bhv®

РОСТИСЛАВ ЧЕБЫКИН

РАЗРАБОТКА И ОФОРМЛЕНИЕ ТЕКСТОВОГО СОДЕРЖАНИЯ САЙТОВ

Более 100 практических советов и рекомендаций

Более 100 конкретных примеров

Более 50 распространенных ошибок разработчиков

Более 30 стандартов и официальных документов

МАСТЕР

Ростислав Чебыкин

РАЗРАБОТКА И ОФОРМЛЕНИЕ ТЕКСТОВОГО СОДЕРЖАНИЯ САЙТОВ

Санкт-Петербург

«БХВ-Петербург»

2004

УДК 681.3.06
ББК 32.973
Ч-34

Чебыкин Р. И.

Ч-34 Разработка и оформление текстового содержания сайтов. — СПб.: БХВ-Петербург, 2004. — 528 с.: ил.

ISBN 5-94157-396-0

Рассматриваются основные аспекты разработки текстового содержания сайтов, прежде всего русскоязычных. Излагаются правила и нормы синтаксического оформления и типографики русского текста, сведения о представлении текстовых данных в компьютере и кодировках символов. Изучается техника разметки и оформления текстового содержания веб-страниц средствами HTML и CSS. Обсуждаются принципы составления текста для сайтов, а также вопросы удобства, доступности, эффективности и привлекательности текстового содержания сайтов для пользователей.

Для веб-разработчиков

УДК 681.3.06
ББК 32.973

Группа подготовки издания:

Главный редактор	*Екатерина Кондукова*
Зам. гл. редактора	*Евгений Рыбаков*
Зав. редакцией	*Григорий Добин*
Компьютерная верстка	*Натальи Караваевой*
Корректор	*Виктория Пиотровская*
Дизайн серии	*Инна Тачина*
Дизайн обложки	*Игоря Цырульникова*
Зав. производством	*Николай Тверских*

Лицензия ИД № 02429 от 24.07.00. Подписано в печать 08.10.04.
Формат 70×100 1/16. Печать офсетная. Усл. печ. л. 42,57.
Тираж 3000 экз. Заказ №
"БХВ-Петербург", 190005, Санкт-Петербург, Измайловский пр., 29.

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

Отпечатано с готовых диапозитивов
в ГУП "Типография "Наука"
199034, Санкт-Петербург, 9 линия, 12

ISBN 5-94157-396-0

© Чебыкин Р. И., 2004
© Оформление, издательство "БХВ-Петербург", 2004

Содержание

Добро пожаловать! Предисловие 1

Summa rei. О чем эта книга 1
Читатель — половина писателя. Для кого эта книга 2
Этапы большого пути. Как устроена эта книга 3
Веселые картинки. Условные обозначения 5
Добавить в «Избранное». Поддержка этой книги в Интернете 5
За все хорошее. Благодарности 6

ЧАСТЬ I. УРОКИ СЛОВЕСНОСТИ. ТЕКСТ — ОСНОВНОЙ НОСИТЕЛЬ ИНФОРМАЦИИ В ИНТЕРНЕТЕ 7

Глава 1. Текст и Сеть. Почему при разработке сайтов необходимо уделять особенное внимание текстовому содержанию 9

1.1. Текст как он есть. Какие свойства отличают текст от других форм представления информации и делают его основным носителем информации в Интернете 10
1.1.1. Статичность 10
1.1.2. Дискретность 11
1.1.3. Линейность 13
1.1.4. Множество представлений 14
1.1.5. Удобство представления в компьютере 15
1.1.6. Текст и браузер 17
1.2. Параллельные миры. Чем сайты и печатные издания похожи друг на друга и чем они различаются 18
1.2.1. Материальное — информационное 19
1.2.2. Традиционное — новое 21
Гиперссылки 23
1.2.3. Линейное — иерархическое 23
1.2.4. Безответное — интерактивное 25
1.2.5. Управляемое — неуправляемое 26
1.2.6. Что же их объединяет? 29

Глава 2. От «А» до «Я». Важные особенности грамматики и типографики для сайтов на русском и английском языках 32

2.1. Что написано пером. Некоторые правила грамматики и типографики 33
2.1.1. Пробелы 34
2.1.2. Скобки и кавычки 35
2.1.3. Знаки препинания 37
Знаки препинания при скобках и кавычках 39
2.1.4. Дефис, тире, минус 40
2.1.5. Математические знаки 43
2.1.6. Числа 45
2.1.7. Специальные знаки 47
2.1.8. Сокращения 50
2.1.9. Переносы 54
2.1.10. Имена собственные, наименования 56
2.1.11. Заголовки и заголовочные надписи 62
2.1.12. Перечни (списки) 64
2.1.13. Выделения 66
2.2. Новое — значит лучшее? Современные явления в сетевом русском языке 68
2.2.1. «Браузер» против «обозревателя». Откуда берутся специальные термины, связанные с Интернетом 69
Норма 70
Традиция 70
Русскоязычный эквивалент 71
Транскрипция или транслитерация 72
Ориентация на аудиторию 73
Благозвучность 74
2.2.2. Админ пингует хост. О специальной терминологии и сетевом жаргоне 75
2.2.3. Знаки времени. Новшества сетевой типографики 78
Новые сокращения 78
Типографские причуды 80
Эмотиконы 81
Обсценная типографика 83
2.2.4. Смесь французского с нижегородским. О включении в текст на русском языке слов, записанных на других языках 84

ЧАСТЬ II. ДЕЛО МАСТЕРА БОИТСЯ. ТЕХНИКА РАЗРАБОТКИ И КОДИРОВАНИЯ ТЕКСТОВОГО СОДЕРЖАНИЯ САЙТОВ 89

Глава 3. Двоичная сюита. Представление текста в компьютере: кодировки символов, форматы текстовых файлов 91

3.1. Азбука для компьютера. Кодировки символов 91
3.1.1. Кодировки и кодовые таблицы 92

3.1.2. Двоичная и шестнадцатеричная системы счисления 95
3.1.3. ASCII 99
Перевод строки 105
Табуляция 107
3.1.4. Расширенные кодировки ASCII 110
Неразрывный пробел и мягкий перенос 114
3.1.5. Кириллица в расширенных кодировках ASCII 115
Проблемы перекодировки 122
3.1.6. Unicode — решение многих проблем 125
ISO 10646 и Unicode 126
UTF-8 и другие UTF 132
Трудности с использованием Unicode на практике 135
3.2. Лики текста. Форматы текстовых файлов 138
3.2.1. Двоичный и текстовый режимы 138
3.2.2. Обычный текст 143
3.2.3. Форматированный текст 150
RTF (Rich Text Format) 150
DOC (Документ Microsoft Word) 152
TEX 154
HTML (HyperText Markup Language) 156
PostScript 158
PDF (Portable Document Format) 159
3.3. При всем богатстве выбора. Программы для работы с текстовыми файлами 161
3.3.1. Текстовые редакторы 163
3.3.2. Текстовые процессоры 165
WYSIWYG 167
Среды разработки сайтов 169
3.3.3. Издательские системы 172

Глава 4. «Блокнот» мой — друг мой. Эффективная работа с текстовым редактором 175

4.1. Кто на свете всех милее? Как выбрать текстовый редактор 176
4.2. От частного к общему. Свойства и возможности текстовых редакторов 182
4.2.1. Текстовый и графический режимы 182
4.2.2. Курсор 186
4.2.3. Набор и редактирование текста 190
4.2.4. Выделение текста 195
4.2.5. Буфер обмена 198
4.2.6. Символы и шрифты 200
4.2.7. Прокрутка и переводы строк 205
4.2.8. Операции с файлами 207
4.2.9. Обработка текста 211

Глава 5. Наука побеждать. Техника разметки и оформления текста с помощью HTML и CSS 214

5.1. С чего начинается Родина. HTML — язык структурной разметки гипертекста 215
5.1.1. Как это понимать? Что означают слова «структурный», «разметка» и «гипертекст» 216
5.1.2. Во первых строках письма... Шаблон документа HTML 220
Пролог документа и версии HTML 221
Валидатор и стандарты W3C 227
Тэги, элементы, атрибуты 231
5.1.3. Уроки хороших манер. Стиль кодирования 234
Основные элементы 237
Вложенные элементы 240
Комментарии 243
Пробельные символы 245
Имена файлов HTML 246
5.1.4. Строительный материал. Структурные элементы HTML 248
Строчные и блочные элементы 248
title — самый главный элемент 250
Заголовки: *h1* — *h6* 256
Текст: *p* и *pre* 260
Внутритекстовые выделения: *em* и *strong* 263
Другие внутритекстовые выделения 265
Блочные цитаты: *blockquote* 268
Перечни: *ul*, *ol*, *dl* 270
Таблицы 278
Гиперссылки: элемент *a* 284
Изображения и их текстовые эквиваленты 296
5.1.5. Полезные мелочи. Подстановки символов 301
Обязательные подстановки 306
Неразрывный пробел 307
Специальное использование подстановок 309
5.1.6. Что дальше? Перспективы развития HTML и XHTML 312
5.2. Красота требует жертв. Оформление текстового содержания веб-страниц с помощью CSS 314
5.2.1. Три кита. Содержание, структура, представление 315
5.2.2. Курс молодого бойца. Основы кодирования таблиц стилей 317
Начальные сведения 317
Замечания о кодировании CSS 320
Виды селекторов CSS 324
Преимущества и недостатки CSS 327
5.2.3. Чтоб картинка ожила. Свойства CSS для оформления текстового содержания веб-страницы 329
Свойства шрифта 330
Свойства блоков текста 344

ЧАСТЬ III. ВСЕ ДЛЯ ЧЕЛОВЕКА. КАК ДОБИТЬСЯ, ЧТОБЫ ТЕКСТОВОЕ СОДЕРЖАНИЕ САЙТА СООТВЕТСТВОВАЛО ИНТЕРЕСАМ И ЗАДАЧАМ ПОЛЬЗОВАТЕЛЕЙ 363

Глава 6. Ради красного словца. Как составлять текст для размещения на сайте 365

6.1. Доброе слово и кошке приятно. Как составить текст, производящий на пользователя нужное впечатление 366
6.1.1. Совершенство малых форм. Самые короткие надписи 366
6.1.2. Туда-сюда. Особенности составления текста для навигации 371
6.1.3. Короче! Соблюдайте краткость в тексте на сайтах 377
6.1.4. Проще простого. Пишите простой текст 382
6.1.5. Сверим наши стили. Придерживайтесь единого стиля 386
6.1.6. И смех и грех. Не злоупотребляйте метафорами и юмором 388
6.1.7. Честно говоря. Размещайте на сайте только достоверную информацию 391
6.1.8. Внесите разнообразие! Не пренебрегайте средствами оформления 394
6.1.9. Заставьте текст работать! Применяйте элементы, с которыми пользователь может взаимодействовать 396
6.1.10. В особо крупных размерах. Что делать с крупными текстами 399
6.1.11. Вы ничего не забыли? Еще несколько распространенных ошибок 403
Время и дата 403
Контактная информация 404
Under construction 405
6.2. Не забывайте про роботов. Как оптимизировать текстовое содержание сайта для поисковых систем 406
6.2.1. Облик Искателей. Общая информация о поисковых системах 406
6.2.2. Разметка превыше всего. Грамотно употребляйте структурную разметку 409
6.2.3. Чтоб язык не заплетался. Учитывайте особенности русского языка 412
6.2.4. Хитрости полезные и вредные. Как еще повысить «находимость» вашего сайта 415

Глава 7. Пользователь всегда прав. Удобство, доступность, эффективность и привлекательность текстового содержания сайта 419

7.1. Чтоб никто не ушел обиженным. Основные принципы удобства текстового содержания 422

7.1.1. Привычка — вторая натура. Учитывайте естественное поведение и привычки пользователей ... 422
7.1.2. Лень — двигатель прогресса. Пользователь не должен прилагать лишних усилий ... 428
Копирование, сохранение и печать текста ... 437
7.1.3. Доверяй, но проверяй. О тестировании удобства сайта ... 442
Редактор моей мечты. Поиск подходящего текстового редактора на сайте *www.listsoft.com* ... 445
Belle! Поиск нот популярного трио из «Notre-Dame de Paris» ... 452
7.1.4. Как вам будет угодно. Персонализация сайтов ... 463
7.2. Это доступно? Критерии доступности текстового содержания сайтов ... 466
7.2.1. Найдите десять отличий. «Машинный фактор» доступности ... 467
Устройства просмотра ... 467
Операционная система ... 469
Видеорежим ... 474
Браузер ... 476
7.2.2. Esse homo. «Человеческий фактор» доступности ... 485
7.2.3. Скрижали Закона. Принципы доступности сайтов от Консорциума W3C ... 489
Стандарты ... 489
Структура информации ... 491
Разделение структуры и представления ... 493
Оформление ... 496
Персонализация ... 497
Степени соответствия WCAG 1.0 ... 498
7.3. ...И люди к вам потянутся. Замечания о привлекательности сайта ... 499
7.3.1. Содержание ... 499
7.3.2. Структура и навигация ... 501
7.3.3. Оформление ... 503

Пора и честь знать. Заключение ... 507

Смотрит в книгу — видит... Рекомендуемая литература ... 513

Орфография, пунктуация и стилистика русского языка ... 513
Издательское дело ... 514
Разработка сайтов ... 517
Ресурсы Всемирной Паутины ... 518
Стандарты и документация ... 518
Другие ресурсы ... 519
Браузеры ... 519

Добро пожаловать!
Предисловие

Когда выходишь на эстраду,
Стремиться нужно к одному:
Всем рассказать немедля надо,
Кто ты, зачем и почему.

Л. Зорин. Покровские ворота

1.7.14. An abstract **should** briefly describe the document. The word “Abstract” **must** be marked up with an `h2` element.

World Wide Web Consortium Publication Rules[1]

Внимание!

Эта книга посвящена всем аспектам разработки текстового содержания сайтов в Интернете, предназначена для составителей сетевого текста, редакторов, корректоров и кодировщиков, а также может быть интересна всем, чья деятельность связана с созданием сайтов.

Страница поддержки книги в Интернете — **www.philigon.ru/books/webtext**.

Summa rei[2]
О чем эта книга

— Расскажите кратенько, в двух словах, о чем ваша книга.
— Эх, если бы я мог рассказать об этом в двух словах, я бы не стал писать целую книгу.

Литературный анекдот

Эта книга — обо всем, что связано с разработкой текстового содержания сайтов в Интернете.

---

[1] 1.7.14. В обзоре следует кратко описывать документ. Слово «Обзор» должно быть размечено элементом `h2`. *(Правила публикаций Консорциума W3C)*

[2] Суть дела *(лат.)*.

Вот наиболее важные темы книги:

- чем сайты и печатные издания похожи и чем различаются; какие практические выводы следуют из этих сходств и различий;
- какие правила русской и английской грамматики и типографики особенно важно учитывать при разработке сайтов;
- что нужно знать разработчику сайтов о технической стороне представления текста в компьютере, о кодировках и форматах текстовых файлов;
- как сэкономить время и повысить эффективность своей работы с текстовым редактором и другими программами, используемыми при разработке сайтов;
- как правильно и грамотно применять языки HTML и CSS для структурной разметки и оформления текстового содержания сайтов;
- как составлять текст, который сделает сайт эффективнее и поможет привлечь и удержать пользователей;
- как улучшить удобство и доступность текстового содержания сайта.

Читатель — половина писателя

Для кого эта книга

> Не давайте святыни псам и не бросайте жемчуга вашего перед свиньями, чтоб они не попрали его ногами своими и, обратившись, не растерзали вас.
>
> *Евангелие от Матфея, гл. 7, ст. 6*

Книга, в первую очередь, предназначена для двух категорий читателей.

Во-первых, для всех, кто разрабатывает или собирается разрабатывать текстовое содержание сайтов:

- авторам сетевого текста, редакторам, корректорам и верстальщикам сайтов она поможет работать более продуктивно, избегать досадных ошибок и делать сайты привлекательнее для пользователей;
- всем, кто учится созданию сайтов, — даст представление о разработке текстового содержания сайтов и позволит принять осознанное решение, стоит ли делать это занятие областью своей специализации.

Во-вторых, эта книга для всех, кому приходится оценивать качество сайтов:

- руководителям сетевых проектов она поможет организовывать деятельность, связанную с разработкой текстов, а также контролировать работу авторов, редакторов и верстальщиков;
- экспертам в области удобства и доступности сайтов — оценивать удобство и доступность текстового содержания.

Автор надеется, что читатели знакомы с основами компьютерной грамотности, что они хотя бы в общих чертах представляют себе, что такое Интернет, и имеют опыт посещения сайтов. Для чтения этой книги больше ничего не потребуется. Хотя более понятна она будет для тех, кто уже занимался разработкой сайтов, — хотя бы на любительском уровне — и имеет представление о HTML и CSS.

Примечание

Ростислав Чебыкин (р. 1977) — преподаватель информационных технологий в Экономическом бизнес-колледже (Москва) и международной школе бизнеса «Орион-Интер». Эта книга основана на одном из разделов курса «Веб-мастер», который автор ведет в течение нескольких лет. Персональный сайт автора — **www.philigon.ru**. Адрес электронной почты: **philigon@master.elserv.msk.su**.

Этапы большого пути

Как устроена эта книга

> Рубрикация — это система заголовков издания и возглавляемых ими подразделов (рубрик), выражающая логическую или иную связь и соподчиненность обозначаемых ими частей текста (произведений, разделов, глав, параграфов).
>
> *А. Э. Мильчин, Л. К. Чельцова. Справочник издателя и автора*

Книга состоит из трех частей.

Часть 1 — «Уроки словесности» — это теоретический базис последующего изложения. Здесь читатель узнает:

- что такое текст, каковы его характерные особенности в сравнении с другими формами представления информации;
- почему текст является основным носителем информации в Интернете;
- какие сходства и различия между сайтами и печатными изданиями должен учитывать разработчик;
- как избегать наиболее распространенных грамматических ошибок в сетевом тексте (прописные и строчные буквы, сокращения, заголовки и т. д.);
- как грамотно использовать в тексте специальные типографские символы (тире, кавычки, пробелы, многоточие и т. д.);

- стоит ли при составлении текста для сайтов употреблять текстовые новшества (смайлики, жаргон, иностранные слова и т. д.).

Часть 2 — «Дело мастера боится» — посвящена практической стороне разработки текстового содержания сайтов. Здесь рассказывается:

- как текст хранится и преобразуется в компьютере, какие программные средства применяются для работы с ним;
- что такое кодировки символов и зачем разработчику сайтов нужно в них разбираться;
- как повысить эффективность своей работы с текстовым редактором и другими программами разработки;
- как правильно использовать язык HTML для структурной разметки текстового содержания;
- как оформлять текстовые элементы сайта с помощью языка CSS.

Часть 3 — «Все для человека» — касается вопросов удобства и доступности текстового содержания сайта для пользователей. Здесь освещены следующие темы:

- как составлять эффективный текст, предназначенный для размещения на сайте;
- как сделать так, чтобы пользователям было удобно работать с текстом на сайте;
- как повысить доступность текстового содержания для различных категорий посетителей, включая пользователей с ограниченными возможностями;
- как текстовое содержание поможет привлечь пользователей и удержать их на сайте;
- как обеспечить единство текста с другими компонентами сайта.

Книга построена как учебник, но ее можно использовать и как справочник. Чтобы читатель мог легко найти нужную информацию, в конце книги есть предметный указатель.

Примечание

Основная часть книги написана в середине 2004 года, и в ней отражена практика, сложившаяся к этому моменту. Читая книгу, не забывайте, что жизнь не стоит на месте.

Веселые картинки
Условные обозначения

Иероглифы представляют собой идеограммы, то есть обозначают целые понятия. Такая форма фиксации информации оказала значительное влияние на характер мышления людей, способствовала формированию структурно-образного восприятия.

Япония от «А» до «Я»

Такой символ (U+26A0) обозначает краткий конспект последующего текста или наиболее важные выводы, на которые рекомендуется обратить особое внимание. Если вам не хочется или некогда читать книгу целиком — прочтите хотя бы то, что помечено этим символом.

Таким символом (U+2611) обозначаются полезные замечания и отступления. Иногда считают, будто подобные замечания появляются потому, что автор очень хотел что-то сказать, но никак не смог уместить это в основном тексте. А иногда оказывается, что без замечаний основной текст не имеет никакого смысла.

Такой символ (U+7EB8, «китайская грамота») обозначает необязательный материал, который можно пропустить при первом чтении книги. Возможно, его даже нужно пропускать: ведь тогда появится стимул перечитать книгу снова!

Добавить в «Избранное»
Поддержка этой книги в Интернете

Один из недостатков обычного издания книги в том, что невозможно мгновенно перепечатать и распространить книгу заново, если появляется необходимость исправить ошибки или внести какие-либо технологические изменения. В Сети же, наоборот, можно мгновенно изменить любую информацию. Для этого я создал сайт, на котором публикую все исправления.

Я. Нильсен. Веб-дизайн

Этой книге посвящена страница **www.philigon.ru/books/webtext**. Там находится список обнаруженных ошибок и опечаток, избранные фрагменты книги, рекомендации для преподавателей и некоторая другая информация.

Примечание

Один из недостатков размещения информации в Интернете — нестабильность и недолговечность. Может быть, сейчас, когда вы держите в руках эту книгу, давно уже нет ни автора, ни его сайта, ни самого Интернета — так что ссылка на страницу поддержки неактуальна. Поэтому автору остается только обещать позаботиться о том, чтобы эта ссылка проработала как можно дольше и чтобы ее исчезновение (которое, увы, рано или поздно наступит) не принесло никому неудобства.

За все хорошее

Благодарности

Ложки у меня пациенты много раз глотали, не скрою, но вот чтобы так — за обедом на десерт и острый предмет — замечательно. За это вам наша искренняя сердечная благодарность.

Г. Горин. Формула любви

Автор благодарен многим людям, которые, сами того не подозревая, оказали неоценимую помощь в работе над этой книгой. Вот лишь некоторые из них: Тим Бернерс-Ли, Людвиг ван Бетховен, Дмитрий Гайдук, Дональд Кнут, Льюис Кэрролл, Леонид Лиходеев, Якоб Нильсен, Чарльз Петцольд, Александр Привалов, Сергей Рахманинов, Владимир Успенский, Джеймс Феличи, Франциск Ассизский, Даниил Хармс, Даглас Хофштадтер, Ян Чихольд, Морис Эшер.

```
<?xml version= 1.0
<!DOCTYPE html PUBL
  "-//W3C//DTD XHTM
  "http://www.w3.or
<html xmlns="http:/
  lang="ru" xml:lan
<!-- Редактировал:
<head>
  <title>Евгений Он
  <link rel="styles
    href="index.css
</head>
```

Часть I

Уроки словесности

Текст — основной носитель информации в Интернете

— Что ж, начинать? — Начнем, пожалуй.

А. С. Пушкин. Евгений Онегин

ТЕКСТ (от лат. textus — ткань, соединение),

1) последовательность предложений, слов (в семиотике — знаков), построенная согласно правилам данного языка, данной знаковой системы и образующая сообщение.

2) Словесное произведение; в художественной литературе — законченное произведение либо его фрагмент, составленный из знаков естественного языка (слов) и сложных эстетических знаков (слагаемых поэтического языка, сюжета, композиции и т. д.).

3) Авторское сочинение без комментариев и приложений к нему.

4) В издании — часть произведения, выраженная в словесной форме, в отличие от рисунков, чертежей, формул и т. п.

5) Типографский шрифт, кегль (размер) которого равен 20 пунктам (около 7,5 мм).

Российский энциклопедический словарь

```
<?xml version= 1.0
<!DOCTYPE html PUBL
  "-//W3C//DTD XHTM
  "http://www.w3.or
<html xmlns="http:/
  lang="ru" xml:lan
<!-- Редактировал:
<head>
  <title>Евгений Он
  <link rel="styles
    href="index.css
</head>
```

Глава 1

Текст и Сеть

Почему при разработке сайтов необходимо уделять особенное внимание текстовому содержанию

Если рассматривать «текст» в самом широком смысле этого слова, то, вероятно, 90 % всего, что мы делаем, на 90 % состоит из обработки текста.

Л. Уолл, Т. Кристиансен, Дж. Орвант. Программирование на Perl

Сейчас мы знаем, что текст состоит не из линии слов, воплощающих единственное «теологическое» значение («сообщение» Автора-Бога), но из многомерного пространства, где обручаются и вступают в распрю всевозможные виды письма, из которых ни один не является изначальным: текст есть производство цитаций, исходящих из тысяч источников культуры.

Р. Барт. Смерть автора

Мы не можем представить себе сайт без текста. Бывают сайты без иллюстраций, без художественного оформления, без специальных объектов вроде фоновой музыки или анимационных роликов, но сайт без текста — это нонсенс. Текст — основная форма представления информации в Интернете, поэтому его разработка заслуживает самого пристального внимания.

1.1. Текст как он есть

Какие свойства отличают текст от других форм представления информации и делают его основным носителем информации в Интернете

Иными словами, откажемся пока от представления о тексте как об отформатированных абзацах на двумерной печатной странице. Будем считать его одномерным потоком букв, цифр, знаков препинания и, возможно, дополнительных символов, указывающих на окончание абзаца.

Ч. Петцольд. Код

Внимание!

Основные свойства текста:

- статичность;
- дискретность;
- линейность;
- множество представлений;
- удобство представления в компьютере.

1.1.1. Статичность

Статичность — это так прекрасно! Так непостижимо величественно!

А. Поникаровская. Промежуточный этаж

Статичность текста означает, что он *не развивается во времени*. Это просто понять, если сравнить текст с музыкальной композицией или видеозаписью. Такие формы представления информации имеют определенную временну́ю продолжительность. А читатель текста сам определяет, с какой скоростью он будет его читать и сколько времени потратит на чтение.

Статичность — одна из причин того, почему многим людям легче читать текст на иностранном языке, чем слушать этот же текст в устном изложении.

Статичность текста нарушается, если он включается в титры, бегущую строку или другие динамические элементы. Это является вынужденной мерой в кино или на рекламных конструкциях, но на сайтах нарушение статичности текста чаще всего ничем не оправдано.

Внимание!

Разрабатывая сайт, не нарушайте статичности текста. Пусть текст на веб-странице будет неподвижным. Не надо бегущих строк, мигающих букв и прочих динамических эффектов. Не мешайте пользователю читать текст с такой скоростью, с какой ему удобнее.

1.1.2. Дискретность

> Атомы Демокрита различаются по форме и по величине; двигаясь в пустоте, они соединяются («сцепляются») между собой в силу различия по форме: у Демокрита есть атомы круглые, пирамидальные, кривые, заостренные, даже «с крючками». Так из них образуются тела, доступные нашему восприятию.
>
> *И. Фролов. Введение в философию*

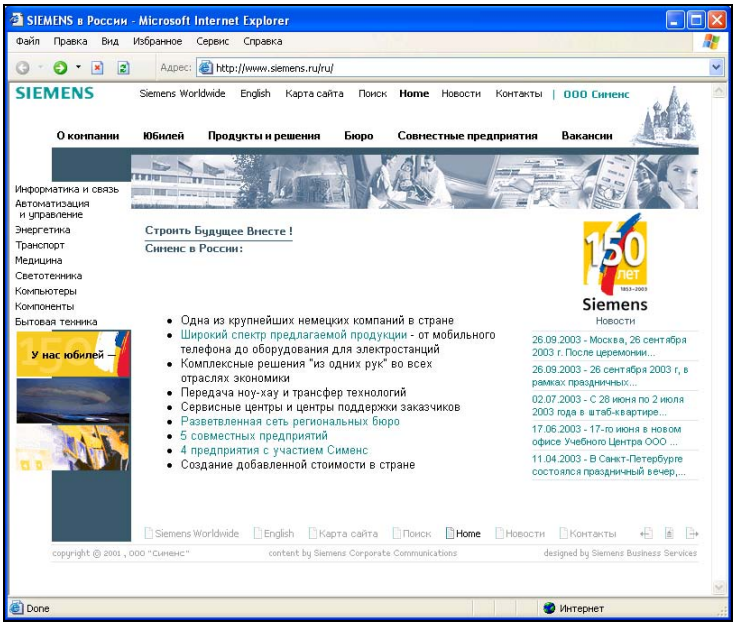
Дискретность текста означает, что он состоит из отдельных элементов — *символов*, каждый из которых рассматривается как единое целое. Этим свойством не обладают, например, изображение или звуковая запись — они являются непрерывными.

Дискретность текста на веб-странице нарушается, если он представлен не в виде последовательности символов, а в виде изображения — например, графического заголовка или панели навигации (рис. 1.1, *а*). Это вызывает сразу несколько неприятных явлений (рис. 1.1, *б*):

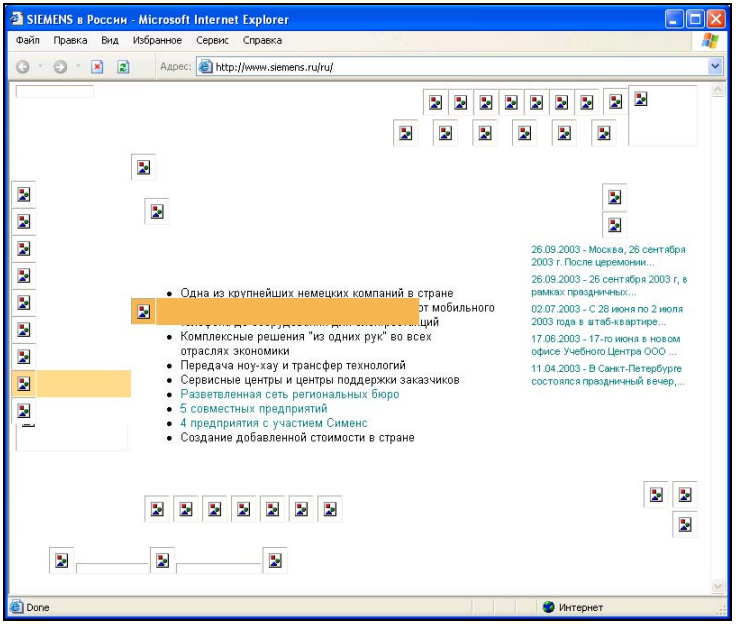
- если в браузере пользователя отключены или не поддерживаются изображения, он не сможет прочитать такой текст;
- если этот текст слишком мелкий или слишком крупный для пользователя, он не сможет увеличить или уменьшить его стандартными средствами браузера;
- если пользователю потребуется сохранить у себя такой текст, он не сможет просто скопировать его в буфер обмена и вставить в свой документ; текст придется перепечатывать вручную;
- поисковые машины не распознают текст, заключенный в изображение, и не индексируют его.

Внимание!

Не нарушайте дискретности текста. Убедитесь, что вся текстовая информация, которая есть на вашем сайте (включая заголовки и средства навигации), представлена именно в форме текста, а не графики.



а



б

Рис. 1.1. www.siemens.ru: если в браузере отключена загрузка изображений, панель навигации на нем становится недоступной. Надо сказать, что таких сайтов с графической навигацией со временем остается все меньше. Еще два года назад этим недостатком страдал почти каждый второй сайт, а теперь автору пришлось некоторое время потрудиться в поисках

Примечание

Все правила и рекомендации, которые приводятся в этой книге, носят самый общий характер. На практике иногда встречаются ситуации, когда правила можно и даже нужно нарушать. Но делать это следует, только если вы хорошо представляете себе, к каким неприятностям приведет нарушение правил, и твердо уверены, что в данном случае эти неприятности допустимы.

Дополнительно

Дискретность текста проявляется на нескольких уровнях. Достаточно опытный читатель воспринимает текст не по отдельным буквам, а целыми блоками. Обычно такими блоками являются слова, знакомые словосочетания и даже целые фразы. Читатель «схватывает» слово или словосочетание целиком, мысленно не разделяя его на более мелкие составляющие. Если в тексте много незнакомых сложных слов, используются громоздкие грамматические и стилистические конструкции, — это затрудняет чтение, нарушая дискретность на данном уровне. Не допускайте таких ошибок, составляя текст для своего сайта.

1.1.3. Линейность

> Наиболее простой способ объединить или связать некоторое множество элементов — это «вытянуть их в линию», организовать список или очередь.
>
> *Н. Вирт. Алгоритмы и структуры данных*

Линейность текста означает, что он представляет собой *одномерную* структуру, последовательность символов. Символы идут строго один за другим, что отличает текст от изображения (двумерная структура) или скульптуры (трехмерная структура).

Именно линейностью объясняется тот факт, что люди научились передавать текстовые сообщения по линиям связи гораздо раньше, чем информацию любого другого характера. Флажковая азбука, азбука Морзе, язык тамтамов — все это средства передачи линейной, последовательной информации.

Если текст заключается в изображение, нарушается не только его дискретность, но и линейность. Это еще одна причина не делать «графического» текста.

1.1.4. Множество представлений

> Когда женщина стареет, с ней могут произойти многие неприятности: могут выпасть зубы, поседеть и поредеть волосы, развиться одышка, может нагрянуть тучность, может одолеть крайняя худоба, но голос у нее не изменится. Он останется таким же, каким был у нее гимназисткой, невестой или любовницей молодого повесы.
>
> *И. Ильф, Е. Петров. Двенадцать стульев*

Один и тот же текст может быть представлен множеством разных способов. Например:

Мой дядя самых честных правил.

Мой дядя самых честных правил.

`Moy djadja samykh chestnykh pravil.`

Несмотря на то, что внешне эти строчки заметно отличаются друг от друга (особенно третья), мы видим, что в них записан один и тот же текст. Это позволяет рассматривать сам текст отдельно от его представления, внешнего вида.

Дополнительно

Например, вы переписываетесь с кем-то по электронной почте. В вашей почтовой программе может быть установлен шрифт Andale Mono, потому что он кажется вам наиболее приятным для глаз и удобным при наборе. А почтовая программа вашего корреспондента настроена так, чтобы отображать сообщения шрифтом Helvetica — потому что так, в свою очередь, ему удобнее читать. Это не вызывает никаких проблем — вы набираете сообщение тем шрифтом, каким удобнее вам, ваш корреспондент читает его тем шрифтом, каким удобнее ему, — а текст при этом остается тем же самым, и оба участника переписки довольны.

Пользователи Интернета, в отличие от читателей печатных изданий, имеют счастливую возможность настраивать внешний вид текста на веб-страницах таким образом, чтобы им было удобнее читать. Настройки браузеров позволяют установить для текста определенный шрифт, цвет, размер и другие параметры.

Внимание!

Не мешайте пользователям настраивать «под себя» внешний вид текста на сайтах. Убедитесь, что на вашем сайте срабатывают все браузерные средства настройки внешнего вида.

Это, кстати, еще один аргумент против «графического» текста. Если текст заключен в изображение, то пользователь не сможет изменить его параметры.

Дополнительно

У текста немало и других свойств, отличающих его от прочих форм представления информации. Например, текст обладает *избыточностью*. Это легко продемонстрировать на следующем примере:

Слв «ткст» знчт ткнь, сцплн, сднн… В лбм слч ткст прдствлт сбй бъднннп смсл пслдвтльнст знквх днц, снвнм свйствм ктрй влтс свзннсть, цльнсть.[1]

Из этого текста удалены все гласные буквы — но его все равно можно понять и восстановить. Это и означает, что полная версия текста — избыточна.

Избыточность текста приводит к двум важным качествам:

- Текст обладает хорошей устойчивостью по отношению к потерям, помехам и искажениям, поэтому его проще передавать по линиям связи. Как видно из приведенного примера, даже если из текста при передаче пропала половина букв, он, тем не менее, поддается восстановлению. А представьте, что из музыкальной мелодии пропала половина нот — как вы будете ее восстанавливать?
- Текст хорошо сжимается без потерь. Программы компрессии иногда позволяют сжать текст в 10 раз, что для других форм представления информации невозможно. Сжатый текст занимает меньше места в устройствах хранения и, что еще более важно, быстрее передается по линиям связи.

1.1.5. Удобство представления в компьютере

Другой аспект, который является общим почти для всех приложений, заключается в том, что при их использовании требуется вводить какой-то текст. (Даже в играх вам иногда приходится вводить собственное имя в случае выигрыша.)

Дж. Раскин. Интерфейс

Внимание!

Удобство представления текста в компьютере:

- простота кодирования;
- точность представления;
- удобство хранения на носителях и передачи по каналам связи;
- удобство операций с текстом.

[1] Н. С. Валгина. Теория текста. М., 2003. С. 12. — *Авт.*

Обладая перечисленными свойствами, текст является самой удобной формой информации для представления в компьютере. Это связано со следующими факторами:

простотой кодирования — из-за дискретности текста его очень просто преобразовывать в двоичный код и обратно: достаточно поставить каждому символу в соответствие код из определенной кодовой таблицы и пользоваться этой таблицей при преобразованиях.

Это даже проще, чем представление числовой информации. Чтобы сохранить в компьютере число, над ним в большинстве случаев приходится производить вспомогательные арифметические операции (например, преобразование в дополнительный код). Кодирование изображений и звуковой информации происходит еще сложнее;

- **точностью представления**. Текст — единственная форма информации, которая может быть сохранена в компьютере с абсолютной точностью. Если снова вспомнить про числа, то даже такое незамысловатое число, как ⅓, при сохранении в компьютере придется округлить до некоторого конечного количества знаков после запятой, а при сохранении изображений и звуковой информации потери еще значительнее. При сохранении текста не теряется *ничего*;
- **удобством хранения на носителях и передачи по каналам связи** — чтобы сохранить информацию в оперативной памяти компьютера, на жестком диске или на любом другом носителе или чтобы передать ее (как между устройствами внутри одного компьютера, так и по линиям связи между разными компьютерами), она должна быть преобразована в линейную последовательность байтов. А текст не надо никуда преобразовывать, он и так линейный;
- **удобством операций с текстом** — обрабатывать текст на компьютере гораздо проще, чем любую другую информацию. Например, из текста легко можно удалить отдельный символ, слово или целое предложение, и текст после этого будет выглядеть так, как будто этого удаленного фрагмента вообще не было. А, скажем, удалить фигуру человека из групповой фотографии так, чтобы фотография после этого казалась нетронутой, — непростая задача, которая иногда не под силу даже профессионалам компьютерной графики;
- **другими факторами** — кроме перечисленных, можно назвать еще множество факторов удобства представления текста в компьютере. Например, работа с текстом требует гораздо меньше вычислительных ресурсов, чем работа с другими формами представления информации — скажем, с изображениями или со звуком, не говоря уже о видео. Текст предъявляет меньше требований к квалификации разработчика. В самом деле, вы легко можете в точности перепечатать на компьютере тот самый текст, который вы сейчас читаете. А можете ли вы так же легко в точности срисовать рис. 1.1?

Дополнительно

У текста есть и недостатки по сравнению с другими формами представления информации. Например, чтобы прочесть текст, нужно знать язык, на котором он написан. Изображения или музыкальные записи не имеют этого недостатка.

С этим связаны споры о том, целесообразно ли использовать на сайтах пиктограммы: например, символическое изображение домика в качестве ссылки на главную страницу или конверта в качестве ссылки на адрес электронной почты. Сторонники пиктограмм утверждают, что такие картинки будут понятны даже тем пользователям, которые не могут прочесть текст на сайте. Противники пиктограмм считают, что они сбивают с толку: например, пользователь может подумать, что изображение домика ведет на страницу, посвященную строительству или домашнему хозяйству.

Впрочем, здесь нетрудно достичь компромисса: можно помещать на страницах пиктограммы и сопровождать их поясняющим текстом.

Примечание

Человеку свойственно ошибаться, и авторы книг — не исключение из этого правила. Какой бы хорошей ни была книга, в ней всегда найдутся какие-то недостатки, всегда обнаружится, что можно исправить и улучшить. Если вы считаете, что автор этой книги в чем-то неправ, — что же, это вполне возможно. Пожалуйста, в этом случае напишите автору по адресу: **philigon@master.elserv.msk.su**. Ведь автор сможет исправить свои ошибки, только если узнает о них!

1.1.6. Текст и браузер

> This guideline emphasizes the importance of providing text equivalents of non-text content (images, pre-recorded audio, video). The power of text equivalents lies in their capacity to be rendered in ways that are accessible to people from various disability groups using a variety of technologies.
>
> *Web Content Accessibility Guidelines 1.0*[2]

Существует огромное количество разнообразных браузеров, с помощью которых пользователи просматривают сайты. В этих браузерах могут не под-

[2] Этот принцип говорит о том, как важно предоставлять текстовые эквиваленты для нетекстового содержания (изображений, аудио- и видеозаписей). Преимущество текстовых эквивалентов в том, что они могут отображаться теми способами, которые доступны для пользователей с ограниченными возможностями, применяющих самые разные технологии доступа к сайтам. *(Принципы доступности содержания сайтов, редакция 1.0)*

держиваться или быть отключены изображения, анимационные ролики, звук, JavaScript, апплеты, таблицы стилей и все остальное. Но текст — это единственное, что поддерживается всегда и в любом браузере.

Таким образом, вы как разработчик можете быть уверены, что ваш текст в точности дойдет до пользователя, независимо от того, чем он смотрит ваш сайт. А для других составляющих сайта этого гарантировать нельзя.

Поэтому все нетекстовые компоненты информационного содержания сайта (например, изображения, анимационные ролики или апплеты) рекомендуется сопровождать альтернативным текстом, который должен увидеть пользователь, если сам нетекстовый объект будет для него недоступен.

1.2. Параллельные миры

Чем сайты и печатные издания похожи друг на друга и чем они различаются

...И плотная сеть паутины
Упала, окутала ложе...

А. Ахматова

...Можно прийти к выводу, что технологии производства аппаратного обеспечения — это сдерживающий фактор, и что только к 2007 году информация, доступная через Сеть, целиком заменит обычные книги. Издатели, будьте бдительны! Это точно произойдет.

Я. Нильсен. Веб-дизайн

Есть мнение, что между размещением информации на сайте и публикацией в печатных изданиях нет никакой принципиальной разницы: просто в одном случае пользователь просматривает документ на экране компьютера, а в другом — на листе бумаги. Поэтому все принципы и правила, относящиеся к традиционной печати, справедливы и в Сети. Но есть и противоположное мнение: Сеть — это принципиально новый носитель информации, поэтому к ней неприменимы принципы бумажных публикаций.

Как это часто бывает, истина находится где-то посередине. Чтобы добиться успеха в создании сайтов, нужно хорошо представлять себе, чем они похожи на печатные издания и чем отличаются.

Внимание!

Печатное издание	Сайт
Материальный объект	Информационный объект
Первые книги — VIII век до н. э.	Первые сайты — конец 1980-х
Линейная структура	Иерархическая структура
Статичное восприятие читателем	Интерактивное взаимодействие с пользователем
Полностью контролируется издателем	В существенной степени контролируется пользователем

1.2.1. Материальное — информационное

> Я люблю чувствовать, осязать, а в Интернете я не чувствую... Вот вас я могу потрогать, а Интернет — не могу.
>
> *Б. Гребенщиков (из интервью)*

Печатные издания. Каждый экземпляр печатного издания представляет собой материальный объект: его можно потрогать, взять в руки, перенести с места на место, — наконец, уничтожить.

Сайты. Сайт — информационный объект. Это значит, что он не связан ни с каким определенным материальным носителем так прочно, как печатный текст. Работоспособность одного сайта может обеспечиваться десятками и даже сотнями различных серверов, разбросанных по всему миру. Одновременно с этим сайт существует в виде электрических сигналов, передаваемых по многочисленным линиям связи, а также в виде страниц, которые отображаются на компьютерах конечных пользователей. Сайт — это скорее не физический предмет, а служба, услуга. В этом смысле сайт похож не на печатные издания, а на такие способы распространения информации, как радио или телевидение.

Сайты обладают всеми свойствами информационных объектов:

- **легко распространяются** — как только веб-страница появляется на общедоступном сервере, ее в ту же минуту можно загрузить на миллионах компьютеров по всему земному шару. Количество просмотров страницы практически не ограничено;
- **легко копируются** — вы всегда можете сохранить веб-страницу на дискете и передать ее своему приятелю, у которого нет доступа в Интернет. Можно гарантировать, что ваш приятель получит в точности ту же страницу, которая до этого была у вас;

Дополнительно

В некоторых источниках можно встретить такое определение: «Если в процессе взаимодействия двух объектов один объект передает другому некую субстанцию, а сам при этом этой субстанции не лишается, — то такое взаимодействие называется информационным, а эта субстанция — информацией». Далеко не все специалисты соглашаются с таким определением, но оно, по крайней мере, дает общее представление о том, чем информация принципиально отличается от материи. В самом деле — если вы отдадите своему приятелю книгу (материальный объект), у вас этой книги не станет. А если вы расскажете ему, в каком магазине эта книга продается (то есть передадите информационный объект), то сведения об этом магазине появятся у вашего приятеля, но и у вас они тоже сохранятся.

- **легко изменяются** — если разработчик захочет исправить обнаруженную на сайте ошибку или внести дополнения, — он легко сможет это сделать, и все пользователи сайта тут же увидят изменения. А можно ли поступить так же, если хочется внести изменения в уже вышедшую книгу?

С другой стороны, сайтам присущи и недостатки информационных объектов.

Дополнительно

Чтобы прочесть книгу или газету, вам не требуется ничего, кроме самой книги или газеты. А чтобы получить доступ к информационному объекту (телепрограмме, радиопередаче, сайту), нужны как минимум три технических устройства:

- **источник** — телепрограммы и радиопередачи транслируются из специально оборудованной студии. Аналогично, для сайта нужен веб-сервер, который обеспечивает его функционирование;
- **приемник** — чтобы просмотреть телепрограмму или прослушать радиопередачу, нужно включить телевизор или радиоприемник. Аналогично, чтобы получить доступ к сайту, нужно устройство, предназначенное для просмотра сайтов (например, персональный компьютер с установленным на нем браузером);
- **линия связи** — наконец, чтобы информационный объект попал от источника к приемнику, между ними должна быть проведена линия связи. Теле- и радиосигналы распространяются по сложной системе кабелей, передаются по воздуху, проходят через ретрансляционные узлы и другие устройства. Аналогично, данные от веб-сервера следуют к приемнику через разнообразные коммуникации — в том числе через те же, которые используются для теле- и радиовещания, телефонной связи и других сигналов.

Еще один недостаток — изменчивость и непостоянство, характерные для Интернета. Сайты появляются и исчезают, меняют свои адреса, содержание и интерфейс. Пользователь не может быть уверен в том, что завтра он посетит свой любимый сайт так же, как делал это вчера и сегодня.

В Сети полно «мертвых» ссылок, ведущих на давно исчезнувшие или изменившиеся сайты. Но особенно нелепо выглядят ссылки на сетевые ресурсы в печатных изданиях, тем более в тех, которые рассчитаны на долгую жизнь: в энциклопедиях, учебниках, словарях. «Мертвую» ссылку в энциклопедии нельзя исправить, и она будет сбивать с толку не одно поколение читателей.

1.2.2. Традиционное — новое

> Тогда телевидение считалось новинкой; так что, даже если актер пукнет в кадре, все начинали говорить, что это свежая концепция и что ничего похожего до сих пор на телевидении не было.
>
> *Дж. Бернс*

Печатные издания. История развития — несколько столетий, а если учитывать изготовление документов на бумаге до изобретения книгопечатания, — несколько тысячелетий.

Существуют общепринятые нормы, правила и традиции, подкрепленные многовековым опытом. Большинство нововведений опираются на эти правила и традиции. Разработаны государственные и международные стандарты, сложилась общепринятая терминология.

Сайты. История развития — менее 15 лет.

Существуют некоторые общепринятые стандарты (например, протокол HTTP, схема именования ресурсов URL), но в целом дела со стандартизацией обстоят плохо. Браузеры по-разному отображают один и тот же код HTML и CSS, по-разному обрабатывают сценарии JavaScript и другие технологии. Разработчикам сайтов приходится прилагать усилия, чтобы сайты были совместимы с различными браузерами.

Дополнительно

Нет единообразия и в подходе к пользовательскому интерфейсу сайтов. Здесь уместно провести параллель с графическим интерфейсом традиционного программного обеспечения. Какую бы программу мы ни взяли — хоть текстовый редактор, хоть почтовый клиент, хоть файловый менеджер, хоть браузер, — всегда сверху мы увидим привычную строку меню с традиционными пунктами **Файл**, **Правка**, **Вид** и т. д., всегда в меню **Файл** можно найти команды **Открыть** и **Сохранить как**, а в меню **Правка** — **Вырезать** и **Копировать**. Под строкой меню всегда находится панель инструментов, в нижней части окна — строка состояния, справа — полоса прокрутки, в правом верхнем углу — кнопки **Закрыть** и **Свернуть**. Все стандартные элементы интерфейса в разных программах имеют один и тот же цвет и размер, команды меню и другие надписи отображаются одним и тем же шрифтом. Поэтому, если пользователь работал

хотя бы с одной такой программой, он легко сможет сориентироваться и в любой другой.

Для сайтов не существует подобного «эталона» интерфейса. Стандартные элементы — заголовки, логотипы, панели навигации, формы поиска и регистрации и др. — каждый разработчик размещает и оформляет по своему усмотрению. Аргументом в защиту такого подхода является то, что таким образом можно создать интерфейс, наиболее соответствующий целям и задачам именно этого сайта; кроме того, это придает сайту индивидуальность и позволяет разработчику выразить свои творческие замыслы. Но противоположная точка зрения заключается в том, что такое разнообразие дезориентирует пользователя, заставляя его на каждом новом сайте тратить лишнее время и интеллектуальные ресурсы, осваивая непривычный интерфейс. Кроме того, чем больше сайтов придется посещать, тем вероятнее, что пользователь запутается и станет регулярно «промахиваться», пытаясь на одних сайтах применить навыки, полученные на других.

В области технологий Интернета нет общепринятой терминологии, причем в нашей стране с ней особенная проблема. Программа для просмотра веб-страниц в одном месте называется «браузер», в другом — «броузер», в третьем — «обозреватель». Слово «страница» относится то к отдельной веб-странице, то к файлу HTML, то к целому сайту.

Новые технологии рождаются и умирают с необычайной скоростью. У новшеств в Сети нет тысячелетнего опыта, на который можно опираться. Поэтому только реальная практика использования сайтов показывает, удачным или неудачным оказалось то или иное решение разработчиков.

Внимание!

Доверяйте только реальной практике. Теоретические соображения, советы, рекомендации, книги, учебные пособия (в том числе эта книга) — это все может быть полезным, но только практическое использование сайтов позволяет делать окончательные выводы о том, как надо и как не надо их разрабатывать.

Примечание

Можно ли считать сайты средствами массовой информации? Сторонники признания их таковыми считают, что в этом случае разработчики и держатели сайтов будут подчиняться законодательству о СМИ, что повысит их ответственность за качество публикуемых материалов. С другой стороны, сайты далеко не во всем похожи на традиционные средства массовой информации — периодическую печать, радио, телевидение. Многие сайты скорее напоминают библиотеки, другие — справочные службы, третьи — сервисные центры... Но, тем не менее, ни один закон, относящийся к этим информационным структурам, на сайты не распространяется. К сожалению, это позволяет распространять через Всемирную Паутину недостоверную информацию, не рискуя понести ответственность.

Гиперссылки

Ссылка можетъ губительно подѣйствовать на пылкій нравъ молодого человека.

Е. А. Энгельгардт — о Пушкине (1820)

Пожалуй, самое значительное новшество, отличающее текст на сайтах, — это *гиперссылки*. Гиперссылки делают текст интерактивным, связывают отдельные веб-страницы в единую информационную систему.

Гипертекстовая страница — это не только документ, предназначенный для чтения, но и средство доступа к различным ресурсам: к другим гипертекстовым страницам, к загружаемым файлам, к службам Интернета и т. д. Поэтому пользователи относятся к веб-страницам иначе, чем к печатным изданиям. Их рассматривают одновременно и как средство информации (аналогично книге или радиопередаче), и как предоставляемую услугу (аналогично библиотеке или магазину). Это первое технологическое явление такого характера. Пожалуй, еще в восьмидесятых годах двадцатого века, если бы мы предложили кому-то представить нечто, что является одновременно книгой и магазином, — нас заподозрили бы в приверженности к абстрактному искусству.

Пользователи взаимодействуют с веб-страницами совсем не так, как с печатными изданиями. Веб-страницы в основном не читают, а просматривают, постоянно переходя с одной на другую. Это нужно учитывать при разработке сетевого текста.

1.2.3. Линейное — иерархическое

Нельзя представить себе явление моста в Беспредельность, ибо мост нуждается в устоях. Но Иерархия, как устои моста, доводит до берега Света.

Агни-Йога. Учение Живой Этики

Печатные издания. Материалы в печатных изданиях, как правило, расположены *линейно*, последовательно. Какие бы два материала (две статьи, две главы, две иллюстрации) мы ни взяли, обычно можно сказать, какой из них расположен раньше, а какой — позже. Конечно, читатель может просматривать материалы издания в произвольном порядке или выборочно (скажем, если речь идет о словаре), но ориентируется он при этом именно по линейной структуре и перемещается по изданию, последовательно перелистывая страницы.

Сайты. Для них характерна *иерархическая* структура, при которой отдельные страницы невозможно «выстроить в очередь». Иерархия начинается с главной

страницы, с ней связано несколько страниц следующего уровня (главные страницы основных разделов сайта), с каждой из этих страниц может быть связано по несколько страниц следующего уровня (главные страницы подразделов), и т. д. При перемещении по сайту пользователь ориентируется не по линейной последовательности, а по «дереву» страниц.

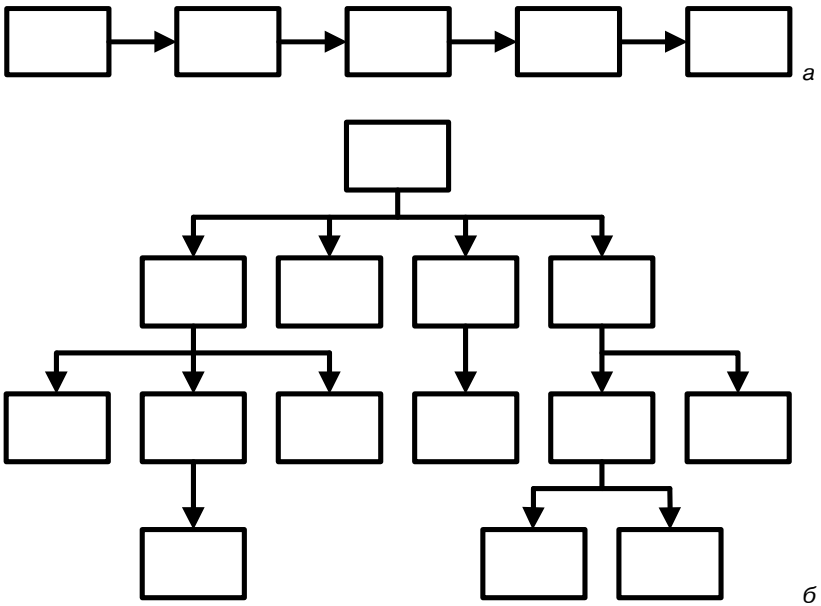


Рис. 1.2. Структура: *а* — линейная; *б* — иерархическая

Разработчику стоит учитывать эту особенность и организовывать структуру разделов таким образом, чтобы она помогала пользователю находить нужный материал.

纸 *Дополнительно*

При написании этой книги автор часто сталкивался с той проблемой, что один и тот же материал одинаково хорошо может разместиться в разных главах, причем без этого материала, наоборот, каждая из них получается менее удачной. Однако в книге было бы нелепо повторять одно и то же в нескольких местах: это, возможно, украсило бы каждую главу в отдельности, но испортило бы книгу в целом. А на сайте ничто не мешает предоставлять доступ к одной и той же странице из разных разделов, если она действительно подходит к каждому из них.

1.2.4. Безответное — интерактивное

> Наверное, когда-то будет создан музей форм взаимодействия, аналогичный музеям старого программного и аппаратного оборудования. Такой музей мог бы быть создан в Интернете.
>
> *Дж. Раскин. Интерфейс*

Пользователь может не только просматривать сайты, но и взаимодействовать с ними. Это достигается несколькими средствами:

- **гиперссылки** — различают гиперссылки нескольких видов, в зависимости от того, куда они ведут пользователя:
  - на веб-страницы:
    - ◊ в пределах одной страницы (как правило, для крупных страниц);
    - ◊ на другие страницы;
  - на файлы для загрузки (например, музыкальные записи или установочные комплекты программ);
  - на серверные сценарии.

  Бывают ссылки, вызов которых не заменяет текущую страницу, а приводит в действие какой-нибудь программный сценарий, выполняющийся на сервере, — например, для регистрации пользователя или для сохранения его ответа на интерактивный опрос. Впрочем, даже в этом случае лучше, если сценарий будет не только выполнять свою непосредственную функцию, но и уведомлять пользователя о том, что его щелчок по ссылке привел к нужному результату;
  - на ресурсы вне Всемирной Паутины (например, на адрес электронной почты или на сервер телеконференций);
- **формы.** Форма — это один или несколько элементов управления (текстовых полей, флажков, кнопок и т. д.), с помощью которых пользователь передает информацию на сервер. Например, простая форма любой поисковой системы состоит всего из двух элементов: строки ввода запроса и кнопки для отправки. А сложная форма для регистрации или оплаты по кредитной карте может включать в себя десятки элементов;
- **другие средства интерактивного взаимодействия** — сюда относятся клиентские сценарии и апплеты, предназначенные, например, для демонстрационных целей или для игр. В отличие от гиперссылок и форм, такие средства могут функционировать не у всех пользователей (например, если они не поддерживаются браузером или отключены из соображений безопасности). Поэтому разработчику следует применять их осторожно и не использовать там, где можно обойтись более универсальными средствами.

1.2.5. Управляемое — неуправляемое

Нам не дано предугадать,
Как слово наше отзовется...

Ф. Тютчев

Печатные издания. Внешним видом издания полностью управляет издатель. Он устанавливает все шрифты, цвета, размеры, относительное расположение элементов на страницах, сорт бумаги, тип печати и остальные параметры издания.

Читатель получает в свое распоряжение все издание целиком, и поэтому он сразу понимает его структуру, композицию и содержание, что позволяет ему выбрать наиболее удобный способ чтения (например, подряд или выборочно).

Сайты. Пользователь не обязательно начинает посещение сайта с главной страницы. Ссылка с другого сайта или из поисковой системы может привести его сразу на внутреннюю страницу. Но даже если пользователь начинает с главной страницы, он все равно не может оценить масштаб сайта так же легко, как это происходит с печатным изданием.

Таким образом, разработчик не может предсказать, с какой страницы пользователь начнет знакомство с сайтом. Значит, придется делать каждую страницу такой, чтобы с нее было удобно начинать.

Разработчик также не должен надеяться, что пользователь будет читать все страницы сайта. Вполне возможно, что нужная пользователю информация или служба сосредоточена всего на одной странице. Пользователь посетит ее и уйдет. Значит, каждая страница в отдельности должна быть самостоятельным, полноценным ресурсом. В газете или журнале одна статья вполне может занимать несколько страниц — в Интернете это нежелательно.

Примечание

Существует немало оформительских решений, которые часто встречаются в печатных изданиях, но практически невоспроизводимы на сайтах. Это, например, вертикальное или диагональное расположение текста, текст с тенью, зеркальное отражение, различные наложения блоков. Некоторые разработчики считают, что это ограничивает творческую свободу создателя сайтов. Но ведь, с другой стороны, сайты позволяют реализовать много других решений, невозможных в печати, — например, в области интерактивного взаимодействия. В самом деле, сайты стоит приравнивать к печатным изданиям не более, чем книги к телепрограммам. Это разные информационные ресурсы, и нет ничего трагического в том, что приемы и решения, применимые для одного из них, неприменимы для другого.

Внешний вид сайта и взаимодействие с ним в существенной степени определяются пользователем. Перечислим факторы, которые на это влияют:

- **аппаратное обеспечение** — от него зависит, какая операционная система доступна пользователю и какие мощности могут быть выделены на работу с веб-страницами. Например, на компьютерах от Apple возможно установить только операционные системы класса Mac OS. Или, например, компьютер с маломощным аппаратным обеспечением может «поперхнуться» страницей, на которой слишком много сложных сценариев JavaScript. И еще пример: в «макинтошах» применяется мышь с одной кнопкой. Поэтому на веб-странице не стоит предлагать пользователю зачем-либо щелкнуть *правой* кнопкой мыши;
- **операционная система** — от нее зависит, какие браузеры могут быть установлены у пользователя, а также некоторые другие аспекты — например, системные шрифты или особенности интерфейса. Например, в графических оболочках для систем класса UNIX может не быть шрифта Arial, поэтому, если этот шрифт используется на странице, пользователь UNIX увидит ее не так, как задумал разработчик;
- **видеорежим** — на самом деле это не один фактор, а несколько. Во-первых, видеорежим может быть графическим или текстовым. (В текстовом режиме работают, например, браузеры Lynx и Links.) Во-вторых, в графическом режиме у разных пользователей разное разрешение; кроме того, окно браузера может быть развернуто не на весь экран. Так что совершенно невозможно предсказать, сколько пикселов в ширину будет занимать веб-страница.

  Длина диагонали экрана также имеет значение. При одном и том же разрешении пользователю с маленьким экраном текст на странице может показаться слишком мелким, а пользователю с большим экраном — слишком крупным.

  Кроме того, пользователь может просматривать сайт не на экране персонального компьютера, а на каком-нибудь другом устройстве — например, на наладонном компьютере или с помощью мобильного телефона;
- **браузер** — как уже говорилось, разные браузеры могут по-разному отображать одну и ту же страницу, и разработчикам приходится прилагать усилия, чтобы добиться совместимости. Впрочем, здесь идет речь не о том, чтобы сайт во всех браузерах выглядел абсолютно одинаково — это невозможно, да и не нужно, — а о том, чтобы он всюду сохранял свою логическую структуру, работоспособность навигации, форм и остальных служб;
- **дополнительные модули** — некоторые сайты требуют для своего просмотра дополнительных модулей и компонентов операционной системы — таких,

как Flash или ActiveX. Если на компьютере пользователя этих модулей нет — сайт может оказаться для него недоступным;

- **настройки пользователя** — наконец, пользователь может настроить режим просмотра веб-страниц «под себя» — например, отключить загрузку изображений (для экономии трафика), увеличить или уменьшить размер шрифта, запретить сценарии JavaScript — и т. д., вплоть до подключения к веб-странице своей собственной таблицы стилей.

纸 Дополнительно

Разработчики относятся к этому по-разному. Одни считают, что пользователь таким образом может нарушить композицию и замысел оформителя, и предлагают делать страницы так, чтобы пользователь не смог их «испортить». Другие возражают, что пользователь лучше знает, как ему удобнее просматривать веб-страницу; кроме того, никто же не протестует по поводу того, что в традиционных программных продуктах пользователь давно уже может настраивать весь интерфейс как ему угодно. Оппоненты приводят новый аргумент: никто ведь не дает посетителям картинной галереи перерисовывать «под себя» Мону Лизу! На это приводится возражение, что сайты все-таки гораздо ближе к программным продуктам, чем к произведениям искусства. И спор продолжается до бесконечности.

Независимо от того, какую сторону в этом споре принять, приходится согласиться с тем, что пользователи, несмотря ни на что, могут настраивать внешний вид сайта. И разработчики вынуждены с этим считаться.

Хотя разработчик не может полностью предугадать, как его сайт будет выглядеть у различных пользователей, существуют компоненты сайта, которые остаются неизменными на любых устройствах просмотра. Вот они:

- содержание;
- структура;
- навигация.

Разработчику следует обращать особенно пристальное внимание на эти компоненты. Если они сделаны качественно, грамотно, логично и удобно, — сайт останется доступен даже для самого экзотического пользователя с самым экзотическим устройством.

1.2.6. Что же их объединяет?

The Hatter opened his eyes very wide on hearing this; but all he said was “Why is a raven like a writing-desk?”

Lewis Carroll. Alice’s Adventures in Wonderland[3]

Внимание!

Сайты и печатные издания:

- предоставляют статичную информацию;
- главенствующую роль в них играет текст.

Получается, что сайт представляет собой информационный ресурс, объединяющий свойства многих объектов: телевидения и радио, традиционных программных продуктов, магазинов и бюро услуг и даже произведений искусства. У сайтов есть и свои собственные свойства — например, гиперссылки. Следовательно, к разработке сайтов нельзя относиться однобоко. Нельзя бездумно переносить на них все принципы, справедливые для какого-то одного из перечисленных объектов.

Но это не означает, что у сайтов нет совсем ничего общего с печатными изданиями. Общее заключается в том, что и те и другие предоставляют в первую очередь *статичное* содержание — т. е. такое, скорость и продолжительность взаимодействия с которым определяет сам пользователь. В отличие от телепрограммы, которая транслируется в строго установленное время, сайт доступен постоянно, и пользователь сам решает, в какое время его посетить. Никто не торопит пользователя при чтении текста или просмотре изображений на сайте, никто не мешает ему неограниченно долго думать над заполнением полей формы.

Конечно, на сайтах могут быть размещены не только статические объекты (такие, как текст или изображения), но и анимационные ролики, звуковые записи и другое динамическое содержание. Но грамотные разработчики размещают их таким образом, чтобы пользователь сам решал, когда запустить или остановить проигрывание записи.

---

[3] Услышав это, Болванщик широко раскрыл глаза, но произнес только: «Чем ворон похож на письменный стол?» *(Л. Кэрролл. Алиса в Стране Чудес)*

Дополнительно

Если не соблюдать это правило, пользователь может столкнуться с трудностями. Например, некоторые разработчики до сих пор встраивают в веб-страницы музыкальные файлы, которые автоматически начинают проигрываться по окончании загрузки страницы. Представьте себе пользователя, который сидит за компьютером, просматривает сайты и одновременно с этим слушает свою любимую музыку. И вдруг при открытии очередного сайта любимая музыка перекрывается громким звучанием MIDI! Что подумает пользователь о разработчиках такого сайта?

И еще одно качество, общее для сайтов и печатных изданий, — это *главенствующая роль текста*. Текст несет основную часть информации на сайтах и в печати, по фрагментам текста пользователь ориентируется на сайте, а читатель — в издании.

Дополнительно

Один из предметов разногласий специалистов — применение *модульной сетки* для разработки внешнего вида сайтов. В традиционной полиграфии модульная сетка служит каркасом для размещения объектов на странице, определяет ее композицию. На сайтах модульная сетка часто организуется с помощью таблиц HTML, задавая жесткие размеры и относительное расположение всех блоков. Многие эксперты в области удобства сайтов склоняются к тому, что это решение не вполне удачно. По их мнению, пользователь должен иметь возможность, например, увеличить размер шрифта на странице, если тот кажется ему слишком мелким и неудобным для чтения. Но если жестко задан размер блока, в котором расположен текст, — то при увеличении шрифта текст перестанет умещаться в этом блоке, и либо композиция страницы «поползет», либо часть текста скроется. Другие специалисты идут еще дальше и рекомендуют оставлять на усмотрение пользователя не только размеры шрифта и блоков, но и их относительное расположение. Они предлагают внедрять средства, которые позволяли бы пользователю «перекраивать» композицию страницы по своему усмотрению. Время покажет, будут ли эти предложения внедрены в жизнь.

Дополнительно

А вот пример, когда, наоборот, понятие из традиционной полиграфии прочно укоренилось в области разработки сайтов.

Издатели газет и другой подобной продукции стараются размещать наиболее важные материалы и заголовки «над сгибом» первого листа, то есть в верхней его половине, чтобы читатель видел их, не разворачивая и не листая газету.

У веб-страницы тоже есть свой «сгиб» — это нижняя граница окна браузера. Конечно, в отличие от газеты, высота окна у всех пользователей разная, но разработчики сайтов стараются, чтобы при всех наиболее распространенных размерах окна самые важные элементы страницы располагались в пределах экрана и были доступны без вертикальной прокрутки. Верхняя часть веб-страницы, умещающаяся на экране, часто так и называется — область «над сгибом».

На этом завершается теоретическое введение, задачей которого было описать текст как форму представления информации и обозначить специфику сайтов, отличающую их от печатных изданий. Автор убежден в том, что без устойчивых теоретических основ не может формироваться никакая эффективная практика. Конечно, в области разработки сайтов именно практика имеет определяющее значение. Но полностью отвергать роль теории было бы ошибкой. Одна из задач этой книги — убедить читателей в том, что в разработке и оформлении текста для сайтов эффективными и полезными на практике оказываются только те принципы и технологии, которые согласуются с изложенным здесь теоретическим базисом.

```
<?xml version= 1.0
<!DOCTYPE html PUBL
  "-//W3C//DTD XHTM
  "http://www.w3.or
<html xmlns="http:/
  lang="ru" xml:lan
<!-- Редактировал:
<head>
  <title>Евгений Он
  <link rel="styles
    href="index.css
</head>
```

Глава 2

От «А» до «Я»

Важные особенности грамматики и типографики для сайтов на русском и английском языках

> Решено было не допустить ни одной ошибки. Держали двадцать корректур. И все равно на титульном листе было напечатано: «Британская энциклопудия».
>
> *И. Ильф*

> ...Это наше дело, мы объединимся и пойдем в деревни, чтобы ликвидировать безграмотность, чтобы наше подрастающее поколение не имело безграмотных.
>
> *В. И. Ленин. Задачи союзов молодежи*

Грамотность текста на современных сайтах оставляет желать много лучшего. С одной стороны, Всемирная Паутина предоставила пользователям небывалую свободу слова: кто угодно может разместить в Сети любую информацию, и в тот же момент эта информация станет доступна всему миру! Но такая свобода имеет и обратную сторону. Ранее уже говорилось о том, что Интернет не защищен от недостоверной информации. Кроме этого недостатка, приходится отметить и удручающую безграмотность сайтов. Даже если разработчики сайта предоставляют проверенную и полезную информацию, им часто не удается изложить ее грамотным русским языком. А это затрудняет чтение и снижает доверие к информации.

Некоторые крупные сайты привлекают профессиональных технических писателей, редакторов и корректоров. Хочется надеяться, что в будущем эта практика распространится повсеместно, и глава, которую вы сейчас читаете, утратит свою актуальность.

Тем не менее, пока поголовного перехода на профессиональную работу с текстом на сайтах не произошло, вам предлагается освежить в памяти некоторые правила русской и английской грамматики и типографики, которые особенно часто нарушаются на сайтах.

2.1. Что написано пером

Некоторые правила грамматики и типографики

Даже молчал он с грамматическими ошибками.

С. Е. Лец

Печатник, не зевай и точку ставь, где надо.
Иного критика ведь может взять досада:
В бессмыслице не видит он вреда,
А точки не найдет — беда!..

Л. Боровиковский

В этой книге не будет идти речь об орфографии и пунктуации. Это не значит, что автор не считает важным соблюдение орфографических правил. Просто правила орфографии и пунктуации для сайтов ничем не отличаются от правил для любого другого текста, и с ними можно ознакомиться в различных справочных пособиях (например, в [1]). Однако автор надеется, что вы не совершаете «школьных» грамматических ошибок, поэтому речь сразу пойдет о том, что обычно выходит за рамки школьной программы.

Примечание

Иногда можно встретить такой вопрос: «Зачем изучать грамматику, если современные текстовые процессоры оснащены средствами проверки орфографии?» Увы, но эти средства не способны заменить человека. Например, ни одна программа не исправит ошибку, если вы написали «эффект» вместо «аффект» или «компания» вместо «кампания». Ни одна такая программа не поставит пропущенной запятой в знаменитой фразе «Казнить нельзя помиловать» и уж тем более не обнаружит стилистические ошибки (например, если вы напишете «играет большое значение» или «одеть пальто»). Автоматические средства проверки орфографии хороши для грамотного человека, причем как вспомогательный инструмент, позволяющий обнаружить случайные опечатки. Неграмотному человеку они, наоборот, могут повредить.

2.1.1. Пробелы

> Полукруглая шпация слишком широка для того, чтобы ее можно было использовать в качестве стандартного пробела между словами. Поэтому круглая шпация была разделена на три вертикальные части, которые называются 3*т*-пробелами.
>
> *М. Клепер. Практическое руководство по цифровой печати*

Казалось бы, с пробелами все понятно. С первого класса известно, что пробелы ставятся между словами. Тем не менее, на веб-страницах встречаются ошибки, связанные с пробелами.

Например, если слова написаны не полностью, а сокращены (допустим, в объявлении: *Мол. чел. жел. позн.*), пробелы между ними не должны никуда пропадать. Поэтому распространенные сокращения *и т. д.*, *и т. п.*, *т. е.*, *т. н.*, *до н. э.*, *у. е.*, *в т. ч.* и им подобные нужно записывать с надлежащими пробелами.

Это же касается личных инициалов: пробелы после них ставятся так же, как если бы инициалы были расшифрованы: *А. С. Пушкин*, *К. Шеннон*.

Наконец, не надо игнорировать пробел между числом и относящимся к нему обозначением единицы измерения: *10 кг*, *2004 г.*

Внимание!

Пробел — признак нового слова. Всякий раз, когда в тексте начинается новое слово, оно должно отделяться от предыдущего текста пробелом. Это правило действует независимо от того, сокращены слова или нет.

Правильно	Неправильно
и т. д.	и т.д.
до н. э.	до н.э.
А. С. Пушкин	А.С.Пушкин, А.С. Пушкин
2004 г.	2004г.

В печатном деле применяются пробелы различной ширины. Ширина стандартного пробела в отечественном издательском деле составляет половину кегля текущего шрифта. Именно этот пробел применяется во всех случаях, если нет никаких специальных причин для какого-нибудь другого. Среди других распространен пробел шириной в 2 типографских пункта (2 п.).

Он ставится, например, между цифрами и знаком в арифметических выражениях: $2+2$. Более подробно об употреблении разных типов пробелов можно прочитать в специальной литературе [3, 4].

Впрочем, технологии разработки сайтов не позволяют следить за шириной пробелов так же внимательно, как это делается в печатных изданиях. Браузеры сами устанавливают ширину всех пробелов, исходя из количества и характера текста. В большинстве случаев это не приводит к заметным проблемам. Тем не менее, во власти разработчика обеспечить соблюдение следующих правил:

- все междусловные пробелы в одной строке должны быть одинаковы по ширине;
- ширина междусловных пробелов не должна быть меньше ¼ и больше ¾ кегля шрифта.

В *главе 5* пойдет речь о том, как практически выполнить эти правила средствами HTML и CSS.

Далее в этой главе вы найдете много правил, касающихся расстановки пробелов в сочетании с другими знаками.

2.1.2. Скобки и кавычки

> Use the straight single and double quotes ' and " only to denote minutes, seconds, feet, and inches; and then only if there is insufficient space to spell out the unit. For all other purposes, use typographer's (curly) quotes.
>
> *Ch. Poynton. Ten Common Mistakes in the Typesetting of Technical Documents*[1]

В русском языке используются круглые скобки и кавычки-«елочки» (они же — угловые, или французские кавычки). Допустимы также кавычки-„лапки“ (обычно в качестве вложенных кавычек).

Круглые скобки	( )
Кавычки-«елочки»	« »
Кавычки-„лапки“	„ “

[1] Используйте прямые одинарные и двойные кавычки ' и " только для обозначения минут, секунд, футов и дюймов; и то только в том случае, если нет места для полного обозначения единицы измерения. Во всех остальных случаях используйте типографские (фигурные) кавычки. *(Ч. Пойнтон. Десять главных ошибок при наборе технической документации.)*

На сайтах и даже в печатных изданиях можно встретить досадную ошибку — прямые кавычки:

Прямые кавычки " "

Дополнительно

Использование прямых кавычек — такое же проявление неграмотности, как применение косой черты в качестве скобок /как в этом предложении/ или дефиса - вместо тире. И прямые кавычки, и косая черта — суррогаты, употреблявшиеся для имитации правильных кавычек и скобок во времена пишущих машинок. В те времена рукопись, набранная на машинке, поступала в типографию, и профессиональный наборщик переносил ее на печатную плату, заменяя суррогаты на нужные символы. Если вы посмотрите на книги, изданные на русском языке до эпохи компьютерных издательских систем, вы не найдете нигде прямых кавычек.

К сожалению, в результате широкого распространения компьютерных технологий издательские мощности начали попадать в руки людей, не знакомых с правилами русского языка. Они набирают прямые кавычки, не подозревая о том, что это неграмотно. Культурному человеку, привыкшему читать профессионально сделанные книги, прямые кавычки «режут глаз» точно так же, как режут слух фальшивые ноты.

Внимание!

Скобки и кавычки не отделяются пробелами от заключенного в них текста.

Правильно	Неправильно
(заметим в скобках)	(заметим в скобках)
«Анна Каренина»	« Анна Каренина »

Квадратные скобки [ ] употребляются в русском языке очень ограниченно: например, для ссылок на литературу (вы встречали такие ссылки несколькими абзацами ранее) или при нумерации формул, да и там их лучше заменять круглыми.

Дополнительно

В английском языке употребляются кавычки, которые так и называются — английские:

Английские кавычки “ ”

Еще один важный символ в английском языке — апостроф. Апостроф выглядит как запятая, но располагается в верхней части строки:

Апостроф ’

Апостроф нельзя путать с прямой одинарной кавычкой. Это такая же ошибка, как употребление в русском языке прямых кавычек вместо «елочек» или косой черты вместо скобок.

Правильно	Неправильно
“Jabberwocky”	"Jabberwocky"
I’m, it’s, O’Reilly	I'm, it's, O'Reilly

纸 ***Дополнительно***

В русском языке недопустимо употреблять апостроф вместо твердого знака. «Об’явление», «под’езд» — это грубые ошибки. Апостроф употребляется только в некоторых иностранных фамилиях: *Д’Артаньян*, *Жанна Д’Арк*, *О’Коннор*.

2.1.3. Знаки препинания

Уже поставил точку,
Труба трубит отбой —
А точка усмехнулась
И стала запятой!

Ю. Ким. Тихий марш

Сейчас вас не будут учить тому, где в тексте нужно ставить запятые и тире. Вместо этого пойдет речь о том, *как* их нужно ставить.

В русском языке употребляется десять знаков препинания, не считая комбинированных:

Точка	.
Запятая	,
Точка с запятой	;
Двоеточие	:
Многоточие	...
Восклицательный знак	!
Вопросительный знак	?
Дефис	-
Короткое тире	–
Длинное тире	—
Комбинированные знаки	?! !.. и др.

Кто-то, возможно, удивится: разве многоточие — это не то же самое, что три точки подряд? Представьте себе, нет: точно так же, как буква «W» — не то же самое, что две буквы «V» подряд. Практически во всех шрифтах,

предназначенных для набора текста, есть специальный символ многоточия. Если вы посмотрите на этот символ, а потом на просто набранные подряд три точки — вы увидите отличия.

Следующим сюрпризом может оказаться различие между дефисом и двумя видами тире. Это действительно три разных символа, и подробнее о них — чуть позже.

Внимание!

Точка, запятая, точка с запятой, двоеточие, многоточие, восклицательный знак, вопросительный знак не отделяются пробелом от предшествующего символа.

Напомним, что, согласно правилу из *разд. 2.1.1*, если после знака препинания начинается новое слово, то оно отделяется от знака пробелом.

Правильно	**Неправильно**
тот, который	тот,который тот , который тот ,который

Это не значит, что после знаков препинания всегда ставится пробел. Вот несколько примеров, когда пробел не нужен, так как новое слово не начинается:

- после знака препинания следует закрывающая скобка или закрывающая кавычка. В этом случае действует правило из *разд. 2.2.2*: *поэма «Хорошо!»*, *пять кочерег (или кочерг?)*;
- точка разделяет уровни нумерации (например, в заголовках или элементах перечня: *2.1.3. Знаки препинания*). Обратите внимание: между цифрами нет пробелов, но после всех цифр, перед словом «Знаки» стоит пробел, как и положено по правилу из *разд. 2.1.1*;
- точка используется в адресах Интернета, именах файлов и других технических обозначениях: *www.bhv.ru*, *info@microsoft.com*, *телеконференция alt.html.tags*. Во многих обозначениях такого рода пробел является значимым символом, таким же, как буква, цифра или точка. Поэтому *www.bhv.ru* и *www. bhv. ru* — два разных адреса.

Примечание

Отдельного упоминания заслуживает ситуация с буквами «е» и «ё». Правила русского языка, действующие с 1956, предписывают употреблять букву «ё» там, где нужно предотвратить неправильное прочтение: *совершённый* (в отличие от *совершенный*), *река Олёкма*.

Не все согласны с этим правилом. Автору нередко приходится встречать противников вспомогательного положения буквы «ё», которые призывают вообще нигде не заменять ее на «е».

Это бесполезно. Даже если употреблять «ё» всюду, где возможно, пользы для читателя не будет. Читатели привыкли к тому, что «ё» обычно не используется. Поэтому, например, встретив слово «совершенный» (через «е»), читатель все равно может подумать, что имеется в виду «совершённый».

Не уделяйте чрезмерного внимания букве «ё». В разработке текста для сайтов есть много других, гораздо более важных аспектов.

Знаки препинания при скобках и кавычках

Если в скобки или кавычки заключено несколько слов (не составляющих целое предложение) и если после этих слов необходим знак препинания, то он ставится после закрывающей скобки или кавычки.

Если в скобки заключено целое предложение, то заключительный знак препинания (например, точка) ставится перед закрывающей скобкой.

Если в кавычки заключено целое предложение, то заключительная точка ставится после закрывающей кавычки, а все остальные знаки препинания (восклицательный, вопросительный, многоточие) — перед закрывающей кавычкой.

Правильно	Неправильно
Даниил Хармс (Ювачев), как известно...	Даниил Хармс (Ювачев,) как известно...
(Это примечание.)	(Это примечание).
Он сказал: «Поехали!»	Он сказал: «Поехали»!
«Счастливые часов не наблюдают».	«Счастливые часов не наблюдают.»

Перед открывающей скобкой не может стоять запятая, точка с запятой или двоеточие.

Если предложение заканчивается сокращенным словом, то после него ставится только одна точка, а не две. Но если это слово заключено в скобки или кавычки, тогда точка, относящаяся к сокращению, ставится внутри скобок (кавычек), а точка, заканчивающая предложение, — снаружи: *...и клиентские технологии (HTML, CSS, JavaScript и т. д.).*

Дополнительно

В комбинированных знаках, составленных из многоточия и вопросительного или восклицательного знака, — ровно три точки, а не четыре: *понятно?.. понятно!..*

Примечание

В докомпьютерную эпоху набором и версткой печатных изданий занимались специально подготовленные наборщики, владеющие правилами грамматики и типографики. Даже если наборщик допускал ошибку, ее исправлял корректор. Если ошибка все-таки просачивалась в готовое издание, наборщикам и корректорам приходилось нести ответственность.

Сейчас — все по-другому. Любой, у кого есть компьютер и принтер, может в одиночку изготовить печатное издание. Но далеко не любой знает правила русского языка, и никто не будет наказывать такого «издателя» за ошибки.

Это касается и разработчиков сайтов, которые редко следят за грамотностью своего текста и соблюдением типографских правил. «Свобода слова» в Интернете приводит к тому, что Всемирная Паутина переполнена безграмотными, дилетантскими текстовыми поделками.

Автор ни в коем случае не предлагает ввести тотальный контроль за грамотностью сетевого текста или цензуру в масштабах Интернета. Автор просто рекомендует вам при разработке сайтов обращать больше внимания на качество текста и надеется, что эта книга поможет улучшить вашу работу.

2.1.4. Дефис, тире, минус

> Если бы назначение тире в полной мере отвечало своему смыслу и если бы тире всегда замещало мысль, то вся книга могла состоять исключительно из тире. Удивительно, как до этого некоторые еще не додумались.
>
> *Я. Чихольд. Облик книги*

Дефис — это короткая черточка, которую обычно можно найти на клавиатуре справа от цифры «0». Дефис соединяет части сложных слов, например:

- в словах с частицами «-то», «-ка», «-либо» и т. д.: *Король-то голый! А ну-ка, девушки!*;
- в наречиях с приставками «по-», «во-» и т. д.: *по-человечески*, *во-первых*;
- в некоторых сложных словах, образованных от двух корней: *генерал-майор*, *премьер-министр*, *горе-работник*;
- в парных сочетаниях и между повторяющимися словами: *жили-были*, *честно-честно*, *путь-дорога*.

Здесь не рассказывается подробно о том, в каких именно случаях должен или не должен ставиться дефис — эта тема освещена в пособиях по русскому языку, например, в [1]. Здесь хочется обратить ваше внимание только на то,

что везде, где правила русского языка предписывают употреблять дефис, не следует ставить вместо него тире или какой-нибудь другой знак.

Дефис не отделяется пробелами от тех частей слова, которые он соединяет.

Дополнительно

Так называемый *висячий дефис* употребляется при сочетании с помощью союзов «и» и «или» нескольких сложных слов с одинаковой второй частью: *радио- и телепередачи*, *двух-, трех- и четырехместные номера*.

Дефис	-	Длинное тире	—
Короткое тире	–	Минус	–

Тире, в отличие от дефиса, употребляется не для соединения частей одного слова, а для разделения разных слов: *Знание — сила*, *курить — здоровью вредить*.

Раньше в отечественной типографской практике существовало только одно тире — длинное, как в этом предложении. Его ширина равна кеглю текущего шрифта, а толщина обычно немного меньше, чем у дефиса. Сейчас получило широкое распространение и короткое тире, которое вдвое короче длинного. Два этих знака поделили «сферы влияния» следующим образом:

Короткое тире употребляется:

- при обозначении пределов (от ... до ...): временных (*1799–1837*), пространственных (*поезд Москва – Петербург*), количественных (*5–6 кг*);
- в названиях физических законов, открытых несколькими учеными (*закон Ломоносова – Лавуазье*) и других подобных сочетаниях.

Дополнительно

Учтите, что короткое тире ставится между фамилиями в том случае, если они принадлежат двум *разным* людям. В двойной или тройной фамилии, принадлежащей одному человеку, используется дефис: *Голенищев-Кутузов*, *Щепкина-Куперник*.

Тире между числами рекомендуется использовать только там, где читатель заведомо не перепутает его со знаком «минус».

Короткое тире не отделяется пробелами, если с обеих сторон от него стоят цифры. В остальных случаях в справочной литературе можно встретить противоречивые предписания (ср. [1] и [6]). Во всех спорных случаях автор предпочитает отделять короткое тире пробелами.

Длинное тире употребляется во всех остальных случаях, где правила русского языка предписывают использовать тире.

Длинное тире всегда отделяется пробелами с обеих сторон.

纸 Дополнительно

Отечественные справочники по набору текста предписывают исключение из этого правила: если перед тире стоит точка, запятая или многоточие, то два этих знака не разделяются пробелом:

Правильно	Неправильно
— Привет,— сказал он.	— Привет, — сказал он.

На практике это исключение почти совсем забыто. Если посмотреть на книги, изданные в нашей стране еще в 60-х годах XX века, там практически нигде не найти пробела между точкой (запятой, многоточием) и тире. Однако с конца 70-х годов этот пробел появляется все чаще, и сейчас встречается в большинстве печатных изданий. Тем не менее, если вы хотите набирать текст строго по правилам, не ставьте этого пробела.[2]

纸 Дополнительно

В номерах телефонов между группами цифр должен стоять дефис, а не тире — ведь это же один целый номер, а не несколько разных.

Правильно	Неправильно
какой-нибудь	какой – нибудь какой — нибудь какой—нибудь
1799–1837	1799-1837 1799—1837 1799 – 1837
123-45-67	123–45–67
Знание — сила	Знание - сила Знание – сила Знание—сила

[2] Пробел между точкой (запятой, многоточием) и тире в «докомпьютерную» эру не ставился в связи с тем, что металлическая точка из ручного набора уже имела определенную толщину, и надобности в специально добавляемом пробеле не было. При компьютерном наборе принято вставлять такой пробел, в частности, все современные справочники, в т. ч. и широко известный «Справочник издателя и автора» А. Э. Мильчина и Л. К. Чельцовой (М.: ОЛМА-Пресс, 2003), набраны подобным образом. Не является исключением и эта книга. — *Ред.*

Дополнительно

Иногда в тексте на сайтах можно встретить конструкцию из двух или даже трех дефисов подряд, которая, как считается, должна заменять тире. Такая «замена» -- неграмотна и совершенно недопустима.

Знак «минус» используется в математических выражениях. Его ширина обычно такая же, как у короткого тире. Но тире располагается примерно на высоте середины строчных букв, а знак «минус» — на высоте середины прописных букв.

Короткое тире	10 – 15
Минус	10 – 15

Речь о математических знаках пойдет в *разд. 2.1.5.*

Дополнительно

Правила расстановки знаков препинания в английском языке другие. Это можно проиллюстрировать на фрагменте из «Алисы в Зазеркалье» Л. Кэрролла:

> “That’s just what I complain of,” said Humpty Dumpty. “Your face is the same as everybody has—the two eyes, so——” (marking their places in the air with his thumb) “nose in the middle, mouth under. It’s always the same. Now if you had the two eyes on the same side of the nose, for instance—or the mouth at the top—that would be *some* help.”[3]

Здесь видно, что длинное тире не отделяется пробелами от окружающих слов, а знаки препинания ставятся перед закрывающей кавычкой, а не после нее. Кроме того, можно заметить одно *очень* длинное (так называемое *фигурное*) тире, которое обычно используется в художественной литературе для выделения прямой речи.

