTML-странице стандартным образом. Но если свойство равно none, элемент исчезает вместе со всем, что находится внутри него.

Первый компонент сворачивающейся страницы — функция, скрывающая и отображающая ваш контент. Функции нужны два параметра: имя изображения "показать/свернуть" и имя элемента, который хотите скрывать или показывать. Функция на самом деле выполняет двойную задачу: проверяет текущее состояние раздела и затем меняет его состояние. Другими словами, благодаря условной логике она автоматически раскрывает свернутый раздел и автоматически сворачивает отображенный раздел. И пока функция это делает, она переключает изображение "показать/свернуть", отображая стрелки двух видов.

> ***Примечание***
>
> Прием, изменяющий текущее состояние элемента на противоположное, на жаргоне программистов называется переключением.

```
<script>
  function ToggleVisibility(image, element){
    // Находит изображение
    var image = document.getElementById(image)

    // Находит элемент, который нужно свернуть/показать
    var element = document.getElementById(element)

    // Проверяет текущее состояние элемента
    if (element.style.display == "none"){
      // Если он скрыт, отображает его
      element.style.display = "block"
      image.src = "open.png"
    }
    else
    {
      // Если он не скрыт, скрывает
      element.style.display = "none"
      image.src = "closed.png"
    }
  }
</script>
```

Код начинается с поиска двух нужных вам объектов и сохранения их в переменных image и element. Далее он приступает к работе. Определяется текущее состояние абзаца и принимается решение (с помощью оператора if) о том, показывать или скрывать текст. Выполняется только одна часть этого условного оператора. Например, если браузер в данный момент прячет изображение (т. е. если вы задали стиль отображения none), функция выполняет только следующие две строки кода, затем переходит вниз и завершается.

```
element.style.display = "block"
image.src = "open.png"
```

С другой стороны, если браузер выводит изображение в данный момент, код получает шанс проверить себя:

```
element.style.display = "none"
image.src = "closed.png
```

Для применения этой функции вам потребуется вставить элемент <img>, выполняющий переключение вашей Web-страницы. Вам также нужно добавить HTML-раздел, содержащий скрываемый контент. Вы можете показывать и скрывать практически любой HTML-элемент, но удачный универсальный вариант — элемент , потому что в него можно вставить все, что вы захотите спрятать. Далее приведен пример.

```
<p>
  <img id="Question1Image" src="closed.png" alt=""
  onclick="ToggleVisibility('Question1Image','HiddenAnswer1')" />
  <b>Where has all the information gone?</b>
</p>

<div id="HiddenAnswer1">
  <p>Now you've found it. We've decided to hide parts of the
  page in these neat little collapsible sections. That way you won't
  see everything at once, panic, and do something drastic.</p>
</div>
```

В первой части разметки (между первыми тегами <p>) задан заголовок, который посетители видят всегда. Он содержит изображение стрелки и вопрос (выводимый жирным шрифтом). Вторая часть (в элементе) — ответ, который ваш код то показывает, то скрывает.

Самое замечательное то, что вы можете многократно применять функцию для создания новых сворачиваемых разделов, потому что поместили в нее всю сложную обработку. Вновь создаваемые разделы будут иметь ту же структуру, но другое содержимое.

```
<p>
  <img id="Question2Image" src="closed.png" alt=""
  onclick="ToggleVisibility('Question2Image','HiddenAnswer2')" />
  <b>Can I read more than one answer at a time?</b>
</p>

<div id="HiddenAnswer2" style="display:none">
  <p>You can expand as many or as few sections as you want.
  Once you've expanded a section, just click again to collapse it back up
  out of sight. The only rule is that when you leave this page and come
  back later, everything will be hidden all over again. That's just the
  way JavaScript and Dynamic HTML work.</p>
</div>
```

Имейте в виду, что каждому элементу <img> и следует присвоить уникальный ID, иначе ваша функция не будет знать, какое изображение менять и какой раздел скрывать.

При желании можно преобразовать эту страницу, немного изменив ее внешний вид и поведение, но сохранив прежнюю способность сворачиваться. Например, вы можете облегчить пользование страницей, позволив посетителю раскрывать и сворачивать разделы, щелкая кнопкой мыши текст заголовка (а не только изображение). Проще всего для этого вставить изображение и отображаемый жирным шрифтом заголовок в элемент и задать для него атрибут event, со значением onclick. Далее приведено изменение, которое нужно внести в код.

```
<div onclick="ToggleVisibility('Question1Image','HiddenAnswer1')">
  <p>
  <img id="Question1Image" src="closed.png">
  <b>Where has all the information gone?</b>
  </p>
  </div>
```

Вы даже можете подчеркнуть текст заголовка, чтобы он выглядел как ссылка, о которой известно, что если ее щелкнуть кнопкой мыши, что-то произойдет. Для этого примените свойство таблицы стилей text-decoration (см. табл. 6.3).

И наконец, если хотите, чтобы все сворачиваемые разделы выводились первоначально свернутыми, необходимо добавить еще один сценарий, выполняющий эту работу. Далее приведен нужный вам блок сценария, который можно поместить в конец вашей страницы, прямо перед закрывающим тегом </body>.

```
<script>
  // Скрыть все разделы, один за другим
  ToggleVisibility('Question1Image','HiddenAnswer1')
  ToggleVisibility('Question2Image','HiddenAnswer2')
...
</script>
```

Вы можете скрыть все сворачиваемые разделы еще проще, если примените стилевое свойство display к каждому элементу с помощью встроенного правила стилей (см. *разд. "Применение встроенных стилей" главы 6*). Но этот прием может создать проблемы для маловероятного события, вызванного тем, что посетитель выключил поддержку JavaScript в своем браузере. В такой ситуации все разделы будут все время оставаться скрытыми. Применив алгоритм, использованный в приведенном ранее коде, вы обеспечите в браузерах с проблемами для JavaScript стандартное отображение всего контента, включая сворачиваемые разделы. Страница не будет столь выразительной, но зато ничего не потеряется. Этот подход, гарантирующий отображение страницы для всех и добавление преимуществ, когда это возможно, называется *прогрессивным улучшением* (progressive enhancement).

ПРИМЕЧАНИЕ

В *главе 16*, когда столкнетесь со сворачивающимся меню, вы увидите дополнительные эффекты сворачивания текста.

Интерактивные формы

HTML-формы — это еще не исследованная вами часть стандарта HTML. Они состоят, например, из текстовых полей, кнопок, флажков и списков. Эти компоненты называют *элементами управления*. Посетители взаимодействуют с ними, отвечая на вопросы и предоставляя информацию.

На рис. 15.10 показан пример действующей HTML-формы.

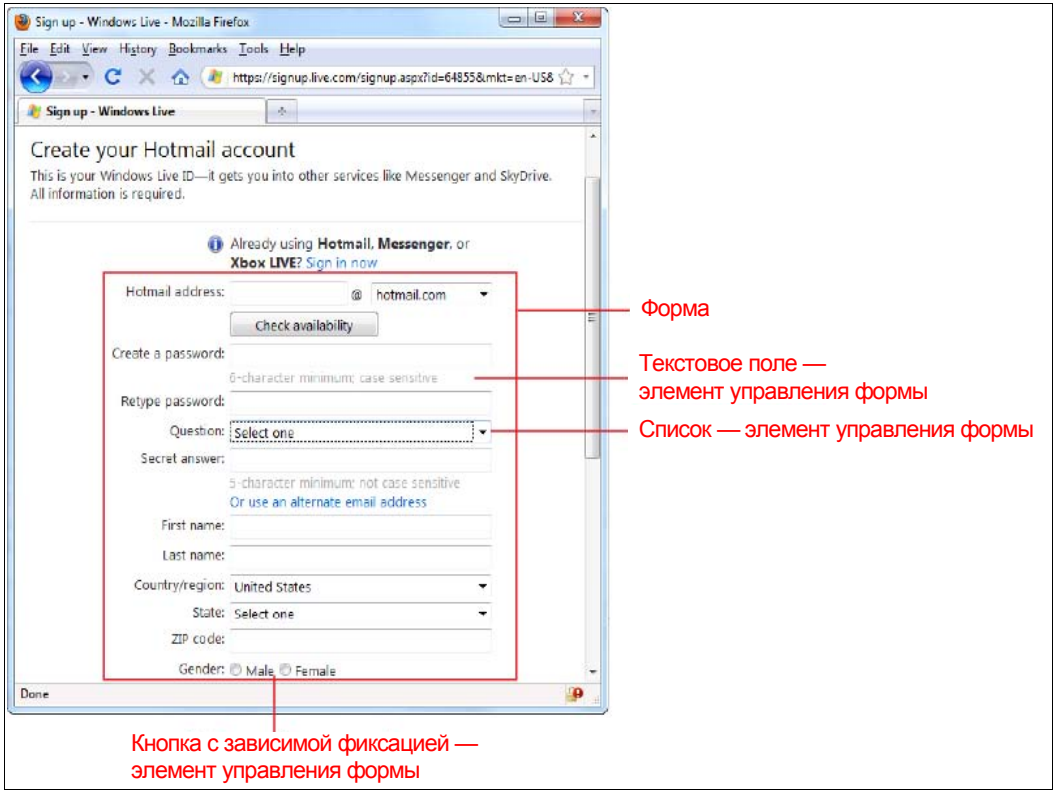


Рис. 15.10. Прежде чем Microsoft предоставит вам учетную запись Hotmail, необходимо отправить некоторую довольно подробную информацию. Текстовые поля, списки и кнопки, которые вы видите, — все это часть HTML-формы

HTML-формы — необъемлемая часть множества Web-сайтов, но вы, вероятно, не получите большого выигрыша от них. Дело в том, что они служат "визитной карточкой" серверных Web-приложений (см. *разд. "Серверное и клиентское программирование" данной главы*). В типовой форме посетители вводят информацию и нажимают кнопку отправки. Далее браузер собирает эту информацию и отправляет ее обратно на Web-сервер для дальнейшей обработки. Эта обработка может включать хранение информации в базе данных или отправку обратно еще одной страницы с другим HTML-кодом (например, сообщение об ошибке, если приложение обнаружило проблему).

Но умелый разработчик на JavaScript все равно сможет извлечь пользу из базовой формы. Отличие состоит в том, что вместо отправки информации обратно на Web-сервер форма собирает данные и посылает их в процедуру на JavaScript. Затем этот программный код на JavaScript делает следующий шаг, например, выполняя вычисления или обновляя страницу.

Элементы формы

Любая HTML-форма начинается с элемента `<form>`. Поскольку элемент `<form>` — это контейнер, HTML интерпретирует все внутри него как часть вашей формы.

```
<form>
   ...
</form>
```

Элементы `<form>` также являются блочными элементами (см. *разд. "HTML-элементы для основного текста" главы 5*). Когда вы создаете форму, ваш браузер вставляет немного свободного пространства, и вы начинаете с новой строки.

Что же происходит внутри вашего элемента `<form>`? Обычный контент (например, абзацы текста) вы можете помещать как внутрь него, так и снаружи. Но элементы управления формы (такие графические компоненты, как кнопки, текстовые поля и списки) всегда следует помещать *внутрь* элемента `<form>`. В противном случае у вас не будет возможности перехватить информацию, введенную посетителями.

Для создания элементов управления вы применяете еще один набор HTML-элементов. Самое интересное — большинство элементов управления формы используют абсолютно один и тот же элемент. Этот элемент называется `<input>`. Он представляет информацию, которую вы хотите получить от посетителя. Тип элемента управления `<input>` вы выбираете с помощью атрибута `type`. Например, для создания флажка используется тип `checkbox`.

```
<input type="checkbox" />
```

Для создания текстового поля (в котором посетитель набирает любой текст) применяется атрибут `text`.

```
<input type="text" />
```

Любой элемент `<input>` также содержит атрибут `value`, который обычно используется для задания начального состояния элемента управления. Например, если хотите поместить какой-то текст в текстовое поле при первом появлении страницы, можно применить следующую разметку.

```
<input type="text" value="Enter the first name here" />
```

Флажки немного отличаются. Вы можете сделать их установленными в начале, добавив атрибут `checked="checked"`, как показано далее.

```
<input type="checkbox" checked="checked" />
```

Не все элементы управления используют элемент `<input>`. В действительности есть два известных исключения. Для захвата текста большого объема, занимающего не-

сколько строк, применяется элемент <textarea>. (Не спрашивайте, почему HTML применяет новый элемент для этой цели, во многом потому, что так исторически сложилось.) Для создания списка, из которого пользователь выбирает варианты, используется элемент <select>. В табл. 15.3 перечислены все самые распространенные элементы управления. В нее не включены новые элементы форм, добавленные в стандарт HTML5 (и не используемые в Internet Explorer 8 и более ранних версиях).

Таблица 15.3. *Элементы управления форм*

Элемент управления	HTML-элемент	Описание
Однострочное текстовое поле	<input type="text" /> <input type="password" />	Отображает текстовое поле, в которое посетители могут ввести текст. Если вы применяете тип password, браузер не будет отображать текст. Вместо него посетители будут видеть по мере ввода их пароля звездочку (*) или маркер (•) на месте каждой буквы, таким образом скрывается пароль от любопытных глаз
Многострочное текстовое поле	<textarea>...</textarea>	Отображает большое текстовое поле, способное вместить несколько строк текста
Переключатель-флажок	<input type="checkbox" />	Отображает флажок, который можно установить или сбросить
Переключатель-кнопка	<input type="radio" />	Отображает кнопку с зависимой фиксацией (кружок, который можно выбрать и сбросить). Обычно у вас есть группа таких кнопок, одна за другой, и в этом случае посетитель может выбрать только одну
Кнопка	<input type="submit" /> <input type="image" /> <input type="reset" /> <input type="button" />	Отображает стандартную кнопку, которую можно щелкнуть мышью. Кнопка submit всегда собирает данные формы и отправляет их по месту назначения. Кнопка image делает то же самое, но позволяет отобразить изображение, которое щелкают мышью, вместо стандартной кнопки с текстом на ней. Кнопка reset очищает установки посетителя и текст во всех полях ввода. Кнопка button ничего не делает, если вы не вставите программный код на JavaScript
Список	<select>...</select>	Отображает список, в котором посетитель может выбрать один или несколько элементов. Для вставки элемента списка вы добавляете элемент <option>

Базовая форма

Для создания законченной формы вы соединяете и подбираете элементы <input> и обычные HTML-элементы. Например, рассмотрим страницу, показанную на рис. 15.11. Она предлагает несколько текстовых полей, в которые посетители вво-

дят числа, и затем использует эти числа для вычисления, когда пользователь щелкает мышью кнопку **Calculate** (Вычислить).

Написать данный пример на удивление просто. Самая трудная часть — создание функции, выполняющей внутренние вычисления.

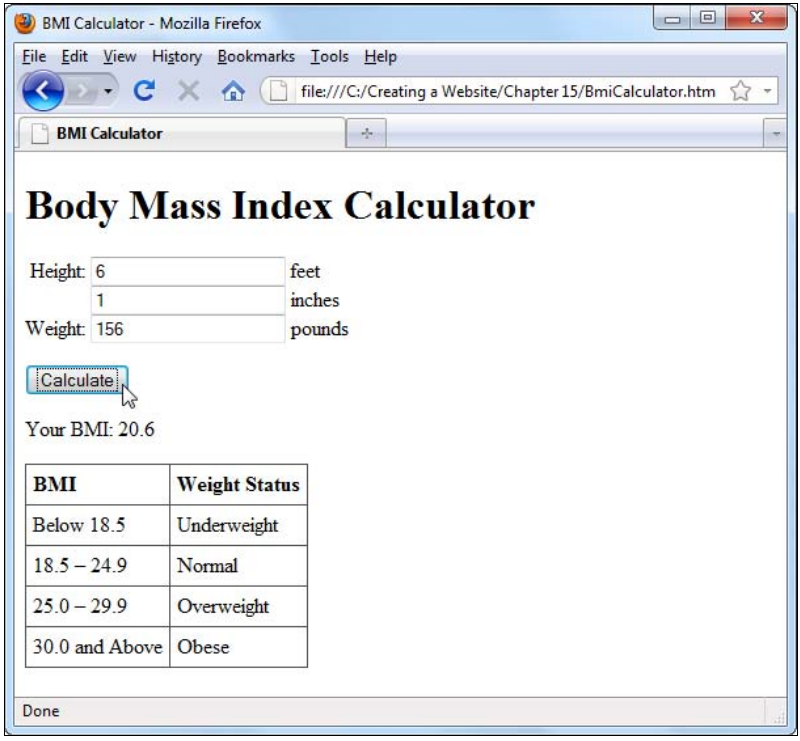


Рис. 15.11. Пока большинство пользователей размышляют о том, что калькулятор BMI (Body Mass Index, Индекс массы тела) говорит об их здоровье, вы находите имеющиеся однострочные текстовые поля и кнопку `submit`

Этой функции требуется несколько порций данных, соответствующих значениям в трех текстовых полях (футы, дюймы и фунты). Функции также нужно задать имя элемента, в котором она отобразит результаты. Далее показано начало функции.

```
<script>
  function CalculateBMI(feet, inches, pounds, resultElementName) {
```

> ***Примечание***
>
> Можно создать функцию `CalculateBMI()`, не использующую никаких параметров. Такая функция сможет просто находить все элементы управления на странице по имени. Но применение параметров — всегда верный подход, делающий ваш код более гибким. Теперь вы можете применять функцию `CalculateBMI()` на всевозможных страницах, с формой или без нее.

Код функции, приведенный далее, не очень отличается от того, что вы видели раньше. Его особенность в том, что он начинается с использования функции

`Number()`, встроенной в язык JavaScript. Эта функция преобразует текст, введенный пользователем, в числа, которые она сможет использовать в вычислении. Если не выполнить этот шаг, вы все же сможете получить правильный ответ (иногда), потому что JavaScript умеет автоматически преобразовывать текстовые строки в числа, как следует. Но тут есть ловушка — если вы решите сложить два числа и JavaScript думает, что они строки, он просто соединит обе строки в один фрагмент текста, т. е. 1+1 даст 11. Эта ошибка может по-настоящему испортить ваши вычисления, поэтому лучше всегда применять функцию `Number()` следующим образом.

```
inches = Number(inches)
pounds = Number(pounds)
feet = Number(feet)
```

Само вычисление не слишком интересно. Оно выполняется в строгом соответствии с определением индекса массы тела, которое можно найти в Интернете.

```
var totalInches = (feet * 12) + inches
```

В заключение функция отображает результат.

```
  var resultElement = document.getElementById(resultElementName)
  resultElement.innerHTML =
    Math.round(pounds * 703 * 10 / totalInches / totalInches) / 10
}
</script>
```

Создать форму, применяющую эту функцию, легко. Нужно только сформировать текстовые поля с помощью элементов `<input>` и дать им имена, которые легко запомнить. В приведенном далее примере форма использует таблицу для того, чтобы четко выровнять текстовые поля относительно друг друга.

```
<form action="">
  <table>
    <tr>
    <td>Height: </td>
    <td><input type="text" name="feet" /> feet</td>
    </tr>
    <tr>
    <td> </td>
    <td><input type="text" name="inches" /> inches</td>
    </tr>
    <tr>
    <td>Weight: </td>
    <td><input type="text" name="pounds" /> pounds</td>
    </tr>
  </table>
```

В заключение в конце формы создается кнопка, вызывающая функцию `CalculateBMI()`, которая использует значения из формы. Для того чтобы заставить кнопку сделать такой вызов, необходимо запрограммировать на вашей странице реакцию на событие `onclick`. Для поиска значения на форме не понадобится функ-

ция `getElementById()`. Вместо этого вы найдете значение по имени с помощью объекта `this.form`, представляющего текущую форму.

```
  <p>
    <input type="button" name="calc" value="Calculate"
    onclick="CalculateBMI(this.form.feet.value, this.form.inches.value,
this.form.pounds.value, 'result')" />
  </p>
</form>
```

И заключительный компонент — элемент, отображающий результат. В данном случае, т. к. вы хотите, чтобы он появился внутри другого абзаца, больше чем элемент `<div>` подойдет элемент `<span>` (обзор отличий см. в *разд. "Разделы и оболочки" главы 5*).

```
<p>
  Your BMI: <span id="result"></span>
</p>
```

Вы можете применять самые разные сценарии, связанные с формами. Например, можно, прежде чем разрешить посетителям переход на следующую страницу, для обнаружения ошибок проверить информацию, которую они ввели в форму. Дополнительные сведения о таких сценариях следует искать в Web, как описано в следующем разделе.

Сценарии в Web-пространстве

JavaScript — по-настоящему мощное средство. Если вы суперумник, любящий писать программы для связи собственной TiVo (система цифровой записи телевизионных программ) со своим карманным компьютером марки BlackBerry, вам понравится программировать на JavaScript долгими бессонными ночами. Но если вам не хочется лежать без сна и размышлять о том, что означает `var howMany = (trueTop>1?"s" : "")`, может быть разумнее доверить эту нелегкую задачу кому-нибудь другому.

Если вы относитесь к лагерю непрограммистов, в этой главе есть полезная для вас информация. Web-пространство завалено бесплатными сценариями на JavaScript. Чаще всего эти сценарии включают пошаговые инструкции, поясняющие, где поместить функции, какие элементы использовать на вашей странице и как с помощью событий связать ваши элементы с функциями.

Но у бесплатного кода есть и недостаток. Как вы узнали в начале этой главы, JavaScript появился на заре Интернета и многие сайты о JavaScript примерно того же возраста. В результате они могут содержать безвкусное форматирование, устаревшую информацию о совместимости браузеров (например, они могут предупреждать, что сценарий не работает в давно усопшем браузере Netscape) и устаревшие подходы, которые следует заменять более современными методами. Некоторые сайты с кодом на JavaScript битком набиты рекламой.

Если эти трудности вас не пугают, далее приводится несколько хороших отправных точек для поиска сценариев на JavaScript.

- **www.dynamicdrive.com**

  Этот сайт предлагает набор достойных сценариев, уделяющих особое внимание Dynamic HTML. Некоторые сценарии создают необычные эффекты, например, ярко светящиеся зеленые буквы, падающие на страницу, в стиле к/ф "Матрица".

- **www.huddletogether.com/projects/lightbox2**

  На этом сайте есть только один сценарий, но он создает один из самых популярных сегодня в Web визуальных эффектов. Когда вы щелкнете кнопкой мыши изображение-миниатюру, она увеличится до истинного размера — в то время как оставшаяся часть страницы незаметно потемнеет. Это разновидность данного эффекта.

- **www.javascriptkit.com/cutpastejava.shtml**

  На этом сайте множество сценариев, пришедших из старого доброго времени Интернета. Здесь есть несколько жемчужин и масса бесплатных сценариев для экспериментов.

- **www.webmonkey.com/tutorial/JavaScript_Tutorial**

  Этот сайт предлагает небольшой набор полезных учебных пособий по JavaScript, которые помогут вам ориентироваться в языке — и выбрать несколько полезных ключевых технических приемов.

С помощью этого перечня вы сможете отыскать все, начиная с незначительных хитростей и заканчивая полными действующими клонами игры Tetris. Но помните о том, что сценарий хорош настолько, насколько совершенен его создатель. Даже на сайтах со строгим контролем вы можете наткнуться на сценарий, не работающий во всех браузерах или очень сильно замедляющий загрузку вашей страницы. Возьмите за правило всегда тщательно проверять сценарий перед применением его на своем сайте.

> ***Совет***
>
> Характерная черта хорошего сайта сценариев — легкость перемещения на нем. Если сайт так завален рекламными объявлениями и всплывающими окнами, что трудно найти сами сценарии, знайте — вы попали на плохой сайт сценариев.

Поиск простого сценария

Готовы искать сценарии в сети? Перечисленные далее шаги проведут вас через весь процесс от начала до конца.

1. Запустите ваш браузер и выберите сайт из перечня, приведенного ранее.

   В данном примере используется сайт **www.dynamicdrive.com.**

2. На исходной странице сайта выберите нужную вам категорию.

   В данном случае перейдите в категорию **Documents Effects** (Эффекты в документах). Дополнительные примеры вариантов сценариев описаны далее, во врезке.

НА ПРОФЕССИОНАЛЬНОМ УРОВНЕ

Категории сценариев

Для того чтобы разобраться в том, какие имеются типы сценариев для Dynamic HTML, просмотрите разные категории на сайте Dynamic Drive (**www.dynamicdrive.com**). Далее приведена выборка из категорий, которые можно найти.

- Сценарии из категории **Calendars** (Календари) формируют остроумную HTML-разметку, которая выглядит как календари и отлично подходит для отображения важных дат или планирования на будущее.
- Категория **Date & Time** (Дата и время) предлагает идущие часы и обратные отсчеты до конкретной даты.
- Категория **Document Effects** (Эффекты в документах) предоставляет визуальные эффекты для переходов и фона страниц (например, фейерверки или летающие звезды).
- В категории **Dynamic Content** (Динамический контент) есть выдвигающиеся меню, наклейки и прокручиваемые панели.
- В категорию **Form Effects** (Эффекты в формах) включены сценарии, позволяющие управлять формами (см. *разд. "Интерактивные формы" данной главы*). Их можно применять для исключения многократной отправки посетителями одной и той же формы, проверки неверно введенных данных и т. д.
- В категории **Image Effects** (Эффекты для изображений) есть сценарии для создания слайдшоу и галереи изображений, а также динамические изображения, сменяющие картинку при перемещении мыши.
- В категорию **Links & Tooltips** (Ссылки и всплывающие подсказки) входят стильные ссылки, которые вспыхивают, причудливые кнопки и всплывающие текстовые поля, привлекающие внимание ваших посетителей.
- Категория **Menus & Navigation** (Меню и навигация) предоставляет удобные сворачивающиеся меню и полосы навигации, позволяющие посетителям перемещаться по вашему сайту (подобные компоненты вы увидите в *главе 16*).
- Категория **Mouse and Cursor** (Мышь и курсор) предлагает сценарии, изменяющие вид указателя мыши и добавляющие так раздражающие мышиные следы (картинки, следующие за указателем мыши повсюду).
- В категории **Scrollers** (Скроллеры) есть прокручиваемый текст, оформленный в виде выделительной рамки, как бегущая строка в телевизионных новостях.
- Сценарии категории **Text Animations** (Текстовые анимации) оживляют текст, заставляя его дрожать, летать, сверкать или приобретать другие причудливые свойства.
- Сценарии категории **User/System Preference** (Параметры пользователя/системы) находят сведения о браузере, который в данный момент отображает вашу страницу.
- В категории **Window and Frames** (Окно и фреймы) есть сценарии для вывода десятков всплывающих окон разных видов.

3. Просмотрите список сценариев в вашей категории (рис. 15.12) и затем щелкните кнопкой мыши один из них.

   В данном случае используется **Top-Down Stripy Curtain Script** (Сценарий ниспадающей полосатой занавески).

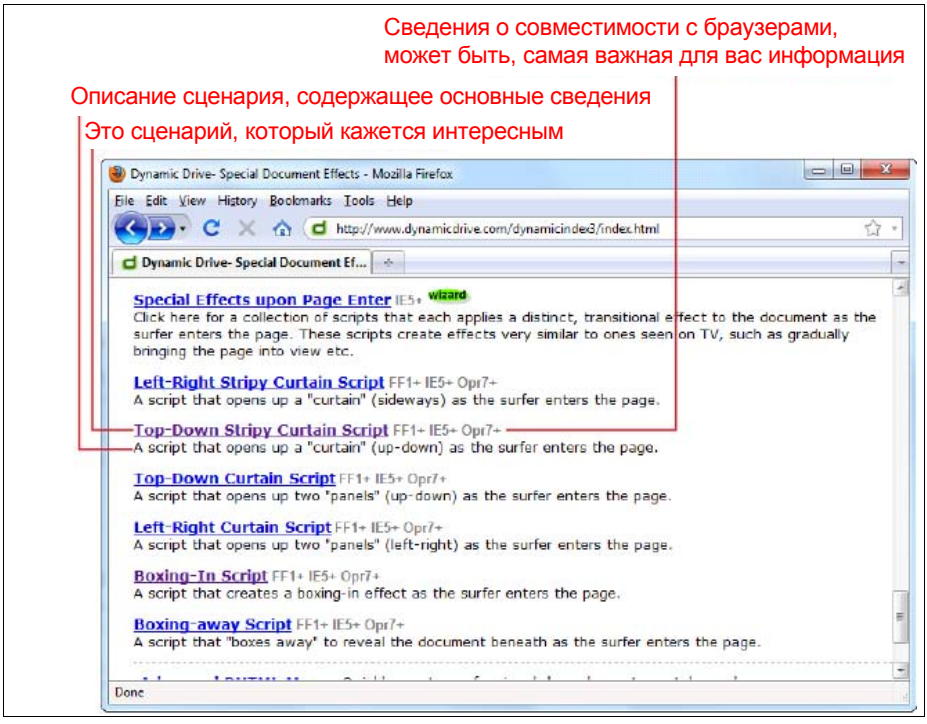


Рис. 15.12. Сценарий **Top-Down Stripy Curtain Script** — хороший выбор. Поддерживается браузерами Firefox, Internet Explorer 5 или более поздними версиями и Opera 7 или более поздними версиями

4. На следующей странице показан пример действия сценария (рис. 15.13).

   Когда страница загрузится, вы увидите описание сценария, имя автора и ссылку на сценарий (если он уже не появился на странице). Под всей этой информацией находятся пошаговые инструкции для сценария.

5. Скопируйте и вставьте на страницу разные части сценария в соответствии с приведенными инструкциями (рис. 15.14).

   Часто предлагается ряд функций, которые необходимо поместить в раздел `<head>` вашей страницы и затем несколько HTML-элементов, которые помещаются в раздел `<body>`. Иногда можно настраивать сценарии — например, для настройки кода сценария вы можете изменять числа и другие значения или для представления другого контента изменять HTML-элементы.

Примечание

Во многие сценарии включен ряд комментариев с информацией автора. Если они есть, стандартная практика — сохранять эти комментарии в вашем файле сценария, чтобы другие разработчики, просматривающие ваш сайт, знали, где взят сценарий. Такое поведение — всего лишь выдача кредита там, где он ожидается. Обычные Web-посетители даже не подумают заглянуть в код сценария, поэтому они не будут знать, кто написал этот сценарий с нуля.