2.1.5. Математические знаки

> Наборщики-формулисты в соответствии с тарифно-квалификационным справочником являются рабочими высшей квалификации, имеющими 6-й тарифный разряд.
>
> *М. Шульмейстер. Набор формул*

Знаки математических операций (+, ·, ×, : и др.) и соотношений (<, >, = и др.) в выражениях отделяются от окружающих операндов (слагаемых,

[3] «О чем я и говорю, — ответил Шалтай-Болтай. — У тебя лицо такое же, как у всех: два глаза (он указал на них большим пальцем), нос посередине, рот внизу. Ничего особенного. Вот если бы у тебя, например, оба глаза были по одну сторону от носа, или рот наверху — это было бы уже кое-что».

множителей и т. д.) пробелами. В печати это должны быть пробелы шириной в 2 п., но на веб-странице их можно достичь только ценой ухудшения удобства и доступности. Поэтому при современном уровне средств просмотра сайтов следует пользоваться обычными пробелами.

Правильно	**Допустимо на веб-странице**	**Неправильно**
2 + 2	2 + 2	2+2
$x = 10$	$x = 10$	x=10

В качестве знака умножения можно в зависимости от ситуации использовать точку на средней линии ( ) или крестик (×), но ни в коем случае не букву «x».

Правильно	**Неправильно**
$2 \times 2 = 4$	2 x 2 = 4
$2 \cdot 2 = 4$	

Если математический символ употребляется не между операндами, а перед одиночным числом (например, чтобы показать его знак или приращение), пробел между символом и числом не ставится.

Правильно	**Неправильно**
+20 °C	+ 20 °C
–5 м (над уровнем моря)	– 5 м (над уровнем моря)

Знаки в верхнем и нижнем индексе не отделяются пробелом от предшествующего символа.

Правильно	**Неправильно**
$м^2$	$м^{\ 2}$
NH_3	$NH_{\ 3}$

Более подробные правила набора математических выражений и формул можно найти в специальной литературе [3, 4, 6].

Для набора формул и других научных записей в отечественной литературе, как и на Западе, фактически действуют правила набора, установленные издательской системой TEX. Если вы собираетесь включать в свой текст формулы сложнее, чем «$2 \times 2 = 4$», то вам обязательно нужно ознакомиться с этой системой.

Примечание

Не забывайте, что эта книга — не исчерпывающее руководство по типографскому оформлению текста. Здесь изложены только самые необходимые основы набора и синтаксического оформления текста. Автор надеется, что если вы встретитесь с проблемами, не описанными в книге, то не станете придумывать собственных правил русского языка, а обратитесь к специальной литературе или за советом к профессионалам типографского дела.

2.1.6. Числа

Позавидовала Единица Десятке.
«Конечно, с такой кругленькой суммой, как этот ноль, я бы тоже кое-что значила!»

Ф. Кривин. Величина

Целые числа в цифровой форме, в которых больше четырех разрядов, следует разбивать пробелами на группы по три цифры (справа налево). Не разбиваются числа, стоящие после знака номера (№), в обозначениях марок машин, в номерах документов.

Правильно	Неправильно
1 234 567	1234567
6789	6 789
№ 101312	№ 101 312

Между группами цифр ставится именно пробел, а не какой-нибудь другой знак. В печатной продукции это снова должен быть пробел в 2 п., но на веб-странице лучше обойтись обычным.

В десятичных дробях между целой и дробной частью ставится запятая (после которой не должно быть пробела). Многие разработчики сайтов, знакомые с программированием, привычно ставят не запятую, а точку — и таким образом совершают ошибку.

Натуральные дроби в тексте записываются через дробную черту, причем числитель указывается в верхнем индексе, а знаменатель — в нижнем. Если дробное число имеет целую часть, она пишется слитно с дробью.

Правильно	Неправильно
12,345	12.345
$^{1}/_{5}$	1/5
$2\,^{1}/_{3}$	$2\ ^{1}/_{3}$

纸 Дополнительно

В английском языке многоразрядные числа также разбиваются на группы, но разделяются не пробелом, а запятой (без пробела после нее). В десятичных дробях, наоборот, целую часть от дробной отделяют точкой.

Русский язык	Английский язык
1 234 567	1,234,567
12,34	12.34

Иногда в тексте используются римские цифры:

I	V	X	L	C	D	M
1	5	10	50	100	500	1000

Римскими цифрами традиционно обозначаются века, номера олимпиад, съездов и других подобных событий. Иногда они используются в кинематографических титрах для обозначения года выхода фильма.

Правила записи чисел римскими цифрами очень просты. Если большая цифра стоит перед меньшей, то они складываются; если меньшая цифра стоит перед большей, то меньшая вычитается из большей.

XXXII = 10 + 10 + 10 + 1 + 1 = 32

XIX = 10 + (10 – 1) = 10 + 9 = 19

MCMLXXIV = 1000 + (1000 – 100) + 50 + 10 + 10 + (5 – 1) = 1974

纸 Дополнительно

Международный стандарт ISO 8601 установил числовые представления даты и времени. В соответствии с этим стандартом дата обозначается как YYYY-MM-DD (четыре цифры года, две цифры месяца и две цифры дня), а время — как HH:MM:SS (по две цифры часов, минут и секунд). Часы записываются в 24-часовой системе, причем полночь обозначается *00:00:00*, а не *24:00:00*.

2003-08-15	15 августа 2003
17:40:59	17 часов 40 минут 59 секунд

Это в основном относится к представлению дат и времени в технических устройствах. В текстах, предназначенных для чтения человеком, в отечественной типографике предпочтительнее употреблять традиционную запись, например: *15.08.2003*, *17.40*. Способ указания минут в верхнем индексе (12^{30}) следует считать устаревшим.

В английском языке для представления времени иногда используется «британская» 12-часовая система, в которой сокращение «a. m.» обозначает «до полудня» (от латинского «ante meridiem»), а сокращение «p. m.» — «после полудня»

(от латинского «post meridiem»). Например: *7 а. т.* (7 часов утра), *4:30 р. т.* (16:30).

Словесная форма записи даты в русском и английском языках различается:

Русский язык	**Английский язык**
15 августа 2003, пятница	Friday, August 15, 2003

Примечание

Не путайте букву «О» с цифрой «0», а букву «З» — с цифрой «3». Также различайте строчную букву «l» (эль — латинская), прописную «I» (и — латинская) и цифру «1». С этими знаками часто возникает путаница при вводе регистрационных имен, паролей, адресов электронной почты и т. д.

2.1.7. Специальные знаки

Глаза привыкают во мраке, —
И вот выступают сквозь мглу
Какие-то странные знаки
На сводах, стенах и полу.

Ф. Сологуб

Знаки номера (№), параграфа (§) и авторского права (©) отделяются пробелом от последующего числа или слова. Знаки номера и параграфа без чисел не употребляются.

Правильно	**Неправильно**
дом № 7	дом №7
см. § 11	см. §11
Этот номер не пройдет!	Этот № не пройдет!

Дополнительно

Согласно Закону Российской Федерации об авторском праве и смежных правах, знак охраны авторского права состоит из трех частей: символа ©, имени или наименования обладателя исключительных авторских прав, года первого опубликования произведения. Эти части следует записывать именно в такой последовательности, причем между именем и годом нужна запятая.

Правила употребления, записи и оформления знака охраны авторского права содержатся в ГОСТ Р 7.0.1–2003.

Не следует использовать вместо знака охраны авторских прав суррогат (с).

Правильно	Неправильно
©	(с), (С)
© Александр Пушкин, 1825	© Александр Пушкин 1825 © 1825 Александр Пушкин © 1825 © Александр Пушкин

Примечание

Не следует использовать вместо знака номера латинскую букву «N» или решетку (#). Решетка как обозначение номера применяется в английском и некоторых других языках, но не в русском.

Знаки единиц измерений, записанные в верхнем индексе (′ , ″ , ° и др.) не отделяются пробелом от предшествующего числа.

Правильно	Неправильно
10′25″	10 ′ 25 ″
40°	40 °

Дополнительно

В отношении некоторых других знаков в справочных руководствах нет единогласия. Например, в [6] предписывается не разбивать пробелом конструкцию «°С», но отделять ее от предшествующего числа: *15 °С*. А в [3], наоборот, знак градуса в этом случае пишется слитно с числом, а буква «С» отделяется пробелом: *15° С*. Похожая ситуация и со знаком процента (%): одни руководства рекомендуют отделять его от предшествующего числа, другие — нет.

Настоящим источником правил по этому поводу следует считать государственный стандарт ГОСТ 8.417–2002, который предписывает в обоих случаях *отделять знаки пробелами*.

Правильно	Неправильно
100 %	100%
15 °С	15°С

Обратите внимание, что если знак градуса употребляется без буквы «С», то он пишется слитно с предшествующим числом: *+10°*.

Примечание

Не забывайте о едином стиле. Сайт будет выглядеть неаккуратно, если в одних местах там основные кавычки — «елочки», а в других — «лапки», если в одних местах между числовыми пределами ставится длинное тире (как предписывают традиционные правила отечественного набора), а в других — короткое (следуя сложившейся практике). Во всех случаях, где официальные правила и стандарты отсутствуют, выбирайте принципы самостоятельно и придерживайтесь их.

Дополнительно

Да, и не путайте знак градуса с цифрой «0» в верхнем индексе. «1°» — это один градус, а «1^0» — это один в нулевой степени. Видите разницу?

Знаки доллара ($), евро (€) и других денежных единиц в тексте на русском языке не употребляются.

Дополнительно

В английском языке применяются специальные знаки денежных единиц — например, доллар ($), евро (€), фунт стерлингов (£). Эти знаки записываются перед цифрой и не отделяются от нее пробелом: *$7.99*, *€200*.

Знак номера (№) в английском языке не употребляется. Вместо него используется решетка (#) или сокращение No. с точкой и пробелом после нее, например: *No. 7*.

В англоязычных странах часто используются такие единицы длины, как футы и дюймы. Для их обозначения употребляются те же знаки, что для минут и секунд. Не путайте эти знаки с прямыми кавычками.

	Правильно	**Неправильно**
Фут, минута	′	'
Дюйм, секунда	″	"

В английском языке распространен *амперсанд* (&), заменяющий слово «and» («и») — чаще всего в заглавиях, на логотипах и в других подобных ситуациях: *Barnes & Noble* (компания, торгующая книгами и другими товарами), *Dungeons & Dragons* (компьютерная игра). Амперсанд нельзя отрывать переводом строки от предыдущего слова и нежелательно отрывать от следующего. В сплошном тексте амперсанд не употребляется.

2.1.8. Сокращения

— Это Изнакурнож?
— Что?
— Я говорю, это изба на курногах или нет? Кто говорит?

А. Стругацкий, Б. Стругацкий.
Понедельник начинается в субботу

В печатных изданиях к сокращениям обычно прибегают для экономии бумаги. Сокращения особенно широко применяются в словарях и энциклопедиях, распространены в справочной литературе. Однако на сайтах сокращения такого характера обычно не оправданы. Экономии бумаги здесь не требуется, а текст без сокращений читается легче и быстрее. Поэтому, даже если вы цитируете на сайте статью из словаря, расшифруйте все сокращения.

Тем не менее, бывают ситуации, когда сокращения в тексте на веб-странице целесообразны. Рассмотрим несколько возникающих при этом сложностей.

Слова «год» и «век», следующие за числом, принято сокращать до одной буквы (с точкой после нее): *1884 г.*, *IV в. до н. э.* Во множественном числе «года» и «века» после чисел сокращаются соответственно до «гг.» и «вв.»: *IX–XI вв.*, *30-е гг.*

Постепенно распространилась практика как в устной, так и в письменной речи опускать слово «год», если из контекста понятно, о чем идет речь: *1 апреля 1977*, *в 2001*. Эта практика особенно уместна для заголовков, пунктов меню и других фрагментов текста, где важна краткость.

При порядковых числительных, обозначенных арабскими цифрами, падежное окончание должно быть однобуквенным, если предпоследняя буква в слове — гласная, и двухбуквенным — если согласная. Например, в слове «пятого» предпоследняя буква «г», следовательно, сокращение будет «5-го».

Правильно	**Неправильно**
2-й (второй), 7-х (седьмых), 10-я (десятая)	2-ой, 7-ых, 10-ая
3-го (третьего), 8-му (восьмому), 30-ми (тридцатыми)	3-его, 8-ому, 30-ыми

Не рекомендуется использовать такие сокращения там, где грамматическая форма числительного может быть непонятна из контекста, например: *5-м* (*пятом* или *пятым*?), *10-й* (*десятый* или *десятой*?).

Дополнительно

Падежное окончание нужно записывать не всегда. Например, при обозначении даты оно не требуется:

Правильно	Неправильно
15 августа 2003	15-го августа 2003

После сокращений «млн» (миллион) и «млрд» (миллиард) точка не ставится: *5 млн экземпляров*, *2 млрд жителей*. На сокращение «тыс.» (тысяча) это правило не распространяется.

Сокращения от единиц измерения употребляются только вместе с числом и записываются по правилам метрической системы, без точки: г (грамм), м (метр), с (секунда) и т. д. (Если числа нет, то единицы измерения записываются полностью: *несколько граммов*, *измеряется в метрах*.)

Единицы измерения, образованные от фамилий, в сокращенной форме обычно пишутся с прописной буквы: *Н* (ньютон), *Па* (паскаль), *Гц* (герц), но в полной форме — только со строчной.

Десятичные приставки записываются по правилам метрической системы: в частности, приставки «милли-» и «кило-» сокращаются строчными буквами «м» и «к» (*мм*, *кг*), а приставки «мега-», «гига-», «тера-» — прописными буквами «М», «Г», «Т» (*МПа*, *ГГц*).

Дополнительно

Единицы измерения информации вызывают существенные затруднения. Во-первых, не существует общепринятых стандартизованных сокращений от слов «бит» и «байт». Впрочем, сейчас распространяется тенденция слово «бит» не сокращать, а слово «байт» сокращать до одной буквы «б». По мнению сторонников такой практики, это не должно вызвать путаницы: обычные пользователи имеют дело с измерением информации в байтах, а не в битах, поэтому они не примут «б» за бит — точно так же, как мы однозначно понимаем запись «длиной 10 м», не принимая «м» ни за мили, ни за микроны.

Тем не менее, пока эта практика не стала общепринятой, рекомендуется всюду, где есть опасность разночтений, не сокращать «бит» и «байт». При этом стоит помнить, что это существительные мужского рода, и они склоняются по падежам (табл. 2.1).

Таблица 2.1. *Склонение существительных «бит» и «байт»*

Падеж	Бит		Байт	
	Единственное число	**Множественное число**	**Единственное число**	**Множественное число**
Именительный	бит	биты	байт	байты
Родительный	бита	битов	байта	байтов
Дательный	биту	битам	байту	байтам
Винительный	бит	биты	байт	байты
Творительный	битом	битами	байтом	байтами
Предложный	бите	битах	байте	байтах

На первый взгляд, эта таблица не вызывает никаких разногласий. В самом деле, никто не будет спорить, что словосочетания вроде «можно пожертвовать десятью *байтами* для улучшения читаемости кода» или «не забывайте о двух *битах* состояния» конструируются именно таким образом. Но сложности возникают, если мы хотим сказать «12 бит»… нет, «12 битов». Или все-таки «12 бит»? С одной стороны, можно рассуждать формально: «двенадцать» управляет ответом на вопрос «кого-чего», нужен родительный падеж, значит, «12 битов». В самом деле: мы сказали бы «12 стульев» (а не «12 стул»), «12 метров», «12 ньютонов»… Но, с другой стороны, в этой же ситуации не станут склоняться «герц», «ватт» и «ампер»: «12 ваттов» выглядит нелепо, а слово «герц» в таком падеже и вовсе невозможно представить.

Что же делать с «битами» и «байтами»? Однозначно можно будет сказать, когда сформируется устойчивая традиция употребления этих слов, как это произошло со словами «метр» и «герц». Автор рискнет предположить, что эта традиция приведет к несклоняемому варианту: «12 бит», «256 байт». Но, пока традиции не сложилось, рекомендуется считать оба варианта допустимыми и следить только за тем, чтобы в одном и том же тексте придерживаться единообразия: либо всюду «12 байт», либо всюду «12 байтов».

Вторая проблема, касающаяся единиц измерения информации, — это кратные единицы: килобайты, мегабайты и т. д. Прежде всего, несмотря на метрические приставки «кило-» и «мега-», — это не десятичные, а двоичные единицы: 1 килобайт равен 1024 (2^{10}) байтам, а не 1000. Иногда это приводит к путанице: если с «килобайтом» все однозначно, то скажем, «килобит» в разных источниках обозначает то 1000, то 1024 бита. Производители устройств хранения информации и поставщики услуг цифровой связи иногда указывают объемы своих услуг из расчета 1 килобайт = 1000 байт (не акцентируя на этом внимание потребителя). Это маркетинговый трюк: например, если жесткий диск хранит 100 000 000 байт информации, то при «честном» подсчете это получается 95 гигабайт, а если считать десятичными величинами — целых 100 гигабайт!

Но к тематике этой книги имеет большее отношение не то, как считать килобайты, а то, как сокращать их при записи. Грамотным вариантом является тради-

ционное сокращение приставки «кило-» и ей подобных: *кб* или *кбайт*, *Мб* или *Мбайт*, *Гб* или *Гбайт* и т. д. Тем не менее, на сайтах часто можно встретить сокращения «Кб», «Кбайт», «КБайт» и «МБ». Сторонники таких обозначений считают, что раз килобайты и мегабайты — не десятичные, а двоичные единицы, то к ним можно не применять правила сокращения, справедливые для десятичных единиц. Пожалуй, это заблуждение. В самом деле, оттого, что в XIX веке слово «пароход» сначала обозначало наземное средство транспорта, а потом стало обозначать водное, — правила грамматики русского языка не перестали на него действовать.

Не забывайте, что между числом и обозначением единицы измерения, независимо от сокращения, обязательно ставится пробел: 128 Мб, 100 кбит/с (а не 128Мб, 100кбит/с).

Помните, что в традиционных сокращениях (*и т. д.*, *и т. п.*, *т. е.*, *до н. э.*, *в т. ч.*, *у. е.*) сокращенные слова разделяются пробелами, как и полные.

Дополнительно

В английском языке для порядковых числительных используются традиционные сокращения с наращениями *st*, *nd*, *rd*, *th*. Эти наращения обычно набирают в верхнем индексе: 1st, 2nd, 3rd, 4th и т. д.

Если сокращение или специальный знак относится к нескольким числам в перечислении или диапазоне, его ставят только у последнего числа.

Правильно	**Неправильно**
5, 9, 12, 16 м	5 м, 9 м, 12 м, 16 м
72, 74, 75 %	72 %, 74 %, 75 %
600–750 Мб	600 Мб – 750 Мб

Примечание

Аббревиатура «вуз» в современном русском языке записывается строчными буквами и склоняется по падежам: *приемная комиссия вуза*, *список вузов Москвы*.

2.1.9. Переносы

> Ни в коем случае нельзя делать переносов, искажающих смысл слова, неблагозвучных или создающих двусмысленные толкования, отвлекающие читателя от правильного восприятия содержания (например, *бри-гады в поле, про-раб* на стройке, *проце-дура*, *по-беда* коллектива и т. п.).
>
> *М. Шульмейстер. Ручной набор*

В русском языке есть много правил о том, как нужно переносить слова. Но на сайтах переносы внутри слов обычно не применяются, поэтому актуальны, наоборот, правила о том, как *не нужно* переносить.

Браузеры сами следят за тем, чтобы не разрывать строку посередине слова или числа, чтобы не отрывать знак препинания от предшествующего слова и т. д. Но во многих случаях браузеры все-таки разрывают строку в неподходящих местах. Разработчик сайта должен это предотвратить. О технических способах установки и предотвращения перевода строки будет рассказано в *главе 5*, а здесь мы перечислим некоторые правила о запрете разрыва строки в русском языке:

- нельзя разрывать строку там, где не ставится пробел. В частности, не может быть перевода строки перед любым знаком препинания, после открывающей или перед закрывающей скобкой (кавычкой), перед дефисом или после него, перед коротким тире (между числами) или после него. Если перенести строку по дефису, то читатель может ошибочно подумать, что это не дефис, а знак переноса;
- можно разрывать строку после длинного тире, но перед ним — никогда. (За исключением ситуаций вроде прямой речи, когда предложение начинается с этого тире.) После короткого тире (между словами) разрыв не рекомендуется;
- нельзя разрывать сокращенные выражения *и т. д.*, *и т. п.*, *т. е.* и им подобные;

> **Дополнительно**
>
> Не рекомендуется оставлять в конце строки сокращение, оканчивающееся точкой: читателю может показаться, что на этом месте кончилось предложение. Также не рекомендуется разрывать строку перед словом, начинающимся с прописной буквы.

- при переносе нельзя отрывать инициалы от фамилии и друг от друга. Нельзя также отрывать имена собственные от предшествующих сокращений (*тов. Иванов*, *г. Тверь*);

- нельзя разрывать число и относящееся к нему сокращенное наименование (*250 кг*, *2004 г.*). Нельзя отрывать знаки номера (№) и параграфа (§) от относящегося к ним числа;
- нельзя оставлять в конце строки предлоги, союзы и частицы, начинающие предложение. Нельзя также оставлять в конце строки однобуквенные предлоги и союзы, в т. ч. стоящие в середине предложения;

Дополнительно

Не рекомендуется разрывать строку после любого однобуквенного слова, а также после любого служебного слова (союза, предлога, частицы).

- нельзя отрывать частицы *же*, *бы*, *ли* от связанного с ними слова;

Дополнительно

Не рекомендуется разрывать устойчивые словосочетания, например, *так как*, *потому что*, *то есть*, *так сказать*, *все равно*, *так точно*, *само собой*.

- нельзя разрывать многоразрядные числа. Согласно правилу из *разд. 2.1.6*, такие числа должны разбиваться пробелами, но разрывать строку на месте таких пробелов недопустимо.

Дополнительно

На самом деле, существует еще множество ситуаций, когда разрыв строки в неподходящих местах ухудшает восприятие и может исказить понимание текста. Например, характерную для математической литературы фразу: «Рассмотрим функцию *f(x)*, где *x* — натуральная переменная» не рекомендуется разрывать перед *f(x)* и перед *x*. В словосочетании «обновите ваш компьютер» строку гораздо лучше разорвать перед «ваш», чем перед «компьютер».

В прошлом профессиональные наборщики отслеживали такие ситуации и предотвращали разрывы строк не только там, где они запрещены формальными правилами, но и там, где они нежелательны по смыслу. К сожалению, сейчас эта практика утеряна. Современные издатели привыкли полагаться на компьютерные технологии, забывая о том, что полностью заменить человека компьютер пока не может.

Неправильно

и‖т.‖д., и‖т.‖п.

когда‖-‖нибудь, меч‖-‖кладенец, Немирович‖-‖Данченко

1799‖–‖1837, рейс Москва‖–‖Одесса

Знание‖— сила

Л.‖Н.‖Толстой, Дж.‖Леннон

проф.‖Преображенский, пос.‖Переделкино, ул.‖Судостроительная, п.‖2.1.9

5000‖км, 60‖Гб, №‖17, 15‖августа

В‖лесу... На‖границе... И‖дольше века... В‖хвост и‖в‖гриву...

так‖же, как‖бы, то‖ли

1‖234‖567

Примечание

Иногда кажется, что эти многочисленные правила совершенно невозможно запомнить. На самом деле, ничего сложного здесь нет. Вы знаете, что многие люди пишут грамотно по интуиции, совершенно не задумываясь ни о каких правилах. Это вовсе не наследственное качество и не сверхъестественная способность (как кажется некоторым). «Врожденная грамотность» развивается на опыте чтения грамотно написанных и оформленных текстов, она обычно присутствует у тех, кто с детства привык читать качественно изданные книги. Хороший способ повысить свою типографскую грамотность — читать больше книг, выпущенных с соблюдением правил. Если вы привыкнете к правильным кавычкам-«елочкам», правильным пробелам и другим знакам, то автоматически будете расставлять эти знаки в своем тексте. Впрочем, хорошие книги полезно читать не только по этой причине.

2.1.10. Имена собственные, наименования

> У нас вся третья эскадра, все офицеры и матросы, — все были с престранными фамилиями: Помойкин, Ярыжкин, Перепреев, лейтенант. А один мичман, и даже хороший мичман, был по фамилии просто Дырка. И капитан, бывало: «Эй ты, Дырка, поди сюда!»
>
> *Н. Гоголь. Женитьба*

К *именам собственным* относятся имена и прозвища людей, клички животных, географические и астрономические названия. Все слова в составе собственных имен (кроме служебных слов и родовых терминов) пишутся с прописной буквы: *Александр Сергеевич Пушкин*, *Ричард Львиное Сердце*, *Ростов-на-Дону*, *Красная площадь*.

Дополнительно

В написании собственных имен, особенно иностранных, иногда возникают трудности. Вот некоторые их них:

- сложные имена в одних справочных изданиях рекомендуется записывать раздельно (*Эрих Мария Ремарк*, *Джон Рональд Руэл Толкин*), а в других — через дефис (*Жан-Жак Руссо*, *Жан-Поль Бельмондо*). Скорее всего, в данном случае имеет смысл ориентироваться на написание в языке оригинала. Если между частями сложного имени решено ставить дефис, он сохраняется и между инициалами: *Ж.-Ж. Руссо*. В этом случае сложное имя склоняется как целое: *произведения Жан-Жака Руссо* (а не *Жана-Жака*);
- артикли, предлоги, частицы «да», «де», «ван», «фон» и другие обычно принято писать со строчной буквы и отдельно от других частей. Однако в некоторых случаях эти частицы слились с фамилией или приобрели устойчивое написание в русском языке: *Ван-Гог*, *Эль Греко*, *Декарт*, *Фонвизин*;
- частицу «дон» в именах «Дон Кихот» и «Дон Жуан» следует писать с прописной буквы. Но справочник [1] предписывает ставить в этих именах дефис и склонять как единое целое, а справочник [6] — писать раздельно;
- некоторые имена, хотя и пишутся раздельно, традиционно склоняются как единое целое. Например, «Робин Гуд»: мы говорим: *легенда о Робин Гуде*, *последователи Робин Гуда*.

Дополнительно

В английском и некоторых других иностранных языках распространена традиция записывать псевдоним или прозвище человека в кавычках между именем и фамилией: *John “Stranger” Smith*, *David “Zeb” Cook*. В русском языке такая запись неграмотна. Во-первых, псевдонимы и прозвища, как и другие личные имена, по-русски не пишутся в кавычках. Во-вторых, в русском языке место между именем и фамилией уже занято: там обычно находится отчество. В русской типографской практике псевдоним традиционно записывается после фамилии в скобках: *Евгений Максимов (Орландо)*, *Владимир Ульянов (Ленин)*. Возможны и другие формы, не противоречащие грамматике, например: *Сергей Винокуров по прозвищу Сокол*.

К *наименованиям* относятся названия учреждений, организаций, средств массовой информации, литературных произведений, праздников и т. д. Существует общее правило употребления прописных букв в наименованиях:

Внимание!

В наименованиях, состоящих из нескольких слов, с прописной буквы пишется первое слово, а также имена собственные, входящие в состав наименования.

Правильно	Неправильно	Примечания
Второе творческое объединение	Второе Творческое Объединение	Слова «творческое» и «объединение» не являются именами собственными, поэтому их не нужно писать с заглавной буквы
Центральный банк России	Центральный Банк России	«Россия» — имя собственное, а «банк» — нет
Московский государственный университет	московский государственный Университет	
Генеральный секретариат Совета Европы	Генеральный секретариат совета Европы	«Совет Европы» — само по себе наименование. Если одно наименование входит в другое, его написание не меняется
Новый год	Новый Год	

На сайтах наблюдается тенденция злоупотреблять прописными буквами, помещая их там, где должны быть строчные. Но некоторые разработчики, наоборот, не используют прописных букв вовсе, объясняя это особенностями творческого стиля. На самом деле, правильное употребление прописных букв помогает чтению, направляет внимание читателя. Исследования показывают ([12], с. 116; [13], с. 157), что если текст написан одними прописными или одними строчными буквами, то читатель тратит больше времени на его восприятие и быстрее устает.

Примечание

На логотипах, эмблемах, заставках и в других случаях художественного оформления текста допустимо отступать от некоторых типографских правил. Например, на эмблеме может быть красивым стилизованным шрифтом написано: «Клуб Любителей Поэзии». Но если это название употребляется в обычном тексте, его нужно привести в соответствие с правилами: «Клуб любителей поэзии».

Несмотря на наличие общего правила, иногда все-таки возникают затруднения в употреблении прописных и строчных букв. Рассмотрим их подробнее:

- существительные и прилагательные, обозначающие географическую, народную и национальную принадлежность, пишутся со строчной буквы. Например, существительные: *русский*, *немец*, *якут*, *папуас*, *москвич*, *харьковчанин*, *сибиряк*; прилагательные в словосочетаниях: *английский язык*, *белорусский народ*, *поволжский городок*, *еврейская песня*, *подмосковные вечера*;

- дни недели и названия месяцев пишутся со строчной буквы: *1 апреля, среда*; *15 августа 2003 года*. Это правило сохраняется при наименовании праздников: *Первое мая*, *Восьмое марта*, *Вербное воскресенье*;

Дополнительно

Исключение делается только тогда, когда название праздника начинается с цифры. В этом случае обозначение месяца пишется с прописной буквы: *7 Ноября*, *9 Мая*.

- прилагательные, образованные от имен собственных, пишутся со строчной буквы: *пушкинский стиль*, *викторианская эпоха*, *сталинские лагеря*;
- родовые наименования, такие, как «улица», «станция», «поселок» и тому подобные, пишутся со строчной буквы: *улица Металлургов*, *бульвар Яна Райниса*, *поселок Переделкино*, *бухта Провидения*.

  В тех случаях, когда бывшие родовые наименования потеряли свое собственное значение, они также пишутся с прописной буквы: *Серпуховский Вал* (улица), *Канин Нос* (мыс), *Марьина Роща* (район), *Море Дождей* (область на Луне);

Примечание

Слово «Интернет» — имя собственное мужского рода. Пишется с прописной буквы, без кавычек, склоняется по падежам: *искать в Интернете*, *службы Интернета*. Не нужно писать «сеть Интернет» или записывать это слово в русском тексте английскими буквами.

- когда слова «север», «юг», «запад», «восток» обозначают части света, географические области, они пишутся с прописной буквы: *народы Севера*, *Восток — дело тонкое*. В значении направления они пишутся со строчной буквы: *повернул на запад*, *на юге Московской области*;
- когда слова «земля», «луна», «солнце» обозначают астрономические объекты, они пишутся с прописной буквы: *расстояние от Земли до Солнца*, *программа исследований Луны*. В нетерминологическом значении они пишутся со строчной буквы: *выть на луну*, *искать по всей земле*;
- слова «Бог», «Господь» и образованные от них прилагательные обычно пишутся с прописной буквы, если речь идет о едином верховном существе: *творение Бога*, *во славу Божью*. Эти слова пишутся со строчной буквы, если обозначают божества политеистических религий (*боги Олимпа*, *бог Перун*) или употребляются в устойчивых разговорных словосочетаниях: *не приведи господь*, *ей-богу*, *бог знает что*, *ради бога*;

Дополнительно

В духовной литературе существуют правила и традиции, связанные с написанием слов «Бог», «Господь» и других религиозных терминов. Но в светском контексте не следует злоупотреблять прописной буквой, чтобы не получалось совершенной нелепицы вроде «все у вас не слава Богу».

- в текстах официальных документов, договоров, актов в обозначениях лиц все слова, кроме служебных (предлогов, союзов, частиц) пишутся с прописной буквы: *Высокие Договаривающиеся Стороны*, *Чрезвычайный и Полномочный Посол*. Традиционным элементом в договорах и других подобных документах являются выражения вроде: *Общество с ограниченной ответственностью «Компания АВС», именуемая в дальнейшем Заказчик...*

  Это правило не дает оснований употреблять прописные буквы вне официального контекста. Например, в информационном сообщении: *заказчик не выполнил условия контракта, поэтому исполнители подали в суд*;

- в названиях литературных и музыкальных произведений, средств массовой информации, как и в других наименованиях, с прописной буквы пишется первое слово, а также имена собственные.

Правильно	Неправильно
Двенадцатая ночь	Двенадцатая Ночь
Белое солнце пустыни	Белое Солнце Пустыни
Аргументы и факты	Аргументы и Факты
Подмосковные вечера	Подмосковные Вечера

Это правило часто нарушается — например, когда с прописной буквы пишут второе слово в названии телепрограммы «Поле Чудес» или фирмы «Седьмой Континент». Это допустимо как художественный прием в заставке или на логотипе, но если такое наименование упоминается в обычном тексте, его нужно привести в соответствие с требованиями грамматики.

Примечание

Местоимения «Вы» и «Ваш» пишутся с прописной буквы при обращении к *одному* лицу в официальных документах и личных письмах. В остальных случаях они пишутся со строчной буквы.

Это правило разработчики сайтов нарушают особенно часто, обращаясь к пользователям с прописной буквы. Из-за желания продемонстрировать показное уважение к пользователям текст выглядит несолидно, по-дилетантски.

Настоящее уважение разработчиков сайта к пользователям проявляется не в том, с какой буквы к ним обращаются, а в том, что сайт полезен и актуален,

что он предоставляет достоверную информацию и эффективные службы, что он удобен в работе.

Иногда возникают сомнения в том, стоит ли заключать то или иное наименование в кавычки. Перечислим несколько распространенных случаев, когда кавычки необходимы:

- названия литературных и музыкальных произведений, газет, журналов и других средств массовой информации, фильмов, телепрограмм: *«Московская правда»*, *«Чапаев и Пустота»*, *«Эхо Москвы»*. В некоторых специальных случаях (например, в подписи под эпиграфом) кавычки не требуются;
- марки автомобилей, самолетов и других транспортных средств: *«ВАЗ-2109»*, *«Боинг-747»*, *вагон «Яуза»*;
- собственные названия компаний, организаций, не включающие в себя родовое понятие: *партия «Единая Россия»*, *завод «Вымпел»*. Но: *Коммунистическая партия России*, *Новороссийский завод металлоконструкций* (включают в себя родовые понятия «партия» и «завод»);
- названия станций метро, остановок наземного транспорта: *«Автово»*, *«Рязанский проспект»*, *«Детская поликлиника»*.

Дополнительно

Названия аэропортов пишутся без кавычек: *аэропорт Домодедово*, *аэровокзал Внуково*. А в отношении железнодорожных станций и платформ определенности нет. Хотя некоторые справочники [6] рекомендуют записывать их без кавычек, логичнее было бы относиться к ним так же, как к станциям метро: *станция «Нижние Котлы»*, *платформа «43-й км»*.

Во всех перечисленных случаях кавычки требуются только тогда, когда наименование употребляется в тексте. Кавычки нужны для его выделения среди окружающих слов. Если окружающих слов нет, и наименование используется само по себе (например, название книги на ее обложке, заголовок статьи над текстом или марка автомобиля в ячейке таблицы), то кавычки не нужны.

Дополнительно

На сегодняшний день нет никаких правил о том, как синтаксически выделять в тексте адреса сайтов. Вот лишь некоторые из вариантов, которые встречаются в печати и на веб-страницах:

- На сайте www.ixbt.com появился обзор…
- На сайте Www.Ixbt.Com появился обзор…
- На сайте «www.ixbt.com» появился обзор…
- На сайте “www.ixbt.com” появился обзор…
- На сайте *www.ixbt.com* появился обзор…

- На сайте **Www.ixbt.com** появился обзор…
- На сайте www.ixbt.com появился обзор…

Для текста, размещенного на веб-странице, автор рекомендует последний вариант: просто делать адрес сайта гиперссылкой на этот сайт и не применять никакого другого выделения. Это самый простой способ: он требует меньше всего усилий по кодированию, не вводит читателя в заблуждение заглавными буквами и дает возможность сразу перейти по ссылке.

Вы можете применять и какой-нибудь другой способ, если на то есть убедительные причины. Но не забывайте всюду придерживаться единого стиля.

纸 *Дополнительно*

В английском языке прописные буквы употребляются существенно чаще, чем в русском. С прописной буквы пишутся:

- названия месяцев и дней недели: *the 1st of April*, *closed on Saturdays*;
- существительные и прилагательные, образованные от имен собственных (в частности, от географических наименований): *He's American*, *my French aunt*, *English-Russian dictionary*;
- названия должностей, титулов и т. д. перед именами и фамилиями: *Colonel Smith*, *Doctor Robert*, *Sir Paul McCartney*.

В наименованиях, в отличие от русского языка, с прописной буквы пишутся все слова, кроме служебных: *National Aeronautics and Space Administration*, *Daily Telegraph*, *All You Need Is Love*.

Для выделения названий в тексте на английском языке часто используются не кавычки, а курсив:

- Jacob Nielsen's *Homepage Usability* is an excellent book;
- The *Wailers*' album *Catch a Fire* was released in 1972.

2.1.11. Заголовки и заголовочные надписи

> Могучий Володя Почкин, как замредактора, исполняющий в основном полицейские обязанности, схватил Дрозда за толстый загривок, согнул его пополам и принялся тыкать носом в газету, приговаривая: «Заголовок где? Где заголовок, Дроздилло?»
>
> *А. Стругацкий, Б. Стругацкий. Понедельник начинается в субботу*

К заголовочным надписям относятся:

- ❒ заглавия, заголовки рубрик текста;
- ❒ заголовки таблиц;

- подписи к иллюстрациям, схемам, чертежам;
- элементы интерактивных меню:
  - в традиционных программных продуктах — например, пункты верхнего меню: **Файл**, **Правка** и т. д.;
  - на сайтах — например, ссылки на основной панели навигации.

Первое слово в заголовочной надписи пишется с прописной буквы.

В конце заголовочной надписи точка не ставится. Если в конце заголовочной надписи необходим другой знак (вопросительный, восклицательный, многоточие), он ставится.

Если заголовочная надпись состоит из нескольких предложений или структурных частей, между ними ставится точка:

> История первая. Суета вокруг дивана
>
> Глава X. Сингулярные интегралы. Тригонометрические ряды. Выпуклые функции
>
> Рис. 14. И. Е. Репин. Запорожцы пишут письмо турецкому султану. 1878–1891

Заголовки граф таблицы подчиняются некоторым особым правилам. В большинстве случаев они ничем не отличаются от заголовочных надписей и тоже пишутся с прописной буквы. Но если в таблице используется многоступенчатый заголовок, то заголовки второй и последующих ступеней пишутся со строчной буквы в том случае, если они грамматически подчинены заголовку предыдущего уровня (рис. 2.1).

Бит	
Единственное число	**Множественное число**

а

Число	
единственное	**множест-венное**

б

Рис. 2.1. Заголовки граф второго уровня: *а* — грамматически не связаны с основным заголовком; *б* — грамматически подчинены основному заголовку

Примечание

Все, о чем говорится в этом разделе, относится к *типографике*. Типографика — это дисциплина, которая занимается оформлением текста типографскими средствами. Не путайте типографику с грамматикой, которая изучается в школе. Грамматика говорит о том, как правильно писать слова и где ставить знаки препинания. А типографика говорит, например, о том, каким шрифтом писать эти слова, как должны выглядеть знаки препинания, в каких случаях использовать то или иное выделение и т. д. Для успешной разработки текста на сайтах необходимо хорошо владеть обеими этими дисциплинами.

2.1.12. Перечни (списки)

> Собственно говоря, это не один список, а целых два. В первом, по большей части, мы находим имена женщин, внушивших наиболее серьезные чувства поэту. На последнем месте здесь поставлена Наталья — его будущая жена.
>
> *П. Губер. Донжуанский список Пушкина*

Перечень — очень удобная форма организации информации списочного характера. Пользователь скорее бегло просматривает веб-страницы, чем читает их, так что перечни, характерные своим внешним видом, особенно хорошо помогают донести информацию.

На сайтах в основном используются такие перечни, каждый элемент которых представляет собой отдельный абзац.

Перечни бывают:

- нумерованные (элементы обозначены последовательными буквами или цифрами) — обычно используются тогда, когда важен порядок элементов, а также когда необходимо ссылаться на отдельные элементы;
- маркированные (элементы обозначены маркерами, как в этом списке) — обычно используются тогда, когда порядок элементов не важен.

Синтаксическое оформление перечня зависит от его содержания.

Если элементы очень просты, состоят из одного или нескольких слов, без знаков препинания внутри:

- в нумерованном списке элементы обозначаются арабскими цифрами или строчными буквами с закрывающей круглой скобкой;
- каждый элемент начинается со строчной буквы;
- элементы разделяются запятыми.

Если элементы состоят из большого количества слов, имеют внутри знаки препинания, то действуют два первых правила, а разделяются элементы точкой с запятой.

Если элементы состоят из нескольких предложений:

- в нумерованном перечне элементы обозначаются арабскими или римскими цифрами или прописными буквами с точкой;
- каждый элемент начинается с прописной буквы;
- элементы разделяются точками.

На рис. 2.2 приведены варианты оформления перечней.

Знаки препинания:

- точка,
- запятая,
- точка с запятой,
- двоеточие,
- многоточие,
- восклицательный знак,
- вопросительный знак,
- дефис,
- тире,
- смешанные знаки.

Заголовки бывают:

1) тематические — слово или словосочетание, выражающее тему озаглавленной части (например: *Пушкин в Михайловском*);
2) родо-нумерационные и родо-литерные — состоят из обозначения рода рубрики (часть, глава, параграф) и пр. и номера или буквы (например: *Часть I*, *§ 9*);
3) смешанные (например, родо-нумерационно-тематический заголовок: *Часть I. Уроки словесности*);
4) прочие.

Афоризмы Козьмы Пруткова:

1. Никто не обнимет необъятного.
2. Смотри в корень!
3. Бди!
4. Не все стриги, что растет.
5. Что имеем — не храним; потерявши — плачем.
6. На чужие ноги лосины не натягивай!
7. Специалист подобен флюсу: полнота его одностороння.
8. Если на клетке слона прочтешь надпись «буйвол», не верь глазам своим.

а *б* *в*

Рис. 2.2. *а* — простой перечень: элементы начинаются со строчной буквы, разделяются запятыми; *б* — элементы перечня синтаксически усложнены, содержат знаки препинания, поэтому разделяются точкой с запятой; *в* — каждый элемент перечня — отдельное предложение, элементы начинаются с прописной буквы и завершаются точкой

При нумерации строчные буквы пишутся слитно с предшествующими цифрами: *рис. 1а*, *пункт IVб*.

Дополнительно

В обозначениях школьных классов употребляется прописная буква, которая заключается в кавычки: 10 «А», 7 класс «В».

Примечание

На сайтах невозможно выполнить все типографские требования, предъявляемые к печатной продукции. Например, в печати есть такое понятие, как «коридор»: он возникает на странице в том случае, если несколько пробелов в соседних строках случайно выстраиваются по одной вертикали или диагонали. Правила набора предписывают не допускать таких «коридоров». Но при наборе текста для веб-страницы гарантировать их отсутствие нельзя. Даже если у вас на экране «коридоров» нет, у какого-нибудь другого пользователя с другими

характеристиками системы и другими настройками браузера они вполне могут появиться.

Впрочем, если некоторые правила набора соблюдать не удается, это не должно мешать соблюдению всех остальных.

2.1.13. Выделения

Умные речи подобны строкам, напечатанным курсивом.

К. Прутков

К выделениям относятся полужирное и курсивное начертание, разрядка, изменение гарнитуры шрифта, гиперссылки и т. д. О видах выделения и их употреблении на сайте будет рассказано в *главе 5*, а здесь пойдет речь о синтаксическом оформлении некоторых видов выделений.

Заголовок в подбор. Это тот самый заголовок, с которого начинается эта строка. После него ставится точка, которая выделяется тем же способом, что и сам заголовок (в данном случае — полужирным шрифтом).

Предложение внутри текста. Если предложение внутри текста выделено целиком, то завершающий его знак препинания также выделяется. *Например, в этом предложении восклицательный знак выделен курсивом!*

Отдельные слова или словосочетания внутри предложения. Если несколько слов или словосочетаний подряд выделены как независимые объекты, то разделяющие их знаки препинания не выделяются:

В сокращениях *и т. д.*, *и т. п.*, *т. е.*, *до н. э.*, *в т. ч.*, *у. е.* не забывайте ставить пробел между сокращенными словами.

Обратите внимание: каждое сокращение в отдельности выделено курсивом, но запятые между сокращениями имеют обычное прямое начертание.

Если словосочетание выделено как единый объект, то знаки препинания внутри него также выделяются.

Знак препинания, стоящий после выделенного фрагмента, не выделяется.

Правильно	Неправильно
Кегль, или размер шрифта...	*Кегль,* или размер шрифта...
	(запятая выделена курсивом)
Выделение **полужирным шрифтом**.	Выделение **полужирным шрифтом.**
	(точка выделена полужирным шрифтом)

Выделения в скобках и кавычках. Если выделенное слово, словосочетание или предложение окружено скобками, сами скобки не выделяются. Кавычки в этом случае, наоборот, выделяются так же, как заключенный в них фрагмент.

Правильно	Неправильно
Размер (*кегль*) шрифта...	Размер *(кегль)* шрифта...
фильм **«Властелин колец»**	фильм «**Властелин колец**»

Дополнительно

Не забывайте, что гиперссылка — также один из видов выделения, и на нее распространяются все правила синтаксического оформления.

Правильно	Неправильно
Наши адреса: info@bhv.ru, mail@bhv.ru	Наши адреса: info@bhv.ru, mail@bhv.ru
На моем сайте (www.philigon.ru) можно найти...	На моем сайте (www.philigon.ru) можно найти...

Примечание

Правила типографского оформления текста, как и правила грамматики, в каждом языке свои. Это необходимо учитывать при переводе.

Например, в английском языке для разделения уровней нумерации может использоваться не точка, а какой-нибудь другой знак — например, короткое тире:

Table 3–11. Document Type Declarations

Номер заголовка может не отделяться точкой:

2 Introduction to HTML 4

В русском языке и то, и другое будет неграмотным. Необходимо привести текст в соответствие с типографскими нормами того языка, на который он переводится:

Английский оригинал	Русский перевод
Table 3–11. Document Type Declarations	Таблица 3.11. Определения типа документа
2 Introduction to HTML 4	2. Введение в HTML 4

Это касается всех аспектов синтаксического оформления текста — в т. ч. пунктуации, скобок, кавычек, прописных букв. Вот несколько распространенных примеров «недоперевода».

- Те, кто привык работать с издательской системой TEX, часто пользуются стандартными для TEX английскими кавычками. Эта неприятность, например, постигла многие книги автора системы TEX Дональда Кнута, переведенные на русский язык. Текст в этих книгах перевели, а кавычки — нет.

- Изготовители музыкальных дисков нередко помещают на обложку список композиций, в названиях которых каждое слово написано с прописной буквы. Совершая эту ошибку, они иногда «превосходят» носителей английского языка, записывая с прописной буквы действительно *каждое* слово, включая даже предлоги, союзы и частицы.
- Бывает, что в тексте, написанном по-русски, названия книг, газет, музыкальных произведений и им подобные не заключаются в кавычки, а выделяются курсивом, как в английском языке. Эта ошибка часто встречается в переводах технической документации.

Соблюдение всех перечисленных правил при разработке текста для сайта требует знания технических тонкостей кодирования HTML и CSS. Этой теме посвящена *глава 5*. Но уже сейчас, вооружившись прочитанным материалом, вы легко можете оценивать грамотность и качество типографского оформления текста на сайте. Между прочим, это опровергает распространенный миф о том, что, прежде чем оценивать что-то, надо уметь это делать. (Автор вообще уверен, что этот миф придумали те, кто сам не умеет ни делать, ни оценивать.)

2.2. Новое — значит лучшее?
Современные явления в сетевом русском языке

Новые сапоги всегда жмут.

К. Прутков

— Не может, — вздохнул Зильберович. — Во время обеда он просматривает читалку.
— Чего просматривает?
— Ну, газету, — сказал раздраженно Лео. — Ты же знаешь, что он борется против иностранных слов.

В. Войнович. Москва 2042

Повсеместное развитие технологий Интернета породило немало новшеств в языке, которым пользуются разработчики и потребители этих технологий. Время покажет, какие из этих новшеств войдут в русский литературный язык, а какие отомрут. Эта глава предназначена в первую очередь для тех, кто привык к «классическому» русскому языку и пока еще плохо ориентируется в новых веяниях.

2.2.1. «Браузер» против «обозревателя»
Откуда берутся специальные термины, связанные с Интернетом

Но *панталоны*, *фрак*, *жилет*,
Всех этих слов на русском нет;
А вижу я, винюсь пред вами,
Что уж и так мой бедный слог
Пестреть гораздо б меньше мог
Иноплеменными словами...

А. С. Пушкин. Евгений Онегин

Непривычные грамматически в русском языке слова *кенгуру*, *какаду*, *пенсне*, *кашне*, *сальдо*, *колибри*, *чохохбили* и т. п. своими «концами» **у**, **е**, **и** не подходят к моделям существительных и поэтому остаются неосвоенными до конца... Слова, содержащие непривычные для русской фонетики звуки или сочетания звуков, также остаются недоосвоенными, например: clанг (с чуждым **l**), *Кёльн* (с чуждым сочетанием **кё**), *Тартарен*...

А. А. Реформатский. Введение в языковедение

Вы наверняка заметили, что в разных книгах и справочной документации к программным продуктам одни и те же вещи называются по-разному. То, что в одном источнике именуется «браузер», в другом становится «обозревателем»; то, что в одной книге — «узел», в другой — «сайт».

Возникает вопрос: какой же из вариантов правильный? Какому варианту следует отдавать предпочтение и почему?

Подробное и серьезное исследование этого вопроса относится к ведению лингвистики и может быть обнаружено в специальной литературе. В этой книге приводятся только самые общие соображения, которые должны помочь как при посещении сайтов, так и при их разработке.

Итак, чем же нужно руководствоваться при выборе русскоязычных терминов для области технологий Интернета?

Внимание!

Чем руководствоваться при выборе терминов?

- Норма.
- Традиция.

- Удачный русскоязычный эквивалент.
- Транскрипция или транслитерация.
- Ориентация на аудиторию.
- Благозвучность.

Норма

Если существует стандартный термин, закрепленный в нормативных словарях и других официальных документах, — следует предпочесть именно его. Например, не нужно вместо «программное обеспечение» писать «софт». Или, если идет речь о таблицах на веб-странице, следует употреблять нормативный термин «графа» (а вовсе не «столбец» или «колонка»).

Традиция

Технологии Интернета развиваются быстрее, чем выпускаются официальные словари. Поэтому в большинстве случаев не существует нормы, которой можно было бы руководствоваться. В этом случае на помощь приходит традиция.

Примечание

К настоящему времени сложилось употребление слова «Интернет», которое можно признать традиционным. Это слово является именем собственным (так как обозначает уникальный предмет) и поэтому пишется с прописной буквы, без кавычек. Оно имеет мужской род и склоняется по падежам: *Интернета*, *Интернету* и т. д. Другие способы использования этого слова в текстах на русском языке (*Internet*, *сеть «Интернет»* и т. д.) постепенно выходят из употребления.

Слово «Интернет» уже встречается в орфографических словарях и речи дикторов радио и телевидения.

Если нормативного термина нет, но есть устойчивая традиция употребления термина в технической документации, книгах и других публикациях, а также в устной речи и переписке, — следует придерживаться этой традиции. Например, программу для просмотра веб-страниц традиционно принято называть «браузер» (название «обозреватель» не прижилось), а элемент управления, который по-английски называется «checkbox» — наоборот, по-русски обычно называют не транслитерацией «чекбокс», — а «флажок».

Дополнительно

Интересная ситуация сложилась со словом «Web». Само это слово в значении «Всемирная Паутина» у нас не закрепилось. В самом деле, неестественно

звучит: «работа в Вебе» или «просмотр Веба» (рис. 2.3). А в качестве приставки это слово прекрасно чувствует себя в русском языке: слова «веб-страница», «веб-мастер», «веб-интерфейс» никого не раздражают.

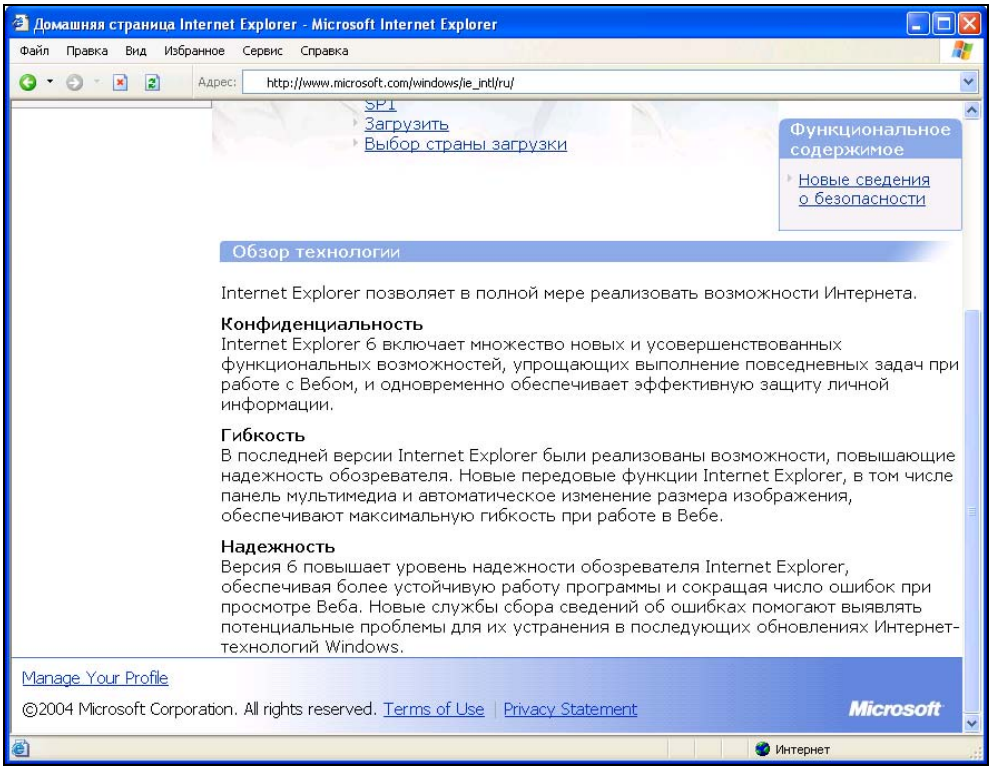


Рис. 2.3. На сайте **www.microsoft.com** отважились употреблять слово «Веб»

Русскоязычный эквивалент

Иногда приходится изобретать новые термины для понятий, которые еще не закрепились в русском языке. В этом случае прежде всего стоит попытаться найти подходящее слово из русского языка, а не заимствовать иностранное. Например, мы говорим и пишем «клавиатура», «указатель мыши», «дисковод», а не «киборд», «маус пойнтер» и «флоппи-драйв». Термины с русскоязычными корнями звучат для носителей русского языка естественнее и воспринимаются легче (табл. 2.2).

Таблица 2.2. *Некоторые термины Интернета*

Англоязычный термин	Предпочтительный эквивалент в русском языке	Другие употребляемые эквиваленты в русском языке
browser	браузер	броузер, обозреватель
click	щелкнуть	нажать, кликнуть
content	содержание	контент
download	загрузить	скачать
frame	фрейм	кадр
interface	интерфейс	
multimedia	мультимедиа	
newsgroup	телеконференция	группа новостей
radio button	переключатель	радиокнопка
screenshot	снимок экрана	скриншот
script	сценарий	скрипт
scrollbar	полоса прокрутки	скроллбар
site	сайт	узел
style sheet	таблица стилей	лист стилей
tag	тэг	дескриптор, контейнер
usability	удобство	юзабилити
World Wide Web	Всемирная Паутина	Веб, Сеть

Для терминов, приходящих к нам из иностранных языков, такой способ особенно часто применяется тогда, когда перевод слова на русский язык уже существует, хотя и применяется в другом значении. Например, это касается слов «mouse» («мышь») и «window» («окно»). В контексте компьютерных технологий эти слова сохранили тот же перевод, который был у них раньше.

Транскрипция или транслитерация

Конечно, не стоит возводить поиски русскоязычного эквивалента в ранг всеобщего закона. Если такого эквивалента подобрать не удается — нет ничего плохого в том, чтобы использовать транскрипцию или транслитерацию:

- *транскрипция* — запись слова одного языка буквами другого таким образом, чтобы максимально точно сохранить звучание исходного слова. Например, «браузер» — транскрипция английского слова «browser», «месье» — транскрипция французского слова «monsieur»;

- *транслитерация* — запись слова одного языка буквами другого таким образом, чтобы максимально точно сохранить написание исходного слова. Например, «броузер» — транслитерация слова «browser», имя «Якоб» — транслитерация от «Jacob» (по-английски оно произносится вовсе не «Якоб», а «Джейкоб»).

Например, путем транскрипции вошли в обиход слова «джойстик», «принтер» и «сканер». Чаще всего транскрипция или транслитерация используются для новых понятий, появляющихся в обращении одновременно с предметами, которые они обозначают.

Дополнительно

При транскрипции слов англоязычного происхождения часто встает вопрос о том, что делать с удвоенными согласными. Во многих словах, уже освоенных русским языком, две согласные превратились в одну: «scanner» — «сканер», «office» — «офис». Но в некоторых других — остаются две согласные: «trimmer» — «триммер», «rally» — «ралли». В настоящее время «в подвешенном состоянии» находятся, например, слова «баннер» и «апплет»: пока еще не сложилась однозначная традиция их употребления ни с одной, ни с двумя согласными. Автор предполагает, что в таких ситуациях нельзя заранее предсказать, на чем остановится выбор. До тех пор, пока слово не появляется хотя бы в нескольких авторитетных орфографических и толковых словарях, автор предпочитает сохранять двойную согласную. Впрочем, вы можете придерживаться и противоположного правила.

Кстати, слово «трафик» пишется с одной буквой «ф». Оно давно зафиксировано в словарях и используется не только в области технологий Интернета, но и, например, в автодорожной области: *трафик машин*, *трафик пешеходов*.

И уж точно не следует писать с удвоенной согласной те транскрибированные слова, которые и по-английски пишутся с одной, — например, «тонер» или «дилер».

Ориентация на аудиторию

Бывает, что разработчики сайтов включают в текст такие термины, которые, на их взгляд, лучше позволят привлечь и удержать потенциальных посетителей. Например, иногда считается, что «прайс-лист» звучит современнее, чем «прейскурант», а использование слова «скачать» вместо «загрузить» придаст сайту более «дружеский» характер. Противники такого подхода считают его признаком непрофессионализма и дурного тона. Но разработчику лучше избегать крайностей и в каждом конкретном случае принимать особое решение, уместен ли тот или иной термин на веб-странице.

Дополнительно

Неграмотность некоторых разработчиков проявляется в употреблении дефисных словосочетаний вроде «FTP-сервер», «HTTP-протокол», «ssh-клиент».

Такие обороты употребляются в английском языке: «NASA project», «CNN news», однако на русский язык они переводятся в обратной последовательности: «проект NASA», «новости CNN». С техническими терминами должно происходить то же самое: «сервер FTP», «протокол HTTP».

Это еще один пример того, как важно при переводе следовать правилам того языка, на который переводится текст.

Благозвучность

Выживают и закрепляются только те термины, которые органично вливаются в язык, которые легко запоминать и произносить. Например, вполне удобными оказались термины «тэг» и «сайт». А вот транслитерации таких слов, как «newsgroup» или «editor», в русском языке не прижились.

Ну и, конечно, не стоит транслитерировать популярные в англоязычной среде термины вроде «e-book» или «e-business», а также называть текстовый файл «raspis.doc».

Дополнительно

Не стоит употреблять избыточные конструкции. Например, совершенно ни к чему говорить «веб-сайт» («интернет-сайт») или «провайдер Интернета». Это все равно, что «музыкальная песня» или «горячий кипяток». Ведь слово «сайт» само по себе не имеет никакого другого значения, да и «провайдер» употребляется, только если мы говорим о поставщике услуг Интернета.

Примечание

У грамотного и литературно эрудированного разработчика обычно присутствует так называемое «чувство языка», которое с достаточной достоверностью позволяет предсказать, какие из новых терминов войдут в язык и закрепятся в нем, а какие вымрут или останутся профессиональным жаргоном.

Отсутствие единой официальной терминологии в области технологий Интернета оставляет разработчику сайтов простор для собственных решений. Единственное правило заключается в том, чтобы эти решения были достаточно продуманы и обоснованы. Плохо, если термин появился в тексте просто потому, что он первым пришел в голову разработчику или потому, что он моден в среде, где разработчик вращается (рис. 2.4).

Если вы относитесь к терминологии серьезно и ответственно — это как минимум приведет к тому, что вам не придется вскоре заново переписывать текст на своем сайте, а как максимум — к тому, что именно ваши термины и понятия войдут в русский литературный язык и станут широко употребляться.

Рис. 2.4. Отечественная поисковая система «Яндекс» популяризирует такие жаргонизмы, как «скачать» и «закачать»

2.2.2. Админ пингует хост
О специальной терминологии и сетевом жаргоне

— Шмутки есть? — спросил Бубновый Валет.
— Вещи на продажу есть? — перевел Тикк.
— Увы, — сказал я, — мы явились только с духовными ценностями.
— Чего этот чувой ботает? — удивленно переспросил Бубновый Валет.
— Он спрашивает, — перевел мне Тикк, — о чем ты говоришь?

Л. Лиходеев. Звезда с неба

В любой профессиональной области есть своя *специальная терминология*. Это слова и словосочетания, которые обозначают понятия, специфичные для данной области, и употребляются специалистами. Классическим примером является флотская терминология, в которой слово «банка» обозначает скамью, а «компас» произносится с ударением на последнем слоге.

Не является исключением и область компьютерных технологий. Непосвященному человеку слова «роутер» или «скрипт» ничего не скажут, а посвященные могут часами спорить о том, не лучше ли говорить «маршрутизатор» и «сценарий».

Однако надо отличать специальную терминологию от *профессионального жаргона*. Специальная терминология — это то, что зафиксировано в профессиональных словарях и справочниках, используется в нормативной документации. А жаргон — это разговорный язык, который используется в неформальной обстановке. Специальные термины чаще всего звучат офи-

циально, консервативно. А жаргонные слова и обороты — наоборот, просторечно и несерьезно. Специальный термин — «исходный код», жаргонный — «сырцы́», специальный термин — «синтаксический разбор», жаргонный — «парсинг».

Жаргонные слова в области компьютерных технологий образуются несколькими путями. Наиболее часто они являются калькой от англоязычного слова — «пинг» и «пинговать», «хаб» и «свитч», «сплэш» и «скринсейвер». Другой вариант — подбор удачного русского слова («звонилка», «скачать», «глючный», «зависнуть»). Третий путь, объединяющий в себе два предыдущих, — русское слово, похожее по звучанию на английский оригинал: «сырцы́», «мыло», «кликнуть».

Дополнительно

Отечественные острословы придумали интересные жаргонизмы для обозначения распространенных браузеров. Internet Explorer — сокращенно IE — сначала превратился из «IE» в «Иа», а потом стал именоваться просто «Осликом». Netscape давно величают «Нетшкафом», а для браузера Mozilla не пришлось изобретать никакого специального наименования: как говорится, «Мазила» — она и есть «Мазила».

Примечание

Автор рекомендует пользоваться жаргоном только в том случае, если вам не составляет труда при необходимости высказать то же самое и на нормативном языке. В противном случае вы рискуете остаться непонятым, если придется разговаривать с человеком, который не владеет компьютерным жаргоном.

Есть и четвертый способ: сокращение или искажение «обычного» термина: «админ», «сервак», «конфа». Наконец, применяются комбинированные способы образования жаргонизмов, приводящие к словам вроде «юзверь» или «Нафигатор».

Стоит ли использовать жаргон? Конечно, стоит! Но только там, где он действительно уместен, — то есть в неформальном разговоре между специалистами. Например, при настройке системы гораздо проще и эффективнее (с точки зрения выполняемой задачи) попросить коллегу-администратора «пропинговать хост», чем вспоминать, что по-русски это будет «прозвонить удаленный узел», а также убеждаться, что коллега тоже владеет не только жаргоном, но и нормативной терминологией.

Дополнительно

Кстати, английское «to ping» — это тоже своего рода жаргон. Слово «ping» в применении к компьютерным сетям — на самом деле аббревиатура

от «Packet InterNet Groper». Но эта аббревиатура удачно совпала с обычным словом «ping», среди значений которого — «трезвонить», «гудеть», «свистеть». С этим словом произошло приблизительно то же, что в русском языке со словом «click», которое трансформировалось в «кликнуть».

Но, к сожалению, жаргон часто используется там, где без него стоило обойтись. В технической документации и справочных руководствах для пользователя не место жаргону. Бывает, что, работая с англоязычной документацией, недостаточно квалифицированный переводчик переводит «router» как «роутер» (правильно — «маршрутизатор»), а «hosting» — как «хостинг» (правильно — «размещение»).

И, конечно, технический жаргон не должен появляться в тексте на сайте. Если только сайт не предназначен исключительно для специалистов, на нем следует обходиться общеупотребительной нормативной лексикой.

Дополнительно

На одном сайте для оптимального просмотра предлагалось «не блокировать попапы, не запрещать куки и включить Flash-плугин». Пользователь, показавший этот сайт автору, поначалу подумал, что от него хотят чего-то неприличного.

Кроме того, даже если пользователи понимают жаргон на сайте, такой сайт обычно выглядит несолидно и несерьезно. Информации, изложенной с употреблением жаргона, доверяют меньше, чем той, которая выражена нормативным языком.

Не путайте специальную терминологию разработчиков сайта и специальную терминологию пользовательской аудитории. Если вы, например, разрабатываете сайт, посвященный лингвистике, то на нем вполне уместны такие непростые слова, как «изоэмия», «агглютинация» и «синекдоха». Но специальные термины, которыми пользуются разработчики, а не посетители сайта («URL», «скрипт» и т. д.), здесь совершенно ни к чему. Или, например, если ваш сайт ориентирован на молодежную аудиторию, на нем могут в особом стилистическом контексте применяться жаргонные слова вроде «тусоваться», «фишка», «заценить» и тому подобные — но ни в коем случае не «хост», «кодинг» или «попап».

Внимание!

На сайте можно употреблять специальную терминологию, а иногда даже жаргон, которым пользуются *посетители* сайта. Но недопустимо употреблять терминологию и тем более жаргон *разработчиков*.

Вообще говоря, граница между специальной терминологией и жаргоном достаточно сильно размыта, особенно в бурно развивающейся области тех-

нологий Интернета. Многие слова, которые первоначально были жаргонными, закрепились в качестве специальных терминов — например, «браузер», «провайдер». Некоторые слова занимают промежуточное положение между специальной терминологией и жаргоном — «скрипт», «прокси-сервер», «юзабилити».

Примечание

Читая печатные издания, выпущенные всего несколько лет назад, можно посмеяться над архаичностью некоторых выражений: «компьютерная сеть „Internet“», «обозреватель страниц Internet», «электронные сообщения в формате „e-mail“». Пожалуй, сейчас все это звучит так же громоздко и старомодно, как «синематограф», «авиатор» или «электронно-вычислительная машина». Вы твердо уверены, что тексты с вашего сайта не будут вскоре приводить в качестве примера подобного «ретро»?

Язык постоянно изменяется и развивается; в нем соседствуют и общеупотребительная лексика («стол», «компьютер»), и специальная профессиональная терминология («изоэмия», «маршрутизатор»), и жаргон («тусовка», «админ»). Каждый из этих типов лексики вполне уместен в своей «экологической нише». Важно уметь отличать их друг от друга и применять только по назначению. Автор считает, что это умение — характерное качество всякого культурного и образованного человека.

2.2.3. Знаки времени
Новшества сетевой типографики

> И если пока еще и в наших строках остались грязные клейма ваших «здравого смысла» и «хорошего вкуса», то все же на них уже трепещут впервые зарницы Новой Грядущей Красоты Самоценного (самовитого) Слова.
>
> *Пощечина общественному вкусу*
> *(манифест футуристов, 1912)*

Новые сокращения

Человеку свойственно упрощать свою деятельность. В частности, это проявляется в сокращении слов в письменной речи, особенно когда требуется записывать быстро или не хочется писать слишком много. В современных средствах обмена информацией сокращения особенно часто применяются в чатах (где важна оперативность) и в сообщениях SMS (в основном из-за неудобного набора).

Однако стиль сокращений изменился по сравнению с «книжной» эпохой. Ранее в русском языке были распространены сокращения путем отсечения

концевой части слова («стр. 5», «техн. обр.», «тов. Иванов») или удаления середины («гражд-во», «отв-сть»). Иногда применялись сокращения через косую черту («с/х», «м/ф», «в/о»). Обратите внимание: во всех этих случаях пропущенная часть слова заменяется специальным символом (точкой, дефисом или косой чертой).

Современные пользователи Интернета стараются обходиться без всяких заменяющих символов. Сейчас характерны такие сокращения, в которых часть слова только выкидывается, но ничего взамен не добавляется: «комп», «проц», «флоп», «Инет».

Это присуще не только носителям русского языка. Например, в английском языке (особенно в американском варианте) марку компьютеров «Macintosh» сокращают до «Mac», а вместо «you» иногда пишут просто «u».

Конечно, это все касается, если можно так выразиться, «разговорного» варианта письменной речи — в чате, по электронной почте, на форуме и т. д. В «серьезной» письменности — например, в печатном издании или на сайте — эти сокращения выглядят нелепо.

Из английского языка к нам пришла мода на числовые сокращения. Например, американцы в электронных сообщениях часто пишут «2» вместо «to» и «4» вместо «for»: *up 2 you*, *mail 4 my friend*. Бывает, что цифры оказываются и в середине слова: *car2n*, *4est*. В русском языке нет таких служебных слов, которые близки по звучанию к числительным, поэтому цифры иногда заменяют часть слова: *по2л*, *о5*.

Это нельзя считать удачной практикой. Такие сокращения, возможно, короче набирать, но труднее читать; кроме того, они портят вид текста, «вылезая» из строки.

纸 *Дополнительно*

Интересный эффект возникает от совпадения некоторых доменных суффиксов с распространенными английскими словами. Скажем, в зоне .to (Тонга) зарегистрирован домен welcome.to, предоставляющий услуги по размещению сайтов, адресом которых будет welcome.to/name, где name — имя компании или организации. Многие телевизионные компании приобрели себе домены в зоне .tv (Тувалу): **www.bbc.tv**, **www.mtv.tv**.

Создатели русскоязычных ресурсов не отстают от зарубежных коллег. Много отечественных сайтов зарегистрировано в зоне .net: **pisem.net**, **sporu.net**, **nichego.net**. Иногда доменный суффикс служит частью слова: **poda.ru**, **kengu.ru**, **www.design.ru/kovodstvo**.

Типографские причуды

Иногда текст «приукрашивают» совершенно чуждыми его природе средствами. Вот несколько примеров.

- Пробелы между буквами в слове: *ф о р у м*, *d e s i g n e r*.
- Разделение букв каким-нибудь другим знаком: *С.А.Й.Т*, *Т:Е:А:М*.
- Замена некоторых букв на цифры или другие похожие по написанию знаки: *о3он*, *usabi1ity*.
- Замена русских букв на латинские буквы похожего написания: *ОВЕС РАСТЕТ* (то же самое строчными буквами: *obec pactet*).

Такие «украшения» затрудняют чтение, но проблема с ними не только в этом. Гораздо серьезнее, что они не позволяют поисковым системам корректно проиндексировать сайт, в результате чего поиск такого сайта и поиск по такому сайту существенно осложняются. В самом деле: поисковая система будет искать слово «форум», но не найдет, потому что такого слова на сайте нет, а есть (с точки зрения поисковой машины) пять однобуквенных слов, разделенных пробелами: «ф о р у м».

Внимание!

Если вы заинтересованы в продвижении своего сайта и его достойной позиции в рейтингах и каталогах поисковых систем, — не употребляйте типографских «украшений», искажающих слова.

Встречаются и более безобидные «украшения» — например, выделение каждой второй буквы в вЕрХнИй рЕгИсТр, — но читать эти изыски крайне неудобно и неприятно.

Дополнительно

В Интернете можно обнаружить более сложные, комбинированные типографские причуды. Вот, например, несколько личных псевдонимов, которые автор встречал в Сети:

```
<==DungeoN==#==KeepeR==>
VЯED1NA
-(**[c0p]**)-
m1a2i2l3e5r
ТелеPUZZIK!
```

Примечание

Кстати, на жаргоне такие типографские причуды называют *кульзипопством*.

Эмотиконы

> Признак того, что вы проводите слишком много времени за компьютером: чтобы улыбнуться, вы поворачиваете голову на 90 градусов.
>
> *Анекдот*

Наличие специального научного обозначения — «эмотиконы» — свидетельствует о том, что явление это достаточно важное. Речь идет о знакомых всем смайликах, или «рожицах»: именно они с недавних пор официально именуются «эмотиконами».

Внимание!

Эмотикон — это текстовый знак, составленный из символов, введенных с клавиатуры, и отражающий настроение автора в момент написания. Обычно эмотикон представляет собой подобие рожицы, повернутой на правый бок.

Можно говорить о сложившемся синтаксисе эмотиконов. В этих знаках выделяется пять составных элементов: волосы, глаза, нос, рот, нижняя часть (табл. 2.3).

Таблица 2.3. *Синтаксис эмотиконов*

Элемент	Употребление	Примеры	
Волосы	Употребляется редко	Ж	растрепанные волосы
Глаза	Употребляется всегда	:	обычная форма
		;	подмигивание
		8	выпученные глаза
		%	перекошенные глаза
Нос	Иногда опускается	–	основная и практически единственная форма
Рот	Употребляется всегда, может редуплицироваться	)	улыбка
		(	грусть, неудовольствие
		|	поджатые губы
		O	разинутый рот
Нижняя часть	Употребляется редко	~	слюна
		Ж	борода

Большинство смайликов являются комбинацией перечисленных элементов и могут выражать самые разные эмоции:

`:-)`	Наиболее употребительный эмотикон. Чаще всего обозначает добрые намерения автора — например, в том случае, если читатель может неправильно понять написанную фразу и обидеться
`;-(`	Печаль с долей иронии
`%-)))))))))`	Безудержный восторг с перекошенными глазами

К эмотиконам относятся по-разному. Одни считают, что они засоряют письменную речь и служат свидетельством дурного тона и неумения выразить свои мысли обычными словами. Другие возражают, что смайлики очень практичны, т. к. позволяют передать интонацию речи гораздо проще и быстрее, чем ее пришлось бы выражать словами.

Примечание

Мнение автора о смайликах аналогично мнению о жаргоне. Эмотиконы рекомендуется использовать только в том случае, если вы уверены, что смогли бы передать нужную интонацию и без их помощи. Иначе вы рискуете оказаться в ситуации, когда по каким-либо причинам смайлики недоступны, а мысль выразить как-то надо.

Эмотиконы имеют свою область применения — это электронная переписка, общение в чатах, на форумах и в телеконференциях, а также другие тексты оперативно-разговорного характера. Но в тексте на сайтах, не говоря уж о печатных изданиях, ;-) эмотиконы совершенно неприемлемы.

Возможно, когда-то и наступит время, когда смайлики превратятся в обычный оборот письменной речи и станут широко применяться — даже в литературных текстах. Но сейчас не стоит щеголять эмотиконами вне их естественной среды обитания.

Дополнительно

Эмотиконы нашли поддержку у некоторых производителей текстовых процессоров. Например, Microsoft Word автоматически заменяет «обычные» смайлики на специальные символы:

:-)	☺
:-(	☹

Впрочем, такие символы не получили широкого распространения. Их трудно ввести с клавиатуры, и они не входят в кодировки, которыми обычно пользуются в сетевом общении.

Примечание

При подготовке этой статьи использованы опубликованные в Интернете материалы Ф. Рожанского.

Обсценная типографика

В школьные годы, начиная изучать новый иностранный язык, автор, как и многие его сверстники, первым делом интересовался, как на этом языке звучат неприличные слова. Это было особое удовольствие — громко произносить эти слова среди тех, кто не владеет этим языком, и знать, что за это ничего не будет.

Собственно говоря, слово «обсценный» в языкознании как раз и обозначает «неприличный», «ненормативный». В Интернете сложились особые традиции употребления обсценной лексики.

На многих форумах, в чатах, в телеконференциях правила запрещают употреблять ненормативную лексику. Но находчивые пользователи придумали способ обойти этот запрет. Они заменяют часть запретного слова или целое слово специальными символами — %, @, # и другими. Вот несколько примеров, взятых в основном из телеконференций сети Fidonet:

- Да за@%%л ты уже со своей @%%ной юзабилити!
- Это еще, так сказать, теоретический п####ц, а скоро у нас наступит п####ц сугубо практический.
- ИДИТЕ ВСЕ В %_@_!!!!! %-(((((

Дополнительно

Запись слов прописными буквами в сетевом общении традиционно означает разговор на повышенных тонах.

Примечание

Автор считает, что ненормативная лексика — это своего рода золотой запас русского языка. Поэтому расходовать ее нужно крайне экономно. Иначе от нее просто не будет никакого эффекта.

2.2.4. Смесь французского с нижегородским
О включении в текст на русском языке слов, записанных на других языках

> Он ездил к ma bonne amie Анне Павловне и ездил dans le salon diplomatique de ma fille и часто, при беспрестанных переездах из одного лагеря в другой, путался и говорил у Анны Павловны то, что надо было говорить у Элен, и наоборот.
>
> *Л. Н. Толстой. Война и мир*

Включение в русскоязычный текст слов, записанных латиницей, — это относительно недавнее явление. Если посмотреть на печатные издания, выпущенные в докомпьютерную эпоху, то можно убедиться, что латиница применяется только в художественных произведениях в особом стилистическом качестве — например, когда персонаж говорит на иностранном языке или когда требуется подчеркнуть, выделить особую «иноязычность» слова или оборота.

Иногда латиницу можно встретить в технической или научной литературе тех времен — но только во вспомогательном качестве, в скобках после термина или фамилии иноязычного происхождения: *волновая механика Шрёдингера (Schrödinger) появилась в 1926 году.*

Во всех остальных случаях иностранные имена и наименования было принято транскрибировать. В отечественных газетах и журналах писали: «Рональд Рейган», «Битлз», «Кока-Кола», «Нью-Йорк Таймс», «НАТО» (а не «Ronald Reagan», «Beatles», «Coca-Cola», «New York Times», «NATO»).

Нарицательные слова, заимствованные из иностранных языков, тоже никогда не сохраняли своего иноязычного написания. Даже когда в нашей стране появлялись новые технологии — например, телевидение или видео, — в печатных изданиях не писали: «tele-передача» или «video-запись». Слова «компьютер», «принтер», «ксерокс» и многие другие появились у нас сразу в кириллическом написании.

Все изменилось после того, как техническую и компьютерную литературу стали писать и издавать люди, хорошо разбирающиеся в своей предметной области, но не знакомые с устоявшимися правилами и традициями русского языка. В русскоязычных текстах стали появляться «компания Microsoft», «программа Photoshop» и «сеть Internet». Впоследствии эта привычка к латинице перекочевала и в тексты общего назначения — например, в лентах новостей и газетных статьях иногда встречается «телеканал CNN», «группа Metallica» и т. д.

Не думайте, что автор считает этот новый стиль кошмаром русского языка и призывает писать «Фотошоп» и «КваркИксПресс». Безусловно, автор по своим взглядам скорее консерватор, но даже консерватор не может отрицать того, что русский язык не стоит на месте, что он развивается и видоизменяется.

Многие соображения говорят в пользу того, что латинское написание — это временное явление, которое должно рано или поздно уйти из языка. В самом деле, так произошло со словом «Интернет». Когда оно употреблялось только узким кругом специалистов, его писали латинскими буквами в литературе и технической документации. Однако по мере того, как это понятие распространялось в общеупотребительной лексике, оно стало записываться русскими буквами и в таком виде появилось в орфографических словарях. Сейчас процесс «русификации» происходит с приставкой «веб-».

纸 *Дополнительно*

Существуют заимствованные слова, употребление и написание которых в русском языке вызывает серьезные затруднения: «online», «cookie», «usability» и некоторые другие. Слово «online» никак не может определиться, какой частью речи ему быть: оно одинаково неестественно смотрится и в качестве существительного, и в качестве наречия. «Cookie» плохо ведет себя при транскрипции, превращаясь во множественное число от неведомого слова — то ли «кук», то ли «кука». «Usability» колеблется между женским и средним родом (хотя в обоих выглядит вызывающе). Время покажет, освоятся ли эти слова и им подобные в русском языке.

Автор хотел бы предложить несколько рекомендаций, основанных на сложившейся в современных текстах традиции употреблять слова в иноязычном написании:

- при написании иностранных слов на языке оригинала строго соблюдайте орфографию источника;

Правильно	Неправильно
Škoda	Skoda
Löwenbräu	Lowenbrau
TEX	TEX, TeX

- не записывайте латинскими буквами личные имена, фамилии, географические наименования. В текстах справочного, технического и научного характера после первого упоминания фамилии или наименования можно в скобках поместить «родное» написание;
- не используйте сложных слов, часть которых записана кириллицей, а часть — латиницей. Либо пишите такие слова целиком по-русски, либо перестраивайте фразу по-другому;

Правильно	Неправильно
сервер FTP	FTP-сервер
веб-страница	web-страница, Web-страница

- наименования — например, названия фирм, торговых марок, программных продуктов — целесообразно записывать в кавычках, как и в русском языке. Особенно это актуально для наименований, состоящих из нескольких слов[4].

Рекомендуется	Не рекомендуется
компания «Microsoft»	компания Microsoft
текстовые процессоры «Microsoft Word» и «OpenOffice»	текстовые процессоры Microsoft Word и OpenOffice

Дополнительно

В последнем примере, если не использовать кавычек, читатель может подумать, что имеются в виду два текстовых процессора компании Microsoft, которые называются Word и OpenOffice.

Во всех перечисленных случаях употребляйте русские кавычки-«елочки», а не английские.

Аббревиатуры (HTTP, W3C, NASA и т. д.) записывайте без кавычек;

- старайтесь не использовать иноязычных наименований в косвенных падежах. Если это все-таки неизбежно, отделяйте окончание апострофом: *нарисовано в «Photoshop'е»*, *в совершенстве владеет TEX'ом*. Впрочем, всегда лучше найти вариант, не требующий изменения слова: *нарисовано в редакторе «Photoshop»*, *в совершенстве владеет системой TEX*.

Если какие-то из этих рекомендаций кажутся вам несостоятельными — вы можете пользоваться собственным стилем написания. Но в любом случае важно соблюдать единообразие. Например, не должно быть такого, чтобы в одном месте англоязычное название компании было написано в кавычках, а в другом — без них.

Русский язык живет и изменяется. Правила и нормы, зафиксированные в нормативной и справочной литературе, не управляют развитием языка, а, наоборот, закрепляют уже сложившиеся принципы и традиции. В языке находится место и специальной терминологии, и жаргону, и ненормативной лексике.

---

[4] Издательство не считает эту рекомендацию правильной — слова на латинице и так весьма выделяются из русского текста и нет смысла «замусоривать» его лишними кавычками. — *Ред.*

Наш язык отражает нашу культуру, наше духовное богатство. Посетитель сайта будет судить о его разработчиках по стилю и грамотности текста, по его типографскому и художественному оформлению. Если для вас важно отношение пользователей к вам и вашему сайту — обращайте внимание на то, чтобы ваш текст был качественным и грамотно оформленным.

От нас в немалой степени зависит, по какому пути пойдет русский язык в своем дальнейшем развитии, как будут говорить и писать наши потомки. И в наших силах сделать так, чтобы язык, обогащаясь какими угодно новшествами, продолжал оставаться великим и могучим, правдивым и свободным — таким языком, которым можно и нужно гордиться.

```
<?xml version="1.0"
<!DOCTYPE html PUBLI
  "-//W3C//DTD XHTM
  "http://www.w3.or
<html xmlns="http:/
  lang="ru" xml:lan
<!-- Редактировал:
<head>
  <title>Евгений Он
  <link rel="styles
    href="index.css
</head>
```

Часть II

Дело мастера боится

Техника разработки и кодирования текстового содержания сайтов

УПРЕМУДРЯТЬ, упремудрить кого, одарить, научить премудрости. -ся, страдат. и возвр. Умаляяйся деянием своим упремудрится, Сир.

В. И. Даль. Толковый словарь живого великорусского языка

Пора уже было переходить от теории к практике, и тут Темный, ругая на все корки скупость ректората, вынужден был сам играть роль оживляемого трупа, накидывая на себя личину мертвеца.

Н. Перумов. Рождение мага

```
<?xml version= 1.0
<!DOCTYPE html PUBL
  "-//W3C//DTD XHTM
  "http://www.w3.or
<html xmlns="http:/
  lang="ru" xml:lan
<!-- Редактировал:
<head>
  <title>Евгений Он
  <link rel="styles
    href="index.css
</head>
```

Глава 3

Двоичная сюита

Представление текста в компьютере: кодировки символов, форматы текстовых файлов

Если присутствие тока можно обнаружить на любом участке контура, то почему бы не передавать данные мгновенно с помощью электричества?

Сэмюэль Морзе (1832)

При подготовке всех знаков шрифта для набора они сортировались по кеглям для размещения в стандартных шрифтовых кассах, рамах или кабинетах. Стандартная касса латинского состоит из 125 секций. Каждая, в зависимости от кегля шрифта, весила от 12 до 25 кг.

Г. Флаке. Производство шрифтов в прошлом и настоящем

3.1. Азбука для компьютера

Кодировки символов

пОБ ОБДЕСМБУШ, ЮФП УХНЕЕФ РТП-ЮЙФБФШ Ч ЛОЙЗЕ ИПФШ ПДОХ УФ-ТБОЙЮЛХ, ОП ЧУЕ ВЩМП ОБРЙУБОП ОБ ЛБЛПН-ФП ОЕРПОСФОПН СЪЩЛЕ.

м. лЬТТПММ. бМЙУБ Ч ъБЪЕТЛБМШЕ

Ты смотришь вокруг и спокойно дышишь,
Но как невозвратны такие дни!

Поэтому все, что в душе запишешь,
И все, что увидишь ты и услышишь,
Запомни, запомни и сохрани!

Э. Асадов. Свидание с детством

Вся информация в компьютере хранится в двоичном коде. Двоичный код на самом деле представляет собой последовательность определенных электрических или магнитных состояний участков носителя (например, жесткого диска или оперативной памяти). Но нам с вами достаточно будет представлять себе двоичный код как привычную последовательность нулей и единиц.

Один элемент кода — 0 или 1 — занимает в памяти компьютера один бит.

Чтобы сохранить в компьютере текст, его надо преобразовать в двоичный код. А при выводе на экран или другое устройство просмотра компьютер должен преобразовать код обратно из нулей и единиц в символы, понятные человеку.

3.1.1. Кодировки и кодовые таблицы

What this specification calls a character encoding is known by different names in other specifications (which may cause some confusion). However, the concept is largely the same across the Internet. Also, protocol headers, attributes, and parameters referring to character encodings share the same name—“charset”—and use the same values from the IANA registry.

HTML 4.01 Specification[1]

Как вы уже знаете из *главы 1*, сохранить текст в двоичном коде легче, чем любую другую информацию. Для этого нужно всего лишь поставить каждому символу текста в соответствие какой-нибудь двоичный код. Обозначим латинскую букву «A» кодом 1000001, букву «B» — кодом 1000010 и т. д. При такой системе, например, название популярной группы «ABBA» будет записано в памяти компьютера так: 1000001 1000010 1000010 1000001.

---

[1] То, что в этой спецификации называется «character encoding» («кодировка символов»), в других спецификациях может встретиться под другими наименованиями (что иногда приводит к путанице). Впрочем, все эти наименования в применении к Интернету соответствуют одной и той же общей концепции. В заголовках протоколов, в атрибутах и параметрах, относящихся к кодировке символов, используется одно и то же ключевое слово — «charset» — и одни и те же значения из реестра IANA. *(Спецификация HTML 4.01)*

Конечно, мы не можем просто так взять и присвоить каждому символу любой код, какой нам захочется. Коды, соответствующие буквам, цифрам, знакам препинания и другим текстовым символам, уже «зашиты» в программное обеспечение наших компьютеров. При вводе и выводе программа или операционная система автоматически преобразует символы в двоичный код и обратно.

Внимание!

Система преобразования символов в двоичный код называется *кодировкой*, а таблица соответствия между символами и кодами — *кодовой таблицей*.

Дополнительно

Термины «кодировка» и «кодовая таблица» в некоторых источниках оказываются синонимами. В английском языке еще сложнее: целых четыре слова «encoding», «coding», «charset» и «codepage» в одних текстах обозначают одно и то же, а в других — нет. Автор надеется, что по крайней мере эта книга не вызовет лишней путаницы.

Чтобы человек мог прочесть сохраненный в компьютере текст, необходимо, чтобы для его декодирования (преобразования из двоичных кодов в символы алфавита) использовалась та же кодовая таблица, что и для его изначального кодирования. В противном случае могут возникнуть проблемы, с которыми иногда сталкиваются посетители сайтов (рис. 3.1).

Для хранения текстовой информации в современных компьютерных системах используется множество различных кодировок. Только для текстов на русском языке существует по меньшей мере три распространенные кодовые таблицы и еще полдюжины менее употребительных. Чтобы не попасть впросак (как на рис. 3.2), разработчику сайтов рекомендуется хорошо ориентироваться в различных кодировках и знать о проблемах, возникающих из-за них.

Важная характеристика всякой кодировки — ее *разрядность*, то есть количество битов, предназначенных для кодирования одного символа. Например, разрядность кодировки, в которой мы записали буквы «А» и «В», равна 7. Во многих современных кодировках разрядность равна 8, но применяются и кодировки, в которых разные символы кодируются различным количеством битов. Например, в кодировке UTF-8 на каждую букву латинского алфавита отводится по 8 битов (1 байт), а на каждую букву кириллицы — по 16 битов (2 байта).

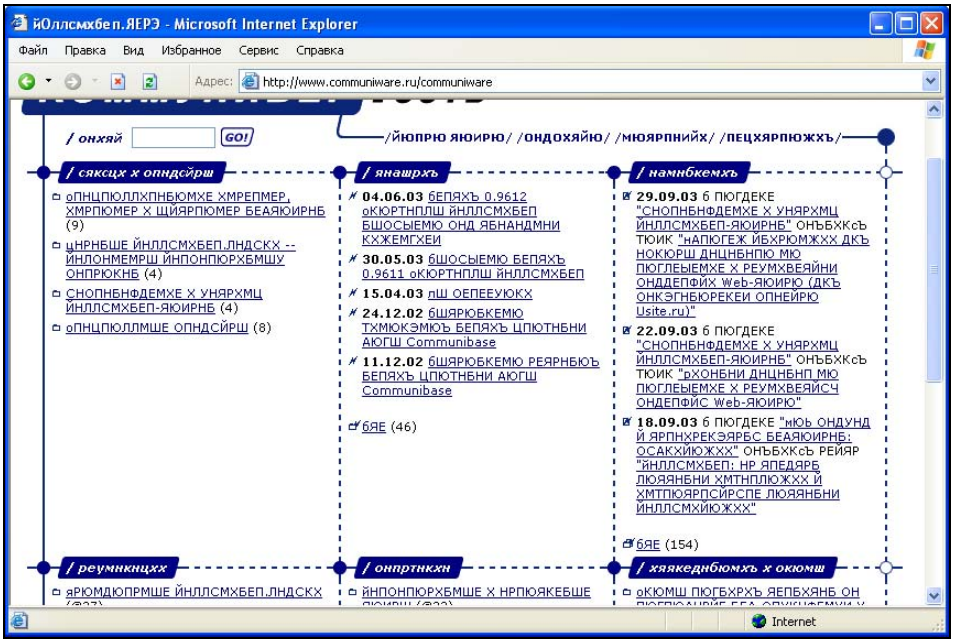


Рис. 3.1. Текст на этом сайте был набран в кодировке Windows-1251, а просматривается в кодировке KOI8-R. Получается так называемая «бНОПНЯ» (именно так выглядит слово «Вопрос», если оно закодировано в Windows-1251, а декодировано в KOI8-R)

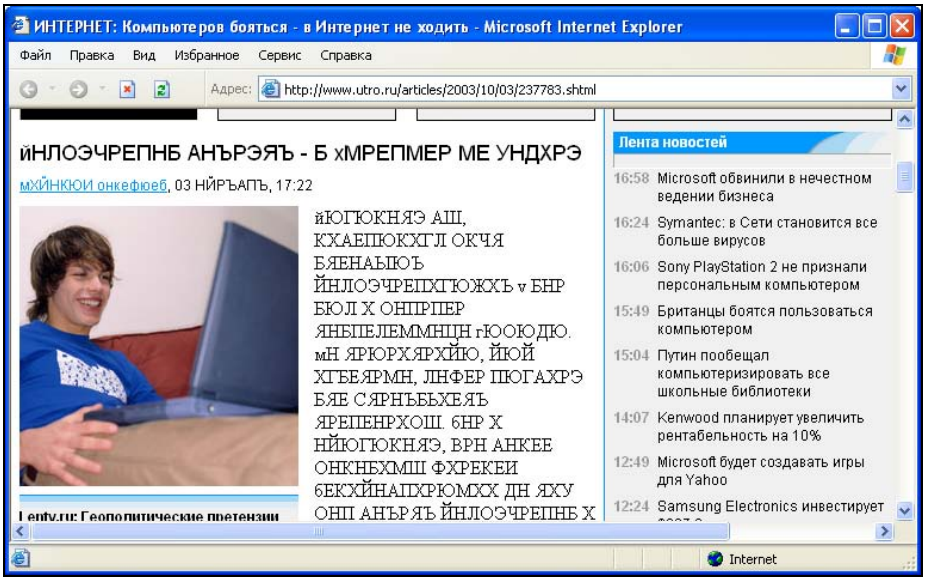


Рис. 3.2. На этой динамически собранной странице часть текста оказалась в одной кодировке, а часть — в другой

Дополнительно

От разрядности кодировки зависит, сколько различных символов в ней можно представить. Если *n* — разрядность кодировки, то количество символов равно 2^n.

Например, в четырехразрядной кодировке можно закодировать только $2^4 = 16$ символов. В эту кодовую таблицу не уместятся даже буквы латинского алфавита! А в 16-разрядной кодировке можно закодировать 65 536 символов: этого хватит и для латинского, и для русского, и для арабского, и для множества других алфавитов.

3.1.2. Двоичная и шестнадцатеричная системы счисления

> Становясь старше, вы обнаруживаете, что шестнадцатеричная система счисления полезна не только применительно к компьютерам. Когда мне исполнилось 40, я сказал друзьям, что мне всего лишь 28_{16}. Причем добавление «$_{16}$» я произнес, конечно, шепотом. В 50 лет мне будет только 32_{16}.
>
> *Дж. Уэйкерли. Проектирование цифровых устройств*

Прежде чем продолжать разговор о кодировках, нам потребуется вспомнить совсем немного школьного материала о системах счисления. Дело в том, что коды символов в технической документации и литературе обычно не обозначают двоичными числами. Вместо этого их переводят в десятичную или шестнадцатеричную систему (табл. 3.1).

Таблица 3.1. *Десятичная, двоичная и шестнадцатеричная системы счисления*

Десятичная	Двоичная	Шестнадцатеричная
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6

Таблица 3.1 *(окончание)*

Десятичная	Двоичная	Шестнадцатеричная
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

Перевод из двоичной системы в десятичную. Надо умножить последнюю цифру числа на 1 (2^0), предпоследнюю — на 2 (2^1), следующую — на 4 (2^2) и т. д., затем сложить полученные результаты.

Например: $101011_2 = 43$.

	1	0	1	0	1	1
×	32	16	8	4	2	1
+	32		8		2	1
=	43					

Дополнительно

В математике систему счисления принято обозначать нижним индексом после числа. Например, 101011_2 — это число, записанное в двоичной системе счисления. После чисел, записанных в десятичной системе, такого обозначения не пишут (43_{10} — неграмотно).

Впрочем, в разговорах о кодировках символов обычно из контекста бывает понятно, о какой системе счисления идет речь, поэтому индекс часто будет отсутствовать.

Перевод из десятичной системы в двоичную. Надо разделить исходное число на 2 с остатком. Неполное частное снова разделить на 2 с остатком и т. д., пока в частном не получится 0. После этого все остатки выписать, начиная с *последнего*.

Например, $43 = 101011_2$.

43	2					
1	21	2				
	1	10	2			
		0	5	2		
			1	2	2	
				0	1	2
					1	0

В шестнадцатеричной системе, кроме десяти обычных цифр, используются еще шесть:

A	B	C	D	E	F
10	11	12	13	14	15

Перевод из шестнадцатеричной системы в десятичную. Надо умножить последнюю цифру числа на 1 (16^0), предпоследнюю — на 16 (16^1), следующую — на 256 (16^2) и т. д., затем сложить полученные результаты.

Например, $12C_{16} = 300$.

	1	2	C
×	256	16	1
+	256	32	12
=	300		

Перевод из десятичной системы в шестнадцатеричную. Надо разделить исходное число на 16 с остатком. Неполное частное снова разделить на 16 с остатком и т. д., пока в частном не получится 0. После этого все остатки выписать, начиная с последнего.

Например, $300 = 12C_{16}$.

300	16		
12 = **C**	18	16	
	2	1	16
		1	0

Дополнительно

Шестнадцатеричная система очень удобна для записи кодов символов, т. к. каждая цифра такого кода соответствует ровно 4 битам. Следовательно, взглянув на код, можно сказать, сколько разрядов он занимает. Например, шестнадцатеричный код 41 (прописная буква «А» в кодировке ASCII) занимает 8 битов, то есть 1 байт. А код E2 99 AD (музыкальный знак «бемоль» в кодировке UTF-8) занимает 3 байта.

Перевод из двоичной системы в шестнадцатеричную. Надо убедиться, что количество цифр в исходном числе кратно 4. Если нет, дописать слева необходимое количество нулей. Затем разбить число на группы по 4 цифры и заменить каждую группу на соответствующую ей цифру из табл. 3.1.

Например, $10100001011_2 = 50B_{16}$.

1. В числе 10100001011 — 11 цифр, значит, дописываем слева нуль, чтобы цифр стало 12: 010100001011.
2. Разбиваем число на группы по 4 цифры: 0101 0000 1011.
3. Заменяем группы в соответствии с таблицей: 0101 = 5, 0000 = 0, 1011 = B. Таким образом, окончательный ответ — **50B**.

Перевод из шестнадцатеричной системы в двоичную. Эта процедура полностью обратна предыдущей. Нужно взять каждую цифру исходного числа и заменить ее на группу из четырех двоичных цифр в соответствии с табл. 3.1.

Дополнительно

В современной вычислительной технике принято записывать коды таким образом, чтобы они занимали целое количество байтов, причем именно такое, которое отведено в компьютере под данный код. Если код занимает отведенное место не полностью, оставшиеся разряды заполняются нулями.

Например, в кодировке ASCII код каждого символа занимает 1 байт. Код символа «&» — 38, или в двоичной системе — 100110. Он занимает всего 6 разрядов, поэтому два оставшихся заполняются нулями, и правильная запись кода выглядит так: 00100110.

В базовом наборе символов Unicode для каждого символа отведено 2 байта. Шестнадцатеричный код латинской буквы «А», как и в ASCII, — 41. Но в Unicode этот код запишется так: 0041.

Примечание

Разработчику сайтов приходится сталкиваться с шестнадцатеричным кодом, когда возникает необходимость вставить в текст символы, которые нельзя ввести непосредственно с клавиатуры — например, угловые кавычки, длинное

и короткое тире, многоточие, знаки «диез» и «бемоль» или китайские иероглифы. В стандарте Unicode и во многих программных средствах (например, в стандартной Таблице символов в Windows) указаны именно шестнадцатеричные коды, соответствующие таким знакам (рис. 3.3).

Рис. 3.3. В стандарте Unicode рядом с каждым символом указан его шестнадцатеричный код

3.1.3. ASCII

> У народов, сохранивших пережитки отдаленных неолитических времен, почти повсеместно находим культ всеобщей Матери... Она мыслится Хозяйкой Вселенной и одновременно Матерью зверей и людей.
>
> *А. Мень. История религии*

Иногда ASCII называют «бабушкой всех кодировок». Это не совсем верно, хотя разработчику сайтов вряд ли придется сталкиваться с кодировками, не происходящими от ASCII. Эта аббревиатура расшифровывается как American Standard Code for Information Interchange (Американский стандартный код для обмена информацией).

ASCII существует с 1967 года. В те времена 8 битов еще не были стандартной минимальной порцией представления информации, и каждый символ в ASCII занимал 7 битов. Следовательно, всего можно было закодировать $2^7 = 128$ различных символов. Двоичные коды этих символов лежали в диапазоне от 0000000 до 1111111 (от 0 до 127 в десятичной системе), и символы эти были такими (табл. 3.2).

Таблица 3.2. *Стандартная кодировка ASCII*

NUL	***SOH***	***STX***	***ETX***	***EOT***	***ENQ***	***ACK***	***BEL***
d 0 h 00	d 1 h 01	d 2 h 02	d 3 h 03	d 4 h 04	d 5 h 05	d 6 h 06	d 7 h 07
b 0000000	b 0000001	b 0000010	b 0000011	b 0000100	b 0000101	b 0000110	b 0000111
BS	***HT***	***LF***	***VT***	***FF***	***CR***	***SO***	***SI***
d 8 h 08	d 9 h 09	d 10 h 0A	d 11 h 0B	d 12 h 0C	d 13 h 0D	d 14 h 0E	d 15 h 0F
b 0001000	b 0001001	b 0001010	b 0001011	b 0001100	b 0001101	b 0001110	b 0001111
DLE	***DC1***	***DC2***	***DC3***	***DC4***	***NAK***	***SYN***	***ETB***
d 16 h 10	d 17 h 11	d 18 h 12	d 19 h 13	d 20 h 14	d 21 h 15	d 22 h 16	d 23 h 17
b 0010000	b 0010001	b 0010010	b 0010011	b 0010100	b 0010101	b 0010110	b 0010111
CAN	***EM***	***SUB***	***ESC***	***FS***	***GS***	***RS***	***US***
d 24 h 18	d 25 h 19	d 26 h 1A	d 27 h 1B	d 28 h 1C	d 29 h 1D	d 30 h 1E	d 31 h 1F
b 0011000	b 0011001	b 0011010	b 0011011	b 0011100	b 0011101	b 0011110	b 0011111
пробел	**!**	**"**	**#**	**$**	**%**	**&**	**'**
d 32 h 20	d 33 h 21	d 34 h 22	d 35 h 23	d 36 h 24	d 37 h 25	d 38 h 26	d 39 h 27
b 0100000	b 0100001	b 0100010	b 0100011	b 0100100	b 0100101	b 0100110	b 0100111
(	**)**	*****	**+**	**,**	**-**	**.**	**/**
d 40 h 28	d 41 h 29	d 42 h 2A	d 43 h 2B	d 44 h 2C	d 45 h 2D	d 46 h 2E	d 47 h 2F
b 0101000	b 0101001	b 0101010	b 0101011	b 0101100	b 0101101	b 0101110	b 0101111
0	**1**	**2**	**3**	**4**	**5**	**6**	**7**
d 48 h 30	d 49 h 31	d 50 h 32	d 51 h 33	d 52 h 34	d 53 h 35	d 54 h 36	d 55 h 37
b 0110000	b 0110001	b 0110010	b 0110011	b 0110100	b 0110101	b 0110110	b 0110111
8	**9**	**:**	**;**	**<**	**=**	**>**	**?**
d 56 h 38	d 57 h 39	d 58 h 3A	d 59 h 3B	d 60 h 3C	d 61 h 3D	d 62 h 3E	d 63 h 3F
b 0111000	b 0111001	b 0111010	b 0111011	b 0111100	b 0111101	b 0111110	b 0111111

Таблица 3.2 *(окончание)*

@	A	B	C	D	E	F	G
d 64 h 40	d 65 h 41	d 66 h 42	d 67 h 43	d 68 h 44	d 69 h 45	d 70 h 46	d 71 h 47
b 1000000	b 1000001	b 1000010	b 1000011	b 1000100	b 1000101	b 1000110	b 1000111
H	**I**	**J**	**K**	**L**	**M**	**N**	**O**
d 72 h 48	d 73 h 49	d 74 h 4A	d 75 h 4B	d 76 h 4C	d 77 h 4D	d 78 h 4E	d 79 h 4F
b 1001000	b 1001001	b 1001010	b 1001011	b 1001100	b 1001101	b 1001110	b 1001111
P	**Q**	**R**	**S**	**T**	**U**	**V**	**W**
d 80 h 50	d 81 h 51	d 82 h 52	d 83 h 53	d 84 h 54	d 85 h 55	d 86 h 56	d 87 h 57
b 1010000	b 1010001	b 1010010	b 1010011	b 1010100	b 1010101	b 1010110	b 1010111
X	**Y**	**Z**	**[**	****	**]**	**^**	**_**
d 88 h 58	d 89 h 59	d 90 h 5A	d 91 h 5B	d 92 h 5C	d 93 h 5D	d 94 h 5E	d 95 h 5F
b 1011000	b 1011001	b 1011010	b 1011011	b 1011100	b 1011101	b 1011110	b 1011111
`	**a**	**b**	**c**	**d**	**e**	**f**	**g**
d 96 h 60	d 97 h 61	d 98 h 62	d 99 h 63	d 100 h 64	d 101 h 65	d 102 h 66	d 103 h 67
b 1100000	b 1100001	b 1100010	b 1100011	b 1100100	b 1100101	b 1100110	b 1100111
h	**i**	**j**	**k**	**l**	**m**	**n**	**o**
d 104 h 68	d 105 h 69	d 106 h 6A	d 107 h 6B	d 108 h 6C	d 109 h 6D	d 110 h 6E	d 111 h 6F
b 1101000	b 1101001	b 1101010	b 1101011	b 1101100	b 1101101	b 1101110	b 1101111
p	**q**	**r**	**s**	**t**	**u**	**v**	**w**
d 112 h 70	d 113 h 71	d 114 h 72	d 115 h 73	d 116 h 74	d 117 h 75	d 118 h 76	d 119 h 77
b 1110000	b 1110001	b 1110010	b 1110011	b 1110100	b 1110101	b 1110110	b 1110111
x	**y**	**z**	**{**	**|**	**}**	**~**	***DEL***
d 120 h 78	d 121 h 79	d 122 h 7A	d 123 h 7B	d 124 h 7C	d 125 h 7D	d 126 h 7E	d 127 h 7F
b 1111000	b 1111001	b 1111010	b 1111011	b 1111100	b 1111101	b 1111110	b 1111111

коды символов:
d (decimal) — десятичный, h (hexadecimal) — шестнадцатеричный, b (binary) — двоичный

Внимание!

Все символы ASCII делятся на две группы:

- отображаемые (десятичные коды от 32 до 126);
- управляющие (от 0 до 31, а также 127) — условные обозначения этих символов набраны более мелким шрифтом.

Начнем с *отображаемых символов*. Эти символы благополучно используются в компьютерной технике до сих пор, их можно найти на любой стандартной клавиатуре.

В ASCII — 95 отображаемых символов:

- 52 буквы латинского алфавита (26 прописных и 26 строчных);
- 10 цифр;
- 33 специальных символа (знаки препинания, обозначения вроде «&» и «@» и др.).

Сразу видно, что нам эта кодировка не подходит. Во-первых, в ней нет русских букв, так что тексты на русском языке в ASCII набирать не получится. Во-вторых, в ней отсутствуют специальные типографские символы, о которых шла речь в *главе 2*, — нет ни тире, ни многоточия, ни кавычек, ни знака авторского права.

Тем не менее, создание ASCII было огромным шагом вперед по сравнению с существовавшими ранее принципами представления текстов в компьютере. ASCII позволяла записывать компьютерные программы и простые тексты на английском языке, как на обычной печатной машинке. Кодировка ASCII стала международным стандартом, благодаря чему тексты, набранные на одном компьютере, могли быть успешно прочитаны на другом.

Поговорим теперь об *управляющих символах*. Чем они отличаются от отображаемых? У каждого отображаемого символа есть определенный внешний вид, позволяющий нам распознать этот символ на экране или бумаге.

Дополнительно

Обратите внимание: пробел (десятичный код 32) — это отображаемый символ. В самом деле, ведь мы *видим* пробел между словами, а значит, у него есть внешний вид!

Управляющие символы (табл. 3.3) — другое дело. Изначально они были предназначены не для отображения на экране, а для управления процессом вывода на дисплей, телетайп или другое устройство. Например, символ с десятичным кодом 12 (FF, расшифровывается как Form Feed) обозначал извлечение страницы. Встретив в тексте такой символ, печатающее устройство не выводило его, а выталкивало текущую страницу и продолжало печатать на следующей.

Примечание

Управляющие символы — это инструкции для устройства вывода. Встретив в тексте управляющий символ, устройство не выводит этот символ, а выполняет связанное с ним действие.

Или, например, символ с десятичным кодом 7 (BEL, Bell) означал звуковой сигнал. С ним вы даже могли встретиться, если застали операционную систему DOS. При попытке просмотреть файл, содержащий символ BEL, компьютер под управлением DOS иногда начинал пищать.

Большинство из управляющих символов ASCII уже давно не используются и сохраняются в современных кодировках, очевидно, только из уважения к традиции.

Дополнительно

Тем не менее, эти символы могут вас заинтересовать, если вы увлекаетесь компьютерной археологией.

Таблица 3.3. *Управляющие символы ASCII*

Десятичный код	Шестнадцатеричный код	Сокращение	Название	Назначение
0	00	NUL	Null	Пусто
1	01	SOH	Start of Heading	Начало заголовка
2	02	STX	Start of Text	Начало текста
3	03	ETX	End of Text	Конец текста
4	04	EOT	End of Transmission	Конец передачи
5	05	ENQ	Enquiry	Запрос
6	06	ACK	Acknowledge	Подтверждение
7	07	BEL	Bell	Сигнал
8	08	BS	Backspace	Возврат на символ назад
9	09	HT	Horizontal Tabulation	Горизонтальная табуляция
10	0A	LF	Line Feed	Перевод строки
11	0B	VT	Vertical Tabulation	Вертикальная табуляция
12	0C	FF	Form Feed	Извлечение страницы
13	0D	CR	Carriage Return	Возврат каретки
14	0E	SO	Shift Out	Выход

Таблица 3.3 *(окончание)*

Десятичный код	Шестнадцатеричный код	Сокращение	Название	Назначение
15	0F	SI	Shift In	Вход
16	10	DLE	Data Link Escape	Авторегистр
17	11	DC1	Device Control 1	Управление устройством 1
18	12	DC2	Device Control 2	Управление устройством 2
19	13	DC3	Device Control 3	Управление устройством 3
20	14	DC4	Device Control 4	Управление устройством 4
21	15	NAK	Negative Acknowledge	Нет подтверждения
22	16	SYN	Synchronous Idle	Синхронизация
23	17	ETB	End of Transmission Block	Конец блока передачи
24	18	CAN	Cancel	Отмена
25	19	EM	End of Medium	Конец носителя
26	1A	SUB	Substitute	Замена
27	1B	ESC	Escape	Сброс
28	1C	FS	File Separator	Разделитель файлов
29	1D	GS	Group Separator	Разделитель групп
30	1E	RS	Record Separator	Разделитель записей
31	1F	US	Unit Separator	Разделитель элементов
127	7F	DEL	Delete	Удаление

Однако среди управляющих символов есть такие, разбираться в которых необходимо каждому разработчику сайтов. Это символы перевода строки (LF, код 10), возврата каретки (CR, код 13) и табуляции (HT, код 9).

Перевод строки

Если вы набираете текст в текстовом редакторе (например, Блокноте или Vim) или в текстовом процессоре (например, Word или OpenOffice), то для перехода на новую строку вы нажимаете клавишу <Enter> (или <Return>). При этом в текст вставляется специальный код, обозначающий перевод строки. Этот код относится к управляющим, т. к. вы не видите в тексте никакого соответствующего ему символа.

Код перевода строки уходит своими корнями в эпоху пишущих машинок. Для того чтобы перейти на новую строку, машинистка должна была выполнить две операции: во-первых, специальным рычагом прокрутить лист на одну строку вперед (вертикальное перемещение), а во-вторых, передвинуть каретку так, чтобы печатающая головка оказалась в начале строки (горизонтальное перемещение). Это и есть операции перевода строки (Line Feed) и возврата каретки (Carriage Return).

Примечание

Если у вас есть старая печатная машинка — попробуйте изучить технологию работы с ней. Это поможет вам понять происхождение многих принципов работы с текстом в современной компьютерной технике.

Старинные компьютеры и другие цифровые устройства для работы с текстом выполняли обе эти операции автоматически, встретив в тексте соответствующие управляющие символы. Встречается LF — устройство прокручивает страницу, встречается CR — устройство переводит позицию вывода в начало строки.

В компьютерах под управлением Windows (и DOS) этот принцип сохранился. Каждый раз, когда вы нажимаете <Enter>, ваш редактор вставляет в текст два символа: сначала CR, затем LF.

Например, если в окне текстового редактора набрано:

```
VENI,
VIDI,
VICI
```

то в памяти компьютера это будет представлено последовательностью двоичных кодов (для краткости и удобства они переведены в шестнадцатеричную систему):

56	45	4E	49	2C	0D	0A	56	49	44	49	2C	0D	0A	56	49	43	49
V	E	N	I	,	CR	LF	V	I	D	I	,	CR	LF	V	I	C	I

Вы можете задаться вопросом: если символы CR и LF встречаются только вместе, почему бы не использовать вместо них обоих какой-нибудь один управляющий символ? Собственно говоря, так и сделано в операционных

системах класса UNIX — там перевод строки обозначается одним символом LF. И в старых операционных системах Mac OS он также обозначается одним символом — но CR. В более современных Mac OS X для перевода строки, как и в UNIX, используется LF (табл. 3.4).

Таблица 3.4. *Символы перевода строки*

Операционная система	Перевод строки
DOS, Windows	CR LF
UNIX, Mac OS X	LF
Mac OS	CR

纸 *Дополнительно*

Эти различия иногда приводят к проблемам. Например, если текстовый файл набран в операционной системе класса UNIX, то при его открытии в Windows все переводы строк могут пропасть, и весь текст запишется в одну строку. Наоборот, если набрать файл в Windows, а потом открыть его в UNIX, то в тексте могут появиться «лишние» символы — оставшиеся неприкаянными CR.

Впрочем, такие проблемы постепенно отходят в прошлое. Большинство современных текстовых редакторов поддерживают переводы строк и в стиле Windows, и в стиле UNIX. Они автоматически распознают переводы строк и позволяют преобразовывать текст из формата Windows в формат UNIX и обратно (рис. 3.4).

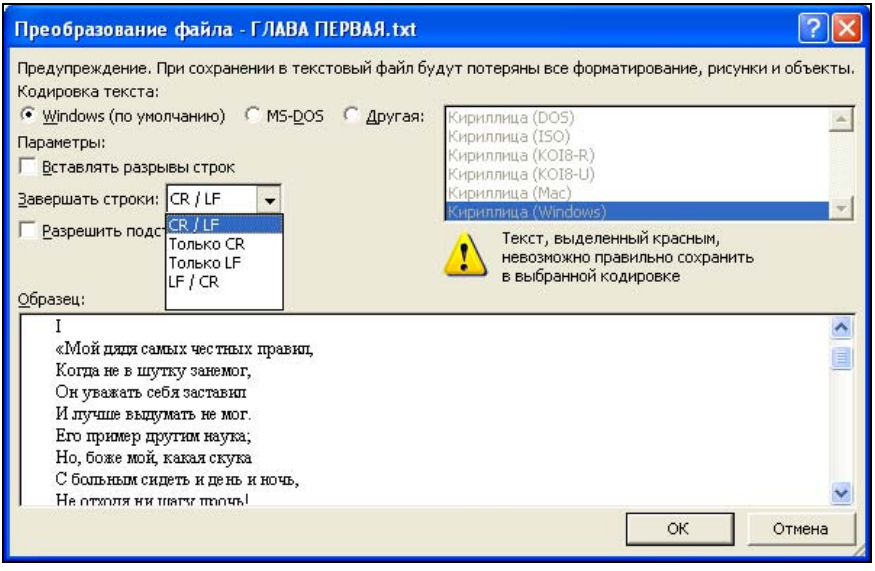


Рис. 3.4. Текстовый процессор Microsoft Word предлагает четыре разных режима перевода строки при сохранении файла

Примечание

Почему разработчику сайтов, особенно кодировщику HTML, важно разбираться в режимах перевода строки? Дело в том, что на практике часто возникают ситуации, когда при просмотре готовой страницы на ней возникают «лишние» переводы строк, которые не были задуманы. Еще чаще бывает наоборот: переводы строк исчезают, «склеиваются», превращаются в пробелы или испытывают еще какие-нибудь неприятные метаморфозы. Иногда причина этих проблем — именно в том, что в коде оказались неподходящие управляющие символы перевода строки.

Табуляция

Управляющий символ табуляции постепенно уходит в историю, и его назначение скоро будет казаться столь же туманным, как у символов ACK или DLE. Тем не менее, во времена простых текстовых форматов этот символ был весьма употребителен. Его использовали тогда, когда требовалось разместить рядом несколько столбцов, выровненных по горизонтали, — например, в текстовой таблице (рис. 3.5).

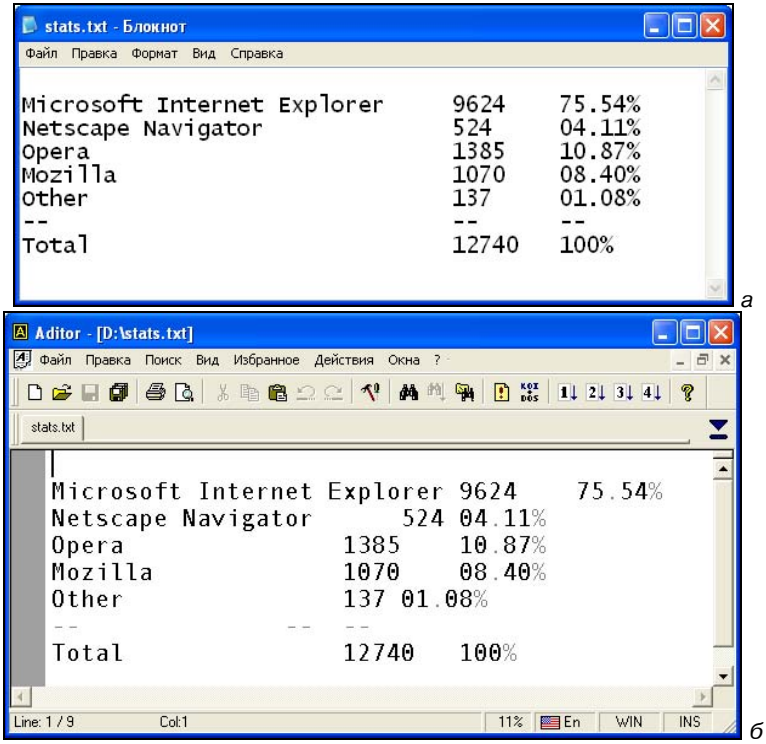


Рис. 3.5. Табуляция позволяет расположить данные ровными столбцами (*а*); но в другом текстовом редакторе ее позиции сместились, и столбцы испортились (*б*)

pushkin.html - Блокнот

Файл Правка Формат Вид Справка

```
<?xml version="1.0" encoding="windows-1251"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"

<html lang="ru" xml:lang="ru" xmlns="http://www.w3.org/19

<head>
  <title>В. Маяковский. Юбилейное</title>
</head>

<body>
<h1>ЮБИЛЕЙНОЕ</h1>
<div>
     Александр Сергеевич,
                        разрешите представиться.
                                                Маяковский.
     Дайте руку!
                  Вот грудная клетка.
                                Слушайте,
                                     уже не стук, а стон;
     тревожусь я о нем,
                        в щенка смирённом львенке.
     Я никогда не знал,
                         что столько
                                       тысяч тонн
     в моей
            позорно легкомыслой головенке.
     Я тащу вас.
                  Удивляетесь, конечно?
     Стиснул?
              Больно?
                      Извините, дорогой.
     У меня,
```

а

В. Маяковский. Юбилейное - Microsoft Internet Explorer

Файл Правка Вид Избранное Сервис Справка

Адрес: D:\pushkin.html

ЮБИЛЕЙНОЕ

Александр Сергеевич, разрешите представиться. Маяковский. Дайте руку! Вот грудная клетка. Слушайте, уже не стук, а стон; тревожусь я о нем, в щенка смирённом львенке. Я никогда не знал, что столько тысяч тонн в моей позорно легкомыслой головенке. Я тащу вас. Удивляетесь, конечно? Стиснул? Больно? Извините, дорогой. У меня, да и у вас, в запасе вечность. Что нам потерять часок-другой?! Будто бы вода — давайте мчать, болтая, будто бы весна — свободно и раскованно! В небе вон луна такая молодая, что ее без спутников и выпускать рискованно. Я теперь свободен от любви и от плакатов, Шкурой ревности медведь лежит когтист. Можно убедиться, что земля поката, — сядь на собственные ягодицы и катись! Нет, не навяжусь в меланхолишке черной да и разговаривать не хочется ни с кем. Только жабры рифм топырит учащённо у таких, как мы, на поэтическом песке. Вред — мечта, и бесполезно грезить, надо весть служебную нуду. Но бывает — жизнь встает в другом разрезе и большое понимаешь через ерунду. Нами лирика в штыки неоднократно атакована, ищем речи точной и нагой. Но поэзия — пресволочнейшая штуковина: существует — и ни в зуб ногой. Например, вот это — говорится или блеется? Синемордое, в оранжевых усах, Новуходоносором библейцем — «Коопсах». Дайте нам стаканы! знаю способ старый в горе дуть винище, но смотрите — из выплывают Red и White Star'ы с ворохом разнообразных виз. Мне приятно с вами,— рад, что вы у столика. Муза это ловко за язык вас тянет. Как это у вас говаривала Ольга?.. Да не Ольга! из письма Онегина к

Готово My Computer

б

Рис. 3.6. Стихотворение, закодированное в HTML с отступами и разбиением на строки (*а*), в браузере превратилось в одну строку (*б*)

Текстовый редактор, поддерживающий табуляцию, как бы мысленно разбивал строку на отдельные группы — чаще всего по 8 символов. Если вставить в такой текст символ табуляции (обычно путем нажатия клавиши <Tab>) — то следующий символ будет выведен ровно в начале следующей группы.

В современных текстовых редакторах пользоваться табуляцией нецелесообразно. Во-первых, для создания красивых таблиц почти во всех текстовых процессорах и программах верстки существуют специальные средства. Во-вторых, очень часто при просмотре табулированного текста не в том редакторе, в котором он создавался, позиции табуляции нарушаются (см. рис. 3.5).

Внимание!

Большинство браузеров особым образом обрабатывают пробелы, символы перевода строки и табуляции, встретившиеся в коде HTML.

Встретив любое количество идущих подряд пробелов, символов перевода строки и табуляции (в любых сочетаниях), браузер вместо всего этого выводит на экран один пробел (рис. 3.6).

Существуют специальные средства HTML, которые позволяют сохранить эти символы на веб-странице. Об этих средствах мы поговорим в *главе 5*.

纸 ***Дополнительно***

Распространение кодировки ASCII даже породило своеобразное направление искусства — ASCII Art, что можно примерно перевести как «Рисование в ASCII». Это искусство заключается в том, чтобы в произвольном текстовом редакторе создать рисунок, используя только символы из стандартной кодовой таблицы ASCII. Вот несколько примеров таких рисунков (рис. 3.7).

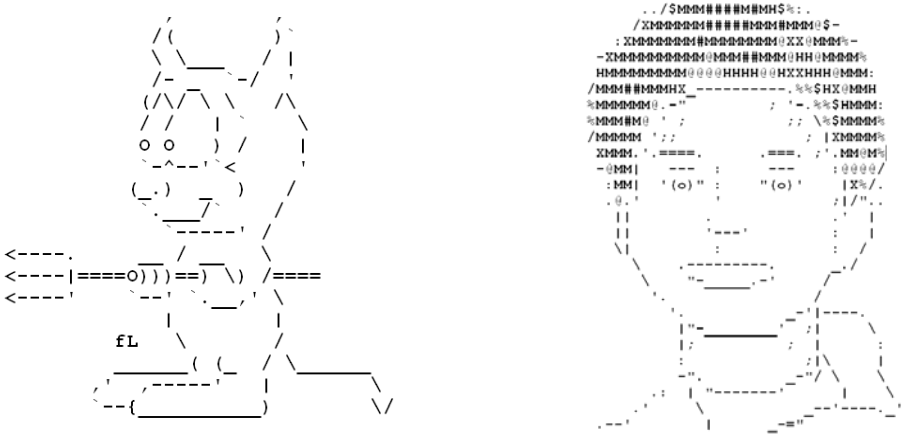


Рис. 3.7. ASCII Art *(начало)*

Рис. 3.7. ASCII Art *(окончание)*

3.1.4. Расширенные кодировки ASCII

> Сейчас важно поднять и расширить сознание могущих идти со Светом, но подымать его можно, лишь пользуясь доступными им методами и образами, лишь постепенно расширяя эти представления до мировых масштабов...
>
> *Е. И. Рерих. Письма*

В предыдущем разделе мы обнаружили, что писать на русском языке в стандартной кодировке ASCII невозможно. Она не подходит и для большинства других языков, включая арабский, китайский, иврит и суахили. Даже в языках, использующих латинский алфавит, обычно присутствуют символы, не входящие в ASCII: например, в немецком (ä, ö, ß), французском (è, é), польском (ł, ź).

纸 *Дополнительно*

Один из первобытных способов решения этой проблемы — исключить из кодовой таблицы некоторые редкие символы (например, {, |, }, ~) и поставить вместо них нужные буквы (например, ä, ö, ü, ß). Впрочем, такой способ ухудшает совместимость и никак не помогает для русского языка или суахили.

纸 Дополнительно

Для записи текстов на языках, отличных от английского, применялся и даже иногда применяется сейчас *транслит* (от слова «транслитерация») — замена национальных символов на знаки из стандартной ASCII или их сочетания.

Основное требование к системе транслита — *взаимная однозначность* кодирования. Необходимо, чтобы любой текст на национальном языке однозначно переводился в транслит и затем при необходимости восстанавливался в точности в первоначальном виде.

В русском языке для транслита есть стандарт — ГОСТ 7.79–2000, действующий с 2002 года. Согласно этому ГОСТу, русские буквы должны заменяться на латинские буквы (и некоторые другие символы ASCII) следующим образом (табл. 3.5).

Таблица 3.5. *Транслит: ГОСТ 7.79–2000*

а	a	к	k	х	kh
б	b	л	l	ц	c
в	v	м	m	ч	ch
г	g	н	n	ш	sh
д	d	о	o	щ	shh
е	e	п	p	ъ	"
ё	jo	р	r	ы	y
ж	zh	с	s	ь	'
з	z	т	t	э	eh
и	i	у	u	ю	ju
й	jj	ф	f	я	ja

Иногда в электронной переписке, а то и на сайтах применяются «самодельные» системы транслита, не обеспечивающие взаимной однозначности. Наиболее распространенные отклонения от стандарта: ё — yo, й — y, х — h, ц — ts, щ — sch, мягкий знак перед гласной — y, э — e, ю — yu, я — ya. Это приводит к неправильному прочтению слов, особенно сложных имен собственных. Например, однаково транслитерируются слова «сэр» и «сер», «Босх» и «Бош», а фамилия литературного персонажа «Чацкий» при обратном преобразовании превращается в «Чатски».

Чтобы расширить кодовую таблицу, было предложено задействовать восьмой бит. В 8-битной кодировке можно сохранить 256 символов, т. е. в два раза больше, чем в стандартной ASCII. Так появились *расширенные кодировки ASCII*.

Внимание!

Кодовая таблица любой расширенной кодировки ASCII состоит из двух частей. Первая часть (десятичные коды от 0 до 127) в точности совпадает со стандартной ASCII (см. табл. 3.2). Вторая часть (коды от 128 до 255) содержит специальные знаки, буквы национальных алфавитов или еще какие-нибудь символы, которые разработчик счел нужным включить.

Стандарт ISO 8859, существующий с середины 1980-х, определил несколько расширенных кодовых таблиц ASCII, каждая из которых покрывала один или несколько языков, отличных от английского. В большинстве этих таблиц коды 128–159 соответствуют дополнительным управляющим символам (так называемым «C1 Controls»), а остальные отвечают за отображаемые символы, не входящие в стандартную кодировку ASCII.

Кодировка ISO-8859-1 (Latin-1) содержит дополнительные символы для немецкого, французского, датского, финского и многих других языков, использующих латинский алфавит (табл. 3.6).

Таблица 3.6. *Кодовая таблица ISO-8859-1 (Latin-1)*

nbsp				¡				¢				£				¤				¥				¦				§			
d	160	h	A0	d	161	h	A1	d	162	h	A2	d	163	h	A3	d	164	h	A4	d	165	h	A5	d	166	h	A6	d	167	h	A7
¨				©				ª				«				¬				- *shy*				®				¯			
d	168	h	A8	d	169	h	A9	d	170	h	AA	d	171	h	AB	d	172	h	AC	d	173	h	AD	d	174	h	AE	d	175	h	AF
°				±				²				³				´				µ				¶				·			
d	176	h	B0	d	177	h	B1	d	178	h	B2	d	179	h	B3	d	180	h	B4	d	181	h	B5	d	182	h	B6	d	183	h	B7
¸				¹				º				»				¼				½				¾				¿			
d	184	h	B8	d	185	h	B9	d	186	h	BA	d	187	h	BB	d	188	h	BC	d	189	h	BD	d	190	h	BE	d	191	h	BF
À				Á				Â				Ã				Ä				Å				Æ				Ç			
d	192	h	C0	d	193	h	C1	d	194	h	C2	d	195	h	C3	d	196	h	C4	d	197	h	C5	d	198	h	C6	d	199	h	C7
È				É				Ê				Ë				Ì				Í				Î				Ï			
d	200	h	C8	d	201	h	C9	d	202	h	CA	d	203	h	CB	d	204	h	CC	d	205	h	CD	d	206	h	CE	d	207	h	CF
Ð				Ñ				Ò				Ó				Ô				Õ				Ö				×			
d	208	h	D0	d	209	h	D1	d	210	h	D2	d	211	h	D3	d	212	h	D4	d	213	h	D5	d	214	h	D6	d	215	h	D7
Ø				Ù				Ú				Û				Ü				Ý				Þ				ß			
d	216	h	D8	d	217	h	D9	d	218	h	DA	d	219	h	DB	d	220	h	DC	d	221	h	DD	d	222	h	DE	d	223	h	DF

Таблица 3.6 *(окончание)*

à				**á**				**â**				**ã**				**ä**				**å**				**æ**				**ç**			
d	224	h	E0	d	225	h	E1	d	226	h	E2	d	227	h	E3	d	228	h	E4	d	229	h	E5	d	230	h	E6	d	231	h	E7
è				**é**				**ê**				**ë**				**ì**				**í**				**î**				**ï**			
d	232	h	E8	d	233	h	E9	d	234	h	EA	d	235	h	EB	d	236	h	EC	d	237	h	ED	d	238	h	EE	d	239	h	EF
ð				**ñ**				**ò**				**ó**				**ô**				**õ**				**ö**				**÷**			
d	240	h	F0	d	241	h	F1	d	242	h	F2	d	243	h	F3	d	244	h	F4	d	245	h	F5	d	246	h	F6	d	247	h	F7
ø				**ù**				**ú**				**û**				**ü**				**ý**				**þ**				**ÿ**			
d	248	h	F8	d	249	h	F9	d	250	h	FA	d	251	h	FB	d	252	h	FC	d	253	h	FD	d	254	h	FE	d	255	h	FF

Кроме символов, характерных для европейских языков, здесь можно увидеть некоторые знаки, употребляющиеся и в тексте на русском языке, — например, кавычки-«елочки» и знак параграфа. (Остальные символы, о которых шла речь в *главе 2*, найдите сами.)

Примечание

Не запутайтесь: в расширенной ASCII всего 256 символов, но десятичный код последнего символа — 255, а не 256. Аналогично, последний код в стандартной ASCII — 127, а не 128!

Дополнительно

Приведем несколько «нестандартных» символов латиницы, которые могут встретиться в текстах на европейских языках (табл. 3.7).

Таблица 3.7. *Дополнительные символы латиницы*

Язык	Символы
Английский	Æ æ Œ œ
Датский, норвежский	Å å Æ æ Ø ø
Немецкий	Ä ä Ö ö Ü ü ß
Польский	Ą ą Ć ć Ę ę Ł ł Ń ń Ó ó Ś ś Ź ź Ż ż
Французский	À à Â â Ç ç É é È è Ë ë Ê ê Î î Ï ï Ô ô Œ œ Ù ù Ü ü Û û

Символы, составленные из двух и более букв (такие, как Æ или Œ), называются *лигатурами*.

Значки, поставленные сверху, снизу или сбоку от символов, называются *акцентировочными* или *диакритическими знаками*. Вот некоторые из часто встречающихся диакритических знаков (табл. 3.8).

Таблица 3.8. *Диакритические знаки*

Название	Диакритический знак	Примеры символов
Акут	´	É é Ź ź
Гравис	`	È è
Умляут	¨	Ä ä Ë ë
Гачек	ˇ	Ě ě Ň ň
Бреве, знак краткости	˘	Й й
Седиль	¸	Ç ç

Неразрывный пробел и мягкий перенос

Два символа Latin-1 заслуживают особого рассмотрения (табл. 3.9).

Таблица 3.9. *Неразрывный пробел и мягкий перенос*

Символ	Условное обозначение	Код	
		десятичный	шестнадцатеричный
Неразрывный пробел	nbsp	160	A0
Мягкий перенос	shy	173	AD

Неразрывный пробел может пригодиться там, где необходимо запретить перевод строки. Например, если в сокращении «и т. д.» поставить не обычные, а неразрывные пробелы, то все три сокращенных слова всегда останутся на одной и той же строке.

Включение в текст неразрывных пробелов — один из способов предотвращения разрыва строки. Но этот способ не всегда является лучшим. В *главе 5* будут подробно рассмотрены особенности запрета перевода строки в коде HTML.

Мягкий перенос, наоборот, употребляется для того, чтобы указать на возможное место перевода строки там, где она обычно не переводится. Такие символы можно расставить внутри слова, обозначив места возможных переносов (например: *не-раз-рыв-ный*). Если слово с мягкими переносами окажется в середине строки, оно будет выглядеть обычным образом, и переносы проигнорируются. Но если слово попадет в конец строки, то многие устройства

просмотра (браузеры, текстовые процессоры и др.) разорвут его на месте одного из мягких переносов и поставят в конце строки обычный знак переноса.

Впрочем, с мягкими переносами связаны отдельные сложности, которые будут также обсуждаться в *главе 5*.

Дополнительно

В некоторых кодировках, отличных от ISO 8859, в позициях 128–159 могут находиться отображаемые символы. Например, в кодировке Windows-1251 символ с кодом 150 — это короткое тире, а символ с кодом 151 — длинное. Однако, даже если вы разрабатываете документ в этой кодировке, не стоит употреблять эти символы. Для того чтобы поместить в документ короткое или длинное тире (или любой другой специальный символ), рекомендуется пользоваться стандартными подстановками HTML или символами Unicode (*см. главу 5*).

Символ	Правильно	Неправильно
Короткое тире	`–` `–`	`–`
Длинное тире	`—` `—`	`—`

Примечание

Вам может показаться, что мы заскочили куда-то далеко вперед: какие-то «подстановки HTML», непонятные коды вроде `—`... Автор спешит напомнить, что фрагменты, помеченные знаком 纸, рекомендуется осваивать, когда вы читаете книгу *второй* раз. К этому времени вы будете знать, и что такое подстановки, и еще многое другое.

3.1.5. Кириллица в расширенных кодировках ASCII

> Кириллица и глаголица долгое время развивались параллельно, хотя кириллица скоро почти полностью вытеснила глаголицу в Болгарии, а затем и в Македонии. В XI–XII вв. глаголица встречается на Руси только в виде вставок в кириллических текстах.
>
> *В. Ефимов. Кириллица*

Среди кодовых таблиц, определенных в стандарте ISO 8859, есть таблица ISO-8859-5 (табл. 3.10), которая содержит символы кириллицы (русского алфавита).

Таблица 3.10. *Кодовая таблица ISO-8859-5*

nbsp				**Ё**				**Ѣ**				**Ѓ**				**Є**				**S**				**І**				**Ї**			
d	160	h	A0	d	161	h	A1	d	162	h	A2	d	163	h	A3	d	164	h	A4	d	165	h	A5	d	166	h	A6	d	167	h	A7
Ј				**Ѣ**				**Ѣ**				**Ѣ**				**Ѓ**				***sh̄y***				**Ў**				**Ц**			
d	168	h	A8	d	169	h	A9	d	170	h	AA	d	171	h	AB	d	172	h	AC	d	173	h	AD	d	174	h	AE	d	175	h	AF
А				**Б**				**В**				**Г**				**Д**				**Е**				**Ж**				**З**			
d	176	h	B0	d	177	h	B1	d	178	h	B2	d	179	h	B3	d	180	h	B4	d	181	h	B5	d	182	h	B6	d	183	h	B7
И				**Й**				**К**				**Л**				**М**				**Н**				**О**				**П**			
d	184	h	B8	d	185	h	B9	d	186	h	BA	d	187	h	BB	d	188	h	BC	d	189	h	BD	d	190	h	BE	d	191	h	BF
Р				**С**				**Т**				**У**				**Ф**				**Х**				**Ц**				**Ч**			
d	192	h	C0	d	193	h	C1	d	194	h	C2	d	195	h	C3	d	196	h	C4	d	197	h	C5	d	198	h	C6	d	199	h	C7
Ш				**Щ**				**Ъ**				**Ы**				**Ь**				**Э**				**Ю**				**Я**			
d	200	h	C8	d	201	h	C9	d	202	h	CA	d	203	h	CB	d	204	h	CC	d	205	h	CD	d	206	h	CE	d	207	h	CF
а				**б**				**в**				**г**				**д**				**е**				**ж**				**з**			
d	208	h	D0	d	209	h	D1	d	210	h	D2	d	211	h	D3	d	212	h	D4	d	213	h	D5	d	214	h	D6	d	215	h	D7
и				**й**				**к**				**л**				**м**				**н**				**о**				**п**			
d	216	h	D8	d	217	h	D9	d	218	h	DA	d	219	h	DB	d	220	h	DC	d	221	h	DD	d	222	h	DE	d	223	h	DF
р				**с**				**т**				**у**				**ф**				**х**				**ц**				**ч**			
d	224	h	E0	d	225	h	E1	d	226	h	E2	d	227	h	E3	d	228	h	E4	d	229	h	E5	d	230	h	E6	d	231	h	E7
ш				**щ**				**ъ**				**ы**				**ь**				**э**				**ю**				**я**			
d	232	h	E8	d	233	h	E9	d	234	h	EA	d	235	h	EB	d	236	h	EC	d	237	h	ED	d	238	h	EE	d	239	h	EF
№				**ё**				**ѣ**				**ѓ**				**є**				**s**				**і**				**ї**			
d	240	h	F0	d	241	h	F1	d	242	h	F2	d	243	h	F3	d	244	h	F4	d	245	h	F5	d	246	h	F6	d	247	h	F7
ј				**ѣ**				**ѣ**				**ѣ**				**ѣ**				**§**				**ў**				**ц**			
d	248	h	F8	d	249	h	F9	d	250	h	FA	d	251	h	FB	d	252	h	FC	d	253	h	FD	d	254	h	FE	d	255	h	FF

В этой таблице мы видим символы, использующиеся в русском, украинском, белорусском, болгарском, сербском, македонском и некоторых других языках. Здесь также появился знак номера, которого не было в Latin-1.

Примечание

Обратите внимание: неразрывный пробел и мягкий перенос остались на тех же местах, что и в Latin-1.

Кириллица ISO не получила широкого распространения. Наиболее популярными стали другие кодировки, которые были встроены в распространенные операционные системы. Например, в DOS и OS/2 «родной» кодировкой была Codepage 866 (табл. 3.11). Ее сокращенно обозначают CP866.

Таблица 3.11. *Кодовая таблица CP866*

А				**Б**				**В**				**Г**				**Д**				**Е**				**Ж**				**З**			
d	128	h	80	d	129	h	81	d	130	h	82	d	131	h	83	d	132	h	84	d	133	h	85	d	134	h	86	d	135	h	87
И				**Й**				**К**				**Л**				**М**				**Н**				**О**				**П**			
d	136	h	88	d	137	h	89	d	138	h	8A	d	139	h	8B	d	140	h	8C	d	141	h	8D	d	142	h	8E	d	143	h	8F
Р				**С**				**Т**				**У**				**Ф**				**Х**				**Ц**				**Ч**			
d	144	h	90	d	145	h	91	d	146	h	92	d	147	h	93	d	148	h	94	d	149	h	95	d	150	h	96	d	151	h	97
Ш				**Щ**				**Ъ**				**Ы**				**Ь**				**Э**				**Ю**				**Я**			
d	152	h	98	d	153	h	99	d	154	h	9A	d	155	h	9B	d	156	h	9C	d	157	h	9D	d	158	h	9E	d	159	h	9F
а				**б**				**в**				**г**				**д**				**е**				**ж**				**з**			
d	160	h	A0	d	161	h	A1	d	162	h	A2	d	163	h	A3	d	164	h	A4	d	165	h	A5	d	166	h	A6	d	167	h	A7
и				**й**				**к**				**л**				**м**				**н**				**о**				**п**			
d	168	h	A8	d	169	h	A9	d	170	h	AA	d	171	h	AB	d	172	h	AC	d	173	h	AD	d	174	h	AE	d	175	h	AF
░				▒				▓				│				┤				╡				╢				╖			
d	176	h	B0	d	177	h	B1	d	178	h	B2	d	179	h	B3	d	180	h	B4	d	181	h	B5	d	182	h	B6	d	183	h	B7
╕				╣				║				╗				╝				╜				╛				┐			
d	184	h	B8	d	185	h	B9	d	186	h	BA	d	187	h	BB	d	188	h	BC	d	189	h	BD	d	190	h	BE	d	191	h	BF
└				┴				┬				├				─				┼				╞				╟			
d	192	h	C0	d	193	h	C1	d	194	h	C2	d	195	h	C3	d	196	h	C4	d	197	h	C5	d	198	h	C6	d	199	h	C7
╚				╩				╦				╠				═				╬				╧				╨			
d	200	h	C8	d	201	h	C9	d	202	h	CA	d	203	h	CB	d	204	h	CC	d	205	h	CD	d	206	h	CE	d	207	h	CF
╤				╥				╙				╘				╒				╓				╫				╪			
d	208	h	D0	d	209	h	D1	d	210	h	D2	d	211	h	D3	d	212	h	D4	d	213	h	D5	d	214	h	D6	d	215	h	D7

Таблица 3.11 *(окончание)*

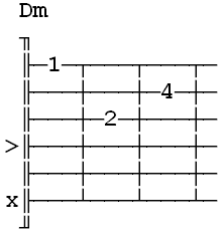
╪	┘	┌	█	▄	▌	▐	▀
d 216 h D8	d 217 h D9	d 218 h DA	d 219 h DB	d 220 h DC	d 221 h DD	d 222 h DE	d 223 h DF
р	**с**	**т**	**у**	**ф**	**х**	**ц**	**ч**
d 224 h E0	d 225 h E1	d 226 h E2	d 227 h E3	d 228 h E4	d 229 h E5	d 230 h E6	d 231 h E7
ш	**щ**	**ъ**	**ы**	**ь**	**э**	**ю**	**я**
d 232 h E8	d 233 h E9	d 234 h EA	d 235 h EB	d 236 h EC	d 237 h ED	d 238 h EE	d 239 h EF
Ё	**ё**	**Є**	**є**	**Ї**	**ї**	**Ў**	**ў**
d 240 h F0	d 241 h F1	d 242 h F2	d 243 h F3	d 244 h F4	d 245 h F5	d 246 h F6	d 247 h F7
°	**•**	**·**	**√**	**№**	**¤**	**■**	***nbsp***
d 248 h F8	d 249 h F9	d 250 h FA	d 251 h FB	d 252 h FC	d 253 h FD	d 254 h FE	d 255 h FF

Здесь стоит обратить внимание на две особенности:

- в кодировке CP866 позиции 128–159 не заняты управляющими символами, а отведены под обычные буквы;
- коды 176–223 соответствуют символам *псевдографики*. К псевдографике относятся не только такие рамки, но и вообще любые значки, пиктограммы и изображения, не являющиеся обычными текстовыми символами, но включенные в кодовую таблицу. Когда-то псевдографика была мощным средством для рисования рамок и схем (рис. 3.8).

DEC	BIN
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

а



б

Рис. 3.8. Набранные с помощью псевдографики таблица (*а*) и схема (*б*)

纸 Дополнительно

На примере кодировки CP866 можно рассмотреть одну проблему, возникающую из-за использования кодов 128–159 под отображаемые символы.

В сети Fidonet именно CP866 была стандартной кодировкой для сообщений на русском языке. Однако большинство программ, служащих для обработки и пересылки сообщений, воспринимали символ с кодом 141 как служебный. Поэтому прописная русская буква «Н» в сообщениях пропадала. Чтобы этого не случалось, приходилось при наборе сообщения заменять русскую «Н» на аналогичную по начертанию латинскую «H» («аш»).

В некоторых случаях возникали проблемы и с русской «р». Ее также приходилось заменять на латинскую «p» («пэ»).

Трудности подстерегали не только при вводе сообщения, но и при последующей работе с ним. Однажды автору передали крупный текст, взятый из Fidonet (о чем автор поначалу не догадывался). В тексте нужно было найти фрагмент, содержащий словосочетание «Новый год» — и программа поиска не справилась с этим простым заданием. В самом деле: ей поручили искать слова по-русски, но в тексте искомое словосочетание начиналось с *латинской* буквы «H»!

Сейчас кодировка CP866 практически вышла из употребления. На русскоязычных сайтах господствуют две кодировки: Windows-1251 (табл. 3.12), принятая в операционных системах Windows, и KOI8-R (табл. 3.13) — «родная» кодировка кириллицы в UNIX.

Таблица 3.12. *Кодовая таблица Windows-1251*

Ђ				Ѓ				‚				ѓ				„				…				†				‡	
d	128	h	80	d	129	h	81	d	130	h	82	d	131	h	83	d	132	h	84	d	133	h	85	d	134	h	86	d	135	h	87
Є				‰				Љ				‹				Њ				Ќ				Ћ				Џ			
d	136	h	88	d	137	h	89	d	138	h	8A	d	139	h	8B	d	140	h	8C	d	141	h	8D	d	142	h	8E	d	143	h	8F
ђ				‘				’				“				”				•				–				—			
d	144	h	90	d	145	h	91	d	146	h	92	d	147	h	93	d	148	h	94	d	149	h	95	d	150	h	96	d	151	h	97
				™				љ				›				‌				‍				‎				‏			
d	152	h	98	d	153	h	99	d	154	h	9A	d	155	h	9B	d	156	h	9C	d	157	h	9D	d	158	h	9E	d	159	h	9F
nbsp				Ў				ў				J				¤				Ґ				¦				§			
d	160	h	A0	d	161	h	A1	d	162	h	A2	d	163	h	A3	d	164	h	A4	d	165	h	A5	d	166	h	A6	d	167	h	A7
Ё				©				Є				«				¬				- *shy*				®				Ї			
d	168	h	A8	d	169	h	A9	d	170	h	AA	d	171	h	AB	d	172	h	AC	d	173	h	AD	d	174	h	AE	d	175	h	AF
°				±				І				і				ґ				µ				¶				·			
d	176	h	B0	d	177	h	B1	d	178	h	B2	d	179	h	B3	d	180	h	B4	d	181	h	B5	d	182	h	B6	d	183	h	B7

Таблица 3.12 *(окончание)*

ё	**№**	**є**	**»**	**j**	**S**	**s**	**ї**
d 184 h B8	d 185 h B9	d 186 h BA	d 187 h BB	d 188 h BC	d 189 h BD	d 190 h BE	d 191 h BF
А	**Б**	**В**	**Г**	**Д**	**Е**	**Ж**	**З**
d 192 h C0	d 193 h C1	d 194 h C2	d 195 h C3	d 196 h C4	d 197 h C5	d 198 h C6	d 199 h C7
И	**Й**	**К**	**Л**	**М**	**Н**	**О**	**П**
d 200 h C8	d 201 h C9	d 202 h CA	d 203 h CB	d 204 h CC	d 205 h CD	d 206 h CE	d 207 h CF
Р	**С**	**Т**	**У**	**Ф**	**Х**	**Ц**	**Ч**
d 208 h D0	d 209 h D1	d 210 h D2	d 211 h D3	d 212 h D4	d 213 h D5	d 214 h D6	d 215 h D7
Ш	**Щ**	**Ъ**	**Ы**	**Ь**	**Э**	**Ю**	**Я**
d 216 h D8	d 217 h D9	d 218 h DA	d 219 h DB	d 220 h DC	d 221 h DD	d 222 h DE	d 223 h DF
а	**б**	**в**	**г**	**д**	**е**	**ж**	**з**
d 224 h E0	d 225 h E1	d 226 h E2	d 227 h E3	d 228 h E4	d 229 h E5	d 230 h E6	d 231 h E7
и	**й**	**к**	**л**	**м**	**н**	**о**	**п**
d 232 h E8	d 233 h E9	d 234 h EA	d 235 h EB	d 236 h EC	d 237 h ED	d 238 h EE	d 239 h EF
р	**с**	**т**	**у**	**ф**	**х**	**ц**	**ч**
d 240 h F0	d 241 h F1	d 242 h F2	d 243 h F3	d 244 h F4	d 245 h F5	d 246 h F6	d 247 h F7
ш	**щ**	**ъ**	**ы**	**ь**	**э**	**ю**	**я**
d 248 h F8	d 249 h F9	d 250 h FA	d 251 h FB	d 252 h FC	d 253 h FD	d 254 h FE	d 255 h FF

Таблица 3.13. *Кодовая таблица KOI8-R*

─	│	┌	┐	└	┘	├	┤
d 128 h 80	d 129 h 81	d 130 h 82	d 131 h 83	d 132 h 84	d 133 h 85	d 134 h 86	d 135 h 87
┬	┴	┼	▀	▄	█	▌	▐
d 136 h 88	d 137 h 89	d 138 h 8A	d 139 h 8B	d 140 h 8C	d 141 h 8D	d 142 h 8E	d 143 h 8F
░	▒	▓	⌠	■	•	√	≈
d 144 h 90	d 145 h 91	d 146 h 92	d 147 h 93	d 148 h 94	d 149 h 95	d 150 h 96	d 151 h 97
≤	≥	*nbsp*	⌡	°	²	·	÷
d 152 h 98	d 153 h 99	d 154 h 9A	d 155 h 9B	d 156 h 9C	d 157 h 9D	d 158 h 9E	d 159 h 9F
═	║	╒	ё	╓	╔	╕	╖
d 160 h A0	d 161 h A1	d 162 h A2	d 163 h A3	d 164 h A4	d 165 h A5	d 166 h A6	d 167 h A7

Таблица 3.13. *(окончание)*

╕	╖	╗	╘	╙	╚	╛	╜
d 168 h A8	d 169 h A9	d 170 h AA	d 171 h AB	d 172 h AC	d 173 h AD	d 174 h AE	d 175 h AF
╝	╞	╟	Ё	╠	╡	╢	╣
d 176 h B0	d 177 h B1	d 178 h B2	d 179 h B3	d 180 h B4	d 181 h B5	d 182 h B6	d 183 h B7
╤	╥	╦	╧	╨	╩	©	
d 184 h B8	d 185 h B9	d 186 h BA	d 187 h BB	d 188 h BC	d 189 h BD	d 190 h BE	d 191 h BF
ю	**а**	**б**	**ц**	**д**	**е**	**ф**	**г**
d 192 h C0	d 193 h C1	d 194 h C2	d 195 h C3	d 196 h C4	d 197 h C5	d 198 h C6	d 199 h C7
х	**и**	**й**	**к**	**л**	**м**	**н**	**о**
d 200 h C8	d 201 h C9	d 202 h CA	d 203 h CB	d 204 h CC	d 205 h CD	d 206 h CE	d 207 h CF
п	**я**	**р**	**с**	**т**	**у**	**ж**	**в**
d 208 h D0	d 209 h D1	d 210 h D2	d 211 h D3	d 212 h D4	d 213 h D5	d 214 h D6	d 215 h D7
ь	**ы**	**з**	**ш**	**э**	**щ**	**ч**	**ъ**
d 216 h D8	d 217 h D9	d 218 h DA	d 219 h DB	d 220 h DC	d 221 h DD	d 222 h DE	d 223 h DF
Ю	**А**	**Б**	**Ц**	**Д**	**Е**	**Ф**	**Г**
d 224 h E0	d 225 h E1	d 226 h E2	d 227 h E3	d 228 h E4	d 229 h E5	d 230 h E6	d 231 h E7
Х	**И**	**Й**	**К**	**Л**	**М**	**Н**	**О**
d 232 h E8	d 233 h E9	d 234 h EA	d 235 h EB	d 236 h EC	d 237 h ED	d 238 h EE	d 239 h EF
П	**Я**	**Р**	**С**	**Т**	**У**	**Ж**	**В**
d 240 h F0	d 241 h F1	d 242 h F2	d 243 h F3	d 244 h F4	d 245 h F5	d 246 h F6	d 247 h F7
Ь	**Ы**	**З**	**Ш**	**Э**	**Щ**	**Ч**	**Ъ**
d 248 h F8	d 249 h F9	d 250 h FA	d 251 h FB	d 252 h FC	d 253 h FD	d 254 h FE	d 255 h FF

Примечание

Аббревиатура «KOI» происходит от русского словосочетания «Код для обмена информацией». Современную кодовую таблицу KOI8-R разработал наш соотечественник Андрей Чернов. Цифра «8» означает, что эта кодировка — 8-битная.

В KOI8-R входят символы псевдографики, практически не нужные в современных условиях. А в Windows-1251, наоборот, можно найти полезные типографские символы — в частности, грамотные кавычки и оба тире.

Значит ли это, что текст на русском языке лучше всего сохранять в кодировке Windows? Да, эта кодировка действительно удачнее других — но только в том случае, если ваши тексты создаются, хранятся и используются на вашем компьютере, и никуда не передаются. При передаче текстовых файлов с одного компьютера на другой и даже при открытии файла не в том приложении, в котором он был набран, могут возникнуть неприятности.

Проблемы перекодировки

Мы уже рассмотрели проблему с переводами строк, которая может возникнуть, если текст передается с одного компьютера на другой. Теперь познакомимся с проблемами, возникающими из-за несовместимости кодировок.

Иногда по техническим соображениям текст необходимо представить в определенной кодировке. Например, чтобы отправить сообщение по сети Fidonet, оно должно быть в кодировке CP866. А почтовый клиент, которым пользуется автор, по умолчанию отправляет все сообщения в кодировке KOI8-R.

Если исходный текст представлен не в той кодировке, которая требуется, его надо перекодировать. Здесь можно столкнуться с затруднениями: не для каждого символа текста найдется соответствие в новой кодовой таблице. С обычными русскими буквами проблем не возникает — они есть во всех кириллических кодировках. Но что делать, если текст набран в кодировке Windows-1251 и содержит типографские символы, — а перевести его надо в CP866 или KOI8-R, где эти символы отсутствуют? Может получиться неприятный эффект, как на рис. 3.9.

«Плащ подберите, — посоветовал я, глядя на него в зеркало заднего вида. — У вас плащ защемляется». Минут через пять все наконец устроилось. Я спросил: «До Соловца километров десять?» — «Да, — ответил горбоносый. — Или немножко больше. Дорога, правда, неважная — для грузовиков». — «Дорога вполне приличная, — возразил я. — Мне обещали, что я вообще не проеду».

а

лПлащ подберите,Ч посоветовал я, глядя на него в зеркало заднего вида.Ч У вас плащ защемляется¬. Минут через пять все наконец устроилось. Я спросил: лДо Соловца километров десять?¬ Ч лДа,Ч ответил горбоносый.Ч Или немножко больше. Дорога, правда, неважная Ч для грузовиков¬.Ч лДорога вполне приличная, Ч возразил я.Ч Мне обещали, что я вообще не проеду¬.

б

"Плащ подберите, - посоветовал я, глядя на него в зеркало заднего вида. - У вас плащ защемляется". Минут через пять все наконец устроилось. Я спросил: "До Соловца километров десять?" - "Да, - ответил горбоносый. - Или немножко больше. Дорога, правда, неважная - для грузовиков". - "Дорога вполне приличная, - возразил я. - Мне обещали, что я вообще не проеду".

в

Рис. 3.9. При преобразовании файла из Windows-1251 (*а*) в CP866 (*б*) кавычки и тире превратились в белиберду. Единственный способ бороться с этим (в данном случае) — заменить типографские символы на суррогаты (*в*)

Внимание!

Если вы создаете страницу в какой-либо из 8-разрядных расширенных кодировок ASCII, пользуйтесь только такими символами, которые не пострадают при перекодировке. К этим символам относятся:

- знаки из стандартной ASCII (десятичные коды 32–127) — они одинаковы во всех кодовых таблицах;
- обычные буквы русского алфавита.

Особенно нежелательно употреблять символы с кодами 128–159.

Чтобы включить в текст дополнительные символы (например, тире, кавычки или знак «№»), пользуйтесь специальными средствами HTML (например, подстановками — *см. главу 5*).

В недавнем прошлом на заглавных страницах многих сайтов пользователю предлагалось выбрать кодировку, в которой он желает просматривать сайт (рис. 3.10). Сейчас это совершенно нецелесообразно: браузеры и веб-серверы давно уже научились «договариваться» между собой о кодировке автоматически, без участия пользователя. В отдельных редких случаях, когда браузеру все-таки не удается воспроизвести страницу в нужной кодировке, пользователь может вручную установить нужную кодировку в настройках браузера.

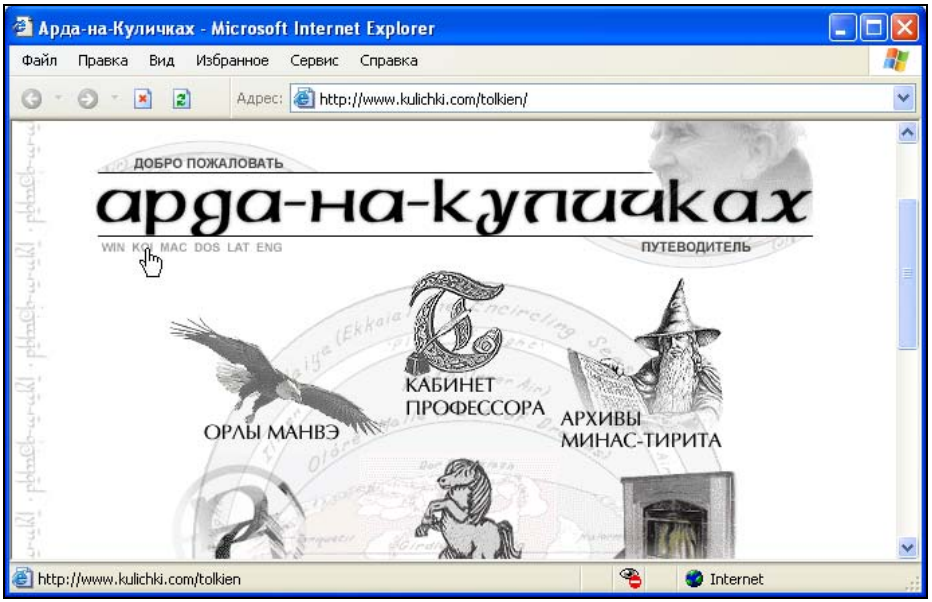


Рис. 3.10. Пользователю предлагается выбрать кодировку для просмотра. На самом деле, к этой функции очень редко обращались даже тогда, когда проблемы кодировок были еще актуальны

纸 Дополнительно

Каким же образом веб-сервер и браузер «договариваются» между собой о кодировке? Если вы интересовались принципами действия протокола HTTP, то, скорее всего, помните, что вместе с веб-страницей сервер передает браузеру некоторое количество служебной информации — в частности, так называемые заголовки HTTP. Каждый заголовок HTTP представляет собой текстовую строку, составленную по определенным синтаксическим правилам и содержащую важную информацию о свойствах самого сервера и запрошенной страницы. Один из этих заголовков — `Content-Type`; он описывает тип передаваемых данных и кодировку (если она важна).

Заголовок `Content-Type`, сопровождающий веб-страницу, может выглядеть, например, следующим образом:

```
Content-Type: text/html; charset=koi8-r
```

Параметр `charset` указывает на то, что страница закодирована в KOI8-R. Получив этот заголовок, браузер подключает нужную кодовую таблицу из своего набора и отображает документ в соответствии с ней.

Настройка заголовков HTTP находится в компетенции администратора веб-сервера. При профессиональном подходе к размещению сайтов разработчик веб-страниц и администратор веб-сервера договариваются между собой о том, чтобы кодировка страниц и параметр `charset` в заголовках HTTP соответствовали друг другу.

Некоторое время назад применялся и другой способ указания кодировки, не связанный с настройками веб-сервера. Разработчик включал в раздел `head` документа тэг `meta` — например, следующего вида:

```
<meta http-equiv="Content-Type" content="text/html; charset=koi8-r" />
```

Этот способ был нежелательным и тогда, когда им широко пользовались, а сейчас его следует окончательно признать устаревшим. Если параметр `charset`, указанный в тэге `meta`, совпадает с параметром из заголовков HTTP, — то тэг `meta` просто лишний. А если не совпадает — то этот тэг тем более не нужен, так как может ввести браузер в заблуждение.

Русский текст сложнее поддается автоматической обработке, чем английский. Например, простая задача — написать программу, которая преобразует букву из прописной в строчную или наоборот — для английского языка решается элементарно: нужно прибавить к коду буквы (или вычесть из него) 32.

В русском языке все гораздо труднее. Чтобы правильно преобразовать букву, нужно, во-первых, знать кодировку текста. А во-вторых, во многих кодировках от самой буквы также зависит, что нужно прибавить или вычесть из ее кода, чтобы сменить регистр. Поэтому вместо простого правила «±32» программисту приходится пользоваться громоздкими алгоритмами или готовыми таблицами преобразования.

Неприятности также подстерегают и при сортировке. Программа может легко отсортировать по алфавиту английские слова или текстовые строки, ори-

ентируясь по положению букв в кодовой таблице. Для русского языка это не всегда удается: например, в кодировке CP866 строчные буквы расположены с разрывом, а в KOI8-R и вовсе переставлены в совершенно диком порядке. Поэтому снова приходится прибегать к сложным алгоритмам или таблицам.

Примечание

Дополнительные сложности связаны с многострадальной буквой «Ё». Многие поисковые системы не отличают ее от «Е». Даже если разработчик сайта постарался и расставил «Ё» всюду, где надо, — такие поисковые системы сведут его усилия на нет. Пользователь будет искать «вёдро» — а найдет множество сайтов, где упоминается «ведро». Он будет искать информацию о фашистском деятеле по фамилии Рём — а найдет сайты, где встречается слово «Рем».

Многие алгоритмы сортировки, применяющиеся в современных программных средствах, также считают «Е» и «Ё» одной и той же буквой, в результате чего помещают слово «ёжик» между «ежевикой» и «ежихой». А некоторые программы и вовсе не считают «Ё» буквой русского алфавита и помещают слова, начинающиеся с «Ё», в самое начало алфавитного перечня.

Если вы разрабатываете текст, с которым должны взаимодействовать программы автоматической обработки и поиска, а тем более если вы сами пишете такие программы, — пожалуйста, предусмотрите возможные проблемы с буквой «Ё» и постарайтесь преодолеть их.

Примечание

Какую же из расширенных кодировок ASCII лучше всего использовать при разработке веб-страниц? Выбор невелик — либо Windows-1251, либо KOI8-R. Используемые сейчас браузеры поддерживают обе эти кодировки одинаково хорошо, так что, скорее всего, имеет смысл пользоваться той, которая наиболее привычна для разработчика. Например, если вы работаете в операционной системе Windows, удобнее разрабатывать страницы в кодировке Windows-1251: это придаст вам уверенность, что такие файлы будут корректно открываться и просматриваться во всех текстовых редакторах и других программах в вашей системе. В системах класса UNIX, наоборот, более целесообразно пользоваться кодировкой KOI8-R.

3.1.6. Unicode — решение многих проблем

Если улицы, мосты,
Переулки, лестницы
Навсегда в себя вместил, —
Все во мне поместится.

Г. Шпаликов

Расширенные кодировки ASCII позволяли представлять тексты на языках, отличных от английского, но это не решало все проблемы.

Например, используя расширенные ASCII, можно было написать текст по-русски или по-немецки, но нельзя было написать текст, в котором присутствуют одновременно и русские буквы, и специфические немецкие символы — ä, ö, ü, ß. Кроме того, существуют языки — например, китайский, — алфавит которых просто не умещается на 128 позициях расширенной ASCII.

Эти проблемы решались по-разному — например, путем создания специальных шрифтов или использования альтернативных кодировок. Впрочем, практически все решения ухудшали совместимость и вызывали дополнительные трудности при переносе текста с одного компьютера на другой.

Удачным выходом из положения стало создание кодовой таблицы, в которую входят сразу *все* символы из всех используемых в мире алфавитов. Эта кодировка наиболее известна как Unicode (Юникод).

ISO 10646 и Unicode

В начале 1990-х несколько компаний и организаций объединили свои усилия по созданию всеобъемлющей кодовой таблицы, и в результате на свет появился стандарт ISO 10646. В этом стандарте определен Универсальный набор символов (Universal Character Set, UCS), содержащий десятки тысяч символов из разных языков мира (табл. 3.14).

Каждому символу в UCS соответствует код из 31 бита. Это значит, что всего в такой кодовой таблице можно закодировать 2^{31} = 2 147 483 648 различных символов. Этого с лихвой хватит на все существующие в мире языки, включая придуманные — вроде эльфийских языков Дж. Р. Р. Толкина или языка клингон из фантастического сериала «Star Track».

Конечно, далеко не все позиции в UCS уже заполнены. Международная организация по стандартизации (International Organization for Standardization, ISO) время от времени добавляет на свободные места новые символы. После того как символ закреплен в UCS, он остается там навсегда, и его код больше не будет меняться.

Внимание!

Каждый символ в UCS обозначается `U+`*`число`*, где *`число`* — это шестнадцатеричный код символа. Кроме того, каждому символу присвоено собственное название.

Таблица 3.14. *Некоторые символы из Universal Character Set*

Символ	Обозначение	Название
Q	U+0051	LATIN CAPITAL LETTER Q
я	U+044F	CYRILLIC SMALL LETTER YA

Таблица 3.14 *(окончание)*

Символ	Обозначение	Название
«	U+00AB	LEFT-POINTING DOUBLE ANGLE QUOTATION MARK
—	U+2014	EM DASH
א	U+05D0	HEBREW LETTER ALEF
ᚨ	U+16A8	RUNIC LETTER ANSUZ A
∮	U+222E	CONTOUR INTEGRAL
☮	U+262E	PEACE SYMBOL
𝄞	U+1D11E	MUSICAL SYMBOL G CLEF

По одной этой таблице можно оценить всеохватность UCS. В самом деле — какой бы знак или символ ни пришел вам в голову, вы можете быть уверены, что он уже есть или скоро будет в UCS.

Наиболее важной частью UCS является набор из первых 65 536 символов — так называемый Базовый многоязычный модуль (Basic Multilingual Plane, BMP). Каждому из таких символов соответствует 16-разрядный код. В шестнадцатеричной системе этот код записывается ровно четырьмя цифрами. Посмотрите: в табл. 3.11 все символы, кроме одного, входят в BMP.

Примечание

Наиболее свежие версии стандарта, описывающего UCS, на момент написания этой книги (середина 2004):

- ISO/IEC 10646-1:2000. Information technology — Universal Multiple-Octet Coded Character Set (UCS) — Part 1: Architecture and Basic Multilingual Plane (общие принципы UCS и Базовый многоязычный модуль);
- ISO/IEC 10646-2:2001. Information technology — Universal Multiple-Octet Coded Character Set (UCS) — Part 2: Supplementary Planes (символы, не входящие в Базовый многоязычный модуль).

Unicode — это стандарт, развивающийся параллельно ISO 10646 и описывающий тот же Универсальный набор символов. Между этими двумя стандартами есть ряд различий, однако они касаются в основном не символов и их кодов, а дополнительной справочной информации — например, описаний символов и различных вариантов их начертаний.

纸 Дополнительно

В стандарте Unicode, например, описаны алгоритмы вывода сложных сочетаний символов (в частности, из арабского языка), сортировки и поиска. А в ISO 10646 содержится гораздо более подробная, чем в Unicode, информация об иероглифах, используемых в восточных языках (так называемые символы CJK — сокращение от «Chinese, Japanese and Korean»).

Еще одно немаловажное различие между ISO 10646 и Unicode — в том, что Международная организация по стандартизации распространяет свои стандарты только на платной основе. А стандарт Unicode распространяется бесплатно и доступен на официальном сайте **www.unicode.org**.

Именно название «Unicode» («Юникод») чаще всего употребляется для обозначения кодировок и кодовых таблиц, соответствующих UCS. В дальнейшем мы тоже будем пользоваться именно этим обозначением, т. к. в этой книге пойдет речь только об общей части обоих стандартов.

Таблица символов Unicode разделена на диапазоны кодов, каждый из которых охватывает какой-нибудь один алфавит или набор знаков. Вот несколько таких диапазонов (табл. 3.15).

Таблица 3.15. *Некоторые диапазоны кодов Unicode*

Диапазон	Стандартное название	Значение	Примеры символов
0000–007F	C0 Controls and Basic Latin	Управляющие символы ASCII и основная латиница	A a 1 2 & ! ( )
0080–00FF	C1 Controls and Latin-1 Supplement	Управляющие символы расширенной ASCII и расширенная латиница	© æ Ü è £ « »
0370–03FF	Greek and Coptic	Греческий и коптский алфавит	αβγΣΨϢϫ
0400–04FF	Cyrillic	Кириллица	АБяѣѢѪЭѾ
0590–05FF	Hebrew	Иврит	וישגבא
0600–06FF	Arabic	Арабский алфавит	ﻋﺤﻗﺑا
2000–206F	General Punctuation	Типографские символы	– — “ ” ... ‰ ※
20A0–20CF	Currency Symbols	Обозначения валют	€ £ ₪ ₧
2100–214F	Letterlike Symbols	Буквоподобные символы	№ °C ™ ℕ ℥ Å
2200–22FF	Mathematical Operators	Математические символы	∀ ∃ ∈ ∫ ∝ ≝

Таблица 3.15 *(окончание)*

Диапазон	Стандартное название	Значение	Примеры символов
2500–257F	Box Drawing	Рамки (псевдографика)	╞ ┼ ╧ ╡ ╎ ┼
2600–26FF	Miscellaneous Symbols	Пиктограммы	☺ ♂ ♏ ♋ ♣ ♛ ♯
4E00–9FAF	CJK Unified Ideographs	Иероглифы	纸 纖 乩 亜 企 倹
FB00–FB4F	Alphabetic Presentation Forms	Лигатуры и комбинированные символы	ﬁ ﬃ ﬩ ﬗ
1D100–1D1FF	Musical Symbols	Музыкальные знаки	𝄞 𝄋 𝅘𝅥𝅮 𝅝

Примечание

Полную таблицу символов Unicode можно найти в Интернете по адресу: **www.unicode.org/charts**.

Как нетрудно заметить, первые 128 символов Unicode (шестнадцатеричные коды 0000–007F) в точности совпадают со стандартной ASCII. А следующие 128 символов — не что иное, как расширенная кодировка ISO-8859-1 (Latin-1). Это сделано из соображений совместимости — чтобы со старыми текстами, набранными в ASCII, можно было работать в современных редакторах и программах просмотра, поддерживающих Unicode.

Не все символы в Unicode входят именно в тот диапазон, который для них предназначен. Например, угловые кавычки по смыслу относятся к типографским символам — но находятся в диапазоне Latin-1. Знаки доллара ($) и фунта (£) пришлись бы к месту в диапазоне для обозначений валют (20A0–20CF), но расположены на других позициях. Это объясняется историческим развитием Unicode и требованием совместимости со старыми кодировками, основанными на ASCII.

Дополнительно

В английском и некоторых других языках используются особые лигатуры, служащие для улучшения внешнего вида текста. Например, буква «f» из-за своего хвостика не очень красиво смотрится в сочетании с некоторыми другими буквами — например, «i», «l» или еще одной «f». Поэтому в типографском деле вместо отдельных литер — например, «f» и «i», используется единая литера «ﬁ». (Присмотритесь внимательно: она выглядит совсем не так, как просто написанные рядом буквы «fi».)

	040	041	042	043	044	045	046	047	048	049	04A	04B	04C	04D	04E	04F
0	Ѐ 0400	А 0410	Р 0420	а 0430	р 0440	ѐ 0450	Ѿ 0460	Ѱ 0470	Ҁ 0480	Ґ 0490	Ҡ 04A0	Ҳ 04B0	І 04C0	Ӑ 04D0	Ӡ 04E0	Ӱ 04F0
1	Ё 0401	Б 0411	С 0421	б 0431	с 0441	ё 0451	ѿ 0461	ѱ 0471	ҁ 0481	ґ 0491	ҡ 04A1	ҳ 04B1	Ӂ 04C1	ӑ 04D1	ӡ 04E1	ӱ 04F1
2	Ђ 0402	В 0412	Т 0422	в 0432	т 0442	ђ 0452	Ѣ 0462	Ѳ 0472	≠ 0482	Ғ 0492	Ӊ 04A2	Ӿ 04B2	ӂ 04C2	Ӓ 04D2	Ӣ 04E2	Ӳ 04F2
3	Ѓ 0403	Г 0413	У 0423	г 0433	у 0443	ѓ 0453	ѣ 0463	ѳ 0473	◌҃ 0483	ғ 0493	ӊ 04A3	ӿ 04B3	Ӄ 04C3	ӓ 04D3	ӣ 04E3	ӳ 04F3
4	Є 0404	Д 0414	Ф 0424	д 0434	ф 0444	є 0454	Ѥ 0464	Ѵ 0474	◌҄ 0484	Ҕ 0494	Ӈ 04A4	Ҵ 04B4	ӄ 04C4	Ӕ 04D4	Ӥ 04E4	Ӵ 04F4
5	Ѕ 0405	Е 0415	Х 0425	е 0435	х 0445	ѕ 0455	ѥ 0465	ѵ 0475	◌҅ 0485	ҕ 0495	ӈ 04A5	ҵ 04B5	Ҋ 04C5	ӕ 04D5	ӥ 04E5	ӵ 04F5
6	І 0406	Ж 0416	Ц 0426	ж 0436	ц 0446	і 0456	Ѧ 0466	Ѷ 0476	◌҆ 0486	Җ 0496	Ӆ 04A6	Ҷ 04B6	ҋ 04C6	Ӗ 04D6	Ӧ 04E6	
7	Ї 0407	З 0417	Ч 0427	з 0437	ч 0447	ї 0457	ѧ 0467	ѷ 0477		җ 0497	ӆ 04A7	ҷ 04B7	Ҍ 04C7	ӗ 04D7	ӧ 04E7	
8	Ј 0408	И 0418	Ш 0428	и 0438	ш 0448	ј 0458	Ѩ 0468	Ѹ 0478	◌҈ 0488	Ҙ 0498	Ҏ 04A8	Ҹ 04B8	Ҍ 04C8	Ә 04D8	Ө 04E8	Ӧ 04F8
9	Ѣ 0409	Й 0419	Щ 0429	й 0439	щ 0449	ѣ 0459	ѩ 0469	ѹ 0479	◌҉ 0489	ҙ 0499	ҏ 04A9	ҹ 04B9	ҍ 04C9	ә 04D9	ө 04E9	ӧ 04F9
A	Ҥ 040A	К 041A	Ъ 042A	к 043A	ъ 044A	ҥ 045A	Ѫ 046A	Ѻ 047A	Ӣ 048A	Ҝ 049A	Ҫ 04AA	Ҫ 04BA	ҥ 04CA	Ӓ 04DA	Ӧ 04EA	
B	Ҧ 040B	Л 041B	Ы 042B	л 043B	ы 044B	ҧ 045B	ѫ 046B	ѻ 047B	ӣ 048B	ҝ 049B	ҫ 04AB	ҫ 04BB	Ҧ 04CB	ӓ 04DB	ӧ 04EB	
C	Ҡ 040C	М 041C	Ь 042C	м 043C	ь 044C	ҡ 045C	Ѭ 046C	Ѽ 047C	Ҍ 048C	Ҟ 049C	Ҭ 04AC	Ҭ 04BC	ҧ 04CC	Ӝ 04DC	Ӝ 04EC	
D	Ѝ 040D	Н 041D	Э 042D	н 043D	э 044D	ѝ 045D	ѭ 046D	ѽ 047D	ҍ 048D	ҟ 049D	ҭ 04AD	ҭ 04BD	Ҩ 04CD	ӝ 04DD	ӝ 04ED	
E	Ў 040E	О 041E	Ю 042E	о 043E	ю 044E	ў 045E	Ѯ 046E	Ѿ 047E	Ҏ 048E	Ҡ 049E	Ҩ 04AE	Ҩ 04BE	ҩ 04CE	Ӟ 04DE	Ӯ 04EE	
F	Ѯ 040F	П 041F	Я 042F	п 043F	я 044F	ѯ 045F	ѯ 046F	ѿ 047F	ҏ 048F	ҡ 049F	ҩ 04AF	ҩ 04BF		ӟ 04DF	ӯ 04EF	

Рис. 3.11. Посмотрите на диапазон кириллицы в Unicode. Здесь можно найти не только буквы дореволюционного алфавита, но и старославянскую азбуку!

В Unicode пять латинских лигатур с буквой «f»:

U+FB00	U+FB01	U+FB02	U+FB03	U+FB04
ff	fi	fl	ffi	ffl

В составе слов они выглядят так:

Без лигатуры	**С лигатурой**
finance	finance
fly	fly
graffiti	graffiti

Лигатуры улучшают внешний вид текста, но при компьютерной обработке текста с ними происходит примерно то же, что с буквой «Ё» в русском языке: программы сортировки, поиска и другой автоматической обработки не распознают лигатуры или распознают их неправильно.