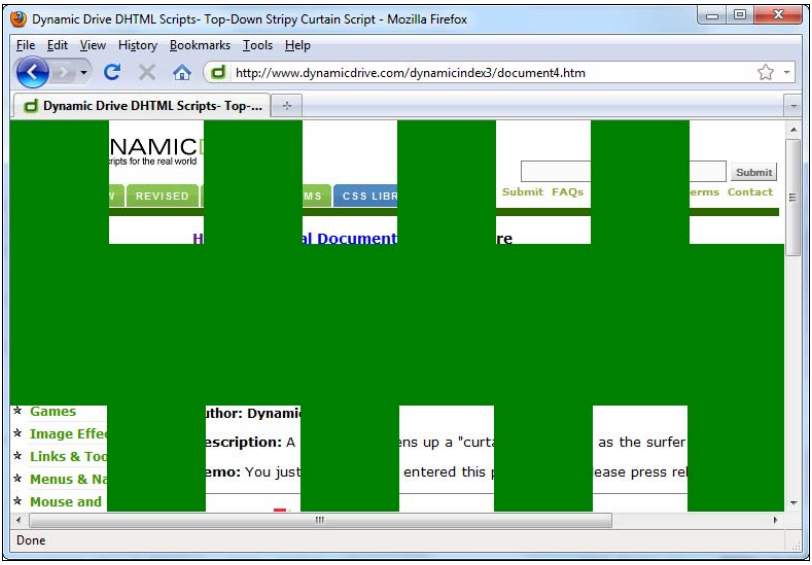


Рис. 15.13. Здесь показан сценарий **Top-Down Stripy Curtain Script** в действии. Он заполняет страницу, отображая сменяющие друг друга полосы, одни сверху вниз, а другие снизу вверх. Все это происходит в одно мгновение

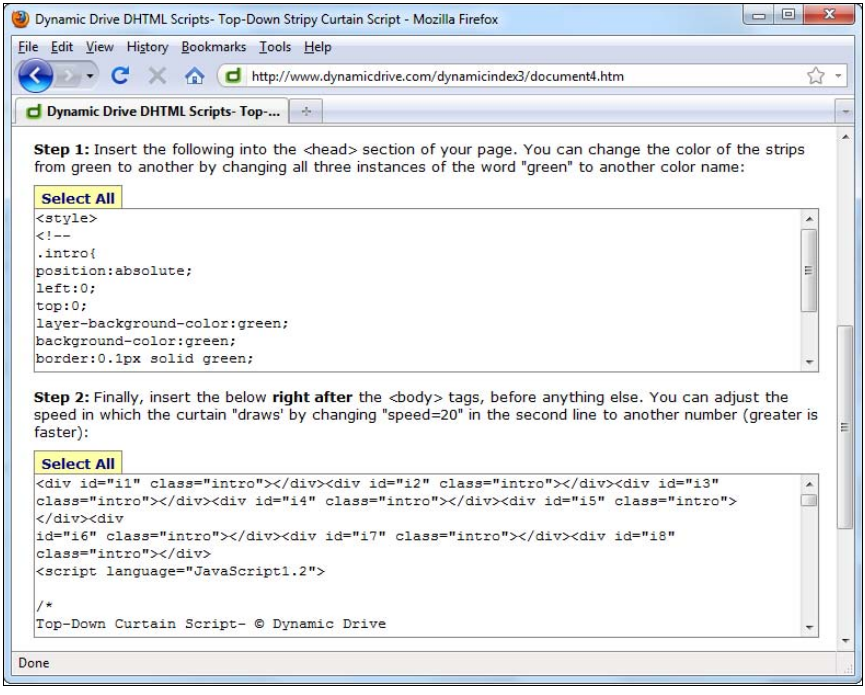


Рис. 15.14. У сценария **Top-Down Stripy Curtain Script** две составляющие. Первая — определение стиля, формирующего сплошную фоновую занавеску, которую содержимое страницы стирает. Вторая составляющая создает фоновую занавеску (как элемент `<div>`) и вставляет код, выполняющий переход. Скопируйте обе составляющие на любую страницу, и все в порядке. (Для лучшей организации подумайте о помещении кода в отдельный файл JavaScript, как описано в *разд. "Внешние файлы сценариев" данной главы*)

Библиотеки на JavaScript

Спустя годы со времени первого появления JavaScript, разработчики перешли к применению *библиотек* JavaScript. Эти библиотеки — не просто мешанина из отдельных сценариев. Они действуют как целый новый слой с расширенными функциями, повышающий качество JavaScript и расширяющий язык настолько, что вам ничего не нужно писать с нуля.

Более того, библиотеки JavaScript гарантируют, что все предлагаемые ими средства работают одинаково во всех современных браузерах, порой применяя для этого скрытый код громадного объема. И самые толковые программисты используют библиотеки JavaScript для создания более полезных сценариев, включая слайдшоу, карусели товаров (интерактивные ряды товаров с прокруткой), динамические диаграммы и лупы для изображений. Эти остроумные сценарии выходят далеко за пределы того, что предлагают примеры, которые можно найти на старомодном сайте с JavaScript. Поэтому профессиональные Web-разработчики почти всегда применяют библиотеки JavaScript.

Далее приведен список самых популярных библиотек JavaScript:

- jQuery (**http://jquery.com**)
- MooTools (**http://mootools.net**)
- Prototype (**www.prototypejs.org**)
- Dojo (**http://dojotoolkit.org**)

Если вы никогда не занимались программированием до чтения этой главы, возможно библиотеки JavaScript не для вас. Они созданы для программистов, решивших создать собственные снабженные JavaScript Web-страницы и не желающих снова изобретать колесо. Их могут применять и простые смертные, но прежде нужно многому научиться. Если вы новичок, можно прочитать книгу "JavaScript: The Missing Manual".

Web-виджеты

Сейчас появилось новое направление, Web-виджеты, вытесняющее JavaScript, распространяющийся с помощью копирования и вставки, и лишенное сложности, связанной с использованием библиотеки JavaScript. По сути, *Web-виджет* — это автономное миниатюрное приложение, которое добавляется на одну или несколько Web-страниц. Типовой виджет помещается в небольшом окне на странице и управляется JavaScript. К популярным примерам относятся виджеты, отображающие постоянно обновляемые новостные заголовки, предлагающие интерактивные игры, проигрывающие музыку, показывающие слайдшоу, подсчитывающие голоса посетителей, собирающие данные из форм и многое другое. Вы можете поискать виджет для вызова со своей страницы во множестве репозитариев виджетов в Web, например, по адресам **www.widgetbox.com**, **www.google.com/ig/directory?synd=open** и **www.widgipedia.com**.

Реальное отличие Web-виджетов от обычного кода на JavaScript, копируемого и вставляемого на страницу, заключается в том, что виджет выполняется на каком-нибудь Web-сервере. Это означает, что копируемый код виджета очень прост. Далее приведен полный код виджета прогноза погоды с сайта **www.widgetbox.com**.

```
<script
src="http://cdn.widgetserver.com/syndication/subscriber/InsertWidget.js">
</script>
<script type="text/javascript">
  if (WIDGETBOX)
    WIDGETBOX.renderWidget('97ec07e5-5918-47da-b3ef-9c4b92013564');
</script>
```

Бóльшая часть кода хранится на Web-сервере **cdn.widgetserver.com**, в файле с именем InsertWidget.js. Благодаря выполнению на другом сервере код на JavaScript имеет доступ к ресурсам этого сервера, например, информации в режиме реального времени, обменным курсам валют или (как в данном примере) прогнозам погоды.

Когда вы открываете страницу, браузер загружает этот файл с кодом на JavaScript, и он выполняется. Этот код создает виджет, который затем отображается на вашей странице (рис. 15.15).

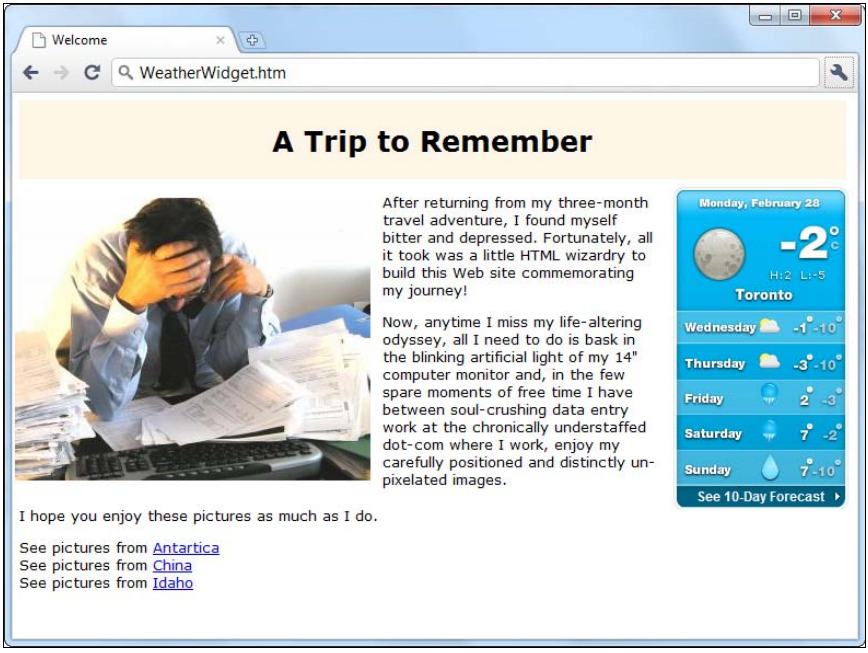


Рис. 15.15. На эту страницу виджет выводит прогноз погоды на неделю, не требуя от вас вставки ни одной строки кода на JavaScript

Примечание

Вы уже видели этот подход в нескольких местах книги. Например, Google применяет JavaScript, чтобы извлечь подходящий рекламный блок для вашей Web-страницы, ис-

пользующей AdSense (см. *разд. "Создание рекламного объявления" главы 14*). Также Yahoo Media Player подключает свой MP3-проигрыватель к вашим Web-страницам (см. *разд. "Плейер Yahoo: воспроизведение всех ваших ссылок" главы 17*).

Web-виджеты — прекрасное средство, если вы найдете именно тот, который вам нужен. Как правило, у виджетов немного настраиваемых параметров: например, вы можете изменить город в прогнозе погоды и размер виджета, но другие особенности его отображения и поведения изменить нельзя. (Некоторые виджеты требуют плату за дополнительные возможности или выводят крошечные рекламные объявления). Если хотите получить больше возможностей управления или создать собственный виджет, придется глубже погрузиться в JavaScript.

16

ГЛАВА 16

Привлекательные кнопки и меню

В последних 15 главах вы основательно набрались Web-учености, узнали, как структурировать Web-страницы с помощью HTML, украшать их средствами таблиц стилей и оживлять их с помощью JavaScript. Теперь пришло время пожинать плоды.

В этой главе вы узнаете, как применять три самых распространенных (и наиболее полезных) технологии Web-дизайна. Вы научитесь создавать причудливые кнопки, которые, например, начинают светиться, когда на них наводят указатель мыши. Вы также воспользуетесь системой всплывающих или раскрывающихся меню, позволяющих посетителям легко и со вкусом кружить по вашему сайту.

Эти средства дадут возможность продвинуться на шаг вперед и усовершенствовать навыки, приобретенные с помощью CSS и JavaScript. Иначе говоря, пришла пора заплатить за вашу тяжелую работу стильными украшениями.

Причудливые кнопки

Направления и стили Web-дизайна все время меняются. В прежние времена Web все пользовались обычными текстовыми ссылками, как те, что вы видели в *главе 8*:

```
<a href="graceland.html">Visit Elvis</a>
```

Спустя какое-то время эти стандартные ссылки стали казаться унылыми. Творческие Web-мастера захотели большего и решили применить вместо них маленькие картинки, которые можно щелкать мышью. Для этого они извлекали знакомый элемент `<img>` и применяли одну из двух методик.

Самый простой подход для создания графической кнопки — вставка изображения в элемент-якорь, как описано в *главе 8*. Вот как это выглядит.

```
<a href="graceland.html"><img src="VisitElvis.jpeg" alt="Visit Elvis" /></a>
```

К сожалению, когда применяется этот метод, HTML вставляет уродливую синюю рамку вокруг изображения для обозначения ссылки. Для того чтобы избавиться от рамки, атрибут свойства CSS `border-style` (см. *разд. "Границы" главы 6*) задается равным `none`.

Второй, отличающийся подход — использование элемента `<img>` в сочетании с атрибутом события JavaScript `onclick` (см. *разд. "События" главы 15*). Вот как это выглядит.

```
<img onclick="GoToGraceland()" src="VisitElvis.jpeg" alt="Visit Elvis" />
```

Этот подход не порождает уродливую синюю рамку и не нужно задавать атрибут `border-style`. Но вы обязательно должны создать функцию JavaScript, именуемую `GoToGraceland()` и содержащую код, который делает все, что вам нужно. Эта методика стоит усилий, если вы не хотите перенаправлять своего гостя на новую страницу, а вместо этого хотите выполнить код на JavaScript. Например, можно выполнить вычисления, изменить содержимое других элементов на вашей странице и т. д. Этот подход применялся в примере калькулятора индекса массы тела в *главе 15* (см. рис. 15.11) — посетители щелкали мышью кнопку, чтобы заставить калькулятор выполнить арифметическую операцию и вывести результат.

Довольно скоро Web-дизайнеры стали недовольны текстовыми ссылками или причудливыми картинками на кнопках. Они захотели большего и для этого воспользовались JavaScript. Основной задачей было создание кнопки нового сорта, использующей технологию JavaScript для переключения изображений, о которой вы узнали в *разд. "Графические ролловеры" главы 15*. Такие динамические кнопки (также называемые кнопками-ролловерами) изменяются мягко, но заметно, когда посетитель наводит на них свой указатель мыши. Этот эффект говорит о том, что посетитель балансирует над действующей кнопкой и для завершения нужно просто щелкнуть кнопку мышью.

Кнопки-ролловеры до сих пор широко распространены, и практически каждый пользовался ими. Но затем разработчики лидирующих Web-сайтов немного устали от всех этих вертящихся и сверкающих кнопок и решили вернуть свои страницы к более классическому виду, лишенному излишеств. Они ограничили применение кнопок-ролловеров и сами эффекты переключения сделали проще. Например, Web-дизайнеры могут изменить цвет фона у кнопки-ролловера и обязательно разместят поверх стандартный HTML-текст. Это не только позволяет создавать менее суетливые и меньше рассеивающие внимание страницы, но и облегчает их сопровождение, т. к. не требуется генерировать десятки разных изображений для кнопок. На рис. 16.1 показан стандартный подход к созданию привлекательной кнопки сегодня.

Вы уже знаете, как связать текст и стандартные неменяющиеся изображения. Вы также знаете основы методики замены изображений на JavaScript (она описана в *разд. "Графические ролловеры" главы 15*). Но вам пока еще неизвестно, как собрать их вместе в современный пакет, гарантирующий сдержанность и привлекательность ваших кнопок, загружающий ваши картинки без раздражающей задержки и применяющий остроумные CSS-хитрости для избавления вашего Web-сайта от запутанного кода на JavaScript. Об этом и пойдет речь.

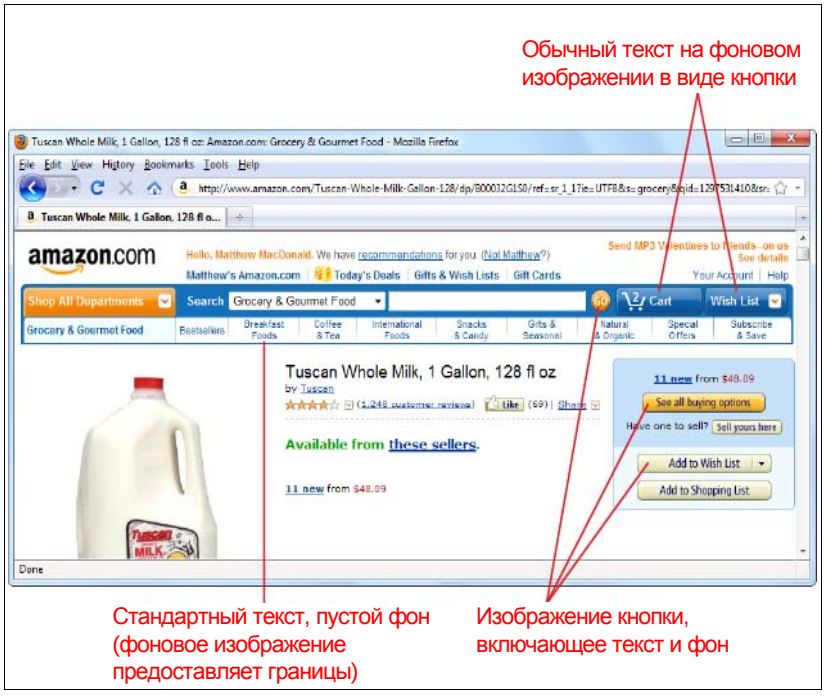


Рис. 16.1. На этой странице Amazon используется три методики создания кнопок: обычные текстовые ссылки в прямоугольной рамке, текст поверх фона в виде кнопки и кнопки, объединяющие текст и фон в одном изображении

Создание изображений ваших кнопок

Прежде чем идти дальше, нужно решить, какой вы хотите видеть свою кнопку и как вы планируете ее создать.

Первое решение, которое следует принять, — будет ваша кнопка только графической или она будет использовать обычный HTML-текст поверх фонового изображения кнопки (рис. 16.2). Первый вариант предоставляет самый полный контроль и допускает самые выразительные визуальные эффекты. Но он потребует больше усилий, так вам придется создать по паре разных картинок для каждой кнопки. Например, если вы хотите разместить на своей Web-странице восемь графических кнопок, потребуется 16 изображений кнопки, по одному для нормального состояния каждой кнопки и по одному для состояния выделенной кнопки (с наведенным на нее указателем мыши). Если же вы примените второй подход (который стал стандартом на более сдержанных, коммерчески ориентированных сайтах), потребуется только два изображения: стандартный фон кнопки и фон для выделенной кнопки. Эти изображения будут использоваться для всех созданных вами кнопок.

Примечание

Применение полностью графических кнопок потребует больше усилий не только при создании Web-сайта, но и в случае его изменения. Например, если вам нужно изме-

нить текст на кнопке, необходимо заново создать два изображения. Но если требуется изменить размер ваших кнопок, придется переделывать все их изображения. Если применяются только два изображения фона кнопок, у вас не будет этих проблем.

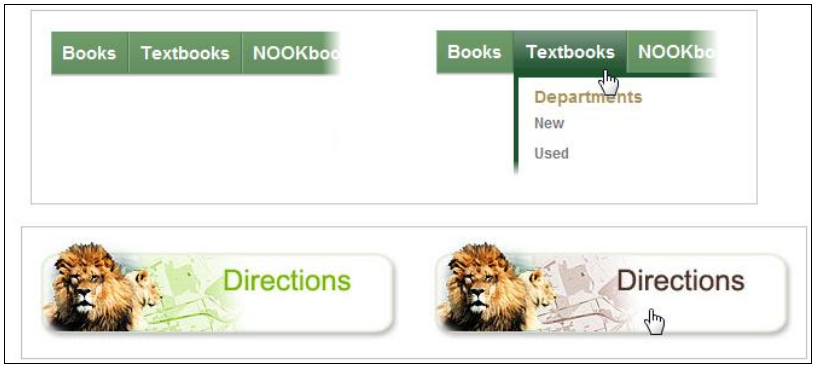


Рис. 16.2. *Вверху*: эта кнопка использует обычный текст с зеленым фоном, который темнеет, когда на него наводится указатель мыши.
Внизу: эта графическая кнопка переключает изображения при наведении указателя мыши

Если вы остановились на графических кнопках, нужен способ их создания. У вас следующие варианты.

- Нарисовать их самостоятельно. Если вы умеете рисовать, создать изображения кнопки можно практически в любой графической программе (две популярные программы, например, Adobe Photoshop и Adobe Fireworks). Но создать внешне привлекательные кнопки иногда совсем непросто. Также тяжело создать их в большом количестве, т. к. придется добиваться точного соответствия на всех кнопках: одни и те же местоположение текста, размер, цветовая палитра и фон.
- Воспользоваться Web-сайтом для создания кнопок. Если ваши художественные способности слабее, чем у рисующей гориллы Коко, есть более простой вариант. Можно использовать специализированную программу создания кнопок. У подобных программ единственная задача — помочь вам создать привлекательные кнопки с текстом, цветами и фоном, которые вы выберите. В Web полно таких программных средств. Они, как правило, начинают с просьбы подробно определить вашу кнопку (указать текст, цвет, фон и т. д.). После завершения вы просто щелкаете мышью кнопку и программа создает изображения кнопки (или кнопок) и отображает их на отдельной странице. Вам остается только загрузить изображения и начать применять их на своем сайте. Примерами сетевых средств создания кнопок могут служить сайты **www.buttongenerator.com** и **www.grsites.com/button**.

Примечание

Не все средства создания кнопок могут создавать изображения для всех состояний кнопки (unclicked (не нажатой), hovered over (с наведенным указателем мыши), clicked (нажатой) и visited (посещенной)). Но, как правило, вы можете запускать генератор кнопок многократно и для создания изображения выделенной кнопки выбрать немного отличающуюся цветовую палитру.

- Применить Expression Web. Популярное средство Web-дизайна обладает инструментом, позволяющим создавать целый ряд изображений кнопок и код на JavaScript для управления ими. Единственный недостаток этого средства — немного громоздкий подход, основанный на JavaScript, а не на технологии CSS, которую в наши дни предпочитает большинство Web-дизайнеров. Для того чтобы опробовать это средство, выберите из меню Expression Web команды **Insert | Interactive Button** (Вставка | Интерактивная кнопка). Вам придется выбирать во впечатляюще оснащенном генераторе кнопок.

Если вы решили использовать обычный фон кнопки с накладываемым на него текстом, ваша жизнь будет гораздо проще. Почти все могут создать два правильных прямоугольника в программе для рисования (один для обычного состоянии кнопки и другой — для выделенного). И поскольку вы создаете только две кнопки и никакого текста, вы не столкнетесь с проблемами нужного выравнивания текста на каждой из них (при неправильном подходе можно создать раздражающе несовместимые кнопки). Если не хотите самостоятельно создавать изображения, можно использовать инструмент Google для поиска готового фона кнопок, который решите применить на своем сайте. Или можно совсем отказаться от изображений и ничего не применять кроме свойств CSS для создания еще более простого фона кнопок, как будет показано далее в *разд. "Кнопки без изображений"*.

В последующих разделах вы увидите, как создать страницу, применяющую простейший и самый современный подход к кнопкам: кнопку-ролловер, использующую фоновое изображение и управляющую ее состояниями с помощью CSS. Вы также узнаете, как настраивать страницу на применение более сложного графически-текстового подхода.

> ***Примечание***
>
> Есть еще одна причина для выбора простого текста, накладываемого на фон кнопки: его предпочитают поисковые машины. Как вы узнали из *разд. "Применяйте информативный текст для ссылок" главы 11*, поисковые машины присваивают определенный вес тексту внутри ссылки-якоря. Но если вы применяете чисто графический подход, вы теряете возможность привлечения некоторого дополнительного внимания.

Создание эффекта ролловера с помощью CSS

После получения нужных вам изображений можно встраивать их в Web-страницу. Скажем, вы хотите создать кнопки, показанные на рис. 16.3.

Структурно это очень простая страница. На ней используется два элемента `<div>`: один для списка ссылок слева и второй для основного контента, занимающего остальное пространство.

```
<div class="MenuBar">
  <a href="...">Rollover Button 1</a>
  <a href="...">Rollover Button 2</a>
  <a href="...">Rollover Button 3</a>
  <a href="...">Rollover Button 4</a>
  <a href="...">Rollover Button 5</a>
```

```
  <a href="...">Rollover Button 6</a>
</div>

<div class="MainContent">
  <h1>Welcome.</h1>
  <p>...</p>
  <p>...</p>
</div>
```

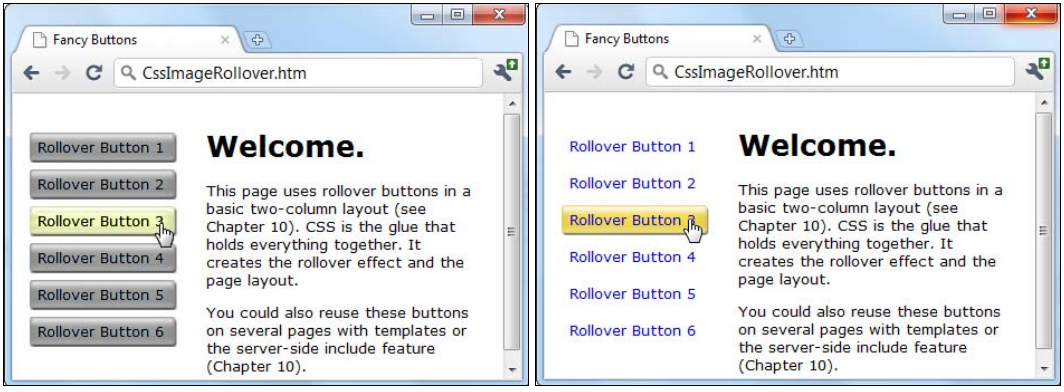


Рис. 16.3. *Слева*: на этой странице ролловер-эффекты заменяют серый с затенениями фон на желтый. *Справа*: порой можно получить более современный внешний вид, применив воздействие, оставляющее первоначальный фон белым

Вся магия, поддерживающая эту страницу, прячется в таблице стилей. Сначала примените ко всей странице базовое правило для элемента `<body>`, чтобы добиться единообразия шрифта.

```
body {
  font-family: Verdana,sans-serif;
  font-size: small;
}
```

Далее слева поместите полосу меню с помощью свойств верстки, с которыми вы познакомились в *главе 9*.

```
.MenuBar {
  position: absolute;
  top: 20px;
  left: 0px;
  margin: 15px;
}
```

Выделите дополнительное пространство для левого поля основного контента, чтобы он не наложился на меню.

```
.MainContent
{
  margin-left: 165px;
```

```
  margin-top: 30px;
  margin-right: 20px;
}
```

Пока все правила были понятны. Но все становится интереснее, когда вводится правило, которое отбирает все элементы-якори в полосу меню (с использованием контекстного селектора; см. *разд. "Сокращение работы с помощью контекстного селектора" главы* 6).

```
.MenuBar a {
  display: block;

  text-decoration: none;
  color: black;

  background: url("NormalButtonBackground.jpg") no-repeat 0 0;
  width: 125px;
  height: 23px;
  margin-bottom: 5px;
  padding-top: 5px;
  padding-left: 7px;
}
```

В этом HTML-коде несколько рычагов воздействия на вашу кнопку. Во-первых, свойство `display` преобразует `<a>` из строчного элемента в блочный (как абзац или заголовок). В этом случае браузер помещает каждую ссылку на боковой панели в отдельную строку, поэтому не нужно вставлять переносы строк.

> ***Примечание***
>
> Другой метод — превратить каждый элемент `<a>` в отдельный пункт маркированного списка (который представлен элементом `<ul>`). Затем можно воспользоваться таблицей стилей для форматирования списка и запрета отображения стандартного маркера рядом с каждым пунктом списка. Многие динамические, раскрывающиеся и сворачивающиеся меню, с которыми вы встретитесь в Web, на самом деле имеют внутреннюю структуру маркированных списков.

Далее свойства `text-decoration` и `color` изменяют текст, превращая его стандартный вид (подчеркнутый, синего цвета) в нечто, более подходящее для кнопки. Но настоящее волшебство начинается со свойства `background`. Оно захватывает изображение NormalButtonBackground.jpg и помещает его на задний план каждого якоря. Уточнение `no-repeat` гарантирует, что браузер не создаст мозаики из изображений, а значения `0 0` помещают изображения в левый верхний угол якоря.

В заключение свойства `width` и `height` задают размер якоря, соответствующий размеру фоновой кнопки (125×23 пиксела), в то время как свойства `margin` и `padding` отделяют кнопки друг от друга и задают отступ для текста внутри кнопки.

Конечно, ни одно из этих свойств не создает ролловер-эффект. Для его получения вам нужен псевдокласс `hover` (впервые описанный в *разд. "Изменения цвета ссылок*

и подчеркивания" главы 8). Он вступает в действие, когда указатель мыши наводится на элемент. В данном случае он изменяет фоновое изображение.

```
.MenuBar a:hover {
  background: url("HighlightedButtonBackground.jpg") no-repeat 0 0;
}
```

> **Совет**
>
> Если вы применяете фон, больше похожий на прямоугольное поле, чем на кнопку, может появиться желание задать в псевдоклассе `hover` свойство `text-decoration` со значением `underline`. В этом случае ссылка подчеркивается, когда кто-нибудь наводит указатель мыши на нее, делая ее назначение (навигацию по страницам) более понятным. Amazon использует этот трюк в своих всплывающих меню и в полосе просмотра категорий, тянущейся вдоль верхнего края некоторых окон (например, на рис. 16.1).

Приведенный код завершает пример. Но вы можете вспомнить, что CSS определяет еще несколько псевдоклассов, и применить их к вашей кнопке. Используйте `visited` для управления внешним видом ссылки после посещения пользователем связанной с ней страницы. Используйте псевдокласс `active` для управления внешним видом ссылки в тот короткий момент, когда посетитель щелкает ее кнопкой мыши. Например, можно немного сдвинуть кнопку в сторону, чтобы создать впечатление нажатой кнопки.

```
.MenuBar a:active {
  margin-left: 1px;
}
```

Еще один вариант — применить другое фоновое изображение.

Кнопки с изображением и текстом

Вы можете переделать предыдущий пример для работы с полностью графическими кнопками, для которых не нужно задавать никакого текста ссылки, но появятся новые проблемы.

Во-первых, нужно подумать о *предварительной загрузке* изображений. Она обеспечивает загрузку браузером изображений всех состояний кнопок при первоначальной обработке страницы, не дожидаясь наведения посетителем указателя мыши на кнопку. Несмотря на то, что при запуске страницы с жесткого диска вашего компьютера вы не заметите разницы, предварительная загрузка изображений делает кнопки более быстро реагирующими, когда посетители взаимодействуют с ними в Интернете, особенно если соединение не слишком быстрое, и ваша страница содержит много кнопок.

Web-разработчики испробовали множество хитростей для предварительной загрузки изображений. В результате они остановились на упаковке обоих изображений кнопки (для нормального и выделенного состояний) в один графический файл. В этом случае браузер захватывает только одно изображение, когда впервые отображает кнопку. На рис. 16.4 показан пример кнопки. (Кстати, эти изображения

были созданы на уже упоминавшемся сайте **www.buttongenerator.com** и затем вставлены вместе в один файл.)

Вы можете превратить эту сдвоенную кнопочную систему в строенную, применив псевдокласс `active`. Некоторые помешанные на производительности Web-диайнеры упаковывают всю страницу, наполненную кнопками и другими графическими украшениями, в единое фоновое изображение. Затем они применяют тщательно выверенные стилевые правила для всевозможной нарезки графики и раздачи ее всем элементам, нуждающимся в ней.

Методика сдвоенных кнопок очень помогает, если создаются кнопки с картинкой и текстом, поскольку у вас будет меньше файлов для управления и меньше файлов для ссылок в вашей разметке.

После того как подготовите несколько кнопок, можно формировать пример, наподобие показанного на рис. 16.5.



Рис. 16.4. На данном рисунке соединены изображения нормального (вверху) и выделенного (внизу) состояний кнопки. Правила вашей таблицы стилей могут захватить только одно изображение и применить его как фон. Учтите, что неважно, какого размера пробельную зону вы поместите между двумя картинками, главное, чтобы этот размер был постоянным для всех ваших картинок



Рис. 16.5. Страница с привлекательными кнопками-ролловерами

Первый шаг при создании графических кнопок — удаление всего текста из ваших якорей, чтобы просматривалась только кнопка. Затем нужно указать подходящие изображения кнопок. Вот тут и начинается путаница. В предыдущем примере вы сохраняли фоновые картинки с помощью двух стилевых правил. Но если вы хотите, чтобы для каждой кнопки применялись разные наборы изображений, потребуется и отдельное правило для каждой кнопки, а это наверняка создаст неразбериху даже в наилучшим образом организованной таблице стилей.

Лучше всего использовать стиль для хранения всей информации *за исключением* имен файлов с изображениями кнопок. Далее приведено чуть более короткое стилевое правило для форматирования элементов-якорей.

```
.MenuBar a {
  display: block;
```

```
  width: 115px;
  height: 18px;

  margin-bottom: 5px;
  padding-top: 5px;
  padding-left: 7px;

  background-repeat: no-repeat;
}
```

Теперь вы задаете графический файл для каждой кнопки с помощью встроенного стиля.

```
<a href="..." style="background-image: url('DogButton.png')"> </a>
<a href="..." style="background-image: url('CatButton.png')"> </a>
<a href="..." style="background-image: url('LemurButton.png')"> </a>
```

Правило в таблице стилей явно задает ширину и высоту кнопок, поэтому для них отводится место, достаточное для отображения только верхней части фонового изображения.

Заключительный компонент — правило, форматирующее выделенные ссылки (с наведенным на них указателем мыши). Оно сдвигает кнопку, только тень (указанием в свойстве `background-position` отрицательного количества пикселов), существенно подтягивая при этом изображение вверх, так что верхняя картинка кнопки попадает за пределы верхнего поля кнопки-якоря, и становится видна нижняя часть фонового изображения. На рис. 16.5 выделенная кнопка сидит на 23 пиксела ниже кнопки в нормальном состоянии, поэтому правило таблицы стилей выглядит следующим образом.

```
.MenuBar a:hover {
  background-position: 0 -23px;
}
```

Кнопки без изображений

Некоторые Web-дизайнеры минималисты создают кнопки совсем без изображений. Они применяют свойства CSS для создания прямоугольной рамки с тенью позади текста. Это немного более скромный подход и он страдает от одного дефекта. Пока окончательно не принят HTML5, браузеры могут не полностью поддерживать некоторые свойства CSS, превращающие обычный фон в настоящий отпечаток кнопки — такие как линейные градиенты, тени и скругленные рамки. В настоящий момент они работают в Firefox, Chrome, Opera и Safari, но не в любой версии Internet Explorer до IE 9.

Если хотите опробовать форматирование только средствами CSS, познакомьтесь с генератором кнопок на сайте CSS-Tricks (**http://css-tricks.com/examples/ButtonMaker**). Он предлагает только цвета и формирование тени для создания кнопки эконом-класса (рис. 16.6).

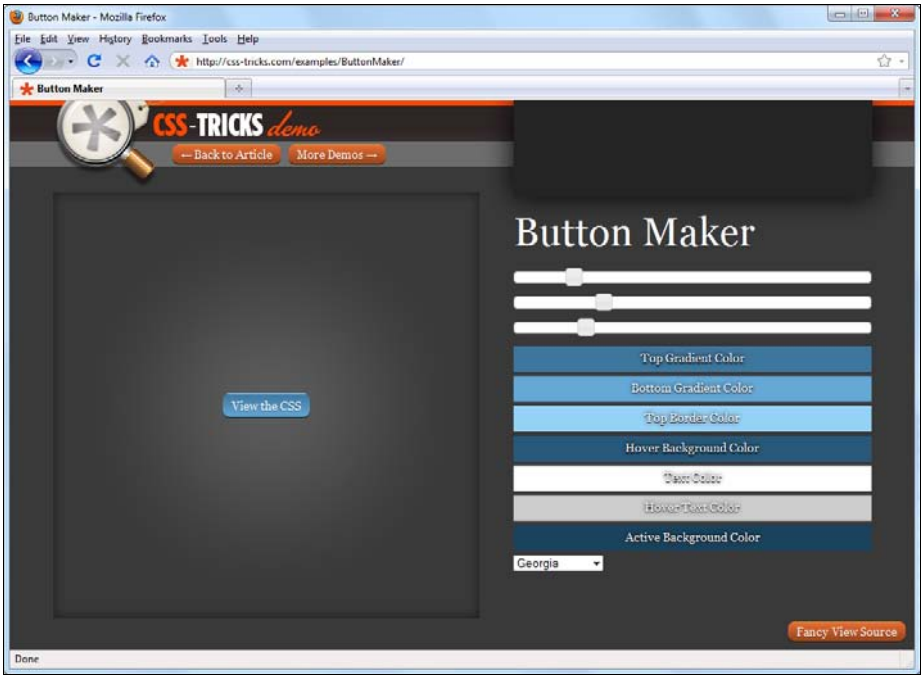


Рис. 16.6. Для создания кнопки на сайте CSS-Tricks просто подвигайте бегунки и щелкните мышью кнопку **View the CSS** (Просмотр CSS), когда останетесь довольны тем, что видите. Но имейте в виду, что скругленная с тенью кнопка, показанная на рисунке, превратится в плоский синий прямоугольник у гостей, пользующихся любой версией Internet Explorer до IE 9

Привлекательные меню

Когда впервые появились кнопки-ролловеры, они были очень популярны. Есть что-то неотразимое в кнопке, которая начинает светиться, когда вы на нее указываете. Тем не менее от хорошей вещи можно получить слишком много и вставка излишнего количества кнопок-ролловеров на страницу — верный путь превращения Web-сайта в перекормленную идейку.

Гораздо позже Web пережил ренессанс простоты и отказа от чрезмерных кнопок-ролловеров. Отчасти причина была в возрастающей сложности Web-сайтов — все очень просто: на современных обычно сложных сайтах горстка кнопок-ролловеров больше не обеспечивала достаточных навигационных средств. Вместо них эти сайты стали применять более подробные многоуровневые меню, заменив десятки кнопок-ролловеров более понятным и четким набором иерархических ссылок.

Типовое меню начинается с набора элементов-якорей, которые на странице группируются вместе. Главная задача — создать из этих ссылок логические группы. Например, Web-сайт компании может включать группу страниц товаров, группу страниц с контактной информацией и местонахождением и еще одну группу со страницами службы техподдержки. Если ссылки организованы в группы, посетители легче смогут найти то, что ищут.

Пока этот дизайн меню не требует ничего специального. Пользуясь навыками работы со ссылками, полученными в *главе 8*, и приемами верстки, освоенными в *главе 9*, вы легко можете создать боковую панель со сгруппированным списком якорей. Но по-настоящему искусные меню добавляют последний штрих — они могут сворачиваться. Это означает, что вы не видите список всех ссылок одновременно. Сначала вы видите только заголовки групп, например, **Товары**, **Контакты** и **Техподдержка**. Когда вы щелкаете мышью заголовок группы, прямо под ним отображается список связанных ссылок.

Создать сворачивающиеся меню можно несколькими способами. Некоторые очень просты, а другие ошеломляюще сложны. В следующих разделах вы узнаете, как сформировать собственное простое сворачивающееся меню, и затем воспользуетесь возможностями более сложного меню, предлагаемого на сайте с бесплатным кодом JavaScript.

Сворачивающиеся меню, сделанные самостоятельно

Вы самостоятельно можете создать привлекательное меню с помощью алгоритма сворачивающегося текста на языке Dynamic HTML, описанного в *разд. "Сворачивающийся текст" главы 15*. Основная идея заключается в использовании JavaScript для скрытия и отображения конкретных HTML-элементов с помощью изменения свойства CSS `display`.

Допустим, вы хотите создать крутое меню с вкладками, показанное на рис. 16.7. Вы делите ссылки на три группы, каждую из которых представляете на экранной вкладке. Одновременно только одна вкладка может вывести свои вложенные ссылки.

Такой двухуровневый дизайн (верхние ссылки на странице представляют первый уровень, а ссылки, появляющиеся при наведении указателя мыши на групповую ссылку, — второй уровень) может показаться немного запутанным, но он состоит только из двух частей: вкладок в верхней части страницы и окон ссылок (меню), которые отображаются динамически под вкладками, когда посетитель наводит указатель мыши на вкладку. Для того чтобы сделать работу с разделами более простой, может быть стоит подумать о заключении их в элементы `<div>` и `<span>`, как вы уже видели в этой книге.

> ***ПРИМЕЧАНИЕ***
>
> В оставшейся части раздела вы найдете подробный обзор решения шаг за шагом. Для того чтобы увидеть законченную страницу, просмотрите загружаемый контент к этой главе, доступный на странице **Missing CD** (Отсутствующий CD) по адресу **www.missingmanuals.com/cds/caw3**.

Поскольку все три вкладки появляются одна за другой в одной строке, для них лучше всего подойдет элемент `<span>`. Напоминаю, что элемент `<div>` вставляет перенос строки и некоторое дополнительное свободное пространство между элементами. Элемент `<span>` строчный элемент, т. е. вы можете вставить его внутрь существующего абзаца и поместить несколько элементов `<span>` один за другим.

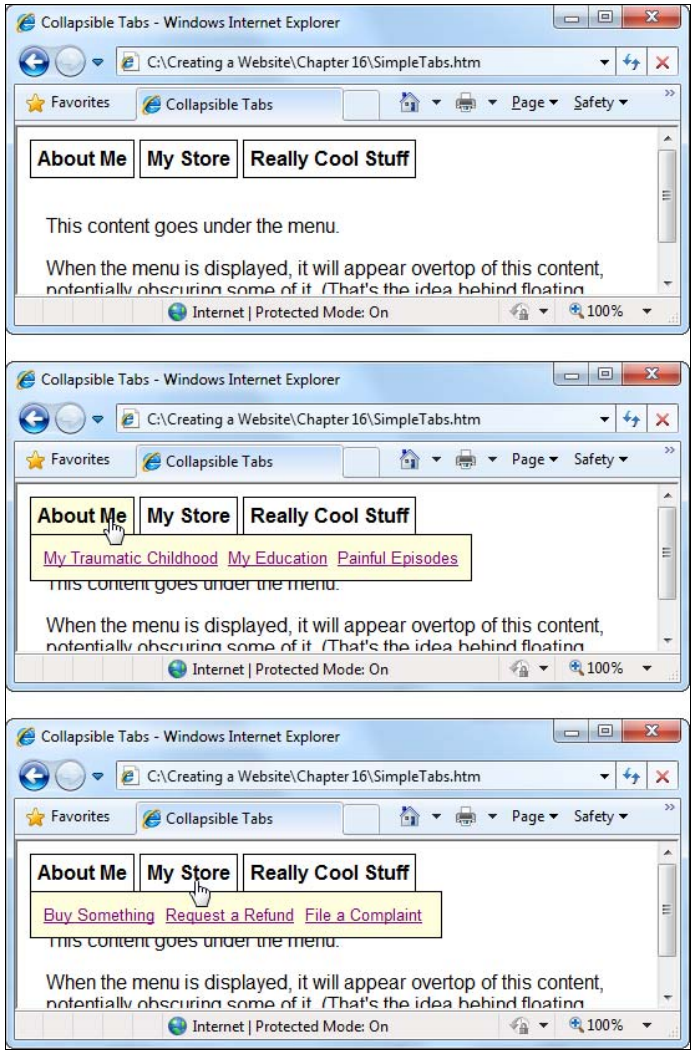


Рис. 16.7. *Вверху*: когда эта страница загружается в первый раз, она представляет посетителям всего три вкладки.
Посередине и *внизу*: когда посетитель наводит указатель мыши на заголовок вкладки, под вкладкой появляется набор связанных ссылок. Эти ссылки "наплывают" на контент страницы

Далее приведен HTML-код, представляющий три вкладки.

```
<div class="TabGroup">
  <span class="Tab">About Me</span>
  <span class="Tab">My Store</span>
  <span class="Tab">Really Cool Stuff</span>
</div>
```

У этих элементов `<span>` описательное имя класса — `Tab`. Оно связывает элементы со следующим правилом таблицы стилей, задающим для вкладок подходящий шрифт и границы.

```
.Tab {
  font-weight: bold;
  padding: 5px;
  border-style: solid;
  border-width: 1px;
  cursor: hand;
}
```

В это правило включено кое-что, чего вы еще не видели — свойство `cursor`. Оно задает стиль указателя мыши, появляющегося, когда гость наводит мышь на ссылку. В данном случае указатель мыши меняется на пиктограмму кисти руки (рис. 16.7, посередине).

Вы заключаете все вкладки в элемент `<div>`, использующий класс `TabGroup`. Это позволит поместить `TabGroup` в определенное место на странице.

```
.TabGroup
{
  position: absolute;
  top: 16px;
  left: 10px;
}
```

После того как вкладки определены, имеет смысл вставить плавающие подменю. Каждое подменю — это просто окно с границами и фоном желтого цвета. Внутри вы найдете группу ссылок. Здесь подходит элемент `<div>`, т. к. вы хотите, чтобы каждое подменю отображалось на странице независимо (вместо соединения всех подменю в одной строке). Кроме того, необходимо присвоить каждому элементу `<div>` уникальный ID, чтобы можно было менять видимость элемента, основываясь на вкладке, которую посетитель щелкнул мышью.

Далее приведены элементы `<div>` для трех групп ссылок.

```
<div id="AboutMe" class="Links">
  <a href="...">My Traumatic Childhood</a>
  <a href="...">My Education</a>
  <a href="...">Painful Episodes</a>
</div>
<div id="MyStore" class="Links">
  <a href="...">Buy Something</a>
  <a href="...">Request a Refund</a>
  <a href="...">File a Complaint</a>
</div>
<div id="ReallyCoolStuff" class="Links">
  <a href="...">Pie Eating</a>
  <a href="...">Harvesting Bananas</a>
  <a href="...">Blindfolded Heart Surgery</a>
</div>
```

Элементы `<div>` отображаются поверх страницы, а это значит, что вы должны применить абсолютное позиционирование для них. Далее приведено соответствующее стилевое правило.

```
.Links {
  position: absolute;
  top: 40px;
  left: 10px;
  border-width: 1px;
  border-style: solid;
  padding: 10px;
  background-color: lightyellow;
  font-size: small;

  display: none;
}
```

Наряду с координатами абсолютного позиционирования (40 пикселов от верхнего края окна браузера, 10 пикселов от левого края) этот стиль задает самую малость параметров форматирования (границы, фон, отступ и размер шрифта). Гораздо важнее то, что он применяет свойство `display` для явного скрытия всех подменю при первоначальной загрузке страницы. Таким образом, несмотря на то, что в этом примере элементы `<div>` для подменю укладываются один на другой, в таком виде вы не увидите их на странице, потому что вы вообще их не увидите сначала.

Внутри плавающих окон задайте для элементов `<a>` небольшое поле, чтобы они не накладывались друг на друга.

```
.Links a {
  margin-right: 5px;
}
```

> ***Совет***
>
> Если вы мечтаете о меню, которое укладывает ссылки одну на другую, можно настроить приведенное стилевое правило и применить вариант `display: block` так же, как вы делали на панели из кнопок-ролловеров в *разд. "Создание эффекта ролловера с помощью CSS" данной главы*.

И в завершение заключите оставшийся контент страницы в элемент `<div>` с именем класса `MainBody`. Задайте для этого `<div>` внушительное верхнее поле, чтобы освободить место для вкладок.

```
.MainBody {
  margin-top: 70px;
  margin-left: 15px;
}
```

Эти правила таблицы стилей и элементы `<span>` и `<div>` формируют базовый каркас вашей страницы. Заключительный шаг — создание сценария, отображающего один из скрытых элементов `<div>` в зависимости от того, какую вкладку выбрал ваш посетитель.

Код, выполняющий эту задачу, аналогичен тому, который применялся в функции `ToggleVisibility()` в *разд. "Сворачивающийся текст" главы 15*. Но в данном слу-

чае вас интересует не скрытие и отображение отдельных разделов; вы хотите показать единственный раздел, зависящий от выбранной вкладки, и скрыть все остальные. Работу выполняют две пользовательские функции: `MakeVisible()`, показывающая подменю для конкретной вкладки, и `ResetAllMenus()`, скрывающая все подменю.

Далее приведена упрощенная версия функции `MakeVisible()`. Как видите, она принимает имя элемента, находит элемент и изменяет параметры стиля так, что он отображается на странице.

```
function MakeVisible(element){
  // Находит element и отображает его
  var element = document.getElementById(element)
  element.style.display = "block"
}
```

Теперь нужно связать эту функцию со всеми вкладками. Тут у вас есть варианты: `MakeVisible()` может реагировать как на щелчок кнопки мыши (событие `onclick`), так и на наведение указателя мыши на элемент (событие `onmouseover`). В данном примере использован последний вариант.

```
<span class="Tab" onmouseover="MakeVisible('AboutMe')">About Me</span>
<span class="Tab" onmouseover="MakeVisible('MyStore')">My Store</span>
<span class="Tab" onmouseover="MakeVisible('ReallyCoolStuff')">Really
  Cool Stuff</span>
```

Страница все еще не до конца корректна. Хотя функция `MakeVisible()` отображает нужные окна подменю, она ничего не скрывает. Это значит, что если вы поочередно наведете указатель на три вкладки, вы увидите все три группы ссылок одновременно, одну поверх другой.

Для скрытия лишних окон потребуется еще немного ловкости. Проблема состоит в том, что `MakeVisible()` знает, какую группу ссылок собирается показать, но ей не известно состояние двух других окон. Для его выяснения ваш код должен выполнить поиск в оставшейся части страницы. В данном примере основной алгоритм — поиск любых элементов `<div>` с именем класса `Links` и скрытие их. Делает это функция `ResetAllMenus()`.

```
function ResetAllMenus() {
  // Получение массива элементов div
  var links = document.getElementsByTagName("div")

  // Поиск в массиве окон ссылок и их скрытие
  for (var j = 0; j < links.length; j++) {
    if (links[j].className == 'Links') links[j].style.display = "none"
  }
}
```

Этот код немного сложен. Сначала функция `getElementsByTagName()` извлекает программный объект, именуемый *массивом*. Массив — это особая переменная, т. к. он содержит не просто один объект, а одновременно группу объектов. В данном слу-

чае массив с именем `links` содержит три объекта, по одному на каждый элемент `<div>` на странице.

Далее применяется программная конструкция, называемая *циклом for*. Он обрабатывает код определенное число раз, применяя встроенный счетчик. В нашем примере счетчик — это переменная `j` с начальным значением 0, продолжающая увеличиваться до тех пор, пока не станет равна переменной `links.length` или, иначе, пока не дойдет до последнего объекта `<div>` в массиве `links`. При условии, что массив `links` содержит три объекта, браузер выполнит следующий оператор три раза:

```
if (links[j].className == 'Links') links[j].style.display = "none"
```

Первый раз `j` равна 0 и код загружает первый объект в список. Второй раз `j` — 1 и извлекается второй объект. Вы можете догадаться, что произойдет в третий раз. Когда код перемещается по этому списку, он проверяет имя класса для каждого элемента `<div>`. Если он определяет, что вы нашли окно ссылок, его отображение стирается со страницы за счет изменения стиля отображения.

> ***ПРИМЕЧАНИЕ***
>
> Если необычные аспекты JavaScript все еще выглядят как датский язык, не беспокойтесь. При желании о программных компонентах, таких как массивы, циклы и операторы `if`, можно узнать на Web-сайтах или из специальной книги (например, "JavaScript: The Missing Manual"). А можно сохранить душевное здоровье и положиться на примеры, предлагаемые в этой книге, и найти прекрасные бесплатные сценарии в сети.

Теперь исправим функцию `MakeVisible()`, чтобы она первоначально скрывала все меню и затем выводила то, которое вам нужно.

```
function MakeVisible(element){
  ResetAllMenus()

  // Находит element и отображает его
  var element = document.getElementById(element)
  element.style.display = "block"
}
```

Было бы неплохо скрыть все плавающие окна меню, если ваш гость отводит указатель мыши от окна ссылок и направляет его на оставшуюся часть страницы. Это действие предполагает, что посетитель решил не щелкать мышью команду меню и вернуться к чтению страницы.

```
<div class="MainBody" onmouseover="ResetAllMenus()">
```

Код в загружаемом примере немного усложнен. Он скрывает границы вкладок и изменяет их фон. Но основной подход тот же самый.

Чужие меню

Если вы потратили достаточно свободного времени на написание своего кода на JavaScript, вы будете рады услышать, что Web-пространство набито бесплатными сценариями меню. Во многих из них гораздо больше блеска, чем в меню с вкладка-

ми, показанном в предыдущем примере. Вы можете найти следующие дополнительные варианты:

- многоуровневые меню, позволяющие вашим посетителям добираться до конкретных подкатегорий;
- меню, в которых можно скрывать и раскрывать подразделы, отображая одновременно все интересующие вас ссылки;
- забавные визуальные эффекты, например, затененная подсветка и прозрачный фон.

Для поиска хорошего меню можно воспользоваться сайтом образцов на JavaScript (см. *разд. "Сценарии в Web-пространстве" главы 15*) или выполнить поиск в Web "Меню JavaScript" или "Меню CSS". Вы увидите, что у меню немного больше разнообразия, чем у кнопок-ролловеров. Все меню выглядят и ведут себя немного поразному. Одни всплывают, другие выскальзывают, а третьи эмулируют внешний вид популярных программ, таких как Microsoft Outlook.

Совет

Не применяйте меню, которые заставляют прятать ваши ссылки в блок кода на JavaScript. Не только потому, что такой подход обеспечивает худшую поддержку, но и потому, что он создает проблемы для поисковых машин, которые не смогут обнаружить (и проиндексировать) все страницы на вашем Web-сайте. Современные меню должны помещать ссылки в реальные элементы <a>, которые затем вы можете расположить один за другим или поместить внутрь маркированного списка.

Посетите сайт Dynamic Drive с набором стильных меню на странице **www.dynamicdrive.com/dynamicindex1** и особенно интересным видом, названным меню

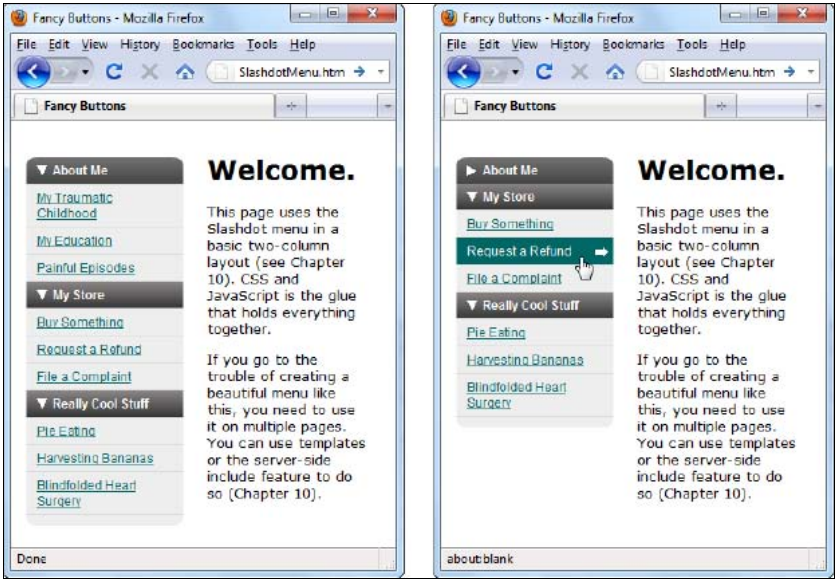


Рис. 16.8. Меню Slashdot сворачиваемое, поэтому можно скрыть разделы, которые вас не интересуют. Более того, можно скрупулезно настроить цвета, пиктограммы и границы меню, что позволит меню вписаться в любой дизайн сайта

Slashdot, на странице **www.dynamicdrive.com/dynamicindex1/slashdot.htm**. На рис. 16.8 показано это меню со структурой, как у меню с вкладками, обсуждавшемся ранее в этой главе.

> ***Совет***
>
> Перед выбором навигационного меню для сайта протестируйте побольше вариантов меню. Этот раздел познакомил вас с процессом, но нужно сравнить результаты с другими меню, прежде чем остановиться на каком-то одном.

В следующих разделах вы загрузите сценарий для создания меню Slashdot и поместите его на страницу.

Получение сценария

Для загрузки меню Slashdot выполните следующие действия.

1. Перейдите на страницу **www.dynamicdrive.com/dynamicindex1/slashdot.htm**.

   На этой странице показан образец меню и предлагаются инструкции по его применению. На ней также описана совместимость браузеров и меню, к счастью, оно работает практически во всех основных браузерах.

2. Найдите ссылку загрузки файла sdmenu.zip. Щелкните ее кнопкой мыши и затем сохраните ZIP-файл где-нибудь на своем компьютере.

   Для использования меню вам понадобятся файл на JavaScript, несколько изображений и таблица стилей. Все они включены в файл ZIP-архива.

3. Распакуйте файл sdmenu.zip. Поместите содержимое в папку сайта на вашем компьютере вместе с остальными страницами вашего сайта.

   Файл sdmenu.zip включает образец страницы с меню (index.html) и подпапку со всеми вспомогательными файлами (sdmenu). Перетащите мышью оба компонента в папку вашего сайта.

   Нет нужды трогать содержимое папки sdmenu (если только вы не хотите переопределить меню для соответствия вашему Web-сайту). Аккуратно заменив некоторые графические файлы, вы сможете изменить фон заголовков (toptitle.gif и title.gif), стрелки (collapsed.gif, expanded.gif и linkarrow.gif) и нижнюю границу (bottom.gif). Раскрыв таблицу стилей (sdmenu.css), можно модифицировать цвета фона, границы и пробельную зону для остальных компонентов меню. Кроме того, папка sdmenu содержит файл на JavaScript (sdmenu.js), который вы, вероятно, не будете редактировать вообще.

4. Создайте новую Web-страницу (или начните редактировать ту, на которой уже есть набор ссылок меню, который вы хотите преобразовать в меню Slashdot).

   Вы можете отредактировать вашу страницу index.html (и это хороший вариант для начала работы с большинством примеров). Однако меню Slashdot простое и похоже на сворачивающееся меню, которое вы сделали самостоятельно чуть раньше, поэтому его будет легко вставить и на новую страницу.

Создание меню

Начать применение меню Slashdot следует с присоединения к вашей странице таблицы стилей и вставки ссылки на файл JavaScript, управляющий меню. Вам также понадобится коротенький сценарий, создающий меню при загрузке страницы. Все три компонента помещаются в раздел <head> вашей страницы. Далее показано, как они выглядят.

```
<head>
  <title>Fancy Buttons</title>

  <link rel="stylesheet" type="text/css" href="sdmenu/sdmenu.css" />
  <script src="sdmenu/sdmenu.js"></script>

  <script>
    var myMenu;
    window.onload = function() {
      myMenu = new SDMenu("my_menu");
      myMenu.init();
    };
  </script>
...
</head>
```

Код сценария стандартный. Вы можете копировать его слово в слово на любую страницу, применяющую меню Slashdot. Следует сделать единственное замечание, касающееся имени меню, используемого в сценарии (в данном примере my_menu), оно должно совпадать с ID элемента на вашей странице, содержащего меню.

Возможно, вы захотите вставить в ваш раздел <head> встроенную таблицу стилей или ссылку на другую таблицу стилей. Например, на рис. 16.8 применяются три базовых правила стиля. Одно задает шрифт по умолчанию на странице, второе определяет местоположение боковой панели, содержащей меню Slashdot, а третье задает местоположение раздела с основным контентом страницы.

```
body {
  font-family: Verdana,sans-serif;
  font-size: small;
}

.MenuBar {
  position: absolute;
  top: 20px;
  left: 0px;
  margin: 15px;
}

.MainContent
{
  margin-left: 180px;
```

```
  margin-top: 30px;
  margin-right: 20px;
}
```

В этих стилях нет ничего нового. Вы видели их в предыдущих примерах.

Пришло время создавать меню. Его структура очень похожа на предыдущие примеры в этой главе. По сути, каждый сворачиваемый раздел меню состоит из контейнера `<div>`, заполненного элементами-якорями. Единственное дополнительное свойство — текст заголовка, содержащийся в элементе `<span>`, который вставлен в начало элемента `<div>`.

```
<div>
  <span>About Me</span>
  <a href="...">My Traumatic Childhood</a>
  <a href="...">My Education</a>
  <a href="...">Painful Episodes</a>
</div>
```

Типичное меню Slashdot содержит несколько сворачиваемых разделов. Вы заключаете их все в еще один элемент `<div>` и присваиваете ему имя, совпадающее с именем меню в сценарии.

```
<div class="sdmenu" id="my_menu">
   ...
</div>
```

Этого достаточно для создания меню Slashdot со всем его форматированием и функциональностью в полном составе. Но возможно вам захочется заключить `<div>` для Slashdot в еще один `<div>`, представляющий боковую панель меню. В этом случае вы сможете поместить боковую панель точно туда, куда захотите, не беспокоясь о конфликтах в таблице стилей или модификации файла sdmenu.css.

```
<div class="MenuBar">
  <div class="sdmenu" id="my_menu">
    ...
  </div>
</div>
```

Далее приведена разметка целиком для меню, показанного на рис. 16.8.

```
<div class="MenuBar">
  <div class="sdmenu" id="my_menu">
    <div>
      <span>About Me</span>
      <a href="...">My Traumatic Childhood</a>
      <a href="...">My Education</a>
      <a href="...">Painful Episodes</a>
    </div>
    <div>
      <span>My Store</span>
      <a href="...">Buy Something</a>
```

```
      <a href="...">Request a Refund</a>
      <a href="...">File a Complaint</a>
    </div>
    <div>
      <span>Really Cool Stuff</span>
      <a href="...">Pie Eating</a>
      <a href="...">Harvesting Bananas</a>
      <a href="...">Blindfolded Heart Surgery</a>
    </div>
  </div>
</div>
```

Примечание

После того как доведете свой Web-сайт до совершенства и будете готовы отпустить его в люди, не забудьте выгрузить подпапку sdmenu и все ее файлы.

ГЛАВА 17

Аудио и видео

На заре Интернета Web-сайты были такими же красочными, как формы Налогового управления США. Вы видели страницы, заполненные обычным текстом, ссылками и снова текстом. Со временем Web возмужал и Web-страницы стали меняться, когда Web-дизайнеры научились использовать возможности цвета, изображения и броские иллюстративные вставки. Но когда острота впечатления стала сглаживаться, появилось новая забава — *мультимедиа*.

Мультимедиа — собирательный термин для множества технологий и типов файлов, предъявляющих очень разные требования к компьютерным системам и ставящих различные задачи в Web-дизайне. Мультимедиа включает все, начиная с вызывающего раздражение звяканья на исходной странице сайта вашего лучшего друга и заканчивая столь популярным видеоклипом кота, играющего на пианино. (Печальный факт: у этого кота 20 миллионов просмотров, вы вряд ли когда-нибудь создадите Web-страницу, которая будет хотя бы на половину столь же популярна.)

В этой главе вы прочтете, как применять несколько типов мультимедиа. Сначала вы научитесь проигрывать фоновую мелодию и помещать современный MP3-плейер на Web-страницу. Потом вы будете встраивать видео — сначала с помощью обычного языка HTML, а затем средствами нового, но еще не совсем готового элемента `<video>` из стандарта HTML5. В заключение вы узнаете, как избежать некоторых серьезных проблем, размещая свои видеофайлы на YouTube.

Примечание

Прежде чем двигаться дальше, найдите время рассмотреть самые плохие примеры неправильного использования мультимедиа. К ним относятся сверкающие рекламные баннеры, раздражающая фоновая музыка, отнимающие время страницы-заставки и поглощающие полосу пропускания коммерческие рекламные объявления. Прежде чем применить мультимедиа, подумайте о том, что хотите сделать. Вы планируете показать ваши музыкальные композиции или предоставить загружаемые записи первых шагов Жуниора (Junior)? Если так, возможно мультимедиа имеет смысл. Но если вы просто ищете способ поразить посетителей логотипом с анимацией, дважды подумайте. Быть может разработка чего-либо, способного только раздражать большую часть посетителей, не стоит затраченных усилий.

Представление о мультимедиа

Пришло время и всем новым Web-дизайнерам захотелось не просто текста и картинок на своих страницах. Украшающие шрифты и элегантные макеты страниц не удовлетворяли, заставляя многих новичков чувствовать острую зависть при обнаружении сайта, загружающего звук и видео. Это понятно: вы тоже хотите поэкспериментировать на ваших страницах со звуком и видеоинформацией. Но прежде чем оживлять свой сайт, следует усвоить некоторые основы.

Связывание, встраивание и хостинг

Ключевое решение, которое вы принимаете, наполняя ваши страницы мультимедиа, — выбор между связыванием или встраиванием файлов.

Связывание с мультимедийным контентом — простейший, но наименее эффектный подход. Ссылка указывает на аудио- или видеофайл, хранящийся вместе с остальными вашими HTML-страницами и файлами. Ничего не стоит установить связь с данными мультимедиа. Вы используете тот же самый прозаичный элемент-якорь и атрибут `href`, который применяли в *главе 8*. Далее приведен пример.

```
Would you like to hear <a href="IndustrialNoiseBand.mp3">Industrial
  Noise</a>?
```

На рис. 17.1 показано, что происходит, когда вы щелкаете кнопкой мыши одну из таких ссылок.

> ***Примечание***
>
> Когда вы вставляете на ваш сайт аудиоданные, абсолютно неважно, какое программное обеспечение установлено на сервере вашего Web-хоста. Когда кто-либо щелкает кнопкой мыши ссылку на аудиофайл, браузер загружает файл на компьютер посетителя и воспроизводит его там, а не на сервере.

Встраивание мультимедиа — гораздо более профессиональный подход. Он включает музыку или видео в состав вашей HTML-страницы. В результате вы можете создавать чудесные комбинации текста, звука и видео.

Но встраивание мультимедиа может стать непростой задачей. В языке HTML нет прямой поддержки аудио и видео. Вместо этого браузеры полагаются на плагины (небольшие программки, расширяющие функциональные возможности браузеров). К несчастью, у вас нет способа узнать, какие плагины есть у ваших посетителей и (в дополнение) какие аудио- и видеоформаты их браузеры умеют воспроизводить. Вывод? Нет гарантии, что ваши посетители смогут увидеть какой-то конкретный тип файла мультимедиа, встроенный в вашу страницу.

Если вы продолжаете настаивать на встроенном мультимедиа, придется использовать HTML-элемент `<embed>` с немного сомнительной репутацией; вы узнаете, как он действует, в *разд. "Элемент <embed>" данной главы*. А в *разд. "Видео в HTML5"* вы прочтете о решении, которое уже не за горами: стандарт HTML5, который вводит соответствующий элемент `<video>`. Загвоздка лишь в том, что компании, выпускающие браузеры, до сих пор спорят о том, какие видеоформаты поддерживать.