Обратите внимание, что символы, имеющие одинаковую графическую форму, но принадлежащие разным языкам, находятся в Unicode на разных позициях. Латинская «A» (U+0041) и русская «А» (U+0410) — два разных символа. Аналогично, латинская «i» (U+0069) по своему положению в таблице отличается от кириллической «і» (U+0456 — используется, например, в украинском языке).

Дополнительно

Есть и более тонкие различия. Например, буква «ℵ» («алеф»), употребляющаяся в тексте на иврите, находится в диапазоне иврита и имеет код U+05D0. А эта же буква, употребляющаяся как обозначение в математических текстах, находится в диапазоне буквоподобных символов, и ее код — U+2135. Буква «Å», присутствующая в некоторых европейских языках, находится в диапазоне расширенной латиницы и имеет код U+00C5. А если она используется для обозначения ангстрема (единицы длины в физике) — она берется из диапазона буквоподобных символов с кодом U+212B.

Некоторые диапазоны Unicode предназначены для частного использования (private use). Это значит, что стандарт не определяет никаких символов в этих диапазонах. Разработчики текстов, шрифтов и программного обеспечения могут размещать в них символы по своему усмотрению. (Разумеется, один и тот же диапазон может использоваться разными разработчиками по-разному, как это происходило раньше с расширенными кодировками.)

Примечание

Наиболее свежая редакция стандарта Unicode на момент написания этой книги — Unicode 4.0.1, выпущенная в марте 2004. Этот стандарт можно загрузить по адресу: **www.unicode.org/versions/Unicode4.0.1**.

В Unicode 4.0 определены шестнадцатеричные коды символов от U+0000 до U+10FFFF — всего 1 114 112 кодов. Из них 96 248 отведены под символы, 137 468 предназначены для частного использования, 2313 заняты под служебные нужды (например, среди них — управляющие символы), и целых 878 083 кода пока еще свободны и будут заполняться новыми символами в будущих версиях Unicode.

Дополнительно

Среди алфавитов, которые рассматриваются ISO и разработчиками Unicode на предмет их включения в новые стандарты, — египетские иероглифы, славянская глаголица, а также тенгвар и кирт из произведений Толкина.

UTF-8 и другие UTF

Unicode сам по себе не является кодировкой. Это просто набор символов, каждому из которых присвоен уникальный номер и название. Чтобы практически закодировать в памяти компьютера каждый символ, применяются специальные основанные на Unicode кодировки, называющиеся *форматами преобразований Unicode* (Unicode Transformation Formats, сокращенно UTF).

Существует несколько таких форматов, но для разработки текстового содержания сайтов наиболее важен один из них — UTF-8.

Принцип кодирования в UTF-8 таков: каждому символу нужно поставить в соответствие один или несколько последовательных 8-битных кодов. (Так как в современной вычислительной технике 1 байт состоит из 8 битов, можно сказать, что UTF-8 преобразует каждый символ в последовательность из одного или нескольких байтов.)

Дополнительно

Алгоритм перевода кода символа в UTF-8 представлен в табл. 3.16.

Таблица 3.16. *Преобразование в UTF-8*

Диапазон кодов	Вид двоичного кода в этом диапазоне	Байты преобразованного кода в UTF-8			
		1-й	2-й	3-й	4-й
U+0000...U+007F	00000000 0xxxxxxx	0xxxxxxx	–	–	–
U+0080...U+07FF	00000yyy yyxxxxxx	110yyyyy	10xxxxxx	–	–
U+0800...U+FFFF	zzzzyyyy yyxxxxxx	1110zzzz	10yyyyyy	10xxxxxx	–
U+10000..U+10FFFF	000uuuuu zzzzyyyy yyxxxxxx	11110uuu	10uuzzzz	10yyyyyy	10xxxxxx

Посмотрим, например, как в UTF-8 будет закодирована прописная русская буква «А». Ее код в Unicode — U+0410, или в двоичной системе — 00000100 00010000. Этот код входит в диапазон из второй строки таблицы:

0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0

0 0 0 0 0 у у у у у х х х х х х

Биты, соответствующие буквам «у», поместим в первый байт результирующего кода, а биты, соответствующие «х» — во второй байт:

1 1 0 1 0 0 0 0 1 0 0 1 0 0 0 0

1 1 0 у у у у у 1 0 х х х х х х

Получился код 11010000 10010000, или в шестнадцатеричной системе — D0 90.

Первые 128 символов Unicode в кодировке UTF-8 кодируются одним байтом и выглядят точно так же, как в стандартной ASCII. Это значит, что любой текст в ASCII полностью совместим с UTF-8.

Обычные буквы многих алфавитов (в частности, русского, греческого, иврита, арабского), а также латинские буквы с диакритическими знаками кодируются двумя байтами. Для большинства типографских символов (тире, английские кавычки, многоточие и др.) требуется три байта. И только особенно экзотические символы (вроде музыкальных знаков) будут закодированы четырьмя байтами.

Кодировка UTF-8, как и другие кодировки Unicode, обеспечивает однозначное представление каждого символа. Это значит, что каждому коду символа соответствует уникальная последовательность байтов, которую можно единственным образом перевести обратно в первоначальный код.

Внимание!

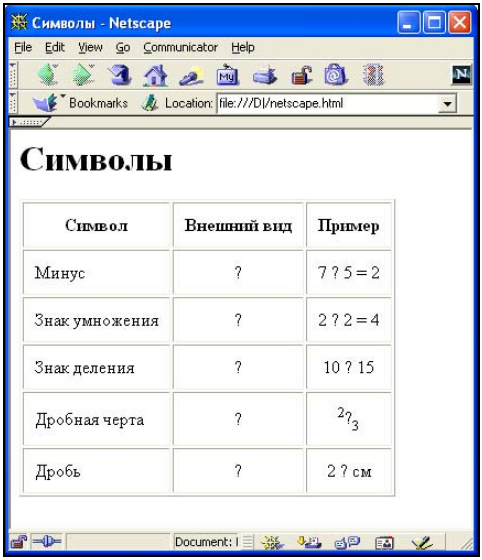
Кодировка UTF-8 позволяет использовать в тексте веб-страницы любые символы из всего набора UCS, не задействуя специальных средств вроде подстановок HTML (*см. главу 5*) или особых шрифтов.

Дополнительно

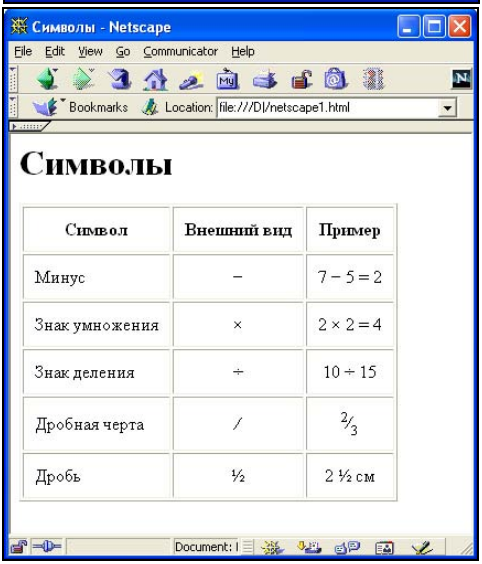
В стандарте Unicode определены еще шесть кодировок: UTF-16, UTF-16BE, UTF-16LE, UTF-32, UTF-32BE, UTF-32LE. Первые три из этих кодировок представляют каждый символ в виде одной или двух цепочек по 16 битов. При сохранении такой цепочки в компьютере или ее передаче по линии связи в кодировке UTF-16BE сначала передается первый (т. н. старший) из двух байтов, а потом второй (младший). В UTF-16LE — наоборот: сначала передается младший байт, потом старший. («BE» расшифровывается как «Big Endian», «LE» — как «Little Endian».) В UTF-16 (без обозначений «BE» или «LE») последовательность байтов предваряется специальной сигнальной цепочкой из двух байтов, по которой обрабатывающая программа судит о порядке байтов в последующем коде.

Аналогично обстоят дела с кодировками UTF-32, UTF-32BE, UTF-32LE, за исключением того, что в них каждый символ представляется в виде одного 32-разрядного блока.

Иногда использование UTF-8 решает некоторые проблемы с совместимостью веб-страниц (рис. 3.12).



а



б

Рис. 3.12. Браузер Netscape 4 не отображает некоторые специальные символы (*а*), если страница создана в одной из расширенных кодировок ASCII. А если преобразовать страницу в UTF-8 — проблемы с этими символами исчезают (*б*)

Трудности с использованием Unicode на практике

Unicode — более молодая технология, чем ASCII, поэтому многие проблемы, связанные с этим набором символов, еще не до конца решены. Нет существенных препятствий для разработки веб-страниц в UTF-8, однако разработчику следует учитывать некоторые возможные трудности:

- операционная система и браузер пользователя должны поддерживать UTF-8. За это можно не беспокоиться: все современные системы хорошо работают с Unicode. Могут возникнуть проблемы с экзотическими символами вроде музыкальных знаков или недавно введенных в стандарт пиктограмм наподобие «⚠». Но за обычные символы, из которых состоит текст, можно не волноваться: программное обеспечение пользователя поймет их правильно;
- на компьютере пользователя должны быть установлены шрифты с поддержкой нужных символов. Недостаточно того, что браузер пользователя понял, какой символ ему встретился; он еще должен иметь возможность отобразить этот символ.

  Ни один из современных шрифтов не включает в себя всех символов Unicode. Даже в самом полном шрифте Arial Unicode MS — немногим более 50 000 символов. (Вспомним, что в стандарте Unicode их почти 100 000!) При этом Arial Unicode MS достаточно редко используется на веб-страницах. В наиболее распространенных шрифтах (для Windows это, например, Times New Roman, Courier New и Arial) символов еще меньше — в среднем 1000–1500.

  Однако в большинстве случаев и это не составляет проблемы. Все символы, необходимые для набора обычного текста на русском и английском языках, входят в любой из распространенных шрифтов. Это касается и символов с диакритическими знаками (например, «Ä» или «Š»), типографских знаков (например, тире или кавычек) и большинства общеупотребительных специальных символов (например, «€», «№» или «©»). Трудности возникнут только в особых случаях: например, если вы хотите набирать текст на старославянском языке или использовать абсолютно все латинские лигатуры. Некоторые рекомендации по разрешению этих трудностей можно найти в *главе 5*;
- редакторы и другие программы разработки не всегда хорошо поддерживают Unicode. Многие программы, которыми разработчики пользуются для кодирования веб-страниц, не поддерживают UTF-8. Кроме того, иногда разработчик испытывает неудобства из-за того, что многие символы Unicode невозможно ввести непосредственно с клавиатуры, как обычные буквы. Для их ввода приходится пользоваться вспомогательными средствами вроде специальных сочетаний клавиш или диалоговых окон.

Примечание

Старайтесь просматривать страницы, которые вы разрабатываете, под разными операционными системами и браузерами. Внимательно следите за тем, не возникает ли где-нибудь проблем с кодировками и отображением символов.

Каждая кодировка имеет свое официальное обозначение. Именно оно должно использоваться в коде HTML, в заголовках протокола HTTP и других технических ситуациях. Приведем обозначения нескольких важных кодировок (табл. 3.17).

Таблица 3.17. *Официальные обозначения некоторых кодировок*

Официальное обозначение кодировки	**Распространенный вариант обозначения**	**Количество символов**	**Нормативный документ, определяющий кодировку**	**Примечания**
us-ascii	ASCII, US ASCII	128	RFC 1345, **www.ietf.org/rfc/rfc1345.txt**	Стандартная ASCII
iso-8859-1	ISO, Latin-1	256	RFC 1345, **www.ietf.org/rfc/rfc1345.txt**	Расширенная кодировка ASCII с поддержкой специальных символов из алфавитов на латинской основе. Не поддерживает кириллицу
cp866	CP 866, DOS	256		Расширенная кодировка ASCII с поддержкой кириллицы. Основная кодировка в DOS и OS/2
windows-1251	WIN, windows, 1251	256		Расширенная кодировка ASCII с поддержкой кириллицы. Стандартная кодировка кириллицы в Windows
koi8-r	KOI, koi-8, KOI-8 R	256	RFC 1489, **www.ietf.org/rfc/rfc1489.txt**	Расширенная кодировка ASCII с поддержкой кириллицы. Стандартная кодировка кириллицы в UNIX

Таблица 3.17 *(окончание)*

Официальное обозначение кодировки	Распространенный вариант обозначения	Количество символов	Нормативный документ, определяющий кодировку	Примечания
x-mac-cyrillic	Mac, MacCyr	256		Расширенная кодировка ASCII с поддержкой кириллицы. Основная кодировка кириллицы в некоторых версиях Mac OS. Обозначение не зарегистрировано в реестре IANA, но эта кодировка фактически распознается большинством браузеров
utf-8	UTF, UTF 8	96 248, может быть расширено		Одна из кодировок Unicode

Полный список официальных обозначений кодировок можно найти в Интернете по адресу: **www.iana.org/assignments/character-sets**.

Примечание

Вы наверняка обратили внимание, что автор придает весьма большое значение нормативным документам, стандартам и спецификациям. В самом деле, для профессиональной деятельности в любой области крайне важно знать принятые в ней официальные стандарты. Одного знания стандартов, конечно, совершенно недостаточно для успешной работы, но без знакомства с ними вообще ни о каком профессионализме не может идти речь.

Чтобы писать текст на русском языке, нужно как минимум знать правила русской грамматики и типографики. Чтобы разрабатывать страницы на языке HTML, нужно как минимум разбираться в официальных спецификациях различных версий этого языка. Чтобы правильно выбрать кодировку для разрабатываемого текста, нужно ориентироваться в нормативных документах, описывающих эти кодировки.

Любые книги, учебники и справочники (в том числе и эта книга) могут использоваться только в качестве вспомогательных пособий. Они помогут получить общее представление о предметной области, донесут до читателя основные принципы и понятия — но нисколько не заменят стандартов и нормативов. Переходя от обучения к профессиональной деятельности, крайне необходимо освоить эти стандарты.

Именно несоблюдение стандартов привело к гибели подводной лодки «К-159» в августе 2003 года. Несоблюдение стандартов в области разработки сайтов, по счастью, редко угрожает здоровью и жизни людей, но это не повод относиться к этим стандартам пренебрежительно. От нарушения стандартов может существенно ухудшиться функциональность, удобство и доступность сайта, сайт получится неэффективным и не будет привлекать пользователей.

Автор надеется, что, прочитав эту книгу, вы также найдете возможность прочесть и все упомянутые в ней нормативные документы.

3.2. Лики текста

Форматы текстовых файлов

Выпуск изданий в нестандартном формате допускается только по разрешению Комитета по печати при Совете Министров СССР.

П. Г. Гиленсон. Справочник технического редактора

...Сегодня прослеживается положительная тенденция: если раньше формат радио зачастую выбирался в соответствии с личными предпочтениями владельца станции, то сейчас станции стараются оперировать данными исследования своей целевой аудитории.

Л. Ксенз. Помехи в эфире («Деловая столица», № 43, 2002)

3.2.1. Двоичный и текстовый режимы

54 6F 20 62 65 2C 20 6F 72 20 6E 6F 74 20 74 6F 20 62 65 3A 20 74 68 61 74 20 69 73 20 74 68 65 20 71 75 65 73 74 69 6F 6E 2E 2E 2E 0D 0A 57 69 6C 6C 69 61 6D 20 53 68 61 6B 65 73 70 65 61 72 65 2E 20 54 68 65 20 54 72 61 67 65 64 79 20 6F 66 20 48 61 6D 6C 65 74 2C 20 50 72 69 6E 63 65 20 6F 66 20 44 65 6E 6D 61 72 6B 0D 0A

Внимание!

Двоичный режим: файл — последовательность байтов.

Текстовый режим: файл — последовательность символов.

Различные редакторы, программы просмотра и другие средства работы с файлами могут выполнять операции над файлами либо в двоичном, либо в текстовом режиме.

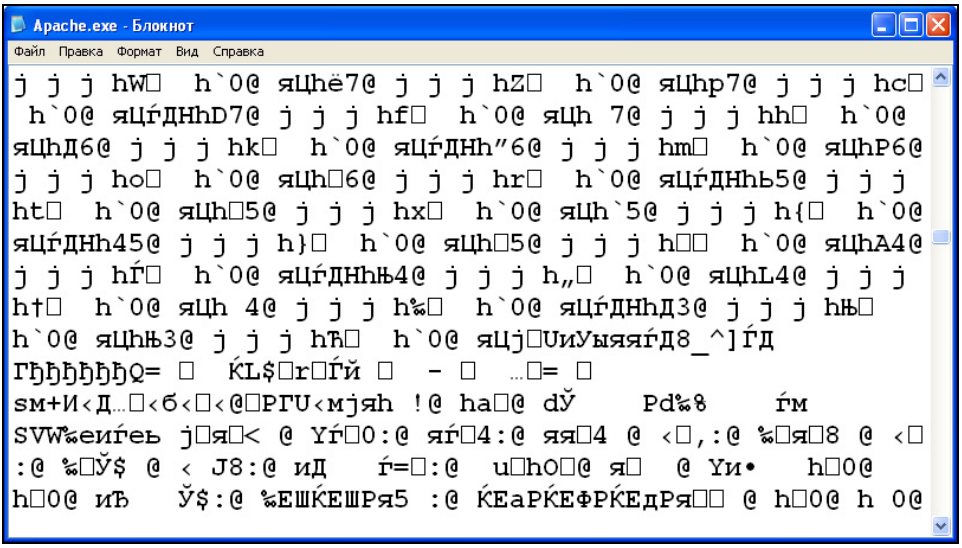
В *двоичном режиме* (который еще называют *бинарным*) файл рассматривается как *последовательность байтов* — т. е. чисел, представленных в двоичной системе счисления. Такой режим характерен для исполняемых файлов (например, в среде Windows к ним относятся файлы с расширениями exe, com и dll), изображений, звуковых файлов, а также архивов.

Если вы откроете такой файл в текстовом редакторе, то, скорее всего, увидите тарабарщину, разобрать которую невозможно (рис. 3.13, *а*). Такие файлы и не предназначены для того, чтобы их открывали в текстовом редакторе. Если действительно требуется внести изменения непосредственно в исполняемый (или звуковой) файл, специалисты пользуются специальным шестнадцатеричным редактором, представляющим каждый байт числа в виде пары шестнадцатеричных цифр (рис. 3.13, *б*).

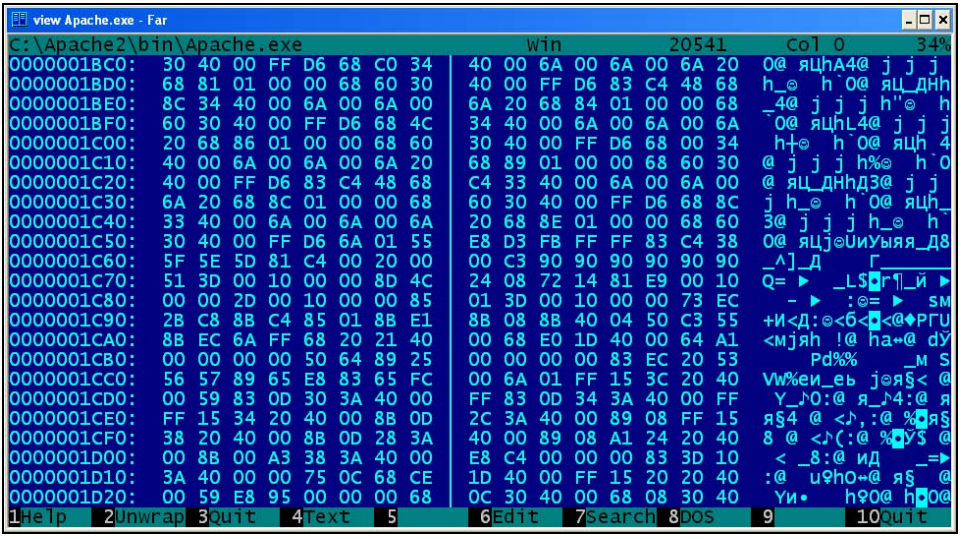
Впрочем, файлы, предназначенные для работы в двоичном режиме, обычно редко подвергаются вмешательству человека. Эти файлы, как правило, автоматически генерируются компьютером. Например, исполняемый файл программы получается в результате компиляции исходного кода, написанного программистом. Если этот исполняемый файл работает не так, как надо, программист не вносит изменения непосредственно в него, а редактирует исходный код и компилирует программу заново.

В *текстовом режиме* файл воспринимается как последовательность символов, представленных в какой-либо кодировке. Большинство текстовых редакторов работают именно в этом режиме. Получив запрос на открытие файла, они выводят на экран не двоичный код (который на самом деле содержится в файле), а символы, соответствующие этому коду. Если редактор при этом использует ту же кодировку, в которой файл был создан, — мы обычно видим на экране осмысленный текст.

Файлы, для работы с которыми требуется текстовый режим, обычно создаются пользователем непосредственно. Это, например, простые текстовые файлы, которые мы набираем в Блокноте или Vim, исходные коды программ, документы HTML и таблицы стилей CSS.



а



б

Рис. 3.13. Исполняемый файл, открытый в текстовом редакторе (*а*), выглядит пугающе. А в специальном шестнадцатеричном редакторе (*б*) он превращается в набор чисел, в которых специалист иногда может разобраться

Дополнительно

Когда-то было распространено такое развлечение программистов, работавших на низком аппаратном уровне: они составляли действующие исполняемые файлы, которые, тем не менее, при просмотре в текстовом режиме выглядели

как текст на английском языке. Эти тексты обычно получались малоосмысленными и неграмотными с точки зрения правописания, но они все же не были такой тарабарщиной, как на рис. 3.13, *а*.

Сам по себе файл не содержит никакой информации о том, «двоичный» он или «текстовый». Любой файл можно открыть и в двоичном, и в текстовом режиме. Просто некоторые файлы предназначены скорее для одного режима, а некоторые — для другого. Мы можем открыть текстовый файл в шестнадцатеричном редакторе, но вряд ли нам доставит удовольствие видеть коды символов вместо самих этих символов (как в эпиграфе к этому разделу). Наоборот, можно «обмануть» операционную систему и запустить текстовый файл на выполнение: например, в детстве автор любил присваивать текстовым файлам имена с расширением exe и смотреть, что происходит с компьютером при попытке запустить такой файл. (Надо сказать, что ничего хорошего с компьютером не происходило.)

Дополнительно

Разница между текстовым и двоичным режимами может стать критичной, например, при передаче файлов по протоколу FTP. Если файлы передаются в текстовом режиме (text mode), то символы перевода строки автоматически конвертируются в формат системы-получателя. (Скажем, если мы передаем файлы из Windows на UNIX, то все CR LF заменяются на LF.) Однако, если передавать в таком режиме исполняемый файл или архив, он может серьезно пострадать — ведь в этом случае символ с кодом 0D вовсе не обозначает возврат каретки, а может оказаться важной частью исполняемой команды или данных. Поэтому такие файлы передаются в двоичном режиме (binary mode), при котором никаких преобразований с байтами не происходит.

Иногда говорят не о двоичном и текстовом режиме, а о *двоичных и текстовых файлах*. Файлы делят на основании того, какой режим наиболее характерен для работы с ними. Нередко это приводит к путанице: например, в одних источниках файлы DOC (создаваемые в Microsoft Word) и PDF считаются двоичными (так как их нельзя редактировать непосредственно без специальных программ), а в других — текстовыми (так как содержат текст, предназначенный для чтения человеком). Автор предпочитает не пользоваться терминами «двоичные файлы» и «текстовые файлы» в этом смысле. В этой книге под текстовым файлом будет пониматься несколько иное.

Внимание!

Текстовый файл — это файл, основным содержанием которого является текст, предназначенный для чтения и (или) редактирования пользователем.

Примечание

Не забывайте, что такое определение действительно в пределах этой книги, но не обязательно распространяется на другую литературу и документацию.

Далее будут рассмотрены несколько распространенных форматов текстовых файлов — в основном с позиций их практического использования в Интернете.

Все текстовые файлы можно разделить на две группы: обычный текст и форматированный текст. *Обычный текст* — это просто последовательность символов, без всякого оформления и верстки; вы можете редактировать обычный текст в такой простой программе, как Блокнот под Windows. *Форматированный текст* — это текст с оформлением, от простого формата RTF до сложных форматов издательских систем вроде QuarkXPress или Corel Ventura. В табл. 3.18 перечислены основные различия между обычным и форматированным текстом. Более подробно об этих различиях рассказано далее в этой главе.

Таблица 3.18. *Обычный и форматированный текст*

Обычный текст	Форматированный текст
Не содержит информации об оформлении; внешний вид текста целиком зависит от настроек программы просмотра или редактора	Содержит информацию об оформлении; в некоторых форматах (например, PDF) внешний вид полностью определяется разработчиком документа и одинаков во всех средствах просмотра
Не содержит нетекстовых объектов — например, иллюстраций, элементов интерактивного взаимодействия (гиперссылок, элементов управления), анимационных роликов, встроенных приложений (например, апплетов) и т. д.	Может содержать нетекстовые объекты — в частности, практически все современные виды форматированного текста, включая DOC, PDF и HTML, поддерживают гиперссылки
Набор символов ограничен выбранной кодировкой и шрифтами, установленными там, где текст просматривается и редактируется	Большинство форматов позволяют включать в текст нестандартные символы, математические формулы и другие текстовые объекты, даже если они не входят ни в одну кодовую таблицу
Текст легко переносится с одного компьютера на другой, может открываться и редактироваться в различных текстовых редакторах, процессорах и издательских системах	Многие форматы просматриваются и редактируются только с помощью специальных программ; могут возникать проблемы при переносе файлов с одного компьютера на другой
Текст легко преобразуется в любой другой текстовый формат	Текст, как правило, плохо преобразуется из одних видов форматированного текста в другие (например, из DOC в HTML) и в обычный текст

Таблица 3.18 *(окончание)*

Обычный текст	Форматированный текст
Используется для простых текстов, где оформление не имеет значения — например, для сообщений электронной почты, записей на форуме	Используется для текстов, где важно оформление и верстка — в т. ч. для текстов, предназначенных для вывода на печать
Файлы обычно занимают маленький объем, быстро передаются по сети	Файлы могут занимать большой объем, не умещаться на некоторые устройства хранения информации (вроде дискет), долго передаваться по сети

3.2.2. Обычный текст

> Тогда речи и чувства текут двумя параллельными потоками, один над другим, и верхний поток, где струятся слова, — это лишь знаки, лишь символы потока глубинного, где мечутся смутные образы.
>
> *А. Моруа. Искусство беседы*

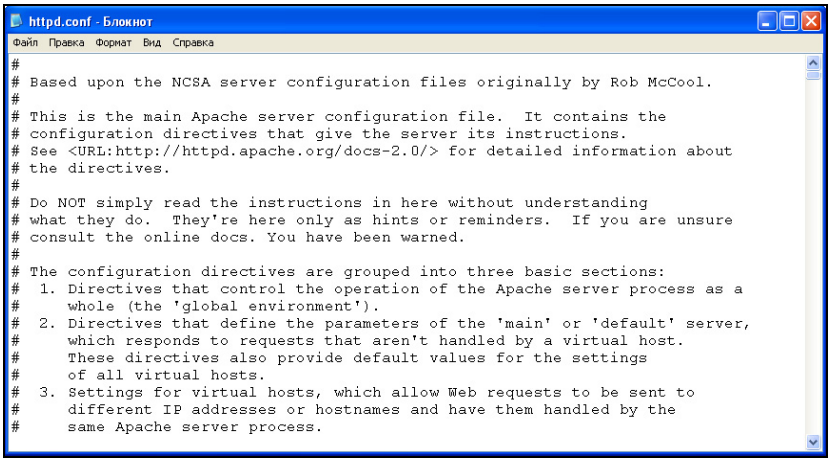
Обычный текст (plain text) — самый простой из текстовых форматов. Он представляет собой последовательность символов, предназначенную для непосредственного чтения человеком. Этот текст может быть открыт и прочитан в любом текстовом редакторе. Чаще всего файлы в формате «обычный текст» имеют расширение txt.

Примечание

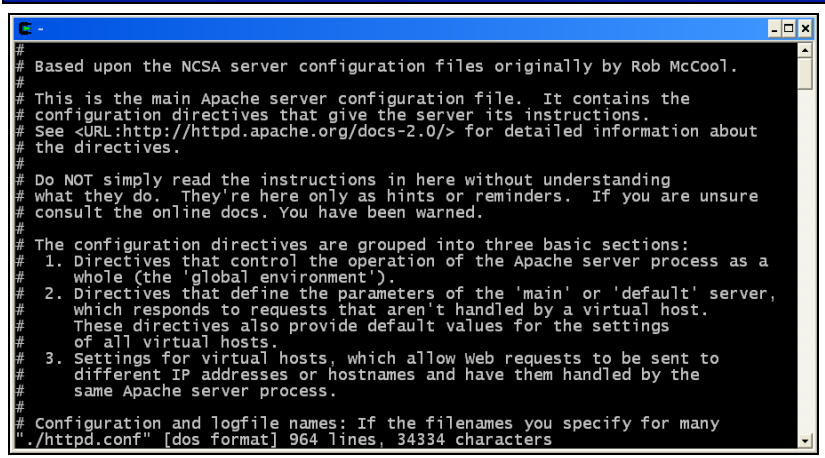
Все современные операционные системы имеют стандартные средства просмотра и редактирования файлов в формате «обычный текст». В Windows таким средством является встроенная программа Блокнот.

Кроме того, все текстовые процессоры (такие, как Microsoft Word или OpenOffice) и издательские системы (например, QuarkXPress) могут открывать обычные текстовые файлы и преобразовывать их в свой формат. Все браузеры также умеют открывать обычный текст для просмотра.

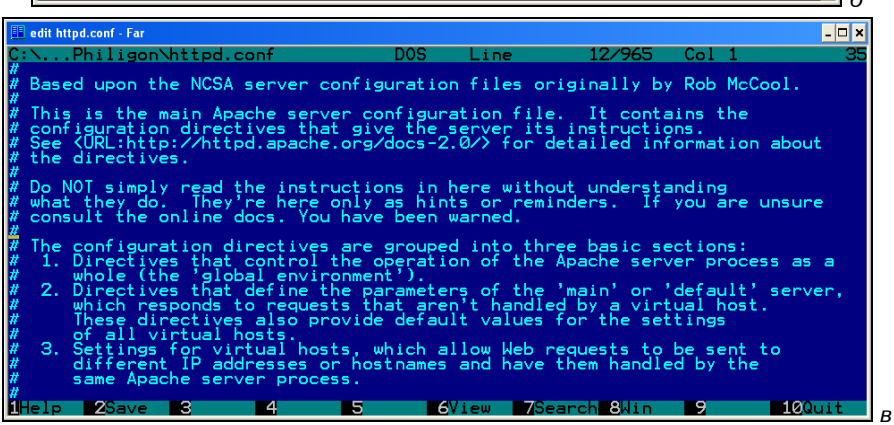
Обычный текст не содержит никакой информации о своем оформлении. В этом формате нельзя указать, что текст должен быть курсивным или подчеркнутым, что он должен выравниваться по центру страницы или иметь размер шрифта 9 пунктов. Внешний вид текста полностью зависит от настроек той программы, в которой его открывают (рис. 3.14).



а



б



в

Рис. 3.14. Один и тот же файл в формате «обычный текст» отображается по-разному в зависимости от настроек редактора: *а* — Блокнот, *б* — vi, *в* — встроенный редактор файлового менеджера FAR Manager

Конечно, редактор или программа просмотра должны уметь правильно интерпретировать кодировку документа. Кроме того, в операционной системе должны присутствовать шрифты, позволяющие отображать текст в этой кодировке.

Обычный текст — наиболее простой, универсальный и экономичный формат. Файлы в этом формате можно легко переносить с одного компьютера на другой, не опасаясь, что там не окажется нужной программы для их чтения и редактирования.

В Интернете формат «обычный текст» особенно широко применяется:

- в сообщениях, отправляемых по электронной почте или в телеконференции;
- в сообщениях на форумах и в гостевых книгах сайтов.

Дополнительно

Многие администраторы локальных сетей и просто достаточно квалифицированные пользователи из соображений безопасности настраивают почтовые программы таким образом, чтобы сообщения отображались только в формате «обычный текст». Что же касается форумов и гостевых книг, то для записи в них обычно используется стандартное текстовое поле (элемент `textarea` в HTML), а в нем пользователь технически не может набрать ничего, кроме обычного текста.

Обычный текст очень легко скопировать из одного документа и вставить в другой, при этом не возникает никаких проблем с несовместимостью форматов.

Формат «обычный текст» прекрасно подходит для постоянного хранения текстовых материалов, так как из него проще всего преобразовывать текст в другие форматы. Кроме того, обычный текст занимает меньше места на диске, чем любой другой текстовый формат.

Примечание

Представьте себе, что вы пишете статью, которую собираетесь представить в издательство в формате DOC, опубликовать в какой-либо телеконференции, а также разместить на своем сайте в виде страницы HTML и в формате PDF — как версию для печати. Вы совершите ошибку, если станете сразу набирать эту статью в Microsoft Word. Ведь для корректного перевода во все остальные форматы вам все равно придется сначала преобразовать файл DOC в обычный текст, чтобы избавиться от лишнего оформления. Если вы не хотите делать двойную работу — сначала наберите и сохраните вашу статью в формате «обычный текст», а потом на его основе создавайте файлы в нужных форматах, добавляя каждый раз оформление теми средствами, которые характерны именно для этих форматов.

Внимание!

Храните тексты на компьютере в формате «обычный текст». Даже если вы пользуетесь текстом в другом формате (например, DOC или RTF), имейте на всякий случай его копию в формате «обычный текст».

Обычный текст имеет и свои недостатки. Во-первых, при сохранении текстовых файлов на русском языке чаще всего применяют какую-либо расширенную кодировку ASCII с поддержкой кириллицы — например, Windows-1251 или KOI8-R. При этом в тексте приходится использовать только такие символы, которые доступны в выбранной кодировке и не пострадают при перекодировке. Практически к этим символам относятся только знаки стандартной 7-битной ASCII и обычные буквы русского алфавита.

Если в текст необходимо включить отсутствующие в кодовой таблице символы, приходится пользоваться суррогатами — символами стандартной ASCII или их сочетаниями (табл. 3.19). Это те же знаки, которые употреблялись на пишущих машинках для имитации типографских символов.

Таблица 3.19. *Некоторые суррогаты для текстовых файлов на русском языке*

Символ		**Суррогат**		**Ошибочные суррогаты**
Название	**Внешний вид**	**Название**	**Внешний вид**	
Кавычки-«елочки» (французские, угловые)	« »	Двойные кавычки ASCII	" "	<< >>
Кавычки-«лапки»	„ “	Двойные кавычки ASCII	" "	,, '' (слева — две запятые подряд, справа — два апострофа ASCII подряд)
Английские кавычки	“ ”	Двойные кавычки ASCII	" "	`` '' (слева — два обратных апострофа ASCII подряд, справа — два апострофа ASCII подряд)
Многоточие	...	Три точки подряд	. . .	
Длинное тире, короткое тире, знак «минус»	— – –	Дефис	-	-- --- (два или три дефиса подряд)

Таблица 3.19 *(окончание)*

Символ		Суррогат		Ошибочные суррогаты
Название	**Внешний вид**	**Название**	**Внешний вид**	
Знак номера	№	Латинская прописная буква «N»	N	#
Знак авторского права	©	Латинская буква «C» в круглых скобках	(C) (c)	

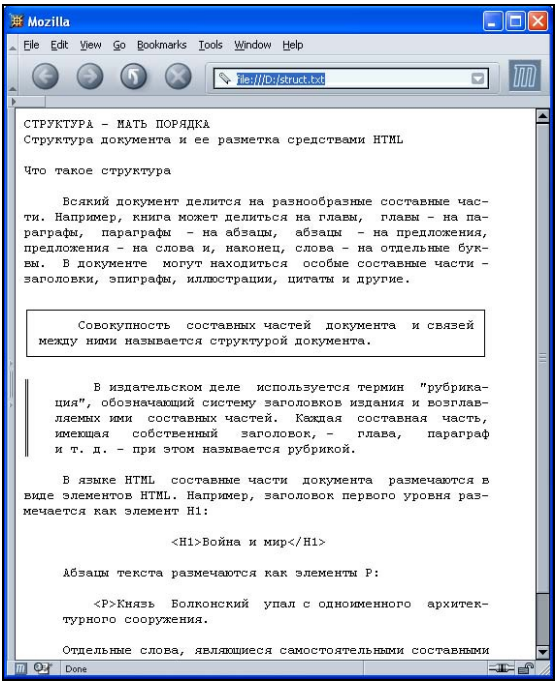
Дополнительно

Получается, что в формате «обычный текст» в расширенной кодировке ASCII нельзя пользоваться грамотной типографикой — нет ни кавычек-«елочек», ни тире, ни многоточия, ни знаков номера и параграфа... Можно набирать текст в какой-либо кодировке Unicode (например, в UTF-8), но это может вызвать проблемы с совместимостью: не на всех компьютерах найдутся текстовые редакторы, поддерживающие кодировки Unicode.

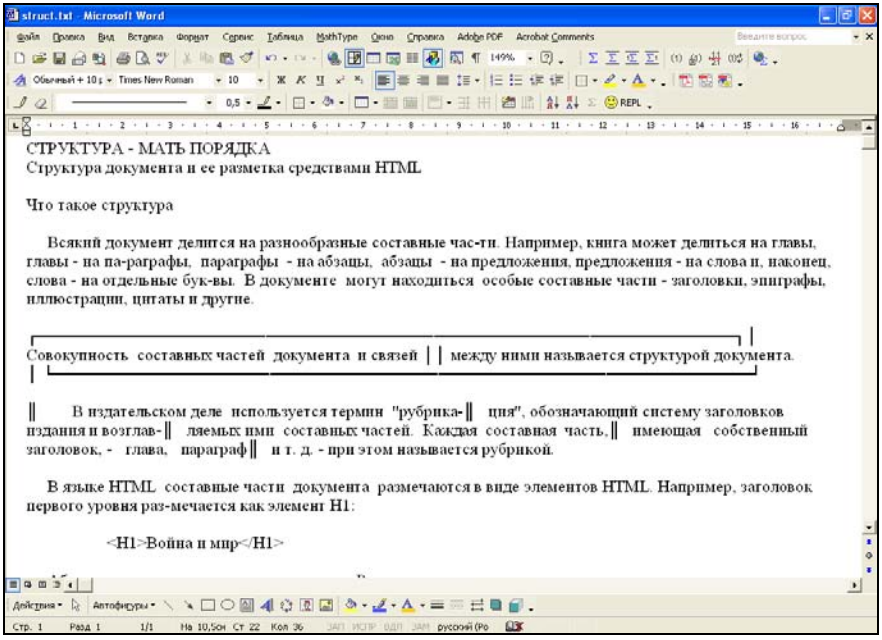
Второй недостаток формата «обычный текст» — это невозможность оформления. Для письма по электронной почте или сообщения на форум никакого особенного оформления не требуется, но для печатной публикации или веб-страницы обычный текст не подходит. В нем нельзя использовать шрифты разного начертания или разного размера, нельзя расположить текст в несколько колонок, нельзя добиться выравнивания. В обычный текст нельзя включить ни изображение, ни гиперссылку, не говоря уж об элементах управления или динамических объектах вроде анимации.

Дополнительно

Иногда встречаются документы в формате «обычный текст» (в основном, это сообщения в телеконференциях), авторы которых постарались все-таки применить к тексту все доступное оформление (рис. 3.15). В результате такой текст может хорошо смотреться в определенных условиях (если просматривается в нужной кодировке и моноширинным шрифтом), но если его захочется сохранить или перевести в другой формат, могут возникнуть существенные сложности. Сначала его придется очищать от имеющегося «псевдооформления» и превращать в настоящий «обычный текст».



а



б

Рис. 3.15. Текст неплохо смотрится в телеконференции: автор снабдил его переносами, выключкой по формату, рамками (*а*). Но если этот текст скопировать в окно редактора — все испортится (*б*)

Все же бывают случаи, когда обычный текст предпочтительнее форматированного — в основном, из-за своей компактности, переносимости и нетребовательности к программному обеспечению. Например, в популярной электронной библиотеке Максима Мошкова (**lib.ru**) литературные произведения представлены в виде обычного текста, поэтому их объем невелик, а загружаются они гораздо быстрее, чем форматированный текст.

Дополнительно

Множество технической документации в Интернете представлено в формате «обычный текст» — например, стандарты из серии RFC. Эта аббревиатура расшифровывается как «Request for Comments», что примерно переводится как «Запрос на комментарии». Несмотря на такое непритязательное название, многие RFC являются общепринятыми сетевыми стандартами. Среди них, например, RFC 2616 — описание протокола HTTP/1.1, по которому передаются все веб-страницы и большинство других ресурсов, доступных во Всемирной Паутине. (См. **www.ietf.org/rfc/rfc2616.txt**.) Некоторые кодировки символов — например, US-ASCII или KOI8-R — стандартизованы как RFC (см. табл. 3.16).

The World Wide Web Consortium (W3C) — организация, также разрабатывающая и поддерживающая многие стандарты (в частности, спецификации HTML и CSS) — предоставляет все свои документы в нескольких форматах, среди которых есть и «обычный текст» (plain text), а также PostScript и PDF.

Многие начинающие пользователи не знают о формате «обычный текст» и набирают все текстовые документы в серьезных текстовых процессорах (например, Microsoft Word) — даже в тех случаях, когда текст не содержит никакого оформления. При этом файл занимает больше места на диске, его можно прочитать только с помощью Word, его будет сложнее преобразовывать в другие форматы (например, чтобы вставить в электронное сообщение) — и все эти затруднения совершенно неоправданны. Грамотные пользователи набирают такие тексты в простом редакторе и сохраняют их в формате «обычный текст».

Файл в формате «обычный текст» можно открыть и редактировать в любом текстовом редакторе (например, Блокнот или Vim), текстовом процессоре (Microsoft Word, OpenOffice) или даже издательской системе (QuarkXPress, Corel Ventura). Если не позволять программе менять формат файла (например, Word, если за ним не уследить, склонен сохранять файлы в собственном формате DOC), то обычный текст останется обычным текстом, и его впоследствии можно будет без проблем открыть в этой же или любой другой программе.

3.2.3. Форматированный текст

> Не видишь ли ты, что сияющие красоты юности уменьшаются в своем совершенстве от чрезмерных, слишком изысканных украшений, не видал ли ты, как горские женщины, закутанные в безыскусственные и бедные одежды, приобретают большую красоту, чем те, которые украшены?
>
> *А. Гастев. Леонардо да Винчи*

В современных компьютерных системах применяется много видов форматированного текста. В отличие от формата «обычный текст», форматированный текст может содержать:

- информацию об оформлении (цвета, шрифты, их размеры, расположение элементов на странице и т. д.);
- информацию о структуре текста (заголовки, перечни, деление текста на логические блоки и т. д.);
- внедренные нетекстовые объекты (иллюстрации, звуковые файлы, динамические элементы — например, анимацию);
- элементы интерактивного взаимодействия (гиперссылки, элементы управления, встроенные программы — например, апплеты);
- параметры вывода на печать;
- метаинформацию (имя автора документа, язык и кодировку символов, сведения о программе, в которой был создан документ, и т. д.).

Рассмотрим несколько распространенных типов форматированного текста.

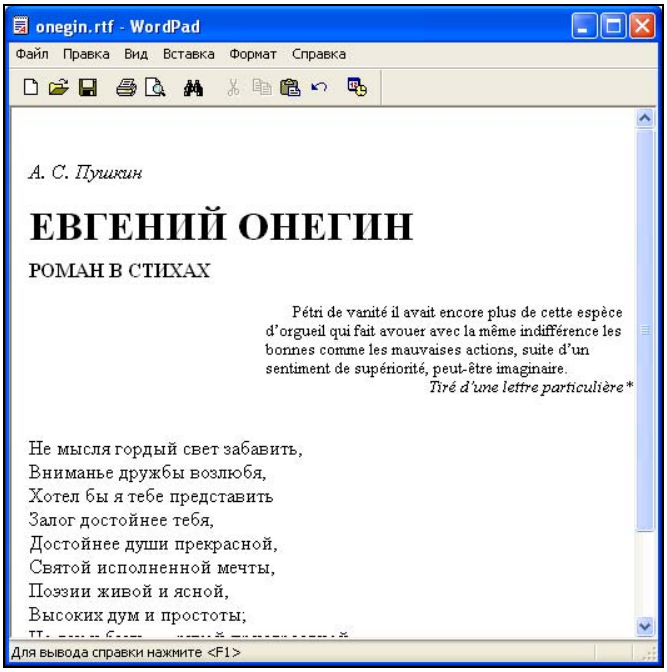
RTF (Rich Text Format)

Сейчас этот формат можно считать устаревшим, однако на его примере удобно показать некоторые особенности, характерные для форматированного текста в целом.

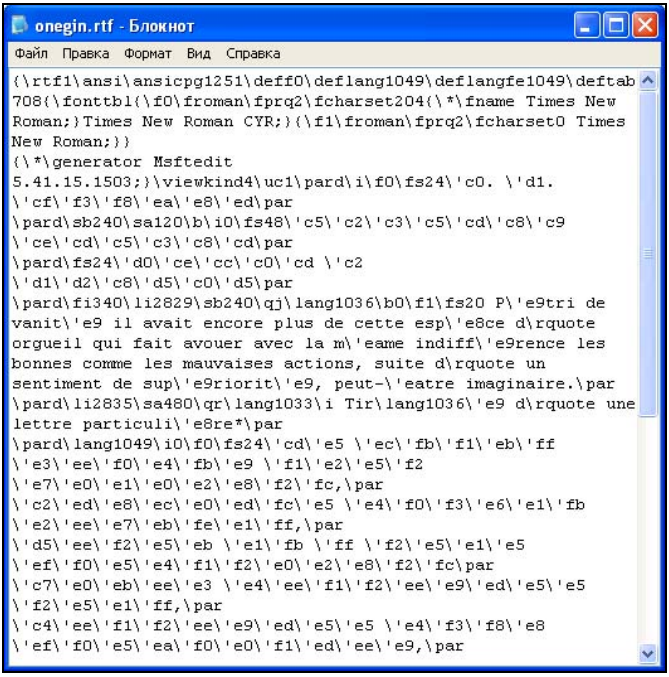
RTF (Rich Text Format, примерный перевод — «обогащенный текстовый формат») был разработан компанией Microsoft как средство создания переносимых документов с оформлением и форматированием, пригодных для просмотра на экране компьютера и для печати. Файлы в этом формате обычно имеют расширение rtf.

Примечание

Слово «переносимый» обычно означает, что документ или программу можно использовать на различных компьютерах с разными операционными системами



а



б

Рис. 3.16. Файл RTF: *а* — при просмотре в специальном редакторе WordPad; *б* — при просмотре в обычном текстовом редакторе

и разным программным обеспечением. Чем меньше файл связан с какими-то определенными операционными системами или программами, тем больше он переносим. Очевидно, среди текстовых файлов максимальной переносимостью обладают файлы в формате «обычный текст».

Если открыть файл в формате RTF в обычном текстовом редакторе (рис. 3.16, *б*), мы увидим набор непонятных знаков. Дело в том, что при сохранении в формате RTF текст определенным образом кодируется (например, слово «текст» превращается в строку «`\'f2\'e5\'ea\'f1\'f2`») и снабжается информацией об оформлении. (Например, если часть текста в документе выделена желтым цветом, то в начале документа присутствует запись «`\red255\green255\blue0`».)

Этот пример показывает следующие свойства, общие для всех видов форматированного текста:

- при сохранении форматированного текста и сам текст, и информация о его оформлении кодируются в соответствии с определенными принципами (эти принципы определяются конкретным форматом);
- для просмотра форматированного текста в «правильном» виде требуется специальная программа, поддерживающая этот формат.

Формат RTF достаточно универсален, его поддерживают многие текстовые процессоры. Он позволяет не только выделять текст различными цветами и применять другое шрифтовое оформление, но и внедрять изображения, размещать сноски и ссылки.

С другой стороны, этот формат достаточно прост. Он не годится для документов со сложным оформлением и форматированием, а также не обеспечивает типографского качества набора и печати текста. В последнее время RTF вытеснен более современными форматами.

Примечание

Спецификация RTF (версия 1.6) находится в Интернете по адресу: **msdn.microsoft.com/library/en-us/dnrtfspec/html/rtfspec.asp**.

DOC (Документ Microsoft Word)

Этот формат, пожалуй, является наиболее популярным в широких кругах пользователей. Набор, редактирование и печать текстовых документов у многих прочно ассоциируется с текстовым процессором Microsoft Word. Однако эта программа не удовлетворяет требованиям профессионального набора.

Дополнительно

Простой пример: попробуйте набрать в Microsoft Word текст, в котором выполнялись бы одновременно следующие требования:

- выравнивание по ширине (на профессиональном языке — «выключка по формату»);
- одинаковая ширина пробелов в каждой строке;
- переходы на новую строку только в разрешенных местах (см. п. 2.1.9).

Автору удавалось выполнить в Word только любые два из трех этих требований.

Файлы в формате DOC занимают достаточно много места по сравнению с другими текстовыми форматами и не очень хорошо переносятся с одного компьютера на другой. Нередки случаи, когда файл DOC, перемещенный на другой компьютер, открывался некорректно.

Тем не менее, приходится считаться с чрезвычайной распространенностью формата DOC всюду, где не требуется типографское и полиграфическое качество: внутрикорпоративный документооборот, рабочие материалы, тезисы докладов и т. д. В конце концов, эту книгу автор тоже набирает в Microsoft Word, потому что таковы требования издательства!

Иногда, если требуется разместить в Интернете крупный текстовый документ со сложным оформлением, автор подготавливает его в формате DOC и предоставляет для всеобщего доступа. Это не оптимальный вариант: например, формат PDF (см. далее) подходит для этой цели намного лучше.

Дополнительно

Формат DOC не всегда хорош и для отправки по электронной почте. Если вы собираетесь отправить текстовый документ, подумайте — может быть, его можно представить в каком-нибудь более экономном формате, вплоть до обычного текста. Кроме того, перед отправкой файл рекомендуется заархивировать. Фактическим стандартом для передачи по Интернету является формат архива ZIP, а он способен сжимать текстовые файлы в несколько раз. Если вы не хотите, чтобы ваш корреспондент был недоволен неоправданно большим объемом файла — постарайтесь уменьшить этот объем. Если получатель письма пользуется модемной связью, то большое письмо будет дольше загружаться, а если во время загрузки случится обрыв связи, то письмо придется запрашивать заново. Если же у получателя выделенная линия, то он, скорее всего, платит за входящий трафик, и будет вынужден раскошелиться на ваше «толстое» письмо.

И уж, конечно, не стоит преобразовывать документы Word в веб-страницы, воспользовавшись пунктом меню **Сохранить как...**. Результат такого преобразования не будет удовлетворять никаким требованиям грамотности, качества и доступности.

TEX

Формат TEX был разработан в конце 1970-х известным математиком и программистом Д. Кнутом. Изначально он предназначался для создания печатных публикаций, прежде всего математического характера. Кнут со всей ответственностью подошел к тому, чтобы публикации соответствовали действующим стандартам типографского набора и верстки. Пожалуй, TEX — это единственная система, не мешающая конечному пользователю создавать документы в строгом соответствии с типографскими правилами.

Впоследствии появилось много расширений TEX (самое известное из них — LATEX), в результате чего эта система стала применяться не только для научных публикаций, но и для подготовки широкого круга печатных изданий.

Документ в формате TEX можно набирать в любом текстовом редакторе, дополняя обычный текст управляющими конструкциями (рис. 3.17). Готовый исходный текст нужно пропустить через обрабатывающую программу — и на выходе получится полноценный документ, пригодный для профессиональной печати или преобразования в другие форматы (например, PostScript или PDF).

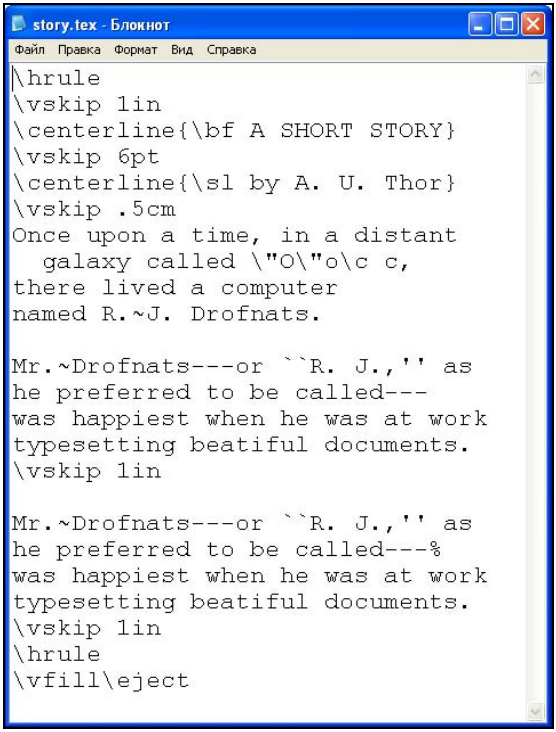


Рис. 3.17, *а*. Документ в формате TEX: набран в текстовом редакторе

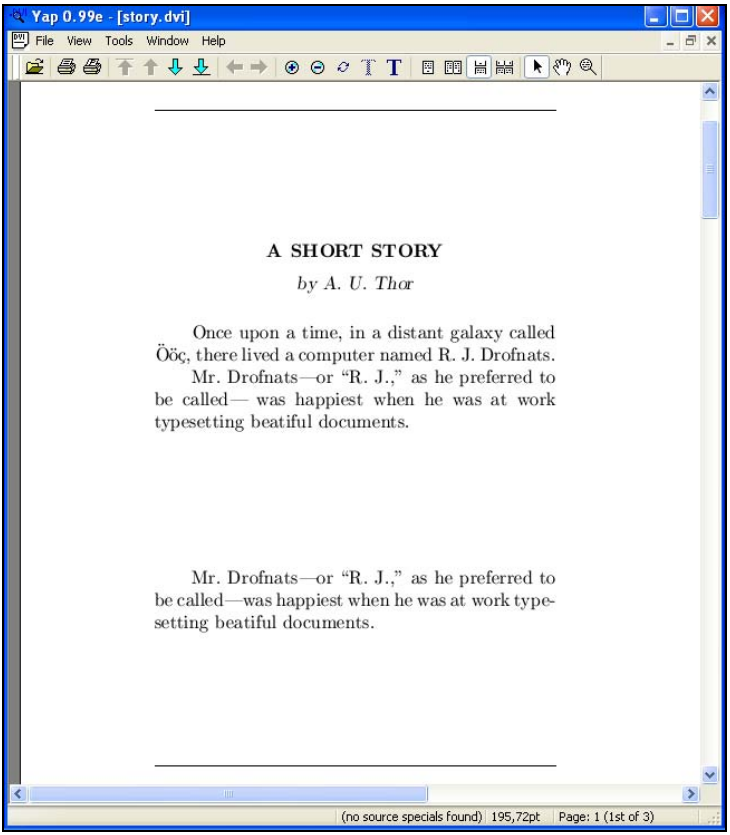


Рис. 3.17, *б*. Документ в формате TEX: с помощью системы TEX преобразован в файл DVI, готовый к выводу на печать

В отличие от форматов наподобие RTF и DOC, разработчик имеет полный контроль над каждым символом и каждой управляющей конструкцией в своем документе. В самом деле, работая в Microsoft Word, вы видите перед собой только готовую страницу и никак не управляете тем, что компьютер на самом деле записывает в файл. И уж тем более вряд ли кому-то придет в голову редактировать файл RTF или DOC в обычном текстовом редакторе (см. рис. 3.16, *б*). А в случае TEX вы непосредственно редактируете исходный файл, не позволяя компьютеру записать туда ни единого байта без вашего ведома.

纸 *Дополнительно*

Система TEX популярна и в России — в основном среди издателей научной (прежде всего, математической) литературы. К сожалению, многие издатели не

учитывают того, что TEX был разработан с учетом правил английской и американской типографики, которые отличаются от отечественных. Отсюда и английские кавычки (“ ”) в русском тексте, и более узкие, чем положено, междусловные пробелы, и употребление короткого тире вместо длинного. Существуют расширения TEX, позволяющие грамотно работать с русским языком, но далеко не все ими пользуются.

HTML (HyperText Markup Language)

Формат HTML также можно отнести к форматированному тексту. В самом деле, если взглянуть на исходный код файла HTML (рис. 3.18), можно убедиться, что он состоит из обычного текста и специальных управляющих конструкций (в частности, тэгов и подстановок символов), которые управляют отображением этого текста и его обработкой.

HTML похож на RTF и DOC тем, что для просмотра файлов в этом формате требуется специальная программа — браузер. (К браузерам, например, относятся Microsoft Internet Explorer, Mozilla и Opera). С другой стороны, HTML похож и на TEX — тем, что разработчик имеет возможность набирать и редактировать исходный код непосредственно в текстовом редакторе, контролируя каждый символ.

Рис. 3.18, *а*. Документ HTML: при просмотре в браузере

www.w3[1] - Блокнот

Файл Правка Формат Вид Справка

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="en-US"
xml:lang="en-US">
<head profile="http://www.w3.org/2000/08/w3c-synd/#">
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<meta name="generator"
content="HTML Tidy for Mac OS, see www.w3.org" />
<meta name="keywords"
content="W3C, World Wide Web, Web, WWW, Consortium, computer, access, accessibility, semantic, worldwide, W3, HTML, XML,
standard, language, technology, link, CSS, RDF, XSL, Berners-Lee, Berners, Lee, style sheet, cascading, schema, XHTML,
mobile, SVG, PNG, PICS, DOM, SMIL, MathML, markup, Amaya, Jigsaw, free, open source, software" />
<meta name="description"
content="W3C is about 400 organizations leading the World Wide Web to its full potential. Founded by Tim Berners-Lee, the
Web's inventor. The W3C Web site hosts specifications, guidelines, software and tools. Public participation is welcome. W3C
supports universal access, the semantic Web, trust, interoperability, evolvability, decentralization, and cooler multimedia."
/>
<title>World Wide Web Consortium</title>
<link id="meta" rel="rel:meta"
href="http://www.w3.org/2000/08/w3c-synd/#meta" />
<link rel="meta" href="Overview-about.rdf" />
<link rel="stylesheet" type="text/css" href="/StyleSheets/home.css" />
<link rel="bookmark" href="#technologies" title="Technologies |" />
<link rel="bookmark" href="#news" title="News |" />
<link rel="bookmark" href="#search" title="Search |" />
<link rel="contents" href="#contents" title="Contents |" />
<link rel="bookmark" href="#systems" title="Systems |" />
<link rel="bookmark" href="#donors" title="Donors |" />
<link rel="bookmark" href="#footnotes" title="Footnotes |" />
<link rel="alternate" type="application/rss+xml" title="RSS"
href="http://www.w3.org/2000/08/w3c-synd/home.rss" />
<style type="text/css">
<!--
html, body { background: #fff; color: #000; } /* body for Windows IE3 */
-->

</style>
</head>
<body>
<h1 id="logo"><img alt="The World Wide Web Consortium (W3C)"
height="48" width="315" src="Icons/w3c_main" /></h1>
```

Рис. 3.18, *б*. Документ HTML: при редактировании в текстовом редакторе

纸 *Дополнительно*

Вы прекрасно знаете, что именно HTML является основным форматом для кодирования веб-страниц. Хотя было бы ошибкой утверждать, что документ HTML и веб-страница — это одно и то же. Под *веб-страницей* мы будем понимать то, что конечный пользователь видит у себя на экране — то есть полноценный информационный ресурс со всеми изображениями, оформлением и функциональными возможностями. При этом веб-страница редко состоит из одного файла HTML. Чаще всего она складывается, кроме документа HTML, из одной или нескольких таблиц стилей, файлов изображений, а также, возможно, внедренных объектов (Flash, апплеты) и отдельных файлов с клиентскими сценариями (обычно на JavaScript).

✓ *Примечание*

Начинающие разработчики HTML иногда задают вопрос: с каким расширением лучше сохранять файлы — `html` или `htm`? В большинстве случаев эти расширения совершенно равноправны, можно спокойно использовать любое из них. Расширение `html` более распространено и считается общепринятым, поэтому,

если у вас нет никаких специальных предпочтений, рекомендуется употреблять именно его.

Иногда от настроек веб-сервера зависит, какое расширение следует придавать файлам HTML. Перед тем как размещать свои страницы во Всемирной Паутине, поинтересуйтесь у администратора веб-сервера, нет ли на этом сервере каких-нибудь специальных правил именования файлов.

Помните, что в системах класса UNIX (под которыми функционирует большинство веб-серверов) прописные буквы отличаются от строчных. В этом случае `index.html` и `INDEX.HTML` — два совершенно разных имени. В именах файлов HTML (опять же, если на сервере не действуют какие-либо особые правила) лучше использовать только строчные буквы.

Дополнительно

И еще о расширениях. В UNIX, в отличие от Windows, расширения не играют специальной роли с точки зрения операционной системы и являются просто частью имени файла. Поэтому пользователи UNIX часто называют их не расширениями, а *суффиксами*. Имя файла в UNIX часто может не иметь суффиксов вовсе, иметь несколько суффиксов (document.new.pdf.tar.gz) или вообще состоять из одного суффикса (.bash_history, .tex).

Особенности кодирования документов HTML и CSS будут подробно обсуждаться в *главе 5*.

PostScript

PostScript — это интерпретируемый язык программирования, обладающий особенно развитыми возможностями в области формирования и вывода векторных изображений. Программа на языке PostScript описывает документ (например, текстовую страницу, предназначенную для вывода на печать) в виде процедур, формирующих совокупность векторных объектов — графических примитивов (отрезков, дуг, кривых Безье и др.), специальных форм (например, текстовых символов) и других. Каждый векторный объект обладает определенными свойствами (может иметь границу определенной формы и цвета, быть закрашенным, совмещаться с другими объектами и т. д.). Кроме того, в документ могут быть встроены сторонние растровые объекты (например, фотографии).

Интерпретаторы PostScript осуществляют *растеризацию* векторного изображения, то есть его преобразование в растровое изображение, которое непосредственно выводится на экран, принтер или другое устройство. В зависимости от размеров, разрешения, цветовых возможностей и других характеристик устройства вывода интерпретатор генерирует наиболее подходящее к нему изображение. Исходный файл на языке PostScript никак не связан с устройством вывода и поэтому обладает хорошей переносимостью.

PostScript — полноценный язык программирования, поддерживающий различные типы и структуры данных (например, числа или текстовые строки, а также массивы) и стандартные алгоритмические конструкции (например, условные переходы, циклы и подпрограммы). Он обеспечивает крайне высокую точность вывода результата (если используются технические средства надлежащего качества): имеются в виду как точность размещения объектов на странице, так и точность передачи цветов.

Благодаря своим свойствам PostScript является популярным форматом для документов, которые нужно вывести на печать с профессиональным качеством.

Файлы в формате PostScript обычно имеют расширение (суффикс) ps.

Примечание

Разработкой и поддержкой языка PostScript занимается компания Adobe. Первая версия этого языка (так называемая LanguageLevel 1) появилась в 1985 году; на момент написания этой книги существует PostScript LanguageLevel 3. Официальная страница формата PostScript — **www.adobe.com/products/postscript**. Стандарт PostScript LanguageLevel 3 можно загрузить по адресу **www.adobe.com/products/postscript/pdfs/PLRM.pdf**.

PDF (Portable Document Format)

PDF — еще один формат, принадлежащий компании Adobe. Эта аббревиатура расшифровывается как «Portable Document Format» и переводится как «переносимый формат документов». Из самого названия следует, что формат PDF предназначен для хранения документов, которые должны одинаково хорошо отображаться на разных компьютерах с разными операционными системами и печататься на разных принтерах и других устройствах.

В отличие от Postscript, PDF не является языком программирования. Это скорее язык *описания* документов (как и HTML). Формат PDF использует ту же базовую модель представления объектов, что и PostScript, но не обладает всей мощью и гибкостью PostScript. Поэтому файлы в формате PDF чаще всего не редактируют. Документ обычно создается и редактируется в каком-либо промежуточном формате (например, в TEX, DOC или том же PostScript), а потом преобразуется в PDF как в конечную форму, предназначенную для распространения. Если в документ необходимо внести изменения, то они обычно вносятся не в файл PDF, а в первоначальный формат, и затем PDF генерируется заново.

Документ PDF содержит не только текст, изображения и другие объекты, непосредственно отображаемые на странице, но и структурную информацию — т. е. сведения о делении документа на логические части (главы,

параграфы и т. д.) и взаимосвязи между этими частями. Кроме того, в PDF могут присутствовать ссылки, ведущие как на другие фрагменты этого же документа, так и к любым ресурсам на текущем компьютере или в Интернете. Также в документ можно вставить мультимедийные объекты и элементы интерактивного взаимодействия (например, формы для заполнения).

Формат PDF стал фактическим стандартом для распространения в Сети документов, предназначенных для печати. Однако его вряд ли стоит применять для документов, которые предполагается просматривать с экрана, и он совершенно не подходит в качестве языка для веб-страниц (вместо HTML). Для просмотра PDF требуются специальные программные средства, файлы имеют слишком большой размер, принципы работы с ними отличаются от работы с обычными веб-страницами — все это причиняет пользователю неудобства и сводит на нет все преимущества PDF.

Файлы в формате PDF обычно имеют расширение pdf.

Дополнительно

Некоторые современные поисковые системы — например, Google (**www.google.com**) — умеют анализировать и индексировать размещенные в Интернете файлы PDF так же хорошо, как и документы HTML. Документы в формате PDF часто можно встретить среди результатов поиска. Это лишний раз показывает распространенность этого формата.

Внимание!

PDF — фактически общепринятый стандарт для распространения документов, предназначенных для печати. Если вы хотите предоставить пользователям возможность распечатывать ваши веб-страницы — подготовьте версии этих страниц в формате PDF и организуйте на каждой странице заметную для пользователей ссылку на файл PDF, соответствующий этой странице.

Примечание

На момент написания этой книги наиболее свежей версией является PDF 1.5. Официальная страница формата PDF — **www.adobe.com/products/acrobat/adobepdf.html**. Спецификацию языка PDF (в формате PDF!) можно загрузить по адресу: **partners.adobe.com/asn/acrobat/sdk/public/docs/PDFReference15_v6.pdf**.

Компания Adobe рекомендует использовать для просмотра документов PDF программу Adobe Reader, которая распространяется бесплатно. Наиболее свежая версия на момент написания этой книги — Adobe Reader 6.0.2, которую можно загрузить со страницы **www.adobe.com/products/acrobat/readstep2.html**.

3.3. При всем богатстве выбора…
Программы для работы с текстовыми файлами

> Нередко случаются травмы, когда пытаются использовать различные предметы и инструменты не по их прямому назначению: утюгом забивают гвозди, примусом разжигают костер, перочинным ножом открывают консервную банку, плоскогубцами удаляют зубы.
>
> *«Горняк», № 53, 2003*

> Умный человек учится в основном на чужих ошибках, глупый — в основном на своих, а дурак вообще никогда не учится.
>
> *Народная мудрость*

Все программы для работы с текстом можно разделить на следующие группы:

- программы просмотра;
- программы редактирования;
- программы преобразования форматов;
- прочие (например, программы проверки орфографии или оптимизации файла).

К *программам просмотра* относятся, например, браузеры и упоминавшийся ранее Adobe Reader. Эти программы не позволяют пользователю изменять файл, они только отображают его и предоставляют дополнительные средства работы с файлом, не воздействующие на сам файл — поиск текста, выделение фрагментов файла и их копирование в буфер обмена, применение интерактивных средств, внедренных в файл и т. д.

Программы редактирования, наоборот, применяются для первоначального создания и последующего изменения файлов. Блокнот, Microsoft Word, Emacs, Adobe PageMaker — все это программы редактирования. Как правило, программы редактирования также обеспечивают все возможности программ просмотра.

Программы преобразования форматов, как несложно догадаться, преобразуют файлы из одного формата в другой. Такова, например, система TEX, переводящая исходный код, написанный на языке TEX, в промежуточный формат DVI, пригодный для вывода на печать и дальнейшего преобразования в другие форматы. Программа Adobe Acrobat превращает файлы различных форматов (в том числе RTF и DOC) в документы PDF.

Наконец, существуют программы, обрабатывающие текст, но не относящиеся ни к одному из перечисленных классов.

Дополнительно

Например, есть программа HTML Tidy, проверяющая код HTML и исправляющая ошибки. Она, как и программы проверки орфографии, не способна выявить все ошибки, но часто оказывается весьма полезным дополнением к остальным средствам проверки кода. Официальная страница этой программы — **tidy.sourceforge.net**.

Многие программы объединяют в себе несколько из перечисленных функций. Таковы практически все текстовые процессоры — скажем, Microsoft Word.

В дальнейшем пойдет речь в основном о программах редактирования. Среди них также можно выделить несколько групп:

- текстовые редакторы;
- текстовые процессоры;
- компьютерные издательские системы.

Чтобы эффективно работать с текстом, нужно разбираться в том, какие программы для чего предназначены. Часто возникают ситуации, когда разработчик использует не ту программу, которая ему действительно нужна.

Дополнительно

Например, некоторые пользователи компьютеров создают веб-страницы в текстовом процессоре Microsoft Word. Результат этой работы обычно не удовлетворяет никаким критериям качества, да и сами создатели таких страниц работают неэффективно, затрачивают лишнее время и усилия. Всех этих проблем можно избежать, если разрабатывать веб-страницы с помощью более подходящих программ — например, простого текстового редактора или интегрированной среды, специально предназначенной для этой задачи.

Другая проблема заключается в том, что пользователи иногда слабо владеют программами, с которыми работают. В результате программы используются малопродуктивно, и результат снова получается неудовлетворительный, а разработчики тратят лишнее время и силы.

Внимание!

Применяйте для каждой задачи тот инструмент, который лучше всего подходит именно для нее.

Освойте программу, с которой собираетесь работать. Научитесь использовать ее возможности наиболее полно и продуктивно.

Примечание

Большинство хороших программ снабжены справочной системой, а также пособиями по работе с ними. К сожалению, пользователи склонны игнорировать эти средства; вместо этого они предпочитают учиться «методом тыка». Автор надеется, что вы не относитесь к таким пользователям.

3.3.1. Текстовые редакторы

> The other fact that you need to understand is that vi is a modeful editor, i.e. you are either entering text or you are executing commands, and you have to be in the right mode to do one or the other.
>
> *vi(1) man page*[2]

Текстовые редакторы предназначены для работы с обычным текстом. Они открывают файл в текстовом режиме, позволяя пользователю просматривать его в «первозданном» виде.

Редакторы обычно используются для набора простых текстов, где не требуется никакого оформления, а также для создания исходных кодов программ, документов HTML, таблиц стилей CSS и других подобных файлов.

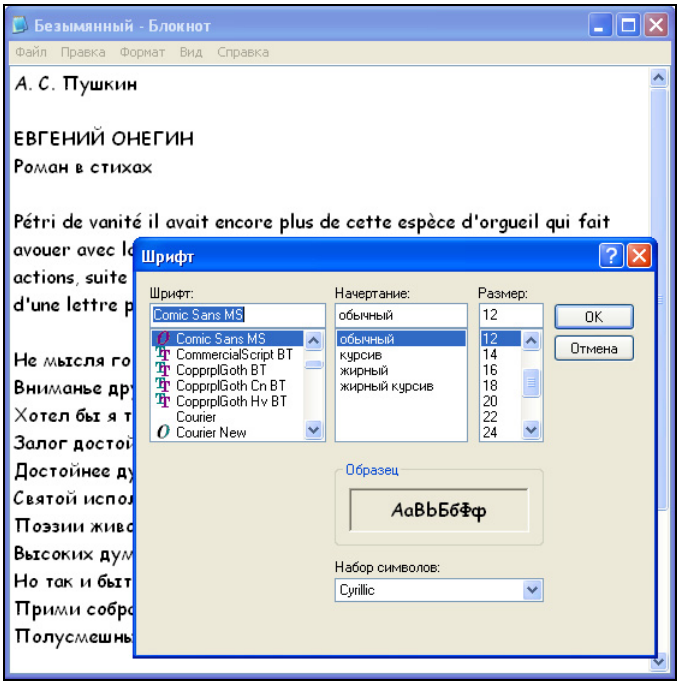
В любой современной операционной системе есть стандартный текстовый редактор. Например, в Windows это Блокнот, а в большинстве систем UNIX — Vi или Vim. Впрочем, многие пользователи используют для работы с обычным текстом не стандартные средства операционных систем, а сторонние программы, обладающие, как правило, более широкими возможностями.

Примечание

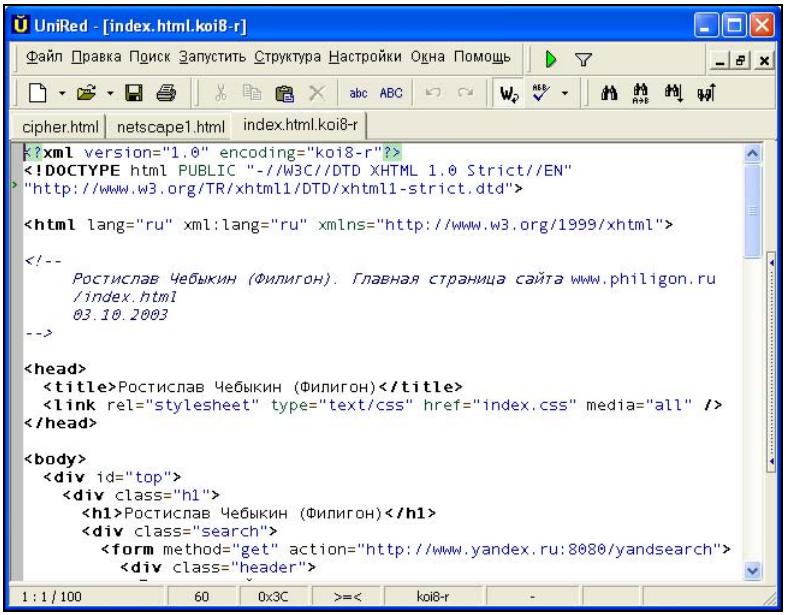
Почти все текстовые редакторы позволяют изменять параметры шрифта (рис. 3.19, *а*). Многие редакторы также обладают синтаксической подсветкой — при этом в тексте выделяются определенные слова или специальные конструкции (например, ключевые слова языка программирования или тэги HTML — рис. 3.19, *б*). Начинающие пользователи часто не понимают, что в самом файле информация о шрифтах или выделениях при этом никак не сохраняется. При переносе файла на другой компьютер или даже в другой редактор шрифт может стать совсем другим, а подсветка и вовсе исчезнет.

---

[2] Другая вещь, которую вам необходимо усвоить, — то, что vi является редактором с переключением режимов, т. е. вы либо вводите текст, либо выполняете команды; для того чтобы выполнить то или другое действие, вы должны находиться в нужном режиме. *(Справочная страница текстового редактора vi)*



а



б

Рис. 3.19. *а* — редактор Блокнот позволяет изменить шрифт; *б* — редактор UniRed обладает синтаксической подсветкой. Но все эти эффекты влияют только на просмотр текстового файла и не сохраняются в самом файле

Важно помнить: в обычном тексте не сохраняется никакой информации о его внешнем виде, оформлении, форматировании. Внешний вид текста целиком зависит от настроек текстового редактора и не сохраняется при переносе файла в другую среду.

Глава 4 полностью посвящена принципам работы с текстовым редактором.

3.3.2. Текстовые процессоры

> Ошибаетесь иногда? Конечно, да, ибо ошибается каждый человек. Но зачем утруждать этим самого себя? Наша засвидетельствованная патентом ошибалка сделает все за вас! Наша новая ошибалка — модель FV 1303 — развивает до 699 ошибок в день!
>
> *К. Чапек. Изобретатель*

К текстовым процессорам относятся программы, предназначенные для создания и редактирования форматированного текста — т. е. текстовых файлов, в которых сохраняется информация об оформлении, структуре и т. д. *(см. разд. 3.2.3)*. Среди текстовых процессоров — уже многократно упоминавшийся Microsoft Word, а также OpenOffice, StarOffice, KOffice и немало других продуктов.

Эти программы очень популярны в широких кругах пользователей, они повсеместно применяются для создания и редактирования текстовых документов.

Несмотря на широкое распространение, пользователи обычно работают с текстовыми процессорами не так грамотно и квалифицированно, как следовало бы. Вот несколько ситуаций, которые подводят неопытных пользователей чаще всего:

- пользователи слишком полагаются на автоматическую проверку орфографии и грамматики. У многих вырабатывается вредный стереотип: «если подчеркнулось — значит, ошибка, если не подчеркнулось — значит, нет ошибки». На самом деле, программы автоматической проверки грамматики далеко не так совершенны, как кому-то кажется. Они не заметят ошибки, если вы напишете «робота» вместо «работа» или «как то» вместо «как-то». Зато, наоборот, они часто считают ошибкой правильно написанные слова — «браузер», «веб-страница», «моноширинный». Никакие ныне существующие программы автоматической проверки не заменят живого человека;
- автоматическая замена и автоматическое форматирование также часто вызывают затруднения. Например, способность некоторых текстовых процессоров исправлять две прописные буквы в начале слова приводит к тому, что название программы KOffice превращается в Koffice. Также

процессоры норовят начинать каждое предложение с прописной буквы, в результате чего вместо «и т. д.» получается «и т. Д.»;

Дополнительно

Автор испытал много проблем с текстовым процессором Microsoft Word, пока набирал эту книгу. Например, нужно написать «(С)» (прописную букву «С» в круглых скобках) — а Word автоматически заменяет эту конструкцию на «©». Нужно поставить прямые кавычки (" ") — а Word автоматически заменяет их на парные (« » или “ ”). Пришлось немало покопаться в настройках программы, чтобы запретить такие замены. Но, к сожалению, это помогло только на компьютере автора. При обработке текста в издательстве снова начали происходить автоматические исправления там, где их совершенно не требовалось.

- текстовые процессоры не умеют сами по себе соблюдать правила компьютерного набора и верстки. Выполнения большинства правил из *разд. 2.1* можно добиться только вручную. К сожалению, пользователи, набирающие тексты с помощью Microsoft Word и других текстовых процессоров, обычно не разбираются в правилах набора, и документы получаются неграмотными и попросту некрасивыми;
- текстовые процессоры не учитывают нашей отечественной специфики набора и типографских традиций. Мало того, многие из них провоцируют на нарушение правил. Например, в настройках автозамены Microsoft Word, даже в русской локализации, есть флажок **Писать названия дней с прописной буквы**, который по умолчанию установлен! (Для тех, кто не понял суть проблемы: это в английском языке названия дней недели обычно пишутся с прописной буквы; в русском же языке они всегда пишутся со строчной, и этот флажок только дезориентирует пользователей.);
- современные текстовые процессоры обладают огромным набором разнообразных возможностей. К сожалению, пользователи обычно не знают об этих возможностях и используют только 5–10 % функций программы. В то же время, если овладеть многими неиспользуемыми функциями, можно существенно повысить производительность собственной работы и улучшить качество документов.

Дополнительно

Например, многие пользователи не подозревают о специальных сочетаниях клавиш, которые позволяют выполнять большинство операций гораздо быстрее, чем с помощью мыши. Простой пример: допустим, нам нужно в Word применить к выделенному фрагменту текста полужирное начертание шрифта. Чтобы сделать это с помощью мыши, мы совершаем следующие действия:

- отнимаем руку от клавиатуры и переносим ее на мышь;

- подводим указатель мыши к кнопке **Ж** на верхней панели инструментов (заметим, что при этом еще требуется прицелиться и попасть в эту кнопку!);
- щелкаем кнопкой мыши;
- отнимаем руку от мыши и переносим ее на клавиатуру.

Нам потребовалось совершить целых четыре разных движения, причем три из них потребовали достаточно далеких перемещений руки. А ведь достаточно просто нажать комбинацию клавиш <Ctrl>+<B> — не отнимая руки от клавиатуры, не совершая никаких посторонних движений, — и выделенный текст станет полужирным!

Использование сочетаний клавиш позволяет выполнять работу в несколько раз быстрее, чем с помощью мыши. Просто удивительно, что люди, постоянно жалующиеся на нехватку времени, продолжают пользоваться мышью!

Еще один пример неиспользуемых возможностей — это стили. Например, набирая эту книгу, автор выделял все адреса Интернета подчеркиванием. Однако издательство порекомендовало убрать подчеркивание и сделать адреса полужирными. Потребовалось внести одно-единственное изменение в соответствующий стиль — и все адреса в тексте, как по мановению волшебной палочки, мгновенно приобрели полужирное начертание! А пользователю, не умеющему работать со стилями, пришлось бы просматривать весь текст и вручную изменять форматирование каждого адреса.

WYSIWYG

Практически все текстовые процессоры работают в режиме WYSIWYG («визивиг»), что расшифровывается как «What You See Is What You Get» («Что видишь, то и получаешь»). Это значит, что, набирая и редактируя текст, разработчик видит его точно таким же, каким он будет представлен читателю или пользователю — со всем оформлением и форматированием.

Это отличается, например, от работы в системе TEX или от создания исходного кода HTML в простом текстовом редакторе вроде Блокнота. В этих случаях разработчик создает код на специальном языке, вручную записывая все управляющие конструкции. Например, чтобы выделить текст полужирным шрифтом, в TEX потребуется написать `\bf`, в прежних версиях HTML — `<b>`, а в CSS — `font-weight: bold;`. Понятно, что читатель публикации или пользователь сайта увидит не эти конструкции, а обычный текст, выделенный полужирным шрифтом.

Получается, что для работы в таком режиме требуется знать определенный язык кодирования — допустим, TEX или HTML. А в случае WYSIWYG достаточно только общей компьютерной грамотности разработчика. Поэтому тысячи пользователей набирают документы в Microsoft Word, и только десятки — в TEX.

Тем не менее, WYSIWYG обладает рядом существенных недостатков, которые ограничивают применение этого режима там, где требуется профессиональное качество результата:

- WYSIWYG часто дезориентирует разработчика. Посмотрите на снимок экрана (рис. 3.20): как вы думаете, слова «Евгений Онегин» выглядят так потому, что действительно написаны белым цветом на черном фоне, или потому, что разработчик пометил их, чтобы применить какое-нибудь оформление?

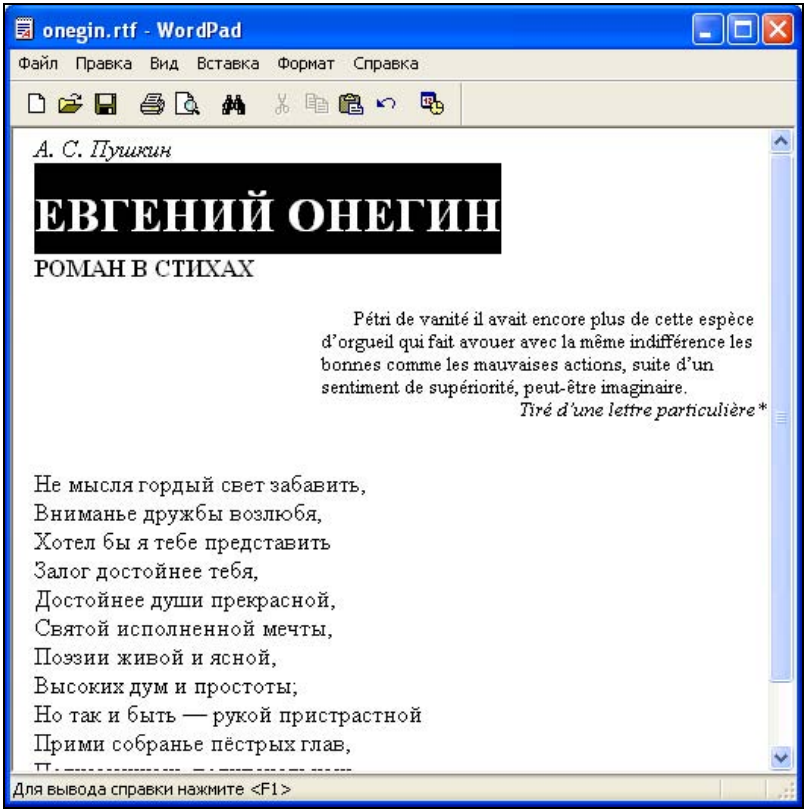


Рис. 3.20. Белым по черному: взглянув на экран, невозможно догадаться, оформление это или временное выделение

- WYSIWYG не дает разработчику полного контроля над редактируемым файлом. Разработчик видит только конечный результат и не может вмешаться в то, как этот результат представлен в исходном коде. При этом становятся нерешаемыми многие проблемы — например, связанные с переносом файлов с одного компьютера на другой. Бывают случаи, когда файл в формате DOC, созданный на одном компьютере, открывается

на другом — и оказывается искаженным, причем причина этого искажения остается загадкой. Если такое происходит с документом HTML — разработчик обычно может просмотреть исходный код и разобраться, что именно в этом коде приводит к различиям;

- разрабатывать документ, наблюдая только его внешний вид, — все равно что строить дом, следя только за его фасадом. Например, обычный пробел и неразрывный пробел выглядят совершенно одинаково; также одинаковы обычный дефис и мягкий перенос. Но у них разное назначение, поэтому в процессе редактирования важно видеть, где в документе обычные пробелы и дефисы, а где — специальные;
- WYSIWYG не позволяет управлять логической и структурной организацией документа. Представьте себе, что все заголовки в документе требуется выделить зеленым цветом. В режиме WYSIWYG придется просматривать весь текст, разыскивать в нем заголовки и выделять каждый из них в индивидуальном порядке. Представьте, сколько времени на это уйдет, если в документе 100 заголовков! А в случае грамотно сверстанного документа HTML достаточно будет изменить одну строчку в таблице стилей — и все заголовки в тексте сразу станут зелеными, будь их хоть десять тысяч.

Надо сказать, что разработчики современных текстовых процессоров приложили усилия к тому, чтобы устранить эти недостатки.

纸 ***Дополнительно***

Например, в Microsoft Word можно переключиться в режим, при котором видны все специальные знаки — в частности, пробелы, неразрывные пробелы, символы табуляции и перевода строки. Можно также пометить фрагменты текста как заголовки.

Однако чем меньше остается перечисленных недостатков — тем больше требований предъявляется к квалификации пользователя. Чтобы работать с Microsoft Word и не испытывать двух последних проблем, нужны специальные знания, вполне сравнимые со знаниями профессионала в TEX или HTML. Таким образом, избавляясь от недостатков WYSIWYG, мы теряем и его преимущества.

Среды разработки сайтов

К разряду текстовых процессоров можно отнести и большинство известных сейчас интегрированных сред разработки сайтов — например, Macromedia Dreamweaver, HomeSite, Microsoft FrontPage. Эти среды обычно позволяют работать в режиме WYSIWYG, редактируя непосредственно внешний вид веб-страницы. На основании этого внешнего вида среда генерирует код HTML и CSS.

Этот принцип вызывает серьезные затруднения даже при подготовке печатных публикаций, а при разработке веб-страниц он становится совершенно неприемлемым. Для веб-страницы внутренняя структура имеет гораздо большее значение, чем внешний вид. Это связано с тем, что, в отличие от печатной публикации, одна и та же веб-страница может совершенно по-разному отображаться на разных компьютерах, в разных операционных средах и браузерах.

Разработчик в режиме WYSIWYG обеспечивает правильное отображение страницы только на своем собственном компьютере, в своем собственном браузере. Он ничего не может поделать с тем, что в других условиях страница будет искажена.

Примечание

Представьте себе, что вы задаете компьютеру внешний вид автомобиля — и компьютер должен на основании этой информации автоматически спроектировать все его внутреннее устройство, от двигателя до руля. Согласитесь, ситуация абсурдная: ведь если мы заменим двигатель, внешний вид машины не изменится. Как же компьютер по внешнему виду должен догадаться, какой именно нужен двигатель? А ведь в точности такой «разработкой» занимаются те, кто создает веб-страницы в режиме WYSIWYG. Внешний вид страницы определяется ее внутренней структурой (и еще некоторыми факторами — например, свойствами компьютера пользователя), а не наоборот.

Внимание!

Не используйте для разработки веб-страниц режим WYSIWYG.

Большинство сред разработки позволяет работать не только в режиме WYSIWYG, но и в режиме редактирования напрямую исходного кода HTML и CSS. Но если разработчик предпочитает редактировать непосредственно исходный код — ему совершенно не нужна громоздкая среда вроде Macromedia Dreamweaver, достаточно простого текстового редактора.

Среды разработки имеют множество других особенностей, которые сбивают с толку начинающих создателей сайтов:

- каждая такая среда обладает специфическим интерфейсом, позволяющим реализовать массу ее возможностей. Чтобы полноценно использовать функции программы, необходимо хорошо изучить ее интерфейс. Таким образом, кроме собственно языков HTML и CSS, а также некоторых других важных дисциплин (например, типографики), разработчику приходится осваивать еще принципы работы с программой (Надо сказать, что иногда это требует больше времени и усилий, чем все перечисленные дисциплины, вместе взятые.);

Дополнительно

Среды разработки не учитывают специфики русскоязычных страниц. Например, для страниц на русском языке совершенно не нужно вставлять в шапку документа мета-тэг, обозначающий кодировку (например: `<meta http-equiv="Content-Type" content="text/html; charset=koi8-r" />`). Вместо этого кодировка (по крайней мере, для русскоязычных страниц) должна быть указана в заголовках HTTP, пересылаемых веб-сервером. Такой мета-тэг в большинстве случаев является попросту лишним и не выполняет никакой полезной функции, а в некоторых ситуациях может даже навредить (например, если указанная в нем кодировка отличается от той, которую сообщил сервер).

Однако многие среды разработки (ориентированные вовсе не на русскоязычные страницы!) автоматически вставляют этот тэг в документ, и неопытные разработчики считают, что так и должно быть.

- среды разработки обычно генерируют избыточный, неоптимальный код. Этот код можно значительно сократить, причем в большинстве случаев сокращение не только не причиняет странице ущерба, но и, наоборот, улучшает ее функциональность, удобство и доступность;
- среды разработки часто создают неграмотный код, не соответствующий официальным стандартам языков HTML и CSS;
- страницы, сгенерированные средами разработки, часто не обладают совместимостью с различными браузерами;
- существуют достаточно распространенные тонкости верстки HTML и CSS, которые можно реализовать, только редактируя вручную исходный код. Некоторые из таких тонкостей будут рассмотрены в *главе 5*.

Чтобы избавить страницу от четырех последних недостатков, приходится вручную исправлять исходный код, сокращать его, оптимизировать, приводить к соответствию стандартам, решать проблемы браузерной совместимости. А это всегда дольше и сложнее, чем «с нуля» набирать грамотный код в простом текстовом редакторе.

Примечание

Исправлять допущенные ошибки — это всегда сложнее, чем не допускать их вовсе. Конечно, в жизни невозможно обойтись совсем без ошибок. Но если в конкретном случае есть возможность их не делать, — зачем же работать в таком режиме, в котором эти ошибки неминуемо возникнут, и их придется потом исправлять?

Вместе с тем, среды разработки имеют много полезных функций — например, разнообразный анализ исходного кода, автоматический расчет размеров

элементов страницы, просмотр страницы в различных видеорежимах без переключения реальных свойств экрана и т. д. Но все эти функции полезны для квалифицированных, опытных разработчиков, которые хорошо разбираются в HTML, CSS, удобстве и доступности сайтов и других дисциплинах. В руках таких специалистов среда разработки облегчает работу и повышает качество результата. Но для начинающих, неопытных разработчиков, по мнению автора, интегрированные среды разработки сайтов крайне вредны, они напоминают осциллограф в руках первобытного человека.

3.3.3. Издательские системы

> С другой стороны, возможен ли Ахиллес в эпоху пороха и свинца? Или вообще «Илиада» наряду с печатным станком и тем более типографской машиной? И разве не исчезают неизбежно сказания, песни и музы, а тем самым и необходимые предпосылки эпической поэзии, с появлением печатного станка?
>
> *К. Маркс. О значении греческого искусства*

Компьютерные издательские системы — это программы, с помощью которых подготавливают печатные публикации: книги, газеты, журналы, листовки и т. д. Здесь о них будет сказано совсем немного, так как прямого отношения к разработке сайтов они не имеют.

Среди издательских систем известны такие программные продукты, как QuarkXPress, Ventura Publisher, Adobe PageMaker и др. Они рассчитаны на разработку документов, которые будут выводиться на печать с полиграфическим качеством.

Издательские системы постоянно развиваются и совершенствуются. Однако можно наблюдать неожиданное явление: несмотря на развитие технологий, часто встречаются газеты, журналы и даже книги, набранные и отпечатанные некачественно, с многочисленными нарушениями правил набора и верстки, а также действующих ГОСТов на издательскую продукцию.

纸 Дополнительно

Посмотрите, например: часто ли вам попадаются издания, в которых соблюдается правило о том, что в конце строки нельзя оставлять однобуквенные предлоги и союзы? А как часто вы видите издания, в которых употребляются грамотные кавычки-«елочки», а не суррогаты (" ") или английские кавычки (“ ”)?

Иногда доходит до смешного: некоторые современные издания, выпущенные с помощью новейших технических и программных средств, оказываются гораздо менее качественными, чем многие старинные книги, набранные вручную.

Почему это происходит? Неужели компьютерные издательские системы — ущербная технология по сравнению с наборными кассами и древними печатными прессами? Неужели для повышения качества издательской продукции следует возвращаться к технике XIX века?

Конечно же, не следует! Проблема заключается вовсе не в технологиях, а в тех людях, которые пользуются этими технологиями. Если вы не умеете играть на фортепиано — то даже если в вашем распоряжении есть самый совершенный концертный рояль, вы все равно не сможете качественно исполнить Второй концерт Рахманинова. Эта мысль кажется очевидной — и тем не менее, когда речь заходит об издательских технологиях или о разработке сайтов, о ней почему-то забывают.

Компьютер — это не волшебный ящик, который без всякого вашего вмешательства сделает всю работу в наилучшем виде. Это всего лишь рабочий инструмент — такой же, как молоток, рояль или осциллограф. Качество инструмента само по себе вовсе не гарантирует того, что выполненная с его помощью работа также будет качественной.

Чтобы работа получалась действительно высокого класса, абсолютно необходимо, чтобы разработчик имел специальное образование, обладал соответствующими знаниями, умениями и навыками, знал правила, стандарты и ГОСТы, ориентировался в истории своей области и ее современных достижениях, а также владел общей професиональной культурой и стремился постоянно учиться и повышать свою квалификацию.

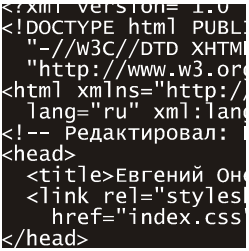
Может быть, когда-нибудь компьютерные технологии и превратятся в те самые волшебные ящики. Но сейчас никакой компьютер не исправит все ошибки орфографии и пунктуации, не проследит за стилистикой текста и соблюдением типографских правил, не разработает логику и структуру документа. Кроме того, немаловажен также привлекательный внешний вид печатной публикации или веб-страницы — а это сможет обеспечить только человек. Ведь компьютеры пока еще не знают, как отличать привлекательный внешний вид от непривлекательного.

Если разработчик слепо полагается на компьютерные технологии, не заботясь о собственных знаниях, умениях и квалификации, у него получаются низкокачественные, безграмотные поделки, даже если он использует самый современный компьютер и новейшие версии программ. Важная часть культуры разработчика состоит в том, чтобы представлять себе, какие задачи можно автоматизировать и поручить компьютеру, а какие должен решать живой человек.

Дополнительно

Допустим, компьютер вполне способен самостоятельно проверить, соответствует ли код вашей веб-страницы стандартам HTML и CSS. Это чисто формальная операция, за которой можно обратиться, например, на сайт **validator.w3.org**. А вот проверить, соответствует ли страница принципам удобства и доступности для пользователя, компьютерам пока не под силу.

Если вы грамотный, квалифицированный разработчик, владеющий необходимыми знаниями, умениями и опытом, компьютерные технологии в ваших руках станут прекрасным подспорьем, помогающим выполнить работу эффективнее и качественнее. Но если вы надеетесь, что компьютер будет работать вместо вас, а вам самим не нужно ничего знать и уметь, — ничего хорошего не получится.



Глава 4

«Блокнот» мой — друг мой

Эффективная работа с текстовым редактором

С «лейкой» и с блокнотом,
А то и с пулеметом
Сквозь огонь и стужу мы прошли.

К. Симонов. Песенка фронтовых корреспондентов

Острый Карандаш трудился с утра до вечера: его и строгали, и ломали, и в работе не щадили. А к Тупому Карандашу и вовсе не притрагивались: раз попробовали его вовлечь, да сердце у него оказалось твердое. А от твердого сердца ни в каком деле толку не жди.

Ф. Кривин. Мемуары

Текстовый редактор — это программа, с помощью которой можно создавать и редактировать файлы в формате «обычный текст», а также исходный код документов HTML, таблиц стилей CSS, компьютерных программ и других специальных файлов.

Пользователям Windows особенно хорошо знаком стандартный текстовый редактор Блокнот. Но существует и множество других редакторов, в том числе совершенно не похожих на Блокнот. Например, к текстовым редакторам относятся:

- обычные редакторы — кроме Блокнота, в эту группу входят UniRed, Aditor, UltraEdit и множество других программ;
- встроенные редакторы интегрированных сред разработки — во многих программных системах, популярных среди начинающих разработчиков (например, Dreamweaver или HomeSite) есть встроенные текстовые редакторы, позволяющие вносить изменения непосредственно в исходный код документов. Встроенные редакторы также есть в некоторых файловых менеджерах — например, в FAR Manager;

Примечание

Вы пользуетесь текстовыми редакторами на каждом шагу, не всегда обращая на это внимание. Например, когда вы набираете сообщение электронной почты, то делаете это с помощью текстового редактора, встроенного в вашу почтовую программу. Даже когда вы вводите запрос в строку поисковой системы (например, Яндекс или Google), эта строка сама по себе является очень простым текстовым редактором.

- текстовые процессоры и издательские системы, работающие в режиме «обычный текст». Например, в Microsoft Word можно создать и сохранить файл в формате «обычный текст». Впрочем, Word и другие подобные программы гораздо лучше подходят для создания форматированного текста и очень редко используются в качестве текстовых редакторов.

В этой главе рассматриваются некоторые общие свойства текстовых редакторов и универсальные приемы работы с ними, благодаря которым можно эффективно разрабатывать документы в самых разных редакторах.

4.1. Кто на свете всех милее?
Как выбрать текстовый редактор

> Уж как трудно решиться, так просто рассказать нельзя, как трудно! Если бы губы Никанора Ивановича да приставить к носу Ивана Кузьмича, да взять сколько-нибудь развязности, какая у Балтазара Балтазарыча, да, пожалуй, прибавить к этому еще дородности Ивана Павловича — я бы тогда тотчас же решилась.
>
> *Н. Гоголь. Женитьба*

> — У меня бы здорово пошло по части музыки, — говорит дядюшка Кэл. — Я выбрал лучший инструмент, какой можно достать за деньги в Сан-Антонио. Ведь верно, Марилла, что пианино хорошее во всех отношениях?
>
> *У. О'Генри. Пианино*

Не существует никакого «самого лучшего» текстового редактора. У каждого редактора есть свои преимущества и недостатки. Разработчику рекомендуется хорошо представлять себе возможности разных программ, их сильные и слабые стороны. Только тогда он сможет выбрать для работы именно тот редактор, который подходит ему лучше всего.

Принимайте во внимание два фактора, влияющих на выбор текстового редактора:

- **задача, для которой применяется редактор** — вы используете редактор для того, чтобы сделать с его помощью текстовый файл — например, сообщение электронной почты, документ HTML или программу на C++. Значит, нужно выбрать именно такой редактор, который позволит вам лучше всего выполнить эту задачу. Вполне возможно, что для разных задач вы будете применять разные текстовые редакторы;
- **ваши предпочтения и привычки** — разным разработчикам удобнее работать с разными редакторами, и в этом нет ничего необычного. Опытным пользователям UNIX привычен и удобен интерфейс таких текстовых редакторов, как Vim или Emacs, в то время как пользователи Windows нередко считают этот интерфейс пугающим.

Примечание

Не утихают «религиозные споры» о том, какой редактор лучше всех. Самое разумное, что можно сделать, — это держаться подальше от таких споров. То же самое верно для любых других программных продуктов: «самый лучший браузер» или «самая лучшая почтовая программа» — это миф. В одной ситуации «самой лучшей» будет одна программа, в другой — другая.

Квалифицированный пользователь или разработчик никогда не высказывается в стиле «Мой любимый редактор — лучше всех!». Для каждой конкретной задачи он выбирает тот инструмент, который соответствует этой задаче наилучшим образом.

Какие требования стоит предъявлять к редактору, кроме того, что он должен соответствовать задаче и быть удобным для вас? Эти требования таковы:

- **контроль над результатами работы** — иногда случается так, что в готовом текстовом файле оказывается что-то, что вы туда не помещали. Или, наоборот, не оказывается чего-то, что, как вы полагали, вы вносили в файл.

  Например, неразрывный пробел ничем не отличается по виду от обычного. Разработчику не всегда удается следить за тем, чтобы обычный пробел не оказывался там, где должен быть неразрывный, и наоборот.

  Некоторые редакторы могут переключаться в режим отображения специальных символов, в котором вместо всех особых знаков на экране показываются другие, ясно различимые символы, например:

Специальный символ	Отображаемый символ
Перевод строки	¶
Пробел (обычный)	•
Пробел (неразрывный)	◦
Табуляция	→

Примечание

Нужно ясно понимать, что на самом деле этих символов в тексте нет, это «фальшивые» знаки, которые текстовый редактор отображает *вместо* настоящих символов.

Однако режим отображения специальных символов тоже имеет свои недостатки. Например, если в документе присутствует настоящий символ «¶» или «→» — вы снова не сможете по внешнему виду отличить подлинный символ от перевода строки или табуляции;

Дополнительно

Вот еще одна иллюстрация к проблеме контроля над результатами работы. Если в стандартном Блокноте сохранить файл в кодировке UTF-8, то Блокнот поместит в начало файла специальный код EF BB BF (в шестнадцатеричном представлении). Это код в кодировке UTF-8, в Unicode ему соответствует служебный символ U+FEFF (Byte Order Mark, BOM). Встретив такой символ в начале файла, обрабатывающая программа должна догадаться, что она имеет дело с такой кодировкой Unicode, в которой старший байт следует перед младшим.

Однако для файлов в UTF-8 эти сложности не нужны, так как проблемы с порядком байтов в этой кодировке не существует. Зато присутствие специального символа в начале файла может привести к неприятностям. Например, если такой символ находится в начале веб-страницы, то некоторые старые браузеры (например, Netscape 4) попытаются его отобразить, из-за чего внешний вид страницы исказится.

Вывод из всего этого следующий: если вы создаете страницы в кодировке UTF-8, убедитесь, что ваш редактор не вставляет в начало файла «лишних» символов. Это можно сделать, например, сохранив файл в указанном редакторе, а потом просмотрев его в каком-нибудь другом редакторе, не поддерживающем кодировок Unicode. Если в начале файла вы увидите «мусор» — значит, ваш редактор «перестарался». Попробуйте изменить его настройки, а если это невозможно — смените редактор.

- **облегчение механической работы** — практически все текстовые редакторы умеют выполнять вместо пользователя стандартные механические действия. Например, если всюду в документе требуется заменить словосочетание «Василий Пупкин» на «Геннадий Уткин» — вам не нужно просматривать документ и вручную перепечатывать имя. В редакторе есть стандартная функция замены, которая сделает это за вас.

  Однако бывают более сложные случаи, в которых проявляется разница между редакторами. Предположим, вам требуется заменить одно имя на другое с учетом всех его форм, то есть словосочетание «Василию Пупкину» — на «Геннадию Уткину», а «Пупкин В. Э.» — на «Уткин Г. Ю». Некоторые редакторы избавляют вас от механической работы даже в таких

случаях (например, с помощью так называемых *регулярных выражений*), а в других редакторах вам ничего не остается, кроме как менять все имена вручную;

Дополнительно

Типичная механическая работа — ввод специальных символов (например, длинного и короткого тире, различных кавычек и т. д.). В большинстве операционных систем существует специальная программа (например, в Windows — Таблица символов), с помощью которой вы можете выбирать нужные символы и копировать их в буфер обмена, чтобы потом вставить в документ.

Однако такая работа со специальными символами малоэффективна. Например, в предыдущем абзаце автор вставил 2 длинных тире и 12 неразрывных пробелов. Если бы за каждым из этих символов пришлось обращаться к Таблице символов — на подготовку книги ушло бы несколько десятилетий.

Многие текстовые редакторы предусматривают специальные сочетания клавиш для вставки в документ особых символов. Например, в Microsoft Word длинному тире соответствует сочетание клавиш <Ctrl>+<Alt>+<->, а неразрывному пробелу — <Ctrl>+<Shift>+<Пробел>. Часто пользователь может сам задавать сочетания клавиш для тех или иных символов.

Если вам приходится много раз подряд выполнять одно и то же действие или последовательность действий — значит, вы имеете дело с типичной механической работой и вам рекомендуется выяснить, не может ли ваш текстовый редактор ее облегчить;

- **удобство настройки и вспомогательные средства** — многие редакторы позволяют настраивать свой интерфейс таким образом, чтобы он лучше соответствовал вашим индивидуальным предпочтениям и выполняемой задаче. Например, практически в любом редакторе можно указать параметры шрифта, с которыми вам удобнее просматривать и редактировать текст.

  Кроме того, редакторы часто предоставляют полезные вспомогательные средства: например, в строке состояния может отображаться номер текущей строки и позиции в этой строке, а также кодировка файла.

Примечание

Зачем может потребоваться номер текущей строки и позиции? Например, если вы проверяете документ HTML с помощью валидатора, он может сообщить о том, что обнаружена ошибка в строке 47, в позиции 134. Разве вам доставит удовольствие отсчитывать по порядку сперва 47 строк с начала документа, а потом — 134 символа в этой строке?

Свойства самого текстового редактора можно разделить на две группы:

- **возможности редактора** — это, фактически, результаты, которые можно получить с его помощью. Например, редактор может уметь или не уметь сохранять файлы в кодировке UTF-8, восстанавливать поврежденный текст и т. д.;

Дополнительно

Некоторые редакторы позволяют «оптимизировать» код HTML, избавляя его от всех необязательных пробелов и переводов строк. Это приводит к некоторому уменьшению размера файла, но делает его код совершенно нечитаемым:

```
<?xml version="1.0" encoding="UTF-8"?><!DOCTYPE html PUBLIC "-
//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd"><html
xmlns="http://www.w3.org/1999/xhtml" xml:lang="en-US" lang="en-
US"><head profile="http://www.w3.org/2000/08/w3c-synd/#"><meta
name="generator" content="HTML Tidy for Mac OS, see www.w3.org"
/><meta http-equiv="Content-Type" content="text/html; char-
set=UTF-8" /><meta name="keywords" content="W3C, World Wide
Web, Web, WWW, Consortium, computer, access, accessibility, se-
mantic, worldwide, W3, HTML, XML, standard, language, technolo-
gy, link, CSS, RDF, XSL, Berners-Lee, Berners, Lee, style
sheet, cascading, schema, XHTML, mobile, SVG, PNG, PICS, DOM,
SMIL, MathML, markup, Amaya, Jigsaw, free, open source, soft-
ware" /><meta name="description" content="W3C's nearly 400 mem-
ber organizations lead the World Wide Web to its full poten-
tial. Founded by Tim Berners-Lee, the Web's inventor. The W3C
Web site hosts specifications, guidelines, software and tools.
Public participation is welcome. W3C supports universal access,
the semantic Web, trust, interoperability, evolvability, decen-
tralization, and cooler multimedia." /><title>World Wide Web
Consortium</title><link id="meta" rel="rel:meta"
href="http://www.w3.org/2000/08/w3c-synd/#meta" /><link
rel="meta" href="/Overview-about.rdf" /><link rel="stylesheet"
type="text/css" href="/StyleSheets/home.css" /><link
rel="bookmark" href="#technologies" title="Technologies |"
/><link rel="bookmark" href="#news" title="News |" /><link
rel="bookmark" href="#search" title="Search |" /><link
rel="contents" href="#contents" title="Contents |" /><link
rel="bookmark" href="#systems" title="Systems |" /><link
rel="bookmark" href="#donors" title="Donors |" /><link
rel="bookmark" href="#footnotes" title="Footnotes |" /><link
rel="alternate" type="application/rss+xml" title="RSS"
href="http://www.w3.org/2000/08/w3c-synd/home.rss" /><style
type="text/css"><!-- html, body { background: #fff; color:
#000; } /* body for Windows IE3 */ --></style></head><body><h1
id="logo"><img alt="The World Wide Web Consortium (W3C)"
height="48" width="315" src="/Icons/w3c_main" /></h1><h2
id="slogan">Leading the Web to Its Full Potential...</h2>
```

Чтобы изменить что-то в таком файле, его придется снова «развернуть» — а многие редакторы этого не умеют, даже те, которые смогли его так «свернуть»!

- **интерфейс редактора** — даже если два текстовых редактора обладают одинаковыми возможностями, может оказаться так, что осуществить эти возможности в одном из них проще и удобнее.

Примечание

Например, встроенный редактор файлового менеджера FAR Manager может работать с файлами в кодировках Unicode, но для этого к нему надо подключить и определенным образом настроить специальные расширения (plug-ins). И даже после этого остаются трудности — например, с отображением и вводом специальных символов, с сохранением файлов. Поэтому, если необходимо редактировать файл в кодировке UTF-8 (или какой-либо другой кодировке Unicode), то не имеет смысла пользоваться этим редактором.

Для того чтобы оценить интерфейс редактора, можно применять вполне формальные критерии. Например, подсчитать, сколько действий нужно выполнить, чтобы получить один и тот же результат в разных редакторах. Если количество действий в одном из редакторов заметно меньше, чем в остальных, — то, вполне возможно, следует счесть его интерфейс более эффективным.

Но, кроме формальных критериев, часто имеет смысл задействовать и неформальные — например, посмотреть, какой из нескольких редакторов кажется вам приятнее и комфортнее. В самом деле, для многих пользователей важно не только работать эффективно, но и получать от работы удовольствие или по крайней мере чувствовать себя комфортно.

Примечание

Если пользователю некомфортно работать, если его что-то нервирует, раздражает, — скорее всего, это отразится на качестве работы. Если пользователь совершает пять действий там, где можно было ограничиться двумя, — это не всегда плохо. Может быть, если заставить пользователя переключиться в «более эффективный» режим работы — он будет испытывать дискомфорт, и в результате его работа на самом деле станет менее эффективной.

Дополнительно

Чаще всего, чтобы уменьшить количество действий, требующихся на ту или иную задачу, пользователю требуется больше нагружать свою память. Например, чтобы с помощью Microsoft Word вставить в документ короткое тире, можно воспользоваться меню **Вставка**, выбрать в нем команду **Символ**, отметить вкладку **Специальные знаки** и щелкнуть там по короткому тире. Того же результата можно добиться короче — нажать сочетание клавиш <Ctrl>+<->. Но ведь для этого нужно помнить об этом сочетании! Причем короткое тире — это не единственный специальный символ; при наборе текста мы употребляем

около двух десятков таких символов. Пользователю не всегда удается помнить сочетания клавиш, соответствующие каждому символу. А использование меню **Вставка** — это универсальный способ, не требующий феноменальной памяти. Поэтому он иногда оказывается эффективнее, чем сочетание клавиш, которое кажется оптимальным с формальной точки зрения.

Вам рекомендуется приобрести опыт работы с разными текстовыми редакторами, а также поинтересоваться у коллег и знакомых, какими редакторами они пользуются и почему. Тогда вы научитесь подходить к выбору текстового редактора осознанно и станете безошибочно выбирать для каждой задачи именно тот редактор, который лучше всего соответствует ей и является наиболее удобным для вас.

4.2. От частного к общему

Свойства и возможности текстовых редакторов

> Вчера на советско-китайской границе был обстрелян мирный комбайн. Экипаж комбайна развернул машину и дал ответный залп из всех орудий, в результате чего противник был уничтожен. Председатель колхоза генерал Петренко предупреждает, что в случае повторения подобных провокаций на поля будут выведены сенокосилки с вертикальным взлетом.
>
> *Анекдот*

4.2.1. Текстовый и графический режимы

> Он узнал, что Человек не всегда был городским жителем и что с тех пор, как машины освободили его от черной работы, наступило вечное соперничество двух разных типов цивилизации.
>
> *А. Кларк. Город и звезды*

Как и любая другая программа, каждый текстовый редактор работает в каком-либо одном из двух режимов:

- ❒ текстовый;
- ❒ графический.

Большинство широко распространенных редакторов предназначены для графического режима; они работают в контексте графического интерфейса операционной системы, будь то интерфейс Windows или графическая оболочка KDE для Linux.

Программа, работающая в графическом режиме, обычно имеет все стандартные средства интерфейса. К ним относятся верхняя строка меню, панель инструментов, строка состояния, полоса прокрутки и другие обычные компоненты. В такой программе можно по необходимости пользоваться мышью. Если вы привыкли работать с Windows или Linux — скорее всего, подавляющее большинство знакомых вам программ работают именно в графическом режиме.

Но существуют и такие программы, которые предназначены для текстового режима. В первую очередь к ним относится популярный файловый менеджер FAR Manager (рис. 4.1) — наследник еще более популярного когда-то Norton Commander.

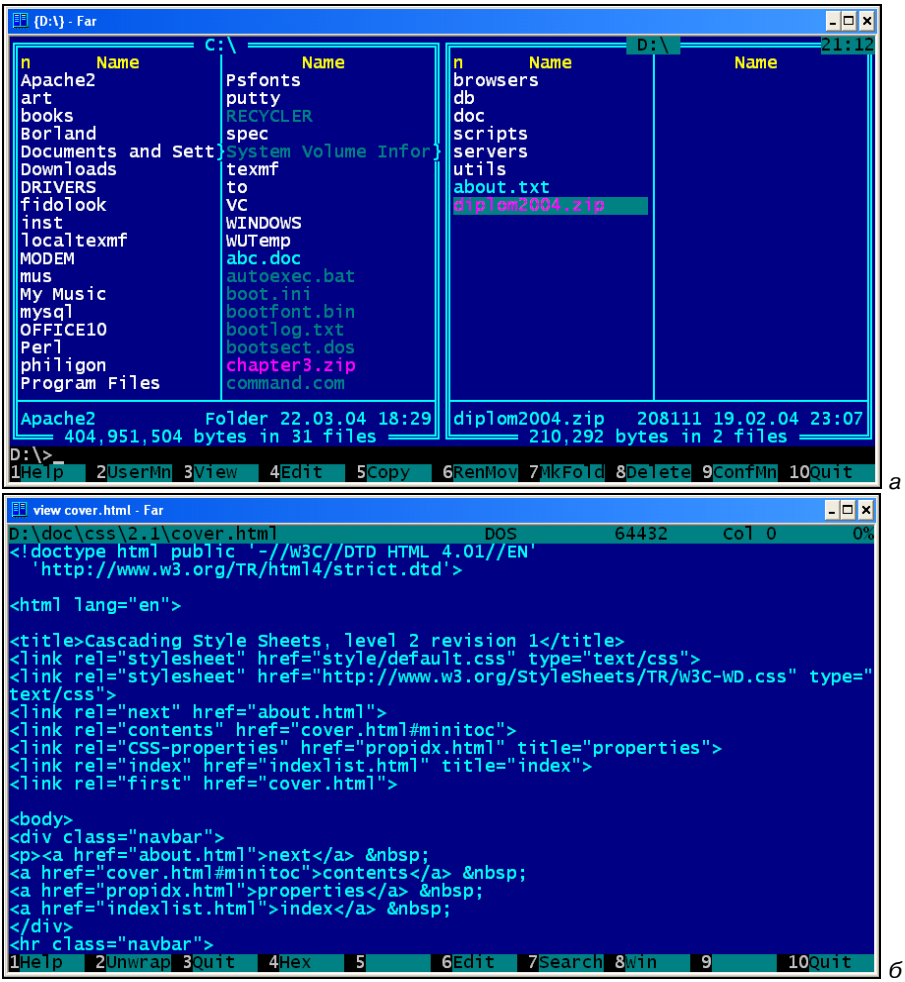


Рис. 4.1. *а* — FAR Manager и *б* — его встроенный текстовый редактор

Окно этой программы совершенно не похоже на то, к чему вы привыкли: нет верхнего меню, панели инструментов и полосы прокрутки; да и применение мыши может привести к неожиданным затруднениям. Все это — следствие текстового режима.

В текстовом режиме окно программы разделено на определенное количество знакомест — например, на рис. 4.1 окно имеет 80 знакомест по горизонтали и 25 по вертикали. Вы не сможете ухватиться мышью за край окна и растянуть его так, как делаете это в других, привычных программах.

Таким образом, всего в окне $80 \times 25 = 2000$ знакомест. Для каждой из таких «клеточек» задаются следующие свойства:

- символ, находящийся на этом знакоместе;
- цвет символа;
- цвет фона.

Дополнительно

Кроме того, символ может мигать, но этот эффект в текстовом режиме уже давно ушел в историю.

Таким образом, в этом режиме в окне программы может отображаться только текст, и ничего больше. Мало того: символы, составляющие этот текст, строго выравниваются по строкам и столбцам, так что буква «Ш» занимает в ширину столько же места, сколько цифра «1».

Единственным доступным эффектом оформления в текстовом режиме является раскрашивание. Даже рамки, которые вы видите в окне FAR Manager, не выведены графически, а составлены из символов псевдографики.

В текстовом режиме вы не можете выделить фрагмент текста другим шрифтом, сделать его полужирным или курсивным. Все знакоместа заполняются символами одной и той же гарнитуры, с одними и теми же характеристиками шрифта. В этом режиме вы не можете просматривать форматированный текст или изображения.

Несмотря на такое несовершенство текстового режима, существуют пользователи, для которых встроенный редактор FAR Manager весьма удобен. С точки зрения этих пользователей, графический режим для редактирования файлов в формате «обычный текст» просто не нужен; кроме того, объединение редактора и файлового менеджера в одной программе имеет свои преимущества.

Примечание

Когда-то персональные компьютеры отображали информацию только в текстовом режиме. Когда появились средства вывода изображений на экран, многие

считали, что «ограниченный» текстовый режим вскоре совершенно отомрет. Однако, как легко убедиться, он применяется до сих пор.

Впрочем, для Windows работа в текстовом режиме — это скорее экзотика. Другое дело — операционные системы класса UNIX, для многих из которых текстовый режим является не только основным, но и единственным. На серверах, маршрутизаторах и других компьютерах служебного назначения под управлением UNIX часто просто не устанавливают никакой графической оболочки. Весь экран представляет собой окно текстового режима, и взаимодействие пользователя с компьютером осуществляется путем ввода команд в командную строку. Квалифицированные пользователи и администраторы UNIX считают такой интерфейс очень удобным.

Даже если вы как разработчик сайта не пользуетесь редактором FAR Manager или чем-то подобным, вам все равно следует знать о том, что такое текстовый режим и командная строка. Зачем могут пригодиться навыки работы в текстовом режиме? Это полезно, например, если потребуется редактировать файл, находящийся на удаленном компьютере (например, один из документов вашего собственного сайта). Конечно, вы можете загрузить этот файл на свой компьютер, внести необходимые изменения и снова передать его на удаленный компьютер, но это не всегда эффективно, так как отнимает больше времени и требует от вас совершить больше действий.

Обычно лучше воспользоваться программой удаленного доступа (той самой, с помощью которой вы совершаете файловые операции на удаленном компьютере) и открыть файл удаленно с помощью текстового редактора, который установлен на том компьютере. А этот редактор почти наверняка окажется в текстовом режиме — например, Vi или Vim. Графическими программами в таких ситуациях почти никто не пользуется.

Дополнительно

При взаимодействии с удаленным компьютером даже в текстовом режиме возникают неприятности. Например, этот компьютер и ваша программа удаленного доступа могут придерживаться разных мнений о том, сколько знакомест по горизонтали и вертикали доступно для вывода; некоторые клавиши могут работать неожиданным образом (например, окажется, что <Backspace> не удаляет символы, а, наоборот, вставляет в текущую строку какой-то «мусор», и т. д.). В графическом режиме проблем еще больше, да и нагрузка на канал связи резко возрастает — поэтому графический режим практически не используется для удаленного взаимодействия.

Примечание

Командная строка, в свою очередь, — также совершенно незаменимое средство при взаимодействии с удаленным компьютером. Если вы захотите совершить

какие-то операции с файлами на вашем веб-сервере (переименовать, скопировать, удалить и т. д.) — в вашем распоряжении не будет ни Проводника, как в Windows, ни, скорее всего, даже такого файлового менеджера, как FAR. Вам придется вводить команды и имена файлов в командную строку.

4.2.2. Курсор

> С какой стороны символа должен появляться курсор: справа или слева? Размещение курсора слева от символа правильно только в том случае, если вы собираетесь сделать в этом месте вставку. Тогда как размещение курсора справа от символа правильно при условии, что вы собираетесь удалить этот символ. По-видимому, компьютер должен знать о вашем намерении перед тем, как правильно разместить курсор.
>
> *Дж. Раскин. Интерфейс*

Как уже известно из *главы 3*, текстовый файл представляет собой последовательность символов. Когда вы открываете файл в текстовом редакторе и работаете с ним, то в каждый момент времени ровно один из этих символов является *текущим*, а место, где он находится, — *текущей позицией*.

Текущая позиция обычно отмечается с помощью *курсора* — специального знака, который не присутствует в файле, но отображается в окне редактора. Чаще всего курсор имеет вид вертикальной черты, которая мигает между символами, или горизонтальной черты, которая мигает под символом.

Курсор	Текущий символ	Пример	Примечания
I	*Справа* от курсора	`Мой дядя с|амых...`	Обычно в графическом режиме
_	*Над* курсором	`Мой дядя с_амых...`	Обычно в текстовом режиме

Примечание

Не путайте курсор и указатель мыши. Начинающим пользователям обычно рекомендуют различать их так: курсор — это то, что перемещается на экране, когда вы нажимаете на стрелки на клавиатуре (<←>,<→>,<↑>,<↓>). А указатель мыши — это то, что перемещается на экране, когда вы двигаете мышь по коврику.

Самая простая и распространенная операция в текстовом редакторе — это смена текущей позиции, то есть перемещение курсора. Начинающие и малоквалифицированные пользователи предпочитают выполнять эту операцию

с помощью мыши, в то время как этот способ крайне неэффективен и неудобен. Чтобы убедиться в этом, перечислим действия, которые совершает пользователь:

- снять одну руку с клавиатуры и перенести ее на мышь;
- прицелиться указателем мыши в нужную позицию на экране (заметим, что для этого пользователю требуется напрягать зрение и очень точно контролировать движения руки);
- щелкнуть мышью;
- снять руку с мыши и перенести ее обратно на клавиатуру.

Таким образом, пользователь вынужден совершать множество действий различной природы, проявляя неординарную меткость и координацию движений. Так как далеко не все пользователи — ворошиловские стрелки, то при манипуляциях с мышью часто происходят ошибки, и приходится прицеливаться заново.

Гораздо удобнее работать с текстом с помощью клавиатуры. Даже если перемещать курсор только с помощью клавиш-стрелочек, это уже исключит всякие упражнения в меткости. Вам потребуется всего лишь нажимать на стрелочки до тех пор, пока курсор не окажется в нужной позиции — а он рано или поздно там обязательно окажется. Согласитесь: для того, чтобы попасть пальцем в клавишу, нужно гораздо меньше усилий, чем для того, чтобы попасть указателем мыши точно между двумя соседними символами в тексте!

Примечание

Конечно, перемещение по тексту с помощью одних только клавиш-стрелочек может отнять слишком много времени. Но если задействовать другие специальные клавиши и сочетания клавиш, о которых рассказывается далее в этой главе, можно добиться практически мгновенного (и при этом безошибочного!) перемещения курсора из любой позиции в любую другую.

Дополнительно

Существует и теоретическое обоснование тому, почему с помощью клавиатуры работать с текстом удобнее, чем с помощью мыши. Как было сказано в *главе 1*, текст обладает свойством дискретности. Клавиатура — также дискретное устройство; она состоит из конечного набора клавиш, каждая из которых в любой момент может быть нажата или отпущена. А мышь — не дискретное устройство; движения мыши по коврику и ее указателя по экрану воспринимаются пользователем как непрерывные. Имея устройство непрерывной природы, обычно трудно работать с дискретными объектами, и наоборот.

Зато для редактирования непрерывных объектов — например, графических файлов — мышь подходит как нельзя лучше.

В самом деле, представьте, что вам требуется редактировать рисунок в каком-нибудь растровом редакторе (например, в Adobe Photoshop), но в вашем распоряжении только клавиатура, и нет мыши. Обычно в растровых редакторах есть «горячие клавиши» и другие средства, позволяющие выполнять любые действия с помощью одной клавиатуры. Но вряд ли ваша работа при этом будет эффективной и доставит вам удовольствие.

То же самое происходит, если пытаться работать с текстом с помощью мыши. Начинающие пользователи, к сожалению, обычно не понимают, насколько при этом снижается эффективность и удобство их работы.

Внимание!

Старайтесь не пользоваться мышью, набирая и обрабатывая текст в текстовом редакторе. Работать с клавиатурой гораздо эффективнее; кроме того, вы сами убедитесь, что это удобнее и комфортнее для вас.

В большинстве текстовых редакторов работают стандартные клавиши и сочетания клавиш для перемещения курсора (табл. 4.1).

Таблица 4.1. *Перемещение курсора*

На один символ:			
<←> влево	<→> вправо	<↑> вверх	<↓> вниз
По текущей строке:			
<Home> в начало строки		<End> в конец строки	
На одно окно:			
<Page Up> вверх	Эти клавиши перемещают курсор на столько строк, сколько их умещается в окне при его текущих размерах. Например, если курсор установлен в самой первой строке окна, то клавиша <Page Down> перенесет его в строку, которая следует за самой последней строкой окна и сейчас не видна на экране		
<Page Down> вниз			
На одно слово:			
<Ctrl>+<→> вправо		<Ctrl>+<←> влево	
В начало и конец документа			
<Ctrl>+<Home> в самое начало документа		<Ctrl>+<End> в самый конец документа	

Обратите особое внимание на четыре последние сочетания клавиш. Иногда они бывают совершенно незаменимы. В длинных документах перемещение в начало или в конец с помощью полосы прокрутки очень утомительно, а сочетания клавиш позволяют выполнить его одним движением.

Примечание

Для того, чтобы быстро установить курсор в нужную позицию в тексте, можно пользоваться следующим алгоритмом:

1. Если нужная позиция находится в том же слове, что и текущая, — нажать <→> или <←> столько раз, сколько требуется.
2. Если нужная позиция находится в той же строке, что и текущая, —
   - с помощью <Ctrl>+<→> или <Ctrl>+<←> добраться до того слова, в котором находится нужная позиция;
   - перейти к п. 1.
3. Если нужная позиция находится в том же окне, что и текущая (то есть в настоящий момент видна на экране), —
   - с помощью <↑> или <↓> добраться до той строки, в которой находится нужная позиция;
   - перейти к п. 2.
4. Если нужная позиция находится за пределами текущего окна —
   - с помощью <Page Up> или <Page Down> добиться того, чтобы нужная позиция оказалась в пределах текущего экрана;
   - перейти к п. 3.

Этот алгоритм может показаться излишне строгим и формальным, однако, начав применять его, вы вскоре будете пользоваться им автоматически, совершенно не задумываясь.

В некоторых редакторах работают и другие сочетания клавиш: например, <Ctrl>+<↑> обеспечивает переход к началу текущего абзаца, а <Ctrl>+<↓> — к его концу. В табл. 4.1 перечислены только наиболее универсальные сочетания клавиш. О том, какие еще сочетания действуют в том или ином редакторе, вы можете прочесть в документации к нему.

Дополнительно

В текстовых редакторах, происходящих из ОС UNIX, для перемещения курсора можно применять те же клавиши, что для набора символов. Например, в Vim и некоторых других программах клавиши <H>, <J>, <K> и <L> в определенном режиме работают так же, как стрелочки. Это связано с тем, что когда-то на клавиатурах вовсе не было стрелочек, да и теперь они не всегда корректно работают при удаленном соединении. С буквенными клавишами, история которых еще древнее, чем история компьютеров, таких проблем гораздо меньше.

4.2.3. Набор и редактирование текста

— С какой скоростью ты набираешь текст?
— Тысяча знаков в минуту! Правда, такая ерунда получается...

Анекдот

На компьютере, в отличие от пишущей машинки, можно набирать текст на многих языках. Однако для ввода букв разных языков применяются одни и те же 30—35 клавиш основной клавиатуры. В каждый момент клавиатура «разговаривает» только на одном языке. Чтобы сменить язык, нужно воспользоваться специальным сочетанием клавиш.

На большинстве персональных компьютеров в нашей стране установлены два языка ввода с клавиатуры — русский и английский. Переключение с одного языка на другой выполняется сочетанием клавиш <Ctrl>+<Shift> или <Alt>+<Shift>. (Пользователь сам может выбрать, какое сочетание клавиш будет отвечать за смену языка.)

Примечание

Как же лучше — <Ctrl>+<Shift> или <Alt>+<Shift>? Вам следует проверить это на собственном опыте. Попробуйте некоторое время пользоваться одним из этих сочетаний, а затем другим. В конце концов остановитесь на том, которое окажется удобнее именно для вас.

Автор использует <Ctrl>+<Shift> — прежде всего потому, что две этих клавиши чуть ближе друг к другу, чем <Alt> и <Shift>.

Дополнительно

Впрочем, сочетание клавиш <Ctrl>+<Shift> иногда может «конфликтовать» с другими сочетаниями. Например, если нужно выделить целое слово, можно нажать <Ctrl>+<Shift>+<→> — и при этом получить совсем не то, что ожидалось: вместо выделения клавиатура переключится на другой язык, а стрелочка сработает сама по себе. Чтобы этого не происходило, автор в таких случаях нажимает управляющие клавиши в обратной последовательности: сначала <Shift>, потом <Ctrl>.

Чтобы нажать сочетание из двух клавиш, начинающие пользователи обычно стараются ударить по обеим клавишам одновременно. Это испытание ловкости — совершенно напрасное: гораздо удобнее не нажимать, а *отпускать* клавиши одновременно. Сначала нажать и удерживать одну клавишу, затем — другую, а затем отпустить обе. То же самое верно для сочетаний из трех

и четырех клавиш (впрочем, четырехклавишные «аккорды» применяются крайне редко).

Примечание

Знаменитая неприятность, связанная с переключением языка ввода: при переходе на другой язык многие знаки препинания и специальные символы «перескакивают» на другие клавиши (табл. 4.2). Это приводит к ошибкам не только у начинающих, но и у достаточно опытных пользователей. Заметим, что свою «прописку» меняют не только относительно редкие символы (среди которых — точка с запятой, апостроф и кавычки ASCII, вопросительный знак), но и самые распространенные знаки препинания: точка и запятая. При этом гораздо менее употребительные символы вроде круглых скобок, дефиса и знака равенства остаются на своих местах. Право слово, было бы лучше, чтобы наоборот.

Таблица 4.2. *Знаки ASCII на клавиатуре*

Знак	**Вид**	**Язык ввода**	
		русский	**английский**
Точка	.	самая правая клавиша в нижнем ряду	там же, где русская буква «Ю»
Запятая	,	<Shift>+<.>	там же, где русская буква «Б»
Точка с запятой	;	<Shift>+<4>	там же, где русская буква «Ж»
Двоеточие	:	<Shift>+<6>	<Shift>+<;>
Вопросительный знак	?	<Shift>+<7>	самая правая клавиша в нижнем ряду (там же, где точка в русском языке)
Восклицательный знак	!	<Shift>+<1>	
Дефис	-	в верхнем ряду, справа от 0	
Одинарная кавычка ASCII	'	нет	там же, где русская буква «Э»
Двойные кавычки ASCII	"	<Shift>+<2>	<Shift>+<'>
Круглые скобки	( )	<Shift>+<9>, <Shift>+<0>	
Квадратные скобки	[ ]	нет	там же, где русские буквы «Х» и «Ъ»
Фигурные скобки	{ }	нет	<Shift>+<[> , <Shift>+<]>

Таблица 4.2 *(окончание)*

Знак	**Вид**	**Язык ввода**	
		русский	**английский**
Знаки «меньше» и «больше» (они же — угловые скобки в HTML)	< >	нет	там же, где русские буквы «Б» и «Ю»
Коммерческое «at» («собачка»)	@	нет	<Shift>+<2>
Знак процента	%	<Shift>+<5>	
Амперсанд	&	нет	<Shift>+<7>
Решетка	#	нет	<Shift>+<3>
Астериск	*	<Shift>+<8>	

Клавиатура компьютера многое унаследовала от пишущих машинок и других старинных устройств — например, раскладку (расположение букв и знаков на клавишах) и специальную клавишу для переключения регистра букв — <Shift>. Так же, как у пишущих машинок, на клавиатуре компьютера две клавиши <Shift>, расположенные симметрично, а также длинная клавиша для пробела.

Однако пользователей, которые действительно умеют работать с клавиатурой, гораздо меньше, чем во времена пишущих машинок. Умение работать с клавиатурой — это прежде всего набор вслепую. Автор считает, что этим навыком должен овладеть каждый, кто набирает на клавиатуре хотя бы несколько строчек в день. Это еще более важно для разработчиков сайтов — прежде всего для авторов и редакторов текста, кодировщиков HTML и CSS, программистов.

Примечание

После того как автор этой книги обучился слепому десятипальцевому методу, он стал набирать текст в 6–7 раз быстрее, чем раньше. На обучение ушло примерно три недели. Эффект заключается не только в экономии времени, но и в субъективном удобстве: руки двигаются по клавишам автоматически; не требуется специально задумываться о том, как найти на клавиатуре нужный символ. Это похоже на то, как мы разговариваем: мы произносим целое слово, совершенно не думая о том, как собрать его из отдельных звуков. Автор настоятельно рекомендует вам, если вы еще не владеете слепым набором, обучиться этой несложной науке.

Эта книга не научит вас набирать вслепую. Но существуют учебные курсы, специальные компьютерные программы и тренажеры, которые помогут вам овладеть слепым десятипальцевым методом. Для русского языка хорошо подходит «Соло на клавиатуре» (**www.ergosolo.ru**).

Дополнительно

Умение набирать текст вслепую избавляет, в частности, от одной hfcghjcnhfytyyjq ghj,ktvs? rjulf gjkmpjdfntkm... Если во время набора постоянно смотреть на экран, а не на клавиатуру, ничего подобного с вами не случится.

Большинство текстовых редакторов позволяют набирать текст в двух режимах:

- режим вставки;
- режим замены.

Режим вставки обычно действует по умолчанию. В этом режиме символы, набранные на клавиатуре, вставляются в текущую позицию, а последующие символы (если они есть) сдвигаются вперед.

В *режим замены* обычно можно перейти, нажав клавишу <Insert> (она же <Ins>). В этом режиме набранные символы так же вставляются в текущую позицию, но ранее находившиеся там символы стираются.

	Исходный текст и курсор в текущей позиции	**Текст после ввода «не␣»**
Режим вставки	`Мой дядя |самых...`	`Мой дядя не |самых...`
Режим замены	`Мой дядя |самых...`	`Мой дядя не |ых...`

Примечание

В некоторых редакторах в режиме замены курсор обычно меняет внешний вид, превращаясь из черточки в мигающий прямоугольник: ▮. В результате пользователь постоянно видит, в каком режиме находится редактор в данный момент.

Кроме того, текущий режим может обозначаться и другими средствами — например, когда Microsoft Word работает в режиме замены, в строке состояния подсвечивается надпись **ЗАМ**.

Текст приходится не только набирать, но и удалять. Для наиболее простых операций удаления предназначены клавиши <Backspace> и <Delete>.

Примечание

Клавиша <Backspace> часто обозначается символом «←», в результате чего некоторые начинающие пользователи путают ее с обычной стрелочкой. В этой книге клавиша <Backspace> всюду обозначается своим полным именем.

Аналогично, на клавише <Delete> иногда пишут просто «Del». Автор надеется, что у вас не возникнет путаницы.

Внимание!

<Backspace>	удаляет символ, предшествующий текущей позиции
<Delete>	удаляет символ в текущей позиции

Дополнительно

Можно было написать, что клавиша <Backspace> удаляет символ слева от текущей позиции. Но это не всегда так: если вы набираете на языке, где пишут справа налево (например, на иврите или арабском), <Backspace> удаляет символ, стоящий справа от текущей позиции. Поэтому, чтобы формулировка была универсальной, мы говорим, что она удаляет символ, предшествующий текущей позиции.

Некоторые пользователи — к сожалению, не только начинающие — активно применяют только одну из двух клавиш удаления. Например, пользователь привык только к <Backspace> — и тогда, если требуется удалить символ в текущей позиции, он делает это в два приема: сначала клавишей <→> перемещается на одну позицию вперед, а затем нажимает <Backspace>. Вряд ли такую работу можно считать эффективной, особенно если требуется удалить не один, а несколько символов подряд.

Примечание

Вот примерный перевод названий клавиш на русский язык:

<Home>	Домой	<Delete>	Удалить
<End>	В конец	<Page Up>	На страницу вверх
<Insert>	Вставить	<Page Down>	На страницу вниз

Кроме того, «Shift» примерно переводится как «сдвиг» (соответствующая клавиша на пишущей машинке «сдвигает» печатающее устройство), «Ctrl» — очевидное сокращение от «control» (контрольная, управляющая клавиша), а «Alt» происходит от «alter» («изменять»).

Возможно, если вы будете знать смысл слов, написанных на клавишах, это поможет вам не путаться в том, какая из них за что отвечает.

4.2.4. Выделение текста

> И со всей ответственностью я заявляю, что все хитроумные построения П. Сороки и Э. Брауна — это просто легкомысленная чушь, очередная попытка взяться правой рукой за левое ухо через-под левое колено.
>
> *А. Стругацкий, Б. Стругацкий.*
> *Волны гасят ветер*

Выделение текста, как и перемещение курсора, совершенно не стоит выполнять с помощью мыши. Это отнимает еще больше времени и требует поистине акробатической ловкости: сначала указателем мыши нужно попасть точно в ту позицию, откуда вы хотите начать выделение, потом мышь нужно переместить точно по горизонтали (все это время удерживая нажатой ее клавишу), потом — отпустить клавишу точно в той позиции, где выделение должно закончиться. Если требуется выделить несколько символов в середине строки, то такая эквилибристика вряд ли получится с первого раза. Вы обязательно либо «заведете» мышь на соседние строки, либо отпустите ее клавишу преждевременно, либо закончите выделение не там, где требуется.

Даже если вы — опытный престидижитатор, и выделение мышью получается у вас без особого труда, подумайте: а стоит ли каждый раз исполнять этот трюк там, где он вовсе не требуется? Выделять текст с помощью клавиатуры — это быстрее, проще и удобнее.

В большинстве редакторов текст выделяется, если перемещать курсор при нажатой клавише <Shift>, например:

<Shift>+<→>	выделить символ, находящийся в текущей позиции
<Shift>+<←>	выделить символ слева от текущей позиции
<Shift>+<Home>	выделить участок текста от текущей позиции (не включая саму эту позицию) до начала текущей строки
<Shift>+<Ctrl>+<End>	выделить участок текста от текущей позиции (включая саму эту позицию) до конца всего текста

Примечание

Обратите внимание: для выделения используются те же самые сочетания клавиш, что и для перемещения курсора (см. табл. 4.1), так что вам не придется заучивать никаких новых сочетаний. Просто добавляйте к уже известным сочетаниям клавишу <Shift> — и вместе с перемещением курсора у вас получится выделение текста.

Выделенный текст чаще всего помечается так называемой выворoткой: цвет фона и цвет текста меняются местами.

Как выделить целую строку текста? Очень просто:

- поместить курсор в эту строку (*см. алгоритм в разд. 4.2.2*);
- поместить курсор в начало этой строки (клавиша <Home>);
- нажать <Shift>+<End>.

Можно действовать и по-другому: поместить курсор не в начало, а в конец строки (клавиша <End>), а затем нажать <Shift>+<Home>.

Дополнительно

Чтобы выделить строку вместе с завершающим ее символом перевода строки — вместо <Shift>+<End> следует нажать <Shift>+<↓>. Вообще, сочетания клавиш <Shift>+<↓> и <Shift>+<↑> очень удобны, когда надо выделить несколько строк подряд (соответственно вниз или вверх от текущей строки).

Когда вы впервые читаете описание этих действий, вам может показаться, что работа с клавиатурой — это какое-то хитрое умение, требующее запоминания огромного количества сочетаний клавиш. Вовсе нет: многие пользователи текстовых редакторов совершают все это автоматически. Наоборот, автору потребовалось специально задуматься, чтобы проанализировать и описать принципы эффективной работы с клавиатурой, в то время как применение этих принципов на практике происходит совершенно интуитивно.

Вам достаточно выполнить любое из клавиатурных действий три-четыре раза — и в дальнейшем вы начнете пользоваться им не задумываясь.

Примечание

При выделении текста текущая позиция также меняется, так как вы перемещаете курсор.

Дополнительно

Выделенный участок текста можно расширить или уменьшить. Например, если вы один или несколько раз нажмете сочетание клавиш <Shift>+<→> или <Shift>+<←>, то к выделению добавится несколько новых символов (или убавится несколько символов, в зависимости от того, в начале или в конце выделенного фрагмента находится курсор).

Таким образом вы можете «наращивать» или «убавлять» только конец выделенного фрагмента; начало выделения сдвинуть уже нельзя. Но обратите внимание, что с помощью мыши нельзя сделать даже этого. Если вы по ошибке

выделили всего на один символ меньше или больше, чем нужно, — с мышью вам придется отменять выделение и выделять фрагмент заново. А на клавиатуре достаточно нажать одно сочетание клавиш, чтобы исправить ошибку.

Как снять выделение с участка текста? Проще всего для этого нажать клавиши <→> или <←> — и текст перестанет быть выделенным. Вообще, для этого можно просто переместить курсор куда угодно.

Если же, выделив участок текста, нажать какую-нибудь «символьную» клавишу (например, букву) — в большинстве текстовых редакторов выделенный участок *удалится*, и на его месте окажется введенный символ. Будьте осторожны в таких ситуациях. Иногда это действие может понадобиться специально, но следите за тем, чтобы удаление выделенного фрагмента не произошло у вас случайно.

Если неприятность все-таки произошла — не расстраивайтесь: большинство редакторов позволяют отменить последнее действие. Проще всего сделать это сочетанием клавиш <Ctrl>+<Z>. В редакторах с графическим интерфейсом обычно также есть команда меню **Edit | Undo** (в русскоязычных программах обычно **Правка | Отменить**), которая делает то же самое. Это работает не только для восстановления нечаянно удаленного выделения, но и для отмены любого действия, которое вы совершили непосредственно перед этим.

Внимание!

Запомните сочетание клавиш <Ctrl>+<Z> — отмена последнего действия. Если вы сделали что-то не то, и текст испортился, — немедленно нажмите <Ctrl>+<Z>, чтобы вернуть вашу работу в предыдущее состояние.

Дополнительно

Некоторые редакторы позволяют отменить не только самое последнее действие, но и предыдущие. Иногда таким образом можно последовательно отменить вообще все действия, которые вы совершали с документом!

Чтобы выделить весь текст сразу, нажмите <Ctrl>+<A>. Это работает не только в текстовых редакторах, но и в браузерах: с помощью этого сочетания клавиш вы можете выделить сразу всю веб-страницу!

Примечание

В большинстве современных текстовых редакторов нельзя выделить несколько несмежных участков текста одновременно. Возможно, в будущем такая возможность получит более широкое распространение; она уже внедрена в популярные текстовые процессоры — например, в Microsoft Word, начиная с версии 2002.

4.2.5. Буфер обмена

Погрожу лишь плеткой —
Мой народец кроткий
Песни распевает,
Денежки сбирает
В мой большой карман,
В мой большой карман...

А. Н. Толстой. Золотой ключик или Приключения Буратино

Очень часто текст выделяют для того, чтобы поместить его в буфер обмена. *Буфер обмена* («карман», clipboard) — это отдельная область памяти, где можно на время сохранить фрагмент текста или другие данные.

Чтобы поместить участок текста в буфер обмена, его надо сначала выделить, а затем можно воспользоваться одним из следующих сочетаний клавиш:

Сочетание клавиш	Действие	Примечания
<Ctrl>+<C> или <Ctrl>+<Insert>	Копировать (Copy)	Выделенный участок копируется в буфер обмена, продолжая также оставаться в тексте
<Ctrl>+<X> или <Shift>+<Delete>	Вырезать (Cut)	Выделенный участок удаляется из текста и перемещается в буфер обмена

Примечание

Если прежде в буфере обмена уже были какие-то данные, они безвозвратно пропадают оттуда.

Текст, находящийся в буфере обмена, можно извлечь оттуда и вставить в любое место документа. Для этого используются следующие сочетания клавиш:

Сочетание клавиш	Действие	Примечания
<Ctrl>+<V> или <Shift>+<Insert>	Вставить (Paste)	Содержимое буфера обмена помещается в документ, начиная с текущей позиции

При вставке текст не пропадает из буфера обмена, так что одно и то же содержимое буфера можно вставить куда-либо много раз — как в один и тот же документ, так и в разные документы.

Дополнительно

При работе с текстом очень полезны шесть управляющих клавиш:

<Insert>	<Home>	<Page Up>
<Delete>	<End>	<Page Down>

К сожалению, не каждая клавиатура приспособлена для того, чтобы эффективно пользоваться этими клавишами. На некоторых клавиатурах блок управляющих клавиш состоит не из двух, а из трех рядов. Третий ряд занимают клавиши <Print Screen>, <Scroll Lock> и <Pause>, которые используются гораздо реже. Эти клавиши мешают работать: например, вы можете промахнуться и вместо <Insert> нажать на <Print Screen>. Удобнее пользоваться клавиатурой, где клавиши <Print Screen>, <Scroll Lock> и <Pause> расположены в отдельном блоке, не примыкающем непосредственно к шести перечисленным клавишам.

Однако бывают еще менее дружелюбные клавиатуры: на них в непосредственной близости от вышеперечисленных клавиш расположены кнопки <Power> (включить или выключить питание), <Sleep> (переход в спящий режим) и <Wake> (выход из спящего режима). Эти кнопки — настоящий кошмар: стоит случайно промахнуться мимо <Delete> или <End> — и ваш компьютер вдруг полностью отключается!

Приобретайте такие клавиатуры, на которых шесть управляющих клавиш образуют единый блок, и поблизости от этого блока нет ничего постороннего.

В некоторых текстовых редакторах предоставляется несколько буферов, с каждым из которых можно работать независимо от других. Например, в редакторе Vim можно воспользоваться 26 буферами, которые называются *регистрами*. (Такое количество регистров установлено потому, что именем каждого из них является буква латинского алфавита.) Однако «основной» буфер обмена хорош тем, что предоставляется не редактором, а операционной системой, так что через него можно переносить данные не только в пределах редактора, но и из одних программ в другие. Если вы хотите вставить в свой документ текст, скопированный с веб-страницы, — вам придется делать это через «основной» буфер.

Примечание

В некоторых редакторах выделенный участок текста помещается в буфер обмена автоматически, как только закончилась операция выделения. Обычно это удобно, так как вам не требуется совершать дополнительных действий по копированию. Но иногда такое поведение редактора — это совсем не то, что вам нужно. Может быть, вы выделяете текст затем, чтобы удалить его и вставить на его место то, что сейчас находится в буфере. В этом случае автоматическое заполнение буфера вам только помешает.

4.2.6. Символы и шрифты

> Достаточно нажать белую кнопку — и мгновенно окажешься в радужном дворце, где находится Голос.
>
> *Я. Вайсс. Дом в тысячу этажей*

По умолчанию текстовые редакторы отображают текст моноширинным шрифтом — т. е. таким, в котором любой символ занимает площадку одной и той же ширины:

Пропорциональный шрифт	Моноширинный шрифт
aaaaaaaaaaaa	aaaaaaaaaaaa
щщщщщщщщщщщщ	щщщщщщщщщщщщ
!!!!!!!!!!!!	!!!!!!!!!!!!
555555555555	555555555555

Это очень удобно при редактировании документов HTML и таблиц стилей CSS, а также исходного кода компьютерных программ.

Примечание

В операционной системе Windows к стандартным моноширинным шрифтам относятся Courier New, Andale Mono, Lucida Console.

Однако бывают случаи, когда более удобно работать с текстом, набранным не моноширинным шрифтом. Большинство текстовых редакторов позволяют переключить шрифт, которым будут отображаться документы, а также установить определенный цвет текста и фона. Учтите, что все эти изменения никак не скажутся на файлах, которые вы создаете. Файлы как были, так и останутся обычным текстом без всякого оформления. Шрифтовые настройки в текстовом редакторе влияют только на то, как файлы будут отображаться в окне этого редактора.

Примечание

В текстовом редакторе обычно есть команда меню **Формат | Шрифт** (или что-то подобное), с помощью которой вы можете установить тот шрифт, который вам лучше подходит.

Еще одно полезное свойство многих текстовых редакторов — синтаксическая подсветка. Редактор распознает управляющие конструкции HTML,

CSS и языков программирования и отображает их со специальным оформлением (рис. 4.2).

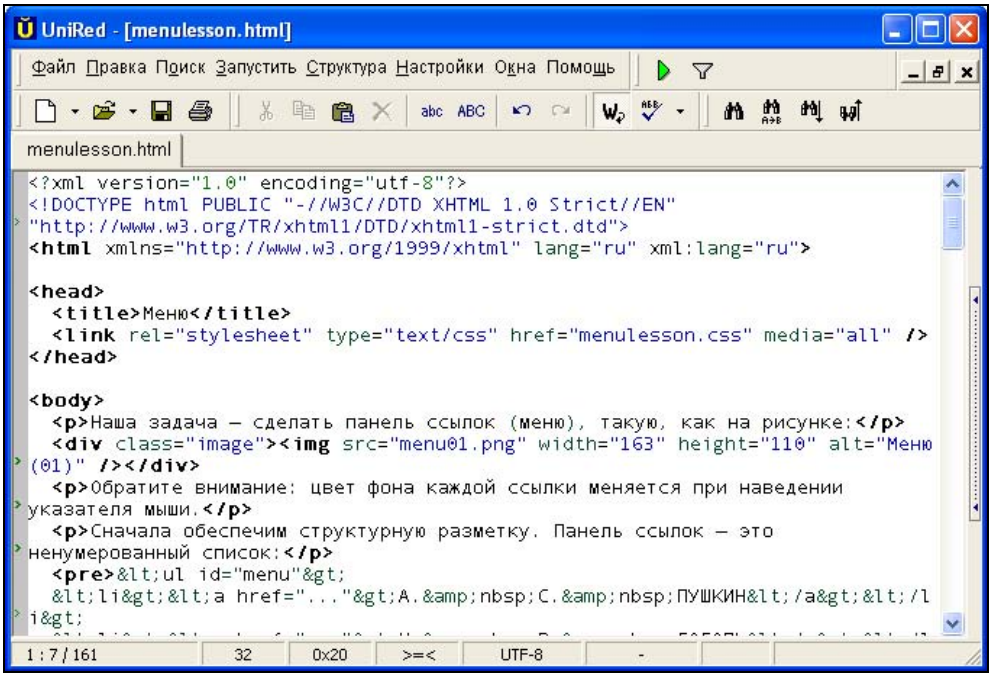


Рис. 4.2. Синтаксическая подсветка в редакторе UniRed: имена и скобки тэгов выделяются полужирным начертанием, имена атрибутов и подстановки символов — зеленым цветом, значения атрибутов — синим цветом

Синтаксическая подсветка делает код более наглядным и позволяет легче в нем ориентироваться. Она также помогает избежать некоторых ошибок и опечаток: например, если вы пропустили закрывающую скобку или кавычку, весь дальнейший текст будет подсвечиваться неправильно, и это укажет вам на ошибку.

Дополнительно

Кроме подсветки, редакторы иногда предлагают и другие средства, облегчающие работу с кодом. Например, некоторые редакторы могут найти тэг, парный к выделенному, подсказать список возможных атрибутов для данного элемента или даже проверить файл на синтаксическую правильность.

При наборе текста могут возникнуть затруднения с вводом символов, отсутствующих на клавиатуре. Прежде всего это типографские знаки — тире, кавычки, многоточие, неразрывный пробел и другие.

Примечание

Прежде чем вставлять в текст такие знаки, убедитесь, что кодировка текста это позволяет. Например, в кодировке Windows-1251 доступны оба тире и кавычки-«елочки», а в KOI8-R эти символы отсутствуют. Некоторые текстовые редакторы позволят вам вставить эти символы и в KOI8, но при следующем открытии файла вместо них отобразятся посторонние значки.

В кодировке UTF-8 доступны абсолютно все символы из Unicode. Если у вас нет каких-либо специальных причин использовать другую кодировку, автор рекомендует при создании веб-страниц применять UTF-8.

Наиболее простой способ вставить «нестандартный» символ — это скопировать его через Таблицу символов операционной системы. Но это — не самый эффективный способ, особенно для знаков, которые требуются вам очень часто. Поэтому многие редакторы предоставляют свои собственные средства для ввода таких символов.

Дополнительно

Например, в редакторе UniRed можно нажать сочетание клавиш <Alt>+<:>, затем набрать десятичный или шестнадцатеричный код нужного символа и нажать <Enter>. Символ появится в документе в текущей позиции.

Самый быстрый способ ввода специальных символов — это использование сочетаний клавиш. Многие редакторы позволяют вам самим назначить эти сочетания.

Примечание

Автор назначил в своих текстовых редакторах следующие сочетания клавиш для ввода специальных символов:

<Ctrl>+<->	—	длинное тире	<Ctrl>+<9>	«	правая кавычка-«елочка»
<Alt>+<->	–	короткое тире	<Ctrl>+<0>	»	левая кавычка-«елочка»
<Ctrl>+<1>	...	многоточие	<Ctrl>+<Пробел>		неразрывный пробел

Это нестандартные сочетания; если вы попробуете применить их в своем редакторе — скорее всего, они не сработают. Однако поинтересуйтесь, нет ли возможности специально настроить ваш редактор таким образом, чтобы эти (или другие удобные вам) сочетания клавиш были связаны с наиболее распространенными типографскими символами.

Если вы сами назначаете сочетания клавиш — следите за тем, чтобы они не совпадали с уже «занятыми» сочетаниями. Например, не стоит связывать какой-либо символ с сочетанием <Ctrl>+<Z>, не говоря уж об <Alt>+<F4>.

Глядя на текст в окне редактора, вы не всегда можете отличить один символ от другого. Например, короткое тире нетрудно перепутать с длинным, а латинскую букву «C» («це») с русской буквой «С» («эс»). (В последнем случае ошибки часто происходят еще и потому, что обе эти буквы находятся на одной и той же клавише.)

Многие редакторы предоставляют средства информации о символах. Например, в редакторе UniRed в строке состояния отображаются два кода текущего символа — в десятичной и в шестнадцатеричной системе (в таблице Unicode). Благодаря этой подсказке вы легко отличите даже обычный пробел от неразрывного — если в текущей позиции обычный пробел, то мы увидим десятичный код 32, а если неразрывный — код 160.

Дополнительно

Вам следует заботиться о том, чтобы вставляемый символ присутствовал в текущей кодировке, но не обязательно — в текущем шрифте. Например, в шрифте Courier New, которым по умолчанию отображается текст во многих редакторах, нет музыкальных знаков бемоля (♭) и диеза (♯). Однако, если вы используете кодировку UTF-8 (или другую, где есть эти символы) — вы можете спокойно пользоваться ими. Сам текстовый файл не связан ни с каким конкретным шрифтом, поэтому ограничения какого-либо определенного шрифта не играют никакой роли.

Важное свойство редактора — это набор кодировок, которые он поддерживает. Если вы разрабатываете веб-страницы на русском языке, то вам рекомендуется пользоваться редактором, поддерживающим все встречающиеся кодировки русского языка (в первую очередь Windows-1251 и KOI8-R), а также UTF-8. К сожалению, немногие популярные текстовые редакторы могут похвастаться полноценной поддержкой UTF-8.

Примечание

Например, редактор MiB Editor не умеет работать с файлами в кодировках Unicode так же, как в расширенных кодировках ASCII. Он может только экспортировать текущий файл в UTF-8 или преобразовать его из этой кодировки в другую, но не позволяет открыть и редактировать файл непосредственно в UTF-8.

Внимание!

Убедитесь, что ваш текстовый редактор поддерживает по крайней мере следующие три кодировки:

- Windows-1251;
- KOI8-R;
- UTF-8.

Во многих текстовых редакторах в строке состояния или в другом месте отображается кодировка текущего файла. Редакторы также дают возможность сменить эту кодировку. Здесь главное — не перепутать: одно дело — сменить кодировку, в которой просматривается файл, и совсем другое — преобразовать сам файл в другую кодировку.

- **Сменить кодировку просмотра.** Эта операция обычно нужна, если файл открылся не в той кодировке, в которой был создан, и вы видите на экране «бНОПНЯ» и «оПХБЕР». Переключение кодировки никак не отражается на самом файле и не изменяет его.

Дополнительно

Многие редакторы умеют распознавать кодировку автоматически. Другие редакторы по умолчанию открывают все файлы в одной и той же кодировке, однако предоставляют пользователю возможность указать при открытии файла другую кодировку.

Кроме распознавания кодировки, некоторые редакторы могут восстанавливать текст, который был некорректно перекодирован. Например, вот какой текст получится, если предыдущий абзац по ошибке *дважды* перекодировать из Windows-1251 в KOI8-R:

```
Клмжуе оедчипмоь якевп очънмцлчачпщ имдуомаия чапмкчпубеъиу. Дояжуе
оедчипмоь нм якмйбчлув мпиоьачвп аъе рчхйь а мдлмх у пмх те имдуомаие,
мдлчим ноедмъпчайэвп нмйщцмачпейв амцкмтлмъпщ яичцчпщ ноу мпиоьпуу
рчхйч дояжяв имдуомаия.
```

Есть редакторы, которые справляются с таким текстом и возвращают его в нормальный вид.

- **Преобразовать файл в другую кодировку.** Эта операция изменяет содержимое самого файла, а не режим его просмотра. Иногда такие операции оказываются необратимыми — например, если вы преобразуете файл из UTF-8 в Windows-1251, то у вас могут пропасть многие специальные символы, которые при обратном преобразовании не вернутся.

4.2.7. Прокрутка и переводы строк

А растаман кассету до конца прокрутил и говорит: а клёво бы сейчас на гитарке поиграть вобще.

Д. Гайдук. Священная история

Если строка текста слишком длинная и не умещается в окне редактора — большинство редакторов предоставляют две возможности:

- позволить строке выйти за правый край окна — обычно этот режим применяется по умолчанию. В этом случае на экране видна только та часть строки, которая уместилась, а остальная строка доступна, если воспользоваться полосой горизонтальной прокрутки;
- «завернуть» строку, расположив ее в окне на нескольких строчках — переключиться в этот режим обычно можно с помощью настройки, которая называется **Переносить по словам** (Word Wrap) или похожим образом.

Дополнительно

Здесь снова всплывает лингвистическая проблема, о которой упоминалось в *главе 2*. Под словом «строка» понимается и участок текста между двумя символами перевода строки[1], и линия текста, видимая в окне редактора. С этой точки зрения, действительно, получается, что одна строка располагается на нескольких строках! Автор надеется, что такая неизбежная омонимия все-таки не вызовет у вас затруднений.

Примечание

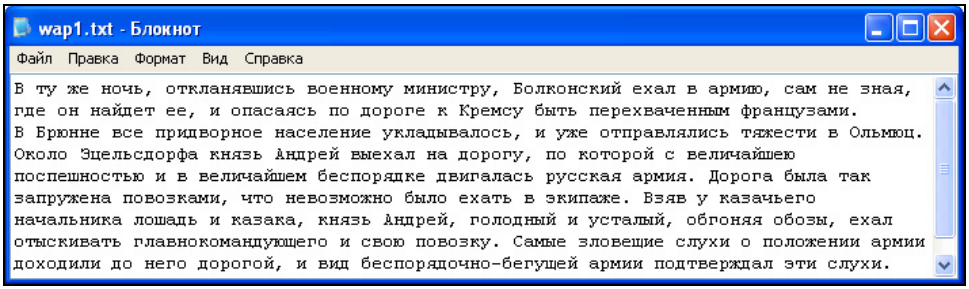
При переносе по словам в самом текстовом файле ничего не меняется; этот режим влияет только на просмотр этого файла в окне редактора.

Если включен режим «Переносить по словам», это может привести к затруднениям при просмотре: вы не всегда сможете понять, где настоящий перевод строки, а где — перенос, сделанный редактором. В некоторых редакторах применяются специальные средства для различения «настоящих» и «фальшивых» переводов строк (рис. 4.3).

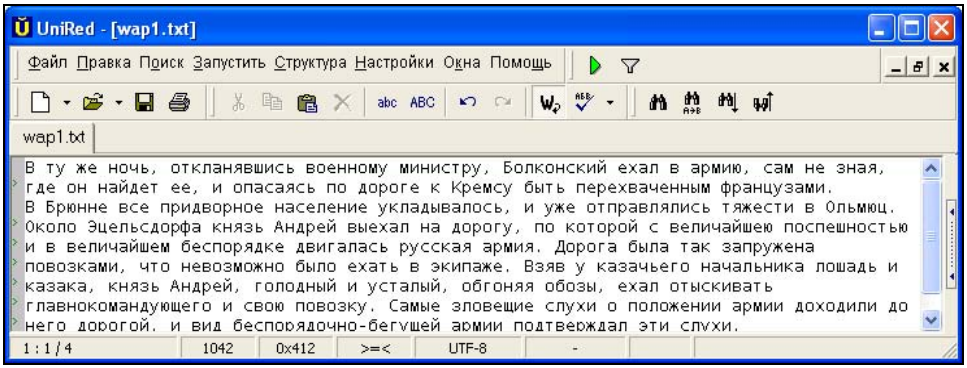
Также можно изменить ширину окна редактора и посмотреть, какие переводы строк останутся на месте — они и будут «настоящими». Старайтесь всегда следить за тем, где в вашем текстовом файле на самом деле переводы строк.

---

[1] Участок текста между двумя символами перевода строки в компьютерной литературе традиционно называется абзацем. Никакой «лингвистической проблемы» по нашему мнению здесь нет. — *Ред.*



а



б

Рис. 4.3. *а* — в Блокноте непонятно, есть ли настоящий перевод строки перед строками 3 и 4; *б* — в Unired'е фальшивые переводы строк отмечены значком «>» слева от текста; понятно, что в строке 3 действительно начинается новый абзац, а строка 4 — продолжение этого абзаца

Примечание

Как вставить в файл пустую строку? Очень просто:

- поместить курсор в начало той строки, *перед* которой нужно вставить пустую (не забывайте про клавишу <Home>);
- нажать <Enter>.

Можно поступить иначе:

поместить курсор в конец той строки, *после* которой нужно вставить пустую (не забывайте про клавишу <End>);

нажать <Enter>.

Дополнительно

Во многих редакторах есть функция автоматического отступа (auto-indent): если нажать <Enter> в конце строки, имеющей в начале один или несколько пробелов, то начало следующей строки автоматически заполнится таким же количеством

пробелов. Иногда эта функция очень удобна при редактировании кода (например, HTML или CSS), когда необходимо, чтобы некоторые строки располагались точно друг под другом.

4.2.8. Операции с файлами

> По своему опыту мы знаем, что имена файлов работают не так, как этого стоило бы ожидать, — они мешают, когда нужно что-то сохранить, и бывают бесполезными, когда нужно что-то найти. Если говорить более конкретно, файловые имена становятся назойливыми, когда вы собираетесь что-то сохранить, так как вам приходится останавливаться в середине самого процесса сохранения, чтобы придумать имя для файла.
>
> *Дж. Раскин. Интерфейс*

Все текстовые редакторы поддерживают стандартные операции с файлами. Почти каждой из этих операций соответствует стандартное сочетание клавиш (табл. 4.3).

Таблица 4.3. *Стандартные сочетания клавиш для операций с файлами*

Операция	Наиболее употребительное англоязычное наименование	Сочетание клавиш
Создать новый файл	New	<Ctrl>+<N>
Открыть существующий файл	Open	<Ctrl>+<O>
Сохранить файл	Save	<Ctrl>+<S>, <F2>
Сохранить файл под другим именем	Save As	<Ctrl>+<Shift>+<S> (только в некоторых редакторах)
Закрыть файл	Close	

Рассмотрим некоторые подробности каждой из этих операций:

- **создать новый файл** — редактор обычно открывает новое, пустое окно; для вновь создаваемого файла устанавливается кодировка по умолчанию;

Дополнительно

Многие редакторы позволяют создавать новые файлы на основе заданных шаблонов. Например, можно заготовить шаблон документа HTML (*см. главу 5*), и настроить редактор таким образом, чтобы этот шаблон загружался при создании каждого нового документа HTML. При этом не потребуется каждый раз вспоминать или добывать откуда-то пролог и другие стандартные компоненты документа.

- **открыть существующий файл** — обычно редактор предлагает диалоговое окно, в котором можно не только указать путь к нужному файлу, но и задать некоторые его параметры — например, кодировку, в которой следует открывать этот файл.

  Некоторые редакторы «запоминают» текущую позицию в файле. Когда этот файл открывается, редактор сразу переводит курсор в то же место, где он был при последнем закрытии файла. Таким образом, пользователь сможет продолжить работу из того же состояния, на котором остановился в прошлый раз;
- **сохранить файл** — начинающие пользователи обычно не понимают разницы между командами **Сохранить** (Save) и **Сохранить как...** (Save As...). Разница эта простая:
  - если файл сохраняется впервые, то обе эти команды приводят к одному и тому же эффекту, и можно использовать любую из них;
  - если файл уже был сохранен, то для его повторного сохранения следует использовать именно **Сохранить**, а не **Сохранить как...**;
  - команда **Сохранить как...** используется в том случае, если требуется сменить имя файла. (Точнее, сохранить текущий файл не под тем именем, которое было раньше, а под другим. Старый файл при этом не пропадает.)

  При первом сохранении файла и при вызове команды **Сохранить как...** предоставляется диалоговое окно. В нем, как и при открытии файла, можно указать имя сохраняемого файла и каталог, в котором он будет расположен, а также другие параметры — в частности, все ту же кодировку.

Внимание!

Сохраняя текстовые файлы в Windows, обязательно указывайте не только имя, но и расширение файла: txt, html, css и т. д.

Примечание

Во многих редакторах есть специальный индикатор, показывающий статус сохранения файла. Этот индикатор активен, если в файл были внесены изменения, и после этого файл еще не был сохранен (рис. 4.4).

```
edit lesson.html - Far
C:\...rn\html\lesson.html *          Win    Line       11/161   Col 25          32
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"   "http://www.w3.org/TR
<html xmlns="http://www.w3.org/1999/xhtml" lang="ru" xml:lang="ru">

<head>
  <title>Меню</title>
  <link rel="stylesheet" type="text/css" href="menulesson.css" media="all" />
</head>

<body>
  <p>Наша задача &#8212;_сделать панель ссылок (меню), такую, как на рисунке:</p
  <div class="image"><img src="menu01.png" width="163" height="110" alt="Меню (О
  <p>Обратите внимание: цвет фона каждой ссылки меняется при наведении указателя
  <p>Сначала обеспечим структурную разметку. Панель ссылок - это ненумерованный
  <pre>&lt;ul id="menu"&gt;
  &lt;li&gt;&lt;a href="..."&gt;А.&amp;nbsp;С.&amp;nbsp;ПУШКИН&lt;/a&gt;&lt;/li&
  &lt;li&gt;&lt;a href="..."&gt;Н.&amp;nbsp;В.&amp;nbsp;ГОГОЛЬ&lt;/a&gt;&lt;/li&
  &lt;li&gt;&lt;a href="..."&gt;И.&amp;nbsp;С.&amp;nbsp;ТУРГЕНЕВ&lt;/a&gt;&lt;/l
  &lt;li&gt;&lt;a href="..."&gt;Л.&amp;nbsp;Н.&amp;nbsp;ТОЛСТОЙ&lt;/a&gt;&lt;/li
&lt;/ul&gt;</pre>
  <p>Получаем обычный список:</p>
  <ul>
    <li><a href="#">А. С. ПУШКИН</a></li>
1Help  2Save  3  4  5  6View  7Search  8DOS  9  10Quit
```

Рис. 4.4. Обратите внимание на астериск (звездочку) в верхней строке окна встроенного редактора FAR Manager. Эта звездочка появляется, если начать редактировать файл, и пропадает, если файл сохранить. Таким образом, пользователь всегда знает, изменялся ли файл после того, как его в последний раз сохраняли

Стандартное сочетание клавиш для сохранения файла — <Ctrl>+<S>, однако многие редакторы поддерживают также сохранение по нажатию клавиши <F2>. Это более экономно и эффективно.

Внимание!

Привыкайте почаще сохранять вашу работу. У многих пользователей эта привычка становится автоматической: после каждого заметного этапа редактирования они нажимают <Ctrl>+<S>, даже не задумываясь об этом.

Примечание

Многие редакторы выполняют автоматическое сохранение файла через установленные промежутки времени — например, через каждые пять или десять

минут. Это небесполезное свойство, но безоговорочно доверять ему не стоит: лучше вручную сохранять файл, как только завершен сколько-либо важный этап работы. Компьютер не может угадать, когда вы завершили важный этап, и не поможет вам своевременно сохранить работу.

Бывают редакторы, которые создают резервные копии обрабатываемых файлов — например, с расширениями bak или tx~. Не забывайте удалять эти файлы после того, как необходимость в них исчезнет, иначе они превратятся в мусор, затрудняющий вашу работу.

Большинство редакторов могут работать с несколькими файлами одновременно. Разные файлы открываются в разных окнах или в разных панелях одного окна (рис. 4.5).

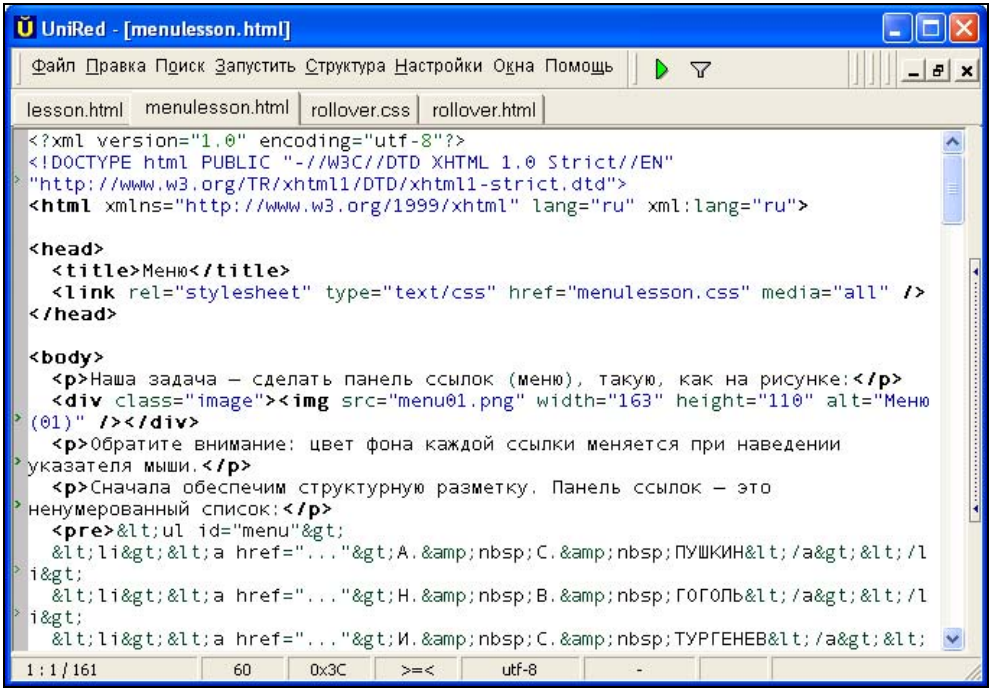


Рис. 4.5. В редакторе UniRed четыре файла открыты в разных панелях

В Windows переключаться между разными окнами можно с помощью клавиш <Alt>+<Tab>, между разными панелями одного окна — <Ctrl>+<Tab>. Автор предпочитает, чтобы разные документы открывались в разных окнах — тогда не возникает ошибок, связанных со случайным нажатием <Alt>+<Tab> вместо <Ctrl>+<Tab> и наоборот.

Текстовые редакторы обычно не выполняют некоторых распространенных файловых операций, например:

- **переименование текущего файла** — команда **Сохранить как...** создает новый файл с другим именем, а не изменяет имя старого файла;
- **удаление файла** — иногда бывает нужно вовсе удалить текущий файл. Для этого приходится отвлекаться от текстового редактора и переключаться в файловый менеджер.

Дополнительно

Текстовый редактор Vim позволяет выполнять любые файловые операции, включая две перечисленные. Для этого можно, нажав специальное сочетание клавиш, прямо в окне редактора вызвать командную строку и ввести в ней команду на переименование или удаление.

Работа с файлами также происходит легко во встроенном редакторе FAR Manager, т. к. в нем можно быстро переключаться между сеансом редактирования и сеансом работы с файловыми панелями.

4.2.9. Обработка текста

> И сразу загремели барабаны, и оловянные солдатики, сделанные из пластмассы, стали переделывать местность. Они пересаживали лесок, перетаскивали песок, переделывали трясину на теснину и перестраивали теснину на трясину.
>
> *Л. Лиходеев. Звезда с неба*

Самыми распространенными операциями обработки текста являются *поиск* и *замена* (табл. 4.4).

Таблица 4.4. *Операции поиска и замены*

Операция	Традиционная команда в меню	Сочетание клавиш
Поиск	**Правка** | **Найти** (Edit | Find)	<Ctrl>+<F>
Замена	**Правка** | **Заменить** (Edit | Replace)	<Ctrl>+<H> (только в некоторых редакторах)

В большинстве редакторов поиск по тексту можно осуществлять одним из следующих способов:

- **литеральный поиск** — вы вводите в строку поиска слово или словосочетание, и редактор ищет всюду в тексте в точности это слово или сочетание;
- **поиск по шаблону** — многие текстовые редакторы и процессоры позволяют писать простые шаблоны, позволяющие искать участки текста, удовлетворяющие определенным условиям;

Примечание

Например, в диалоговом окне поиска Microsoft Word сочетание символов `^#` соответствует любой цифре, а сочетание `^p` обозначает перевод строки. Таким образом, по шаблону `^p^#` будут найдены цифры, стоящие в начале абзаца (за исключением самого первого абзаца).

- **поиск с использованием регулярных выражений** — регулярные выражения — это целый язык, позволяющий задавать сколь угодно сложные шаблоны соответствия строк. Механизм обработки регулярных выражений встроен во многие современные языки программирования, включая Perl и Java.

Дополнительно

Цифра в начале абзаца соответствует регулярному выражению `^[0-9]`. А такому регулярному выражению: `\$[0-9]+(\.[0-9][0-9])?` соответствует любая денежная сумма в долларах (и, возможно, центах), например: «$10», «$14.99», «$0.07» и т. д.

С помощью регулярных выражений можно задать шаблон, соответствующий любому правильно составленному адресу электронной почты или URL, всякому тэгу или элементу HTML и вообще любому фрагменту текста, который можно описать словами.

В этой книге не описывается язык регулярных выражений. Всю необходимую информацию можно найти в специальной литературе, например, Дж. Фридл. Регулярные выражения. СПб.: Питер, 2003. Овладев регулярными выражениями, вы сможете быстро строить сложные конструкции для поиска и замены текста в своих документах.

Замена текста позволяет не только заменять одно сочетание символов другим, но и преобразовывать текст более сложными способами. Например, используя регулярные выражения, ничего не стоит обнаружить в тексте все числа и переставить в них цифры в обратном порядке.

Более жизненный пример: допустим, нам требуется устранить из текста все пробелы в конце строки. В Microsoft Word для этого можно искать все, что соответствует шаблону `␣^p`, и заменять этот текст на один перевод строки `^p`. В других редакторах вместо `^p` может использоваться `\n` — стандартное обозначение перевода строки во многих языках программирования.

Дополнительно

Если в тексте есть строки, где перед концом строки стоит несколько пробелов, то вышеописанную замену потребуется произвести несколько раз. Можно также составить регулярное выражение, которое учтет произвольное количество пробелов.

Вообще, готовый текстовый файл имеет смысл «чистить»: удалять пробелы в конце строки и лишние пробелы в начале строки (оставляя, впрочем, те, которые поставлены в соответствии со стилем кодирования), превращать случайные двойные пробелы в один, удалять ненужные пустые строки и т. д. Такая «чистка» — часть культуры кодирования; это полезная привычка, которую рекомендуется взять на вооружение любому кодировщику.

Разумеется, «чистить» код не обязательно вручную. Можно написать и подключить к текстовому редактору простую программу с использованием регулярных выражений, которая будет совершать все необходимые операции автоматически.

Всякий текстовый редактор предлагает различные настройки поиска и замены. Например, можно искать и заменять текст глобально во всем документе, от текущей позиции до конца или до начала документа, в выделенном фрагменте и т. д.

При поиске и замене можно указать, следует ли учитывать регистр символов. Например, при поиске без учета регистра слова «мур» и «МУР» будут обработаны как одинаковые, а с учетом регистра — как два разных слова.

Примечание

При поиске на русском языке снова возникают проблемы, связанные с особенностями русского алфавита — например, во многих редакторах средства поиска и замены некорректно обрабатывают слова с буквой «Ё». Или, наоборот, обрабатывают их чересчур корректно — считают «Е» и «Ё» совершенно разными буквами и не учитывают того, что в большинстве текстов «Е» употребляется вместо «Ё».

Обработка текста не ограничивается поиском и заменой. Многие текстовые редакторы умеют сортировать абзацы текста (например, по алфавиту), а также позволяют пользователю разрабатывать и подключать собственные средства для обработки текста.

Например, существуют небольшие программы, приводящие текст в соответствие с типографскими правилами. Они заменяют кавычки ASCII на грамотные «елочки» (правильно определяя, где должна быть правая кавычка, а где — левая), суррогат-дефис — на короткое или длинное тире (опять же, определяя, где какое тире должно быть) и т. д.

Полезное вспомогательное средство многих текстовых редакторов — это создание сводки по документу. Редактор может подсчитать, сколько в документе знаков, сколько слов, какие слова повторяются наиболее часто и т. д.

Если вы будете хорошо представлять себе возможности, преимущества и недостатки различных текстовых редакторов, вы сможете не только выбрать наилучший редактор для своей работы, но и использовать его возможности наиболее полно и эффективно.