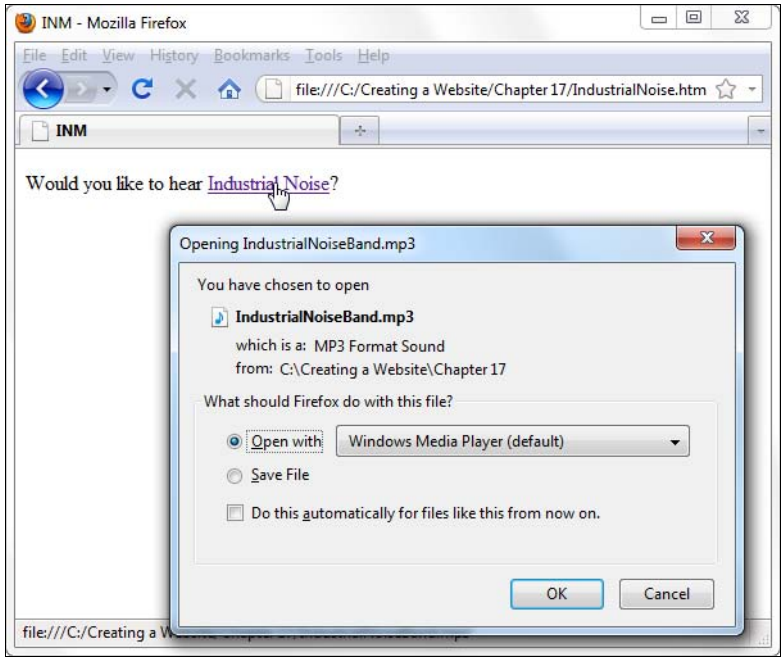


Рис. 17.1. Когда вы щелкаете кнопкой мыши ссылку на мультимедийный файл, ваш браузер предлагает сохранить файл или сразу открыть его. Если вы выбираете последнее, браузер сначала загружает файл, а затем запускает его с помощью отдельной программы. Программа, выбираемая браузером для воспроизведения файла, зависит от программного обеспечения, установленного на вашем компьютере. Например, это может быть Windows Media Player или QuickTime Player компании Apple

Примечание

Разница между связыванием и встраиванием мультимедиа такая же, как разница между ссылкой на изображение (с помощью элемента `<a>`) и встраиванием его непосредственно в вашу страницу (с помощью элемента `<img>`). Единственное отличие состоит в том, что изображения — базовая, хорошо поддерживаемая часть HTML-стандарта, поэтому встроенные изображения никогда не вызывают больших хлопот. Встроенные мультимедийные данные погружают в неизведанные пучины.

Но есть еще один способ управления мультимедиа — *выложенное мультимедиа* (hosted multimedia), файлы мультимедиа, хранящиеся на чьем-либо сервере, но выводящиеся или указанные в ссылках на вашей Web-странице. Самый известный пример выложенного мультимедиа — YouTube, ужасно популярный сайт, проигрывающий сотни миллионов видеоклипов ежедневно.

Выложенное мультимедиа — отличный вариант, если вы хотите отображать действительно большие файлы, в особенности кинофрагменты. Этот способ не снижает вашу пропускную способность (см. *разд. "Определение ваших потребностей" главы 3*) и работает практически со всеми браузерами и операционными системами. Единственный недостаток — потеря существенной доли контроля. Например, если вы выложили ваши видеофайлы на YouTube, вы не сможете показать кинофильмы, длящиеся более 15 минут, и YouTube может несколько снизить качество ваших фильмов, добиваясь их приемлемого воспроизведения. (Технически YouTube

уменьшает размер видеофайлов, чтобы браузеры могли быстрее загружать их, в этом случае посетители не ощущают задержки при воспроизведении, когда нажимают соответствующую кнопку.) Но YouTube до сих пор вне всякого сомнения является самым практичным способом совместного использования клипов. Он даже объединяется с вашим Web-сайтом, и для демонстрации собственных видеофайлов вы можете поместить окно YouTube на любую Web-страницу. Как применять YouTube, вы узнаете в *разд. "Выгрузка вашего видео на YouTube" данной главы*.

Типы файлов мультимедиа

Решение связать или встроить файлы зависит по крайней мере отчасти от типа мультимедийных данных, которые вы хотите показать. Набор возможных вариантов может слегка сбить с толку.

- *Synthesized music* (Синтезируемая музыка): MIDI — Musical Instruments Digital Interface, Интерфейс электронных музыкальных инструментов. Файлы MIDI хранят ноты, генерируемые вашим компьютером во время воспроизведения песни в отличие от воспроизведения записи музыкального инструмента. Как результат, у файлов MIDI маленький размер, но сомнительное качество. Если вам не нужен эффект модной клавиатуры Casio, вы, вероятно, и не будете ими пользоваться.
- *Digital audio* (Цифровое аудио): WAV и MP3. Файлы этих типов хранят записанный звук, что означает более высокое качество, чем у файлов MIDI. Но файлы формата WAV громадные, что делает их непригодными за исключением раздутых Web-сайтов. MP3-файлы составляют одну десятую часть размера файла WAV, но, когда вам потребуется воспроизвести их, у некоторых браузеров может не оказаться нужного плагина.
- *Digital video* (Цифровое видео): MPEG, AVI, MOV и WMV. Файлы этих типов — крупнейшие игроки на поле мультимедиа. Они позволяют воспроизводить видео самого разного качества, начиная с крошечных окон с прерывистым воспроизведением и заканчивая фильмами с качеством как на DVD-дисках. Необработанные файлы с цифровым видео — проблема для любого создателя Web-страницы из-за их гигантских размеров. Для того чтобы добиться воспроизведения цифровых видеоданных без сбоев, вам, возможно, придется сжать, сократить и уменьшить размер вашего клипа и снизить его качество с помощью программы редактирования видео.
- *Animated GIF*. Файлы в формате Animated GIF состоят из последовательности маленьких изображений-кадров, воспроизводимых с большой скоростью одно за другим, напоминая разновидность мультипликации (кинеограф). В старые добрые времена Интернета GIF-анимация применялась всюду, но в наши дни она становится историей, и лишь немногие Web-разработчики с печалью наблюдают за этим процессом.
- *Flash*. Flash — это многофункциональный плагин, имеющий двойное назначение. С одной стороны, это программное средство для создания интерактивных

Web-приложений высокого качества (например, игры на странице **www.ferryhalim.com/orisinal**). В то же время Flash — это способ воспроизведения видео (хранящегося в файлах с расширением FLV) на компьютерных устройствах практически любого типа за исключением iPad и iPhone (компания Apple не поддерживает Flash-стандарт для мобильных устройств).

Несмотря на эти впечатляющие плюсы, у Flash есть три недостатка. Во-первых, для создания Flash-приложения или видеопроигрывателя Flash необходимо специальное программное обеспечение компании Adobe стоимостью в несколько сотен долларов. Во-вторых, даже если вы раскошелитесь на оплату Flash, необходимо еще и обучение. Для создания обычного видеоплейера не обойтись без руки помощи, а для разработки настоящего Flash-приложения потребуются серьезные профессиональные знания. И наконец, посетители не увидят Flash-видеоклипы, если у них нет установленного плагина Flash. (При этом по достоверным оценкам предполагается, что у более 99% настольных компьютеров с выходом в Интернет уже есть этот плагин.)

ПРИМЕЧАНИЕ

Компании, предлагающие услуги хостинга мультимедиа, такие как YouTube, для воспроизведения своих видеоклипов применяют Flash (рис. 17.2). Дело в том, что Flash обеспечивает наилучшее сочетание возможностей настройки, производительности и совместимости. Конечно, у этих ведущих компаний полно денег, чтобы платить своим программистам.

Понятно, что переварить всю эту информацию сразу довольно трудно. Если вы все еще не определились с выбором, просмотрите сценарии в табл. 17.1, которые помогут вам разобраться в особенностях воспроизведения мультимедиа разных типов.

Таблица 17.1. *Сценарии работы с мультимедиа*

Если вы хотите:	То используйте:	Вид размещения
Воспроизводить непрерывно в фоновом режиме короткий фрагмент цифрового аудио	Flash (можно также применить формат MP3, но его поддерживают не все браузеры и многократное повторение будет менее гладким)	Встроенное
Дать возможность посетителям увидеть ваш домашний видеофильм	Файлы форматов MPEG, AVI, WMV или MOV. (Если размер файла слишком велик, обязательно примените программу редактирования видео для уменьшения размера файла)	Размещенное на хосте (на сервисе, таком как YouTube)
Показать экран-заставку с анимацией, коммерческую рекламу или сюжет	Flash	Встроенное

ПРИМЕЧАНИЕ

Если вы планируете создать Web-сайт с большим объемом аудио и видео, надо пересмотреть требования сайта к объему дискового пространства и величине Web-трафика (см. *разд. "Контрольный список для выбора Web-хоста" главы 3*). В отличие

от обычных HTML-страниц и Web-графики файлы мультимедиа могут быть очень большими, способными переполнить отведенное пространство на Web-хосте и превысить пропускную способность. Этой проблемы можно избежать, используя сервис хостинга мультимедиа, например, YouTube, тогда просмотры видео не будут требовать от вас ни Web-пространства, ни пропускной способности.

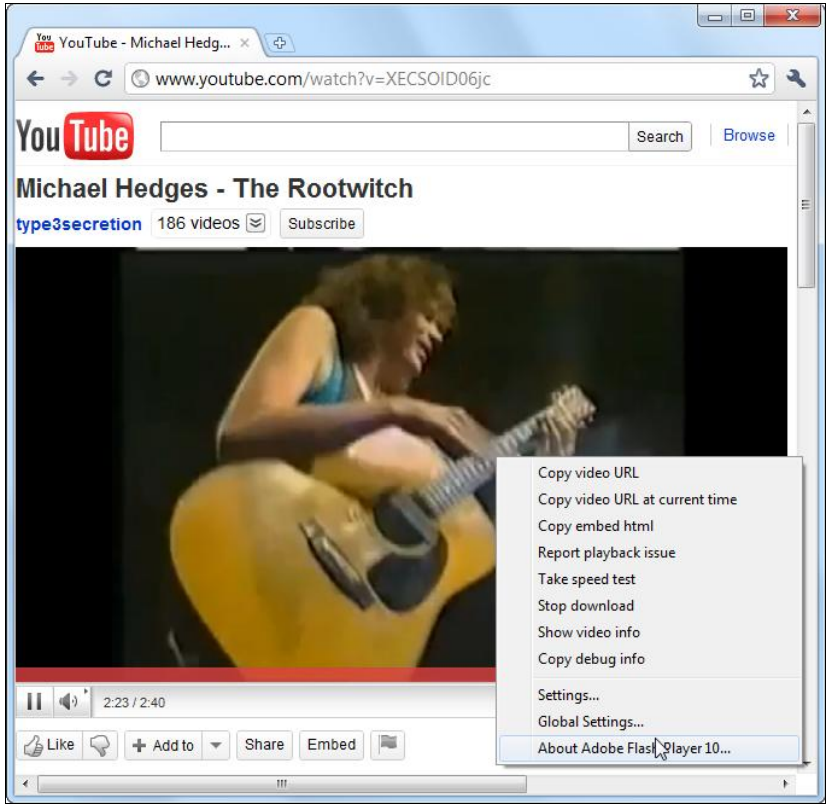


Рис. 17.2. Если вы предполагаете, что анимационная коммерческая реклама, видеоклип или игра — Flash-приложение (обычно оно так и есть), щелкните его правой кнопкой мыши (на машинах Mac левой кнопкой с нажатой клавишей <Ctrl>). Если перед вами Flash-файл, в меню, которое появится, будет строка наподобие "About Flash" (О Flash)

Стандартная фоновая музыка

Большинство людей любит просматривать Web в умиротворяющей тишине. Это означает без транс-гипно-обволакивающих композиций, без странных ритмов диско и без грохота тарелок. Подобная антипатия к шуму, быть может, объясняется тем, что примерно 98% всех Web-просмотров фиксируется в рабочее время.

Но если вы любите пугать и раздражать людей или абсолютно убеждены, что вашей Web-аудитории необходимы привычные музыкальные ритмы, продолжайте чтение данной главы, ведущее прямо к фоновой музыке.

Элемент <embed>

Несмотря на то что стандарт HTML не поддерживает фоновую музыку, почти все браузеры поддерживают элемент <embed>, впервые введенный Netscape в дни зарождения Web. Элемент <embed> можно поместить в любое место вашей страницы. Далее приведена типовая страница, использующая его для проигрывания фоновой музыки.

```
<!DOCTYPE html>

<html>
<head>
  <title>Background Music</title>
</head>
<body>
<h1>Automatic, Unsolicited Music</h1>
<p>The music now blaring from your speakers is
Scarlatti's first sonata (K. 500).
I hope you didn't tell your colleagues you were working!</p>
<embed src="scarlatti.mp3" type="audio/mpeg" />
</body>
</html>
```

Элемент <embed> предлагает массу параметров, управляющих воспроизведением. Если применить элемент без параметров (как в предыдущем примере), ваши посетители увидят страницу, показанную на рис. 17.3, и услышат ее аудиофайл, проигрываемый автоматически.



Рис. 17.3. Если вы применяете на вашей странице простейший элемент <embed>, там куда вы поместили элемент, появляются кнопки управления воспроизведением. Конкретный стиль кнопок управления меняется в зависимости от вашей операционной системы, браузера и аудиоплагинов

Воспроизведение музыки не всегда проходит гладко. Поскольку каждый браузер обрабатывает встроенную музыку немного по-разному, можно столкнуться с такими проблемами, как приведенные на рис. 17.4. Лучше всего протестировать стра-

ницу как минимум в трех основных браузерах (Internet Explorer, Firefox и Chrome) и подумать о вставке на вашу страницу текста, поясняющего проблемы, с которыми могут столкнуться гости.

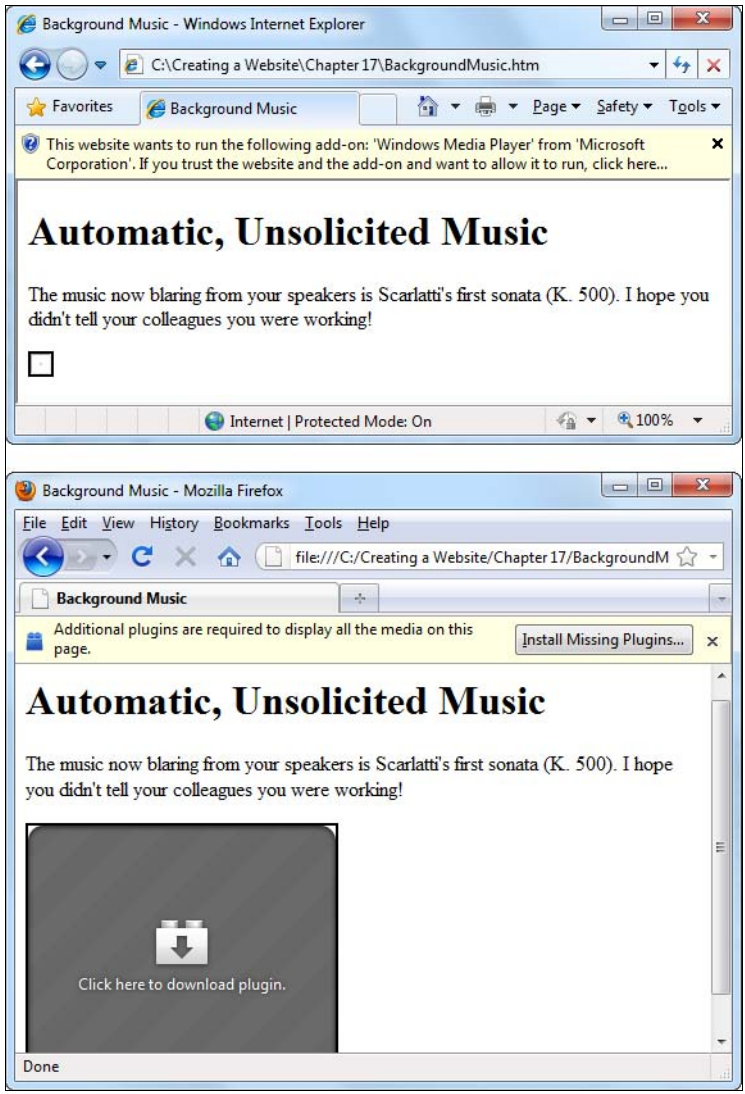


Рис. 17.4. *Вверху*: параноидальные браузеры могут блокировать вашу музыку из-за настроек их системы безопасности.
Внизу: в зависимости от того, что ваш посетитель установил или деинсталлировал, браузер может не обнаружить компонентов, необходимых ему для воспроизведения вашей фоновой мелодии

Параметры встроенного звука

Обычно элемент `<embed>` начинает проигрывать мелодию, как только ваш браузер загрузит заданный файл. Посетители могут убрать звук, быстро щелкнув мышью

кнопку останова (при условии отображения элементов управления воспроизведением), но если вы не ожидаете услышать музыкальное сопровождение, оно может ударить по нервам.

Более вежливый способ обработки фонового звука, отобразить элементы управления воспроизведением и позволить вашим посетителям решать, когда нажать кнопку воспроизведения или не нажимать ее никогда. Сделать это легко — просто примените атрибут `autoplay`.

```
<p>If you'd like some soft music to browse by, click the play button.</p>
<embed src="scarlatti.mp3" type="audio/mpeg" autoplay="false" />
```

Отключение автопроигрывания — считается признаком хорошего тона в Web-пространстве. Гораздо хуже использование атрибута `hidden`, позволяющего прятать все элементы управления воспроизведением. Чаще чем хотелось бы, встречаются Web-страницы, применяющие следующие элементы `<embed>`:

```
<embed src="scarlatti.mp3" type="audio/mpeg" hidden="true" />
```

В этом примере звуковой файл воспроизводится автоматически. И поскольку вы скрыли элементы управления, единственный способ остановить музыку — броситься к элементу управления громкостью. Web-сайты, подвергающие своих посетителей такому суровому испытанию, редко видят повторные визиты.

В элемент `<embed>` включены дополнительные параметры, подробно описанные в табл. 17.2.

Таблица 17.2. *Атрибуты элемента* `<embed>`

Атрибут	Описание
`src`	URL, который указывает на аудиофайл
`autoplay`	Значение `true` или `false`, обозначающее немедленное воспроизведение аудио (`true`) или ожидание нажатия кнопки воспроизведения (`false`)
`hidden`	Значение `true` или `false`, обозначающее видимость элементов управления воспроизведением
`loop`	Значение `true` или `false`, обозначающее однократное воспроизведение мелодии (от начала до конца) или бесконечное ее повторение. Во время повторного воспроизведения мелодии вы заметите в начале явную паузу
`volume`	Значение от 1 до 100, задающее громкость воспроизведения как процент от максимального значения. 100% — самый сильный звук, который можно получить. 50% стремится обеспечить стандартную громкость на компьютерах с Windows; на компьютерах Mac этот же эффект достигается при 75%. Если задать громкость 100%, можно быть уверенным, что вернувшихся посетителей не будет. Когда задаете атрибут `volume`, указывайте только число (без знака %)

Печально, но элемент `<embed>` не поможет создать такие прекрасные повторяющиеся саундтреки, которые вы, возможно, слышали на других Web-сайтах. Несмотря на то что `<embed>` поддерживает атрибут `loop`, результаты скверные, потому что мелодия замирает каждый раз, когда доходит до конца звукового файла. Если вы хо-

тите сгладить повторяющуюся мелодию, необходимо воспользоваться Flash, как описано в следующем разделе.

> ***Совет***
>
> Вам необходимо получить файлы MP3? Множество отличных свободно распространяемых программ могут помочь записать файлы WAV и превратить их в более компактный формат MP3. Есть две дешевые, которые можно опробовать бесплатно, — GoldWave (**www.goldwave.com**) и FlexiMusic (**www.fleximusic.com**). Если вам нужно только превратить имеющиеся файлы WAV в формат MP3, можно воспользоваться программой iTunes компании Apple с бесплатными версиями для Windows и Mac (**www.apple.com/itunes**). Получить файл можно, щелкнув правой кнопкой мыши (или левой кнопкой мыши с нажатой клавишей <Ctrl> для Mac) название мелодии и выбрав из раскрывающегося меню команду **Create MP3 Version** (Создать MP3-версию).

Flash-плейеры MP3

Как вы уже знаете, Flash — это браузерный плагин, позволяющий вставлять на Web-страницу видеоклипы, анимацию и даже миниатюрные программы, например игры. Несмотря на то что создание Flash-программы с нуля потребует больших усилий (и кое-какого недешевого программного обеспечения), вставить музыкальный Flash-плейер на вашу страницу далеко не так трудно. И все потому, что множество людей уже сделали основную работу за вас. В действительности Web наводнен бесплатными музыкальными Flash-плейерами.

Для того чтобы найти некоторые из них, задайте в строке поиска Google "flash mp3 player". Большая их часть на удивление отшлифована, поддерживает списки мелодий, кнопки воспроизведения и настраиваемые цвета и схемы. В следующих разделах вы увидите, как воспользоваться двумя хорошими образцами.

Плейер Premiumbeat: воспроизведение списка мелодий

Web-сайт **www.premiumbeat.com** предлагает набор профессионально выглядящих Flash-плейеров[1].

- Мини-плейер (Mini player). Он выглядит как кнопка воспроизведения, спаренная с крошечной пиктограммой в виде полоски (которая оживает во время воспроизведения). Используйте этот плейер, если на странице важнее всего свободное пространство.
- Узкий плейер (Thin player). Он выглядит как мини-плейер, но с длинной узкой полосой, позволяющей видеть, в каком вы месте звучащей мелодии. Это еще один вариант для страниц, не желающих привлекать внимание к музыкальному сопровождению.

[1] На момент выхода русского перевода книги указанный сайт начал перенаправлять посетителей со своей страницы **www.premiumbeat.com/flash_music_players** на дружественный сайт с плейерами. — *Прим. ред.*

- Стандартный плейер (Basic player). Этот вариант содержит полный набор кнопок управления, включая кнопки "следующая/предыдущая" и бегунок уровня громкости. Визуально это небольшое окно, выглядящее как кнопки управления на рис. 17.3, но при желании можно дополнительно вывести название мелодии и метку со временем.
- Плейер со списком воспроизведения (Player with playlist). Он выглядит как стандартный плейер, но содержит список всех мелодий, которые вы выбрали для воспроизведения. Этот плейер удобно использовать, если вы предлагаете несколько мелодий. Его можно увидеть на рис. 17.5.

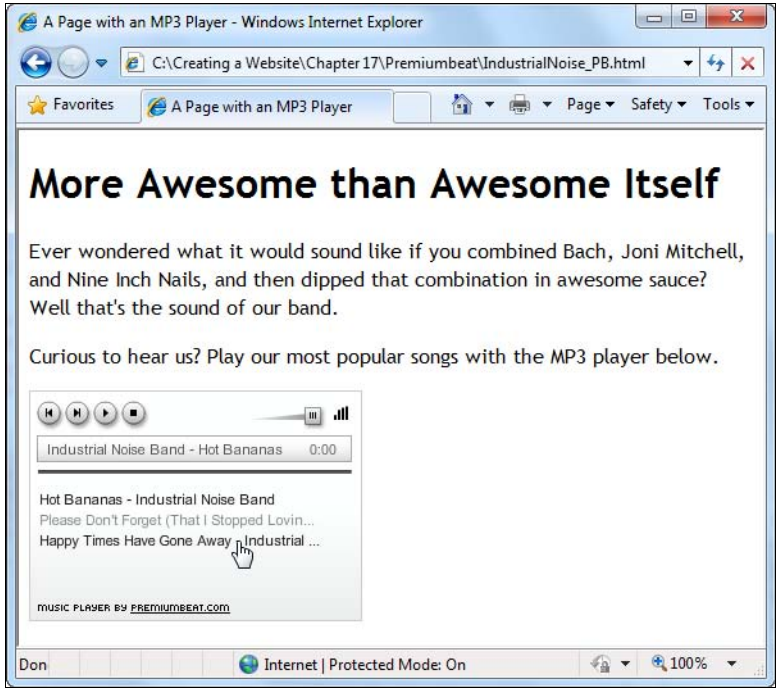


Рис. 17.5. Нажмите кнопку воспроизведения, и этот плейер проиграет весь свой список от начала до конца. Или дважды щелкните кнопкой мыши конкретную мелодию, чтобы послушать именно ее

Все плейеры предлагаются в двух вариантах: бесплатная версия и версия премиум. Если вы заплатите за версию премиум, то сможете удалить мелкий текст "Music Player by Premiumbeat.com", расположенный в нижней части плейера.

Для использования одного из плейеров Premiumbeat на своих страницах следуйте приведенным инструкциям.

1. Перейдите на страницу **www.premiumbeat.com/flash_music_players**. Выберите понравившийся вам плейер и щелкните мышью расположенную рядом кнопку **Download Free Version** (Загрузить бесплатную версию).

   Будет загружен ZIP-архив с небольшим набором файлов. Файл с расширением swf — это и есть настоящий музыкальный Flash-плейер. Вы не сможете открыть

его сразу, т. к. его нужно выгрузить на ваш Web-сайт, чтобы ваши страницы могли им пользоваться. Имя файла зависит от точной версии загруженного вами плейера. Если вы загрузили полнофункциональный плейер с поддержкой списка воспроизведения, вы увидите файл с именем OriginalMusicPlayerPlaylist.swf.

2. Создайте список воспроизведения.

   В списке воспроизведения перечисляются по порядку все файлы, которые вы хотите загрузить в плейер.

   Загруженный ZIP-файл включает пример списка воспроизведения, названный playlist.xml. Можно отредактировать этот файл или создать новый с другим именем.

   Строго говоря, список воспроизведения — это XML-файл, т. е. у него есть теги, как в HTML-файле. (Но в отличие от HTML эти теги вставлены для определения разных фрагментов информации для музыкального плейера, а не для подготовки документа к отображению в Web-браузере.) Структура списка воспроизведения очень проста. HTML заключает весь список в элемент `<playlist>` и вставляет в него по одному элементу `<item>` для каждой мелодии. Каждый элемент `<item>` в свою очередь включает элементы `<title>`, `<artist>` и `<path>`. Информацию из элементов `<title>` и `<artist>` плейер отображает во время воспроизведения, а элемент `<path>` указывает на файл MP3, который вы хотите услышать.

   Далее приведен полный список воспроизведения с тремя мелодиями.

```
<playlist>

  <item>
    <title>Hot Bananas</title>
    <artist>Industrial Noise Band</artist>
    <path>HotBananas.mp3</path>
  </item>

  <item>
    <title>Please Don't Forget (That I Stopped Loving You)</title>
    <artist>Industrial Noise Band</artist>
    <path>LoveSong.mp3</path>
  </item>

  <item>
    <title>Happy Times Have Gone Away</title>
    <artist>Industrial Noise Band</artist>
    <path>HappyTimes.mp3</path>
  </item>

</playlist>
```

СОВЕТ

В элементе `<path>` применяется та же система относительных ссылок, которую вы применяли раньше в якорях и изображениях (см. *разд. "Относительные ссылки и*

папки" главы 8). Это означает, что ссылка всегда задается относительно местоположения HTML-страницы, отображающей плейер. Для лучшей организации вашего сайта поместите все мелодии в подпапку (скажем, названную MP3) и затем измените в вашем плейере путь к мелодии (например, с HotBananas.mp3 на MP3/HotBananas.mp3).

3. Вставьте плейер на вашу страницу.

   Вам нужен только корректный элемент `<object>`. Легче всего получить его, скопировав содержимое из файла sampleEmbedObject.html, входящего в загруженный ZIP-файл. Элемент `<object>` выглядит следующим образом.

```
<object type="application/x-shockwave-flash" id="player1"
  allowscriptaccess="always" allowfullscreen="true"
  data="OriginalMusicPlayerPlaylist.swf" width="260" height="180">
    <param name="movie" value="OriginalMusicPlayerPlaylist.swf" />
    <param name="FlashVars" value="playlistXmlPath=playlist.xml" />
</object>
```

   Беспокоиться следует только об имени файла со списком воспроизведения (выделено жирным шрифтом в примере). Если вы дадите вашему списку другое имя, замените его соответственно. При желании можно задать свойства ширины и высоты, чтобы плейер лучше вписался в вашу страницу.

   В загруженном с Premiumbeat файле есть еще два образца страниц. На странице sampleEmbedJavascript.html плейер создается с помощью кода на JavaScript. Это остроумный прием, но полезный только при интеграции плейера в Web-приложения определенного типа. В образце sampleEmbedMultiplePlayers.html показано, как можно поместить несколько плейеров на одну страницу, хотя такой вариант может запутать ваших посетителей.

4. Испытайте плейер.

   Теперь, открывая свою Web-страницу в браузере, вы должны увидеть медиаплейер с вашим списком воспроизведения. Щелкните кнопкой мыши каждую мелодию, чтобы убедиться, что все ссылки действуют.

   Когда будете готовы выгрузить завершенный продукт, не забудьте включить в него вашу Web-страницу, файлы MP3, XML-файл со списком воспроизведения и файл плейера с расширением swf (например, OriginalMusicPlayerPlaylist.swf).

> ***Совет***
>
> Если хотите настроить шрифт плейера и цвета, посетите страницу **www.premiumbeat.com/flash_player_customization_tool**. Вы сможете выбрать нужные параметры, и средство настройки создаст разметку, которую вы должны будете вставить на вашу Web-страницу. Если же вы хотите более радикальных изменений, поищите Flash-плейер MP3, умеющий полностью изменять внешний вид. Такие плейеры позволяют использовать пользовательскую графику для всех деталей оформления, включая фон и кнопки воспроизведения.

Плейер Yahoo: воспроизведение всех ваших ссылок

Другой способ интеграции плейера с вашими страницами — использование медиаплейера Yahoo (**http://mediaplayer.yahoo.com**). Он особенно подойдет для про-

граммистов в цейтноте (или ленивых), потому что не требует практически никакой работы. Вам не нужно загружать ни единого файла, поскольку код для плейера находится на Web-серверах Yahoo. Вы просто вставляете на свою Web-страницу единственную ссылку на сценарий, например, такую:

```
<script src="http://mediaplayer.yahoo.com/js"></script>
```

Как вы узнали в *разд. "Внешние файлы сценариев" главы 15*, ссылка на сценарий позволяет вашей Web-странице применять код на JavaScript, хранящийся в отдельном файле. Это избавляет вас от необходимости копирования и вставки больших фрагментов кода на каждую страницу. Код на JavaScript для медиаплейера Yahoo особенно остроумен. Он реагирует на события на Web-странице. Как только браузер загружает вашу страницу, код на JavaScript просматривает ее в поисках обычных ссылок `<a>`, указывающих на поддерживаемые медиафайлы, например формата MP3. Каждый раз, когда он находит связанный MP3-файл, браузер вставляет рядом со ссылкой крошечную пиктограммку воспроизведения. (Конечно, весь процесс занимает всего несколько микросекунд.) На рис. 17.6, вверху, показан результат.

Самое интересное происходит, когда вы щелкаете кнопкой мыши одну из этих пиктограмм. Вместо предложения загрузить файл или попытки загрузки плагина, Yahoo раскрывает плавающий музыкальный Flash-плейер (рис. 17.6, внизу). В плавающем окне начинает проигрываться песня без всяких раздражающих предупреждений системы безопасности.

Далее приведена разметка для страницы, показанной на рис. 17.6. Как видите, это обычная страница с гиперссылками. Магия медиаплейера создается присутствием единственного элемента `<script>`.

```
<!DOCTYPE html>
<html>

<head>
  <title>A Page with the Yahoo Media Player</title>
  <style>...</style>
  <script src="http://mediaplayer.yahoo.com/js" />
</head>

<body>
  <h1>More Awesome than Awesome Itself</h1>
  <p>My band has released a number of hit songs that are so awesome, they
  turn sand to diamonds and make small crying babies cry less. Check them
  out.</p>
  <p>Our first release was <a href="song.mp3">Please Don't Forget (That I
  Stopped Loving You)</a>.
  <p>We followed up with <a href="soundfile.mp3">Happy Times Have Gone
  Away</a> and <a href="song.mp3">Hot Bananas</a>.
</body>
<html>
```

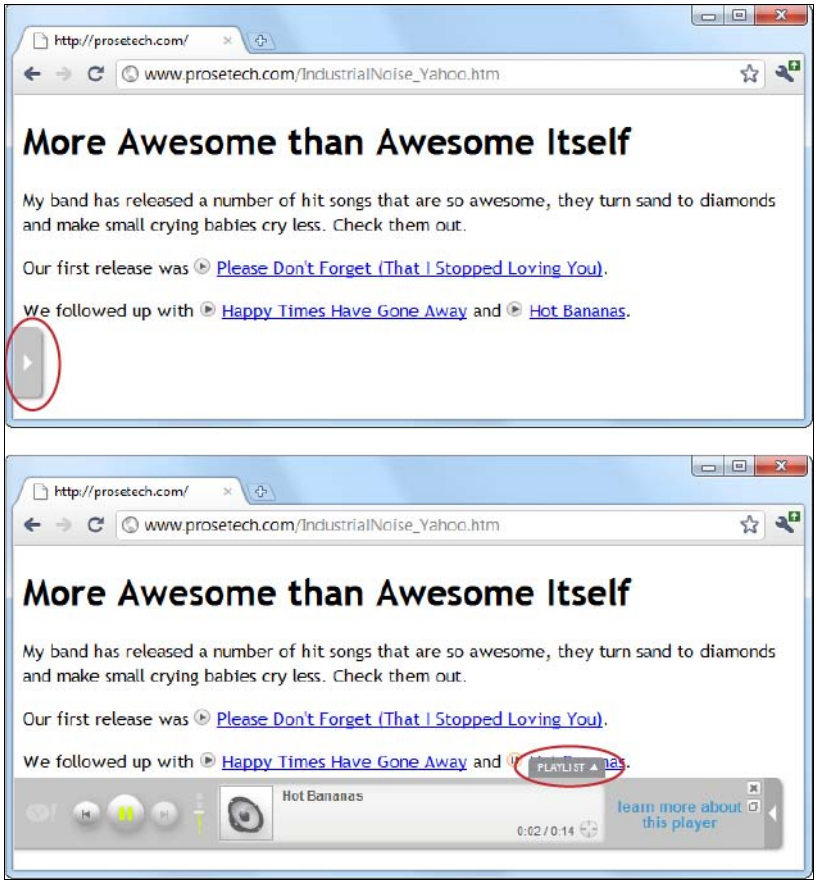


Рис. 17.6. *Вверху*: стрелка внизу справа на этой странице представляет автоматически созданный, но пока скрытый медиаплейер Yahoo. Щелкните стрелку кнопкой мыши, чтобы раскрыть ее, или щелкните мышью одну из пиктограмм кнопки воспроизведения, чтобы начать проигрывание. *Внизу*: медиаплейер Yahoo плавает поверх контента вашей страницы. Он добавляет все медиассылки на вашей странице в единый список воспроизведения, который можно увидеть, щелкнув мышью ссылку Playlist (обведена)

Хочу сделать два предупреждения. Во-первых, медиаплейер Yahoo относительно новый и не лишен иногда проявляющихся странностей. (Например, временами раскрывшийся плейер может исчезнуть, хотя кнопки воспроизведения рядом со ссылками продолжают работать). Во-вторых, он работает, только если вы выгрузили ваш сайт на реально действующий Web-сервер. Если попробовать открыть локальную копию страницы, использующей плейер (это страница, хранящаяся на вашем компьютере), он загружает себя, но ничего не воспроизводит. Проблема в том, что ваши MP3-файлы замурованы на жестком диске, куда медиаплейер Yahoo не может получить доступ.

Лупы формата Flashtrak

В предыдущих разделах вы видели два отличных способа вставки MP3-плейера на любую Web-страницу. Но порой вам нужно нечто, более простое. Вместо предос-

тавления вашим посетителям возможности перебирать коллекцию песен, вы можете просто осчастливить их бесконечной фоновой музыкой.

Хотя многие Web-сайты продают звуковые лупы (короткие мелодии, повторяющиеся бесконечное число раз), можно загрузить и бесплатные с сайта Flash Kit, **www.flashkit.com/loops** (рис. 17.7). Flash Kit предлагает большой отличный каталог примерно из 10 000 лупов в разных стилях, от обычных до урбанистических.

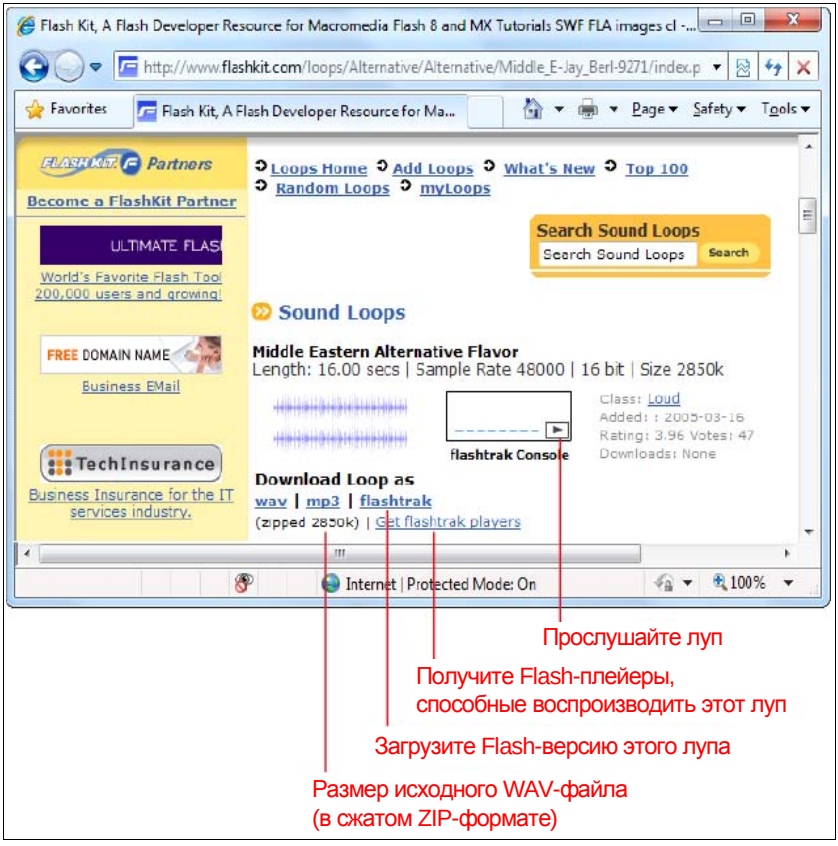


Рис. 17.7. Вы можете предварительно прослушать в вашем браузере лупы с сайта Flash Kit, не загружая их. После того как нашли то, что нужно, щелкните кнопкой мыши ссылку **flashtrak** для загрузки звука в одном из трех форматов: MP3 (обычный), WAV (хорош для последующего редактирования) или в формате Flashtrak, работающем со специализированными плейерами, имеющимися на сайте. В данном примере высококачественный 10-секундный файл WAV весит почти 2 Мбайт, MP3-версия — более приемлемые 700 Кбайт. Размер файла в формате Flashtrak и того меньше — всего 200 Кбайт

Примечание

Лупы — эквивалент обойной мозаики. Это короткие музыкальные фрагменты, разработанные специально так, что начало подхватывается, как только завершился конец. Вы можете проигрывать луп снова и снова и в результате получите бесперебойную фоновую композицию. В лупе высшего разряда повторение не сразу угадывается, и вы можете с наслаждением слушать его в течение нескольких минут.

Если вы загрузите один из этих лупов как MP3-файл, вы сможете использовать его с бесплатным Flash-плейером MP3, как уже обсуждавшиеся ранее файлы. Но есть и другая альтернатива, файл, применяющий сокращенный аудиоформат, называемый Flashtrak. Файлы формата Flashtrak загружаются вмиг, поэтому вашим посетителям никогда не придется ждать, чтобы ощутить атмосферу вашего сайта. Для этих файлов нужен плейер Flashtrak, который также можно загрузить бесплатно с Web-сайта Flash Kit. (Плейер Flashtrak — это Flash-программа, как и музыкальные плейеры Premiumbeat и Yahoo, но она не воспроизводит MP3-файлы.)

Для загрузки плейера Flashtrak найдите ссылку **Get flashtrak players** (Получить плейеры flashtrak), расположенную прямо под ссылками на лупы (см. рис. 17.7). Вы можете выбирать из более десяти стилей плейера. У большинства из них есть отличные визуальные эффекты во время проигрывания мелодии, например пульсирующие линии или расширяющиеся окружности.

Примечание

После загрузки плейера Flashtrak вы можете обнаружить лишние файлы. Например, вам не нужны и могут быть удалены любые файлы с расширением fla (это исходные Flash-файлы, которые могут редактироваться только Flash-программой). Кроме того, после загрузки плейера вы, возможно, найдете у себя массу дополнительных файлов с песнями. Удалите те, которые вам не нужны, иначе ваш плейер будет воспроизводить их все по очереди.

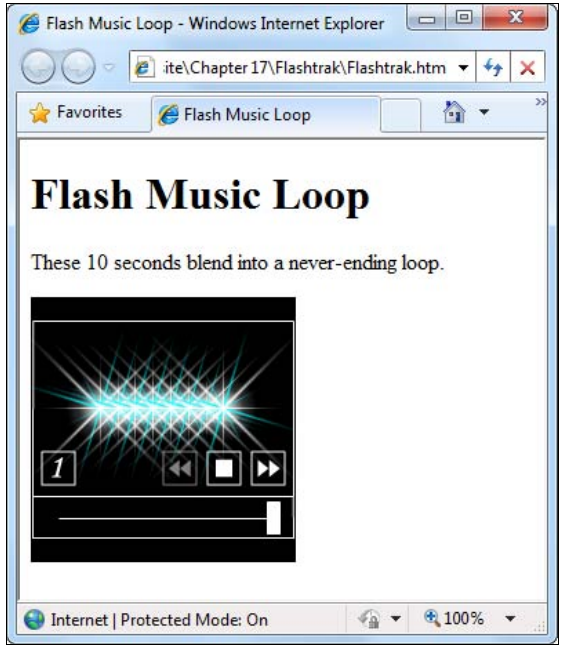


Рис. 17.8. На рисунке показаны в действии звуковые элементы управления на базе Flash-технологии, укомплектованные кнопками воспроизведения и нераздражающей графикой. Лучший способ опробовать этот пример (и примеры MP3-плейеров, приведенные ранее) — загрузить содержимое примеров к данной главе со страницы **Missing CD** по адресу **www.missingmanuals.com/cds/caw3**

После загрузки плейера, его можно встраивать в вашу Web-страницу. Позаботиться об этом можно с помощью простого элемента <embed>, указывающего на файл плейера.

```
<embed src="StarPlayerMultiTrackWithAutoStart.swf"></embed>
```

На рис. 17.8 показано, что вы увидите, когда откроете страницу, содержащую этот элемент.

Видеоклипы

Теперь, когда вы справились с проблемами звука и узнали, как поместить на ваши Web-страницы все, начиная со звуковых эффектов и заканчивая повторяющейся фоновой мелодией, можно переходить к решению следующей задачи — *видеоконтенту*.

Хотя браузеры для воспроизведения видеофайлов применяют многие из средств, используемых для проигрывания звуковых файлов (плагины Windows Media Player, QuickTime и Flash), есть ряд серьезных отличий. Самое важное — большой размер видеофайлов. Самый маленький из них во много раз больше аудиозаписи симфонии Малера без сокращений и купюр. Обработка таких данных — настоящий тест на долготерпение для ваших посетителей. В следующих разделах вы узнаете, как подготовить ваши видеофайлы для публикации в Web и дать возможность вашим посетителям увидеть их.

Подготовка видеофайлов

Вставка персональных видеофайлов на Web-сайт — задача, посильная для трудолюбивых знатоков мультимедиа. Главный камень преткновения — огромный размер цифрового видео. На цифровой видеокамере каждая секунда видео съедает от 1 до 3 Мбайт (в зависимости от качества записи и выбранного вами формата). Смонтируйте 10-минутный клип, и вы с изумлением увидите перед собой файл размером от 600 до 1800 Мбайт. Он не только неудобен в управлении, но и отхватывает значительный кусок от выделенных Web-разработчику пространства на сервере и пропускной способности.

Что можно сделать, чтобы Web-видео и выглядело хорошо, и исполнялось как следует? Вы всегда можете воспользоваться чужими видеофайлами, подготовленными для Web (или заплатить уйму денег компании, занимающейся редактированием видео, чтобы она сократила ваше видео до размеров, приемлемых в Web). При условии, что это не то, что вам нужно, остается два варианта.

- Запись с более низким качеством. Многие видеокамеры позволяют записывать с заданным низким качеством с единственной целью — поместить видео в Интернет. Мобильные телефоны, Web-камеры, мини-видеокамеры и цифровые фотоаппараты создают фильмы низкого качества, позволяя избежать проблем преобразования и отправить видео сразу на ваш сайт.

- Последующее снижение качества. Гораздо чаще вам придется проходить через длительный процесс перекодирования вашего высококачественного видео для получения размера файла, подходящего для Web. Для этого вам потребуется программа редактирования видео. Видеокамеры обычно поставляются с каким-нибудь программным средством, способным помочь, но возможно вы решите заплатить за более мощную программу. Есть два основных варианта — Windows Movie Maker, поставляемая с ОС Windows, и iMovie для компьютеров Mac. У многих программ редактирования видео есть средство, автоматически выбирающее настройки низкого качества для видеофайлов, которые вы хотите выгружать в Интернет.

Далее перечислены действия, которые нужно выполнить для получения видео, готового для Web.

1. Снимите ваш фильм.

   Возьмите пару уроков у искушенных любителей видео и снимите ваш видеоклип так, чтобы легче было его сжать и при этом внести меньше искажений: выполняйте плавные и короткие движения камерой и не снимайте сложные картины. Ваше сжатое видео в этом случае будет выглядеть лучше.

2. Запустите программу видеозахвата, поставляемую с вашей видеокамерой. Воспользуйтесь ею для загрузки вашего видео на жесткий диск своего компьютера.

   Обычно этот шаг включает подсоединение камеры к вашему компьютеру с помощью кабеля с интерфейсом FireWire. Хотя USB-кабели недостаточно быстры для поддержки громадных порций необработанных видеоданных, вы можете применять их для передачи видео с менее мощных устройств, например фотоаппаратов или мобильных телефонов, записывающих короткие клипы.

3. Воспользуйтесь программой редактирования видео для отрезания именного того видеофрагмента, который хотите опубликовать.

   Некоторые программы позволяют в этот момент добавить музыку или спецэффекты.

4. Перекодируйте видеофрагмент в сильно сжатый формат. Если все сведения о формате в вашей программе выглядят как китайская грамота, при сохранении вашего клипа поищите вариант, в котором явно указано "web video".

   На этом этапе у вас есть три варианта: вы можете задать видеоформат (стандарт, который ваша программа редактирования видео применит для кодирования вашего видеофайла), размеры окна воспроизведения (скажем, 640×360 для стандартного окна на Web-странице) и качество видео (как в случае файлов JPEG, чем сильнее сжатие, тем больше деталей теряется).

   Перекодирование видео — длительная операция. Даже самый быстрый компьютер затратит в пять раз больше времени, чем длительность исходного клипа. Но в конце процесса вы получите более управляемый видеофайл, готовый для публикации в Web, скажем файл размером 20 Мбайт для полноценного трехминутного клипа.

Связанное и встроенное видео

Как ни удивительно, но вы можете поместить видео на свою Web-страницу, пользуясь той же методикой, которую применяли к цифровому звуку (см. рис. 17.1). Это означает, что вы можете сослаться на видеофайл, и он откроется в новом окне браузера.

```
Click to download or open my home movie
<a href="ouch.mpg">Ouch, That Hurts</a>.
```

Или же можно применить элемент `<embed>` и поместить окно видео внутрь собственной Web-страницы.

```
<embed src="ouch.mpg" autoplay="false" />
```

Если применяете элемент `<embed>`, обязательно убедитесь, что кнопка автовоспроизведения *вы*ключена. Видеоклипы, начинающие воспроизводиться автоматически, вызывают еще большее раздражение, чем автоматически проигрываемые мелодии.

Окно видео отображается там, куда вы поместите элемент `<embed>` (рис. 17.9).

Если все кажется слишком простым, чтобы быть правдой, значит, так оно и есть. Несмотря на то что эта простая тестовая страница большую часть времени работает

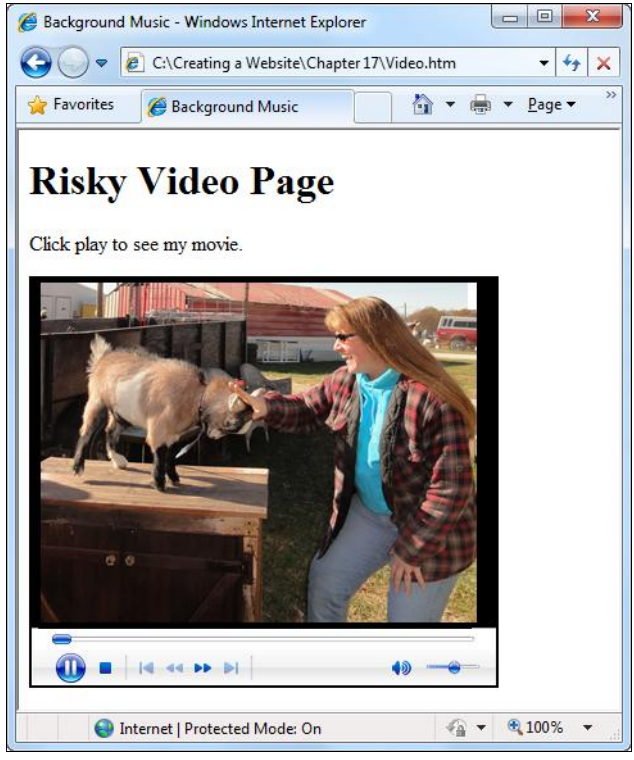


Рис. 17.9. Вставить окно с видео на вашу Web-страницу почти так же легко, как добавить базовое воспроизведение аудио. Если не зададите фиксированный размер, окно автоматически примет размеры вашего видео

корректно, она не вполне надежна. Вашим посетителям, возможно, придется загрузить весь фильм, прежде чем они смогут просмотреть его, все зависит от формата кодирования вашего фильма и настроек браузера. А если у их браузеров нет нужного плагина, или он некорректно установлен, ваше видео может вообще не воспроизводиться.

В следующих двух разделах вы рассмотрите два решения, требующие изрядной мощной поддержки Flash и HTML5. Затем вы познакомитесь с почти безупречной альтернативой с использованием YouTube.

Flash-видео

В настоящее время самое распространенное решение проблем подключения видео — технология Flash. В действительности подавляющее большинство видеоклипов, разбросанных в Web, применяют Flash. Это означает две вещи: видео, которое кодируется в формате FLV (Flash video), и видеоплейер — маленькое Flash-приложение, которое разные Web-сайты адаптируют каждый по-своему.

Один из способов получения надежного видео в Web — применение Flash на ваших Web-страницах. К сожалению, Flash — не HTML и применение этой технологии требует полностью нового набора инструментов (и абсолютно нового обучающего процесса). Скорее всего вы пойдете этим путем, если захотите достичь чего-то еще. Например, захотите создать отличный настраиваемый видеоплейер или использовать другие свойства Flash, такие как способность создавать напоминающие мультфильмы анимационные сюжеты, игры-аттракционы или другие типы приложений с ослепительной графикой. Для начала подберите посвященную Flash книгу, например, "Flash CS5: The Missing Manual".

Второй вариант — применить готовый Flash-видеоплейер, подобный Flash-плейерам MP3, которые использовались вами раньше. Два таких видеоплейера — Flowplayer (**http://flowplayer.org**) и Premiumbeat (**www.premiumbeat.com/flash_video_players**). У обоих стандартные версии бесплатные. К сожалению, Flash-видеоплейеры умеют воспроизводить видео только в FLV-формате, поэтому вам потребуется средство преобразования видео. Вы сможете найти бесплатные варианты на сайтах свободно распространяемых программ, таких как **www.download.com**; просто поищите "FLV encoder" (FLV-преобразователь).

Если вам кажется, что оба эти варианта требуют слишком много хлопот, вы — отличный претендент на сервис видеохостинга, например, такого чрезвычайно популярного, как YouTube. Как им пользоваться, вы узнаете в *разд. "Выгрузка вашего видео на YouTube" данной главы*.

СЕКРЕТЫ РЕМЕСЛА

Видео средствами Microsoft Silverlight

Чтобы создать видео для Web, можно воспользоваться еще одним решением на основе плагина — Microsoft Silverlight, смертельного врага Flash.

В великих войнах плагинов Silverlight — аутсайдер, т. к. его первая версия появлялась только в 2007 г., намного позже того времени, когда Flash колонизировала Web. По этой причине корпорация Microsoft проявляет неестественную щедрость в отношении

средств Silverlight. Вы можете загрузить со страницы **http://tinyurl.com/** их бесплатный пакет Expression Encoder, который не только преобразует ваше видео в сжатый формат, ориентированный на Web, но и создаст отличную управляемую Silverlight страницу видеоплейера. (Если вы амбициозны, можете даже настроить видеоплейер, чтобы получить уникальный внешний вид для своего Web-сайта.)

Единственная загвоздка Silverlight — скорость принятия. Несмотря на то что за последние несколько лет это программное обеспечение существенно распространилось, оно все еще установлено на примерно 60% компьютеров, подключенных к Web, в то время как Flash присутствует на 99% из них. Да, Web-искатели могут быстро и бесплатно загрузить плагин Silverlight, но если они до сих пор не сделали этого, то вряд ли будут хлопотать, а значит, не смогут увидеть ваш видеоклип.

Видео в HTML5

Web стоит на пороге больших перемен. HTML5 не за горами, а этот стандарт вводит более простой способ встраивания видео и аудио в ваши страницы. Вместо хлопот с элементом `<embed>` и плагинами браузеров вы сможете использовать по-настоящему поддерживаемый элемент `<video>`, который выглядит следующим образом.

```
<video src="ouch.mpg" controls></video>
```

В этом примере важно слово `controls`, оно заставляет браузер вставить элементы управления воспроизведением, позволяющие посетителям просматривать видео. Если его пропустить, придется написать код на JavaScript, делающий то же самое.

Поддержки HTML5 пока еще нет, по крайней мере повсеместной. Пользователи Internet Explorer не получат некоторые элементы HTML5 до версии IE 9, которая во время написания книги еще не была выпущена. Для разрешения этой ситуации вам придется вставить (резервное содержимое) внутрь элемента `<video>`. В этом случае, если браузер не понимает элемента `<video>`, он отобразит замещающий контент, например, ссылку на загрузку видео.

```
<video src="ouch.mpg" controls>
  Click here to download the <a href="ouch.mpg">ouch.mpg</a> video.
</video>
```

Конечно, ссылка не так хороша, как настоящее окно с видео. Для компенсации можно поместить внутрь элемента `<video>` элемент `<embed>` для Flash (при условии, что у вас есть Flash-видеоплейер и еще одна копия вашего видео в FLV-формате). Или вы можете вложить внутрь окно YouTube, которое внесет путаницу, но даст вашим посетителям возможность увидеть ваш видеоклип.

Если вам кажется, что элемент `<video>` не стоит создаваемых им проблем, на самом деле все обстоит еще хуже. Во время написания книги создатели самых популярных браузеров все еще не могли договориться о том, какие видеоформаты поддерживать. Поэтому, если хотите применять HTML5, не отказываясь ни от одного современного браузера, придется создавать несколько версий вашего видеофайла и писать скандально неряшливую разметку, использующую приемы, описанные на странице **http://camendesign.com/code/video_for_everybody**. (Но мы предупрежда-

ем, что этот Web-сайт и сам использует теги HTML5, и владельцы Internet Explorer не смогут прочесть его в любой версии, кроме IE 9.)

При таких сложностях захочет ли кто-нибудь вообще применять элемент `<video>` из HTML5? Одна из веских причин — Apple, компания, занимающаяся высокими технологиями и наводнившая мир iPhones и iPads. Apple упорно отказывается разрешить Flash выполняться на своих устройствах отчасти по техническим, а отчасти по политическим причинам. Но Apple согласилась на поддержку HTML5 в своем браузере Safari. Другими словами, если хотите, чтобы ваши видео можно было смотреть на сегодняшнем поколении мобильных устройств Apple, вам нужен стандарт HTML5, а не Flash.

Если же вообще хотите избежать этих проблем, YouTube — гораздо лучшее решение. Его страницы достаточно сообразительны, чтобы определить, нуждается браузер в HTML5 или во Flash, и отправить подходящее содержимое. И в дополнение сайт предоставляет всем вашим видеофайлам бесплатное Web-пространство и полосу пропускания.

Выгрузка вашего видео на YouTube

До появления YouTube видеоклипы не пользовались успехом в Web. Они были неудобны во всех отношениях: медленно загружались, воспроизведение часто было дрожащим и прерывистым. Теперь ситуация изменилась. Web-подключения стали быстрее, и плагины браузера, поддерживающие воспроизведение видео (такие как Flash и Silverlight), стали более распространенными. Обычные пользователи владеют способными снять короткий клип видеоустройствами разного сорта, начиная с настоящих видеокамер и заканчивая цифровыми фотоаппаратами, мобильными телефонами и Web-камерами. Популярные клипы облетают мир со скоростью ракеты, превращаясь в интернет-сенсацию за считанные часы. Члены семьи, авантюристы и как бы политические комментаторы регулярно используют видео для того, чтобы поддерживать связь, продемонстрировать свое мастерство и распускать сплетни.

YouTube (**www.youtube.com**) — на переднем крае в этой революции. Несмотря на то что этот сайт относительный новичок в Web-пространстве (YouTube был создан в 2005 г.), он в настоящее время занимает третье место по популярности среди сайтов (пропуская вперед только Google и Facebook). Содержимое YouTube невероятно разнообразно. Вы с помощью быстрого поиска можете натолкнуться на любительский и профессиональный контент, включая забавное домашнее видео, обзоры товаров и рекламные объявления, самодельные музыкальные видеоклипы, видеоролики фильмов и телевизионных шоу и массу творений обычных людей практически на любую тему (это направление названо *видеоблогами* (video blogging)).

Если вы все еще ищете возможность выложить ваше видео в сеть, есть три веские причины для применения именно YouTube.

- Он хорошо воспроизводит. YouTube применяет Flash, чтобы практически все браузеры могли воспроизводить его видеоклипы. Кроме того, его видео поддер-

живает *поэтапную загрузку* (progressive downloading), т. е. вы можете смотреть начало видеоклипа, пока браузер загружает оставшуюся часть, а не ждать завершения загрузки.

- Он расширяет возможности вашего Web-сайта. YouTube — один из самых популярных сайтов в Web. Удачные видеоклипы могут увеличить свою аудиторию от нескольких человек до миллионов страстных любителей клипов. Помещая ваши видеоролики на YouTube, вы увеличиваете вероятность того, что кто-то откроет их и после этого возможно посетит ваш сайт. Например, самые популярные клипмейкеры зарабатывают на своей YouTube-популярности, продавая тематические товары на своих сайтах. Другие применяют AdSense (см. *разд. "Google AdSense" главы 14*), вставляющую специальные рекламные окна, способные ненавязчиво разместиться в нижней части окна видео.
- Он поддерживает HTML5. YouTube одним из первых внедряет средства видео из HTML5. Но сайту хватает сообразительности применять страницы, управляемые HTML5, только если запрашивающий их браузер поддерживает стандарт HTML5 (в противном случае он использует лучше поддерживаемые Flash-видеофайлы). Подобная ловкость гарантирует, что владельцы устройств, не поддерживающих Flash (например, iPad), все равно увидят свои видеоклипы.

У YouTube один недостаток — вы теряете контроль над качеством вашего видео. YouTube печально известен применением мощного сжатия для уменьшения размеров видеофайлов, после этого некоторые видеоклипы выглядят ужасно. Больше всего страдают клипы, уже закодированные, поскольку YouTube кодирует их во второй раз, уничтожая в процессе мелкие детали.

В следующих разделах вы увидите, как выгрузить ваше первое видео на YouTube, и даже узнаете, как встроить его в окно на одной из ваших Web-страниц.

Регистрация на YouTube

Полный каталог видеоклипов YouTube (6 миллионов клипов во время написания книги) может просматривать любой. Но выгружать собственные видеоролики вы сможете, имея учетную запись YouTube (аккаунт). Вот как ее создать[1].

1. Перейдите на сайт **www.YouTube.com**. Щелкните кнопкой мыши ссылку **Sign Up** (Зарегистрироваться) в правом верхнем углу страницы.
2. Введите данные вашего аккаунта.

   От вас потребуются обычные сведения, включая ваш адрес электронной почты, пароль, местонахождение и дату рождения. В отличие от других Web-сайтов, опознающих вас исключительно по вашему адресу электронной почты, YouTube требуется имя пользователя, представляющее собой строку из букв и цифр, например, JoeTheMovieMaker403. С учетом популярности сайта, возможно, придется сделать несколько попыток, чтобы подобрать незанятое имя. Для выясне-

[1] На момент выхода русского перевода книги интерфейс YouTube изменился, но общие принципы работы с сайтом остались прежними. — *Прим. ред.*

ния доступности выбранного имени щелкните кнопкой мыши ссылку **Check Availability** (Проверить доступность) после того, как введете имя.

Примечание

Если у вас уже есть учетная запись Google, вы можете использовать ее имя пользователя и пароль для регистрации на YouTube. Прокрутите страницу регистрации к левому нижнему углу и щелкните ссылку **Sign in with your Google Account** (Зарегистрируйтесь с помощью своего аккаунта Google). Вам все равно придется выбрать имя пользователя YouTube и ввести адрес вашей электронной почты, но вы сможете входить во все сервисы Google с одной и той же комбинацией пароля и адреса электронной почты.

3. Щелкните мышью кнопку **Create My Account** (Создать мой аккаунт).

   YouTube пришлет подтверждение по вашему адресу электронной почты. После его получения щелкните кнопкой мыши ссылку в письме, чтобы подтвердить ваш аккаунт.

Подготовительный этап

Теперь вы готовы к публикации видео. Но сначала неплохо разобраться в принципах работы YouTube и типах видеофайлов, на которые он рассчитывает. Далее перечислены некоторые существенные характеристики YouTube.

- YouTube принимает видео практически в любом формате. Это означает, что вам не нужно преобразовывать ваш файл перед отправкой. К популярным форматам, поддерживаемым YouTube, относятся MPEG2, MPEG4, MOV, DivX, WMV, WebM, AVI и FLV.
- YouTube автоматически перекодирует видеоклипы, которые вы выгружаете, чтобы Web-посетители могли загружать их без сводящих зубы задержек. По этой причине YouTube рекомендует выгружать оригинальную версию вашего видеоклипа. Другими словами, не перекодируйте ваше видео. Вы только зря потратите время и снизите качество вашего клипа, поскольку YouTube при подготовке вашего клипа к просмотру перекодирует его. Но есть одна загвоздка: в зависимости от длины выгружаемого клипа и применяемого видеоформата ваш исходный файл может быть гигантского размера (легко доходящего до нескольких сотен мегабайтов). Даже при этом YouTube разрешает выгружать файлы размером до 2 Гбайт, видео такого размера не годится ни для кого (начнем с того, что оно будет выгружаться вечно).

Примечание

Если у вас медленное Web-соединение или ваш интернет-провайдер ограничивает объем данных, передаваемых за месяц, вам, быть может, придется уменьшать ваши видеофайлы. В этом случае лучше всего попытаться постепенно уменьшить размер исходного файла; другими словами, применить формат записи с самым низким возможным качеством. Но если файлы все равно остаются слишком большими или вам нужно сохранить высокое качество, чтобы можно было использовать их для других целей (например, записи на DVD-диски), вам, вероятно, придется перекодировать ваше видео с помощью процесса, описанного в *разд. "Подготовка видеофайлов" данной главы*.