```
<?xml version= 1.0
<!DOCTYPE html PUBL
  "-//W3C//DTD XHTM
  "http://www.w3.or
<html xmlns="http:/
  lang="ru" xml:lan
<!-- Редактировал:
<head>
  <title>Евгений Он
  <link rel="styles
    href="index.cs
</head>
```

Глава 5

Наука побеждать

Техника разметки и оформления текста с помощью HTML и CSS

Изначально разметкой назывались комментарии технического редактора к рукописи, которые поясняли наборщику, как отформатировать документ. Они состояли из рукописных замечаний, например: «Набрать этот заголовок курсивным шрифтом Helvetica 12 пунктов, в тексте шрифт 10 пунктов, выключная строка на шпон 22 пик, слева отступы 1 em, справа без отступов».

М. Гуссенс, Ф. Миттельбах, А. Самарин.

Путеводитель по пакету LaTeX

и его расширению LaTeX 2_ε

Он никогда не считал себя воином света. А женщина, похоже, прочла его мысль и сказала: «На это способны все. И, хотя никто не считает себя воином света, каждый человек может стать им».

П. Коэльо. Книга воина света

HTML (HyperText Markup Language) — это основной язык, с помощью которого создаются веб-страницы. Страница, которую пользователь видит в своем браузере, может состоять из множества разных файлов — например, изображений, анимационных роликов, сценариев JavaScript, апплетов и т. д. — но основой страницы практически всегда является документ HTML. Другие языки разметки (прежде всего многочисленные приложения XML) пока еще слабо поддерживаются браузерами и поэтому не вытесняют HTML на компьютерах обычных пользователей.

CSS (Cascading Style Sheets) — это язык, предназначенный для оформления веб-страниц и некоторых других видов документов. Разработчик стандартов HTML и CSS, Консорциум Всемирной Паутины (The World Wide Web Consortium, W3C) рекомендует разделять структуру и представление — т. е.

кодировать в HTML только логическую структуру документа, а все, что связано с отображением документа на экране или представлением на других устройствах, выполнять средствами CSS.

Эта книга — не учебник по HTML и CSS. Хотя в этой главе содержится справочный материал по основам обоих языков, все же для их настоящего изучения автор рекомендует обратиться к официальной документации Консорциума W3C. Бóльшая часть материала этой главы предназначена для тех, кто уже хотя бы немного владеет HTML и CSS.

5.1. С чего начинается Родина
HTML — язык структурной разметки гипертекста

Размечают наши души
Те, которые велят.
Здесь — подправить, там — разрушить,
Здесь — поштучно, там — подряд.

Ю. Устинов. Золотой мотылек

Using markup improperly — not according to specification — hinders accessibility. Misusing markup for a presentation effect (e.g., using a table for layout or a header to change the font size) makes it difficult for users with specialized software to understand the organization of the page or to navigate through it.

Web Content Accessibility Guidelines 1.0[1]

Аббревиатура «HTML» расшифровывается как «HyperText Markup Language» (в переводе на русский язык — «язык разметки гипертекста»). Взрывной рост Всемирной Паутины в начале 1990-х во многом обусловлен широким распространением этого языка и браузеров, отображающих написанные на нем страницы.

Примечание

Статью об истории HTML можно найти по адресу: **www.w3.org/MarkUp/historical**.

---

[1] Применение разметки не по назначению — не в соответствии со спецификацией — наносит ущерб доступности сайта. Из-за употребления разметки для достижения определенного внешнего вида (например, при использовании таблицы для верстки страницы или заголовка для получения шрифта определенного размера) пользователи со специфическим программным обеспечением испытывают трудности с пониманем организации сайта или с навигацией по нему. *(Принципы доступности содержания сайтов, редакция 1.0)*

Все три слова в названии языка — «структурный», «разметка» и «гипертекст» — нуждаются в дополнительных пояснениях, которые сейчас и будут даны.

5.1.1. Как это понимать? Что означают слова «структурный», «разметка» и «гипертекст»

> Разметку производят на разметочной плите. Для правильной установки заготовок на разметочной плите пользуются специальными приспособлениями: призмами, домкратами, подкладками и др.
>
> *Большая советская энциклопедия*

Гипертекст — это просто текст со ссылками. Текст практически каждой веб-страницы является типичным примером гипертекста: можно щелкнуть по ссылке, и вы немедленно перейдете к другой странице, к загружаемому файлу или еще к какому-нибудь ресурсу, на который указывает эта ссылка.

Внимание!

Гипертекст — текст со ссылками.

Дополнительно

Некоторые источники приводят в качестве примеров гипертекста также тексты, созданные задолго до компьютерной эпохи. Например, с их точки зрения, синодальный перевод Библии, который широко используется у нас, является типичным гипертекстом: основной текст сопровождается так называемыми «параллельными местами», то есть ссылками на другие фрагменты Библии, имеющие отношение к данному участку текста.

Кто-то возражает, что этот «гипертекст», в отличие от сетевого, не обеспечивает средств перехода по ссылке (проще говоря, по параллельному месту в Библии нельзя щелкнуть мышью) и поэтому не является настоящим гипертекстом.

Мы не будем обсуждать вопрос о том, применимо ли понятие «гипертекст» к печатным изданиям. В этой главе обсуждается разработка веб-страниц, и нам вполне достаточно того, что на них уж точно есть настоящий гипертекст.

Теперь поговорим о разметке. Как вам известно, в коде документа HTML находится не только сама информация, которую пользователь увидит на веб-странице, но и некоторые инструкции о том, как браузер пользователя будет обрабатывать эту информацию. Например, можно включить в документ такую конструкцию:

```
<h1>Евгений Онегин</h1>
```

Пользователь увидит на странице только сам текст «Евгений Онегин». Окружающие его тэги `<h1>` и `</h1>` не отображаются на странице, но снабжают браузер информацией, позволяющей правильно интерпретировать этот текст. Они обозначают, что такой текст является заголовком первого уровня. Большинство современных графических браузеров по умолчанию выделяют этот заголовок полужирным шрифтом увеличенного кегля. (Забегая вперед, заметим, что многие поисковые системы придают заголовкам больший вес, чем обычному тексту, а заголовкам первого уровня — в свою очередь, больший вес, чем прочим заголовкам.)

Такие конструкции, которые сами не являются содержанием страницы, но указывают браузеру, как интерпретировать содержание, и называются разметкой.

Внимание!

Разметка — это код, который сопровождает содержание документа и предоставляет браузеру (или другой обрабатывающей программе) информацию о том, как следует интерпретировать те или иные фрагменты содержания.

Язык HTML предназначен в первую очередь для структурной разметки, т. е. для обозначения каждого элемента в соответствии с его местом и ролью в структуре документа. Например, тэг `<p>` обозначает, что элемент является просто абзацем обычного текста, тэг `<ul>` — что элемент является перечнем, тэг `<blockquote>` — что элемент является цитатой и т. д.

В HTML есть также тэги, управляющие внешним видом отдельных элементов: например, тэг `<i>` выделяет текст курсивом, тэг `<font>` изменяет различные параметры шрифта, а тэг `<br />` вставляет перевод строки. Все такие тэги не относятся к структурной разметке, и применять их не рекомендуется. Все, что касается оформления веб-страницы, ее внешнего вида, можно и нужно делать с помощью CSS.

Дополнительно

Одна из многочисленных причин, почему оформление следует делать средствами CSS, заключается в том, что оформительские возможности HTML очень скудны. Например, с помощью HTML можно сделать рамки вокруг таблицы — но нельзя сделать рамки с разных сторон разными. А что касается оформления текста (собственно, ему и посвящена эта книга) — HTML не предоставляет для этого практически никаких средств. С ним нельзя сделать абзацный отступ, разрядку, управлять междустрочным интервалом и междусловными пробелами… Практически все типографское оформление шрифта можно сделать только с помощью CSS.

Примечание

Существует много эффектов, которых можно добиться как средствами HTML, так и средствами CSS. Например, чтобы закрасить фон страницы в зеленый цвет, можно написать в HTML:

```
<body bgcolor="#99ff99">,
```

а можно вместо этого написать в CSS:

```
body { background-color: #99ff99; }
```

По многим причинам использование CSS гораздо предпочтительнее. Эти причины будут подробно рассмотрены в *разд. 5.2*.

Структурная разметка не имеет никакого отношения к внешнему виду документа. Документ может быть отображен на экране компьютера, написан от руки или напечатан на пишущей машинке; в конце концов, его можно прочитать вслух — но все заголовки в нем останутся заголовками, все перечни — перечнями, а цитаты — цитатами. Структурная роль элементов документа, в отличие от оформления, никак не меняется в зависимости от того, какими средствами просматривают этот документ.

Внимание!

Структурная разметка — это разметка, обозначающая роль каждого элемента в структуре документа.

Иными словами, структурная разметка говорит о том, чем *является* тот или иной элемент, а не о том, как его следует или не следует *отображать*.

Грамотная структурная разметка обеспечивает независимость документа от устройства вывода. Сравните две записи:

- `<font size="+2"><b>Евгений Онегин</b></font>`
- `<h1>Евгений Онегин</h1>`

Первая запись говорит о том, что данный текст нужно выделить полужирным шрифтом увеличенного размера. А вторая — что данный текст является заголовком первого уровня. Во многих графических браузерах обе записи приводят к одному и тому же внешнему виду. Но что делать неграфическим браузерам — например, текстовым, речевым или браузерам Брайля? Они просто по конструкции не могут отобразить полужирный шрифт увеличенного размера. Поэтому при воспроизведении этими браузерами текст первой записи останется невыделенным.

А вторая запись гарантирует, что *любой* качественный браузер выделит этот текст доступными ему средствами. Например, текстовый браузер может отобразить его более ярким цветом, чем основной текст, а также пропустить строку перед ним и после него. Таким образом, пользователь обязательно увидит, что это — заголовок, а не просто обычный текст. Речевой браузер может произнести текст с увеличенной громкостью, с интонационным выделением, а также с паузами перед ним и после него.

Структурная разметка не говорит о том, как конкретно должен быть выделен текст. Но это не мешает разработчику оформить его нужным образом с помощью CSS. Например, можно написать в таблице стилей:

```
h1 {
  color: #009933;
  font: 1.5em bold sans-serif;
}
```

и в тех браузерах, которые поддерживают графическое оформление, заголовок первого уровня отобразится полужирным шрифтом без засечек темно-зеленого цвета, с увеличенным размером. При этом в текстовых, речевых и брайлевских браузерах этот текст также будет как-то выделен. Технология CSS позволяет даже написать отдельные таблицы стилей для каждого возможного устройства отображения.

Внимание!

Структурная разметка и оформление не противоречат друг другу. На грамотной веб-странице должно быть и то, и другое. Структурную разметку рекомендуется выполнять средствами HTML, а оформление — средствами CSS.

Дополнительно

Хорошая структурная разметка также облегчает автоматическую обработку документа. Например, когда необходимо пронумеровать все заголовки или составить оглавление документа — программа легко справится с этой задачей, если заголовки помечены соответствующими структурными тэгами. Можно при этом включить в середину документа новые заголовки — программе не составит труда перенумеровать все заново и обновить оглавление. Если не использовать для заголовков правильные тэги `<h1>`, `<h2>` и т. д., то их нумерация или составление оглавления превратится в кропотливую и неэффективную ручную работу.

5.1.2. Во первых строках письма… Шаблон документа HTML

> Помнится, Наполеон писал: «On s'engage et puis on voit». В вольном русском переводе это значит: «Сначала надо ввязаться в серьезный бой, а там уже видно будет».
>
> *В. И. Ленин. О нашей революции*

Вспомним, как выглядит шаблон документа HTML в наиболее общем виде (листинг 5.1).

Листинг 5.1. Шаблон документа HTML

```
01  <?xml␣ version="1.0"␣ encoding="..."?>
02  <!DOCTYPE␣ html␣ PUBLIC␣ "-//W3C//DTD␣ XHTML␣ 1.0␣ Strict//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
03
04  <html␣ xmlns="http://www.w3.org/1999/xhtml"␣ lang="..."␣ xml:lang="...">
05
06  <head>
07  ␣ ␣ <title>...</title>
08  ␣ ␣ <link␣ rel="stylesheet"␣ type="text/css"␣ href="..."␣ media="all"␣ />
09  ...
10  </head>
11
12  <body>
13  ...
14  </body>
15
16  </html>
```

Обратите внимание: все пробелы в этом коде обозначены специальным знаком «␣ » (U+2423). Все строки пронумерованы, чтобы не возникало путаницы с переводом строк. Например, строка 02 здесь просто не умещается на странице, поэтому она разбита пополам. На самом деле, в коде это должна быть одна длинная строка.

Строки 03, 05, 11 и 15 — пустые.

Дополнительно

Возникают определенные сложности с тем, что в русскоязычной технической литературе слово «строка» употребляется в трех различных значениях.

В английском языке слово «string» чаще всего обозначает строковый тип данных (в отличие от, допустим, числового); вы встречались с ним, если занимались программированием или хотя бы работали с электронными таблицами. Слово «line» обозначает строку в том смысле, в каком это слово встречается двумя абзацами выше — то есть участок кода, завершающийся переводом строки. (Обратите внимание: «перевод строки» по-английски — «line break», а не «string break»!) И, наконец, слово «row» относится к строке таблицы, то есть к ряду данных. По-русски все три этих различных объекта обычно называются одним словом — «строка», что может привести к путанице. Например, попробуйте перевести на русский язык предложение: «Only string data is allowed in the second row of the table which begins at line 41». У вас получится: «Во второй строке таблицы, которая начинается с 41-й строки, допустимы только строки». Все понятно, не так ли?

Обычно даже у тех, кто раньше сталкивался с HTML, возникают вопросы о смысле некоторых конструкций в этом шаблоне. Попробуем объяснить все по порядку.

Пролог документа и версии HTML

> Он великий учитель: весь пролог наизусть!
> Ваши слова прямо в Евангелие, а наши и в пролог не годятся!
>
> *В. И. Даль. Толковый словарь живого великорусского языка*

Строки 01 и 02 — это так называемый *пролог* документа. Прежде всего он нужен для того, чтобы обозначить версию HTML, на которой написан этот документ.

Действительно, существует множество разных версий языка HTML. Когда-то была широко распространена версия HTML 2.0, затем ее надолго сменил язык HTML 3.2. Сейчас чаще всего применяется HTML 4.0 или его ближайшая разновидность — HTML 4.01.

Дополнительно

Многие версии HTML дополнительно делятся на типы. Например, существует три типа HTML 4.0 и HTML 4.01: Strict, Transitional и Frameset. Transitional («переходный») — это особенно распространенный тип, на нем написано большинство существующих страниц и именно ему посвящена основная масса учебников и справочников по HTML. Strict («строгий») отличается от переходного тем,

что в нем недопустимы тэги и элементы, отвечающие за внешний вид, представление документа. Например, в строгом типе нет тэгов `<font>` и `<u>`, атрибутов `bgcolor` и `alink`. Тип Frameset («фреймовый»), наоборот, еще либеральнее переходного, так как допускает все то же, что и переходный, а также поддерживает фреймы.

Приведенный здесь шаблон соответствует более современной версии языка HTML — XHTML 1.0 Strict. Эта версия обладает существенными достоинствами по сравнению со всеми предыдущими, поэтому автор рекомендует использовать именно ее.

Примечание

Все примеры из этой книги написаны в соответствии с XHTML 1.0 Strict. Автор не гарантирует, что они будут работать, если ваш документ не соответствует этой версии языка HTML.

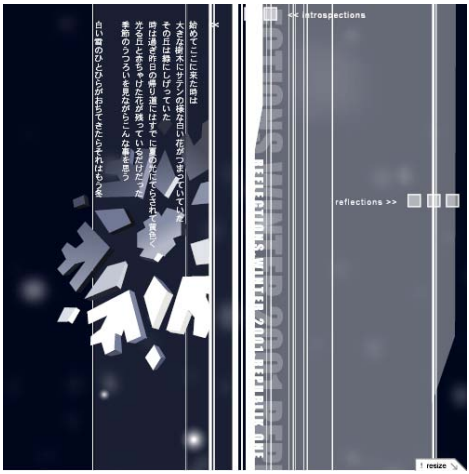
Главное достоинство XHTML — в том, что он совместим с более общим языком XML (Extensible Markup Language, «расширяемый язык разметки»). Это значит, что всякий документ, написанный на языке XHTML, одновременно является и документом, написанным на языке XML. Технология XML приобретает все более широкое распространение, поэтому совместимость с ней обеспечивает бóльшую универсальность документов и облегчает их обработку стандартными средствами работы с XML.

Дополнительно

Многие современные языки разметки совместимы с XML. К таким языкам, например, относится SVG (Scalable Vector Graphics, язык описания векторной графики) и MathML (язык описания математических формул) – рис. 5.1. Многие приложения переходят на внутреннее представление данных в формате, совместимом с XML.

Для того чтобы документ был совместим с XML, необходимо соблюдать определенные правила кодирования. Для HTML 4 и предыдущих версий большинство этих правил были не обязательными, но в XHTML им нужно следовать:

- у каждого открывающего тэга должен быть соответствующий ему закрывающий. В старых версиях HTML некоторые тэги (например, `<p>` или `<td>`) можно было не закрывать, но в XHTML закрывающие тэги обязательны;
- имена всех тэгов и атрибутов записываются строчными буквами. В XML регистр букв имеет значение, поэтому `<head>` и `<HEAD>` — два разных тэга;
- значения атрибутов обязательно заключаются в кавычки;



$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

```
<svg viewBox="0 0 1000 500" x="0" y="0"
preserveAspectRatio="xMaxYMin slice"
xml:space="preserve">
 <g id="P60bab30" transform="matrix(1 0
0 1 763 -4)">
  <defs>
   <path id="B60b9090"
d="M0,0q0,508,0,508q9.47,0,9.47,0q0-
347.32,0-
347.32Q18,112.78,18,112.78Q18,2,18,2Q0,0
,0,0z"/>
  </defs>
  <use xlink:href="#B60b9090"
id="U60bc440" style="fill:#FFFFFF;"/>
 </g>
 <g id="P60b9e10" transform="matrix(1 0
0 1 785 -5)">
  <defs>
   <path id="B60b9a00"
style="fill:none;stroke-
width:0.5;stroke-linecap:round;stroke-
linejoin:round;"
d="M0,0q0,512.5,0,512.5"/>
  </defs>
  <use xlink:href="#B60b9a00"
id="U60ba120"
style="fill:none;stroke:#FFFFFF;"/>
 </g>
  ...
```

а

```
<mrow>
  <mi>x</mi>
  <mo>=</mo>
  <mfrac>
    <mrow>
      <mrow>
        <mo>-</mo>
        <mi>b</mi>
      </mrow>
      <mo>&PlusMinus;</mo>
      <msqrt>
        <mrow>
          <msup>
            <mi>b</mi><mn>2</mn>
          </msup>
          <mo>-</mo>
          <mrow>
            <mn>4</mn>
            <mo>&InvisibleTimes;</mo>
            <mi>a</mi>
            <mo>&InvisibleTimes;</mo>
            <mi>c</mi>
          </mrow>
        </mrow>
      </msqrt>
    </mrow>
    <mrow>
      <mn>2</mn>
      <mo>&InvisibleTimes;</mo>
      <mi>a</mi>
    </mrow>
  </mfrac>
</mrow>
```

б

Рис. 5.1. Исходный код и результат просмотра в браузере: *а* — SVG; *б* — MathML

- тэги пустых элементов (`link`, `img`, `br` и др.) завершаются косой чертой, например: `<br␣ />`. (Пробел, стоящий перед косой чертой, не обязателен в XML, но рекомендуется для совместимости с устаревшими браузерами, понимающими только «традиционный» HTML.)

Правильно в XHTML	Неправильно в XHTML
`<td>Пушкин</td>`	`<td>Пушкин`
`<td>Гоголь</td>`	`<td>Гоголь`
`<td>Толстой</td>`	`<td>Толстой`
`<title>Война и мир</title>`	`<TITLE>Война и мир</TITLE>`
`<a href="myfile.html">`	`<a href=myfile.html>`
`<br␣ />`	` ` (также не рекомендуется ` `)

Дополнительно

В XHTML недопустима также сокращенная запись атрибутов. Например, чтобы отметить флажок или переключатель, недостаточно написать просто `checked`, надо написать полностью: `checked="checked"`.

В XHTML 1.0 Strict нет тэгов и атрибутов, отвечающих за оформление текста — например, тэгов `<center>` и `<font>`, атрибутов `alink`, `bgcolor`, `color` и многих других, к которым привыкли некоторые разработчики. Все, что касается внешнего вида документа, должно быть закодировано не в HTML, а в CSS.

Пролог состоит из двух строк. Первая строка — это *декларация XML*. В ней записано, какой версии XML соответствует этот документ. На сегодняшний день существует две редакции XML — 1.0 и 1.1. Редакция 1.1 появилась в феврале 2004 и пока еще не получила широкого распространения, поэтому на момент написания этой книги рекомендуется для документов XHTML применять XML 1.0 и указывать именно эту версию в атрибуте `version`. Атрибут `charset` предназначен для кодировки, в которой представлен документ — например, Windows-1251, KOI8-R и т. д.

Дополнительно

Может показаться, что с атрибутом `charset` возникает та же проблема, что описана в *главе 3* в связи с тэгом `<meta>`. Там говорилось, что кодировка должна сообщаться не в самом документе, а в заголовках протокола HTTP при передаче этого документа.

Дело в том, что язык XML предназначен не только для передачи данных по протоколу HTTP, но и для использования во многих других условиях, где нет

возможности указать кодировку отдельно от документа. Поэтому в декларации XML должен присутствовать атрибут `charset`.

Впрочем, существует компромисс между передачей страниц по HTTP (при которой кодировку не рекомендуется указывать в самом документе) и языком XML (в котором кодировку, наоборот, желательно указывать). Если документ представлен в кодировке UTF-8 (а также в некоторых других кодировках Unicode), то атрибут `charset` является необязательным. Таким образом, чтобы избежать затруднений, нужно создавать страницы в UTF-8 — благо, эта кодировка имеет и ряд других достоинств по сравнению с 8-битными расширенными ASCII. В этом случае декларация XML будет выглядеть следующим образом:

```
<?xml␣version="1.0"?>
```

Обратите внимание, что декларация XML начинается с символов `<?` и заканчивается символами `?>`. Не забывайте про вопросительный знак.

Вторая строка пролога — *объявление типа документа* (Document Type Declaration). Именно в этой строке записано, какой версии и какому типу языка HTML соответствует данный документ. Объявление типа составляется по определенным правилам, подробное описание которых выходит за рамки этой книги.

Примечание

Подробно про объявление типа документа для документов XML (в том числе XHTML) рассказывается в стандарте XML 1.1: **www.w3.org/TR/2004/REC-xml11-20040204/#sec-prolog-dtd**.

Для начала достаточно запомнить, что если документ соответствует XHTML 1.0 Strict, то объявление типа документа должно выглядеть в точности таким образом. Просто вставляйте эту строчку в начало каждого документа, который написан на этой версии HTML. Объявление типа документа для других версий HTML можно найти на сайте Консорциума W3C (**www.w3.org**).

Дополнительно

Формальные правила XHTML 1.0 Strict, как и любой другой версии HTML, описаны в специальном нормативном документе, который называется *определением типа документа* (Document Type Definition, сокращенно DTD). Именно в DTD указано, какие тэги и атрибуты вообще бывают в этой версии, какие элементы можно, а какие нельзя вкладывать друг в друга, какие атрибуты могут и не могут относиться к определенным элементам и т. д. Если вы ищете первоисточник информации о синтаксических правилах XHTML 1.0 Strict — обратитесь к соответствующему DTD. Его адрес указан в объявлении типа документа — **www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd**.

К сожалению, начинающим разработчикам не всегда удается разобраться в самом DTD. Например, взгляните на фрагмент DTD XHTML 1.0 Strict (листинг 5.2).

Листинг 5.2. Фрагмент DTD XHTML 1.0 Strict

```
<!--=================== Document Body ====================================-->

<!ELEMENT body %Block;>
<!ATTLIST body
  %attrs;
  onload          %Script;   #IMPLIED
  onunload        %Script;   #IMPLIED
  >

<!ELEMENT div %Flow;>  <!-- generic language/style container -->
<!ATTLIST div
  %attrs;
  >

<!--=================== Paragraphs ======================================-->

<!ELEMENT p %Inline;>
<!ATTLIST p
  %attrs;
  >

<!--=================== Headings ========================================-->

<!--
  There are six levels of headings from h1 (the most important)
  to h6 (the least important).
-->

<!ELEMENT h1  %Inline;>
<!ATTLIST h1
   %attrs;
   >

<!ELEMENT h2 %Inline;>
<!ATTLIST h2
   %attrs;
   >
```

В этом фрагменте определены элементы `body`, `div`, `p`, `h1` и `h2`. Вполне возможно, что вам эти определения кажутся китайской грамотой. (В конце концов,

давно пора: недаром дополнительные разделы обозначены именно иероглифом.) Не расстраивайтесь: для того чтобы разрабатывать качественный код HTML, не обязательно быть специалистом по чтению DTD. Однако автор рекомендует вам как-нибудь найти время и разобраться в правилах, по которым записываются DTD. Вводную статью на эту тему можно прочитать в спецификации HTML 4.01, в разделе «On SGML и HTML» (**www.w3.org/TR/html4/intro/sgmltut.html**).

Примечание

Не путайте объявление типа документа и определение типа документа. Объявление типа документа (Document Type Declaration) — это строчка, которая включается в документ HTML и обозначает версию, на которой он написан. А определение типа документа (Document Type Definition) — это стандарт, содержащий формальные правила языка разметки.

Валидатор и стандарты W3C

> Воинственная раса отличается жесткой дисциплиной и слишком высокими стандартами поведения. До тех пор, пока они замкнуты на себя, они непобедимы.
>
> *Л. Спрэг де Кемп. Воинственная раса*

Для того чтобы проверить, соответствует ли ваш документ той или иной версии HTML (XHTML), лучше всего использовать так называемый *валидатор*. Это программа, которая проверяет, соответствует ли код документа формальным правилам того DTD, который объявлен в прологе.

Лучший из существующих валидаторов расположен в Сети по адресу **validator.w3.org** (рис. 5.2). С его помощью вы можете проверить любой документ, расположенный во Всемирной Паутине или на вашем собственном компьютере.

Автор не рекомендует пользоваться другими валидаторами — по крайней мере до тех пор, пока вы сами не научитесь отличать настоящие валидаторы от программ, которые просто так называются. Во многих интегрированных средах разработки (например, HomeSite или Dreamweaver) есть встроенные «валидаторы», которые, к сожалению, часто ошибаются: не замечают многих синтаксических ошибок, запрещенных в DTD, зато, наоборот, считают ошибками некоторые синтаксически правильные, хотя и малоупотребительные конструкции.

Документ, соответствующий объявленному DTD, называется *валидным*, или *действительным*.

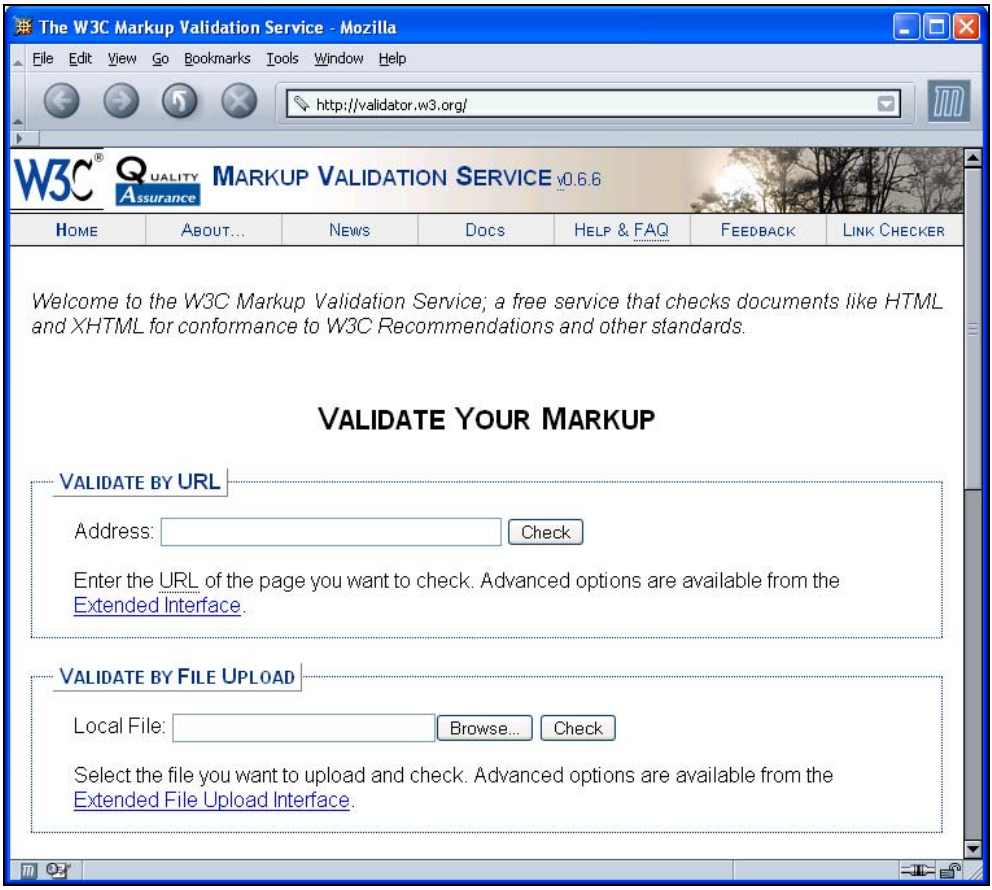


Рис. 5.2. Сетевой валидатор Консорциума W3C. Введите адрес или имя файла на вашем компьютере — и вы узнаете, соответствует ли этот файл стандарту HTML. Проверяя файл со своего компьютера, рекомендуется также указывать его кодировку — в некоторых случаях валидатор не распознает ее и обрабатывает документ некорректно

Внимание!

Обязательно проверяйте свои страницы валидатором. Пользуйтесь валидатором Консорциума W3C (**validator.w3.org**).

Большинство существующих в Интернете страниц, к сожалению, не являются валидными. Многие разработчики считают, что абсолютно строгое соблюдение стандартов HTML — вовсе не обязательное условие того, чтобы сайт получился качественным и эффективным. Действительно, существует множество весьма успешных сайтов, код которых не проходит придирчивой проверки валидатором.

Автор этой книги согласен, что можно сделать хороший сайт, не добиваясь полного соответствия кода формальным правилам языка. Однако разработка валидного кода имеет ряд преимуществ как для самого разработчика, так и для конечного пользователя. Автор настоятельно рекомендует писать грамотный код и проверять его валидатором. Это полезно по следующим причинам:

- проверка документов валидатором позволяет избежать мелких досадных ошибок — неправильно вложенных тэгов, пропущенных скобок и кавычек и т. д. Если страница отображается не так, как хотел разработчик, — в половине случаев это происходит из-за какой-нибудь мелкой неаккуратности в коде. Валидатор поможет вам автоматически исключить такие ошибки.

  Примечание

  Коварство разработки веб-страниц заключается в том, что разработчик не может предсказать все проблемы, которые могут возникнуть у различных пользователей с его страницами. Даже если страница была тщательно протестирована — обязательно найдутся пользователи, у которых неожиданно появляется горизонтальная полоса прокрутки, блоки на странице «соскальзывают» друг под друга, возникают неожиданные отступы, создающие ненужные пустые пространства на странице и т. д. Проверка кода валидатором не спасет от всех неожиданностей, но убережет по крайней мере от половины. Если вы хотите наполовину сократить время, которое вы тратите на тестирование и отладку своих страниц, — пишите грамотный код и обязательно проверяйте его валидатором.

  В самом деле, ошибки при отображении страницы в браузере могут возникать по многим причинам, и в том числе — из-за неграмотного кода. Если вы избавились от неграмотного кода — вам будет легче тестировать страницы, так как исчезнет одна из самых распространенных причин ошибок;

- современные браузеры поддерживают стандарты W3C гораздо лучше, чем их прежние версии. Эта тенденция сохраняется, так что соответствие DTD приобретает все большую важность для правильного отображения страниц в браузерах.

  Многие браузеры применяют разные алгоритмы анализа и отображения страниц в зависимости от того, какой DTD указан в прологе и указан ли он вообще. Если DTD не указан, браузер вынужден «угадывать», как отображать страницу — и часто не может угадать правильно. Если DTD указан, но код страницы ему не соответствует, — страница также может отобразиться некорректно;

Дополнительно

Хотя браузеры поддерживают стандарты Консорциума W3C все лучше и лучше, до полного соответствия пока еще далеко. При разработке страниц приходится уделять много внимания их совместимости с конкретными браузерами. Однако это не противоречит тому, чтобы писать валидный код. Хороший код является одновременно и валидным, и совместимым с наиболее популярными браузерами — одно другому не мешает.

- так как валидный код соответствует определенным формальным правилам, его легче интерпретировать и обрабатывать. Он быстрее анализируется и отображается в браузере, с ним легче работать поисковым и индексирующим системам. Валидная страница лучше поддается автоматической обработке — например, если требуется совершить какое-нибудь однотипное редактирование над множеством страниц, то задача существенно упростится, если эти страницы будут соответствовать определенному DTD;

Дополнительно

Вот пример задачи, которую однажды пришлось выполнять автору. Требовалось обработать код страниц таким образом: если в коде присутствует абзац текста, за которым следует любой элемент, кроме абзаца, включающий в себя только текст, причем за ним следует еще один абзац, — то добавить перед вторым абзацем определенный графический элемент. Программа на Perl, решающая эту задачу для кода, соответствующего XHTML 1.0 Strict, была написана за 15 минут. Но когда через некоторое время ее потребовалось применить к невалидным файлам — пришлось промучиться над модификацией программы полдня, и в результате все равно выполнять половину работы вручную.

- валидность кода гарантирует совместимость ваших страниц не только с нынешними, но и с будущими версиями браузеров. Вам не придется переписывать ваши страницы после выхода новой версии Internet Explorer или Opera. Разработчики браузеров гарантируют все более и более полную поддержку стандартов. А вот корректное отображение невалидных страниц никто не гарантирует!

Дополнительно

Многие браузеры поддерживают тэги и атрибуты, отсутствующие в стандарте языка HTML, — например, тэги `<nobr>` и `<marquee>`, атрибуты `marginwidth` и `marginheight` и т. д. Эти тэги и атрибуты не рекомендуется использовать: всегда можно добиться нужного эффекта, не нарушая стандартов.

Заботясь о грамотности страниц, об их соответствии стандартам W3C, не надо впадать в крайность и думать, что если страница успешно проверена валидатором, то она автоматически является качественной и эффективной.

Грамотный код — это важный, но далеко не единственный показатель качества страницы. Действительно, ведь можно написать текст на русском языке, который будет полностью удовлетворять грамматическим правилам — и при этом окажется неудачным или вовсе бессмысленным. Точно так же и со стандартами языка HTML: их нужно соблюдать, но ими одними не стоит ограничиваться.

Собственно говоря, соблюдение формальных правил, описанных в DTD, — это самая легкая часть разработки страниц. Валидатор всегда абсолютно точно скажет, соответствуют ли ваши страницы этим правилам. Гораздо сложнее обстоят дела с удобством, доступностью, эффективностью и привлекательностью страниц: нет никакой программы, которая может проверить, является ли ваша страница эффективной и привлекательной. Здесь разработчику приходится опираться на собственный интеллект, опыт, здравый смысл и творческие способности.

Встретив в этой книге или еще где-нибудь очередные правила, рекомендации или советы, — не торопитесь безоговорочно принимать или отвергать их. Подумайте, откуда взялись эти правила, почему и в каких случаях автор рекомендует им следовать. Посмотрите, какие преимущества приносит соблюдение этих правил и от каких недостатков оно может избавить. Проанализируйте собственный опыт и опыт ваших коллег-разработчиков. И только потом принимайте решение, следовать ли тем или иным советам в вашей конкретной ситуации.

Большинство неудачных решений разработчиков происходит не оттого, что они следуют или не следуют тем или иным стандартам и правилам, а оттого, что они следуют им необдуманно, не зная, когда на самом деле нужно или не нужно применять эти правила. Квалифицированный разработчик не только хорошо разбирается во всех стандартах, но и знает, для каких случаев эти стандарты пригодны, а для каких — нет. Автор надеется, что читатели этой книги либо уже являются, либо собираются становиться именно такими разработчиками.

Тэги, элементы, атрибуты

Wie alles sich zum Ganzen webt,
Eins in dem andern wirkt und lebt!

J. W. von Goethe. Faust[2]

Давайте определимся с технической терминологией, которая будет использоваться далее в этой книге. Если вы хорошо разбираетесь в HTML, пропустите этот параграф.

---

[2] В каком согласьи, боже мой,
Все элементы меж собой!
(И. В. Гете. Фауст)

Основной конструкцией разметки в HTML являются *тэги*. Тэг — это то, что заключено в угловые скобки («<» и «>» — они же знаки «меньше» и «больше»). Тэги бывают *открывающие* и *закрывающие*, например:

Открывающие тэги	Закрывающие тэги
`<head>`	`</head>`
`<p>`	`</p>`
`<html␣ lang="..."␣ xml:lang="..."␣ xmlns="http://www.w3.org/1999/xhtml">`	`</html>`
`<a href="myfile.html">`	`</a>`

Дополнительно

Бывают еще *тэги пустых элементов* — каждый такой тэг является как бы одновременно открывающим и закрывающим, например:

```
<link␣ rel="stylesheet"␣ type="text/css"␣ href="index.css"␣ media="all"
␣ />
<img␣ src="photo.jpg"␣ width="420"␣ height="300"␣ alt="Зимний␣ лес.␣
Фотография"␣ />
<br␣ />
```

Обратите внимание: тэг пустого элемента завершается пробелом, косой чертой и правой угловой скобкой.

В XHTML для каждого открывающего тэга обязательно должен быть парный ему закрывающий. А тэги пустых элементов, наоборот, всегда бывают только одиночными.

Открывающий тэг, закрывающий тэг и все, что находится между ними, называется *элементом*. Таким образом, можно сказать, что документ HTML состоит из элементов.

То, что находится между открывающим и закрывающим тэгами, называется *содержанием* элемента. Как видно хотя бы в приведенном шаблоне, элементы могут быть вложенными друг в друга. Например, содержанием элемента `head` здесь являются элементы `title` и `link`.

Именем (или *названием*) элемента является то, с чего начинается открывающий тэг. Например, элемент `<body> ... </body>` имеет имя `body`, элемент `<a href="myfile.html">` ... `</a>` имеет имя `a`.

Внимание!

Не путайте тэги, элементы и имена элементов! Например, `<title>` — это *тэг*. А такая конструкция: `<title>Евгений Онегин</title>` — это *элемент*, причем `title` — *имя* этого элемента.

В шаблоне, приведенном в листинге 5.1, обозначено 5 элементов. Две строки пролога к элементам не относятся, а все остальное — элементы, вложенные друг в друга. Например, все, что заключено между открывающим тэгом `<html ... >` и закрывающим `</html>` (то есть практически весь документ) — это все один элемент.

Дополнительно

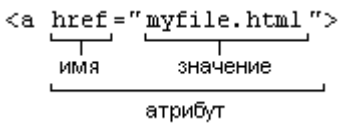
Тэг пустого элемента сам по себе является элементом. Например, строка 08 в листинге — это элемент `link`.

Дополнительно

Вообще говоря, разница между тэгами и элементами имеет более сложный характер. Элемент — это структурный компонент документа. А тэги применяются для *обозначения* этих элементов. Чтобы это понять, вспомните, в чем разница между словом и значением этого слова в обычном русском (или любом другом) языке. Слово — это просто некоторая последовательность букв. А значение слова — это смысл, который несет эта последовательность. Надеюсь, вы понимаете, что слово — это совсем не то же самое, что значение этого слова. Так же отличаются друг от друга тэги и элементы: тэг — это последовательность символов, а элемент — это некоторый смысл, который обозначается этой последовательностью.

Впрочем, если вы не хотите вдаваться в глубокий философский смысл формы и содержания — можете просто считать, что элемент — это два парных тэга и все, что находится между ними. Для большинства практических задач этого достаточно.

В некоторых открывающих тэгах, кроме имени элемента, содержатся *атрибуты*, описывающие свойства этого элемента. Атрибут состоит из *имени* и *значения*, причем значение заключается в кавычки ASCII и соединяется с именем атрибута знаком равенства, например:



Примечание

Не забывайте: в XHTML, в отличие от прежних версий HTML, имена атрибутов записываются только строчными буквами, а значения обязательно заключаются в кавычки.

В листинге 5.1 у элемента `html` три атрибута, у элемента `link` — четыре. Если у элемента несколько атрибутов, они отделяются друг от друга пробелом.

Примечание

Атрибуты записываются только в открывающем тэге элемента. Не допускайте распространенной ошибки начинающих разработчиков, записывая атрибуты в закрывающем тэге:

Правильно	Неправильно
`<a href="myfile.html"> ... </a>`	`<a href="myfile.html"> ... </a href="myfile.html">` `<a href="myfile.html"> ... </a href>`

Если один элемент имеет несколько атрибутов, то их можно указывать в любом порядке. Например,

```
<link␣ rel="stylesheet"␣ type="text/css"␣ href="..."␣ media="all"␣ />
```

— это то же самое, что

```
<link␣ type="text/css"␣ rel="stylesheet"␣ media="all"␣ href="..."␣ />.
```

В DTD указано, какие значения может иметь каждый атрибут, а также какие атрибуты можно присваивать тем или иным элементам.

5.1.3. Уроки хороших манер

Стиль кодирования

> От возмущения она хлопала ресницами.
> — Кто вас воспитывает, скажите, пожалуйста?
> — Когда папа Карло воспитывает, а когда никто.
> — Теперь я займусь вашим воспитанием, будьте покойны.
> «Вот так влип!» — подумал Буратино.
>
> *А. Н. Толстой. Золотой ключик или Приключения Буратино*

При разработке документов HTML, таблиц стилей CSS, а также при написании исходного кода компьютерных программ грамотные разработчики предпочитают придерживаться определенного *стиля кодирования*.

Чтобы понять, что такое стиль кодирования, можно посмотреть на два следующих листинга (5.3, *а* и 5.3, *б*), на которых закодирован один и тот же фрагмент документа HTML.

Листинг 5.3, *а*. Фрагмент документа HTML, вариант 1

```
...
<div class="resume">
  <div id="header">
    <h2>Статистика посещений сайта</h2>
    <div class="subh2">Данные за 14&#8211;21 декабря 2003</div>
    <!-- следующий отсчет - 29.12.2003, 05.17 -->
  </div>
  <div class="table">
    <table>
      <thead>
        <tr>
          <th>Браузер</th>
          <th>Количество посещений</th>
          <th>В % от общего количества посещений</th>
        </tr>
      </thead>
      <tfoot>  <!-- По стандарту должен быть перед tbody -->
        <tr>
          <th>Итого</th>
          <td>4458</td>
          <td>100,00</td>
        </tr>
      </tfoot>
      <tbody>
        <tr>
          <td>Microsoft Internet Explorer</td>
          <td>3786</td>
          <td>84,93</td>
        </tr>
        <tr>
          <td>Opera</td>
          <td>303</td>
          <td>6,80</td>
```

```
        </tr>
        <tr>
          <td>Mozilla</td>
          <td>251</td>
          <td>5,63</td>
        </tr>
        <tr>
          <td>Другие</td>
          <td>118</td>
          <td>2,64</td>
        </tr>
      </tbody>
    </table>
  </div>
</div>
...
```

Листинг 5.3, *б*. Фрагмент документа HTML, вариант 2

```
...
<div class="d4">  <div id="d5">
    <h2> Статистика посещений сайта
    </h2><div class="d6sub">Данные за 14&#8211;21 декабря 2003
</div> </div>
<div class="d6t"><table>
 <thead><tr>
   <th>Браузер </th><th>Количество посещений</th>  <th>В %
   от общего количества посещений
      </th></tr></thead>
 <tfoot><tr><th>Итого </th> <td>4458</td> <td> 100,00</td></tr>
</tfoot>
  <tbody> <tr>
<td>Microsoft Internet Explorer</td><td>3786 </td> <td>84,93</td>
           </tr>
           <tr>
<td>Opera</td><td>303 </td> <td>6,80</td>
     </tr><tr><td>Mozilla </td> <td>251 </td> <td>5,63</td>
</tr> <tr> <td>Другие</td>
```

```
   <td>118 </td> <td>2,64</td></tr></tbody>
 </table>  </div>
   </div>
...
```

Оба фрагмента соответствуют правилам XHTML 1.0 Strict и успешно проходят проверку валидатором. Оба приводят к одному и тому же результату на странице. Но даже если вы еще слабо разбираетесь в HTML, вам должно быть очевидно, что первый из двух этих листингов выглядит аккуратнее и понятнее, его легче читать и редактировать.

Это происходит потому, что в первом фрагменте присутствует определенный стиль кодирования, т. е. принципы, в соответствии с которыми в коде расставляются пробелы, отступы и переводы строк.

Не существует никакого «самого правильного» стиля кодирования. Далее описан тот стиль, который предпочитает автор этой книги. Вы можете пользоваться им, можете позаимствовать стиль других разработчиков, а можете придумать свой собственный. Главное, чтобы этот стиль подчинялся единым логичным принципам.

Внимание!

Существует много удачных стилей кодирования, и вы можете выбрать любой из них. Но, выбрав свой стиль, строго придерживайтесь его в своих разработках.

Преимущества хорошего стиля кодирования проявляются наиболее ясно, когда вам приходится редактировать документ HTML, которым вы последний раз занимались несколько месяцев назад. Если этот документ достаточно сложный, в нем много уровней вложенных элементов, много специальных идентификаторов (значений атрибутов `id` и `class`) — при плохом стиле кодирования вы просто не сможете в нем разобраться.

Основные элементы

> И вот, Кеваддха, этот монах приблизился к великому Брахме и, приблизившись, сказал великому Брахме: «Где же, достопочтенный, уничтожаются без остатка эти четыре великих элемента, а именно: элемент земли, элемент воды, элемент огня и элемент воздуха?»
>
> *Кеваддха сутта*

Главными элементами в документе HTML являются `html`, `head`, `title` и `body`.

`html` — это так называемый *корневой элемент*. В шаблоне документа (см. листинг 5.1) он начинается в строке 04 и заканчивается в строке 16. Можно видеть, что `html` включает в себя все остальные элементы, которые есть в документе.

Атрибуты `lang` и `xml:lang` элемента `html` служат для указания языка, на котором написан документ. В данном случае предполагается, что документ будет по-русски, поэтому эти атрибуты имеют значение `"ru"`. Если документ написан на английском языке, их значение будет `"en"`.

Двухбуквенные коды некоторых естественных языков, установленные стандартом ISO 639 (см. **www.w3.org/WAI/ER/IG/ert/iso639.htm**), приведены в табл. 5.1.

Таблица 5.1. *Двухбуквенные коды некоторых языков*

Код	Язык	Код	Язык
en	английский	de	немецкий
ar	арабский	pl	польский
be	белорусский	ru	русский
el	греческий	sa	санскрит
da	датский	tr	турецкий
he	иврит	uk	украинский
es	испанский	fr	французский
it	итальянский	cs	чешский
zh	китайский	sv	шведский
la	латинский	ja	японский

纸 *Дополнительно*

Вы можете удивиться, что два атрибута элемента `html` выполняют одну и ту же функцию. Дело в том, что `lang` — это старый атрибут «исторического» HTML, который понимают браузеры прежних поколений. А `xml:lang` — это атрибут XML, он предназначен для более новых браузеров, а также поисковых систем и других средств, обрабатывающих документы HTML и XML. Следует использовать оба атрибута, чтобы информацию о языке воспринимали и старые, и новые браузеры.

Значением атрибута `xmlns` является идентификатор пространства имен XHTML (XHTML Namespace). Если вы хотите подробнее узнать, что такое этот идентификатор и зачем он нужен, обратитесь к стандарту «Namespaces in XML» (**www.w3.org/TR/REC-xml-names**).

Элемент `head` — это так называемая *шапка* документа. В ней содержится заглавие документа (элемент `title` — далее о нем будет рассказано подробнее), а также могут содержаться некоторые дополнительные элементы (например, `link`, как в листинге 5.1, или `meta`).

То, что находится в `head`, не отображается непосредственно на веб-странице (за исключением `title`), но предоставляет браузеру дополнительную информацию об этой веб-странице и о том, как ее отображать.

Наконец, `body` — это *тело* страницы. Именно внутри элемента `body` находится все то, что пользователь увидит непосредственно на веб-странице у себя в браузере. Подавляющее большинство элементов, о которых пойдет речь далее, должны располагаться в `body`.

Дополнительно

Если вы внимательно читали стандарт XHTML, то могли убедиться, что элемент `body`, как это ни странно, не является обязательным. Да, можно делать страницы без `body`. Вообще, единственным обязательным элементом документа, согласно стандарту, является `title`.

В самом деле, читайте официальные стандарты и другую документацию. Эта книга, как надеется автор, будет вам полезна, но она, как и любая другая книга на схожую тематику, не заменит стандартов.

Что касается стиля кодирования, то автор предпочитает размещать пролог и основные элементы именно так, как показано в листинге 5.1: первые две строки занимает пролог, затем идет пустая строка, затем — открывающий тэг `<html ... >` и т. д. Обратите внимание на пустые строки 03, 05, 11 и 15: они полезны для зрительного отделения друг от друга основных структурных частей документа.

Элементы `title` и `link`, вложенные в элемент `head`, набраны с отступом шириной в 2 пробела от левого края документа. Этот отступ помогает подчеркнуть, что элементы вложены в `head`. Вы увидите, что в дальнейшем этот подход распространяется и на остальные вложенные элементы.

Примечание

Не забудьте, что в этом разделе описывается тот стиль кодирования, которого придерживается автор. Этот стиль, разумеется, не является единственно правильным и обязательным к применению. Конечно, автор будет доволен, если вы выберете именно его для своих разработок. Но вам ничто не мешает использовать какой-нибудь другой стиль, а этот рассмотреть просто в качестве примера.

Вложенные элементы

— У меня смерть, — говорит он, — в таком-то месте; там стоит дуб, под дубом ящик, в ящике заяц, в зайце утка, в утке яйцо, в яйце моя смерть.

Русская народная сказка «Кощей Бессмертный»

Если в какой-либо элемент не вложено ничего, кроме текста, то весь этот элемент располагается на одной строке.

Рекомендуется	Не рекомендуется

`<p>Все смешалось в доме Облонских.</p>`	`<p>`
	`Все смешалось в доме Облонских.`
	`</p>`

`<td>2004</td>`	`<td>`
	`  2004`
	`</td>`

Обратите внимание: между тэгами и заключенным в них текстом не делается пробелов.

Рекомендуется	Не рекомендуется

`<title>Евгений Онегин</title>`	`<title> Евгений Онегин</title>`
	`<title>Евгений Онегин </title>`
	`<title> Евгений Онегин </title>`

Если в какой-либо элемент заключены другие элементы, то:

- открывающий и закрывающий тэги внешнего элемента располагаются в отдельных строках, первый символ закрывающего тэга расположен точно под первым символом открывающего;
- каждый внутренний элемент расположен в отдельной строке, его открывающий тэг расположен с отступом в 2 пробела от тэгов внешнего элемента.

Рекомендуется	Не рекомендуется

`<tr>`
`␣ ␣ <td>Mozilla</td>`
`␣ ␣ <td>251</td>`
`␣ ␣ <td>5,63</td>`
`</tr>`
	`<tr>`
	`<td>Mozilla</td>`
	`<td>251</td>`
	`<td>5,63</td>`
	`</tr>`

(окончание)

Рекомендуется	Не рекомендуется
	`<tr>`
	␣ ␣ `<td>Mozilla</td><td>251</td><td>5,63</td>`
	`</tr>`
`<div>`	␣ `</div>`
`...`	`...`
`</div>`	`</div>`
	`<div>`
	`...`
	␣ `</div>`

纸 *Дополнительно*

Если в элемент вложены и другие элементы, и просто текст, вопрос со строками и пробелами решается в зависимости от конкретной ситуации. Чаще всего в таких случаях все содержимое внешнего элемента целесообразнее располагать в одну строку, как обычный текст. Вот типичный пример такой ситуации — здесь элементы `strong` и `em` вложены в элемент `p`:

```
<p>17 декабря родился <strong>Людвиг ван Бетховен</strong>
(1770&#8211;1827), знаменитый немецкий композитор, написавший
9 симфоний, 5 концертов для фортепиано
с оркестром, десятки камерных произведений, песен, хоров. Среди
наиболее известных произведений Бетховена &#8212; соната
<strong>&laquo;Апассионата&raquo;</strong> для фортепиано
и <strong>9&#8209;я симфония</strong> &#8212; вершина твор-
ческой деятельности композитора. (<em>По материалам БСЭ</em>)</p>
```

Обратите внимание также на неразрывные пробелы и специальные подстановки для двух видов тире, кавычек и неразрывного дефиса.

Чтобы не запутаться с вложенностью элементов, рекомендуется набирать закрывающий тэг сразу после того, как набран открывающий, и только потом набирать то, что должно быть между этими тэгами.

В качестве примера рассмотрим таблицу из листинга 5.3, *а*. Для начала набираем открывающий тэг `<table>`, и сразу после него на следующей строке — парный ему закрывающий тэг `</table>`:

```
<table>
</table>
```

Обратите внимание: начало обоих тэгов расположено строго на одном вертикальном уровне!

Непосредственно в элемент `table` входят три других элемента: `thead`, `tfoot` и `tbody`. Набираем открывающие и закрывающие тэги этих элементов, не забывая делать перед ними отступ в 2 пробела:

```
<table>
␣␣ thead>
␣␣ </thead>
␣␣ <tfoot>
␣␣ </tfoot>
␣␣ <tbody>
␣␣ </tbody>
</table>
```

В элемент `thead` входит один элемент `tr`. Набираем его тэги, делая отступ еще в 2 пробела:

```
<table>
␣␣ <thead>
␣␣␣␣ <tr>
␣␣␣␣ </tr>
␣␣ </thead>
␣␣ <tfoot>
␣␣ </tfoot>
␣␣ <tbody>
␣␣ </tbody>
</table>
```

Наконец, в этот элемент `tr` входят три элемента `th`, а в каждый `th` — просто текст. Набираем и эти элементы, не забывая про отступы:

```
<table>
␣␣ <thead>
␣␣ <tr>
␣␣␣␣␣␣ <th>Браузер</th>
␣␣␣␣␣␣ <th>Количество посещений</th>
␣␣␣␣␣␣ <th>В % от общего количества посещений</th>
␣␣␣␣ </tr>
␣␣ </thead>
␣␣ <tfoot>
␣␣ </tfoot>
␣␣ <tbody>
␣␣ </tbody>
</table>
```

Аналогично набирая остальные вложенные `tr`, `td` и `th`, получаем полностью закодированную таблицу.

Описанный подход особенно полезен, если вы вкладываете друг в друга одноименные элементы (чаще всего элементы `div`). Если не следовать этому принципу, вы очень скоро запутаетесь, какой закрывающий `</div>` относится к какому открывающему `<div>`!

Комментарии

— А в армии я, потому что меня жена с тещей хотели в сумасшедший дом отдать. За убеждения. Видишь суслика?
— Нет...
— И я не вижу. А он есть!

И. Охлобыстин, Р. Качанов. ДМБ

Иногда бывает полезно сопровождать код HTML комментариями. Комментарии не обрабатываются браузером, они не видны конечному пользователю, но они могут оказаться важными для разработчика, который собирается редактировать код в будущем.

Комментарий в документе HTML начинается с конструкции `<!--` и заканчивается конструкцией `-->`. Внутри комментария не может быть двух дефисов подряд (`--`). За исключением этого ограничения, вы можете писать внутри комментария все, что вам заблагорассудится. То, что вы напишете, будет видно только тому, кто пожелает просмотреть исходный код документа HTML.

В листинге 5.3, *а* можно видеть два комментария, которые, вероятно, содержат важную для разработчика информацию.

Не рекомендуется злоупотреблять комментариями. Комментарии, как и любой другой код, занимают место и увеличивают размер файла. Три-четыре важных и не очень длинных комментария могут быть вполне допустимы, но если комментариев становится больше, чем основного кода, — разработчику следует пересмотреть свой стиль кодирования.

Дополнительно

Комментарии также бывают полезны при тестировании страницы. Если какой-то блок кода вызывает сомнения или просто мешает — можно «отключить» его, поставив перед ним `<!--`, а после него — `-->`. Браузер перестанет обрабатывать такой блок — как будто его вовсе нет в документе. Когда настанет время работать с кодом этого блока, его можно будет «включить» обратно, убрав символы комментария.

Примечание

Автор предпочитает снабжать собственные документы HTML комментарием, информирующим о назначении, дате последней редакции и некоторых других свойствах документа. Вот пример одного из таких комментариев — обычно он ставится после открывающего тэга `<html ... >` (листинг 5.4).

Листинг 5.4. Пример «заголовочного» комментария

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

<html lang="ru" xml:lang="ru" xmlns="http://www.w3.org/1999/xhtml">

<!--
     Сайт клуба «Школа сопереживаний». Главная страница
     www.teleska.ru/
     06.09.2003
-->

<head>
  <title>Школа сопереживаний. Телесно-ориентированная психотерапия</title>
...
```

Как показывает этот пример, один комментарий может занимать и несколько строк.

Дополнительно

Кстати, посмотрите внимательно на этот фрагмент кода. Он набран в кодировке UTF-8, поэтому в декларации XML нет атрибута `charset`. Эта кодировка позволяет включать специальные типографские символы непосредственно в документ. Например, здесь в документ напрямую включены кавычки-«елочки», для которых в расширенных кодировках ASCII пришлось бы использовать подстановки `«` и `»`.

Дополнительно

Присваивайте элементам осмысленные идентификаторы (значения атрибутов `id` и `class`). В листинге 5.3, *б* употребляются неудачные идентификаторы вроде `d4` и `d6t`. Они только затрудняют понимание кода. Гораздо лучше такие идентификаторы, как в листинге 5.3, *а*: `resume`, `header` и др. Они несут дополнительную информацию о смысле и назначении элементов.

Не присваивайте элементам идентификаторы, говорящие об их оформлении и расположении, — например, `green`, `left`, `small`. Лучше используйте идентификаторы, сообщающие о структурном или логическом значении элементов, — например, `heading`, `photo`, `comment`, `about` и т. д. Один и тот же элемент в зависимости от устройства просмотра и многих других условий может быть оформлен совершенно по-разному, но его структурное значение всегда останется неизменным.

Пробельные символы

Есть пробелы в памяти, бельма
На глазах: семь покрывал...
Я не помню тебя — отдельно.
Вместо черт — белый провал.

М. Цветаева. Поэма горы

К *пробельным символам* (white-space characters) в XHTML относятся:

- обычный пробел (U+0020);
- горизонтальная табуляция (U+0009);
- перевод строки (LF, U+000A);
- возврат каретки (CR, U+000D)

и еще некоторые более редкие символы.

Обычно пробельные символы обрабатываются браузерами не так, как большинство отображаемых символов.

Внимание!

Один или несколько следующих подряд пробельных символов в большинстве случаев отображаются как один обычный пробел.

Это значит, что в приведенном далее примере текст отобразится без увеличенных пробелов и переводов строки:

В коде HTML	В окне браузера
`<p>Если␣ ␣ ␣ хочешь␣ быть␣ красивым,␣ ␣ ␣ поступи␣ в␣ гусары.</p>`	Если хочешь быть красивым, поступи в гусары.

Эта особенность HTML позволяет использовать различные стили кодирования, расставляя в документе пробелы и переводы строк, которые предназначены исключительно для улучшения внешнего вида кода.

Тем не менее, следует избегать «посторонних» пробелов и переводов строки. В приведенном только что примере удвоенные и утроенные пробелы не вызывают никаких проблем, но и не являются необходимыми. В данном случае повторяющиеся пробелы лучше устранить. Не рекомендуется также оставлять пробелы в самом конце строки (как в первой строке кода примера).

Дополнительно

Спецификация HTML утверждает, что пробелы, следующие непосредственно после открывающего тэга и непосредственно перед закрывающим, должны игнорироваться. Следовательно, код

```
<p>Это<a␣ href="file.html">␣ гиперссылка␣ </a>.</p>
```

должен отобразиться следующим образом:

Этогиперссылка.

Не все браузеры следуют этому правилу, и вышеприведенный код часто отображается так:

Это гиперссылка .

Чтобы избежать неприятностей в подобных ситуациях, не ставьте пробелов непосредственно после открывающих тэгов и непосредственно перед закрывающими.

Посторонние пробелы «выскакивают» и в некоторых других случаях — например, при заключении изображения в блочный элемент:

```
<div>
  <img src="lev.jpg" width="250" height="374"
alt="Лев Толстой. Фото 1907 г." />
</div>
```

Некоторые браузеры отображают переводы строк после открывающего тэга `<div>` и перед закрывающим тэгом `</div>` как пробелы, в результате чего вокруг изображения появляются ненужные отступы. Чтобы устранить эту проблему, имеет смысл обойтись без переводов строк:

```
<div><img src="lev.jpg" width="250" height="374"
alt="Лев Толстой. Фото 1907 г." /></div>
```

Имена файлов HTML

Придумай имя для птицы,
Имя для зверя,
Для камня, для леса.
Придумай птицу для рыбы,
Рыбу для моря,
Море для неба...

О. Арефьева. Придумай имя

Многие начинающие разработчики задаются вопросом о том, какие имена следует присваивать файлам HTML. Особенно часто интересуются, какая разница между расширениями `htm` и `html`.

Наиболее универсальный ответ на все вопросы об именах файлов звучит так: допустимые имена файлов зависят от настроек, установленных на веб-сервере, где предполагается разместить сайт. Какими должны быть имена файлов — следует спрашивать у администратора вашего веб-сервера.

Впрочем, в большинстве случаев достаточно придерживаться нескольких простых рекомендаций:

- используйте в именах файлов и каталогов только латинские буквы (желательно только строчные) и цифры. Не стоит называть файл `Страница.html` или вовсе `Мой␣сайт.html` (с пробелом посередине);
- старайтесь давать файлам и каталогам короткие имена. Когда-то, во времена операционной системы DOS, длина имени файла не должна была превышать 8 символов. Технически это ограничение уже давно не действует, но на практике, чем короче имя файла, тем легче его запомнить и с ним работать;
- вместе с тем, делайте имена файлов и каталогов достаточно информативными. Не называйте файлы 01.html или a1b2c3.html. Гораздо удачнее для файлов будут такие имена, как news.html и contacts.html, а для каталогов — products или links.

Примечание

В большинстве случаев между расширениями htm и html нет никакой разницы, и вы можете присваивать своим файлам любое из них. Автор предпочитает расширение html — в самом деле, ведь язык разметки называется HTML, а не HTM!

Некоторые веб-серверы настроены так, что по-разному обрабатывают файлы с расширениями htm и html. Такое бывает достаточно редко, но все же поинтересуйтесь об этом у администратора того сервера, на котором вы планируете разместить свой сайт.

Дополнительно

На самом деле, если пользователь загружает веб-страницу из Сети, а не со своего собственного диска, расширение файла чаще всего вообще не играет роли. Получая страницу от веб-сервера, браузеры ориентируются вовсе не на название или расширение файла, а на заголовки протокола HTTP. Поэтому, если сервер настроен сответствующим образом, имя файла может заканчиваться хоть на .123 или .xyz или вовсе не иметь расширения — браузеру это будет безразлично.

На крупных, профессиональных сайтах большинство страниц вообще не хранятся на веб-сервере, а каждый раз генерируются заново по запросу пользователя. Таким образом, пользователь получает данные, которые создаются «на лету» и вообще не имеют никакого имени! Тем не менее, именно благодаря заголовкам HTTP браузер воспринимает эти данные как документ HTML и отображает веб-страницу, которая получается из этого документа.

5.1.4. Строительный материал
Структурные элементы HTML

> Только после явления Неба и соединения Земли с Небом можно говорить о реальном начале космогонического процесса, в результате которого возникают все структурные элементы мироздания.
>
> *М. Евзлин. Космогония и ритуал*

Здесь будут перечислены основные структурные элементы HTML и даны рекомендации о том, как их лучше всего кодировать.

Строчные и блочные элементы

> Все люди делятся на две группы: те, кто делит людей на две группы, и те, кто этого не делает.
>
> *Анекдот*

Все элементы в XHTML делятся на две категории:

- строчные (inline);
- блочные (block-level).

Строчные элементы — это элементы уровня текста, то есть такие, которые могут находиться внутри текстовой строки. *Блочные элементы* обычно образуют самостоятельные блоки в документе (рис. 5.3).

Рис. 5.3. *а* — строчный элемент; *б* — блочный элемент

Примечание

Простое правило, позволяющее отличать строчные элементы от блочных: если в документе HTML поместить несколько строчных элементов подряд — то они выстроятся друг за другом на одной строке. А если поместить несколько блочных элементов подряд — то они разместятся друг под другом.

Если вы не знаете, строчным или блочным является тот или иной элемент, — проведите эксперимент: повторите этот элемент в коде HTML несколько раз подряд и посмотрите, как он будет вести себя при просмотре в браузере.

Дополнительно

Далеко не всегда строчные элементы отображаются частью строки, а блочные — отдельными блоками. С помощью CSS и тем, и другим элементам можно придать какой угодно внешний вид. Деление элементов на строчные и блочные соответствует скорее не тому, как эти элементы отображаются, а тому, какую структурную роль в документе они выполняют (табл. 5.2).

Таблица 5.2. *Некоторые строчные и блочные элементы XHTML*

Строчные элементы	Блочные элементы
a	blockquote
em	div
img	h1, h2, h3, h4, h5, h6
span	ol
strong	p
sub	pre
sup	table
	ul

Внимание!

Блочные элементы не могут быть вложены в строчные.

Синтаксические правила XHTML, описанные в DTD, налагают также много других ограничений на вложенность элементов. Например, внутри элемента `a` не может быть других элементов `a`, внутри элемента `ul` должен быть хотя бы один элемент `li` и т. д. Некоторые из таких ограничений будут описаны далее в этой главе. Но если вы хотите получить самую надежную информацию о том, какие элементы могут или не могут быть вложены друг в друга — читайте официальную документацию Консорциума W3C.

Дополнительно

Одна из строк DTD XHTML 1.0 Strict гласит:

```
<!ELEMENT body %Block;>
```

Это значит, что непосредственно в элемент `body` могут быть вложены только блочные элементы — `p`, `h1`, `div` и т. д. Обычный текст, изображение или гиперссылка не могут находиться внутри элемента `body`, если они не заключены в какой-либо блочный элемент.

Допустимо	Недопустимо

`<body>`
`...`
`<p>Все смешалось в доме Облонских.</p>`
`...`
`</body>`
	`<body>`
	`...`
	`Все смешалось в доме Облонских.`
	`...`
	`</body>`

`<body>`
`...`
`<div class="image"><img src="lev.jpg" width="250" height="374" alt="Лев Толстой. Фото 1907 г." /></div>`
`...`
`</body>`
	`<body>`
	`...`
	`<img src="lev.jpg" width="250" height="374" alt="Лев Толстой. Фото 1907 г." />`
	`...`
	`</body>`

title — самый главный элемент

Представителей одной из групп безумцев от литературы Ломброзо назвал графоманами. К графоманам он причислял и тех, кто дает своим книгам чрезмерно длинные заглавия. Как пример приводит он попавшуюся ему в руки книгу, заглавие которой состояло ни больше ни меньше, как из восемнадцати строк!

Рат-Вег Иштван. Комедия книги

Элемент `title` — это *заглавие* (или *название*) страницы. Пожалуй, это самый важный элемент, поэтому ему стоит уделять особое внимание.

Важность `title` становится понятна, если вспомнить, что именно заглавие страницы фигурирует в результатах поисковых систем и в каталогах ресурсов Интернета. Удачное заглавие помогает странице попасть в верхние позиции этих результатов и каталогов, а пользователю — выбрать одну страницу из длинного списка. Неудачное заглавие может привести к тому, что пользователь вообще не узнает о существовании данной страницы (рис. 5.4).

Примечание

Попробуйте рассуждать как пользователь. Вот вы вводите в поисковой системе запрос, и она выдает вам несколько сотен результатов. Вряд ли вы будете открывать все эти страницы подряд одну за другой. Гораздо вероятнее, что вы, глядя на страницу с результатами, сразу отбросите одни из них, а другие, наоборот, сочтете достойными проверки. Обратите внимание: в этот момент вы делаете вывод о том, интересна ли вам та или иная страница, не видя самой этой страницы, а только читая ее заглавие в результатах поиска!

Идеальное заглавие страницы должно быть таким, чтобы, увидев его в списке среди множества других заглавий, пользователь захотел щелкнуть именно по нему.

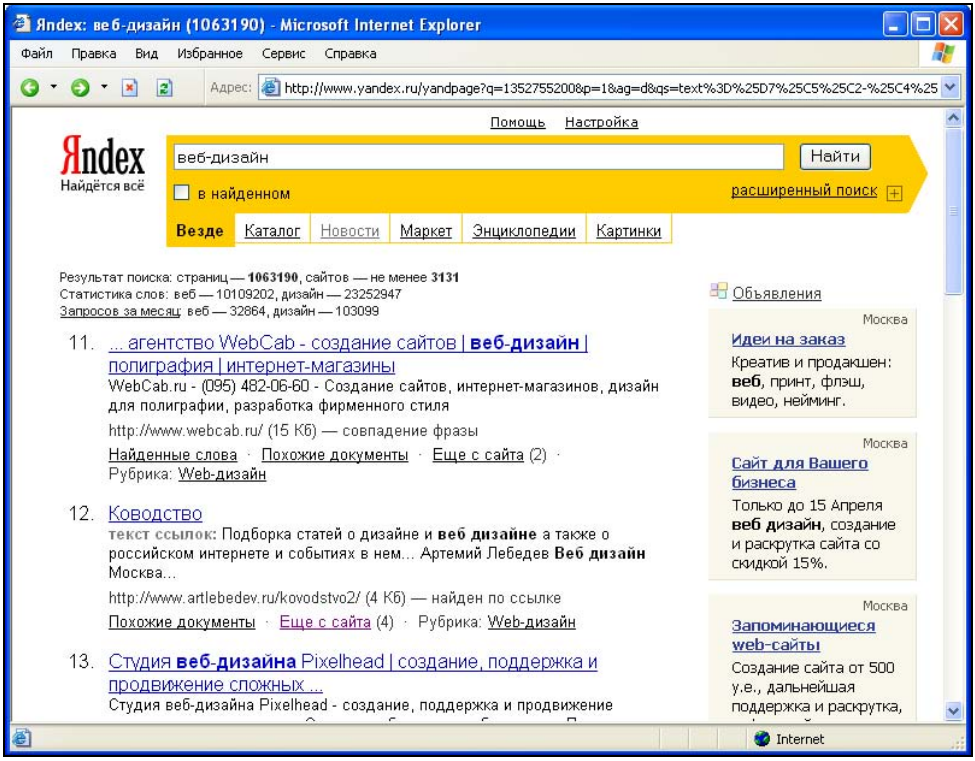


Рис. 5.4. Если в поисковой системе Яндекс ввести запрос `веб-дизайн` — в результатах поиска не окажется сайта той самой студии, которая разработала Яндекс! Только на 12-м месте можно увидеть ссылку на одну из внутренних страниц этого сайта

Заглавие должно быть коротким. Чем короче заглавие, тем охотнее пользователь его прочтет и тем вероятнее он его запомнит. Это относится не только к заглавию сайта, но и к любому другому тексту.

Длина заглавия ограничена также техническими условиями. Если окно браузера открыто на полный экран, то в верхней строке окна умещается 75—100 символов. Если учесть, что пользователи не всегда разворачивают браузер на весь экран, то ограничение длины становится еще более критичным.

Количество символов	Рекомендации
30 и меньше	Идеально
31–50	Допустимо
50–70	Допустимо в некоторых специальных случаях
71 и больше	Не рекомендуется

Дополнительно

Имеет смысл ограничивать не только количество символов в заглавии, но и количество слов — прежде всего значащих (т. е. всех слов, кроме частиц, союзов, предлогов и других служебных частей речи).

Количество значащих слов	Рекомендации
1–3	Идеально
4–5	Допустимо
6–7	Допустимо в некоторых специальных случаях
8 и больше	Не рекомендуется

В самом деле, чем меньше слов, тем легче читатель «схватывает» фразу, тем прочнее она запоминается. Вы же наверняка хотите, чтобы пользователь обратил внимание на заглавие вашей страницы!

Заглавие должно быть содержательным. Не забывайте, что во многих случаях пользователь именно по заглавию делает вывод о том, стоит ли ему открывать страницу. Если заглавие не будет соответствовать содержанию страницы — пользователь будет обманут. Либо он, увидев заглавие, откроет эту страницу, увидит, что на самом деле она ему не нужна, и разочаруется. Либо, наоборот, он посмотрит на заглавие и подумает, что эта страница ему неинтересна, — а на самом деле, если бы он ее открыл, он бы не пожалел.

Внимание!

Заглавие страницы (элемент `title`) — это, по существу, кратчайшее описание всего содержания страницы. Оно должно так полно, как только возможно, информировать пользователя о содержании страницы.

Недопустимы заглавия вроде: «Добро пожаловать!», «Главная страница», «Начало», «Здравствуйте» и т. д. Это все «пустышки», которые существенно ухудшают эффективность сайта.

Рассмотрим пример того, как можно улучшить неудачное заглавие сайта. Допустим, для сайта фирмы «Юникомп», занимающейся продажей компьютеров и комплектующих, предложено заглавие:

```
Официальный сайт фирмы «Юникомп»
```

В чем недостатки этого заглавия? Во-первых, совершенно ни к чему слова «официальный сайт». В самом деле, зачем на сайте писать, что это сайт? Неужели найдется пользователь, который не догадается, что перед ним именно сайт, а не, например, книга или радиопередача?

«Официальный» статус сайта тоже не должен нуждаться в специальном описании. Обычно характер информации, представленной на сайте, а также его структура и оформление дают ясно понять, что это именно официальный сайт. Возможно, указание на «официальность» было бы уместно на сайте какого-нибудь популярного эстрадного исполнителя (чтобы отличать его собственный сайт от любительских сайтов, сделанных поклонниками), но маловероятно, чтобы у компьютерной фирмы появились «неофициальные» сайты.

Таким образом, слова «официальный сайт» устраняются. Заглавие приобретает следующий вид:

```
Фирма «Юникомп»
```

К сожалению, из такого заглавия совершенно непонятно, чем занимается эта фирма. Поэтому добавим в заглавие содержательную часть:

```
Фирма «Юникомп». Продажа компьютеров и комплектующих
```

纸 *Дополнительно*

Посмотрите: мы разделили две части заглавия точкой, как наиболее типографски правильным символом. Многие разработчики, следуя моде, поставили бы на этом месте дефис, тире или двоеточие. Хотя каждый из этих символов в принципе допустим, именно точка в наибольшей степени отвечает отечественным типографским традициям.

Теперь посчитаем символы, входящие в заглавие. Получается 52 символа, а это уже немного выходит за разумные пределы (50 символов). Что же можно убрать из заглавия, чтобы при этом не нарушить его смысл? Прежде всего, стало совершенно ненужным слово «Фирма». Такие общие слова — «фирма», «компания», «предприятие» и т. д. — чаще всего не сообщают пользователю никакой полезной информации, так что без них практически всегда можно обойтись.

Наконец, можно избавиться от слова «продажа». В самом деле, если известно, что фирма занимается компьютерами и комплектующими, пользователь обычно ожидает, что она их именно продает (а также — что возможно, но необязательно — собирает и ремонтирует). Наоборот, если бы фирма имела

отношение к компьютерам, но занималась бы при этом не продажей, а чем-то другим, — вот тогда об этом следовало бы написать в заглавии.

Предпримем еще одно маленькое техническое усовершенствование: уберем кавычки, ставшие ненужными, и поставим слова «компьютеры» и «комплектующие» в именительный падеж:

```
Юникомп. Компьютеры, комплектующие
```

Получается 34 символа. Это немного не дотягивает до идеала, но все же существенно лучше, чем 52.

Внимание!

Заглавие должно начинаться с самого важного слова или словосочетания. Например, заглавие на главной странице сайта фирмы или организации должно начинаться с названия этой организации.

Рекомендуется	Не рекомендуется
Бизон. Интернет-магазин	Интернет-магазин «Бизон»
Всероссийское общество филателистов	Сайт Всероссийского общества филателистов
Kulina.Ru: Кулинарные рецепты	Сайт кулинарных рецептов KULINA.RU

Дополнительно

Добивайтесь, чтобы все важные слова в заглавии были в именительном падеже. В самом деле, в поисковых системах обычно ищут «кулинарные рецепты», а не «кулинарных рецептов» или «кулинарным рецептам».

Для большинства заглавий также будут полезны дополнительные рекомендации:

- соблюдайте единообразие заглавий на всех страницах сайта. Плохо, если одна страница озаглавлена «Юникомп. Новости», другая — «Специальные предложения фирмы „Юникомп”», а третья — «Гостевая книга — Юникомп». Гораздо лучше назвать все страницы в едином стиле, например: «Юникомп. Новости», «Юникомп. Специальные предложения», «Юникомп. Гостевая книга»;

Примечание

Нужно соблюдать не только смысловое, но и синтаксическое единообразие. Например, не рекомендуется называть одну страницу «Юникомп. Новости»,

а другую — «Юникомп — специальные предложения». И то, и другое синтаксически допустимо, но остановитесь на чем-нибудь одном: либо везде разделяйте заглавие точкой, либо везде — тире, либо еще чем-то, но везде одинаково!

- не украшайте заглавия посторонними символами, как, например:

  << ПОРТАЛ >>

  :: Юникомп :: Копмьютеры :: Комплектующие ::

  ~Зеленая~ ~Волна~

  Каждый лишний символ заглавия уменьшает вероятность того, что пользователь заинтересуется именно вашей страницей, увидев ее в списке среди прочих. Наоборот, вы должны по возможности избавить заглавие от всех символов, без которых можно обойтись, — скажем, в приведенном примере мы из этих соображений убрали кавычки. Конечно, не надо пренебрегать правилами русского языка и убирать, например, скобки или запятые там, где они нужны, — но всюду, где позволяет грамматика, следует устранять любые символы, без которых можно обойтись;

  Дополнительно

  Кроме того, во многих случаях использование посторонних символов вызывает проблемы при индексации страниц поисковыми системами. «Украшение» заглавий — верный способ добиться, чтобы на первые места в результатах поиска вышли сайты ваших конкурентов.

- не используйте в заглавиях метафоры, аллегории, иносказания, рекламные слоганы и т. д. Заглавие должно быть информативным, а не вызывающим.

  В самом деле, представьте себе, что вы ищете в Сети информацию о фирмах, торгующих компьютерами в кредит. Среди результатов поиска вы видите два таких заглавия:

  НЬЮКОМ — Добро пожаловать в реальный мир!

  Юникомп. Компьютеры в кредит

  Если вас интересует именно продажа в кредит — по какой из этих ссылок вы щелкнете в первую очередь?

  Дополнительно

  Кстати, последний вопрос не совсем корректен. Есть заметная разница между тем, что пользователь предполагает о своих действиях на сайте, и тем, что он реально делает, посетив этот сайт. Например, при беглом взгляде на сайт пользователь одобряет оригинальное оформление — а поработав с сайтом несколько раз, убеждается, что это оформление неудобно и мешает ему выполнять свои задачи. Бывает и наоборот: например, при взгляде на сайт пользователь считает, что его цветовая гамма неудачна — а в то же время пользо-

ватели, которые посещают этот сайт по делу, вообще не обращают внимания на цветовую гамму и не испытывают никаких затруднений из-за нее.

Поэтому только реальная практика использования сайта в состоянии показать его достоинства и недостатки. Разработчику могут помочь квалифицированные советы и рекомендации специалистов, справочники и учебники (вроде этой книги) и другие источники информации, но все это — только вспомогательные средства. Главное, на основании чего разработчик должен делать выводы, — это реальный опыт посещения сайта реальными пользователями, которые пришли туда для выполнения своих практических задач, а не просто поглазеть.

Элемент `title` может включать в себя только текст. Внутри `title` не может быть других элементов.

Примечание

Не пишите в заглавии главной страницы слова «Главная страница», «Домашняя страница», «Начало» и им подобные. Это все равно что на обложке книги писать «Обложка».

Заголовки: *h1 — h6*

> Ему бросается в глаза заголовок первой статьи «Ребенок как творец», он читает первые строки и начинает плакать от радости.
>
> *Ф. Кафка. Дневники*

Заголовки — это опора всей структуры документа. Они имеют первостепенное значение для всех, кто знакомится с содержанием страницы:

- пользователь сначала окидывает страницу беглым взглядом и обращает внимание только на наиболее заметные ее элементы — в первую очередь, на заголовки. Ознакомившись с заголовками, он решает, стоит ли читать текст под ними;
- поисковые системы и каталогизаторы ресурсов, занося страницу в свои базы данных, придают заголовкам большое значение. От того, что написано в заголовках, зависит, в какую тематическую категорию будет отнесена страница.

Примечание

Заголовок и заглавие — разные вещи. В традиционной печати заглавием называется название целой книги, а заголовками — обозначения ее отдельных структурных частей (разделов, глав, параграфов, таблиц и т. д.). В этой книге значение слов «заглавие» и «заголовок» распространяется на веб-страницы похожим образом: под *заглавием* понимается элемент `title` (т. е. то, что

обычно отображается в верхней строке окна браузера), а под *заголовками* — элементы `h1` – `h6` (присутствующие на самой веб-странице).

В XHTML 1.0 существует шесть уровней заголовков. Заголовок первого уровня — `h1` — это заголовок целого документа. Такой заголовок рекомендуется размещать на каждой веб-странице.

Внимание!

Помещайте на каждой странице заголовок первого уровня (`h1`). На странице должен быть только один заголовок первого уровня.

Дополнительно

Во многих случаях заголовок первого уровня (`h1`) совпадает с заглавием страницы (`title`). Например, на первой странице персонального сайта и заглавие, и основной заголовок, скорее всего, будут одинаковыми — «Василий Пупкин». Но, скажем, если страница содержит текст известного романа в стихах, то ее заглавие будет «А. С. Пушкин. Евгений Онегин», а заголовок на странице — «Евгений Онегин». На самой странице имя автора не будет внесено в заголовок, а составит содержание другого элемента (вероятнее всего, `address`).

Если страница разделена на равноправные структурные части, и каждая из этих частей имеет собственный заголовок — это заголовки второго уровня (`h2`). Если, в свою очередь, какие-либо из этих структурных частей тоже разделены на еще более мелкие части — то каждая из таких частей имеет заголовок третьего уровня (`h3`) и т. д.

Проводя аналогию с книгой, можно сказать, что `h1` — это заголовок целой книги, `h2` — это заголовок части, `h3` — заголовок главы и т. д.

Разработчики языков HTML и XHTML посчитали, что заголовки мельче шестого уровня вводить нецелесообразно. В самом деле, на практике даже `h5` используются крайне редко.

Внимание!

Соблюдайте иерархию заголовков. В документе не может быть заголовков третьего уровня (`h3`), если отсутствуют заголовки второго уровня (`h2`).

Примечание

Некоторые начинающие разработчики совершают такую ошибку: допустим, если веб-страница состоит из нескольких коротких заметок с заголовками, они

выделяют заголовок первой заметки тэгами `<h1>...</h1>`, заголовок второй заметки — `<h2>...</h2>`, третьей — `<h3>...</h3>` и т. д. На самом деле элемент `h1` должен быть заголовком всей страницы целиком, а заголовок каждой заметки в данном случае — `h2`.

Основное отличие заголовков на веб-странице от заголовков в печати — это повышенное требование к их конкретности и информативности. В газетах и журналах заголовки обычно делаются броскими, кричащими, они рассчитаны на привлечение внимания читателя. «Сенсация!», «Вы такого еще не видели!», «Политический скандал века!» — это, возможно, хорошо для желтой прессы, но не для сайта. Заголовок на веб-странице должен *информировать*, а не шокировать пользователя.

Внимание!

Заголовки должны передавать конкретную информацию о содержании возглавляемого ими текста. В заголовках нет места преувеличениям и эмоциям.

Рекомендуется	Не рекомендуется
Пожар уничтожил 15 гектаров леса	Разгул огненной стихии
Mozilla 1.8: улучшенная защита от спама в почтовом клиенте	Новый браузер Mozilla 1.8
Результаты чемпионата: Россия на первом месте	Наши взяли золото!

Дополнительно

Если на сайте публикуется статья или заметка о каком-либо событии, заголовок лучше сделать утвердительным предложением, выражающим основную тему этого события. Например, заголовок «В Москве состоялась встреча министров иностранных дел» удачнее, чем «Встреча министров иностранных дел в Москве». Второй заголовок ничего не говорит о том, была ли эта встреча или только будет; чтобы узнать это, пользователю придется читать текст статьи. А первый заголовок сразу сообщает наиболее важную информацию. Если пользователю нужно было всего лишь узнать, прошла ли уже встреча министров, он сэкономит свое время и скажет спасибо разработчику сайта.

В самом деле, заголовок должен не только *обозначать* какую-то информацию, но и непосредственно *передавать* ее. Пользователи предпочитают сайты с информативными заголовками, на которые достаточно бегло взглянуть, чтобы получить представление о содержании сайта. Если заголовки не позволяют получить такого представления или, что еще хуже, вводят пользователя в заблуждение, — он попросту уйдет на другой сайт, который больше

его устраивает. Неужели вы хотите, чтобы пользователь ушел с вашего сайта к вашему конкуренту?

Примечание

Буква «h» в именах элементов `h1 – h6` происходит от слова «heading» («заголовок»).

Для большинства заголовков также будут полезны дополнительные рекомендации:

- не делайте заголовки графическими. Текст вообще не следует заключать в изображения, а тем более такой важный текст, как заголовки. Помимо известных затруднений для пользователя (*см. разд. 1.1.2*), графические заголовки не индексируются поисковыми системами;
- не подменяйте элементы `h1 – h6` другими элементами, обеспечивающими такой же внешний вид. (Например, некоторые разработчики, чтобы сделать заголовок, пишут: `<b><font size="+3">Евгений Онегин</font></b>`.) Внешний вид — это еще не весь заголовок. Если вы не используете элементы `h1 – h6`, ваши «заголовки» будут выделены только в графических браузерах, но останутся обычным текстом в текстовых и речевых браузерах, а также (что еще более важно) — поисковые системы тоже сочтут их обычным текстом;
- в заголовках, как и в заглавиях, не рекомендуется использовать метафоры и иносказания. Не надо называть статью о реструктуризации государственного долга «Долг платежом красен» или «Где деньги, Зин?». Иначе эта статья рискует оказаться в поисковых системах в тематическом разделе «Русские пословицы и поговорки» или «В. С. Высоцкий».

Дополнительно

Если заголовок слишком длинный, или если пользователь установил слишком малую ширину окна браузера, то заголовок будет размещаться в нескольких строках. Нужно следить за тем, чтобы разбиение на строки происходило в соответствии со смыслом заголовка:

Рекомендуется	Не рекомендуется
Песни нашего века	Песни нашего века
Красная Шапочка и Серый Волк	Красная Шапочка и Серый Волк
Фильму «Унесенные ветром» — 80 лет	Фильму «Унесенные ветром» — 80 лет

Чтобы добиться такого эффекта, можно разбить слова в заголовке на 2–3 единые группы, связанные неразрывными пробелами — например, так:

```
<h1>Красная Шапочка и Серый Волк</h1>
```

В этом случае, если заголовок не будет умещаться на одной строке, он разобьется только в разрешенном месте — после слова «Шапочка».

Внутрь заголовка можно помещать только строчные элементы. Например, между тэгами `<h1>...</h1>` можно вставить `<em>...</em>`, `<span>...</span>` или `<a href="...">...</a>` (если эти элементы находятся на своем месте согласно структуре документа), но нельзя вставить `<div>...</div>`, `<table>...</table>` или, например, другой заголовок.

Дополнительно

Особая проблема — подзаголовки. В HTML нет специального тэга для подзаголовков, поэтому кодировщикам приходится придумывать обходные пути. Например, один из возможных вариантов выглядит так:

```
<div class="h1">
  <h1>Глава 5. Наука побеждать</h1>
  <div class="subtitle">Техника разметки и оформления сайтов</div>
</div>
```

Естественно, имена классов `h1` и `subtitle` здесь не имеют никакого особенного значения и применяются только для связи документа с таблицей стилей.

Текст: *p* и *pre*

> Кстати, сверхпопулярная ныне эпопея о Гарри Поттере удостоилась издевательского антиприза «Абзац» — за худший перевод и худшую редактуру.
>
> *А. Немзер. Дело привычное*

Элемент `p` — это абзац обычного текста. Например, текст, который вы читаете прямо сейчас, в документе HTML следовало бы заключить в тэги `<p>...</p>`.

Многие разработчики пренебрегают элементом `p` там, где он должен быть, и заключают текст в другие элементы (например, в `<div>...</div>`). Это не такая серьезная ошибка, как применение элементов `font` и `b` вместо `h1`, но ее все же рекомендуется избегать.