- YouTube воспроизводит как стандартное, так и широкоэкранное видео с разными разрешениями. Если вы снимаете видео стандартного размера (с соотношением высоты и ширины 4:3), подойдет разрешение 480×360. Для широкоэкранного видео подходящее разрешение — 640×360. YouTube также поддерживает более высокие разрешения (640×480 и 1280×720). Выгрузите видео с этими разрешениями, и YouTube создаст копии разного качества, так что посетители смогут выбрать версию для просмотра, основываясь на скорости своего соединения. Важно не преобразовывать широкоэкранное видео в стандартное или наоборот, т. к. вы получите в результате видео маленького размера, окруженное черными полосами, компенсирующими отличающиеся контуры экрана.
- Читайте справку YouTube. Время от времени YouTube меняет свои рекомендации или вводит новые поддерживаемые форматы. Для получения свежей информации перед выгрузкой видео посетите страницу **http://tinyurl.com/66onkub**.

Выгрузка видео

После того как видеофайл подготовлен, пора помещать его в Интернет. Процесс до приятного прост.

1. Вернитесь на YouTube и зарегистрируйтесь.

   Вы увидите, что YouTube предлагает массу средств, которые помогут вам исследовать до конца понравившиеся вам видео. Он хранит список просмотренных вами видеоклипов, рекомендует похожие видео и позволяет подписаться на конкретные видеогруппы. Прямо сейчас не обращайте внимания на эти средства и сосредоточьтесь на вставке нашего собственного творения в эту структуру.

2. Щелкните кнопкой мыши ссылку **Upload** (Добавить видео) в верхней части страницы.

   Вы увидите раскрывающееся окно выгрузки и кнопку **Browse** (Обзор).

3. Перетащите мышью файл в окно выгрузки или щелкните мышью кнопку **Upload video** (Добавить видео), чтобы найти файл на своем жестком диске.

   В любом случае вы можете выгрузить одновременно несколько видеофайлов.

   После того как вы выбрали свои клипы, YouTube начинает выгружать их (рис. 17.10). Сейчас можно потратить время на ввод сопутствующей информации, т. к. вы инициировали длительный процесс выгрузки.

4. Введите подробные характеристики вашего видео.

   Вам нужно ввести название и описание, которое YouTube отображает на вашей видеостранице и в своих результатах поиска. Вы также должны задать категорию вашего видео и вставить один или несколько тегов. Когда другие люди ищут в YouTube видео с помощью ключевых слов, совпадающих с вашими тегами, возрастает вероятность того, что ваше видео окажется в результатах поиска.

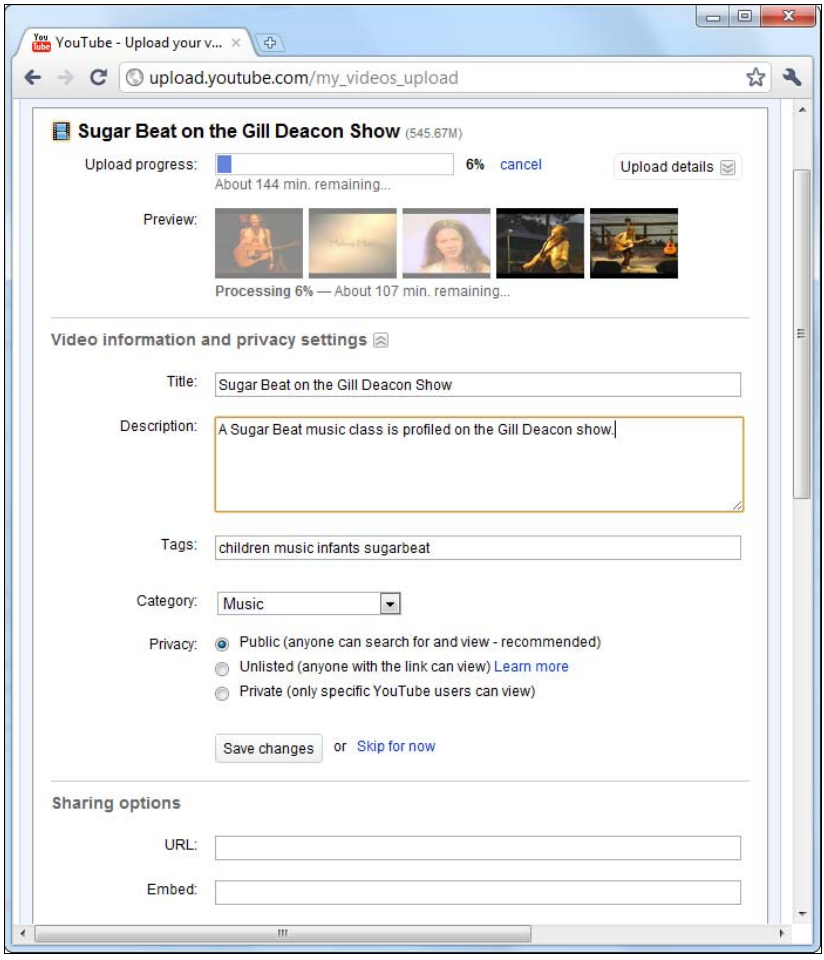


Рис. 17.10. Когда YouTube выгружает ваш файл, он выводит примерную оценку времени, необходимого для полной выгрузки. Когда подходящая порция будет выгружена, он начнет обрабатывать файл. Под полосой индикатора выполнения выгрузки появятся уменьшенные изображения для предварительного просмотра

Примечание

Не беспокойтесь, если введенная вами информация не совсем точна. Ее можно изменить в любое время. Найдите свое видео в разделе **My Videos** (Мои видео), выделите его и щелкните мышью ссылку **Edit** (Изменить видео).

5. Задайте настройку **Privacy** (Конфиденциальность).

   Общедоступное видео включается в результаты поиска YouTube и любой может его просматривать.

   Любой может просматривать и не включенное в список видео, но при этом нужно знать правильный URL для поиска, т. к. такое видео не включается в результаты стандартного поиска.

Если вы пометите ваш клип как **Private** (Частное), только пользователи YouTube, которых вы указали явно, смогут увидеть видео и только после того, как войдут в YouTube.

6. Ждите.

   YouTube утверждает, что для выгрузки каждого мегабайта видеоклипа обычно требуется от 1 до 5 минут при высокоскоростном подключении, поэтому сейчас самое время для второй чашечки кофе.

   Во время ожидания можно перейти к следующему окну выгрузки, выбрать еще один видеофайл и начать и его выгрузку. Если же вы закончили выгрузку, щелкните мышью кнопку **Cancel** (Отмена), чтобы завершить добавление видео.

   Когда выгрузка видео завершится и YouTube преобразует ваше видео, вы увидите в верхней части страницы выгрузки сообщение (рис. 17.11).



Рис. 17.11. В конце каждой выгрузки YouTube предоставляет две порции важной информации: ссылку, которая позволит просмотреть ваше видео, и фрагмент HTML-разметки, который можно вставить на любую Web-страницу для встраивания вашего видео

7. При желании отредактируйте ваше видео, настроив несколько дополнительных параметров.

   Для этого щелкните кнопкой мыши ваше имя пользователя (в правом верхнем углу страницы) и выберите ссылку **My Videos** (Мои видео). Найдите видео, которое только что выгрузили, и щелкните кнопку **Edit** (Изменить), расположенную рядом с ним.

   Вы можете откорректировать информацию, предоставленную вами при первой выгрузке видео, и задать следующие дополнительные параметры.

   - Выбрать **Video Thumbnail** (Значок видео), единственный кадр из вашего видео, который будет появляться, когда YouTube отобразит ваше видео в результатах поиска.
   - Использовать параметры раздела **Date and Map** (Дата и карта), чтобы определить, когда и где вы записали свое видео.
   - Применить параметры **Comments** (Комментарии), **Comment Voting** (Оценка комментария), **Video Responses** (Видеоответы) и **Ratings** (Оценки) для управления откликами людей на ваше видео. Например, можно запретить комментарии или разрешить только одобренные вами комментарии. Вы также можете определить, могут ли люди оценивать комментарии других людей, публиковать видеоответы (видео, связанное с вашим) и оценивать ваши видео (с помощью кнопок с большим пальцем, поднятым вверх или опущенным вниз). Обычно YouTube разрешает все эти режимы, создавая на сайте атмосферу шумного содружества. В некоторых случаях имеет смысл ограничить комментарии и оценки (например, если у вас деликатное видео, и вы боитесь оскорбительных комментариев). Но подавляющее большинство видеоклипов на YouTube разрешают комментарии (и активно комментируются). Это видеоклипы, которые не могут быть проигнорированы.
   - Используйте раздел **Embedding** (Встраивание) для управления возможностью встраивания вашего видео на чужие Web-страницы. Но имейте в виду, что если вы его запретите, то не только помешаете другим людям демонстрировать ваше видео, но и себя лишите возможности встроить собственное видео на свой Web-сайт.

   Совет

   На странице **Uploaded Videos** (Выгруженные видео) можно решать и другие задачи управления. Например, выберите видеоклип и нажмите кнопку **Delete** (Удалить) для его удаления, кнопку **Edit** (Изменить) для изменения видеоинформации и параметров, заданных вами при выгрузке клипа, и кнопку **Insight** (Статистика) для получения кое-какой увлекательной статистики о людях, его посмотревших.

8. Когда закончите, щелкните мышью **Save Changes** (Сохранить изменения).

   Теперь ваше видео полностью настроено и начинает жить в огромном Web-мире.

Просмотр видео

После подготовки видео его можно просматривать несколькими способами.

- Можно искать на YouTube.
- Можно просмотреть все видео в вашем аккаунте. Войдите на YouTube, щелкните кнопкой мыши ваше имя пользователя (на странице вверху справа) и выберите раздел **My Videos** (Мои видео).
- Можно просмотреть его в окне YouTube на одной из ваших Web-страниц. Для этого скопируйте на вашу страницу разметку, приведенную на рис. 17.11. (Если вы не скопировали эту разметку, когда выгружали свое видео, перейдите на страницу YouTube с вашим видеоклипом и щелкните мышью кнопку **Embed** (Встроить), чтобы получить ее снова.)
- Можно поместить на вашу Web-страницу ссылку, которая приведет к вашей видеостранице на YouTube. Просто перейдите на страницу и скопируйте URL из адресной строки.

Из всех этих вариантов самый интересный — встраивание вашего видео в окно YouTube на собственной Web-странице (рис. 17.12). Он позволяет сочетать внешнее представление самостоятельно хранимого видеоклипа с высокой производительностью YouTube и основательной поддержкой браузеров. Но самое замечательное — легкость встраивания видеоклипов, заключающегося в простом копировании фрагмента HTML-разметки на вашу страницу. Вы просто помещаете эту разметку в то место, где хотите вывести видеоокно. Часто ее заключают в элемент `<div>` и применяют стилевые правила для позиционирования элемента (как описано в *главе 9*).

Далее приведена полностью разметка, создающая страницу, показанную на рис. 17.12.

```
<!DOCTYPE html>

<html>
<head>
  <title>A Taste of YouTube</title>
</head>

<body>
  <h1>A Taste of YouTube</h1>
  <p>The following video window is brought to you by the fine people
    at YouTube.</p>

  <object width="425" height="344">
    <param name="movie"
      value="http://www.youtube.com/v/qx0QIUkRAGQ&hl=en&fs=1"></param>
    <embed src="http://www.youtube.com/v/qx0QIUkRAGQ&hl=en&fs=1"
      type="application/x-shockwave-flash" allowfullscreen="true"
      width="425" height="344">
```

```
    </embed>
  </object>

  <p>Click it to start playing.</p>
</body>
</html>
```



Рис. 17.12. Встраивание позволяет просматривать ваше видео на странице собственного производства

Внимательный читатель заметит, что это видеоокно состоит из элемента `<embed>`, заключенного в элемент `<object>`. Эта запутанная разметка отлично подходит для обеспечения совместимости браузеров, потому что браузеры, поддерживающие элемент `<object>`, проигнорируют внутреннее содержимое, а браузеры, не знакомые с элементом `<object>`, воспользуются элементом `<embed>`.

Совет

Для изменения цвета рамки вокруг вашего видеоокна заставьте его запускаться автоматически, задайте поддержку полноэкранного режима или настройте одну из нескольких других деталей, необходимых для регулировки параметров внутри разметки. Более подробную информацию см. на странице **http://tinyurl.com/2kxnv3**.

1

ПРИЛОЖЕНИЕ 1

Краткий справочник по HTML

HTML — язык Web-пространства. Его можно применять для создания любой Web-страницы, рекламирующей продажу местной выпечки или управляемой компанией из списка "Fortune 500". В *главе 2* было приведено подробное описание HTML и с этого момента вы постоянно пополняли ваш арсенал HTML-элементов.

Данное приложение предлагает краткую справку по всем HTML-элементам, с которыми вы встречались в этой книге (и даже немного больше). В каждом разделе приводится краткое описание назначения элемента и во многих разделах даются ссылки на более подробные примеры в книге. Вам также предлагается обзор самых полезных новых элементов в HTML5 (тех, которые можно заставить работать в Internet Explorer 8 и более ранних версиях) и краткая памятка о символьных HTML-последовательностях, позволяющих отображать специальные символы на Web-странице.

HTML-элементы

Как вы уже знаете, в основе HTML-стандарта лежат элементы — специализированный программный код в угловых скобках, сообщающий браузеру о том, как форматировать текст, когда вставлять изображения и как связывать вместе разные документы. На протяжении всей книги вы рассмотрели практически все важные HTML-элементы, применяемые в настоящее время, за исключением новейших элементов из HTML5, способных нарушить работу вашего браузера.

В этом справочнике элементы перечислены в алфавитном порядке. В начале каждого раздела приводятся две важнейшие характеристики элемента.

- *Блочный или строчный.* Блочный элемент можно помещать непосредственно в раздел Web-страницы `<body>`. Со строчным элементом так поступать нельзя; его необходимо поместить *внутрь* блочного элемента. К некоторым элементам применяются более строгие правила и их можно вставлять в элементы только определенных типов. Например, элемент `<area>`, определяющий активную область на карте изображения, ограничен элементом `<map>`, описывающим карту изображения (см. *разд. "Изображения-ссылки и карты изображений" главы 8*).

Примечание

В официальном словаре HTML5 блочные элементы называются *потоковыми элементами*, а строчные элементы — *фразовыми элементами*.

- *Контейнер или пустой.* Элемент-контейнер может содержать текст или другие элементы. Пустой элемент должен оставаться пустым, поэтому его обычно записывают как один тег с синтаксисом пустого элемента (см. *разд. "Представление об элементах" главы 2*).

Новые элементы HTML5 вы найдете в *разд. "Семантические элементы HTML5" данного приложения*.

<a> (элемент-якорь)

Строчный элемент, элемент-контейнер.

У элемента-якоря `<a>` две роли. Самая распространенная — создание ссылки, которая при щелчке ее кнопкой мыши переносит вас с одной страницы на другую. Для вставки этого типа ссылки вы задаете URL назначения, используя атрибут `href`, и помещаете текст ссылки между открывающим и закрывающим тегами.

```
<a href="LinkedPage.htm">Click Me</a>
```

При задании атрибута `href` можно использовать *относительный* URL, указывающий на страницу на вашем собственном Web-сайте, или *абсолютный* URL, начинающийся с http:// и указывающий на любую страницу в Web. Отличие относительных и абсолютных ссылок и особенности их применения см. в *разд. "Внутренние и внешние ссылки" главы 8*.

Создавать графические ссылки так же легко, как текстовые. Хитрость заключается во вставке элемента `<img>` внутрь элемента `<a>` следующим образом.

```
<a href="LinkedPage.htm"><img src="MyPic.gif" alt="My Pic" /></a>
```

И наконец, можно создать ссылку, которая вместо переноса посетителя на другую страницу открывает бланк письма электронной почты с заполненной адресной информацией. Делается это с помощью ссылки `mailto`, как показано далее.

```
<a href="mailto:me@myplace.com">Email Me</a>
```

Дополнительную информацию о ссылке `mailto` см. в *разд. "Почтовые ссылки" главы 8*.

Вы можете применять в элементе-якоре атрибут `target`, заставляющий браузер открыть страницу назначения в особом фрейме или новом окне браузера следующим образом.

```
<a href="LinkedPage.htm" target="_blank">Click Me</a>
```

Элемент-якорь также работает с *закладками*, позволяющими переносить гостей Web-сайта к конкретному месту на Web-странице. Для создания закладки необходимо задать имя элемента в его атрибуте `id` следующим образом.

```
<h2 id="Canaries">Pet Canaries</h2>
```

После того как закладка создана, можно писать URL, указывающий на эту закладку. Хитрость в том, что информацию о закладке нужно добавить в конец URL. Для этого вставьте знак решетки (#) и следом за ним имя закладки, как показано далее.

```
Learn about recent developments in <a href="sales.htm#Canaries">canary
sales</a>.
```

Дополнительную информацию о закладках и обычных ссылках можно узнать в *главе 8*.

<address>

Блочный элемент, элемент-контейнер.

Web-разработчики применяют элемент <address> лишь иногда, он предоставляет посетителям возможность связаться с авторами Web-страницы. Эта контактная информация может состоять из адреса электронной почты или Web-ссылки (два самых распространенных варианта) или из почтового адреса. Далее приведен пример.

```
Our website is managed by:
<address>
<a href="mailto:jsolo@mysite.com">John Solo</a>,
<a href="mailto:lcheng@mysite.com ">Lisa Cheng<a>, and
<a href="mailto:rpavane@mysite.com ">Ryan Pavane</a>.
</address>
```

Но если на вашей Web-странице есть другая адресная информация, скажем физический адрес вашего магазина, не используйте для нее элемент <address>. По традиции он зарезервирован для человека или нескольких человек, осуществляющих сопровождение Web-страницы.

Большинство браузеров выделяют адреса курсивом, как если бы вы применили элемент <i>. Единственный смысл в использовании элемента <address> — возможность извлекать полезную адресную информацию автоматически выполняющимся программам, сканирующим Web-страницы.

<area> (карта изображения)

Допускается только в <map>, пустой элемент.

Элемент <area> определяет область, реагирующую на щелчок кнопкой мыши (называемую *активной областью*) на карте изображения (которую вы генерируете с помощью элемента <map>; см. *разд. "<map> (карта изображения)" данного приложения*). При определении активной области необходимо указать конечный URL (с помощью атрибута href), ее контур (с помощью атрибута shape) и координаты этого контура (с помощью атрибута coords). Контуром может быть окружность, прямоугольник или многоугольник.

Для окружности задают координаты в следующем порядке: точка центра (координата x), точка центра (координата y), радиус. Для любого другого контура задаются координаты углов по порядку, как последовательность координат x-y, например,

так: x1, y1, x2, y2 и т. д. Далее приведен пример, создающий активную область в виде прямоугольника.

```
<area href="page1.htm" shape="square" coords="5,5,95,195" alt="A clickable
square" />
```

Прямоугольник невидимый. Если вы щелкнете кнопкой мыши где-нибудь внутри этого прямоугольника, вы перейдете на page1.htm. Дополнительную информацию см. далее в разделе, посвященном элементу <map>. Законченный пример с картой изображения см. в *разд. "Изображения-ссылки и карты изображений" главы 8*.

<b> (жирный шрифт)

Строчный элемент, элемент-контейнер.

Элемент <b> отображает текст жирным шрифтом. Официальные правила полагают, что вы применяете элемент <b> для стилистического выделения текста, который должен быть представлен жирным шрифтом, но не более важен, чем остальной контент. Он может включать ключевые слова, названия товаров и что-либо еще, что следует выводить жирным шрифтом при печати.

```
Make sure you buy the <b>Super-Fraginator</b> today!
```

Если вы хотите отформатировать текст жирным шрифтом и придать ему дополнительную важность, **рекомендуемый вариант** — элемент (хотя визуальный эффект один и тот же).

<base> (базовый URL)

Разрешен только в элементе <head>, пустой элемент.

Элемент <base> определяет базовый или основной URL документа, т. е. Web-адрес, применяемый для расшифровки всех относительных путей. Элемент <base> следует помещать в раздел страницы <head> и можно применить два атрибута — href (поставляющий базовый URL) и target (задающий конечный фрейм для ссылок).

Например, если у вас есть ссылка, указывающая на файл с именем MySuperSunday.htm, и базовый URL **http://www.SundaysForever.com/Current/**, браузер интерпретирует ссылку как **http://www.SundaysForever.com/Current/MySuperSunday.htm**. Web-профессионалы редко применяют базовый URL подобным образом, т. к. для браузера почти всегда удобнее использовать текущую страницу как опорную точку для всех относительных URL. Другими словами, если вы ищете **http://www.SundaysForever.com/Current/Intro.htm**, браузер уже знает, что базовый URL — **http://www.SundaysForever.com/Current/**. Дополнительную информацию об отличии абсолютных и относительных ссылок см. на стр. 237.

<blockquote> (блочная цитата)

Блочный элемент, элемент-контейнер.

Элемент <blockquote> обозначает длинную цитату (которая отображается отдельно, вне других абзацев) как блочный элемент.

```
<blockquote><p>It was the best of times, it was the worst of times.</p>
</blockquote>
```

Обычно браузеры выделяют элемент <blockquote> отступами слева и справа. Но не следует применять <blockquote> исключительно для форматирования. Используйте этот элемент по прямому назначению для выделения цитируемого из книги фрагмента текста. Для текста <blockquote>, как и для любого другого элемента, можно задать стилевое правило, изменяющее стандартное форматирование браузера.

Если хотите вставить короткую цитату в блочный элемент (такой как абзац), используйте вместо элемента <blockquote> элемент <q>.

<*body*> (тело документа)

Разрешен только в элементе <html>, элемент-контейнер.

Элемент <body> — базовая часть структуры любого HTML-документа. Вы вставляете его сразу за окончанием раздела <head>, и он содержит весь контент вашей Web-страницы, включая ее текст, URL изображений, таблицы и ссылки.

<*br*> (перенос строки)

Строчный элемент, пустой элемент.

Перенос строки (<br>) — строчный элемент, который переносит следующий за ним текст на новую строку без дополнительной пробельной зоны. Например, можно использовать элемент
 для разделения адресной информации в абзаце.

```
<p>Johny The Fever<br />
200 Easy Street<br />
Akimbo, Madagascar</p>
```

<*button*> (кнопка)

Строчный элемент, элемент-контейнер.

Элемент <button> позволяет создать в форме реагирующую на щелчок мыши кнопку с любым контентом в ней (например, можно поместить фразу или изображение между открывающим и закрывающим тегами элемента <button>). Как для любого другого элемента управления формы нужно задать уникальное имя и значение, которое форма отправит, когда посетитель щелкнет кнопку мышью. Контент кнопки помещается между открывающим и закрывающим тегами.

```
<button name="submit" value="order" type="button">Place Order</button>
```

Можно создать три типа кнопок, в зависимости от значения, заданного в атрибуте type. Значение button задает стандартную кнопку без встроенных хитростей (вставьте код на JavaScript, чтобы заставить ее что-то делать). Значение reset очищает содержимое элементов ввода в форме, а значение submit отправляет данные

формы обратно на Web-сервер, что полезно, если вы создаете приложение, которое выполняется на сервере.

Для создания кнопок элемент <button> подходит больше, чем элемент <input>, потому что он помещает на поверхность кнопки любой нужный вам контент, включая изображения.

```
<button name="submit" value="order" type="button">
  <img src="Order.gif" alt="Place Order" />
</button>
```

<caption> (заголовок таблицы)

Разрешен только в элементе <table>, элемент-контейнер.

Элемент <caption> определяет текст заголовка таблицы. Его следует сделать первым в элементе <table>.

```
<table>
  <caption>Least Popular Vacation Destinations</caption>
  ...
</table>
```

HTML не применяет никакого форматирования к заголовку; он просто помещает заголовок над таблицей как обычный текст (и делит его на строки в соответствии с шириной таблицы). С помощью правил таблицы стилей можно задать любое форматирование заголовка.

<cite> (опубликованный источник)

Строчный элемент, элемент-контейнер.

Элемент <cite> задает источник, который может быть ссылкой на книгу, печатную статью или другую публикацию.

```
<p>Charles Dickens wrote <cite>A Tale of Two Cities</cite>.</p>
```

Обычно браузеры отображают элемент <cite> курсивом. Но применять элемент <cite> только для форматирования не следует. Используйте его по прямому назначению (например, когда ссылаетесь на опубликованную работу, которую цитируете) и вставляйте правила таблицы стилей для применения определенного форматирования.

<code>

Строчный элемент, элемент-контейнер

Применяйте элемент <code> для вставки в него небольших фрагментов программного кода (например, на Web-странице, представляющей учебное пособие по программированию). Браузеры отображают такой код моноширинным шрифтом.

<dd> (словарное определение)

Разрешен только в элементе <dl>, элемент-контейнер.

Элемент <dd> задает определение в словарном списке. Дополнительную информацию см. в описании элемента <dl> или в *разд. "Списки определений" главы 5*.

<del> (удаленный текст)

Блочный элемент или строчный элемент, элемент-контейнер.

Web-разработчики редко применяют элемент <del>; он обозначает текст, который присутствовал, но теперь удален. Браузеры, поддерживающие этот элемент, отображают перечеркнутый текст для представления удаленного материала. Еще один элемент, иногда применяемый Web-профессионалами для обозначения переработанной версии, — элемент <ins> (см. соответствующий раздел далее).

<dfn> (определяемый термин)

Строчный элемент, элемент-контейнер.

Разработчики сайтов редко используют элемент <dfn>; он обозначает определяемый экземпляр термина. Например, когда вы первый раз встречаете новый термин в этой книге, такой как *froopy (крутой)*, он выделяется курсивом. В данном случае он является определяемым экземпляром и за ним, как правило, следует определение. Браузеры отображают элемент <dfn> курсивом.

(типовой блочный контейнер)

Блочный элемент, элемент-контейнер.

Элемент группирует вместе один или несколько блочных элементов. Например, вы можете собрать вместе несколько абзацев, абзац и заголовок и т. д. Далее приведен пример.

```
<div>
  <p>...</p>
  <p>...</p>
</div>
```

Сам по себе элемент не делает ничего. Однако он очень полезен для применения форматирования таблицы стилей. В приведенном примере вы можете задать форматирование элемента , и браузер распространит его на два вложенных абзаца (при условии, что стилевые свойства, использованные вами, поддерживают наследование, как описано в *разд. "Наследование" главы 6*).

Дополнительную информацию об использовании элемента для применения стилевых правил см. в *разд. "Сокращение работы с помощью элемента " главы 6*. Вы также можете ссылаться на элемент (см. соответствующий раздел далее), который задает форматирование внутри блочного элемента. Если же вы хотите применить более информативные контейнеры, подумайте о замене вашего

 одним из семантических элементов HTML5, обсуждение которых начинается в *разд. "Семантические элементы HTML5" данного приложения*.

<dl> (словарный список)

Блочный элемент, элемент-контейнер.

Элемент <dl> описывает список определений (также называемый словарным списком), представляющий собой последовательность терминов, за каждым из которых следует определение в текстовом блоке с отступами, появляющемся сразу под термином. Теоретически вы можете поместить любое содержимое в словарный список, но рекомендуется следовать его прямому назначению и включать в список термины и их определения. Далее приведен пример.

```
<dl>
  <dt>tasseomancy</dt>
  <dd>Divination by reading tea leaves.</dd>
  <dt>tyromancy</dt>
  <dd>Divination by studying how cheese curds form during cheese
     making.</dd>
</dl>
```

<dt> (словарный термин)

Разрешен только в элементе <dl>, элемент-контейнер.

Элемент <dt> обозначает термин в словарном списке. Дополнительную информацию см. в описании элемента <dl>, приведенном ранее, или в *разд. "Списки определений" главы 5*.

(акцент)

Строчный элемент, элемент-контейнер.

Элемент оказывает тот же эффект, что и элемент <i> (наклонный шрифт или курсив), но у него несколько иное назначение. Он предназначен для текста, который выделяется интонацией при чтении вслух, например:

```
Make sure you <em>don't</em> use the wrong element for your italics.
```

Если сравнить с элементом <i>, последний предназначен для написания курсивом текста, не имеющего интонационного выделения.

С помощью таблицы стилей можно изменить форматирование элемента и выделять его содержимое без использования курсива (например, окрасив текст в красный цвет).

<embed> (встроенный объект)

Строчный элемент, элемент-контейнер.

Элемент <embed> встраивает специальные объекты на вашу страницу, например аудио, видео и апплеты (миниатюрные программы, работающие внутри Web-

страницы). Например, вы можете применить элемент <embed> для вставки на вашу Web-страницу плейера с фоновой музыкой или видео с YouTube, как описано в *главе 17*.

Элемент <embed> тесно связан с элементом <object>, хотя и отличается синтаксисом. Поскольку некоторые очень старые браузеры поддерживают <embed> и не поддерживают <object>, порой эти элементы применяются вместе, и элемент <embed> предоставляет замещающий контент, если элемент <object> не работает, как показано в *разд. "Просмотр видео" главы 17*.

<form> (интерактивная форма)

Блочный элемент, элемент-контейнер.

Элемент <form> создает интерактивную форму, в которую вы помещаете графические виджеты, такие как текстовые поля, флажки, списки выбора и т. д. (представляемые элементами <input>, <textarea>, <button> и <select> соответственно). Помещая эти виджеты в элемент <form>, можно создавать страницы, собирающие информацию от посетителей и отправляющие ее Web-приложению. Web-приложения не рассматриваются в этой книге, но вы можете увидеть, как применять элемент <form> с JavaScript в *разд. "Базовая форма" главы 15*, а также познакомиться с бесплатными сервисами создания и обработки форм в *разд. "Электронные информационные рассылки" главы 12*.

<h1>, <h2>, <h3>, <h4>, <h5>, <h6> (заголовки)

Блочный элемент, элемент-контейнер.

Заголовки — это названия разделов. Они используют жирное начертание и шрифт разного размера. Размер шрифта зависит от уровня заголовка. Шесть уровней заголовков начинаются с <h1> (самый крупный) и снижаются до <h6>. У заголовков 5-го и 6-го уровня размер шрифта меньше размера обычного текста. Далее приведен пример заголовка <h1>.

```
<h1>Important Information</h1>
```

Пользуясь заголовками, всегда добивайтесь, чтобы ваша страница соответствовала логической структуре, в которой все начинается с <h1> и постепенно переходит к более низким уровням. Не начинайте с заголовка <h3>, только потому, что его форматирование выглядит привлекательнее. Вместо этого применяйте таблицы стилей для изменения форматирования заголовков всех уровней, добиваясь подходящего внешнего вида, и используйте уровни заголовков для описания структуры вашего документа.

<head> (шапка документа)

Разрешен только в элементе <html>, элемент-контейнер.

Раздел <head> располагается перед разделом страницы <body>. В то время как элемент <body> содержит контент Web-страницы, элемент <head> включает в себя дру-

гую информацию, например название Web-страницы (элемент <title>), описательные метаданные (один или несколько элементов <meta>) и стили (элементы <style> или <link>).

<hr> (горизонтальная линия)

Блочный элемент, пустой элемент.

Элемент <hr> задает горизонтальную линию (сплошную), которая применяется для разделения блочных элементов.

```
<p>...</p>
<hr />
<p>...</p>
```

Хотя элемент <hr> всегда работает корректно, HTML стремится к использованию для этой цели границ в правиле таблицы стилей, чтобы получить больше контроля над стилем линии и ее цветом. Далее приведен пример, задающий правило таблицы стилей для сплошной синей линии.

```
.border { border-top: solid medium navy }
```

А вот как его можно применить.

```
<p>...</p>
<div class="border"></div>
<p>...</p>
```

Дополнительную информацию о настройке границ в таблице стилей см. в *разд. "Создание улучшенных границ" главы 6.*

<html> (документ)

Элемент <html> — первый элемент в HTML-документе. Он включает в себя весь документ. Если вы создаете обычную Web-страницу, элемент <html> содержит два основных компонента — элемент <head>, определяющий название страницы, метаданные и связанные таблицы стилей, и элемент <body>, содержащий реальный контент.

<i> (наклонный шрифт)

Строчный элемент, элемент-контейнер.

Элемент <i> отображает текст курсивом или наклонным шрифтом, не выражая ни особого логического ударения, ни интонации. Далее приведен пример.

```
The mattress label says <i>do not remove under penaltyof law</i>
```

Если хотите вывести текст, не только наклонный типографически, но и подразумевающий логическое ударение, больше подойдет элемент . Но форматирование у обоих элементов одно и то же.

<iframe> (внутристрочный фрейм)

Строчный элемент, элемент-контейнер.