Примечание

Название элемента `p` происходит от «paragraph». (Заметим, что это слово переводится как «абзац». Знак параграфа — §, U+00A7 — по-английски называется не «paragraph», а «section sign».)

Практически любой связный текст, будь то статья, объявление, аннотация или еще что-то подобное, состоит из абзацев. Такие абзацы должны быть закодированы с помощью элементов `p`.

Дополнительно

Некоторые разработчики применяют `div` вместо `p`, потому что в большинстве браузеров элемент `p` по умолчанию отображается с небольшими вертикальными отступами перед и после. Это не должно останавливать вас от грамотной разметки: ничто не мешает отменить эти отступы средствами CSS.

Элемент `pre` — более редкий. Он обозначает *преформатированный текст* (preformatted text), т. е. такой текст, в котором должны сохраняться все пробелы, табуляции и переводы строк. Когда-то этот элемент широко применялся для текстовых таблиц и схем (см. рис. 3.8), но со временем необходимость в таких суррогатах отпала. Хотя осталось немало других случаев, когда целесообразно употреблять элемент `pre`:

- **для выделения программного кода и подобного текста** — если вы размещаете на веб-странице фрагменты исходного кода программ или файлов HTML, их следует заключать в элемент `pre`. В противном случае будет потерян стиль кодирования, а иногда могут даже возникнуть синтаксические ошибки;

  Дополнительно

  Не забывайте, что в исходном коде, размещенном на веб-странице, следует применять специальные подстановки для служебных символов HTML. Например, если вы хотите, чтобы пользователь увидел в точности такой текст:

  <h2 id="ch2">Глава вторая</h2>

  — то его нужно закодировать следующим образом:

  ```
  &lt;h2␣ id=&quot;ch2&quot;&gt;Глава␣ вторая&lt;/h2&gt;
  ```

  (если вы помещаете этот текст не в элемент `pre`, то вместо обычных пробелов нужны неразрывные).

- **для стихотворного текста** — в стихах, как и в кодах программ, пробелы и переводы строк являются важной частью содержания, и элемент `pre` позволяет их сохранить — например, так:

```
<pre>
Александр Сергеевич,
                    разрешите представиться.
                                            Маяковский.
Дайте руку!
         Вот грудная клетка.
```

```
                                Слушайте,
                                        уже не стук, а стон;
тревожусь я о нем,
                    в щенка смиренном львенке.
...
</pre>
```

Именно стихотворный текст приводится в официальной спецификации HTML в качестве примера грамотного использования элемента `pre`.

Иногда для стихов используют другие способы кодирования — например, разрывают строки с помощью `<br />`, а для отступов применяют неразрывные пробелы или CSS. Все эти способы приводят к неоправданному «разбуханию» кода. Например, в приведенном примере, чтобы обойтись без `pre`, следует вставить девять `<br />` и 160 (!) неразрывных пробелов, что приведет к увеличению объема текста в целых 4 раза!

Примечание

Графические браузеры по умолчанию отображают текст, заключенный в `pre`, моноширинным шрифтом. Однако для стихов лучше установить в таблице стилей какой-нибудь более подходящий шрифт.

Дополнительно

Обратите внимание: если заключить текст в `pre`, как показано в последнем примере, то перед этим текстом и после него появятся вертикальные отступы. Это неудивительно: ведь `pre` действительно сохраняет *все* переводы строк — в т. ч. те, которые находятся сразу после открывающего тэга `<pre>` и сразу перед закрывающим тэгом `</pre>`. Если вы хотите избавиться от отступов, не ставьте этих переводов строк:

```
<pre>Александр Сергеевич,
                    разрешите представиться.
                                        Маяковский.
...
Обожаю
       всяческую жизнь!</pre>
```

Элемент `pre` не только сохраняет существующие разрывы строк, но и не вставляет новых. Если строка слишком длинная и не помещается в окно браузера, она не разобьется, как в случае элемента `p`, а уйдет за границу окна. При этом появится горизонтальная полоса прокрутки.

Примечание

Вообще говоря, горизонтальная полоса прокрутки обычно неудобна для пользователя. Добивайтесь того, чтобы при наиболее распространенных режимах просмотра ваших страниц эта полоса не возникала.

Внутритекстовые выделения: *em* и *strong*

> Иногда слова, напечатанные курсивом, много несправедливее тех, которые напечатаны прямым шрифтом.
>
> *К. Прутков. Плоды раздумья*

Особенно важные участки текста, на которых должно быть сделано логическое ударение, следует выделять с помощью элементов `em` и `strong`.

Современные графические браузеры отображают элемент `em` курсивом, а элемент `strong` — полужирным начертанием. Но это не значит, что `<em>...</em>` — то же самое, что `<i>...</i>`, а `<strong>...</strong>` — то же самое, что `<b>...</b>`. Элементы `em` и `strong` указывают прежде всего на *структурное* выделение участка текста, а это позволяет выделять эти элементы даже тем браузерам, которые не поддерживают курсивного и полужирного начертания.

Элемент `em` рекомендуется использовать тогда, когда имеется в виду интонационное выделение, например:

В коде HTML	В окне браузера
`<p>Это <em>мой</em> брат.</p>`	Это *мой* брат.
`<p>Это мой <em>брат</em>.</p>`	Это мой *брат*.

Благодаря логическому выделению два этих предложения несут различную смысловую нагрузку: в первом автор специально обращает внимание читателя на то, что это его брат, а не еще чей-нибудь; во втором автор отмечает, что имеется в виду брат, а не, например, отец или дядя.

Примечание

Название элемента `em` происходит от «emphasis» («ударение», «акцент», «выделение»).

Дополнительно

Элемент `em` имеет смысл использовать и во многих других случаях, например:

- для выделения имени собственного, указанного на языке оригинала:
- …изобретатель HTML Тим Бернерс-Ли (*Tim Berners-Lee*) предполагал, что…
- для выделения дополнительной информации, относящейся к загружаемому файлу:

  Симфония № 5 (*MP3, 11 Мб*)

Элемент `strong`, являющийся более сильным выделением, чем `em`, можно использовать для акцентирования ключевой информации. Типичный пример применения `strong` — выделение наиболее важных слов в информационной заметке:

> **Выборы мэра г. Озерецка** признаны состоявшимися. В голосовании приняло участие более 70 % избирателей. По предварительным результатам, нынешний градоначальник **Василий Пупкин** значительно опережает своих соперников.

Дополнительно

Не забывайте, что знаки препинания и скобки, окружающие выделенный участок текста, сами не выделяются. А кавычки, наоборот, выделяются (*см. разд. 2.1.13*):

Правильно	Неправильно
`<em>Кегль</em>, или размер шрифта…`	`<em>Кегль,</em> или размер шрифта…`
`Выделение <strong>полужирным шрифтом</strong>.`	`Выделение <strong>полужирным шрифтом.</strong>`
`Размер (<em>кегль</em>) шрифта…`	`Размер <em>(кегль)</em> шрифта…`
`Фильм <strong>«Властелин колец»</strong>`	`Фильм «<strong>Властелин колец</strong>»`

Не стоит злоупотреблять логическими выделениями. Если в тексте выделено слишком много отдельных участков, в результате ни один из них не привлекает к себе внимание читателя. Рекомендуется использовать не более одного-двух выделений в пределах каждого абзаца текста.

Другие внутритекстовые выделения

И долго я стоял у речки,
И долго думал, сняв очки:
«Какие странные дощечки
И непонятные крючки!»

Д. Хармс. Что это было?

Сейчас пойдет речь о выделениях, которые используются еще реже, чем `em` и `strong` — но не потому, что менее актуальны, а потому, что многие разработчики просто не подозревают об их существовании. К таким выделениям относятся внутритекстовые (табл. 5.3), а также нижний и верхний индексы (табл. 5.4).

Таблица 5.3. *Внутритекстовые выделения*

Элемент	Значение элемента
`abbr`	аббревиатура
`acronym`	акроним
`code`	фрагмент исходного кода
`dfn`	определение термина
`kbd`	текст, введенный на клавиатуре
`samp`	выходные данные
`var`	имя переменной или параметра
`address`	информация об авторе и документе

Элемент `abbr` — это аббревиатура, например, HTTP, WWW, МГУ, ООН и т. д. Аббревиатуры рекомендуется использовать вместе с атрибутами `lang` и `title`, сообщающими об их языке и расшифровке:

```
<abbr lang="en" title="World Wide Web">WWW</abbr>
<abbr lang="ru" title="Московский государственный университет">МГУ</abbr>
```

Примечание

Если используется атрибут `title`, то во многих браузерах при наведении указателя мыши на аббревиатуру появляется всплывающая подсказка с расшифровкой.

Дополнительно

Элемент `abbr` можно использовать также для сокращений, например:

```
<abbr lang="ru" title="кандидат физико-математических наук">к. ф.-м. н.</abbr>
```

Акроним — это сокращение, превратившееся в самостоятельное слово: *радар*, *модем*, *колхоз* и т. д. В большинстве случаев такие слова не требуется специально выделять, поэтому элемент `acronym` используется гораздо реже, чем `abbr`.

Элемент `dfn` предназначен для выделения определяемых терминов и понятий, например:

```
<p><dfn>Браузер</dfn> &#8212; это программа, предназначенная для просмотра веб-страниц.</p>
```

Примечание

Термин, заключенный в `<dfn>...</dfn>`, рекомендуется ставить в именительном падеже единственного числа, чтобы облегчить обработку текста специальными программами (например, поисковыми системами).

Дополнительно

Остальные выделения используются в специальных случаях, например:

```
...подставим в уравнение значения переменных <var>x</var> и
<var>y</var>...
```

```
...в поле <samp>Имя пользователя</samp> введите
<kbd>anonymous</kbd>...
```

```
...Элемент <code>dfn</code> предназначен для выделения определяемых
терминов и понятий...
```

Таблица 5.4. *Нижний и верхний индексы*

Элемент	Значение элемента
`sub`	нижний индекс
`sup`	верхний индекс

Примечание

По-английски нижний индекс — «subscript», а верхний — «superscript».

Индексы используются в формулах, обозначениях и других специальных случаях:

В коде HTML	В окне браузера
`H<sub>2</sub>SO<sub>4</sub>`	H_2SO_4
`<var>a</var><sup>2</sup> +  <var>b</var><sup>2</sup>`	$a^2 + b^2$
`E<sub>7</sub><sup>sus<sub>4</sub></sup>`	$E_7^{sus_4}$

纸 *Дополнительно*

Нужно ли пользоваться специальными символами в верхнем и нижнем индексе (², ³, ½, ™ и т. д.) или следует употреблять обычные буквы и цифры, заключенные в тэги `<sub>...</sub>` и `<sup>...</sup>`?

Обычно это зависит от ситуации. Например, знак торговой марки ™ давно уже превратился в самостоятельный символ, поэтому лучше использовать его (например, с помощью подстановки `™`), а не `<sup>TM</sup>`.

Специальные символы, обозначающие числовые показатели степеней и обыкновенные дроби, можно применять в том случае, если на протяжении всего документа удается обойтись только ими. Документ будет выглядеть нелогично и некрасиво, если для второй степени в нем будет применяться подстановка `²`, а, скажем, для восьмой — конструкция `<sup>8</sup>`.

Если вы не пользуетесь специальными символами десятичных дробей (½, ¼, ¾), а конструируете эти дроби «вручную», не забывайте, что между числителем и знаменателем должна стоять специальная дробная черта (U+2044, подстановка `⁄`), а не обычная косая черта, которую можно найти на клавиатуре.

Вот как отображаются некоторые комбинации символов в браузере Mozilla.

Обычные символы	Специальные символы
`<var>c</var><sup>2</sup>  = <var>a</var><sup>2 </sup> + <var>b </var><sup>2</sup>`	`<var>c</var>² =  <var>a</var>²  + <var>b</var> ²`
$c^2 = a^2 + b^2$	$c^2 = a^2 + b^2$
`<sup>1</sup>⁄<sub>2</ sub> стакана муки и <sup>1</sup>⁄<sub>4</ sub> стакана сахара`	`½ стакана муки и ¼ стакана сахара`
$^1/_2$ стакана муки и $^1/_4$ стакана сахара	½ стакана муки и ¼ стакана сахара

Многие символы, в состав которых входят знаки в верхнем и нижнем индексе, вообще не следует «собирать по частям», как дроби; нужно пользоваться только уже готовыми «целыми» символами. К таким символам, например, относятся знаки процента (%) и промилле (‰).

Элемент `address` применяется, если вы хотите вставить в документ подпись, дату, место написания и другую подобную информацию, например:

```
...
И вдруг умел расстаться с ним,
Как я с Онегиным моим.</pre>
<address>А. С. Пушкин</address>
```

Блочные цитаты: *blockquote*

Я знал человека до такой степени необразованного, что ему приходилось самому выдумывать цитаты из классиков.

С. Е. Лец. Непричесанные мысли

Элемент `blockquote` предназначен для выделения цитат, составляющих один или несколько абзацев текста, например:

> Причудливые мультимедийные эффекты на сайтах обычно не выдерживают проверки практическим использованием. Простой текст и ясные фотографии не только лучше доносят информацию до пользователя, но и позволяют ему полнее контролировать свое взаимодействие с Сетью. А это способствует укреплению Всемирной Паутины в качестве информационной среды с мгновенной отдачей.
>
> Почти каждое исследование удобства сайтов, которое мы проводили, показывает, что наиболее простые средства передачи информации оказываются предпочтительнее для пользователей, чем «продвинутые» мультимедийные средства. Немногочисленные исключения, встречавшиеся нам, только подтверждают правило о приоритетности обычного текста и статических изображений: пользователи хотят контролировать то, с чем они взаимодействуют.

Эта цитата из статьи Я. Нильсена будет закодирована в HTML следующим образом:

```
<blockquote cite="http://www.useit.com/alertbox/20030421.html">
  <p>Причудливые мультимедийные ... с мгновенной отдачей.</p>
  <p>Почти каждое ... с чем они взаимодействуют.</p>
</blockqoute>
```

Атрибут `cite`, как понятно из этого примера, указывает на источник цитаты. Он не является обязательным, но его лучше использовать всегда, когда известен сетевой адрес цитируемого текста.

Примечание

Название элемента `blockquote` составлено из двух слов: «block» и «quote». Его можно перевести как «блочная цитата», что как раз соответствует структурному назначению этого элемента.

Обратите внимание: согласно синтаксическим правилам XHTML, нельзя заключать текст непосредственно в элемент `blockquote`. Даже если текст составляет всего один абзац, его нужно поместить в тэги `<p>...</p>` и только потом — в `<blockquote>...</blockquote>`.

Правильно	Неправильно

`<blockquote>`
`<p>Все смешалось в доме Облонских.</p>`
`</blockquote>`
	`<blockquote>`
	`Все смешалось в доме Облонских.`
	`</blockquote>`

Примечание

Современные браузеры по умолчанию отображают блочные цитаты с увеличенным отступом от левого края внешнего блока. Средства CSS позволяют не только увеличить, уменьшить или вовсе убрать этот отступ, но и расположить цитату на экране каким угодно образом — например, в виде врезки.

Одно из наиболее явных применений `blockquote` — это эпиграфы. Например, эпиграф к *разд. 5.1.4* может быть закодирован следующим образом:

```
<blockquote>
  <p>Только после явления Неба и соединения Земли с Небом можно
говорить о реальном начале космогонического процесса, в результате
которого возникают все структурные элементы мироздания.</p>
  <address>М. Евзлин. Космогония и ритуал</address>
</blockqoute>
```

Дополнительно

Элемент `blockquote` можно также использовать для примечаний, комментариев, врезок и других текстов дополнительного характера. Например, блок текста, который вы читаете прямо сейчас, автор закодировал бы в HTML как `<blockquote class="zhi">`.

Перечни: *ul*, *ol*, *dl*

Несчастные! должны ль упреки несть
От подражательниц модисткам?
За то, что смели предпочесть
Оригиналы спискам?

А. С. Грибоедов. Горе от ума

Элемент `ul` — это *маркированный (неупорядоченный) перечень (список)*, а `ol` — *нумерованный (упорядоченный)*. Согласно правилам XHTML, каждый из этих элементов должен содержать в себе один или несколько элементов `li` — пунктов перечня:

В коде HTML	В окне браузера

`<ul>`
`  <li>Красный</li>`
`  <li>Желтый</li>`
`  <li>Зеленый</li>`
`</ul>`
	• Красный
	• Желтый
	• Зеленый

`<ol>`
`  <li>Иванов</li>`
`  <li>Петров</li>`
`  <li>Сидоров</li>`
`</ol>`
	1. Иванов
	2. Петров
	3. Сидоров

Маркированные перечни (`ul`) используются тогда, когда порядок следования элементов не важен. Они позволяют представить информацию более наглядно, разбив ее на отдельные логические единицы. Сравните два следующих примера:

В операционной системе Windows сайт рекомендуется тестировать на совместимость с браузерами Internet Explorer (версии 6.0, а также 5.0 и 5.5), Mozilla (подойдет любая версия, начиная с 1.0) и Opera 6–7.

Браузеры, рекомендуемые для тестирования сайта

Операционная система:

- Windows

Браузеры:

- Internet Explorer (5.0, 5.5, 6.0)
- Mozilla (1.x)
- Opera (6.x, 7.x)

Вариант с перечнями воспринимается легче, особенно если это не печатный текст, а текст на веб-странице. Напомним, что пользователи чаще не читают веб-страницы, а бегло просматривают их. Упорядоченный перечень скорее привлечет внимание пользователя, тем более если этот перечень качественно оформлен с помощью маркеров, шрифтов и отступов.

纸 Дополнительно

Стандартная панель навигации (ее часто размещают в верхней левой части страницы) структурно является типичным маркированным перечнем. Не нужно кодировать ее с помощью таблиц, блоков (`div`) или абзацев (рис. 5.5 и листинг 5.5).

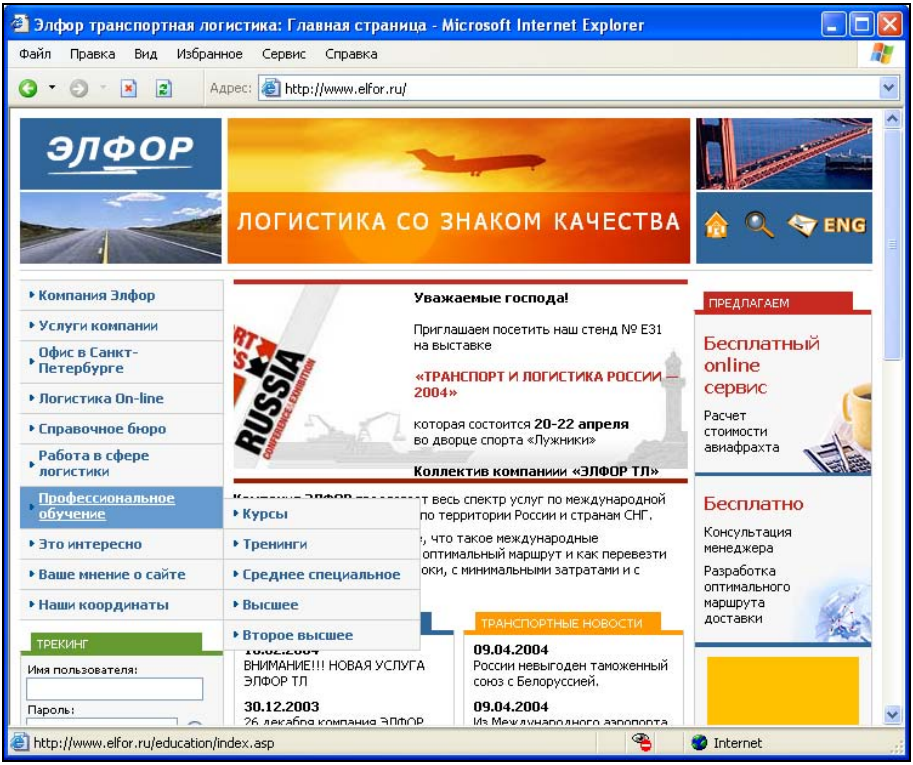


Рис. 5.5. Панель навигации на сайте **www.elfor.ru** грамотно закодирована как маркированный перечень. Все оформление, которое вы видите, сделано средствами CSS

Листинг 5.5. Панель навигации на сайте www.elfor.ru

```
<ul id="menu">
  <li><a href="/company/index.asp">Компания Элфор</a>
    <div>
      <ul>
        <li><a href="/company.asp">О компании</a></li>
        <li><a href="/articles.asp">Элформация</a></li>
        <li><a href="/company/team.asp">Команда</a></li>
```

```
        <li><a href="/vacancies.asp">Вакансии</a></li>
        <li><a href="/opinion.asp">Отзывы</a></li>
      </ul>
    </div>
  </li>
  <li><a href="/services/services.asp">Услуги компании</a>
    <div>
      <ul>
        <li><a href="/services/agents.asp">Агенты</a></li>
        <li><a href="/services/start.asp">Международная доставка
грузов</a></li>
        <li><a href="/services/routes.asp">Маршруты</a></li>
        <li><a href="/services/rates.asp">Расценки</a></li>
        <li><a href="/services/statistics.asp">Статистика</a></li>
        <li><a href="/inquiry.asp">Запрос на перевозку</a></li>
      </ul>
    </div>
  </li>
  <li><a href="/offices/index.asp">Офис в Санкт-Петербурге</a></li>
  <li><a href="/online/index.asp">Логистика On-line</a>
    <div>
      <ul>
        <li><a href="/articles.asp">Транспортные новости</a></li>
        <li><a href="/articles.asp">Тема для размышления</a></li>
        <li><a href="/consult.asp">Консультации по ВЭД</a></li>
        <li><a href="/vote.asp">Вопрос недели</a></li>
      </ul>
    </div>
  </li>
  <li><a href="/information/index.asp">Справочное бюро</a>
    <div>
      <ul>
        <li><a href="/articles.asp">Инкотермс</a></li>
        <li><a href="/information/dictionary.asp?lang=ru">Толковый
словарь логистика</a></li>
        <li><a href="/information/dictionary.asp?lang=en">Англо-
русский словарь</a></li>
        <li><a href="/laws.asp">Юридический справочник</a></li>
        <li><a href="/information/time.asp">Сверим часы</a></li>
        <li><a href="/urls.asp">Ресурсы интернета</a></li>
```

```
      </ul>
    </div>
  </li>
  <li><a href="/job/index.asp">Работа в сфере логистики</a>
    <div>
      <ul>
        <li><a href="/job/vacancies.asp">Вакансии</a></li>
       <li><a href="/job/resumes.asp">Резюме</a></li>
      </ul>
    </div>
  </li>
  <li><a href="/education/index.asp">Профессиональное обучение</a>
    <div>
      <ul>
        <li><a
href="/education/courses.asp?status=0">Курсы</a></li>
        <li><a
href="/education/courses.asp?status=1">Тренинги</a></li>
        <li><a href="/education/courses.asp?status=2">Среднее
специальное</a></li>
        <li><a
href="/education/courses.asp?status=3">Высшее</a></li>
        <li><a href="/education/courses.asp?status=4">Второе
высшее</a></li>
      </ul>
    </div>
  </li>
  <li><a href="/interest/index.asp">Это интересно</a>
    <div>
      <ul>
        <li><a href="/articles.asp">Знаете ли вы, что...</a></li>
        <li><a href="/articles.asp">Элфория</a></li>
      </ul>
    </div>
  </li>
  <li><a href="/feedback.asp">Ваше мнение о сайте</a></li>
  <li><a href="/coords.asp">Наши координаты</a></li>
</ul>
```

Нумерованные перечни (`ol`) применяются тогда, когда важен порядок следования пунктов — например, для перечисления действий, которые должен совершить пользователь, или каких-нибудь объектов, которые он должен перебрать в определенной последовательности.

Примечание

- `ul` — сокращение от «unordered list» («неупорядоченный перечень»);
- `ol` — сокращение от «ordered list» («упорядоченный перечень»);
- `li` — сокращение от «list item» («пункт перечня»).

По умолчанию нумерованный перечень отображается с нумерацией арабскими цифрами с точкой, но с помощью CSS нумерацию можно изменить — например, на римские цифры или строчные буквы латинского алфавита. Номера проставляются браузером автоматически, так что если вы вставите новый пункт `<li>...</li>` в середину существующего нумерованного перечня, нумерация «пересчитается» заново.

Вложенные перечни вызывают особенные проблемы у начинающих кодировщиков. Рассмотрим такой перечень (в нем перечислены некоторые распространенные браузеры для разных операционных систем):

- Microsoft Windows
  - Microsoft Internet Explorer
  - Mozilla
  - Opera
- UNIX/Linux
  - Konqueror
  - Mozilla
  - Opera
- Mac OS
  - Safari
  - Camino

Закодируем сначала внешний перечень (это очень легко):

```
<ul>
  <li>Microsoft Windows</li>
  <li>UNIX/Linux</li>
  <li>MacOS</li>
</ul>
```

Теперь добавим первый из внутренних перечней. Это следует сделать так (то, что мы добавили, выделено полужирным шрифтом):

```
<ul>
  <li>Microsoft Windows
    <ul>
      <li>Microsoft Internet Explorer</li>
      <li>Mozilla</li>
      <li>Opera</li>
    </ul>
  </li>
  <li>UNIX/Linux</li>
  <li>MacOS</li>
</ul>
```

Весь новый перечень должен быть добавлен внутрь первого элемента `li`, то есть после текста «Microsoft Windows» и *перед* закрывающим тэгом `</li>`. Не совершайте распространенной ошибки, как в этом коде (полужирным шрифтом выделен тэг `</li>`, стоящий не на своем месте):

```
<ul>
  <li>Microsoft Windows</li>  <!-- Здесь ошибка! -->
  <ul>
    <li>Microsoft Internet Explorer</li>
    <li>Mozilla</li>
    <li>Opera</li>
  </ul>
  <li>UNIX/Linux</li>
  <li>MacOS</li>
</ul>
```

В листинге 5.6 приведен полностью закодированный перечень. Чтобы подчеркнуть правильную вложенность элементов, каждый элемент обведен рамкой. Здесь наглядно видно, что каждый из элементов `ul`, обозначающих внутренние перечни, вложен в один из элементов `li` внешнего перечня (а не во внешний `ul`, как в ошибочном примере ранее).

Менее распространен третий вид перечней — `dl`, *перечень определений*. Этот перечень вначале предназначался для словарных статей и других подобных текстов. В элемент `dl` входят элементы `dt` (термин) и `dd` (определение), которые могут следовать друг за другом в любом порядке и в любых количествах.

Листинг 5.6. Вложенные перечни

```
<ul>
  <li>Microsoft Windows
    <ul>
      <li>Microsoft Internet Explorer</li>
      <li>Mozilla</li>
      <li>Opera</li>
    </ul>
  </li>
  <li>UNIX/Linux
    <ul>
      <li>Konqueror</li>
      <li>Mozilla</li>
      <li>Opera</li>
    </ul>
  </li>
  <li>MacOS
    <ul>
      <li>Safari</li>
      <li>Camino</li>
    </ul>
  </li>
</ul>
```

Вот пример «традиционного» перечня определений и его код (листинг 5.7):

Веб-страница

Ресурс, который пользователь просматривает с помощью браузера. В основе веб-страницы лежит документ HTML, к которому могут быть подключены таблицы стилей, изображения и другие внешние файлы.

Сайт

Совокупность веб-страниц, объединенных общей тематикой и структурой.

Всемирная Паутина (World Wide Web)

Совокупность всех сайтов и связанных с ними ресурсов (в частности, загружаемых файлов).

Листинг 5.7. Перечень определений

```
<dl>
  <dt>Веб-страница</dt>
  <dd>Ресурс, который пользователь просматривает с помощью браузера. В основе веб-страницы лежит документ HTML, к которому могут быть подключены таблицы стилей, изображения и другие внешние файлы.</dd>
  <dt>Сайт</dt>
  <dd>Совокупность веб-страниц, объединенных общей тематикой и структурой.</dd>
  <dt>Всемирная Паутина (World Wide Web)</dt>
  <dd>Совокупность всех сайтов и связанных с ними ресурсов (в частности, загружаемых файлов).</dd>
</dl>
```

Примечание

- `dl` — сокращение от «definition list» («перечень определений»);
- `dt` — сокращение от «definition term» («определяемый термин»);
- `dd` — сокращение от «definition description» (в данном контексте — «определение термина»).

Дополнительно

Элемент `dt` может содержать в себе только строчные элементы, а элемент `dd` — также и блочные.

Еще одно применение перечня определений — титры и подобные им тексты, например:

В коде HTML	В окне браузера

`<dl>`
`  <dt>Авторы сценария</dt>`
`  <dd>Аркадий Стругацкий</dd>`
`  <dd>Борис Стругацкий</dd>`
`  <dt>Режиссер</dt>`
`  <dd>Константин Бромберг</dd>`
	Авторы сценария
	Аркадий Стругацкий
	Борис Стругацкий
	Режиссер
	Константин Бромберг

(окончание)

В коде HTML	В окне браузера
`<dt>В ролях</dt>`	В ролях
`<dd>Александр Абдулов</dd>`	Александр Абдулов
`<dd>Александра Яковлева</dd>`	Александра Яковлева
`<dd>Валентин Гафт</dd>`	Валентин Гафт
`<dd>Михаил Светин</dd>`	Михаил Светин
`<dd>Эммануил Виторган</dd>`	Эммануил Виторган
`<dd>и другие</dd>`	и другие
`</dl>`	

Дополнительно

Обратите внимание, что в последнем примере автор немного отступил от принятого ранее стиля кодирования. Хотя элементы `dd` не входят в `dt`, они размещены с дополнительным отступом от левого края. Это сделано для наглядности кода, чтобы подчеркнуть разницу между `dt` и `dd`.

На практике очень редко встречаются случаи, когда имеет смысл нарушать устоявшийся стиль кодирования.

Таблицы

> Должна быть закономерность, должны быть таблицы, а то вечно изобретаешь велосипед.
>
> *К. Булычев. Поселок*

Таблицы — важная и объемная часть языка XHTML. Эта книга не учебник по XHTML, поэтому здесь будет дан только общий обзор табличных элемен-тов, необходимый для обсуждения кодирования текстового содержания сайтов (табл. 5.5).

Примечание

Более подробную информацию о таблицах можно найти в официальной спецификации HTML 4.01, на странице **www.w3.org/TR/html4/struct/tables.html**.

К сожалению, даже многие опытные разработчики не используют табличные элементы на полную мощность.

Таблица 5.5. *Элементы XHTML, связанные с таблицами*

Элемент	Значение элемента
`table`	таблица
`caption`	заголовок таблицы
`thead`	шапка таблицы
`tbody`	тело таблицы
`tfoot`	«подвал» таблицы
`tr`	строка
`th`	ячейка с заголовком
`td`	ячейка с данными
`col`	графа (столбец)
`colgroup`	группа граф (столбцов)

Дополнительно

В DTD XHTML 1.0 Strict указано следующее:

```
<!ELEMENT table
    (caption?, (col*|colgroup*), thead?, tfoot?, (tbody+|tr+))>
```

Это означает, что элемент `table` содержит в себе следующие элементы:

- `caption` (0 или 1);
- `col` или `colgroup` (0 или больше);
- `thead` (0 или 1);
- `tfoot` (0 или 1);
- `tbody` или `tr` (1 или больше).

Другими словами, таблица обязательно должна содержать по крайней мере одно тело (`tbody`) или строку (`tr`); также таблица может содержать заголовок (`caption`), шапку (`thead`) и «подвал» (`tfoot`), но каждый из этих элементов, если он присутствует, может быть только в одном экземпляре.

Пример таблицы в HTML приведен в листинге 5.3. На рис. 5.6 показано, как эта таблица отображается в окне браузера (рамки и форматирование в ячейках выполнены с помощью CSS).

tr	Браузер	Количество посещений	В % от общего количества посещений	thead
tr	Microsoft Internet Explorer	3786	84,93	tbody
tr	Opera	303	6,80	
tr	Mozilla	251	5,63	
tr	Другие	118	2,64	
tr	Итого	4458	100,00	tfoot

Ячейки таблицы: □ — td, ■ — th

Рис. 5.6. Таблица и ее элементы

Примечание

В современных браузерах по умолчанию таблицы отображаются без рамок. Рамки таблицы можно сделать средствами CSS.

Не пренебрегайте элементами `thead`, `tbody` и `tfoot`. Кроме того, что они помогают отразить логическую структуру таблицы, также благодаря им облегчается оформление таблицы с помощью CSS.

Дополнительно

Например, вы можете написать в таблице стилей:

```
thead {
  background-color: #ff99cc;
}
```

— и фон сразу всех ячеек в шапке таблицы станет бледно-малиновым.

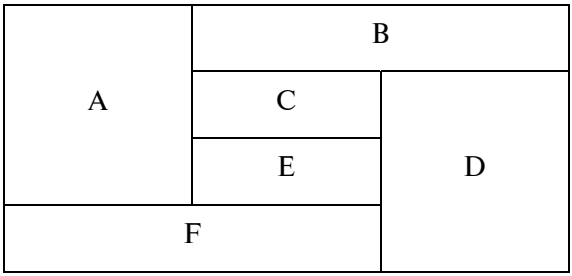
Если таблице нужен заголовок, употребляйте элемент `caption`, который существует специально для этой цели. Его нужно размещать сразу после открывающего тэга `<table>`:

```
<table>
  <caption>Таблица 5.3. Внутритекстовые выделения</caption>
  <tbody>
    ...
  </tbody>
</table>
```

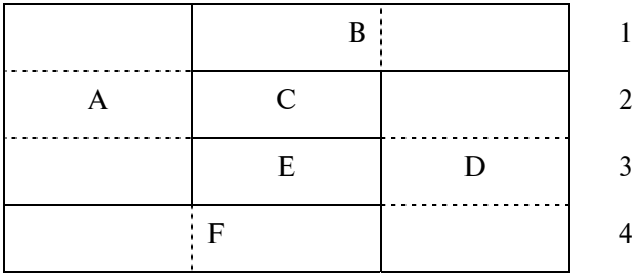
Дополнительно

Обратите внимание: «подвал» таблицы (`tfoot`) в коде следует сразу за шапкой (`thead`), в то время как на странице (см. рис. 5.6) он отображается в конце всей таблицы. Этого требуют правила XHTML: такое «преждевременное» кодирование «подвала» нужно для того, чтобы браузер мог отобразить его, еще не закончив загружать тело таблицы.

Атрибуты `colspan` и `rowspan` элементов `td` и `th` позволяют распространить одну ячейку на несколько строк или столбцов. С помощью этих атрибутов можно кодировать сложные таблицы. Рассмотрим один из примеров такого кодирования. Предположим, нам нужно создать таблицу следующего вида:



Сначала мысленно проведем все недостающие линии, чтобы таблица стала «полной»:



Теперь сосчитаем количество строк в полученной таблице — их получается четыре. Затем определим, какие ячейки относятся к каждой из этих четырех строк. Если одна ячейка распространяется на несколько строк, в XHTML она считается относящейся к верхней из этих строк. (Например, в нашей таблице ячейка A относится к строке 1, а ячейка D — к строке 2.) Окончательно получаем: строка 1 — ячейки A и B, строка 2 — ячейки C и D, строка 3 — ячейка E, строка 4 — ячейка F.

Затем присвоим атрибуты `colspan` и `rowspan` всем ячейкам, которым они нужны. Это те ячейки, которые оказались разделенными на несколько частей пунктирными линиями. Если ячейка разделена вертикальными линиями (например, B) — ей присваивается атрибут `colspan`, если горизонтальными (например, A) — атрибут `rowspan`. Значение этого атрибута равно количеству частей, на которые разделена ячейка. Например, для ячейки A мы запишем: `rowspan="3"`.

Окончательный код таблицы приведен в листинге 5.8.

Листинг 5.8. Сложная таблица

```
<table>
  <tbody>
    <tr>
      <td rowspan="3">A</td>
      <td colspan="2">B</td>
    </tr>
    <tr>
      <td>C</td>
      <td rowspan="3">D</td>
    </tr>
    <tr>
      <td>E</td>
    </tr>
      <td colspan="2">F</td>
    <tr>
    </tr>
  </tbody>
</table>
```

Примечание

Некоторые разработчики в похожих случаях предпочитают использовать вложенные таблицы. Это неудачное решение, так как вложенные таблицы увеличивают объем кода HTML, и, кроме того, браузеру сложнее их отобразить.

Таблицы часто используются для верстки веб-страниц. Это устаревшая технология; табличная верстка была актуальна во времена старых браузеров, для которых не существовало других практических способов создать более или менее сложную композицию страницы. Сейчас возможности CSS позволяют не только повторить любую верстку, которая раньше делалась с помощью таблиц, но и продвинуться еще дальше — например, добиться перекрытия или наложения блоков.

Современному разработчику рекомендуется забыть табличную верстку, как пережиток прошлого, и использовать таблицы только для того, для чего они были предназначены изначально — для информации табличного характера.

纸 Дополнительно

Элементы `col` и `colgroup` служат для обозначения отдельных граф (столбцов) таблицы и групп граф. Рассмотрим, как можно применить эти элементы к следующей таблице:

Звуковые карты	Цена	I	II	Гарантия, мес.
Creative SB Audigy 2 Platinum EX 6.1, RTL	5977,44	196	192	6
Creative SB Audigy 2, OEM	2409,12	79	77	6
Creative SB Audigy 2, RTL	3323,88	109	106	6
Creative SB Audigy 5.1, OEM	1860,36	61	59	6
Creative SB Live Player 5.1	1036,98	34	33	6

Эта таблица будет закодирована следующим образом (однотипные строки для краткости пропущены):

```
<table>
  <colgroup class="name">
  </colgroup>
  <colgroup class="price">
    <col class="rubles">
    </col>
    <col class="dollarsI">
    </col>
    <col class="dollarsII">
    </col>
  </colgroup>
  <colgroup class="warranty">
  </colgroup>
  <thead>
    <tr>
      <th>Звуковые карты</th>
      <th>Цена</th>
      <th>I</th>
      <th>II</th>
      <th>Гарантия, мес.</th>
    </tr>
  </thead>
  <tbody>
    <tr>
```

```
      <td>Creative SB Audigy 2 Platinum EX 6.1, RTL</td>
      <td>5977,44</td>
      <td>196</td>
      <td>192</td>
      <td>6</td>
    </tr>
    ...
  </tbody>
</table>
```

Применение `col` и `colgroup` имеет два положительных эффекта. Во-первых, эти элементы позволяют браузеру рассчитать количество граф в таблице еще до того, как загрузится вся таблица целиком. Браузеру не нужно будет ждать, пока поступит вся таблица, прежде чем он начнет отображать ее. Во-вторых, во многих браузерах (включая Internet Explorer) можно однотипно оформить все ячейки одной графы, присвоив какое-либо свойство CSS элементу `col` или `colgroup`. Например, здесь:

```
col.rubles {
  text-align: center;
}
```

Похожим образом заданы стили для всех остальных граф. К сожалению, некоторые браузеры (в частности, Mozilla до версии 1.6 включительно) не поддерживают такого способа оформления граф (формально ячейки не входят в элемент `col`, поэтому и оформление, указанное для `col`, не должно на них распространяться). Если мы хотим добиться однотипного оформления всех ячеек одной графы в таких браузерах, придется записывать для каждой из этих ячеек `<td class="rubles">...</td>`.

Гиперссылки: элемент *a*

> Председательствующий м-р Леонард. Но разве кто-нибудь ставит эти положения под сомнение?
> М-р Эндрюс. Тем не менее надо подчеркивать их снова и снова.
>
> *К. Саймак. Утраченная вечность*

Гиперссылки — наиболее заметное отличие текстового содержания сайтов от печатных изданий. Благодаря гиперссылкам все существующие сайты связаны в единое информационное пространство, которое по праву называется Всемирной Паутиной (World Wide Web, WWW).

Чтобы превратить текст в гиперссылку, нужно заключить его в элемент `a` с атрибутом `href`. Значение этого атрибута — адрес (URL) того ресурса, к которому ведет ссылка:

В коде HTML	В окне браузера
`<a href="http://www.w3.org/">Консорциум W3C</a>`	Консорциум W3C

В современных графических браузерах гиперссылки по умолчанию отображаются с подчеркиванием, а также выделяются ярко-синим цветом. Если подвести к гиперссылке указатель мыши, он превращается в «указующий перст». В таком состоянии можно щелкнуть мышью, чтобы переместиться к другому ресурсу.

Внимание!

Ссылки должны быть подчеркнутыми. Когда они подчеркнуты, то пользователь сразу видит, что на странице является ссылками, а что — нет. Если нарушать это правило, то пользователь потратит лишнее время, работая с веб-страницей. Еще хуже, если на странице подчеркнут текст, не являющийся ссылкой: многие пользователи попробуют щелкнуть по нему — и обманутся.

Чаще всего используются следующие значения атрибута `href`:

- ссылка на главную страницу какого-либо сайта, например: `href="http://www.mozilla.org/"`. Обратите внимание: URL такого вида начинается с `http://` (не перепутайте: *одно* двоеточие и *две* косые черты!), за которым следует имя сайта, и заканчивается косой чертой. Подавляющее большинство ссылок такого характера работают и без завершающей косой черты, но указывать ее считается признаком хорошего тона, так как это обычно упрощает обработку запроса на веб-сервере, где расположен сайт;

Дополнительно

Обязательно ли имя сайта начинается с «www»? Вовсе нет: это просто широко распространенная традиция, которой обычно следуют те, кто размещает сайты в Сети. Большинство сайтов «откликаются» на имя без «www» так же хорошо, как с «www». Например, адреса **lib.ru** и **www.lib.ru** приводят к одному и тому же ресурсу.

Бывают сайты, в адресах которых нет никакого «www» — например, **msdn.microsoft.com** или **mail.yandex.ru**.

Если вы хотите установить ссылку на чужой сайт — на всякий случай проконсультируйтесь с администратором того веб-сервера, на котором этот сайт размещен. Если один и тот же сайт доступен под несколькими именами — возможно, какое-то из этих имен является более предпочтительным, и администратор сообщит вам об этом.

- ссылка на конкретную страницу, например: `href="http://www.adobe.com/svg/viewer/install/main.html"`. Для разделения имени сервера, каталогов на этом сервере и имени файла используется косая черта: `/` (а не обратная косая черта: `\`, как в командной строке DOS или Windows);

Примечание

В URL важен регистр букв. Например, `index.html` и `INDEX.HTML` — два разных файла. Некоторые серверы настроены таким образом, чтобы и тот, и другой вариант обрабатывались одинаково, но в общем случае ссылка может не работать, если вы указали URL не в том регистре.

- ссылка на загружаемый файл, например: `href="http://www.w3.org/TR/xhtml1/xhtml1.zip"`. Если щелкнуть по такой ссылке, браузер обычно открывает диалоговое окно, в котором спрашивает, что делать с этим файлом — например, открыть или сохранить на диске. В большинстве случаев загружаемые файлы предназначены именно для того, чтобы пользователь сохранял их у себя на диске и затем открывал на своем компьютере;

Дополнительно

Если вы размещаете на своем сайте ссылки на загружаемые файлы, делайте эти ссылки прямыми. Некоторые разработчики используют «косвенные» ссылки: они ведут не непосредственно к файлу, а к серверной программе, которая вызывает загрузку файла. (Например, ссылка `http://www.kilor.ru/download.php?id=10537&action=1` на музыкальном сайте на самом деле ведет к файлу `http://www.philigon.ru/mp3/lament.mp3`.) Во многих случаях это неудобно для пользователя, который хочет загрузить файл не стандартными средствами браузера, а какой-нибудь другой программой (например, менеджером загрузки наподобие GetRight или FlashGet). Не скрывайте от пользователей истинный адрес файла, который им предстоит загружать.

- относительные ссылки на страницы текущего сайта, например: `href="news.html"` или `href="../../about.html"`. Полный адрес такой страницы определяется в зависимости от адреса текущей страницы.

Примечание

Более подробно про синтаксис URL можно прочитать, например, в стандарте RFC 2396 (**www.ietf.org/rfc/rfc2396.txt**).

Предположим, мы находимся на странице **http://www.somesite.ru/docs/opera/opera7.html**. Посмотрим, к каким файлам приведут различные относительные ссылки:

Относительная ссылка	Полный адрес файла
`href="index.html"`	**http://www.somesite.ru/docs/opera/index.html**

(окончание)

Относительная ссылка	Полный адрес файла
`href="bugs/latest.html"`	**http://www.somesite.ru/docs/opera/bugs/latest.html**
`href="compliance/css/css1.html"`	**http://www.somesite.ru/docs/opera/compliance/css/css1.html**
`href="../index.html"`	**http://www.somesite.ru/docs/index.html**
`href="../mozilla/bugs/latest.html"`	**http://www.somesite.ru/docs/mozilla/bugs/ latest.html**
`href="../../feedback/mailto.html"`	**http://www.somesite.ru/feedback/mailto.html**

Обратите внимание: две точки обозначают подъем на один уровень вверх в структуре каталогов сайта;

- ссылки на адрес электронной почты, например: `href="mailto:info@yandex.ru"`. Обычно при активизации этой ссылки вызывается почтовая программа, позволяющая отправить сообщение по указанному адресу.

Текст гиперссылки должен подчиняться следующим требованиям:

- создавайте короткие гиперссылки, желательно не более 3—4 слов. Старайтесь, чтобы каждая гиперссылка располагалась на одной строке;
- ссылки должны соответствовать тому ресурсу, на который они ссылаются. Например, ссылка в тексте `…компания «ABC», которая примет участие в конференции…`, должна вести на официальный сайт компании «ABC», а не на другую статью, в которой тоже упоминается эта компания. Ссылка `Загрузить Mozilla 1.8` должна указывать непосредственно на загружаемый файл, а не вести на веб-страницу этого браузера.

  Если, например, ссылка ведет на статью, то текст ссылки должен соответствовать заголовку статьи. Если статья называется «Особенности объектной модели браузеров Internet Explorer 5.0 и 5.5», то не рекомендуется делать такие ссылки:

  Статья про IE 5.0 и 5.5

  DHTML в Internet Explorer

  ...об этом можно прочитать в одной статье на Webmascon.Com...

- текст ссылки не должен содержать слова о том, что это ссылка и что по ней надо щелкнуть. В большинстве случаев не нужно и слов вроде «Перейти».

Рекомендуется	Не рекомендуется
Василий Пупкин	Щелкните здесь, чтобы перейти на страницу В. Пупкина
Каталог товаров	Ссылка на каталог товаров
Полный текст статьи	Перейти к полному тексту статьи...

- в тексте ссылки не должно быть слов «Назад», «Вперед», «На предыдущую страницу», «На следующую страницу» и т. д. Вы же не знаете, в каком порядке пользователь просматривает страницы, и какие страницы являются для него предыдущей и следующей.

  Автор несколько раз наблюдал, как пользователь нажимал на ссылку **Назад** или **Вернуться**, рассчитывая, что он при этом переместится на ту страницу, которую посетил раньше. Разумеется, ссылка вела совсем на другую страницу (обычно на ту, которую пользователь до этого вообще не посещал).

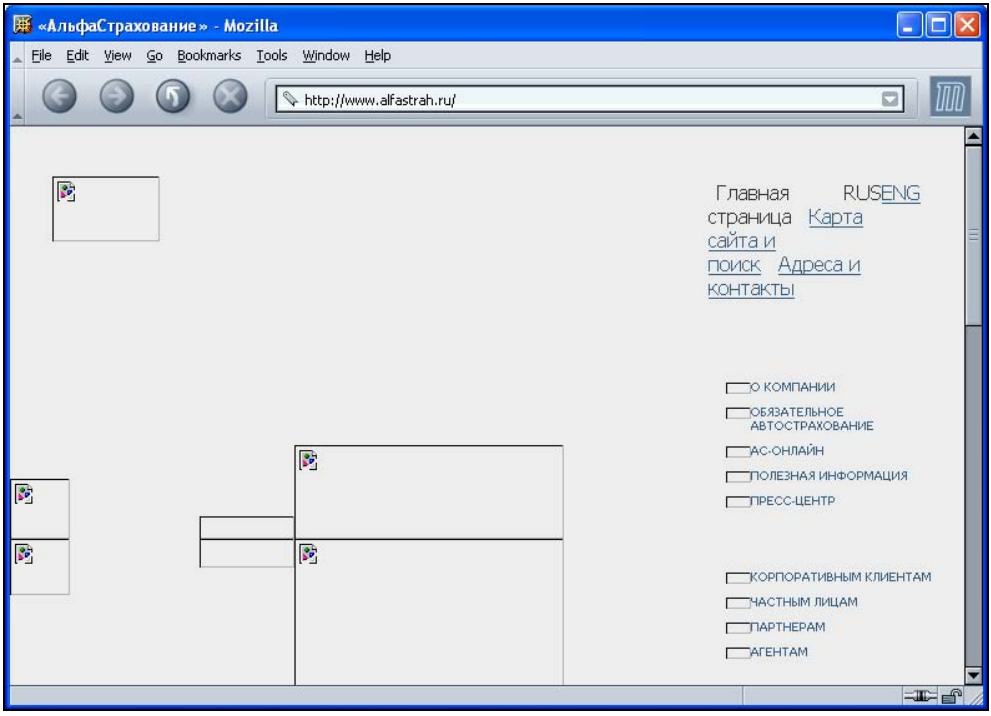
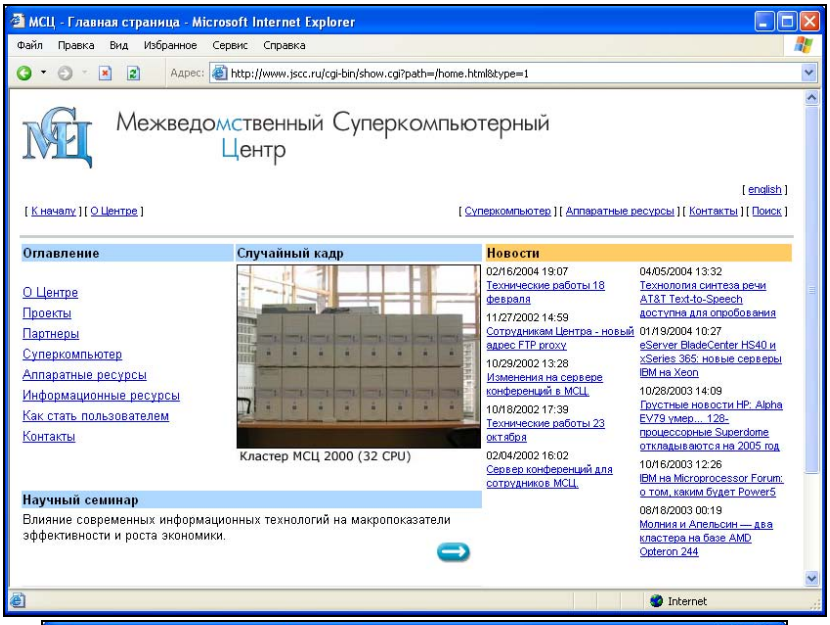
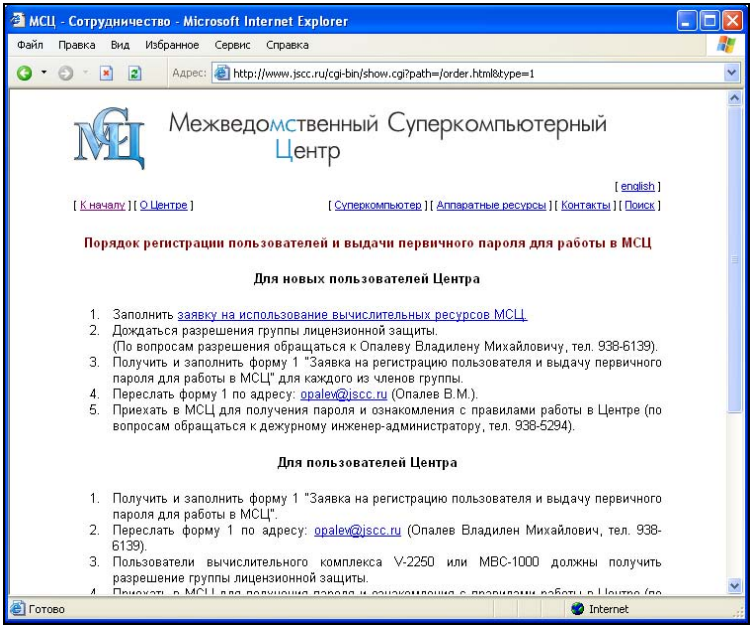


Рис. 5.7. www.alfastrah.ru: в правом верхнем углу — на самом деле две ссылки: «Карта сайта и поиск» и «Адреса и контакты». Видите, что с ними случилось?



а



б

Рис. 5.8. Сайт Межведомственного суперкомпьютерного центра (**www.jscc.ru**) — один из рекордсменов по неправильным ссылкам. Например, ссылка «Как стать пользователем» на главной странице (*а*) ведет на страницу, озаглавленную... «МСЦ - Сотрудничество» (*б*). А ссылка «Контакты» ведет на страницу, которая называется не «Контакты», а «МСЦ - Адрес» (при том, что это заглавие тоже не вполне соответствует содержанию страницы)

Примечание

Бывает, что разработчик не делает ссылкой текст, который следует сделать ссылкой. Например, в новостях сайта объявлено:

> В разделе «Программирование» появилась новая статья «Алгоритмы двоичного поиска»

Лучший вариант здесь — сделать слово «Программирование» ссылкой на главную страницу одноименного раздела, а название статьи — ссылкой на страницу, содержащую саму эту статью:

> В разделе «Программирование» появилась новая статья «Алгоритмы двоичного поиска»

К сожалению, не все разработчики следуют этой рекомендации. Прочитав эту новость без ссылок, пользователю приходится обращаться к панели навигации и «вручную» переходить в нужный раздел, а там — к нужной статье.

Размещая ссылку на загружаемый файл (например, звукозапись или установочный комплект программы), обязательно нужно указать:

- размер файла (в килобайтах или мегабайтах);
- формат (тип) этого файла.

Например, для музыкального файла сто́ит сообщить тип (например, MP3, OGG или WAV), но такие подробности, как частота оцифровки или качество сжатия, обычно излишни:

Плохо	Соната № 5
Лучше	Соната № 5 (*MP3, 7,5 Мб*)
Избыточно	Соната № 5 (*MP3, 44 кГц, 192 bps, 7,5 Мб*)

Если ссылка ведет на изображение, рекомендуется указать, кроме формата и размера, ширину и высоту этого изображения в пикселах.

Дополнительно

А вот имя загружаемого файла в большинстве случаев указывать совсем необязательно. Пользователю обычно нет никакой разницы, как называется файл — sonate.mp3, sonata5.mp3, beethoven_sonate_no_5.mp3 или как-то еще. Если пользователь все-таки захочет увидеть имя файла, он посмотрит в строку состояния в нижней части окна браузера.

Иногда ссылку бывает полезно сопроводить описанием или аннотацией. Например, ссылку на текст статьи хорошо дополнить аннотацией этой статьи. Аннотацию лучше написать заново, а не использовать для нее фрагмент самой статьи. И не нужно делать ссылкой саму эту аннотацию — достаточно, чтобы ссылкой был заголовок статьи.

Рекомендуется	Не рекомендуется
Алгоритмы двоичного поиска	**Алгоритмы двоичного поиска**
Василий Пупкин	*Василий Пупкин*
При реализации алгоримов двоичного поиска многие разработчики сталкиваются со стандартными затруднениями...	При реализации алгоримов двоичного поиска многие разработчики сталкиваются со стандартными затруднениями...

Следующие рекомендации касаются внешнего вида ссылок:

- всегда подчеркивайте ссылки. Неподчеркнутые ссылки — это дурной тон, свойственный в основном начинающим разработчикам, которые не разбираются в удобстве и доступности сайтов. Если ссылки не подчеркнуты, пользователь не может сразу определить, что на странице является ссылкой, а что — нет, по какому тексту можно щелкнуть мышью, а по какому — нельзя.

  Кроме того, даже если пользователь честно избороздил мышью всю страницу и выяснил, где на ней ссылки, а где — обычный текст, это не значит, что он все это запомнил. При следующем посещении страницы он с большой вероятностью снова будет вынужден угадывать, по какому тексту здесь можно щелкнуть, а по какому — нет.

  Чтобы пользователю не приходилось тратить лишнее время на исследование страницы, а также проявлять чудеса памяти, — делайте все ссылки подчеркнутыми;

  Внимание!

  Всегда подчеркивайте ссылки. Неподчеркнутые ссылки приводят к ухудшению удобства и доступности сайта. Также не подчеркивайте текст, не являющийся ссылками.

  Примечание

  Одно из важных правил разработки сайта гласит: не применяйте ни одного оформительского или технологического решения только потому, что у вас есть возможность его применить. Технологии CSS позволяют подчеркивать, не подчеркивать или вовсе зачеркивать ссылки; в конце концов, ссылки вообще можно сделать невидимыми или заставить их убегать от подведенного к ним указателя мыши. Но это не значит, что все это *нужно* делать.

- посещенные ссылки должны отличаться по цветовому оформлению от непосещенных. Это позволит пользователю не путаться в том, какие страницы он уже посетил, а какие — еще нет.

Для ссылок лучше всего использовать оттенки стандартных цветов: синий — для непосещенных, бордовый — для посещенных. Не обязательно применять те же самые оттенки, которые установлены в браузере по умолчанию; подберите такие оттенки синего и бордового, чтобы они удачно сочетались с остальными цветами сайта, а также чтобы посещенные ссылки выглядели бледнее, чем непосещенные;

Дополнительно

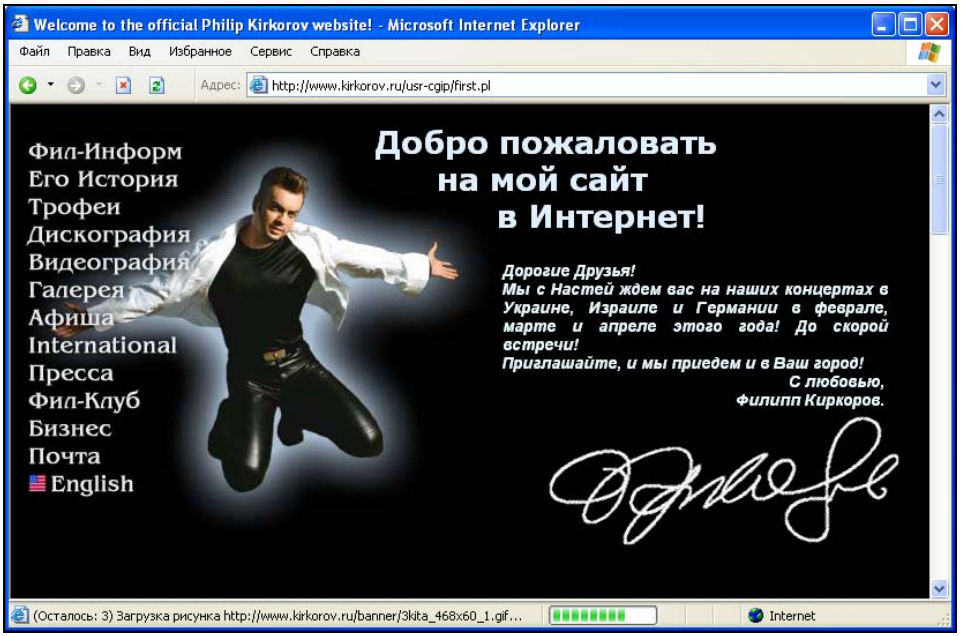
Во многих современных графических браузерах непосещенные ссылки по умолчанию отображаются цветом `#0000ff`, а посещенные — цветом `#800080`. Это не очень удачный вариант. Ярко-синий `#0000ff` чаще всего выглядит слишком броско и «режет глаза». На что его заменить — зависит от сочетания цветов остальных элементов страницы. Например, в некоторых случаях хорошо подходят `#3333cc` или `#0066cc`.

А цвет `#800080` для посещенных ссылок неудачен потому, что не относится к безопасной палитре. Ближайшие к нему оттенки из безопасной палитры — `#660066` и `#990099`.

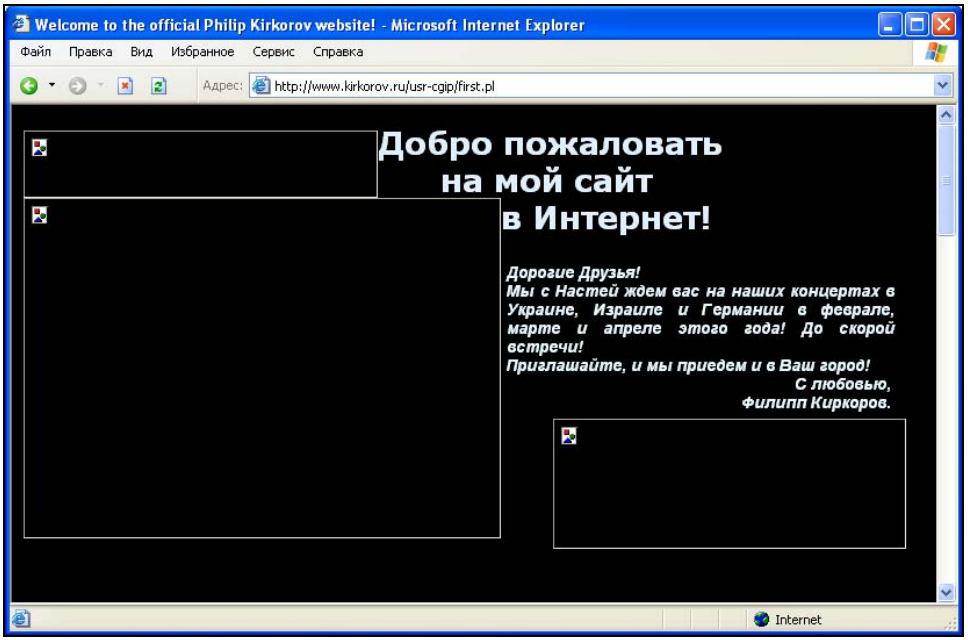
- не делайте ссылки графическими — текст, заключенный в изображение, приводит к неудобствам для пользователя (*см. разд. 1.1.2*). А ссылки — это не просто текст, это элемент функциональности сайта, поэтому для них требования удобства и доступности особенно важны. Тем более не стоит использовать для ссылок карты-изображения (рис. 5.9);
- некоторые разработчики предпочитают подсвечивать ссылки при наведении на них указателя мыши. Это не причинит пользователю неудобств только в том случае, если ссылка действительно слегка подсвечивается, а не изменяется в размерах, радикально перекрашивается или перескакивает на другое место. Вполне достаточно, чтобы чуть-чуть изменился цвет текста ссылки или фона под ней (но не то и другое одновременно!).

Дополнительно

Некоторые разработчики делают так, что ссылки подчеркиваются, когда пользователь подводит к ним указатель мыши. Это еще хуже, чем не подчеркивать ссылки вовсе. Во-первых, пользователь испытывает все ранее описанные неприятности от неподчеркнутых ссылок. Во-вторых, при наведении указателя мыши на ссылку этот указатель сам по себе изменяет форму, так что пользователь и так видит, что это ссылка. Если теперь, когда пользователь уже понял, что это ссылка, она наконец еще и подчеркнется, — это можно расценить только как дополнительное издевательство над пользователем.



а



б

Рис. 5.9. www.kirkorov.ru (*а*): пользователь, в браузере которого отключена загрузка изображений (*б*), не может воспользоваться навигацией на этом сайте

С помощью атрибута `title` ссылку можно снабдить дополнительным текстом, который в большинстве браузеров отображается в виде всплывающей подсказки при наведении указателя мыши на ссылку.

В коде HTML	В окне браузера
`<a href="index.html" title="На главную страницу">ABC</a>`	ABC На главную страницу

Такие всплывающие подсказки уместны тогда, когда они предоставляют полезную дополнительную информацию о ресурсе, на который ведет ссылка. Например, одно из удачных применений `title` — в панели навигации сайта, когда при наведении на название раздела всплывает подсказка, сообщающая о том, как давно этот раздел обновлялся:

```
<a href="products.html" title="Последнее обновление:
11.04.2004">Программные продукты</a>
```

Наоборот, не нужно делать подсказок, повторяющих текст ссылки или просто не несущих никакой полезной информации для пользователя. Вот неудачные примеры:

```
<a href="products.html" title="Программные продукты">Программные продукты</a>
<a href="products.html" title="Наши новейшие разработки">Программные продукты</a>
```

Примечание

Не путайте атрибут `title` (всплывающая подсказка) с элементом `title` (заглавие документа HTML). Это совершенно разные вещи, хотя и называются они одинаково.

Поведение ссылок должно быть разумным и предсказуемым, удобным для пользователя:

- не размещайте на веб-странице ссылок, ведущих на эту же самую страницу. Щелкая по ссылке, пользователь ожидает, что он перейдет на *другую* страницу. Если он останется там же, где был, его ожидания обманутся;
- иногда на странице, где размещен крупный текст, в нем помещают ссылки **К началу страницы**. Это совершенно лишняя мера: чтобы переместиться к началу страницы, пользователь привык применять прокрутку или сочетание клавиш <Ctrl>+<Home>;
- не создавайте навигацию с использованием JavaScript, Flash, ActiveX, апплетов и других модных технологий. Каждая из таких технологий дает

прекрасные результаты в своей области применения, но для навигации обычный HTML подходит лучше их всех, вместе взятых;

Дополнительно

JavaScript позволяет изменять текст в строке состояния (обычно слева в нижней части окна браузера). Начинающие разработчики злоупотребляют этой возможностью, так что при наведении на ссылку в строке состояния отображается не адрес, на который она ведет, а посторонний текст, не несущий пользователю никакой полезной информации.

Не подменяйте текст в строке состояния. Для многих пользователей он несет важную информацию о назначении ссылки. Иногда именно адрес в строке состояния позволяет пользователю решить, нужно ли щелкать по этой ссылке.

- не открывайте ссылки в новом окне. Щелкая по ссылке, пользователь в большинстве случаев ожидает, что она откроется в текущем окне, заменив собой его прежнее содержимое. Новое окно должно открываться только тогда, когда пользователь хочет этого специально. Для этого он применяет специальные средства браузеров — например, в большинстве браузеров можно открыть ссылку в новом окне, щелкая по ней с удерживанием клавиши <Shift>;

Примечание

Довольно часто встречается следующая ситуация: пользователь щелкает по ссылке, и страница открывается в новом окне, однако пользователь этого не замечает, потому что новое окно полностью загораживает собой старое. Ознакомившись с содержанием страницы, пользователь хочет перейти обратно к предыдущей странице и нажимает для этого кнопку **Назад** в своем браузере. Однако попасть на прежнюю страницу таким образом не удается: в «свежем» окне кнопка **Назад** попросту не работает. Такие ситуации дезориентируют и раздражают пользователей.

- если на вашем сайте присутствуют ссылки на другие сайты, время от времени проверяйте их, избавляясь от «мертвых» ссылок и обновляя ссылки на страницы, адрес которых изменился. К сожалению, в Сети очень много ссылок на давно исчезнувшие страницы. Не умножайте количество таких ссылок.

Гиперссылки — один из важнейших элементов наполнения веб-страницы. От правильной организации ссылок во многом зависит эффективность всего сайта и его привлекательность для пользователей.

Изображения и их текстовые эквиваленты

— Что ты там видишь, осел? — спрашивает петух.
— Вижу я, — отвечает осел, — сидят за столом разбойники, едят и пьют.

Я. Гримм, В. Гримм. Бременские музыканты

Эта книга посвящена текстовому содержанию сайтов, поэтому графическое оформление и композиция веб-страниц здесь не рассматриваются. Из этого не нужно делать вывод, будто автор отвергает всякое графическое оформление и считает, что страницы должны состоять из одного текста. Автор любит красиво оформленные сайты (разумеется, при условии, что ими удобно пользоваться). Просто тематика данной книги не имеет отношения к визуальному оформлению. По этим же соображениям здесь не идет речи о конфигурации веб-сервера, программировании на Java или настройке DNS. Обо всех этих вещах (безусловно, полезных и важных для разработчика сайтов) можно прочитать в другой хорошей литературе.

Тем не менее стоит сказать несколько слов про изображения, точнее — про связанный с ними текст.

Изображения включаются в документ HTML с помощью элемента `img`, например:

```
<img src="summer.jpg" width="180" height="235" alt="Летний пейзаж" />
```

Атрибуты элемента `img` имеют значения, приведенные в табл. 5.6.

Таблица 5.6. *Значения атрибутов элемента* `img`

Атрибут	Значения атрибута
`src`	адрес (URL) файла изображения
`width`	ширина изображения (в пикселах)
`height`	высота изображения (в пикселах)
`alt`	альтернативное текстовое описание

Обратите внимание: `img` — это строчный элемент. Если вы хотите расположить изображение в отдельном блоке, его можно включить в подходящий блочный элемент (например, `div`).

Дополнительно

Для тех изображений, которые должны располагаться в отдельном блоке, можно также указать свойство CSS `display` со значением `block`, например:

```
img#logo {
  display: block;
}
```

Внимание!

В XHTML 1.0 атрибут `alt` является обязательным для элемента `img`. Не забывайте указывать его.

Альтернативное текстовое описание предназначено для отображения в условиях, когда само изображение по каким-либо причинам недоступно. Это не такие редкие причины, как может показаться на первый взгляд:

- пользователь может отключить загрузку изображений из-за экономии трафика. При просмотре веб-страниц изображения в среднем составляют до ¾ загружаемой информации. Так как абсолютное большинство изображений не содержат ничего полезного для посетителя сайта, разумные пользователи предпочитают не загружать их;
- многие пользователи подключаются к Интернету по низкокачественным, медленным линиям связи. Не надо думать, что таковыми являются только небогатые люди, живущие в провинции. Все больше состоятельных людей постоянно имеют при себе компьютеры типа Notebook и с их помощью выходят в Интернет, находясь в дороге, на переговорах или на отдыхе. Качество связи при этом определяется местными условиями, которые не всегда благоприятны. В таких ситуациях отключение изображений обеспечивает гораздо более быструю загрузку веб-страниц;
- в нашей стране повсеместно распространяются портативные информационные устройства, с помощью которых пользователи также посещают сайты. Это, например, наладонный компьютер, специализированные сотовые телефоны, коммуникаторы. Экраны таких устройств не приспособлены для полноценного показа изображений — прежде всего из-за малого размера. Поэтому многие пользователи портативных компьютеров также отключают загрузку изображений;
- наконец, бывают пользователи, которые применяют текстовые или речевые браузеры. Это не только люди с ограниченными возможностями, но и те, кто временно оказывается в ситуации, когда обычные средства просмотра сайтов недоступны. Например, в автомобиле может быть установлен речевой браузер, с помощью которого водитель во время движения может прослушивать сайты с новостями и информацией о дорожных пробках.

Во всех перечисленных случаях альтернативное текстовое описание поможет пользователю, не загружая изображения, понять его суть.

Примечание

Конечно, речь не идет о случаях, когда изображения сами по себе являются той информацией, ради которой пользователь пришел на сайт. Например, если

пользователь хочет увидеть изображения товаров в сетевом магазине или репродукции в виртуальной картинной галерее, он сам позаботится о том, чтобы его техника отображала графическую информацию. Но если пользователь пришел на сайт не за изображениями, — использование сайта не должно быть для него затруднено.

Еще один важный аспект пользы от атрибута `alt` — это его важность для поисковых систем. Многие современные поисковые системы осуществляют поиск среди изображений. При этом поиске они в основном опираются на названия файлов и на содержание атрибута `alt`. Разве вы не хотите, чтобы изображения с вашего сайта оказывались в первых строчках результатов поиска?

纸 *Дополнительно*

В браузере Internet Explorer содержимое атрибута `alt` отображается в виде всплывающей подсказки даже тогда, когда сам рисунок доступен. Во многих других браузерах (например, Mozilla) оно при этом не отображается. Если вы хотите сделать «настоящую» всплывающую подсказку, используйте для этого не `alt`, а `title`:

```
<img src="summer.jpg" width="180" height="235"
alt="Летний пейзаж" title="Фото В. Пупкина" />
```

Текст «Летний пейзаж» будет показан вместо изображения, если оно по каким-то причинам недоступно. А текст «Фото В. Пупкина» будет показан, если изображение загружено и на него наведен указатель мыши.

Как же сделать альтернативное текстовое описание наиболее грамотно и эффективно? Как и составление заголовков, а также любых других текстов, это во многом творческая задача, для которой невозможно составить четкий формальный алогоритм решения. Тем не менее, следует соблюдать ряд рекомендаций, которые сами по себе не гарантируют идеального текстового описания, но по крайней мере позволяют избежать ряда ошибок и сделать это описание наиболее удобным и доступным для пользователя:

- пишите конкретно, что изображено на рисунке. Не нужно общих слов: «Фото», «Картинка», «Портрет» и т. д.;

Рекомендуется	Не рекомендуется
А. С. Грибоедов	Портрет
В. Пупкин выступает на конференции «Наука-2004»	В. Пупкин Фото с конференции Выступление
Рост продаж продукции «ABC» за июль – декабрь 2003	Продажи График продаж ABC

- пишите кратко — альтернативное текстовое описание предназначено не для того, чтобы полностью заменять собой изображение. (Если изображение можно без ущерба заменить текстом — тогда вообще никакого изображения не надо.) Такое описание нужно, чтобы пользователь получил только общее представление о сути изображения. Это позволит ему, в частности, решить, стоит ли загружать и просматривать само это изображение.

  Для большинства изображений удается составить текстовое описание, не превышающее 5–7 слов. Подбирайте такие слова, которые характеризуют наиболее важные детали изображения.

Примечание

Описание изображения зависит не только от самого этого изображения, но и от задач, для которых оно применяется. Например, на обучающей странице, рассказывающей об устройстве персонального компьютера, может быть помещена фотография с описанием: «Персональный компьютер». А на сайте торговой фирмы эта же фотография может быть размещена с описанием: «Компьютер ABC-Starlight на базе AMD Athlon».

Ставьте ключевые слова в именительный падеж, чтобы облегчить поиск ваших изображений с помощью поисковых систем. Наиболее важные слова помещайте в начале описания.

Рекомендуется	Не рекомендуется
В. Пупкин выступает на конференции «Наука-2004»	Выступление В. Пупкина на конференции «Наука-2004»
Михаил Щербаков. Концерт 31 октября 2003	Михаил Щербаков на концерте 31 октября 2003

Дополнительно

Разработчик практически не имеет возможности придать всплывающей подсказке (например, `title`) какое-то специальное оформление. Ее внешний вид определяется операционной системой и браузером пользователя. В частности, не существует универсального способа вставить во всплывающую подсказку перевод строки. Тем не менее, можно и нужно, наоборот, предотвращать перевод строки в тех местах, где он нежелателен. Например, в приведенных выше случаях:

```
В. Пупкин выступает на конференции «Наука&#8209;2004»
```

```
Михаил Щербаков. Концерт 31 октября 2003
```

Не путайте изображения, предназначенные для передачи содержания сайта, и изображения, играющие роль оформления. Содержание — это то, ради

чего пользователь, собственно, заходит на сайт. Поэтому к удобству и доступности содержания предъявляются повышенные требования. Если пользователь по какой-то причине не увидит оформления сайта (рамочек, логотипов, виньеток, шрифтовых выделений и т. д.) — это, может быть, печально, но не катастрофично. Но если он не увидит содержания, — это может оказаться фатально.

Конечно, если пользователь хочет увидеть изображение того продукта, который собирается купить, или репродукцию в виртуальной картинной галерее — он сам позаботится о том, чтобы его устройство просмотра поддерживало изображения. Но если пользователь хочет прочесть последние новости или узнать расписание работы Третьяковской галереи — он вполне может попытаться сделать это с помощью устройства, которое не показывает изображений. Такой пользователь будет очень разочарован, если без изображений он не сможет добраться до нужной ему информации (например, если на сайте графическая навигация, карты-изображения или графические заголовки на страницах).

Внимание!

Отключите изображения у себя в браузере и убедитесь, что в этом режиме текстовое содержание вашего сайта все равно остается доступным.

Примечание

Какое значение атрибута `alt` указывать, если изображение не несет никакой информации (допустим, оно является элементом оформления)? Не писать атрибут `alt` нельзя, потому что в XHTML он обязателен. Значит, остается указать пустое значение: `alt=""` (между кавычками нет ничего, даже пробела!).

Например: `<img src="bullet.gif" width="16" height="16" alt="" />`.

Во многих случаях удается отказаться от изображений благодаря тому, что нужного эффекта можно достичь с помощью CSS или с использованием специальных текстовых символов. Например, вместо того, чтобы делать графические кнопки в панели навигации, можно использовать обычные текстовые ссылки, а под них средствами CSS «подложить» красивые фоновые рисунки.

Дополнительно

Многие модные элементы оформления можно вставить в документ как обычные символы (табл. 5.7).

Таблица 5.7. *Примеры оформляющих символов*

Символ	Код Unicode	Десятичная подстановка
→	U+2192	`→`
►	U+25BA	`►`
•	U+25CF	`●`
☺	U+263A	`☺`
♥	U+2665	`♥`
♫	U+266B	`♫`

5.1.5. Полезные мелочи

Подстановки символов

> Смотрите, не презирайте ни одного из малых сих; ибо говорю вам, что ангелы их на небесах всегда видят лице Отца моего небесного.
>
> *Евангелие от Матфея, гл. 18, ст. 10*

В *главе 2* шла речь о том, что в тексте следует грамотно употреблять типографские символы — тире, кавычки, многоточия и т. д. Пришло время узнать, как практически вставлять эти символы в документ HTML.

Если вы набираете документ в кодировке UTF-8 — вам не приходится беспокоиться о специальных символах. Любой символ, будь то длинное тире, стрелка или рожица, вы можете вставить непосредственно в документ, пользуясь возможностями своего текстового редактора или таблицей символов.

А в расширенных кодировках ASCII для вставки специальных символов используются *подстановки символов* двух типов:

- ❒ именованные подстановки;
- ❒ числовые подстановки.

Именованная подстановка имеет вид:

`&имя;`

Например: `—` — длинное тире, `½` — дробь ½, `Ü` — латинская буква «u» с умляутом: ü. Многие распространенные подстановки (включая все типографские символы, упомянутые в *главе 2*) перечислены далее в табл. 5.8.

Дополнительно

В XHTML 1.0 более 250 именованных подстановок. В официальной документации они разделены на три группы и описаны в файлах:

www.w3.org/TR/xhtml1/DTD/xhtml-lat1.ent

www.w3.org/TR/xhtml1/DTD/xhtml-special.ent

www.w3.org/TR/xhtml1/DTD/xhtml-symbol.ent

С помощью подстановок можно грамотно разметить текст, приведенный на рис. 3.9:

```
&laquo;Плащ подберите, &mdash; посоветовал я, глядя на него
в зеркало заднего вида. &mdash; У вас плащ
защемляется&raquo;. Минут через пять все наконец устроилось.
Я спросил: &laquo;До Соловца километров десять?&raquo;
&mdash; &laquo;Да, &mdash; ответил горбоносый. &mdash;
Или немножко больше. Дорога, правда, неважная &mdash;
для грузовиков&raquo;.&mdash; &laquo;Дорога вполне
приличная, &mdash; возразил я. &mdash; Мне обещали, что
я вообще не проеду&raquo;.
```

Далеко не для всех символов существуют именованные подстановки. Существует более универсальный способ вставить в документ произвольный символ из диапазона Unicode. Для этого надо знать десятичный или шестнадцатеричный код нужного символа, чтобы сделать из него числовую подстановку.

Числовая подстановка имеет вид:

`&#код;` (если код десятичный) или `&#xкод;` (если код шестнадцатеричный).

Например, составим подстановку для знака номера (№). Код этого символа в Unicode — U+2116, следовательно, шестнадцатеричная подстановка будет выглядеть так: `№`. Если перевести число 2116 из шестнадцатеричной системы в десятичную, получим 8470. Следовательно, десятичная подстановка для знака № будет такова: `№`. Вы можете использовать любую из этих подстановок, чтобы вставить знак номера в свой документ. (Правда, традиционно используются именно десятичные, а не шестнадцатеричные подстановки.)

Внимание!

Именованная подстановка:

`&имя;`

Числовая подстановка:

`&#код;` (десятичная) или `&#xкод;` (шестнадцатеричная).

Не забывайте точку с запятой в конце подстановки!

Таблица 5.8. *Подстановки символов*

Код	Символ	Описание	Подстановки		
			именованная	десятичная	шестнадцатеричная
U+0022	"	двойные кавычки ASCII (суррогат)	"	"	"
U+0026	&	амперсанд	&	&	&
U+0027	'	одиночная кавычка ASCII (суррогат)	'	'	'
U+003C	<	знак «меньше»	<	<	<
U+003E	>	знак «больше»	>	>	>
U+00A0		неразрывный пробел			
U+00A3	£	знак фунта	£	£	£
U+00A5	¥	знак иены	¥	¥	¥
U+00A7	§	знак параграфа	§	§	§
U+00A9	©	знак авторского права	©	©	©
U+00AB	«	левые кавычки-«елочки»	«	«	«
U+00AD	-	мягкий перенос			
U+00AE	®	знак зарегистрированной торговой марки	®	®	®
U+00B0	°	знак градуса	°	°	°
U+00B1	±	плюс-минус	±	±	±
U+00B2	²	цифра 2 в верхнем индексе	²	²	²
U+00BB	»	правые кавычки-«елочки»	»	»	»
U+00BD	½	обыкновенная дробь ½	½	½	½
U+00D7	×	знак умножения	×	×	×
U+00F7	÷	знак деления	÷	÷	÷
U+0301	́	знак ударения (акут)		́	́

Таблица 5.8 *(окончание)*

Код	Символ	Описание	Подстановки		
			именованная	десятичная	шестнадцатеричная
U+03B1	α	альфа	α	α	α
U+03C0	π	пи	π	π	π
U+2009		короткий пробел			
U+2011	-	неразрывный дефис	–	‑	‑
U+2013	–	короткое тире	–	–	–
U+2014	—	длинное тире	—	—	—
U+2019	’	апостроф	’	’	’
U+201C	“		“	“	“
U+201D	”		”	”	”
U+201E	„		„	„	„
U+2022	•	маркер	•	•	•
U+2026	…	многоточие	…	…	…
U+2032	′	минуты	′	′	′
U+2033	″	секунды	″	″	″
U+2044	⁄	дробная черта	⁄	⁄	⁄
U+2048	⁇		–	⁈	⁈
U+2049	⁉		–	⁉	⁉
U+20AC	€	знак евро	€	€	€
U+2116	№	знак номера	–	№	№
U+2122	™	торговая марка	™	™	™
U+2212	−	минус	−	−	−
U+263A	☺	рожица	–	☺	☺
U+266D	♭	бемоль	–	♭	♭
U+266F	♯	диез	–	♯	♯

Зачем нужны именованные подстановки, если для любого символа есть числовые? Прежде всего, разработчику легче запомнить имена подстановок, чем их коды. Согласитесь, имена `−` и `€` гораздо удобнее для запоминания, чем коды `−` и `€`.

Примечание

В старых браузерах возникали проблемы с некоторыми подстановками. Например, Netscape 4 не отображает тире, включенные в документ с помощью именованных подстановок `—` и `–`. Но числовые подстановки `—` и `–` он прекрасно понимает.

Подстановки иногда вызывают проблемы при просмотре в браузере пользователя. Чтобы браузер мог отобразить нужный символ, этот символ должен находиться хотя бы в одном из шрифтов, доступных на компьютере. Например, символ ⚠ (U+26A0), хотя и присутствует в стандарте Unicode, пока еще не введен ни в один шрифт, поэтому браузер не сможет правильно обработать подстановку `⚠` или `&#26A0;`.

В современных операционных системах есть средства, позволяющие просмотреть список всех шрифтов, доступных на компьютере, и всех символов, имеющихся в этих шрифтах. (Например, в Windows для этого предназначена стандартная программа Таблица символов).

Изучите ассортимент шрифтов, которые поставляются по умолчанию с распространенными операционными системами. (Например, для Windows можно взять в качестве отправной точки те шрифты, которые присутствуют в Windows NT 4.0.) Даже если у вас на компьютере есть экзотический шрифт, в котором находятся символы ⚑, ♩ или 𝄞, — не стоит включать эти символы в веб-страницу с помощью подстановок, т. к. у подавляющего большинства пользователей они не будут правильно отображаться. Постарайтесь обходиться только теми символами, которые обычно бывают у пользователей. (Таких символов гораздо больше, чем вы думаете: например, символы карточных мастей ♠, ♣, ♦, ♥, математические знаки ≤, ≥, стрелки ←, ↑, →, ↓ и геометрические фигуры ■, ▲, ● — все эти и многие другие символы будут без проблем отображаться у большинства пользователей.)

Дополнительно

Иногда с необычными символами возникают проблемы, которые приходится решать специальными средствами. Предположим, браузер встретился с символом, который отсутствует в шрифте, применяемом для текущего текста. Как он должен поступить? Многие браузеры (например, Mozilla) ищут нужный символ в других комплектах шрифтов, установленных на компьютере. Символ, полученный таким образом, обычно заметно отличается по стилю от окружающих знаков, но это все же лучше, чем если бы его вообще не было! Другие браузеры ограничиваются только текущим шрифтом, так что вместо отсутствующих символов отображаются бесполезные квадратики или другие знаки (рис. 5.10).

Пустое множество □: ∀x x □ □. *а*

Пустое множество ∅: ∀x x ∉ ∅. *б*

Рис. 5.10. *а* — если не указать для этих символов специального шрифта, Internet Explorer их не отображает; *б* — а Mozilla не только отображает, но и выравнивает символы из разных шрифтов по размеру и базовой линии (если просто поместить друг за другом символы из разных шрифтов, такой набор будет выглядеть очень неровно)

Чтобы помочь браузерам находить нужный шрифт, его можно указать в таблице стилей. Например, чтобы в приведенном примере Internet Explorer показал недостающие символы, в файле HTML можно записать так:

```
Пустое множество <span class="math">&empty;</span>: <span
class="math">&forall;</span><var>x</var> <var>x</var> <span
class="math">&notin;</span> <span class="math">&empty;</span>
```

А в файле CSS при этом можно записать:

```
span.math {
  font-family: 'Arial Unicode MS', sans-serif;
}
```

Обязательные подстановки

Символ амперсанда (&) имеет вид стилизованной интерпретации латинского слова *et*, означающего «и». Само слово происходит от словосочетания «and per se and».

М. Клепер. Практическое руководство по цифровой печати

Как было сказано ранее, подстановки используются для вставки в документ символов, которых нет в текущей кодировке. Но существует несколько символов, которые есть в любой кодировке, так как они относятся к обычной ASCII, — и тем не менее их нельзя включить в документ никак, кроме как с помощью подстановок (обычно именованных).

Эти символы указаны в табл. 5.8, но имеет смысл повторить их отдельно:

Внимание!

<	`<`
>	`>`
&	`&`

Символ «<» означает начало тэга HTML. Поэтому, если такой знак встречается в документе, браузер не отобразит его на странице, а попытается интерпретировать как элемент разметки. Поэтому, чтобы все-таки вывести на страницу сам символ «<», нужно использовать именованную подстановку `<`. Из этих же соображений, чтобы вывести символ «>», применяется подстановка `>`.

К сожалению, не все разработчики знают о том, что сам символ «&» (амперсанд) требует подстановки. Причина здесь такая же, как в случае «<»: амперсанд в HTML обозначает начало подстановки, следовательно, если включать его в документ непосредственно, браузер может запутаться.

Примечание

Иногда браузеры снисходительно относятся к ошибкам разработчиков: бывают случаи, когда страница отображается как задумано, даже если на ней присутствуют символы «<» и «>» в отрыве от тэгов, символ «&» сам по себе, если некоторые подстановки не завершаются точкой с запятой. Но из этого не следует, что можно писать неграмотный код. Такой код, конечно, может сработать... но в какой-нибудь ситуации может привести и к неправильному отображению. Чтобы полностью исключить такие ошибки, соблюдайте правила XHTML.

Дополнительно

Для символа «"» (двойные кавычки ASCII) также существует именованная подстановка `"`. Необходимость в ней возникает очень редко, особенно если в тексте вы используете не этот суррогат, а грамотные парные кавычки. Пример практического применения `"` — вставка этого символа в значение атрибута (которое само ограничено кавычками ASCII), например:

```
<img src="quot.png" width="11" height="20" alt="Символ &quot;" />.
```

В этом случае, если изображение недоступно, на его месте отобразится текстовый эквивалент:

```
Символ "
```

Неразрывный пробел

> Для чего, зачем, почему так неразрывно соединил вас навеки?
>
> *М. Норбеков. Слово о полку рыцарей*

Неразрывный пробел (` ` или ` ` — числовая подстановка менее употребительна) применяется тогда, когда нужно предотвратить разрыв строки на месте пробела. Это может потребоваться по правилам набора (*см. главу 2*) или по другим причинам — например, для того, чтобы длин-

ный заголовок, не умещающийся на одной строке, разбивался не в произвольном месте, а там, где это подходит по смыслу.

Правильно	Неправильно
и т. д.	и␣т. ␣д. и␣т.д.
А. С. Пушкин	А. ␣С. ␣Пушкин А.С.Пушкин
в лес␣на охоту	в␣лес␣на␣охоту

Примечание

Не запутайтесь: неразрывный пробел применяется *вместо* обычного, а не вместе с ним. Если вы напишете `А.␣ ␣С.␣ ␣Пушкин` — никакого прока от неразрывных пробелов не будет.