Элемент <iframe> создает внутристрочный фрейм, встроенное, прокручиваемое окно, отображающее Web-страницу внутри другой страницы. Вы задаете атрибуты src (страницу, которую браузер должен отобразить во фрейме), name (уникальное имя фрейма) и width и height (размеры фрейма в пикселах). Можно отключить рамку вокруг фрейма, задав атрибут frameborder="0", или прокрутку, добавив атрибут scrolling="no". Далее показано, как используется элемент <iframe>.

```
<iframe src="MyPage.html" width="100" height="250"></iframe>
```

Внутрь элемента <iframe> можно поместить контент, который браузер отобразит, если не поддерживает элемент <iframe>.

```
<iframe src="MyPage.html" width="100" height="250">
  <p>To see more details, check out <a href="MyPage.html">this
    page</a>.</p>
</iframe>
```

<img> (изображение)

Строчный элемент, пустой элемент.

Элемент <img> указывает на изображение, которое вы хотите отобразить на странице. Атрибут src задает изображение с помощью относительной или абсолютной ссылки (см. *разд. "Внутренние и внешние ссылки" главы 8*). Атрибут alt содержит текст, который браузер выводит, если не может отобразить изображение.

```
<img src="OrderButton.gir" alt="Place Order" />
```

Internet Explorer выводит замещающий текст во всплывающем текстовом окне, в то время как другие, знающие о стандартах браузеры (а именно Firefox) нет. В любом случае вы можете предоставить всплывающее текстовое окно в практически любом браузере с помощью атрибута title. Это наилучший способ вставить всплывающий текст к изображению.

Элемент <img> также поддерживает атрибуты height и width, которые можно использовать для явного задания размеров изображения.

```
<img src="photo01.jpg" alt="A photo" width="100" height="150" />
```

В данном примере у изображения ширина 100 пикселов и высота 150 пикселов. Если эти размеры не соответствуют действительным размерам исходного изображения, ваш браузер растянет или сожмет изображение, добиваясь соответствия заданным размерам.

Для изменения размера изображения никогда не применяйте атрибуты height и width. Вместо этого отредактируйте исходное изображение в подходящем графическом редакторе. Вы можете применять атрибуты height и width для того, чтобы сообщить браузеру о размере своего изображения, и он сможет сверстать страницу

перед загрузкой всего изображения, тогда макет страницы сохранится, даже если браузер не найдет файла с изображением.

Дополнительную информацию о поддерживаемых типах файлов, способах размещения изображений на странице и поиске наилучшего материала см. в *главе 7*.

И наконец, вы можете создавать на изображении области, реагирующие на щелчок кнопкой мыши, определив карту изображения и затем связав эту карту с вашим изображением с помощью атрибута `usemap` элемента `<img>` Дополнительную информацию см. далее в разделе, посвященном элементу `<map>`.

<*input*> (элемент ввода)

Разрешен только в элементе `<form>`, пустой элемент.

Элемент ввода — самый распространенный компонент HTML-формы (которая сама представляется элементом `<form>`). Элемент `<input>` может представлять разные экранные виджеты (называемые элементами управления), собирающие информацию от Web-посетителей.

Атрибут `type` определяет вид элемента управления, который вы собираетесь создать. В табл. П1.1 перечислены самые распространенные варианты. Кроме того, каждому элементу управления следует присвоить уникальное имя с помощью атрибута `name`.

Таблица П1.1. *Элементы управления HTML-формами*

Элемент управления	HTML-элемент	Описание
Однострочное текстовое поле	`<input type="text" />`	Отображает текстовое поле, в которое посетитель может вводить любой текст
Текстовое поле пароля	`<input type="password" />`	Отображает текстовое поле, в которое посетитель может ввести любой текст, но браузер не отображает текст. Он выводит звездочку (*) на месте каждой введенной буквы, чтобы скрыть текст от любопытных глаз
Флажок	`<input type="checkbox" />`	Отображает переключатель-флажок, который можно установить или сбросить
Зависимая кнопка	`<input type="radio" />`	Отображает зависимую кнопку (кружок, который можно установить или сбросить). Обычно задается группа кнопок, одна за другой, из них посетитель выбирает только одну
Кнопка **Submit**	`<input type="submit" />`	Отображает стандартную кнопку для нажатия, которая отправляет форму и все ее данные на сервер
Кнопка **Reset**	`<input type="reset" />`	Отображает стандартную кнопку, которая очищает все настройки посетителя и введенный им текст во всех элементах ввода формы

Таблица П1.1 *(окончание)*

Элемент управления	HTML-элемент	Описание
Графическая кнопка	`<input type="image" />`	Отображает кнопку **Submit** отличающуюся тем, что вы представляете ее как изображение. Для задания файла с изображением указывается атрибут `src`
Обычная кнопка	`<input type="button" />`	Отображает стандартную кнопку, не делающую ничего, если вы не связали ее с некоторым кодом на JavaScript (см. *главу 15*)

Далее показан элемент `<input>`, создающий текстовое поле. Когда посетитель отправляет страницу, посылается все, что было введено в поле, вместе с описательным идентификатором `LastName`:

```
<input type="text" name="LastName" />
```

Конечно, формы полезны только тогда, когда у вас есть программный код, обрабатывающий их данные. В *разд. "Базовая форма" главы 15* вы видели, как гости сайта могут взаимодействовать с формами с помощью JavaScript, но гораздо чаще формы применяются вместе с серверными приложениями. Например, сценарий на сервере может получить из формы некоторую информацию от посетителя и сохранить ее в базе данных.

<ins> (вставленный текст)

Блочный или строчный элемент, элемент-контейнер.

Элемент `<ins>` обозначает недавно вставленный текст (например, его можно использовать для демонстрации изменений страницы в ходе самого последнего редактирования). По умолчанию текст внутри элемента `<ins>` подчеркивается. Из-за его узкоспециализированного назначения вы редко встретите страницы, использующие элемент `<ins>`.

Вы можете применять элемент `<ins>` вокруг блочных элементов или внутри них. Элемент `<del>` — еще один элемент редактирования и его можно использовать вместе с элементом `<ins>`.

<li> (пункт списка)

Разрешен только в элементах `<ol>` и `<ul>`, элемент-контейнер.

Элемент `<li>` представляет отдельный пункт в нумерованном или маркированном списках. Дополнительную информацию см. далее в описании элемента `<ol>` для нумерованных списков и в описании элемента `<ul>` для маркированных списков.

<link> (взаимосвязи документа)

Разрешен только в разделе <head>, пустой элемент.

Элемент <link> описывает взаимосвязи текущего документа и другого документа. Например, его можно применять для указания на предыдущую версию текущего документа. Более часто его используют для ссылки на *внешнюю таблицу стилей*, предоставляющую инструкции для форматирования текущей страницы. Вы всегда помещаете элемент <link> в раздел <head> вашей страницы. Далее приведен пример возможного применения.

```
<link rel="stylesheet" href="MyStyles.css" />
```

Пользуясь внешними таблицами стилей, можно описать ваши стили в одном файле и применять их на нескольких страницах. Дополнительную информацию о таблицах стилей и вариантах их применения см. в *главе 6*.

<map> (карта изображения)

Строчный элемент, элемент-контейнер.

Элемент <map> описывает *карту изображения*, изображение с одной или несколькими областями, реагирующими на щелчок кнопкой мыши. Когда вы создаете карту изображения, с помощью атрибута name задается уникальное имя, идентифицирующее карту изображения. Далее внутрь элемента <map> для каждой активной области вставляется элемент <area>, задающий ее координаты и конечный URL. (См. ранее дополнительную информацию об атрибуте coords в описании элемента <area>.) Далее приведен пример карты изображения с тремя активными областями.

```
<map id="Three Squares" name="ThreeSquares">
  <area href="page1.htm" shape="square" coords="5,5,95,195"
   alt="Square #1" />
  <area href="page2.htm" shape="square" coords="105,5,195,195"
   alt="Square #2" />
  <area href="page3.htm" shape="square" coords="205,5,295,195"
   alt="Square #3" />
</map>
```

В заключение для использования вашей карты изображения необходимо связать ее с изображением с помощью атрибута usemap. Этот атрибут совпадает с именем карты, но начинается со знака решетки (#), сообщающего браузеру о том, что карта изображения находится на текущей странице.

```
<img src="image.gif" usemap="#ThreeSquares" alt="Image with Hotspots" />
```

Вы не сможете увидеть активные области карты изображения (если не обведете их на изображении). Однако при наведении указателя мыши на активную область, пиктограмма указателя изменится на кисть руки. Щелчок кнопкой мыши активной области создает тот же эффект, что и щелчок мышью обычной ссылки <a> — вы

немедленно переходите на новый URL. Полноценный пример с картой изображения см. в *разд. "Изображения-ссылки и карты изображений" главы 8*.

<meta> (метаданные)

Разрешен только в разделе <head>, пустой элемент.

Метаэлементы позволяют встраивать в ваши Web-страницы описательную информацию. Ваши посетители никогда не увидят ее, но автоматические программы, например поисковые машины, смогут найти ее во время сканирования вашего сайта. Метаданные вставляются с помощью элементов <meta> в раздел <head> вашей страницы.

Все элементы <meta> содержат атрибут name (обозначающий тип вставляемой вами информации) и атрибут content (содержащий информацию как таковую). Хотя нет ограничения на количество потенциальных метаэлементов, самые распространенные — description и keywords, поскольку некоторые поисковые машины используют их.

```
<meta name="description" content="Sugar Beat Music for Children offers
age-appropriate music classes for children 4 months to 5 years old" />
```

В *разд. "Вставляйте описание страницы" главы 11* приводится более подробная информация о метаэлементах и поясняется, как их используют поисковые машины.

<noscript> (замещающий контент сценария)

Блочный элемент, элемент-контейнер.

Элемент <noscript> задает контент, который браузер должен отобразить, когда он не может выполнить сценарий. Элемент <noscript> должен располагаться сразу за элементом <script>. Чаще всего браузер не может выполнить сценарий, если посетитель специально отключил в настройках браузера поддержку сценариев.

Дополнительную информацию о сценариях см. в *главе 15*.

<object> (встроенный объект)

Строчный элемент, элемент-контейнер.

Элемент <object> встраивает специальные объекты на вашу страницу, такие как аудио, видео и даже апплеты (маленькие программки, выполняющиеся на вашей Web-странице). Например, элемент <object> можно использовать для вставки Flash-видеоклипа на Web-страницу, как описано в *главе 17*.

 (нумерованный список)

Блочный элемент, элемент-контейнер.

Нумерованный список начинается с элемента <ol> и содержит множественные пункты, каждый из которых представлен элементом <li>. В нумерованном списке

ваш браузер последовательно помечает каждый пункт списка с помощью цифр, букв или римских цифр.

Далее приведен простой нумерованный список из трех пунктов, пронумерованных от 1 до 3.

```
<ol>
  <li>Buy bread</li>
  <li>Soak stamps off letters</li>
  <li>Defraud government with offshore investment scheme</li>
</ol>
```

Для того чтобы пронумеровать список не с 1, задайте атрибут `start` и укажите начальное значение. Для изменения формата нумерации используйте атрибут `type` с одним из следующих значений: 1 (числа), a (строчные буквы), A (заглавные буквы), i (строчные римские цифры), I (заглавные римские цифры).

В HTML5 добавлен атрибут для обратно нумеруемых списков. Для создания такого списка в элемент `<ol>` вставляется атрибут `reversed="reversed"`. Но в настоящее время ни один из браузеров не поддерживает этот прием.

Дополнительную информацию о нумерованных списках см. в *разд. "Нумерованные списки" главы 5*.

<option> (пункт меню)

Разрешен только в элементе `<select>`, элемент-контейнер.

Элемент `<option>` определяет пункт в элементе управления, именуемом списком выбора, внутри элемента `<select>`. Например, если вы хотите создать раскрывающееся меню, позволяющее посетителям выбирать цвет из списка вариантов, включая Синий, Красный и Зеленый, вам понадобится один элемент `<select>` и три элемента `<option>` внутри него.

Когда описывается элемент `<option>`, можно использовать атрибут `selected="selected"`, чтобы сообщить браузеру о выборе этого элемента списка при первом отображении страницы. Можно также применить атрибут `value` для связи однозначно определяемой порции информации с элементом `<option>`, которая включается в данные формы при отправке ее посетителем на сервер.

Базовый пример см. в описании элемента `<select>`.

<p> (абзац)

Блочный элемент, элемент-контейнер.

Элемент `<p>` содержит абзац текста

```
<p>It was the best of times, it was the worst of times ...</p>
```

Поскольку абзацы — блочные элементы, браузер автоматически вставляет перенос строки и небольшую пробельную зону между абзацами или между абзацем и другим блочным элементом, таким как список или заголовок.

Пустые абзацы браузеры игнорируют. Для создания пустого абзаца, занимающего пространство стандартного размера, пользуйтесь неразрывным пробелом следующим образом:

```
<p> </p>
```

<param> (параметр объекта)

Разрешен только в элементе <object>, пустой элемент.

Элемент <param> задает дополнительную информацию в элементе <object>, которую браузер отправляет в апплет или плагин.

<pre> (предварительно отформатированный текст)

Блочный элемент, элемент-контейнер.

Предварительно отформатированный текст нарушает обычные правила форматирования, принятые в HTML. Внутри элемента <pre> браузер дублирует каждый встреченный пробел или перенос строки, сохраняя ваше исходное форматирование. Кроме того, браузер выводит весь контент моноширинным шрифтом (обычно Courier), что не всегда дает хорошие результаты. Элемент <pre> — это легкий и быстрый способ расположения текста на странице в соответствии с вашим желанием, который полезен для визуального представления стихотворения или фрагмента программного кода. Но его не следует применять для выравнивания больших порций обычного текста. Для этих задач используйте правила позиционирования CSS (см. *главу 9*).

```
<pre>
Tumbling-hair
                    picker of buttercups
                                          violets
    dandelions
And the big bullying daisies
                                through the field wonderful
    with eyes a little sorry
Another comes
                 also picking flowers
</pre>
```

<q> (короткая цитата)

Строчный элемент, элемент-контейнер.

Элемент <q> определяет короткую цитату в блочном элементе, например, абзаце.

```
<p>As Charles Dickens once wrote, <q>It was the best of times, it was the
worst of times</q>.</p>
```

Обычно браузеры отображают элемент <q> как курсивный текст, а некоторые браузеры, например Firefox, заключают текст в кавычки. Но не применяйте элемент <q>

просто для форматирования. Используйте его по прямому назначению, а затем добавьте правила таблицы стилей для задания нужного вам форматирования.

Если вы хотите применить длинную цитату, представляющую собой отдельный блочный элемент, воспользуйтесь элементом <blockquote>, описанным ранее.

<samp> (стандартный вывод)

Строчный элемент, элемент-контейнер.

Вы можете применить элемент <samp> для формирования компьютерного вывода — например, текста, отображаемого в консольном окне после выполнения вами какого-нибудь управляющего сценария. Этот редко используемый элемент просто выводит свое содержимое моноширинным шрифтом, как <pre> и <code>.

<script> (клиентский сценарий)

Блочный элемент, элемент-контейнер.

Элемент <script> вставляет клиентский сценарий на вашу Web-страницу. Сценарий или скрипт — это набор инструкций, написанных на облегченном языке программирования, например, JavaScript. Web-разработчики применяют сценарии для создания более интерактивных Web-страниц, добавляя, например, кнопки, меняющие свой цвет при наведении на них указателя мыши. Узнать больше об основах JavaScript и увидеть сценарии в действии можно в *главе 15*.

<select> (список выбора)

Разрешен только в элементе <form>, элемент-контейнер.

Элемент <select> описывает внутри формы элемент управления "список". Ваш посетитель может выбрать из списка один элемент (или несколько элементов, если вы добавите атрибут multiple="multiple"). Атрибут name применяется для однозначной идентификации вашего элемента управления, как в следующем примере.

```
<select name="PromoSource">
  <option value="Ad">Google Ad</option>
  <option value="Search">Google Search</option>
  <option value="Psychic">Uncanny Psychic Intuition</option>
  <option value="Luck">Bad Luck</option>
</select>
```

Обычно элементы управления формируют списки выбора как раскрывающиеся меню. Но вы можете с помощью атрибута size создать прокручиваемый список. Просто задайте количество строк, которые хотите видеть одновременно.

```
<select name="PromoSource" size="3">
  ...
</select>
```

Пример формы см. в *разд. "Базовая форма" главы 15*.

<small> (мелким шрифтом)

Строчный элемент, элемент-контейнер.

Используйте элемент <small> для хранения "мелкой печати", например, специальной юридической пометки в нижней части контракта. Визуально мелкий шрифт уменьшает размер текста на одну градацию, но если вам это не подходит, с помощью таблицы стилей можно изменить это форматирование.

(типовой строчный контейнер)

Строчный элемент, элемент-контейнер.

Используйте элемент для обозначения внутри блочного элемента текста, который хотите отформатировать. Например, можно задать особый формат одного слова в абзаце, целого предложения и т. д. Далее приведен пример.

```
<p>In this paragraph, some of the text is wrapped in a span element.
That <span>gives you the ability</span> to format it in some fancy
way later on.</p>
```

Сам по себе элемент не делает ничего. Но это очень полезный способ применения гибкого и многократного форматирования, заданного в таблице стилей.

Вы также можете обратиться к элементу , который применяется для одновременного форматирования нескольких блочных элементов и описан ранее.

(особая важность)

Строчный элемент, элемент-контейнер.

Элемент создает тот же визуальный эффект, что и элемент <b> (текст жирным шрифтом), но официальные правила HTML полагают, что вы применяете его к тексту, имеющему особую важность по сравнению с окружающими словами. Далее приведен пример.

```
I'm <strong>very</strong> sorry for all the trouble.
```

Если хотите выделить жирным шрифтом текст, не имеющий особого значения по сравнению с окружающими словами, лучше воспользоваться элементом <b>. Также имеет смысл с помощью таблицы стилей изменить форматирование элемента , если хотите выделить особо важный текст иным способом.

<style> (внутренняя таблица стилей)

Разрешен только в разделе <head>, элемент-контейнер.

Используйте элемент <style> для задания правил CSS (Cascading Style Sheet, Каскадная таблица стилей), форматирующих Web-страницу. Всегда помещайте элемент <style> внутрь раздела <head> на Web-странице.

Элемент <style> позволяет определить стиль прямо внутри Web-страницы. Его называют в этом случае *внутренней таблицей стилей*. Далее приведен пример, задающий цвет фуксия для заголовков <h1>.

```
<style>
  h1 { color: fuchsia }
</style>
```

Гораздо чаще вы будете применять элемент <link> вместо элемента <style>, чтобы установить связь с отдельным файлом, определяющим ваши стили. В этом случае вы сможете применять одни и те же стили к нескольким страницам, не засоряя вашу разметку. Дополнительную информацию о таблицах стилей и способах их применения см. в *главе 6*.

<*sub*> (нижний индекс)

Строчный элемент, элемент-контейнер.

Элемент <sub> форматирует текст так, что он отображается меньше и ниже (средняя линия такого текста выравнивается по нижней линии текущей строки). Лучше всего не пользоваться этим приемом форматирования (применяйте таблицы стилей), но это очень удобный способ для форматирования научных формул и обозначений, например H_2O. Далее показано, как его применять.

```
Water is H<sub>2</sub>O
```

<*sup*> (верхний индекс)

Строчный элемент, элемент-контейнер.

Элемент <sup> форматирует текст так, что он отображается меньше и выше (средняя линия такого текста выравнивается по верхней линии текущей строки). Лучше всего не пользоваться этим приемом форматирования (применяйте таблицы стилей), но это очень удобный способ для форматирования, например, показателей степени (3^3). Далее показан элемент <sup> в действии.

```
3<sup>3</sup> is 27
```

<*table*> (таблица)

Блочный элемент, элемент-контейнер.

Элемент <table> — элемент внешнего уровня в описании таблицы. Внутри элемента <table> вы задаете строки с помощью элемента <tr> и внутри каждой строки применяете элемент <td> для описания отдельных ячеек и их содержимого. Далее приведена разметка стандартной таблицы.

```
<table>
  <tr>
    <td>Row 1, Column 1</td>
    <td>Row 1, Column 2</td>
   </tr>
```

```
  <tr>
    <td>Row 2, Column 1</td>
    <td>Row 2, Column 2</td>
   </tr>
</table>
```

Выглядит она следующим образом:

Строка 1, Столбец 1	Строка 1, Столбец 2
Строка 2, Столбец 1	Строка 2, столбец 2

Дополнительную информацию о создании таблиц см. в *разд. "Простая таблица" главы 5*, а о задании размеров таблицы в *разд. "Задание размеров таблиц" главы 9*.

<td> (ячейка данных в таблице)

Разрешен только в элементе <tr>, элемент-контейнер.

Элемент <td> представляет отдельную ячейку внутри строки таблицы (элемент <tr>). Каждая вставка элемента <td> формирует столбец. Но вполне корректно иметь разное количество столбцов в последовательных строках (хотя это может выглядеть несколько странно). Пример стандартной таблицы см. в описании элемента <table>, приведенном ранее, а дополнительные пояснения, касающиеся таблиц, — в *главе 9*.

<textarea> (многострочное текстовое поле ввода)

Разрешен только в элементе <form>, элемент-контейнер.

Элемент <textarea> отображает большое текстовое поле в форме, способное вместить много строк текста. Как и в случае других элементов ввода формы, необходимо идентифицировать этот элемент управления с помощью уникального имени. Кроме того, вы можете задать размер текстового поля с помощью атрибутов rows и cols.

Если хотите, чтобы при первой загрузке страницы элемент <textarea> отображался с текстом внутри, поместите этот текст между начальным и конечным тегами следующим образом.

```
<textarea name="Comments">Enter your comments here.</textarea>
```

<th> (ячейка шапки таблицы)

Разрешен только в элементе <tr>, элемент-контейнер.

Элемент <th> представляет отдельную ячейку таблицы с текстом шапки таблицы. Применяйте элемент <th> так же, как элемент <td>. Разница лишь в том, что обычно его резервируют для первой строки таблицы (он представляет заголовки столбцов),

и текст в элементе <th> отображается жирным шрифтом и центрированным (что конечно можно изменить с помощью таблиц стилей).

<title> (название документа)

Разрешен только в разделе <head>, элемент-контейнер.

Элемент <title> задает название Web-страницы. Браузер отображает этот текст в строке заголовка и использует его как текст закладки, если посетитель создает закладку для данной страницы. Этот элемент следует помещать в раздел страницы <head>.

```
<title>Truly Honest Car Mechanics</title>
```

<tr> (строка таблицы)

Разрешен только в элементе <table>, элемент-контейнер.

Элемент <tr> представляет отдельную строку в таблице (элемент <table>). Для вставки ячеек с данными необходимо добавить внутрь элемента <tr> элемент <td>. Пример стандартной таблицы см. в описании элемента <table>, приведенном ранее.

 (маркированный список)

Блочный элемент, элемент-контейнер.

Маркированный список начинается с элемента <ul> и включает в себя множественные пункты списка, каждый из которых представляется элементом <li>. Браузер создает отступ для каждого элемента в списке и вставляет маркер перед ним. Далее приведен простой маркированный список.

```
<ul>
  <li>Buy bread</li>
  <li>Soak stamps off letters</li>
  <li>Defraud government with offshore investment scheme</li>
  </ul>
```

Web-разработчики часто применяют <ul> для создания меню команд. (Вы также можете использовать многоуровневые вложенные списки для создания иерархического меню.) Пользуясь этим приемом, применяйте правило таблицы стилей для удаления маркеров (задайте свойство list-style-type, равным none) и настройки свойств margin и padding.

Семантические элементы HTML5

Как вы узнали из *главы 2*, существует новый стандарт языка, потрясший Интернет: HTML5. И хотя HTML5 полон усовершенствований, он все равно столкнулся со значительным сопротивлением. Поскольку он столь нов, многие браузеры не под-

держивают его в полной мере. Internet Explorer особенно неповоротлив: вы не получите поддержки, пока не воспользуетесь абсолютно новой версией IE 9.

Из-за проблем с поддержкой браузерами Web-разработчики вынуждены применять HTML5 с опаской. Обходные приемы позволяют создавать страницы, применяющие свойства HTML5 и не оставляющие за бортом более старые браузеры, но многие из предлагаемых решений порождают больше проблем, чем они того заслуживают. (Например, в *разд. "Видео в HTML5" главы 17* рассказано о значительных трудностях, которые придется преодолеть, чтобы добиться поддержки HTML5-видео.)

Тем не менее одну группу элементов можно использовать уже сейчас и без особых усилий. Это *семантические элементы* из HTML5, делающие структуру ваших страниц более отчетливой. Например, можно в вашей разметке показать, где находится верхний колонтитул, где расположены навигационные ссылки и т. д. Ваши посетители никогда не увидят эту информацию, но вы сможете воспользоваться ею для различных целей. Например, ее могут использовать поисковые машины, чтобы узнать больше о вашем Web-сайте, с ее помощью программы считывания с экрана (экранные дикторы) смогут лучше представить ваше содержимое людям с ограниченными возможностями, и другие инструменты смогут извлекать ваши данные и повторно использовать их различными способами (например, в новостных рассылках, см. *разд. "Агрегирование" главы 13*). Все эти сценарии постоянно эволюционируют, но если вы планируете поддерживать свой Web-сайт длительное время, стоит освоить некоторые из новейших соглашений, чтобы получить от них выгоду, когда HTML5 станет официальным стандартом.

Если вы решите применить семантические элементы из HTML5, вам все же придется побеспокоиться об одной проблеме. Более старые браузеры (например, IE 8) не распознают эти элементы, они просто их проигнорируют. Это нормально до определенного момента, т. к. семантические элементы не касаются форматирования, они описывают структуру. Но многие из новых семантических элементов блочные, следовательно, браузер должен выводить их в отдельных строках на конечной Web-странице с небольшой пробельной зоной между ними и предыдущим (и последующим) элементом. Браузерам, не распознающим элементы HTML5 как блочные, вам придется объяснить, что делать с помощью правил таблицы стилей. Далее приведено суперправило, применяющее одновременное форматирование к отображаемому блоку из девяти элементов HTML5.

```
article, aside, figure, figcaption, footer, header, hgroup, nav, section,
summary {
  display: block;
}
```

Это правило таблицы стилей не окажет никакого влияния в браузерах, уже узнающих HTML5, поскольку они уже установили в свойстве `display` значение `block`.

Суперправило таблицы стилей решает часть проблемы, но не всю. Оно работает во всех старых браузерах, не понимающих HTML5, кроме Internet Explorer. Вся беда в том, что IE отказывается применять форматирование из таблицы стилей к незна-

комым элементам. К счастью, у вас есть обходной прием: вы можете обманом заставить Internet Explorer узнавать чужеродный элемент, зарегистрировав его с помощью команды на JavaScript. Далее приведен блок сценария, дающий возможность IE узнавать и форматировать элемент <header>.

```
<script>
  document.createElement('header')
</script>
```

Вместо самостоятельного написания подобного кода можно воспользоваться готовым сценарием, который делает все за вас (описан на странице **http://tinyurl.com/nlcjxm**). Для применения этого сценария вы просто вставляете ссылку на него в раздел своей страницы <head>, например, следующим образом.

```
<head>
  <title>...</title>
  <script src="http://html5shim.googlecode.com/svn/trunk/html5.js"></script>
<head>
```

Данный код берет сценарий с Web-сервера html5shim.googlecode.com и выполняет его до того, как браузер начнет обрабатывать остальную часть страницы. Сценарий короткий, и по существу он использует для создания всех новых элементов HTML5 прием на JavaScript, описанный ранее. Это позволяет форматировать элементы с помощью правил таблиц стилей. Добавьте приведенное суперправило в вашу таблицу стилей, и новые элементы будут отображаться как корректные блочные элементы. Вам остается только применить элементы и добавить собственные правила таблицы стилей для их форматирования.

В следующих разделах перечислены все новые семантические элементы из HTML5. Отсутствуют только те элементы из HTML5, для которых нет удобных обходных приемов и хитростей, позволяющих обрабатывать их в старых браузерах. К таким относятся элементы HTML5 для работы с аудио- и видеоданными (которые до сих пор вовлечены в войны видеокодеков, описанные в *разд. "Видео в HTML5" главы 17*), новые виджеты HTML5, которые применяются при конструировании форм (см. *разд. "Интерактивные формы" главы 15*), и новый элемент <canvas>, используемый для рисования фигур с помощью JavaScript.

> ***Совет***
>
> Самую последнюю версию спецификации HTML5, содержащую все элементы, которые выводят старые браузеры из игры, см. на странице **http://dev.w3.org/html5/markup**. Узнать больше обо всех новых элементах HTML5, включая поддерживаемые стандарты для построения улучшенных Web-приложений с JavaScript, можно из книги "HTML5: The Missing Manual".

<*article*> (статья)

Блочный элемент, элемент-контейнер.

Представляет все, что вы воспринимаете как статью — раздел законченного контента, такого как газетная статья, публикация на форуме или публикация в блоге

(не включая такие дополнения, как комментарии или биографические данные автора).

Как и большинство семантических элементов HTML5, элемент <article> не содержит никакого встроенного форматирования. Воспринимайте его как более информативную версию типового контейнера .

НА ПРОФЕССИОНАЛЬНОМ УРОВНЕ

Структурирование документа с помощью корневых узлов сегментирования

Элемент <article> — пример немного абстрактного понятия из HTML5, именуемого *корневыми узлами сегментирования* (sectioning roots). По существу корневые узлы сегментирования формируют структурный план вашей страницы. Если у страницы хороший план, другим инструментам и программам легче понять ее структуру.

HTML5 считает корневыми узлами сегментирования следующие элементы: <body>, <section>, <article>, <aside>, <footer>, <header> и <nav>. Все они — элементы верхнего уровня, формирующие структурный план страницы. Другими словами, если вы вставляете на вашу страницу корневой узел сегментирования, например, <article>, вы создаете новый структурный план. (Остальная часть плана создается из заголовков внутри этого сегмента. Начинаете с заголовка уровня <h1>, далее используете дополнительные уровни, например, <h2>, <h3> и т. д., сколько потребуется.)

Концепция корневых узлов сегментирования приобретает важность, если у вас несколько порций законченного контента на одной и той же странице. Рассмотрим, к примеру, новостной сайт, отображающий несколько статей одну за другой или страницу блога с несколькими публикациями одновременно. В традиционном HTML весь этот контент объединяется в один плохо структурированный макет. Но благодаря корневым узлам сегментирования HTML-разметка четко показывает разделение, поскольку каждая статья или публикация приобретает собственный корневой узел.

К несчастью, концепция корневых узлов сегментирования и структурных планов страниц все еще скорее теоретическая, т. к. редакторы Web-страниц, браузеры, поисковые машины и устройства-помощники (например, считыватели экрана) до сих пор не получают никаких преимуществ от всей этой информации. Когда это произойдет, корневые узлы сегментирования станут ключевой концепцией, помогающей Web-разработчикам создавать хорошо структурированные страницы. В настоящее время вы можете изучить макет любой HTML5-страницы с помощью средства определения структуры на странице **http://gsnedders.html5.org/outliner**.

<aside> (боковая панель)

Блочный элемент, элемент-контейнер.

Представляет законченную порцию контента, который отделен от основного содержимого страницы. Например, имеет смысл использовать <aside> для создания рядом с основным контентом боковой панели со связанным контентом или ссылками. Его также можно применить для формирования блока рекламных объявлений.

<figcaption> (подрисуночная надпись)

Блочный элемент, элемент-контейнер.

Элемент <figcaption> включает в себя текст подрисуночной надписи, связанной с элементом <figure> (см. дальше). Цель — четко обозначить связь между изобра-

жением и его подрисуночной надписью. Конечно, <figcaption> не ограничивается только текстом. Можно использовать HTML-элементы, имеющие смысл, например, включить в него ссылки или крошечные пиктограммы.

<figure> (рисунок)

Блочный элемент, элемент-контейнер.

Элемент <figure> обрамляет изображение и связанную с ним подрисуночную надпись. Далее приведен пример.

```
<figure>
   <img src="planetree.jpg" alt="Plane Tree" />
   <figcaption>The bark of a plane tree</figcaption>
</figure>
```

В большинстве случаев для позиционирования элемента <figure> придется использовать класс в таблице стилей. Например, можно выбрать плавающий элемент, прижатый к левому или правому краям вашей страницы.

<footer> (нижний колонтитул или подвал)

Блочный элемент, элемент-контейнер.

Нижний колонтитул — это порция контента, находящаяся внизу страницы. В нее могут входить юридические пометки мелким шрифтом, знак авторского права и небольшой набор ссылок (например, **About Us** (О нас) или **Get Support** (Техническая поддержка)).

<header> (верхний колонтитул)

Блочный элемент, элемент-контейнер.

Элемент <header> представляет шапку, включающую заглавие и некоторый контент. Например, шапка перед статьей может содержать заглавие и имя автора или заглавие и несколько ссылок на подтемы, обсуждаемые в статье. Далее приведен пример.

```
<header>
  <h1>An HTML5 Investigation</h1>
  <p>Prepared by Steven Smith</p>
</header>
```

Элемент <header> может включать в себя заголовочный раздел Web-сайта (например, баннер с логотипом компании). Вполне приемлемо иметь на Web-странице несколько секций <header> при условии, что каждая шапка принадлежит разному контенту.

Если нужно объединить название и подзаголовок, следует использовать элемент <hgroup>, т. к. он допускает заголовки разных уровней. Но если есть смысл, можно сочетать элементы <header> и <hgroup>.

```
<header>
  <hgroup>
    <h1>An HTML5 Investigation</h1>
    <h2>Part 1: Getting the Real Story</h2>
  </hgroup>
  <p>Prepared by Steven Smith</p>
</header>
```

<hgroup> (группа верхнего колонтитула)

Блочный элемент, элемент-контейнер.

Элемент <hgroup> — особая форма верхнего колонтитула, группирующая несколько элементов-заголовков и ничего более. Его основное назначение — отобразить вместе название и подзаголовок, например, так:

```
<hgroup>
  <h1>An HTML5 Investigation</h1>
  <h2>Part 1: Getting the Real Story</h2>
</hgroup>
```

Со структурной точки зрения, заголовки, сгруппированные вместе в элементе <hgroup>, браузер трактует как единый суперзаголовок. Если же применить ту же разметку, но без элемента <hgroup>, элемент <h2> считается подсекцией в заголовке <h1>. Это не влияет на внешний вид Web-страницы в браузере, но меняет структуру вашей страницы, что может повлиять на другие программы, анализирующие эту структуру, например, средства Web-дизайна, поисковые машины и экранные дикторы.

<mark> (выделенный текст)

Строчный элемент, элемент-контейнер.

Элемент <mark> представляет порцию текста, которую вы хотите выделить для обозначения чего-либо. Например, его можно использовать для пометки изменений, ошибок или ключевых слов.

Если вы решите применять элемент <mark> сейчас, необходимо задать форматирование для браузеров, до сих пор не поддерживающих HTML5. Далее приведен пример нужного вам стилевого правила.

```
mark {
  background-color: yellow;
  color: black;
}
```

<nav> (навигационные ссылки)

Блочный элемент, элемент-контейнер.

Элемент <nav> представляет секцию страницы, содержащую ссылки. Эти ссылки могут указывать на рубрики на текущей странице или на других страницах Web-

сайта. В действительности нет ничего необычного в том, что на странице есть несколько секций <nav> — одна для навигационного меню всего Web-сайта, вторая для рубрик в текущей статье и т. д.

Следует отметить, что необязательно каждый блок ссылок включать в элемент <nav>. Web-разработчики обычно резервируют его для самых больших и наиболее важных навигационных секций страницы. Элемент <nav> также можно помещать в другой элемент HTML5. Например, если у вас есть набор навигационных ссылок прямо в верхнем колонтитуле, вы можете поместить элемент <nav> внутрь элемента <header>. Или, если решите, что ссылки не являются явно частью верхнего колонтитула, можно вставить элемент <header>, а следом за ним элемент <nav>. В спецификации HTML5 четко указывается, что подобные решения — дело вкуса.

<*section*> (секция)

Блочный элемент, элемент-контейнер.

Элемент <section> — самый общий тип узла сегментирования (см. врезку в описании элемента <article>). Он представляет документ, который должен начинаться с заголовка любого уровня.

Если есть возможность, следует применять контейнер, более специализированный, чем <section>. Например, у элементов <article> или <aside> конкретное назначение, что делает их более удобными, чем элемент <section>. Если вы не можете описать свой контент как статью или некое отступление (например, это история болезни, извлеченная из базы данных), тогда элемент <section> вполне подходит для его идентификации.

Одна деталь, сбивающая с толку, — вы можете использовать элемент <section> с *частями* Web-страницы. Например, на исходной странице вашего Web-сайта можно поместить публикации из блога, новости и блок рекламных объявлений на боковой панели, заключив каждую часть в отдельный элемент <section>.

> ***Примечание***
>
> Когда применение таких элементов, как <section>, станет для Web-разработчиков привычным делом, люди начнут полагаться на несколько общеупотребительных приемов использования. А до тех пор в HTML5 остается некоторая неоднозначность, касающаяся наилучших способов применения его элементов.

<*time*> (дата или время)

Строчный элемент, элемент-контейнер.

Информация о дате и времени часто отображается на Web-страницах. Например, она появляется в конце большинства публикаций в блогах. К сожалению, нет стандартных тегов для дат, поэтому у других программ (таких как роботы поисковых машин) нет легкого способа извлечения даты без подсчетов и допущений. Элемент <time> решает эту проблему двумя способами. Он не только позволяет пометить

имеющуюся дату, но и вставить атрибут `datetime`, который можно использовать для задания даты в стандартной форме, понятной любой программе.

Далее приведено несколько примеров использования элемента `<time>`.

```
<time datetime="2011-11-30">30<sup>th</sup> of November</time>
<time datetime="20:00">8:00pm</time>
<time datetime="2011-11-30T20:30">8:30 PM on November 11, 2011</time>
```

Имейте в виду, информация из атрибута `datetime` не будет отображаться на вашей Web-странице. Она предназначается для считывания другими программами.

Поскольку элемент `<time>` чисто информативный и не имеет никакого связанного с ним форматирования, его можно применять в любом браузере.

Символьные HTML-последовательности

Символьные HTML-последовательности — это коды, которые браузер при отображении Web-страницы преобразует в другие символы. Все символьные HTML-последовательности начинаются с амперсанда (&) и заканчиваются точкой с запятой (;).

Для применения символьных HTML-последовательностей есть две веские причины. Прежде всего, вам может понадобиться символ, имеющий специальное назначение в HTML-стандарте. Например, если вы набираете в HTML-документе символ <, браузер полагает, что начинается описание элемента, и это затрудняет написание логических выражений, таких как "2 < 3". Для решения этой проблемы символ < заменяют символьной последовательностью, которая *представляет* символ "меньше чем". Затем при отображении страницы браузер вставляет нужный символ <.

Вторая причина применения символьных HTML-последовательностей — использование специальных символов, которые непросто ввести, например, буква с диакритическим знаком или символ валюты. Вполне возможно, что подобные символы вообще отсутствуют на вашей клавиатуре.

В табл. П1.2 приведены самые часто используемые HTML-последовательности. Более полный список, содержащий множество интернациональных языковых символов, см. на странице **www.webmonkey.com/reference/Special_Characters**. Вы также можете вводить определенные специальные символы с помощью неанглийской клавиатуры или выбрать международные языковые символы из специальной программы. Дополнительную информацию об этих возможностях см. в *разд. "Иностранные языки" главы 5*.

Таблица П1.2. *Символьные HTML-последовательности*

Символ	Название символа	Что вводить
<	Знак "меньше"	`<`
>	Знак "больше"	`>`

Таблица П1.2 *(окончание)*

Символ	Название символа	Что вводить
&	Амперсанд	`&`
"	Кавычки	`"`
©	Копирайт	`©`
®	Зарегистрированный товарный знак	`®`
¢	Цент	`¢`
£	Фунт стерлингов	`&pound`
¥	Йена	`¥`
€	Евро	`€` (но у `€` лучше поддержка)
°	Градус	`°`
±	Плюс-минус	`±`
÷	Знак деления	`÷`
×	Знак умножения	`×`
µ	Знак микро	`µ`
¼	Дробь, одна четверть	`¼`
½	Дробь, одна вторая	`½`
¾	Дробь, три четверти	`¾`
¶	Знак абзаца	`¶`
§	Знак параграфа	`§`
"	Открывающие угловые кавычки	`«`
"	Закрывающие угловые кавычки	`»`
¡	Перевернутый восклицательный знак	`¡`
¿	Перевернутый вопросительный знак	`¿`
æ	Строчный дифтонг ae (лигатура)	`æ`
ç	Строчная c, седиль	`ç`
è	Строчная e, с тяжелым ударением	`è`
é	Строчная e, с восходящим акцентом	`é`
ë	Строчная e, трема или умляут	`ë`
ö	Строчная o, трема или умляут	`ö`
É	Заглавная E, с восходящим акцентом	`É`

Названия цветов в HTML

Официально HTML-стандарт распознает только 16 цветов. Они перечислены в табл. П1.3.

Таблица П1.3. *Названия цветов в HTML*

Aqua (голубой)	Navy (темно-синий)
Black (черный)	Olive (оливковый)
Blue (синий)	Purple (лиловый)
Fuchsia (фуксия)	Red (красный)
Gray (серый)	Silver (серебристый)
Green (зеленый)	Teal (сине-зеленый)
Lime (ярко-зеленый)	White (белый)
Maroon (темно-бордовый)	Yellow (желтый)

Несмотря на то что многие браузеры распознают больше цветов, надежный способ получения нужной цветовой гаммы — применение кодов цветов (см. *разд. "Задание цвета" главы 6*).

2

ПРИЛОЖЕНИЕ 2

Полезные Web-сайты

Прочитав эту книгу, вы узнали о множестве замечательных Web-сайтов, позволяющих загрузить полезное программное обеспечение или получить важную информацию. Скорее всего вы захотите снова посетить некоторые из них, чтобы отработать Web-навыки (или просто получить бесплатный материал). Для того чтобы вам не пришлось перелистывать сотни страниц, в этом приложении приводятся все ссылки, сгруппированные по главам.

Вам не нужно скрупулезно вводить эти URL в адресную строку вашего браузера и приобретать туннельный синдром запястья. Просто воспользуйтесь электронной версией приложения на странице Missing CD (**www.missingmanuals.com/cds/caw3**). В этом случае, когда нужный сайт найден, щелкните его кнопкой мыши. Кроме того, на этой странице можно найти свежие изменения (например, URL, перемещенные в другое место).

Ссылки по главам

В следующих таблицах приведены ссылки, найденные в каждой главе. Ссылки перечисляются в том порядке, в каком они встретились в тексте. Здесь вы найдете все виды ссылок. Некоторые указывают на полезные обучающие сайты, другие — на забавные Web-курьезы, а большинство — на удачные бесплатные средства или загружаемые изображения и медиаданные.

Глава 1. *Подготовка к вступлению в Web*

Описание	URL
История Интернета	www.isoc.org/internet/history www.walthowe.com/navnet/history.html
Статистика применения браузеров	http://en.wikipedia.org/wiki/Usage_share_of_web_ browsers/
Internet Explorer (браузер)	www.microsoft.com/ie
Firefox (браузер)	www.mozilla.org/firefox

Глава 1 *(окончание)*

Описание	URL
Google Chrome (браузер)	www.google.com/chrome
Safari (браузер)	www.apple.com/safari
Opera (браузер)	www.opera.com
Сводка о браузерах Mac	www.knutson.de/mac/www/browsers.html
Один из множества бесплатных сервисов ведения блогов (дополнительный перечень см. в *главе 13*)	www.blogger.com
Чего не следует делать на Web-странице	www.angelfire.com/super/badwebs
Яркие примеры плохого Web-дизайна	www.worstoftheweb.com

Глава 2. *Создание вашей первой Web-страницы*

Описание	URL
Flash-игры	www.ferryhalim.com/orisinal
HTML-валидатор	http://validator.w3.org

Глава 3. *Размещение вашей страницы в Web*

Описание	URL
Домены верхнего уровня, ориентированные на конкретную страну	http://tinyurl.com/rnlmf
Smartdots (бесплатные имена поддоменов)	www.smartdots.com
Сервис Google Sites (упрощающий разработку страниц)	http://sites.google.com
Brinkster (Web-хост)	www.brinkster.com
DreamHost (Web-хост)	www.dreamhost.com
GoDaddy (Web-хост)	www.godaddy.com
HostGo (Web-хост)	www.hostgo.com
Insider Hosting (Web-хост)	www.insiderhosting.com
OCS Solutions	www.ocssolutions.com
Pair Networks (Web-хост)	www.pair.com
Sonic.net (Web-хост)	www.sonic.net
WebHostingTalk (форум)	http://tinyurl.com/5zffwp
TinyURL (сервис сокращения URL)	http://tinyurl.com
Каталог бесплатных Web-хостов	www.free-webhosts.com

Глава 4. *Специализированные средства*

Описание	URL
Условно-бесплатные программы	www.download.com
Список редакторов Web-страниц	http://en.wikipedia.org/wiki/Comparison_of_HTML_editors
Blue Griffon (редактор Web-страниц)	www.bluegriffon.org
Amaya (редактор Web-страниц)	www.w3.org/Amaya
HTML-Kit (редактор Web-страниц)	www.htmlkit.com
CoffeeCup (редактор Web-страниц)	www.coffeecup.com/free-editor
Expression Web (пробная версия)	www.microsoft.com/Expression/try-it
Dreamweaver (пробная версия)	www.adobe.com/products/dreamweaver

Глава 5. *Текстовые элементы*

Описание	URL
Специальные символы в HTML	www.webmonkey.com/reference/Special_Characters

Глава 6. *Таблицы стилей*

Описание	URL
CSS-совместимость таблиц для разных браузеров	www.quirksmode.org/css/contents.htmlhttp://caniuse.com
Сетевые цветовые палитры	www.colorpicker.com www.colorschemer.com/online.html
Поддержка шрифтов в разных операционных системах	http://tinyurl.com/cr9oyx http://tinyurl.com/325f9qs
Поддержка встроенных шрифтов в разных браузерах	http://tinyurl.com/cluxzz
Комплекты готовых шрифтов	www.fontsquirrel.com/fontface
Каталог шрифтов Google	http://code.google.com/webfonts
Генератор комплектов шрифтов	www.fontsquirrel.com/fontface/generator

Глава 7. *Добавление графики*

Описание	URL
Применение графики в формате SVG	http://tinyurl.com/2gy2vyg
Использование альфа-смешивания в Photoshop	http://tinyurl.com/ypf78g

Глава 7 *(окончание)*

Описание	URL
Бесплатные графические редакторы	www.gimp.org (все платформы) www.getpaint.net (только Windows) www.aviary.com (сетевой)
Бесплатные фоновые изображения	www.grsites.com/textures www.backgroundsarchive.com
Средство Google для поиска графики (использование изображений не всегда бесплатное)	http://images.google.com
Stock.XCHNG (бесплатные изображения)	http://sxc.hu
Flickr (страница расширенного поиска)	www.flickr.com/search/advanced
Обзор сайтов для поиска бесплатных изображений	http://tinyurl.com/49yquv3
Коммерческие сайты изображений	www.istockphoto.com www.fotolia.com www.dreamstime.com
Иллюстративные заготовки Microsoft Office	http://office.microsoft.com/images

Глава 8. *Связывание страниц*

Описание	URL
Средство кодирования текста электронного письма	http://meyerweb.com/eric/tools/dencoder
Проверка ссылок	http://validator.w3.org/checklink

Глава 9. *Макет страницы*

Описание	URL
Расширение Web Developer для Firefox и Chrome	http://chrispederick.com/work/web-developer
Огромный каталог примеров верстки средствами таблиц стилей	www.csszengarden.com
Создание клейкого нижнего колонтитула	www.cssstickyfooter.com
Пример резинового макета	www.dccdesign.co.uk
Пример эластичного макета	http://tinyurl.com/67v6hb6
Статья о фиксированной, резиновой и эластичной верстке	http://tinyurl.com/q54tnl

Глава 10. *Страницы с многократно используемыми элементами*

Описание	URL
Электронная книга о шаблонах страниц в Expression Web	http://tinyurl.com/9whbc7

Глава 11. *Обнародование вашего сайта*

Описание	**URL**
Каталог Open Directory Project	http://dmoz.org http://dmoz.org/add.html (правила подачи заявки) http://dmoz.org/guidelines/ (пособие по редактированию)
Каталог Google	http://directory.google.com
Каталог Yahoo	http://dir.yahoo.com
Руководство по оформлению заявок в каталог Yahoo	http://tinyurl.com/47quxhk www.apromotionguide.com/yahoo.html (неофициальное) https://ecom.yahoo.com/dir/submit/intro (сервис Yahoo Directory Submit)
Заявка на регистрацию в Google	www.google.com/addurl
Заявка на регистрацию в Bing	www.bing.com/webmaster/SubmitSitePage.aspx
Как работает система PageRank в Google	www.markhorrell.com/seo/pagerank.html http://en.wikipedia.org/wiki/PageRank
Google Toolbar	www.google.com/toolbar (загрузка) http://tinyurl.com/64bjmtd (включение средства PageRank)
Новости индустрии поисковых машин	www.webmasterworld.com www.searchengineland.com
Список Web-роботов	www.robotstxt.org
Wayback Machine (архив Web-страниц)	www.archive.org
Сервис Google Webmaster Tools	www.google.com/webmasters/tools www.google.com/webmasters/tools/removals (удаление сайта)
Сервис Google Analytics	www.google.com/analytics

Глава 12. *Продвижение Web-сайта*

Описание	**URL**
Сервис Google Places	http://places.google.com/business
Google AdWords	http://adwords.google.com
О цене слов на AdWords	www.iterature.com/adwords
Сценарий отправки формы	www.freecontactform.com/free.php
Сервисы отправки форм	www.emailmeform.com www.123contactform.com
Twitter	http://twitter.com

Глава 12 *(окончание)*

Описание	**URL**
Самые популярные подписчики Twitter	http://twitaholic.com/top100/followers
Сервис Google Groups	http://groups.google.com http://groups.google.com/googlegroups/overview.html
Примеры Fan-страниц в Facebook	www.facebook.com/kristof www.facebook.com/benjerry http://tinyurl.com/6amf47k
Facebook	www.facebook.com/pages/create.php (создание Fan-страницы) http://tinyurl.com/4z9wzlz (получение привлекательного URL) http://developers.facebook.com/docs/plugins (получение кнопки **Like**)

Глава 13. *Блоги*

Описание	**URL**
Поисковые машины для блогов	http://technorati.com
Примеры блогов	http://stefaniewildertaylor.com www.rottengods.com www.schneier.com http://wilwheaton.typepad.com http://thesartorialist.blogspot.com http://roseandsnail.com
Примеры групповых блогов	http://lifehacker.com www.dailykos.com www.thehuffingtonpost.com
Сетевые программы чтения RSS-лент	www.google.com/reader www.newsgator.com
Windows-программа чтения RSS-лент	www.feeddemon.com
Программа чтения RSS-лент для Mac	http://netnewswireapp.com
Браузерные программы чтения RSS-лент	www.microsoft.com/windows/rss (IE) www.apple.com/safari/features.html (Safari)
Блоги с самостоятельным хостингом	http://wordpress.org www.movabletype.org
Блоги на базе сервиса хостинга	www.blogger.com http://wordpress.com www.typepad.com

Глава 13 *(окончание)*

Описание	URL
Системы управления контентом	http://drupal.org www.joomla.org
Информация сервиса Blogger	www.bloggerforum.com (дискуссионный форум) http://help.blogger.com (официальное справочное руководство) http://tinyurl.com/n2zsbl (использование пользовательского домена) http://tinyurl.com/4tbxvl8 (скрытие NavBar) http://tinyurl.com/295vg5 (справка по тегам шаблона) http://tinyurl.com/3n4hvu (создание закладки BlogThis)

Глава 14. *Как зарабатывать деньги на своем сайте*

Описание	URL
Google AdSense	www.google.com/adsense (регистрация) www.google.com/services/adsense_tour www.google.com/adsense/taxinfo www.google.com/adsense/policies
Amazon Associates	http://affiliate-program.amazon.com http://tinyurl.com/46cqc84 (баннеры)
CafePress	www.cafepress.com
Zazzle	www.zazzle.com
PayPal	www.paypal.com www.paypal.com/chargeback www.paypal.com/SellerProtection www.paypal.com/fees

Глава 15. *JavaScript: добавка интерактивности*

Описание	URL
Свойства JavaScript	www.w3schools.com/htmldom/dom_reference.asp
События JavaScript	www.w3schools.com/jsref
Примеры на JavaScript	www.dynamicdrive.com www.huddletogether.com/projects/lightbox2 www.javascriptkit.com/cutpastejava.shtml www.webmonkey.com/tutorial/JavaScript_Tutorial

Глава 15 *(окончание)*

Описание	URL
Библиотеки JavaScript	http://jquery.com http://mootools.net www.prototypejs.org http://dojotoolkit.org
Библиотеки виджетов	www.widgetbox.com www.google.com/ig/directory?synd=open www.widgipedia.com

Глава 16. *Привлекательные кнопки и меню*

Описание	URL
Генератор изображений кнопки	www.buttongenerator.com www.grsites.com/button
Генератор CSS-кнопок	http://css-tricks.com/examples/ButtonMaker
Меню Slashdot	www.dynamicdrive.com/dynamicindex1/slashdot.htm

Глава 17. *Аудио и видео*

Описание	URL
Flash-игры	www.ferryhalim.com/orisinal
Редакторы WAV/MP3	www.goldwave.com www.fleximusic.com
iTunes	www.apple.com/itunes
Flash-плейер MP3	www.premiumbeat.com/flash_music_players
Flash-лупы для фоновой музыки	www.flashkit.com/loops
Flash-видеоплейеры	http://flowplayer.org www.premiumbeat.com/flash_video_players
Expression Encoder	http://tinyurl.com/243b7sr
Поддержка видеостандарта HTML5 разными браузерами	http://camendesign.com/code/video_for_everybody
YouTube	www.youtube.com http://tinyurl.com/66onkub (рубрики справки) http://tinyurl.com/2kxnv3 (как настроить окно видео)

Описание	**URL**
Обходной прием на JavaScript для HTML5 в Internet Explorer	http://tinyurl.com/nlcjxm
Спецификация HTML5	http://dev.w3.org/html5/markup
Построитель схемы для HTML5	http://gsnedders.html5.org/outliner
Специальные символы в HTML	www.webmonkey.com/reference/Special_Characters

Предметный указатель

A

Adobe Dreamweaver 95
AdWords 349
Amaya 93
Amazon Associates 444
Animated GIF 538
AVI 538

B

Bing 323
Blogger 383, 391
BlueGriffon 91

C

CoffeeCup Free HTML Editor 94
CSS 118, 154
CSS-комментарии 196

D

DNS 58
DNS-сервер 66
Doctype 35
Document Type Definition, DTD 35
Dreamweaver 124, 135, 141
Dynamic HTML 485

E

Expression Web 124, 141

F

Facebook 371
Fan-страница 372
Firefox 12
Flash 538
FLV 539
FTP-браузер 81
FTP-протокол 81

G

Google AdSense 424
Google Analytics 333
Google Chrome 13
Google Groups 361
Google Places 347
Google Webmaster Tools 329

H

HTML 23
HTML 4.01 29
HTML Kit Tools 93
HTML5 29, 30, 470
HTML-комментарий 44
HTML-таблица 137
HTML-элемент 32
HTTP 24, 56

I

Internet Explorer 11
IP-адрес 58

J

JavaScript 467

K

KompoZer 91

M

Mark of the Web 474
Microsoft Silverlight 555
MIDI 538
MP3 538
MPEG 538

N

NavBar 414
Nvu 91

O

ODP 319
Open Directory Project 319
Opera 13

P

PayPal 453

R

Rbrowser 83
RSS 388

S

Safari 13
Search Engine Optimization 325
SEO 325
String 473

T

Technorati 386
TinyURL 76
Twitter 358

U

URL 55
URL-адрес 21
Usenet 361

W

WAV 538
Web Developer 264
Web-браузер 10
Web-виджет 510
Web-приложение 30, 295
Web-сайт 103
Web-сервер 14
Web-страница
◊ пропорционального размера 264
◊ фиксированной ширины 264
Web-хост 14, 55

X

XHTML 1.0 29
XHTML 2 29

Y

Yahoo 323
YouTube 537, 557

Z

Z-индекс 283

А

Абсолютный URL 237
Агрегатор новостных лент 388
Агрегирование 387
Активная область изображения 249
Альфа-смешивание 214
Аргумент 481
Артефакты сжатия 211
Архивирование 404
Атрибут
◊ action 460
◊ alt 204, 317
◊ autoplay 543
◊ cite 130
◊ class 199
◊ colspan 140
◊ content 316
◊ event 494
◊ font-family 177
◊ hidden 543
◊ href 156, 235, 536
◊ id 252, 486
◊ name 316
◊ rel 156
◊ rowspan 141
◊ shape 250
◊ src 204
◊ start 133
◊ target 239
◊ title 205
◊ type 133, 134, 501
◊ value 501

Б

Базовая линия текста 215
Бесплатное программное обеспечение 91
Бесшовный фрейм 294
Блог 383
Блокировщик всплывающих окон 239
Блочный элемент 48, 120

В

Валидатор 52
◊ ссылок 257
Вернувшиеся посетители 351
Видеоблоги 557
Вложенный список 135
Вложенный элемент 33
Внешняя ссылка 237
Внешняя таблица стилей 152, 159
Внутренняя ссылка 237
Внутренняя таблица стилей 152, 159
Водяной знак 224
Возвратный платеж 453
Возмещение 453
Всемирная паутина 10
Встраивание мультимедиа 536
Встроенный стиль 153, 159
Выделенный сервер 76
Выложенное мультимедиа 537
Выравнивание по ширине 171

Г

Гаджет 412
Глубина рекурсии 259

Д

Домашняя страница 239
Домен верхнего уровня 61
Доменные сквоттеры 60

Е

Единица измерения em 283

З

Заголовки 124
Закладка 254

И

Идентификатор партнера 445
Иконка сайта 352
Имя домена 21
Интернет 10, 55
Интранет 59

К

Канал Adsense 433
Карта сайта 330
Каталог 319
◊ Yahoo 322
Клейкие нижние колонтитулы 282
Клиентские приложения 469

Клиентский скрипт 72
Код отслеживания 336
Кодировка Unicode 147
Команда return 483
Коммерческий счет 453
Комплект шрифтов 185
Контактные формы 356, 358
Контекстный селектор 166, 201
Корзина для покупок 461
Корневая папка 239
Кэширование 160

Л

Личный счет 454
Логический элемент 116
Лупы 550

М

Маркированный список 132, 519
Массив 528
Медиаплейер Yahoo 547
Метаданные 106
Метатег 315
Метаэлемент 315
Метод объекта 486
Модерация 417
Моноширинный шрифт 127
Мультимедиа 535

Н

Наследование стилей 161
Невидимое изображение 459
Неразрывный пробел 122
Нумерованный список 132

О

Объект 485
◊ document 486
◊ image 486
◊ this.form 505
Опробование домена 62
Относительный URL 237
Охват ячеек 139

П

Параметр 481
Парковка домена 66
Партнерская программа 421
Переадресация домена 66
Перекрывающиеся слои 283
Переменные 475
Перенаправление 260
◊ домена 66
Песочница 468
Пиксел 206, 265
Плагин 50
Поддомен 68
Показатель отказов 340
Пользовательская функция 479
Потеря ссылок 255
Потоковый элемент 121
Почтовая ссылка 246
Предварительная загрузка 520
Предварительно отформатированный текст 127
Презентационный элемент 116
Программа
◊ чтения RSS-лент 388, 390
◊ -робот 328
Прогрессивное улучшение 499
Пропорциональный шрифт 127
Пропускная способность 71
Псевдокласс 245
◊ :active 245
◊ :hover 245, 519
◊ :link 245
◊ :visited 245
Пустой элемент 32

Р

Разметка 31
Разрешение экрана 265
Регистрация домена 64
Редактируемая область 300
Редактор с двойным окном 89
Режим обратной совместимости IE 36
Резиновые макеты 282
Ролловер 494

С

Свойства объекта 486
Свойство
◊ @font-face 184
◊ background 519
◊ background-attachment 225
◊ background-color 166, 192
◊ background-image 224, 225

◊ background-position 225
◊ background-repeat 225
◊ border-color 191
◊ border-style 191
◊ border-width 191
◊ clear 219
◊ color 166
◊ cursor 526
◊ display 496, 519, 527
◊ float 217
◊ font-size 182
◊ height 269, 291
◊ innerHTML 487
◊ list-style-image 230
◊ margin 173, 192
◊ max-width 278
◊ min-width 278
◊ min-height 279
◊ padding 192
◊ position 270, 281
◊ right 275
◊ text-align 171
◊ text-decoration 246
◊ vertical-align 215
◊ white-space 174, 175
◊ width 289
Сворачивающаяся страница 496
Связывание мультимедиа 536
Секция CDATA 474
Селектор
◊ идентификатора 165
◊ класса 164
◊ типа 163
Семантический элемент 116
Семантическое Web-пространство 118
Серверное включение 294
Серверное приложение 468
Серверные языки программирования 72
Серверный скрипт 74
Серверный сценарий 356
Сервис создания форм 357
Символьная HTML-последовательность 34, 145
Синдикация 387
Синтаксис пустых элементов 32
Система управления контентом 391
Скрипт 469
Событие 492
Список
◊ воспроизведения 545
◊ определений 132
Средства проверки ссылок 257
Ссылка относительно
◊ документа 243
◊ корня сайта 243
Стандартный режим браузера 35
Строка запроса 57
Строчный элемент 49
Сценарий 469
Счет
◊ business 455
◊ premier 454

Т

Таблица стилей 118, 151
Твит 359
Тег 31
Текст в заданном формате 127
Текстовый HTML-редактор 89
Текстовый процессор 115
Технология Flash 555
Типографический элемент 116
Транзитный счет продавца 453

У

Условно-бесплатное программное обеспечение 91
Учетная запись OpenID 416

Ф

Файл
◊ robots.txt 327
◊ cookie 335
Фиксированный контент 300
Формат
◊ Flashtrak 551
◊ GIF 208
◊ JPEG 208
◊ PNG 208
◊ SVG 208
Фрагмент 252
◊ страницы 57
Фразовый элемент 121
Функция 479
◊ alert 472
◊ local() 187
◊ Number() 504
◊ setTimeout() 491
◊ toDateString() 477

Функция *(прод.)*
◊ url() 187
◊ write() 478

Ц

Цикл for 529

Ш

Шаблон страницы 294, 302
Шрифт без засечек 177

Э

Эластичные макеты 283
Элемент
◊ <a> 235
◊ <area> 250
◊ <b> 32, 117, 119, 142
◊ <body> 37
◊ <blockquote> 128
◊
 32, 122
◊ <cite> 130
◊ <code> 127
◊ <dd> 135
◊ <del> 143
◊ 130, 199, 267
◊ <dl> 135
◊ <dt> 135
◊ 142
◊ <embed> 536, 541, 554
◊ <figcaption> 221
◊ <figure> 221
◊ <font> 117
◊ <form> 459, 501
◊ <head> 37
◊ <h1> 120
◊ <html> 37
◊ <hr> 126
◊ <i> 33, 117, 142
◊ <iframe> 294
◊ <img> 45, 119, 204
◊ <input> 501
◊ <item> 546
◊ 42, 132, 134
◊ <link> 156
◊ <map> 250
◊ <mark> 143
◊ <meta> 316, 492
◊ <noscript> 475
◊ 132
◊ <p> 38, 120, 121
◊ <path> 546
◊ <playlist> 546
◊ <pre> 127
◊ <q> 129
◊ <samp> 127
◊ <script> 471
◊ <select> 502
◊ <small> 143
◊ 131, 524
◊ 142
◊ <style> 159
◊ <sub> 143
◊ <sup> 143
◊ <table> 137
◊ <td> 137
◊ <textarea> 502
◊ <th> 138
◊ <title> 38, 314
◊ <tr> 137
◊ <u> 117, 144
◊ 42, 134
◊ <video> 536, 556
Элемент-контейнер 32
Элемент-якорь 449
Элементы управления 500