Бывает, что неразрывные пробелы используются не по назначению. Например, начинающие разработчики любят кодировать абзацный отступ, размещая несколько ` ` подряд:

```
<p>     Все смешалось в доме Облонских.</p>
```

Это совершенно неграмотное решение. Для абзацного отступа (как и для любых других отступов и интервалов) следует использовать свойства CSS, речь о которых пойдет в *разд. 5.2*.

Также начинающие разработчики пользуются тэгом `<nobr>`, что тоже неграмотно: этого тэга нет в официальной спецификации HTML и XHTML, его поддержка — частная инициатива разработчиков некоторых браузеров. Если вы хотите разрабатывать код грамотно, не применяйте `<nobr>`.

Дополнительно

Использование ` ` не всегда приводит к оптимальному эффекту. Например, при выравнивании текста по ширине (точнее — при *выключке по формату*) многие браузеры растягивают обычные пробелы, но не изменяют неразрывные. В результате текст выглядит некрасиво, так как в одной строке получаются пробелы разной ширины (рис. 5.11).

Сначала машина двигалась скачками, и я был озабочен тем, чтобы удержаться в седле, обвившись ногами вокруг рамы и изо всех сил цепляясь за рулевую дугу. Краем глаза я смутно видел вокруг какие-то роскошные призрачные строения, мутно-зеленые равнины и холодное, негреющее светило в сером тумане неподалеку от зенита. Потом я сообразил, что тряска и скачки происходят от того, что я убрал ногу с акселератора, мощности двигателя (совсем как это бывает на автомобиле) не хватает, и машина, двигаясь неравномерно, то и дело натыкается на развалины античных и средневековых утопий. Я подбавил газу, движение сразу стало плавным, и я смог наконец устроиться поудобнее и оглядеться.

Рис. 5.11. Internet Explorer не масштабирует неразрывные пробелы

Автор решает эту проблему следующим образом. В тексте, выключенном по формату, неразрывные пробелы не используются. Вместо этого участки текста, внутри которых не должно быть перевода строки, заключаются в тэги `<span>...</span>`, а в таблице стилей при этом записывается:

```
span {
  white-space: nowrap;
}
```

Так как элемент `span` «в чистом виде» (то есть без атрибутов `id` или `class`) больше нигде не используется, такое его применение вполне целесообразно.

Это решение позволяет предотвращать разрыв строки и там, где нет пробелов — например, в случае обычного дефиса: `<span>как-нибудь</span>`. Наиболее грамотным решением в этой ситуации является неразрывный дефис (`‑`), но он поддерживается далеко не всеми браузерами.

Специальное использование подстановок

Тогда англичане позвали государя в самую последнюю кунсткамеру, где у них со всего света собраны минеральные камни и нимфозории, начиная с самой огромнейшей египетской керамиды до закожной блохи, которую глазам видеть невозможно, а угрызение ее между кожей и телом.

Н. Лесков. Левша

Включения текста на иностранных языках. С помощью подстановок вы легко можете включать в свой документ фрагменты текста почти на любом из существующих в мире языков, например:

В коде HTML	В окне браузера
`Слово «техника» происходит от греческого τέχνη`	Слово «техника» происходит от греческого τέχνη.
`«Pour les siècles à venir»,— поет Гренуар.`	«Pour les siècles à venir», — поет Гренуар.

Ударения. В русском языке иногда требуется ставить знак ударения над гласными буквами — например, чтобы различать «большу́ю часть» и «бо́льшую часть», «стре́лки» и «стрелки́», «уже́» и «у́же». Это можно сделать двумя способами:

- использованием специальной подстановки `́` (в Unicode это символ U+0301 — диакритический знак *акут*), которая помещается после ударной гласной:

В коде HTML	В окне браузера
`стре#769;лки`	стре́лки
`стрелки#769;`	стрелки́

- для букв *а*, *е*, *о*, *у* пользоваться символами латиницы с «готовыми» ударениями, например:

В коде HTML	В окне браузера
`большýю часть`	большу́ю часть
`бóльшую часть`	бо́льшую часть

К сожалению, оба этих способа создают проблемы для поисковых систем: многие системы не воспринимают слова с ударением. Кроме того, второй способ менее грамотен (ведь на самом деле в текст вставляются не русские, а латинские буквы!), хотя обеспечивает более красивый внешний вид текста. Но все же ударениями приходится пользоваться, если есть основания опасаться неправильного прочтения.

В электронной почте, телеконференциях и некоторых других случаях употребления формата «обычный текст» авторы иногда обозначают ударения прописными буквами: «большАя часть», «стрЕлки». Такое выделение не соответствует правилам русского языка и недопустимо в тексте на веб-страницах. Также недопустимо выделение ударной гласной с помощью курсива: «больш*а*я часть», «стр*е*лки».

纸 *Дополнительно*

Переносы. В современных браузерах нет встроенных средств для переноса слов, поэтому текст, выключенный по формату, часто смотрится некрасиво: во-первых, образуются слишком большие интервалы между словами, а во-вторых, часто бросается в глаза разница между этими интервалами в соседних строках.

Некоторые разработчики, желающие улучшить внешний вид текста, вручную или с помощью специальных программ (например, модуля работы с русским языком из TEX) расставляют в тексте сразу все возможные переносы с помощью подстановки `` (или ``, мягкий перенос). Это выглядит примерно следующим образом:

```
Во дни сом&shy;не&shy;ний, во дни тя&shy;гост&shy;ных
раз&shy;ду&shy;мий о судь&shy;бах мо&shy;ей ро&shy;ди&shy;ны,
— ты один мне под&shy;держ&shy;ка и опо&shy;ра, о
ве&shy;ли&shy;кий, мо&shy;гу&shy;чий, прав&shy;ди&shy;вый и
сво&shy;бод&shy;ный рус&shy;ский язык!
```

Тем не менее, такие переносы имеют много недостатков. Многие старые браузеры (Netscape 4, Internet Explorer 5.0) отображают мягкие переносы как обычные дефисы, в результате чего текст становится совершенно нечитаемым. Другие браузеры (например, современная Mozilla) просто игнорируют эти переносы, нередко нарушая при этом ровный правый край текста.

Автор в общем случае рекомендует разработчикам не пользоваться мягкими переносами, либо пользоваться в отдельных случаях на свой страх и риск, тщательно рассмотрев все преимущества и недостатки.

Математические формулы. HTML плохо подходит для набора текста с формулами. Тем не менее, в документ можно вставить некоторые простые формулы, пользуясь подстановками, например:

В коде HTML

```
&exist;<var>u</var> &forall;<var>z</var> 
(<var>z</var> &isin; <var>u</var> &hArr; 
&exist;<var>x</var> &isin; <var>v</var>: 
<var>&phi;</var>(<var>x</var>,<var>z</var>) &and; 
&forall;<var>w</var> (<var>&phi;</var>(<var>x</var>,<var>w</var>)
 &rarr; <var>z</var> = <var>w</var>))
```

В окне браузера

$$\exists u \forall z (z \in u \Leftrightarrow \exists x \in v : \varphi(x, z) \wedge \forall w (\varphi(x, w) \rightarrow z = w))$$

✓ *Примечание*

Не забывайте: подстановки символов (как именованные, так и числовые) можно свободно использовать не только в тексте страницы, но и в ее заглавии (элемент `title`), а также в значениях некоторых атрибутов — например, `alt` и `title`:

```
<img src="ill17.jpg" width="340" height="178" alt="Фрегат
&laquo;Паллада&raquo;" />
```

```
<a href="http://www.lowenbraubeer.com" title="Официальный
сайт L&ouml;wenbr&auml;u">Главная страница</a>
```

Также не забывайте, что, если вы разрабатываете страницу в кодировке UTF-8, вам нет необходимости пользоваться подстановками. Вводите все нужные символы в документ непосредственно — так же, как обычные буквы, цифры и знаки препинания.

5.1.6. Что дальше? Перспективы развития HTML и XHTML

> "A slow sort of country!" said the Queen. "Now, *here*, you see, it takes all the running *you* can do, to keep in the same place. If you want to get somewhere else, you must run at least twice as fast as that!"
>
> *Lewis Carroll. Through the Looking-Glass*[3]

Вы не достигнете успеха в разработке сайтов, если не будете следить за тенденциями в области технологий Всемирной Паутины. Сайты, которые вы создаете сейчас, рано или поздно придется обновлять, изменять, переделывать. Значит, имеет смысл сейчас делать сайты так, чтобы впоследствии не приходилось совершать лишнюю работу. Это не означает, что нужно гоняться за всеми модными новинками. Скорее наоборот — здоровый консерватизм чаще приводит к успеху. Но, вместе с тем, нельзя упускать из виду, что Сеть меняется. Меняются стандарты HTML, меняются компьютеры и браузеры, меняются пользователи. Если вы поймете, куда ведут эти изменения, ваша работа станет более эффективной и востребованной.

Язык HTML прекратил свое развитие еще в 1999 году, когда вышла его последняя редакция — HTML 4.01. С тех пор развиваются только языки разметки, базирующиеся на XML. Среди них — XHTML 1.0, которому посвящена эта глава, SVG (язык разметки векторной графики), MathML (язык разметки математических формул) и некоторые другие.

Дополнительно

XHTML 1.1, появившийся в 2001 году, не обеспечивает такой же хорошей совместимости со старыми редакциями HTML и старыми браузерами, как XHTML 1.0. Вместе с тем XHTML 1.1 не позволяет на практике достичь чего-либо, что было бы невозможно средствами XHTML 1.0. Поэтому, если вы разрабатываете

[3] «Какая медленная страна! — сказала Королева. — Нет, здесь, как видишь, приходится бежать так быстро, как только можешь, только чтобы остаться на месте. А если ты хочешь попасть куда-то еще, тебе придется бежать как минимум вдвое быстрее!» *(Л. Кэрролл. Алиса в Зазеркалье)*

страницы, рассчитанные на широкий круг пользователей Интернета, остановитесь на XHTML 1.0, а не на XHTML 1.1.

Новые браузеры все более внимательно относятся к версии языка разметки, указанной в прологе документа (в строке `<!DOCTYPE ... >`). Они применяют различные алгоритмы обработки страницы в зависимости от того, что именно написано в `<!DOCTYPE ... >`.

Примечание

О том, как Internet Explorer 6.0 по-разному обрабатывает страницы с разным `<!DOCTYPE ... >`, можно прочесть в Базе знаний Microsoft: **msdn.microsoft.com/library/en-us/dnie60/html/cssenhancements.asp**.

Дополнительно

В свою очередь, Mozilla также ориентируется на объявление типа документа и обрабатывает документ в соответствии с этим типом. Если `<!DOCTYPE ... >` не указан, Mozilla переключается в так называемый «quirks mode» (что можно приблизительно перевести как «режим уловок»); фактически браузеру приходится «угадывать», что имел в виду разработчик. А это угадывание не всегда приводит к правильному отображению страницы. Если же `<!DOCTYPE ... >` указан, но код страницы ему не соответствует, это еще хуже: Mozilla «поверит» такому коду и будет обрабатывать его так, как будто он написан грамотно, а это приводит к неправильному отображению еще чаще, чем в «quirks mode». (См. **www.mozilla.org/docs/web-developer/quirks**.)

Таким образом, разработчику следует указывать объявление типа документа и создавать код, соответствующий этому объявлению. Это важно для уже существующих браузеров и будет еще важнее для их следующих поколений. Стоит напомнить, что документ, соответствующий XHTML 1.0, автоматически соответствует также и HTML 4.0, так что проблем с поддержкой старых браузеров не будет. Зато, разрабатывая документы в строгом соответствии с XHTML 1.0, вы обеспечите их поддержку и в будущих версиях браузеров, так что вам не придется в спешке переделывать свой код, когда выйдет и распространится очередная версия Internet Explorer.

Между тем, на подходе новая редакция XHTML — язык XHTML 2.0. Консорциум W3C подготовил черновой проект этой редакции (см. **www.w3.org/TR/xhtml2**). Наступит время, когда браузеры станут поддерживать этот язык. Скорее всего, вы заинтересованы в том, чтобы облегчить себе переход на новую редакцию XHTML и опередить менее расторопных конкурентов. Значит, уже сейчас вы должны представлять себе, чем новый язык отличается от того, к чему вы привыкли.

XHTML 2.0 обеспечивает полное отделение содержания документа от его представления. В нем нет ничего отвечающего за внешний вид документа — нет

элементов `font`, `i`, `b`, `center`, нет атрибутов `align`, `size` и `bgcolor`. Предполагается, что вся информация об оформлении документа содержится в отдельной таблице стилей.

Собственно говоря, если вы разрабатываете документы на XHTML 1.0 Strict и при этом не пользуетесь никакими «оформительскими» элементами и атрибутами — считайте, что переход на XHTML 2.0 пройдет для вас совершенно безболезненно. Практически единственное, что вам потребуется сделать, — это заменить строку `<!DOCTYPE ... >` в прологе.

Дополнительно

На самом деле, потребуются еще некоторые поправки. Например, придется отказаться от атрибутов `name` и `lang`. Кроме того, в XHTML 2.0 вводится много новых элементов — например, элемент `section`, обозначающий структурную часть документа, и элемент `h` — заголовок общего вида (уровень заголовка будет определяться уровнем той структурной части, к которой он относится). Но даже с учетом подобных изменений переход на XHTML 2.0 будет значительно легче, если вы уже сейчас грамотно отделяете структуру документа от его представления.

5.2. Красота требует жертв

Оформление текстового содержания веб-страниц с помощью CSS

— И все-таки мы должны тебя оформить, — сказал Луи.
— Но не оформим, — сказал Билл, — запишем, что ты ехал 75, хотя ты ехал 80.

Ч. Буковски. Макулатура

Иногда, глядя с крыльца на двор и на пруд, говорил он о том, как бы хорошо было, если бы вдруг от дома провести подземный ход или чрез пруд выстроить каменный мост, на котором бы были по обеим сторонам лавки, и чтобы в них сидели купцы и продавали разные мелкие товары, нужные для крестьян.

Н. Гоголь. Мертвые души

Таблицы стилей уже широко распространились во Всемирной Паутине. Многие создатели сайтов на собственном опыте убедились, что технологии CSS предоставляют широкие возможности для оформления внешнего вида сайтов и позволяют оптимизировать процесс разработки.

Консорциум W3C рекомендует использовать таблицы стилей для того, чтобы отделять содержание и структуру страниц от их представления. Мы начнем разговор про CSS с разъяснения того, что же такое содержание, структура и представление и почему их важно разделять.

5.2.1. Три кита
Содержание, структура, представление

Divide et impera[4].

Н. Маккиавелли

Внимание!

Содержание — та информация, которую получает пользователь.

Структура — принцип организации этой информации.

Представление — внешний вид, который воспринимает пользователь.

Содержание веб-страницы — это то, что эта страница сообщает пользователю; та информация, которую она передает. Содержание — наиглавнейшая составляющая страницы, то, ради чего эта страница вообще существует.

Эта книга посвящена текстовому содержанию сайтов, но вообще содержание может быть не только текстовым. Например, в виртуальной картинной галерее содержанием являются цифровые репродукции картин, а в архиве бесплатно распространяемых программ содержанием могут являться файлы с исходным кодом и установочными комплектами этих программ.

Интерес пользователей к сайту определяется в первую очередь его содержанием. Пользователи посещают сайты именно потому, что хотят получить содержание этих сайтов. Поэтому внимание разработчика должно быть направлено в первую очередь именно на содержание.

Примечание

Многие разработчики сайтов любят говорить о таких вещах, как «концепция», «стиль», «дизайн», «креатив» и т. д. Конечно, оригинальная концепция и великолепный стиль могут существенно улучшить эффективность сайта и его привлекательность для пользователя — но только при условии, что сайт предоставляет нужное и интересное содержание.

---

[4] Разделяй и властвуй (*лат.*).

В самом деле, первый вопрос, который должен задавать себе разработчик, — «*Зачем* пользователи будут заходить на мой сайт?» Если вовремя не задуматься над этим — может получиться красивый, но никому не нужный сайт.

Структура страницы — это взаимосвязь ее логических элементов: заголовков, обычного текста, перечней, гиперссылок и т. д. Пользователю удобнее воспринимать содержание, если оно организовано в стройную и логичную структуру. В самом деле, подумайте, как вы пользовались бы словарем, если бы слова в нем шли не по алфавиту, а в случайном порядке? Алфавитный порядок — один из типичных примеров того, как содержанию придается определенная структура.

На веб-страницах для структурной организации используются элементы HTML, которым была посвящена первая часть этой главы.

Наконец, *представление* страницы — это та форма, в которой пользователь ее воспринимает. Большинство пользователей просматривают страницы на экранах компьютеров и других устройств. Значит, для них представление страницы — это то же самое, что ее внешний вид, оформление.

Если пользователь прослушивает веб-страницу с помощью речевого браузера или «прощупывает» ее браузером Брайля — значит, представлением является ровно то, что он слышит или нащупывает.

Таблицы стилей CSS как раз и предназначены для описания представления страницы.

Примечание

Научитесь четко различать, что относится к содержанию страницы или документа, что — к структуре, а что — к представлению. Например, посмотрите на ближайший предыдущий заголовок: «Три кита». Содержание этого текста — это и есть два слова: «Три кита», и больше ничего. Роль этого текста в структуре заключается в том, что он является заголовком части главы. Информация о том, что это не просто текст, а заголовок, позволяет нам включить его в содержание книги и использовать при поиске. А представление этого текста — его внешний вид: шрифт Arial, кегль 9 пунктов, светлый, выключенный по ширине, с отступом слева 1 см.

Содержание, структура и представление — три разные вещи, но они тесно связаны между собой. Структура и представление страницы должны соответствовать ее содержанию и помогать пользователю воспринимать это содержание. С другой стороны, разумеется, для одного и того же содержания нередко можно придумать несколько различных удачных структур и много разных вариантов представления.

По этой причине рекомендуется разделять содержание, структуру и представление веб-страниц. Разработчик должен иметь возможность легко внести изменения в любой из этих компонентов, не затрагивая при этом два остальных.

5.2.2. Курс молодого бойца
Основы кодирования таблиц стилей

> Они улыбались и говорили: «освоитесь». Вот тебе и освоитесь. А если грыжу привезут? Объясните, как я с ней освоюсь?
>
> *М. Булгаков. Записки юного врача*

Начальные сведения

> Я спросил мастера-наладчика о порядке ее работы, на что он мне ответил:
> — Слухай сюда! Положь колдобину со стороны загогулины и два раза дергани за пимпочку. Опосля чего долбани плюхалкой по кувыкалке и, кады чвакнет, отскочь дальшее, прикинься ветошью и не отсвечивай.
>
> *М. Задорнов. Задание выполнено*

В строке 08 листинга 5.1 находится элемент `link`, позволяющий подключить к документу внешнюю таблицу стилей. Если вы недостаточно хорошо представляете себе, как это делается, — сейчас самое время восполнить этот пробел.

Возьмите какой-нибудь документ XHTML, в котором есть хотя бы один элемент `h1` (заголовок 1-го уровня). Убедитесь, что этот документ соответствует шаблону, приведенному в листинге 5.1. На всякий случай проверьте его валидатором.

Теперь создайте в текстовом редакторе еще один файл. Содержимым этого файла будут всего три строчки (листинг 5.9).

Листинг 5.9. Простая таблица стилей

```
h1 {
  color: #009933;
}
```

Сохраните этот файл под именем file.css в том же каталоге, где находится ваш документ HTML. Сейчас вы сделали простую таблицу стилей.

Теперь вернитесь к документу HTML, найдите там элемент `link`, о котором говорилось ранее, и атрибут `href` этого элемента. Запишите в кавычках после `href=` имя таблицы стилей: `file.css`. В результате элемент `link` будет выглядеть следующим образом:

```
  <link rel="stylesheet" type="text/css" href="file.css" media="all" />
```

Теперь сохраните измененный документ HTML и откройте его в браузере. Если вы все сделали правильно, то заголовок `h1`, имеющийся у вас в документе, окрасится в зеленый цвет.

Разберем теперь подробнее тот код CSS, который вы написали. Каждая конструкция в этом коде имеет свое собственное название и назначение (рис. 5.12).

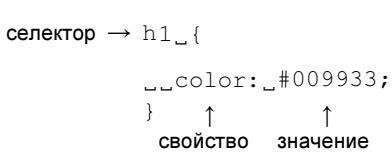


Рис. 5.12. Основные конструкции CSS

Все начинается с *селектора*. Он показывает, какие именно элементы мы собираемся оформлять. В данном случае мы написали `h1`, и оформление подействовало на заголовки первого уровня. Замените `h1` на `p`, и вы увидите, что заголовок перестанет быть зеленым, зато окрасятся абзацы обычного текста. Попробуйте подставлять вместо `h1` имена других элементов, которые есть в вашем документе, и убедитесь, что оформление каждый раз действует именно на эти элементы.

纸 *Дополнительно*

В качестве селектора может быть указано не имя элемента, а что-то другое. Например, если вы напишете

```
* {
  color: #009933;
}
```

(в качестве селектора здесь стоит звездочка) — то в современных браузерах оформление подействует на *все* элементы документа.

Правила составления селекторов в CSS 2 достаточно разнообразны и сложны. Эта книга — не учебник по CSS, поэтому здесь будут рассмотрены только некоторые, наиболее употребительные виды селекторов. Если вы хотите освоить CSS в полном объеме — пожалуйста, прежде всего прочтите официальную

документацию от консорциума W3C, которую можно найти по адресу **www.w3.org/TR/CSS2**.

После селектора следует открывающая фигурная скобка. Затем, на следующей строке, располагается пара, состоящая из свойства и его значения. *Свойство* — это та характеристика элемента, которую вы хотите изменить. В данном случае мы решили изменить цвет элемента и для этого написали свойство `color`. Попробуйте написать не `color`, а `background-color` и посмотрите, к чему это приведет.

CSS позволяет изменить не только цвет, но и любые другие параметры оформления — мы можем изменить размер шрифта, сделать его полужирным или курсивным, выровнять по левому или правому краю страницы, а также по ширине, заключить в рамку, расположить в определенном месте экрана или вовсе сделать невидимым.

Примечание

Полный список свойств, имеющихся в CSS 2, можно найти на странице **www.w3.org/TR/CSS2/propidx.html**.

Наконец, *значение* свойства — это то, каким мы хотим сделать указанное свойство. Например, цвет может быть зеленым, белым, желтым или еще каким угодно. В самом деле, попробуйте заменить `#009933` на `#993300`.

Свойство и его значение разделяются двоеточием. После значения обязательно следует точка с запятой.

Наш код завершается строкой, состоящей из одной закрывающей фигурной скобки. Эта скобка — парная к той, которая открылась в первой строке.

Внимание!

Синтаксис CSS:

```
селектор␣{
␣␣свойство:␣значение;
␣␣свойство:␣значение;
␣␣свойство:␣значение;
␣␣...
}
```

Не забывайте точку с запятой после каждой строки!

Можно присвоить одному и тому же элементу два и больше свойств, например, так:

```
h1 {
  color: #009933;
  background-color: #ffffcc;
}
```

Если требуется оформить несколько различных элементов, в таблицу стилей можно дописать код CSS для этих элементов:

```
h1 {
  color: #009933;
  background-color: #ffffcc;
}
h2 {
  color: #006600;
}
p {
  background-color: #ffffcc;
}
```

Попробуйте сделать так, чтобы каждый элемент, имеющийся в вашем документе HTML, приобрел какой-нибудь особый цвет. Возможно, страница при этом будет выглядеть слишком пестро, зато вы освоите основы CSS.

Замечания о кодировании CSS

И человека растянуть
Он позволял не как-нибудь,
Но в строгих правилах искусства,
По всем преданьям старины.

А. С. Пушкин. Евгений Онегин

В CSS, как и в HTML, рекомендуется придерживаться какого-либо определенного стиля кодирования. Вы можете использовать тот стиль, который описан в этой книге, или какой-нибудь еще — но, в любом случае, соблюдайте единообразие и аккуратность. Это сэкономит ваше время и силы, если потребуется вносить изменения в файл CSS, который вы в последний раз редактировали полгода назад.

Автор применяет такой стиль:

- селектор и открывающая фигурная скобка записываются на одной строке;
- каждое свойство и его значение записываются в отдельной строке, причем перед свойством делается отступ величиной в два пробела;
- закрывающая фигурная скобка записывается на отдельной строке, без отступов.

Если вам удобнее следовать какому-то другому стилю — вы имеете на это полное право. Например, можно записывать открывающую фигурную скобку в отдельной строке:

```
h1
{
  color: #009933;
  background-color: #ffffcc;
}
```

а можно, наоборот, писать все в одной строке:

```
h1 {color: #009933; background-color: #ffffcc;}
```

Правда, в первом случае увеличивается количество строк в файле CSS, и файл приходится больше прокручивать, чтобы добраться до нужной конструкции. А во втором случае строки становятся слишком длинными и неудобными для редактирования.

Для одного и того же селектора можно записать несколько конструкций:

```
h1 {
  color: #009933;
}
h1 {
  background-color: #ffffcc;
}
```

а можно объединить свойства в одной конструкции, как в предыдущих примерах.

Примечание

В некоторых старых браузерах могут возникать проблемы из-за того, что в CSS пропущен пробел или точка с запятой. Синтаксис CSS действительно позволяет во многих местах обходиться без этих знаков — например, можно не ставить точку с запятой после самого последнего свойства, перед закрывающей фигурной

скобкой. Но чтобы гарантированно избежать затруднений со старыми браузерами — стоит придерживаться следующих правил:

- после каждой пары *свойство: значение* ставьте точку с запятой;
- после двоеточия между свойством и значением ставьте пробел;
- если несколько свойств записаны в одной строке, ставьте пробел после каждой точки с запятой.

Рекомендуется	Не рекомендуется

h1␣{
␣␣color:␣#009933;
␣␣background-color:␣#ffffcc;
}
	h1␣{
	␣␣color:␣#009933;
	␣␣background-color:␣#ffffcc
	}

␣␣color:␣#009933;	␣␣color:#009933;
h1␣{color:␣#009933;␣ background-color:␣#ffffcc;␣}	h1␣{color:␣#009933;background-color:#ffffcc;}

В таблицу стилей можно вставлять комментарии. Комментарий начинается парой символов /* (косая черта, звездочка) и заканчивается парой этих же символов в обратном порядке: */, например:

```
p␣{
␣␣background-color:␣#ffffcc; /* бледно-желтый фон */
}
```

Комментарии можно вставить в любую строку, где уже записана какая-либо конструкция CSS (как в примере ранее), а можно записать отдельной строкой или распространить на несколько строк.

Примечание

Автор предпочитает снабжать собственные таблицы стилей комментарием, информирующим о назначении, дате последней редакции и некоторых других свойствах файла. Вот пример одного из таких комментариев (обычно он ставится в самом начале файла):

```
/*
  www.philigon.ru
  /css/index.css
  Главная таблица стилей
```

```
  10.04.2004
*/

* {
  margin: 0;
  padding: 0;
}
body {
  background-color: #ffffff;
}
...
```

В предыдущих примерах файл CSS располагался в том же каталоге, что и файл HTML, к которому он подключен. Разумеется, это не обязательно. Таблица стилей может располагаться и в любом другом каталоге. Чтобы подключить таблицу стилей, находящуюся в другом каталоге, нужно указать в атрибуте `href` путь к файлу так же, как это делается для гиперссылок, например:

```
<link␣rel="stylesheet"␣type="text/css"␣href="../css/file.css"␣media="all"␣/>
```

Такая запись означает, что для доступа к таблице стилей нужно подняться на один уровень относительно текущего документа HTML, там зайти в каталог `css` и подключить файл `file.css`, находящийся в этом каталоге.

Примечание

Для имен файлов CSS справедливы все те же принципы, о которых рассказывалось ранее в *разд. «Имена файлов HTML»*. Рекомендуется использовать в именах файлов только строчные латинские буквы и цифры, не делать имена слишком длинными, но вместе с тем добиваться, чтобы они были информативными.

Всегда указывайте расширение файла `css`.

К одному и тому же документу HTML можно подключить две и более разных таблицы стилей, записав для каждой из них отдельный элемент `link`:

```
<link␣rel="stylesheet"␣type="text/css"␣href="../css/main.css"␣media="all"␣/>

<link␣rel="stylesheet"␣type="text/css"␣href="special.css"␣media="all"␣/>
```

Если инструкции CSS в этих файлах не противоречат друг другу, то веб-страница будет оформлена с учетом обеих таблиц стилей.

Наконец, один и тот же файл CSS можно подключить к нескольким разным документам HTML. Это одно из самых полезных свойств таблиц стилей, так как оно позволяет оформить несколько страниц единообразно, не указывая правила оформления отдельно для каждой страницы.

Виды селекторов CSS

> Так психиатр и провел всю вторую половину этого прохладного, напоенного кондиционированным воздухом, такого долгого дня. Телефон, радио, селектор, телефон, радио, селектор, телефон, радио, селектор, телефон, радио...
>
> *Р. Брэдбери. Убийца*

В CSS существует множество видов селекторов. Перечислим те из них, которые употребляются наиболее часто и хорошо поддерживаются браузерами (табл. 5.9).

Внимание!

Таблица 5.9. *Виды селекторов CSS*

Селектор	Значение селектора
Селектор по элементу	`имя элемента`
Селектор по `id`	`#значение_атрибута_id`
	`имя_элемента#значение_атрибута_id`
Селектор по классу	`.значение_атрибута_class`
	`имя_элемента.значение_атрибута_class`
Контекстный селектор	`селектор1␣селектор2␣...␣селекторN`

Селектор по элементу. Это тот самый селектор, который употребляется во всех предыдущих примерах. В качестве селектора указывается имя элемента HTML, и он действует на все элементы в документе, имеющие это имя.

Селектор по `id`. Если требуется особенным образом оформить *один-единственный* элемент во всем документе, можно присвоить этому элементу атрибут `id`, например:

```
<h2 id="new">Красота требует жертв</h2>
```

Затем в таблице стилей нужно записать селектор, состоящий из символа решетки (#), непосредственно за которым следует значение атрибута id (в данном случае — new):

```
#new {
  color: #996633;
}
```

Можно указать перед решеткой имя элемента HTML:

```
h2#new {
  color: #996633;
}
```

Если у нас только один документ HTML, нет никакой разницы, указывать ли в этом селекторе имя элемента. Всякое значение id может быть только у одного элемента в документе: например, если уже есть <h2 id="new">, то в этом же файле HTML нельзя записать <p id="new"> или даже еще один заголовок <h2 id="new">.

Дополнительно

Но если одна и та же таблица стилей применяется для оформления нескольких страниц, разница между селекторами #new и h2#new может проявиться: в одном документе атрибут id="new" может быть у одного элемента, а в другом этот же атрибут — совершенно у другого.

Селектор по классу. Если требуется одинаково оформить несколько элементов, можно присвоить каждому из этих элементов атрибут class с одним и тем же значением, например:

```
<h3 class="abstract">...</h3>
...
<p class="abstract">...</p>
```

Затем в таблице стилей нужно записать селектор, состоящий из точки (.), непосредственно за которой следует значение атрибута class (в данном случае — abstract):

```
.abstract {
  background-color: #cccccc;
}
```

Этот селектор действует на все элементы, имеющие атрибут class со значением abstract. В отличие от атрибута id, атрибут class может иметь одно и то же значение у нескольких элементов.

Селектор по классу также можно использовать вместе с именем элемента, например:

```
h3.abstract {
  background-color: #cccccc;
}
```

В этом случае селектор будет действовать только на заголовки третьего уровня (`h3`), имеющие атрибут `class` со значением `abstract`, и не будет действовать, например, на абзацы текста (`p`) или элементы перечня (`li`), даже если в их открывающем тэге написано `class="abstract"`.

Контекстный селектор. Этот селектор применяется, когда оформление элемента должно зависеть не только от самого этого элемента, но и от того, в какой элемент он вложен. (Например, мы хотим раскрасить в определенный цвет каждый отдельный пункт маркированного перечня, но при этом оставить нераскрашенными нумерованные перечни.)

Контекстный селектор состоит из двух или нескольких селекторов одного из предыдущих типов, описанных ранее. Эти селекторы отделяются друг от друга пробелами, например:

```
ul␣li␣{
␣␣color:␣#663300;
}
```

Такой селектор будет действовать на все элементы `li` при условии, что эти элементы вложены в элемент `ul`. На элементы `li`, вложенные в `ol`, он не действует.

Можно составить более сложный контекстный селектор:

```
table#main␣thead␣tr.category␣th.price
```

Такой селектор будет действовать на элемент `th` с атрибутом `class="price"`, вложенный в элемент `tr` с атрибутом `class="category"`, который вложен в элемент `thead`, вложенный в элемент `table` с атрибутом `id="main"`.

Примечание

Пробелы в контекстных селекторах играют очень важную роль. Например, такой селектор:

```
p.special
```

действует на элемент `p`, у которого есть атрибут `class="special"`. А такой селектор:

```
p␣.special
```

действует на любой элемент с атрибутом `class="special"`, который вложен в элемент `p`.

Если записать несколько селекторов через запятую и пробел, получится *объединенный селектор*, который будет действовать на все перечисленные элементы. Например, такая конструкция:

```
h1, h2, h3 {
  color: #003366;
}
```

раскрасит в темно-синий цвет все заголовки `h1`, `h2` и `h3`.

Дополнительно

Одному и тому же элементу можно присвоить атрибуты `id` и `class`, например:

```
<p class="new" id="summary">
```

Чтобы оформить с помощью CSS именно этот элемент, можно записать такой комбинированный селектор:

```
p.new#summary
```

Синтаксис HTML не запрещает присвоить одному и тому же элементу несколько классов, например, так:

```
<ul class="toc special named">
```

Такой элемент `ul` должен «откликаться» на любой из трех селекторов CSS:

```
ul.toc
ul.special
ul.named
```

К сожалению, не все браузеры хорошо поддерживают множественные классы. Поэтому этот прием стоит использовать с большой осторожностью, проверив его работу в разных браузерах.

Преимущества и недостатки CSS

Как отделить от дегтя мед,
Иль деготь выбросить из меда?

П. Вяземский

Если вы немного знакомы с HTML, то знаете, что оформить веб-страницу можно и без помощи CSS. Например, в HTML 4.01 Transitional можно было применить такую конструкцию:

```
<h2><font size="6" color="#006633" face="Verdana"><b>Глава 3</b></font></h2>
```

— и заголовок «Глава 3» отображался крупным полужирным шрифтом зеленого цвета, гарнитуры Verdana. Зачем же использовать CSS?

Дело в том, что таблицы стилей обеспечивают определенные преимущества, наиболее важные из которых следующие:

- в таблице стилей можно задать оформление не только для какого-то одного конкретного элемента, но и для всех элементов, удовлетворяющих указанному селектору. Например, можно написать:

  ```
  h2 {
    color: #006633;
    font: bold 2em Verdana, Arial, Helvetica, sans-serif;
  }
  ```

  — и *все* заголовки второго уровня приобретут одинаковое оформление. Не требуется записывать в HTML оформление для каждого заголовка.

  Если теперь потребуется изменить оформление — например, во всех заголовках на всех страницах сайта поменять шрифт Verdana на шрифт Tahoma — достаточно будет обновить одну строку в таблице стилей. Если бы оформление создавалось средствами HTML, пришлось бы в каждом файле перебирать все заголовки и менять «Verdana» на «Tahoma»;

- многие эффекты оформления доступны только в CSS, их нельзя добиться с помощью HTML. Среди таких эффектов, например, р а з р я д к а текста, надчеркивание и измененный междустрочный интервал, разнообразные рамки, а также наложение элементов друг на друга. Чтобы не запутаться в том, какое оформление сделано средствами HTML, а какое — средствами CSS, лучше делать все с помощью таблиц стилей;

- таблицы стилей позволяют отделить содержание и структуру страницы от ее представления. Если требуется изменить внешний вид страницы, не меняя содержания, можно внести изменения только в файл CSS и при этом быть твердо уверенным, что в содержании и структуре документа вы ничего не нарушили, даже случайно.

  Кроме того, для одной и той же страницы вы можете сделать две или несколько таблиц стилей — например, одну для обычных брузеров, а другую — для наладонных компьютеров и других портативных устройств.

Вместе с тем, у CSS есть и недостатки. Главный недостаток заключается в том, что браузеры не всегда корректно интерпретируют те или иные инструкции CSS, и внешний вид страницы искажается. Впрочем, этот недостаток касается скорее верстки страницы и расположения элементов, чем оформления текстового содержания.

纸 Дополнительно

В самом деле — многие разработчики применяют верстку с помощью таблиц, т. к. она позволяет «закрепить» композицию страницы и обеспечить одинаковое относительное расположение элементов при просмотре в разных браузерах. Такое использование таблиц противоречит рекомендациям Консорциума W3C:

он предлагает применять таблицы HTML только для разметки табличных данных, а не для оформления внешнего вида страницы.

Однако несовершенная поддержка CSS в современных браузерах (прежде всего в Internet Explorer) приводит к тому, что средствами CSS иногда довольно трудно осуществить сложную верстку. Кроме того, приходится уделять чрезмерно много внимания тестированию сайта в разных браузерах и решению проблем совместимости.

Тем не менее, если ограничиться только современными браузерами (начиная с Internet Explorer 5.0), для них всегда можно сверстать страницу с помощью CSS. Нет ничего такого, чего можно было бы добиться «табличной версткой» и нельзя было бы осуществить через таблицы стилей. Правда, поиск решения иногда требует слишком много времени и повышенной квалификации разработчика.

Автор рекомендует всегда, когда позволяют ваши возможности, верстать страницы средствами CSS и не использовать «табличной верстки». Однако эта рекомендация — не непреложный закон: существует немало хороших, эффективных и удобных сайтов, где верстка страниц сделана с помощью таблиц.

В этой главе рассказывается только про оформление текстового содержания веб-страниц, и нет ничего про верстку с помощью CSS. Вы можете найти весь недостающий материал в официальной спецификации CSS, а также в книгах (например, [19], [20], [21]).

В том, что касается оформления текста, поддержка CSS браузерами тоже далека от совершенства. Здесь, впрочем, не возникает таких серьезных проблем, как при верстке страницы, поэтому применение CSS для оформления текста — практически всегда более правильное решение, чем использование оформительских тэгов HTML вроде `<b>` или `<font>`.

Далее в этой главе будут перечислены все свойства CSS, имеющие отношение к оформлению текстового содержания.

5.2.3. ...Чтоб картинка ожила
Свойства CSS для оформления текстового содержания веб-страницы

Мы притягиваем большие, общие окружающие света, способные исправить самые разные свойства в каждом.

Из бесед М. Лайтмана

Свойства шрифта

> Большинство читателей не замечают ни шрифта, ни букв. Зато видят слова. И не наслаждаются красотой композиции, а просто читают. И ничего более.
>
> *Дж. Феличи. Типографика*

> Ничего, я вас не совсем изобью, только малость разукрашу, чтоб вспомнили, как это бывает!
>
> *У. Фолкнер. Город*

В этом разделе рассказывается о таких свойствах CSS, которые могут быть применены к произвольному участку текста — в том числе к одному-единственному символу. Впрочем, все эти свойства можно применять и к сколь угодно крупным участкам текста — например, к заголовку, к целому абзацу или вовсе ко всей веб-странице.

Цвет: *color* и *background-color*

Как понятно из предыдущих примеров, с помощью CSS можно придать определенный цвет тексту и фону под ним, используя для этого свойства `color` и `background-color`.

Внимание!

`color` — цвет текста

`background-color` — цвет фона

Использование:

- `color: #xxxxxx;`
- `background-color: #xxxxxx;`

Значения цветов лучше всего указывать в стандартной *шестнадцатеричной модели RGB* (Red-Green-Blue):

#xxxxxx

Здесь вместо каждого символа × должна стоять шестнадцатеричная цифра (от 0 до f). Первые две цифры отвечают за красную (red) составляющую цвета, вторые — за зеленую (green), третьи — за синюю (blue). Чем больше число в каждой из трех позиций, тем ярче эта составляющая в данном цвете.

Зная это, можно строить многие цвета «из головы». Например, код `#000000` означает, что яркость каждой составляющей равна 0 — следовательно, это черный цвет. Соответственно, белый цвет получается, если все составляю-

щие будут максимальны: `#ffffff` (напомним, $FF_{16} = 255$). Если взять максимальное значение красной составляющей и оставить `00` в двух остальных, получится ярко-красный цвет: `#ff0000`. Таким же образом можно получить ярко-синий и ярко-зеленый цвет.

Смешивая основные цвета друг с другом, можно получить дополнительные цвета. Например, если смешать ярко-красный и ярко-синий, получится ярко-фиолетовый: `#ff00ff`. Основные и дополнительные цвета перечислены в табл. 5.10.

Таблица 5.10. *Основные и дополнительные цвета*

Код	Цвет	Код	Цвет
`#000000`	белый	`#0000ff`	ярко-синий
`#ffffff`	черный	`#ff00ff`	ярко-фиолетовый
`#ff0000`	ярко-красный	`#00ffff`	ярко-бирюзовый
`#00ff00`	ярко-зеленый	`#ffff00`	ярко-желтый

В любом графическом редакторе есть палитра цветов, из которой можно узнать шестнадцатеричный код любого нужного вам цвета (рис. 5.13).

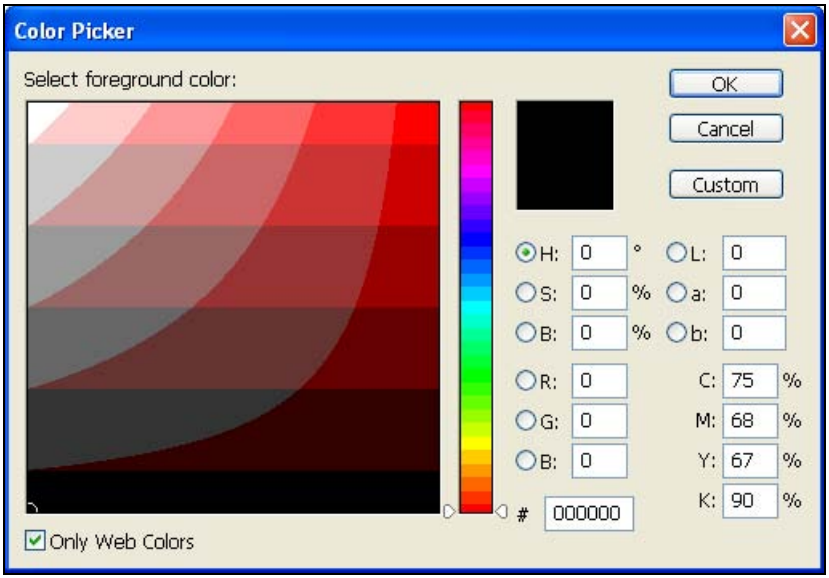


Рис. 5.13. Инструмент **Color Picker** в Adobe Photoshop позволяет посмотреть код выбранного цвета в любой из четырех цветовых моделей, причем в модели RGB — и в десятичном, и в шестнадцатеричном коде. Не забывайте установить флажок **Only Web Colors**

Для цвета текста, фона и рамок рекомендуется использовать цвета из так называемой *безопасной палитры* (или *веб-палитры*, Web colors). Эти цвета характеризуются тем, что каждая из трех составляющих цвета может иметь только одно из следующих значений: `00`, `33`, `66`, `99`, `cc`, `ff`. Например, цвета `#ccff66` и `#000099` относятся к безопасной палитре, а цвета `#808080` и `#abcdef` — не относятся.

Если употреблять для текста, фона и рамок цвета не из безопасной палитры, могут возникнуть проблемы с их отображением в браузере, например:

- браузеры используют разные алгоритмы для преобразования кода в цвет, который пользователь видит на экране. Один и тот же код (если он не относится к безопасной палитре) может привести к немного разным оттенкам в разных браузерах;
- иногда браузеры не отображают точно тот цвет, который требуется, а «округляют» код до ближайшего, который могут отобразить. В этом случае, если не используются цвета из безопасной палитры, может получиться так, что по художественному замыслу разработчика два элемента должны чуть-чуть отличаться по цвету, а в браузере у пользователя они окрашиваются одинаково;
- если требуется, чтобы цвет текста, фона или рамки совпал с цветом графического элемента (например, чтобы обеспечить «слияние» изображения-логотипа и находящегося под ним текстового блока с навигацией) — при использовании «небезопасных» цветов этого нельзя гарантировать. Автор неоднократно сталкивался с ситуациями, когда на компьютере разработчика фон графического и текстового элемента сливается, а на другом компьютере разница между цветами заметна невооруженным глазом.

Внимание!

Для текста, фона и рамок рекомендуется использовать цвета только из безопасной палитры:

```
00      66      cc
33      99      ff
```

В веб-палитру входит 216 цветов. Этого более чем достаточно: в самом деле, если на странице для текста, фона и рамок применяется больше 10 различных цветов — это чаще всего выглядит слишком пестро и некрасиво. Так что даже 10 цветов вам должно хватить, не говоря уж о 216.

Дополнительно

Некоторые начинающие разработчики хотят использовать тот или иной «небезопасный» цвет, объясняя это тем, что их художественный замысел

требует именно этого цвета. Такой подход уместен в печатных изданиях, но если вы разрабатываете сайт, вы должны понимать, что разные мониторы будут отображать совершенно разные оттенки, и большинство пользователей все равно не увидят «тот самый» цвет. Во-первых, у разных мониторов разная «цветовая температура», во-вторых, пользователи по-разному настраивают яркость и контрастность мониторов, в-третьих, видимый цвет в немалой степени зависит от освещения, падающего на экран. Разница между цветами, которые видят при этом разные пользователи, обычно значительно больше, чем между «тем самым» цветом и ближайшим к нему оттенком из безопасной палитры. Поэтому всегда лучше выбирать цвета из безопасной палитры.

Для основного текста на сайте нельзя придумать ничего лучше, чем черный текст (`#000000`) на белом фоне (`#ffffff`). Это наиболее привычная и удобочитаемая цветовая гамма. С помощью CSS вы можете раскрасить текст в какой угодно цвет, но удобочитаемость текста от этого пострадает, и глаза пользователя будут быстрее уставать.

Некоторые разработчики считают, что в выборе нестандартного цвета для текста проявляется их творческая концепция. На самом деле не должно быть таких «концепций», которые ухудшают удобство и доступность сайта для пользователей. Наоборот, разработчику следует применять свои творческие способности к тому, чтобы сделать сайт как можно более удобным.

Примечание

Рекомендация о применении оттенков из безопасной палитры распространяется только на текст, фон и рамки, цвета которых вы задаете с помощью CSS. Для изображений, внедренных в веб-страницу, эта рекомендация не актуальна. В них вы можете использовать любые цвета, которые позволяет формат изображения — за исключением случаев, когда требуется, чтобы цвет какой-либо части изображения совпадал с цветом текста, фона или рамки.

Дополнительно

В CSS существуют и другие способы указания цвета, среди которых:

- стандартное имя цвета, например: `color: green; background-color: yellow`. Этот способ устарел и не рекомендуется еще и потому, что большинство «стандартных» цветов не входят в безопасную палитру;
- сокращенный шестнадцатеричный код: например, вместо `#33cc99` можно написать просто `#3c9`, а вместо `#666666` — просто `#666`;
- десятичный код в модели RGB, например:

  ```
  color: rgb(255,0,0)
  ```

Здесь в каждой позиции может находиться целое число от 0 до 255.

Более подробно о различных способах указания цвета в CSS можно прочесть в официальной документации от Консорциума W3C. Впрочем, в большинстве случаев не требуется прибегать к каким-то иным способам задания цвета, кроме обычного шестнадцатеричного кода.

Гарнитура шрифта: *font-family*

Внимание!

`font-family` — гарнитура (вид) шрифта

Использование:

- `font-family:␣`*`шрифт`*`,␣`*`стандартное семейство`*`;`
- `font-family:␣`*`шрифт1`*`,␣`*`шрифт2`*`,␣...,␣`*`шрифтN`*`,␣`*`стандартное семейство`*`;`

Свойство `font-family` задает *гарнитуру* шрифта — то, что многие привыкли называть просто словом «шрифт».

Все гарнитуры делятся на два типа в соответствии с наличием или отсутствием засечек на концах штрихов:

С засечками	Без засечек (рубленые)
Times New Roman	Arial
Georgia	Verdana
Courier New	Comic Sans MS
Palatino Linotype	Lucida Console

Примечание

Когда лучше использовать шрифты с засечками, а когда — без засечек? Шрифты с засечками лучше всего подходят для основного текста страницы, шрифты без засечек — для заголовков и панелей ссылок. Засечки помогают направлять внимание читателя вдоль строки; шрифт без засечек обращает на себя больше внимания, но плохо подходит для долгого чтения. Впрочем, если шрифт имеет маленький размер (что допустимо в примечаниях и других небольших текстах вспомогательного характера) — то он также лучше воспринимается без засечек.

Шрифты также можно разбить на две группы по другому признаку: *пропорциональные* и *моноширинные*. Шрифт, которым набран этот текст, относится к пропорциональным. Такие шрифты наиболее распространены и характеризуются различной шириной символов. Вы видите, что буквы «А» и «Н» имеют примерно одинаковую ширину, буквы «М» и «Ш» — шире, а знаки препинания «!» и «.» — заметно у́же. А в моноширинном шрифте всякий символ имеет одну и ту же ширину:

Пропорциональные шрифты	Моноширинные шрифты
Times New Roman	Courier New
Arial	Lucida Console
Book Antiqua	Andale Mono
Century Gothic	

Примечание

Для оформления текста почти всегда следует использовать пропорциональные шрифты, так как они выглядят аккуратнее и читаются легче. Моноширинные шрифты допустимы только для создания каких-то особых эффектов оформления — например, в этой книге моноширинным шрифтом набраны фрагменты кода HTML и CSS.

Дополнительно

Моноширинные шрифты когда-то использовались в пишущих машинках и телетайпах, а теперь их используют в основном программисты и кодировщики для набора и редактирования в своем текстовом редакторе исходных текстов программ, файлов HTML и CSS и других подобных данных.

Одинаковая ширина всех символов обеспечивает наглядность кода — особенно это касается горизонтальных отступов.

Разнообразие операционных систем и браузеров приводит к тому, что вы не всегда можете задать в CSS какой-то определенный шрифт. Например, на компьютерах под управлением операционной системы Linux часто не бывает шрифта Arial, а на компьютерах под управлением Windows редко встречается шрифт Helvetica. Поэтому в CSS определено несколько так называемых *стандартных семейств шрифов*. Вместо того чтобы указывать конкретный шрифт, можно указать одно из стандартных семейств. Браузер сам подберет доступный шрифт из этого семейства.

Наиболее употребительны три стандартных семейства шрифтов:

- `serif` — пропорциональный шрифт с засечками. Шрифтом из этого семейства большинство графических браузеров по умолчанию отображают текст, для которого не задано никакой конкретной гарнитуры шрифта. В Windows самым распространенным шрифтом из этой группы является Times New Roman;
- `sans-serif` — пропорциональный шрифт без засечек. В Windows самым распространенным шрифтом из этой группы является Arial;

Дополнительно

Обозначения «serif» и «sans-serif» происходят из французской полиграфии: слово «serif» по-французски и на некоторых других языках обозначает «засечка», предлог «sans» — «без».

- `monospace` — моноширинный шрифт (обычно с засечками, но в общем случае зависит от системы и браузера). В Windows самым распространенным шрифтом из этой группы является Courier New.

Значение свойства `font-family` рекомендуется указывать в виде списка альтернативных шрифтов, разделенных запятой и пробелом, например, так:

```
h2 {
  font-family: Verdana, Arial, Helvetica, sans-serif;
}
```

Первым в этом списке идет та самая гарнитура, которой разработчик хочет оформить текст — в данном случае Verdana. Но может случиться так, что на компьютере пользователя не окажется шрифта Verdana. Тогда будет использован следующий шрифт в списке — Arial. Если у пользователя отсутствует и Arial (например, у него Linux) — текст будет отображаться шрифтом Helvetica. И, наконец, если нет ни одного шрифта, перечисленного в списке, — сработает стандартное семейство шрифтов. По крайней мере один шрифт из этого семейства на компьютере найдется всегда.

Примечание

Если название шрифта состоит из нескольких слов, разделенных пробелами, то в значении свойства `font-family` этот шрифт надо заключать в одинарные или двойные кавычки:

Правильно	Неправильно
`font-family: 'Courier New', monospace;`	`font-family: Courier New, monospace;`
`font-family: "Courier New", monospace;`	

Не следует злоупотреблять количеством разных гарнитур на одной веб-странице. Гарнитуру шрифта для основного текста вообще лучше специально не указывать: пусть текст отображается шрифтом, установленным в браузере по умолчанию — т. е. именно тем шрифтом, к которому привык пользователь. Для заголовков, панелей ссылок и других подобных элементов можно указать какой-нибудь шрифт без засечек. Обычно этого достаточно: если на стра-

нице использовать три и более различных гарнитур, это чаще всего портит ее внешний вид, нарушает единый стиль.

Примечание

Всегда указывайте стандартное семейство шрифтов в конце списка.

Рекомендуется	Не рекомендуется
`font-family: 'Courier New', monospace;`	`font-family: 'Courier New';`
`font-family: Tahoma, Arial, Helvetica, sans-serif;`	`font-family: Tahoma, Arial, Helvetica;`

Дополнительно

Многие шрифты относятся к так называемым *акцидентным* — они предназначены для заголовков и специальных выделений, но не годятся для набора сплошного текста. Если основной текст набран акцидентным шрифтом, скорость его чтения становится медленнее и глаза читателя быстрее устают. К акцидентным шрифтам относятся, например, Arial Black, Impact и Comic Sans (которым набран этот абзац).

Размер (кегль) шрифта: *font-size*

Внимание!

`font-size` — размер (кегль) шрифта

Использование:

`font-size: X`*em*`;`

(X — целое или дробное число)

Свойство `font-size` позволяет установить размер шрифта (в типографском деле размер шрифта называется *кеглем*). Значение этого свойства лучше всего указывать в условных единицах, называющихся `em`, например, так:

```
h3 {
  font-size: 1.2em;
}
```

Обычный размер шрифта равен 1em. Следовательно, если указать размер 2em, то шрифт станет в два раза крупнее, а если 0.5em — в два раза мельче (табл. 5.11).

Таблица 5.11. *Примеры использования свойства* `font-size`

Свойство *font-size*	**Примеры**
`font-size: 0.5em;`	Слово
`font-size: 0.75em;`	Слово
`font-size: 1em;`	Слово
`font-size: 1.2em;`	Слово
`font-size: 1.5em;`	Слово
`font-size: 2em;`	Слово
`font-size: 3em;`	Слово

Обратите внимание на синтаксис записи. Обозначение единицы измерения (em) пишется непосредственно после числа, оно не отделяется от числа ни пробелом, ни каким-либо другим символом. Число может быть как целым, так и дробным; целая часть отделяется от дробной с помощью точки (а не запятой, как наши десятичные дроби).

Примечание

Когда-то величина em означала ширину прописной буквы «M». Для управления размером шрифта на веб-страницах это значение em не актуально; достаточно просто помнить, что обычный шрифт имеет кегль 1em.

Не путайте обозначение единицы em в CSS и элемент `em` в HTML. Их одинаковое название — случайное совпадение, и никакого отношения друг к другу они не имеют.

Дополнительно

В англоязычной типографской практике часто встречаются термины, включающие в себя «em» или «en». Единица em соответствовала так называемой *круглой*, или *кегельной* — высоте площадки, на которой располагалась типографская литера. Единица en соответствовала *полукегельной* — т. е. половине em. Отсюда происходят em-dash (длинное тире, U+2014) и en-dash (короткое тире, U+2013), а также em-space (пробел величиной в кегельную, U+2003) и en-space (пробел в полукегельную, U+2002).

Заметим, что современные браузеры не очень хорошо поддерживают пробелы разной ширины. Большинство из них отображает em-пробел и en-пробел как обычный пробел. Также некоторые браузеры «не понимают» подстановок,

соответствующих этим пробелам, и отображают на их месте пустые квадратики или другие замещающие символы.

Существуют и другие единицы измерения, которые можно использовать в значении свойства `font-size`. Однако все они по разным причинам менее предпочтительны, чем `em`.

- **Фиксированные единицы**: `px` (пикселы), `pt` (типографские пункты), `cm` (сантиметры). Во многих браузерах (например, в Internet Explorer) указание размера шрифта в фиксированных единицах приводит к тому, что пользователь не может средствами браузера масштабировать шрифт, если он кажется ему слишком мелким или слишком крупным. Это вызывает неудобства при чтении. С другой стороны, некоторые другие браузеры (например, Mozilla) позволяют масштабировать шрифт в любом случае, так что указание конкретного размера в пикселах или пунктах становится просто бессмысленным.
- **Стандартные обозначения**: `large`, `medium`, `small`, `x-small` и др. Браузеры по-разному обрабатывают эти размеры: например, при указании любого из них шрифт в Mozilla будет отображаться мельче, чем в Internet Explorer. Такой способ указания кегля шрифта следует считать устаревшим.
- **Проценты**, например: `font-size: 110%`. Этот способ тоже менее универсален, чем указание кегля с помощью em. В некоторых случаях он, как и стандартные обозначения, может приводить к различному отображению текста.

Насыщенность шрифта: *font-weight*

Внимание!

`font-weight` — насыщенность шрифта

Использование:

- `font-weight: bold;`
- `font-weight: normal;`

Насыщенность шрифта — то же самое, что его «жирность». Чтобы сделать шрифт полужирным, нужно написать:

```
font-weight: bold;
```

Дополнительно

Иногда требуется, наоборот, сделать обычным (светлым) шрифт, который по умолчанию отображается полужирным — например, шрифт элемента `th` или `strong`. В этом случае нужно написать:

```
font-weight: normal;
```

Иногда удается регулировать насыщенность шрифта более детально. В качестве значения `font-weight` можно указать любое из чисел 100, 200, 300, 400, 500, 600, 700, 800, 900. Для того чтобы это сработало, нужно, чтобы и сам шрифт, и браузер поддерживали разные степени насыщенности. В большинстве случаев значения от 100 до 500 (в некоторых браузерах до 600) эквивалентны `normal`, а от 600 (в некоторых браузерах от 700) до 900 эквивалентны `bold`. Но иногда при значении 500 или 600 браузеры отображают насыщенность, «промежуточную» между `normal` и `bold`.

Примечание

Полужирный шрифт подходит для заголовков и выделения отдельных слов в тексте. Он совершенно не годится для сплошного текста.

Начертание шрифта: *font-style*

Внимание!

`font-style` — начертание шрифта

Использование:

- `font-style: italic;`
- `font-style: normal;`

Свойство `font-style` позволяет выделить текст курсивом. Для этого надо написать:

```
font-style: italic;
```

(слово «italic» во многих европейских языках обозначает «курсив»).

Дополнительно

Иногда требуется, наоборот, сделать обычным (прямым) шрифт, который по умолчанию отображается курсивом — например, шрифт элемента `caption` или `em`. В этом случае нужно написать:

```
font-style: normal;
```

Примечание

Курсив, как и полужирный шрифт, не годится для сплошного текста. Чаще всего он используется для выделения в тексте отдельных слов.

Не злоупотребляйте курсивом. Если на странице слишком много курсивного шрифта, она выглядит некрасиво. Особенно некрасиво выглядит ***полужирный курсив*** — такое оформление лучше вообще никогда не употреблять.

Межбуквенный интервал: *letter-spacing*

Внимание!

`letter-spacing` — межбуквенный интервал

Использование:

- `letter-spacing: Xem;`
- `letter-spacing: normal;`

Свойство `letter-spacing` позволяет регулировать межбуквенный интервал, т. е. «прижимать» или «раздвигать» соседние буквы.

Примечание

Разрядка. Увеличенный межбуквенный интервал называется *разрядкой*. Разрядка используется для выделения отдельных слов в тексте, а также для заголовков «в подбор» (как в начале этого примечания). Ни в коем случае не нужно применять разрядку для выделения сплошного текста — это смотрится очень некрасиво и плохо читается.

Значения `letter-spacing` лучше всего указывать, как и размер шрифта, в em. Отрицательные значения приводят к уплотнению набора, положительные — к разрядке. Рекомендуется указывать очень малые значения: в случае разрядки они не должны превышать ¼ кегля шрифта, иначе слово перестанет восприниматься как единое целое. (Например, если кегль равен 1em, текст можно разрежать до 0.2em; 0.25em — это уже предел.) В случае уплотнения ограничения еще серьезнее: в большинстве шрифтов уже при уплотнении до $^{1}{}_{10}$ кегля соседние буквы наезжают друг на друга (табл. 5.12).

Таблица 5.12. *Примеры использования свойства* `letter-spacing`

Свойство *letter-spacing*	**Примеры**
`letter-spacing: -0.1em;`	Слово
`letter-spacing: -0.05em;`	Слово
`letter-spacing: 0;`	Слово
`letter-spacing: 0.05em;`	Слово
`letter-spacing: 0.1em;`	Слово
`letter-spacing: 0.2em;`	Слово

Обычному межбуквенному интервалу соответствует значение `0` или `normal`.

Примечание

Указывая значение 0, единицу измерения лучше опускать:

Рекомендуется	Не рекомендуется
`letter-spacing: 0;`	`letter-spacing: 0em;` `letter-spacing: 0px;`

Дополнительно

В CSS также существует свойство `word-spacing`, с помощью которого можно увеличивать или уменьшать промежутки между словами. В зарубежной типографике стандартный междусловный пробел равен ⅓ em; это же значение по умолчанию сохраняется в браузерах. В отечественных правилах набора рекомендуется оптимальный междусловный пробел в ½ em.

Межстрочный интервал: *line-height*

Внимание!

`line-height` — межстрочный интервал

Использование:

- `line-height: X%;`
- `line-height: Xem;`

Межстрочный интервал — это расстояние между уровнями, на которых располагаются соседние строки. В большинстве случаев его не требуется менять. Но если все-таки возникает необходимость изменить его, следует воспользоваться свойством `line-height`, например:

```
p {
  line-height: 130%;
}
```

Величину `line-height` можно указывать в em или в процентах. По умолчанию она обычно равна 1.2em (то есть 120 %). Высота строки занимает 1em, и еще 0.2em отводится на промежуток между строками.

Примечание

Уменьшать межстрочный интервал не рекомендуется никогда. Иногда для достижения определенных эффектов оформления его можно увеличить, но не более, чем до 150 %.

Обобщенное свойство *font*

Внимание!

`font` — свойства шрифта

Использование:

```
font: font-style font-weight font-size/line-height font-family;
```

Для одного и того же элемента можно задать множество разных свойств шрифта, например:

```
h2 {
  font-style: normal;
  font-weight: bold;
  font-size: 1.25em;
  line-height: 130%;
  font-family: Verdana, Arial, Helvetica, sans-serif;
}
```

Но вместо этого все такие свойства можно объединить в одном свойстве `font`. Они записываются в определенной последовательности с соблюдением определенных синтаксических правил, например:

```
h2 {
  font: normal bold 1.25em/130% Verdana, Arial, Helvetica, sans-serif;
}
```

Обратите внимание: все значения разделяются пробелами, кроме пары значений `font-size` и `line-height`, между которыми ставится косая черта.

В свойстве `font` обязательно указывать только два значения: `font-size` и `font-family`. Остальные значения необязательны; например, можно записать:

```
  font: 1.25em Verdana, Arial, Helvetica, sans-serif;

  font: bold 1.25em/130% Verdana, Arial, Helvetica, sans-serif;
```

Однако все значения, которые присутствуют, должны идти друг за другом строго в указанном порядке: `font-style`, `font-weight`, `font-size`, `line-height`, `font-family`.

Примечание

Обратите внимание: свойство `letter-spacing` не входит в `font`, так что его требуется указывать отдельно.

Дополнительно

В CSS существуют и другие свойства шрифта, например:

- `font-variant` — позволяет сделать КАПИТЕЛЬ;
- `font-stretch` — позволяет сжать или растянуть каждый символ по ширине;
- `font-size-adjust` — регулирует соотношение между размерами прописных и строчных букв;
- `text-transform` — позволяет сделать все буквы строчными или прописными.

Эти свойства пока еще плохо поддерживаются браузерами, поэтому в настоящее время не рекомендуются к употреблению.

Свойства блоков текста

А вокруг — приметы нови,
Блочных зданий серый ряд.

А. Городницкий. Голицыно

Иногда мы открываем книгу и обнаруживаем огромные серые блоки из маленьких, плотно сбитых букв — и думаем: «Нет, это мне не прочесть».

Дж. Феличи. Типографика

В этом разделе рассказывается о свойствах CSS, которые могут быть применены к блокам текста, но не к отдельным символам. Например, можно выровнять блок текста по ширине или установить для него абзацный отступ, но нельзя применить эти свойства к отдельной букве.

Выравнивание (выключка): *text-align*

Горизонтальное выравнивание текста в абзаце или другом текстовом блоке называется *выключкой*. Различают выключку *в левый край* (выравнивание по левому краю), *в правый край* (выравнивание по правому краю), *в красную строку* (выравнивание по центру) и *по формату* (выравнивание по ширине). Например, основной текст этой книги выключен по формату.

Внимание!

`text-align` — выключка (выравнивание) текста в блоке

Использование:

`text-align: значение;`

(значение — одно из следующих: `left`, `right`, `center`, `justify`)

Для выключки текста в блоке применяется свойство `text-align`, которое может иметь одно из следующих значений:

- `left` — в левый край;
- `right` — в правый край;
- `center` — в красную строку (по центру);
- `justify` — по формату (по ширине).

Дополнительно

В издательском деле «выключка в красную строку» действительно означает выравнивание по центру. Более знакомое нам значение этого термина (строка с абзацным отступом), хотя и употребляется профессионалами, но гораздо реже.

Например, можно применить это свойство следующим образом:

```
h1 {
  text-align: center;
}
p {
  text-align: justify;
}
```

Примечание

По умолчанию большинство браузеров отображают текстовые блоки с выключкой в левый край, поэтому значение `text-align: left;` практически никогда не требуется указывать.

Значение `text-align: right;` следует использовать только в особенных случаях — например, для размещения подписи справа под текстом. Для сплошного текста выключка в правый край не годится ни в каком случае. (В самом деле — посмотрите, как некрасиво и неудобочитаемо выглядит этот абзац.)

В печатной продукции основной текст обычно выключается по формату, заголовки — по левому краю или в красную строку (по центру). Но прежде

чем переносить эту традицию на текстовое содержание веб-страниц, следует задуматься об отличиях Всемирной Паутины от традиционной полиграфии.

В то время как основное заглавие в книгах обычно помещается в центре обложки, заголовки веб-страниц и логотипы имеют тенденцию располагаться в левом верхнем углу. Пользователи привыкли к такому их расположению, и поэтому при открытии новой страницы внимание пользователя обычно падает в первую очередь на верхний левый угол.

Основная панель навигации сайта также нередко располагается у левого края страницы, либо в виде строки под главным заголовком — но, опять же, начинаясь от левого края. «Хлебные крошки», распространенные на сайтах, тоже обычно «прижимаются» влево.

Поэтому заголовки на веб-страницах рекомендуется также выравнивать по левому краю. Если заголовок размещен в центре страницы, многим пользователям приходится переключать внимание на этот центр. А при выключке по левому краю он оказывается как раз там, где уже сосредоточено внимание пользователя.

Примечание

Разумеется, эта рекомендация, как и остальные рекомендации в этой книге, не является непреложной истиной. Бывают ситуации, когда ее можно нарушать. Но учтите, что выключка заголовков не в левый край в большинстве случаев снижает удобство и эффективность страницы. Значит, размещать заголовок по центру или прижимать его к правому краю страницы можно только тогда, когда вы уверены, что в результате удобство и эффективность страницы только повысится.

Для Сети также плохо подходит выравнивание основного текста по ширине. Здесь отличия от печатных изданий таковы:

- в печатных изданиях набор «подгоняется» под заданную ширину полосы. А в Сети ширина, которую займет текстовый блок на экране, зависит от конечного пользователя и может варьироваться в самых широких пределах;
- в печатных изданиях контролируется ширина междусловных пробелов. По правилам отечественной типографики (*см. главу 2*) отимальная ширина пробела составляет ½ кегельной (т. е. 0.5em) и может изменяться в пределах от ¼ до ¾ кегельной. Кроме того, верстальщик следит за тем, чтобы междусловные пробелы в соседних строках не сильно отличались друг от друга. А при отображении текста на веб-странице браузер не будет отслеживать этих тонкостей, в результате чего многие строки получаются чрезмерно разреженными и образуется существенная разница в ширине пробелов в соседних строках. Из-за этого текст становится неудобочитаемым и некрасиво смотрится;

- аккуратный вид текста, выключенного по формату, в печатных изданиях в значительной степени обеспечивается переносами слов. Если не делать переносов — в тексте появятся огромные междусловные пробелы. Но в браузерах расстановка переносов не предусмотрена. Кодировщику остается только вручную расставить все возможные переносы на своей странице (с помощью подстановки ``), но и это не позволяет избежать всех проблем, которые решены в традиционной печати;

Дополнительно

Если вы хотите расставлять переносы слов вручную, учтите, что вам придется отслеживать и не допускать неблагозвучные переносы — например, «попадает» или «ссы-лок». Автоматические средства расстановки переносов не «отлавливают» таких курьезов.

- кроме того, текст выглядит некрасиво, если несколько строк подряд (обычно больше 3) оканчиваются переносом. Также некрасиво, если несколько строк подряд оканчиваются чем-то, кроме букв (переносами, знаками препинания, скобками). В печатных изданиях это легко предотвратить, но на веб-странице никак нельзя предсказать, как именно браузер пользователя разобьет ваш текст на строки.

Поэтому в подавляющем большинстве случаев основной текст на веб-странице рекомендуется выключать влево (то есть выравнивать по левому краю). Конечно, текст с неровным правым краем выглядит не так аккуратно, как в книгах. Но текст с огромными междусловными пробелами выглядит еще хуже и читается гораздо труднее.

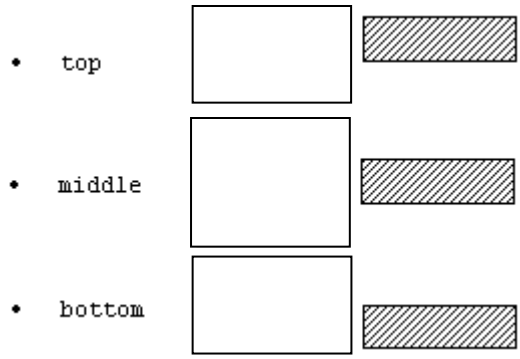
Примечание

Не следует выравнивать основной текст по центру. Неровный правый край — это не очень красиво, но вполне терпимо. А вот неровный левый край, который образуется при выключке в правый край или в красную строку, затрудняет чтение, так как читателю сложнее переходить со строки на строку.

Выравнивание по центру иногда подходит для заголовков, но никогда — для основного текста.

Дополнительно

Иногда требуется выровнять блок текста не только горизонтально, но и вертикально. Для этого используется свойство `vertical-align`, которое может иметь следующие значения:



Существуют другие допустимые значения `vertical-align`, но они пока еще плохо поддерживаются современными браузерами.

Абзацный отступ: *text-indent*

Внимание!

`text-indent` — величина абзацного отступа

Использование:

- `text-indent: X`em`;`
- `text-indent: X%;`

Абзацный отступ употребляется в крупных текстах — например, в длинных статьях или литературных произведениях. Отступ обычно неуместен в тексте объемом менее 3 абзацев, особенно если эти абзацы короткие.

Чтобы сделать абзацный отступ, применяется свойство `text-indent`, например:

```
p {
  text-indent: 1.5em;
}
```

Оптимальная величина абзацного отступа — от 1em до 2em. Для обычного текста его рекомендуется устанавливать равным 1.5em.

Дополнительно

Типографские правила велят делать во всем издании абзацные отступы одного и того же размера. Если в одном и том же тексте употребляются абзацы разного кегля (например, основной текст, размер шрифта в котором 1em, перемежается примечаниями с размером 0.8em), то в тексте меньшего кегля требуется указывать больший абзацный отступ:

Правильно	Неправильно
Вот абзац основного текста, кегль которого составляет 1em. А вот абзац вспомогательного текста, кегль которого составляет 0.8em.	Вот абзац основного текста, кегль которого составляет 1em. А вот абзац вспомогательного текста, кегль которого составляет 0.8em.

`p.major {`
`  font-size: 1em;`
`  text-indent: 1.5em;`
`}`
`p.minor {`
`  font-size: 0.8em;`
`  text-indent: 1.95em;`
`}`
	`p.major {`
	`  font-size: 1em;`
	`  text-indent: 1.5em;`
	`}`
	`p.minor {`
	`  font-size: 0.8em;`
	`  text-indent: 1.5em;`
	`}`

В зарубежных изданиях часто не используют абзацного отступа, зато отделяют абзацы друг от друга увеличенным вертикальным интервалом. Собственно, именно так браузеры по умолчанию отображают содержимое тэгов `<p>...</p>`.

Но применять одновременно такой «зарубежный» стиль и отечественный абзацный отступ не рекомендуется. Оформляйте абзацы либо с помощью отступа, либо с помощью увеличенного интервала, но не тем и другим способом одновременно.

Дополнительно

Иногда в зарубежной литературе применяется следующий стиль: все абзацы начинаются с отступа, но самый первый абзац в тексте — без отступа. Этот стиль применяется в некоторых системах компьютерной верстки (например, TEX), в результате чего сейчас и в нашей стране выпускаются издания, где первый абзац в каждой рубрике не имеет отступа.

Обработка пробельных символов: *white-space*

Внимание!

`white-space` — режим отображения пробельных символов

Использование:

- `white-space: pre;`
- `white-space: nowrap;`

Напомним, что к пробельным символам в XHTML относятся пробел, табуляция, возврат каретки (CR) и перевод строки (LF). Если в тексте какого-либо элемента присутствуют несколько любых пробельных символов подряд (одинаковых или разных), браузеры по умолчанию отображают всю эту последовательность как один пробел.

Если пробельные символы важны, то текст можно заключить в элемент `pre`, по умолчанию сохраняющий пробелы и переводы строк.

CSS также предоставляет средство для контроля над пробельными символами — свойство `white-space`. Оно может иметь следующие значения:

- `pre` — пробельные символы в тексте обрабатываются так же, как в элементе `pre`. Но, в отличие от элемента `pre`, текст при этом автоматически не выделяется моноширинным шрифтом. Например, если написать:

  ```
  p.poem {
    white-space: pre;
  }
  ```

  — то содержимое элемента `p` с классом `poem` отобразится с сохранением всех пробелов и переводов строк;
- `nowrap` — пробельные символы, как в обычном режиме, «сжимаются» в один пробел, но при этом внутри текстового блока не происходит переводов строк. Если текст слишком длинный, он уходит за границу экрана, но не разбивается на строки.

纸 Дополнительно

Один из характерных примеров использования значения `nowrap` — предотвращение переноса строки по дефису. Многие браузеры по умолчанию считают обычный дефис (-) символом, после которого можно разрывать строку. Чтобы предотвратить это, нужно заключить слово, содержащее дефис, в элемент `span`, для которого в CSS указано: `white-space: nowrap;`. Например, так:

в HTML:
```
<span class="nonbreaking">когда-нибудь</span>
```

в CSS:
```
span.nonbreaking {
  white-space: nowrap;
}
```

Не забывайте, что значение атрибута `class` (в данном случае `nonbreaking`) мы придумали «из головы». Вместо `nonbreaking` можно было написать `nowrap`, `onestring`, `vasya2004` или любое другое значение, допустимое с точки зрения синтаксиса CSS. Главное только, чтобы и в документе HTML, и в таблице стилей было записано одно и то же значение.

Можно немного сократить запись — например, сделать «неразрывными» все элементы `span`:

```
span {
  white-space: nowrap;
}
```

При этом слова с дефисами можно заключать просто в тэги `<span>...</span>` без атрибутов.

Рамки: *border* и другие свойства

С помощью CSS можно заключить блок текста в рамку произвольной толщины, стиля и цвета. Можно также поместить рамку не со всех четырех сторон от элемента, а с каких-то выбранных. Можно сделать разные рамки с разных сторон.

Чтобы задать стиль для рамки с какой-то одной стороны, нужно использовать свойства CSS:

- `border-top-style` — сверху;
- `border-bottom-style` — снизу;
- `border-left-style` — слева;
- `border-right-style` — справа.

Наиболее распространенные значения этих свойств таковы:

`none`	нет линии
`solid`	
`double`	
`dotted`	
`dashed`	

Например, чтобы рамки сверху и снизу от элемента `blockquote` стали двойными, надо написать:

```
blockquote {
  border-top-style: double;
  border-bottom-style: double;
}
```

Если требуется задать один и тот же стиль рамки со всех четырех сторон, можно использовать свойство `border-style`, например:

```
h2 {
  border-style: solid;
}
```

Чтобы задать толщину рамки, используются следующие свойства:

- `border-top-width` — сверху;
- `border-bottom-width` — снизу;
- `border-left-width` — слева;
- `border-right-width` — справа.

Ширину рамки рекомендуется задавать в пикселах (px), например:

```
h2 {
  border-bottom-width: 2px;
}
```

Если требуется задать одну и ту же ширину рамки со всех четырех сторон, можно использовать свойство `border-width`, например:

```
ul {
  border-width: 1px;
}
```

Примечание

Чтобы рамка с какой-то стороны блочного элемента действительно появилась, для нее надо задать и стиль, и толщину.

Обратите внимание, что минимально допустимая толщина рамки зависит от ее стиля. Например, для обычной сплошной рамки (`solid`) минимальная толщина составляет 1px. А для двойной рамки (`double`) требуется как минимум 3px: хотя бы по одному пикселу на каждую линию и хотя бы один пиксел между ними. Если задать для двойной рамки ширину 1px или 2px — появится обычная сплошная рамка.

Дополнительно

Браузеры по-разному поддерживают различные стили рамок. Многие браузеры (включая Internet Explorer 5) не поддерживают стилей `dotted` и `dashed`, более современные версии поддерживают, но отображают немного по-разному, не всегда красиво и аккуратно.

Прежде чем задавать какой-либо стиль рамки, отличный от `solid` и `double`, тщательно проверьте результат в различных браузерах.

Чтобы задать цвет рамки, используются следующие свойства:

- `border-top-color` — сверху;
- `border-bottom-color` — снизу;
- `border-left-color` — слева;
- `border-right-color` — справа.

Значениями этих свойств являются шестнадцатеричные коды цветов (такие же, как в свойствах `color` и `background-color`), например:

```
ul {
  border-left-color: #006633;
}
```

Если требуется задать один и тот же цвет рамки со всех четырех сторон, можно использовать свойство `border-color`, например:

```
h2 {
  border-color: #996633;
}
```

Наконец, можно одной строчкой задать сразу толщину, стиль и цвет рамки с какой-либо одной стороны. Для этого используются свойства:

- `border-top` — сверху;
- `border-bottom` — снизу;
- `border-left` — слева;
- `border-right` — справа.

Параметры рамки задаются в следующем формате:

```
свойство:␣толщина␣стиль␣цвет;
```

Обратите внимание, что параметры разделены пробелами, так же, как в свойстве `font`. Например:

```
table {
  border-right:␣2px␣solid␣#006699;
}
```

Кроме того, если со всех сторон от блока требуется задать совершенно одинаковые рамки, можно воспользоваться обобщенным свойством `border`, синтаксис которого такой же, как у предыдущих четырех свойств. Например:

```
blockquote {
  border: 1px solid #cc0099;
}
```

В результате этого каждая блочная цитата будет заключена в тонкую сплошную рамку вишневого цвета.

Поля и отступы: *margin* и *padding*

По умолчанию текст в блоке прижимается вплотную к рамке (если она есть). В свою очередь, сам текстовый блок (вместе с рамкой) вплотную прижимается к окружающим его блокам.

Расстояние от содержания текстового блока до рамки и от текстового блока до соседних блоков можно регулировать, изменяя свойства `margin` и `padding` для этого блока:

Можно установить разные размеры полей и отступов с разных сторон. Для *отступов* (расстояния от содержания блока до окружающей его рамки) применяются следующие свойства:

- `padding-top` — сверху;
- `padding-bottom` — снизу;
- `padding-left` — слева;
- `padding-right` — справа.

Например, чтобы между левой рамкой и текстом элемента `h2` появился отступ величиной 1.5em, можно записать:

```
h2 {
  padding-left: 1.5em;
}
```

Если требуется задать одну и ту же величину отступа со всех четырех сторон, можно использовать свойство `padding`, например:

```
td {
  padding: 0.75em;
}
```

Для *полей* (расстояние от рамки до соседних блоков) применяются следующие свойства:

- `margin-top` — сверху;
- `margin-bottom` — снизу;
- `margin-left` — слева;
- `margin-right` — справа.

Если требуется задать одну и ту же величину поля со всех четырех сторон, можно использовать свойство `margin`, например:

```
p {
  margin: 0;
}
```

Примечание

Горизонтальные поля и отступы (слева и справа) рекомендуется задавать в em или в процентах. Проценты при этом отсчитываются от ширины внешнего блока. Так, если элемент `h1` находится непосредственно внутри элемента `body`, то `margin-left: 5%;` для `h1` будет означать отступ величной 5 % от ширины всей страницы.

Вертикальные поля и отступы (сверху и снизу) рекомендуется задавать только в em.

В исключительных случаях можно указывать поля и отступы в пикселах (px).

Дополнительно

Поля и отступы могут быть отрицательными, например: `margin-left: -2.5em;`. Это позволяет совмещать блоки и накладывать их друг на друга. Такой возможностью, как и указанием полей и отступов в пикселах, следует пользоваться очень осторожно, убедившись, что это не вызывает проблем с просмотром страницы в разных браузерах под разными операционными системами, с разным разрешением экрана и разной шириной окна браузера.

Некоторые элементы HTML по умолчанию отображаются в браузерах с «готовыми» значениями `margin` и `padding`. Например, абзац обычного текста (элемент `p`) по умолчанию имеет поля сверху и снизу, блочная цитата (элемент `blockquote`) — поле слева. Чтобы устранить эти поля и отступы, в таблице стилей нужно указать нулевые значения, например:

```
p {
  margin-top: 0;
  margin-bottom: 0;
}
```

Дополнительно

Существуют также свойства `margin` и `padding`, позволяющие задать поля и отступы сразу со всех сторон. Если в качестве значения такого свойства указано одно число, эта величина применяется со всех сторон, например:

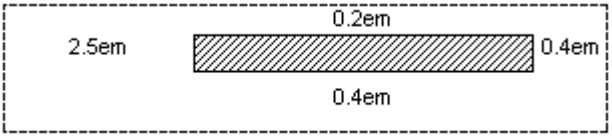
```
td {
  padding: 0.75em;
}
```

Если указаны два числа, разделенные пробелом, то первое из них применяется сверху и снизу, а второе — по бокам, например:

```
h1 {
  margin: 0 0.25em;
}
```

Если указаны три числа, разделенные пробелами, то первое из них применяется сверху, второе — по бокам, а третье — снизу. Наконец, если указаны четыре числа, то они применяются в следующем порядке: сверху, справа, снизу, слева (то есть по часовой стрелке), например:

```
blockquote {
  margin: 0.2em 0.5em 0.4em 2.5em;
}
```



Ширина блока: *width*

Внимание!

`width` — ширина блока

Использование:

- `width: X%;`

По умолчанию любой блочный элемент занимает всю доступную ему ширину. Если бы не было средств преодолеть это свойство, то было бы невозможно расположить два или несколько блоков рядом друг с другом.

В этой книге не рассматриваются вопросы верстки страницы. Чтобы узнать, как разместить блоки рядом и добиться других эффектов относительного расположения, обратитесь к официальной документации Консорциума W3C (**www.w3.org/TR/CSS2**) и к книгам по CSS (например, [19], [20] и [21]).

Тем не менее, автор считает полезным упомянуть об употреблении свойства `width`. Это свойство позволяет задавать конкретную ширину блока, например:

```
blockquote {
  width: 50%;
}
```

Напомним, что проценты, как и в случае `margin` и `padding`, отсчитываются от ширины внешнего блока.

Примечание

В большинстве случаев не рекомендуется указывать конкретную ширину блока. Всегда следует организовывать страницу таким образом, чтобы размеры всех блоков определялись их содержимым.

Рассмотрим связанные с этим проблемы на примере двух блоков, расположенных рядом.

Здесь указана конкретная ширина обоих блоков: по 50 %

— Ну, — замялась Алиса, — если вы не возражаете, я хотела бы чуточку подрасти! Я ведь сейчас с палец ростом. Подумайте, это прямо стыдно быть такого роста!	— Ну, — замялась Алиса, — если вы не возражаете, я хотела бы чуточку подрасти! Я ведь сейчас с палец ростом. Подумайте, это прямо стыдно быть такого роста!

Здесь конкретная ширина блоков не указана; она рассчитывается, исходя из их содержимого

— Ну ,— замялась Алиса, — если вы не возражаете, я хотела бы чуточку подрасти! Я ведь сейчас с палец ростом. Подумайте, это прямо стыдно быть такого роста!	— Ну ,— замялась Алиса, — если вы не возражаете, я хотела бы чуточку подрасти! Я ведь сейчас с палец ростом. Подумайте, это прямо стыдно быть такого роста!

Как видно, если в обоих блоках одинаковое количество содержимого, то и в случае с заданной шириной, и в случае без заданной ширины блоки выглядят одинаково. Теперь добавим содержания в один из блоков:

— Ну, — замялась Алиса — если вы не возражаете, я хотела бы чуточку подрасти! Я ведь сейчас с палец ростом. Подумайте, это прямо стыдно быть такого роста! — Таким ростом можно только гордиться! — сердито закричал Червяк, вытягиваясь во весь рост. Он был как раз длиной в палец.	— Ну, — замялась Алиса, — если вы не возражаете, я хотела бы чуточку подрасти! Я ведь сейчас с палец ростом. Подумайте, это прямо стыдно быть такого роста!

Здесь ширина блоков осталась по 50 %, в результате чего в правом блоке образовалось большое пустое место.

— Ну, — замялась Алиса, — если вы не возражаете, я хотела бы чуточку подрасти! Я ведь сейчас с палец ростом. Подумайте, это прямо стыдно быть такого роста! — Таким ростом можно только гордиться! — сердито закричал Червяк, вытягиваясь во весь рост. Он был как раз длиной в палец.	— Ну, — замялась Алиса, — если вы не возражаете, я хотела бы чуточку подрасти! Я ведь сейчас с палец ростом. Подумайте, это прямо стыдно быть такого роста!

Здесь ширина левого блока автоматически стала немного больше, а правого — немного меньше. В результате этого ни в одном блоке не возникает ненужной пустоты, и высота обоих блоков одинакова.

Верстка, при которой ширина блоков не задается изначально, а «подгоняется» браузером под размер окна и содержание блоков, называется *«резиновой»* версткой. К сожалению, многие разработчики не пользуются ей или пользуются очень ограниченно, в результате чего страницы хорошо просматриваются только в «стандартных» условиях (обычный графический браузер на персональном компьютере, среднее разрешение экрана, окно браузера развернуто на весь

экран). Стоит только пользователю изменить ширину окна, масштабировать шрифт или просмотреть страницу с помощью наладонного компьютера — верстка «плывет» и искажается, появляется горизонтальная полоса прокрутки и другие неприятности, затрудняющие просмотр страницы.

Качественная «резиновая» верстка — это настоящее искусство. Грамотное владение такой версткой свидетельствует о высокой квалификации кодировщика и обеспечивает ему признание и авторитет в профессиональной среде, а пользователям — наиболее удобные и доступные страницы.

Дополнительно

В CSS также есть свойство `height` (высота блока), но употреблять его обычно не следует. Очевидно, если для блока задано свойство `width`, то высота этого блока однозначно определяется его содержанием. Если задать высоту блока «насильственно», она будет конфликтовать с высотой, рассчитанной по содержанию.

Один из редчайших случаев, когда высоту требуется указывать, — это ситуация, когда в блочный элемент заключено растровое изображение или другой объект, имеющий собственные жестко определенные размеры. Например, пусть изображение вложено в элемент `div`:

```
<div id="tolstoy"><img src="lev.jpg" width="250"
height="374" alt="Лев Толстой. Фото 1907 г." /></div>
```

В этом случае в CSS можно записать:

```
div#tolstoy {
  width: 250px;
  height: 374px;
}
```

Оформление таблиц

CSS предоставляет широкий набор средств для оформления таблиц. Во-первых, к тексту, находящемуся в таблице, можно применять все обычные свойства CSS. Кроме того, существуют свойства, управляющие рамками таблицы, и главное из них — `border-collapse`.

Внимание!

`border-collapse` — режим отображения рамок вокруг ячеек таблицы

Использование:

- `border-collapse: collapse;` — рамки соседних ячеек сливаются;
- `border-collapse: separate;` — рамки соседних ячеек разделены.

Чтобы задать рамки вокруг ячеек таблицы, можно воспользоваться свойством `border`, которое было описано ранее. Например, так:

```
th, td {
  border: 1px solid black;
}
```

Однако при этом таблица отобразится с «разделенными» ячейками (рис. 5.14, *а*), так как свойство `border-collapse` по умолчанию имеет значение `separate`. Чтобы рамки соседних ячеек сливались, для всей таблицы нужно задать значение `collapse` (рис. 5.14, *б*):

```
table {
  border-collapse: collapse;
}
```

Браузер	Количество посещений	В % от общего количества посещений
Microsoft Internet Explorer	3786	84,93
Opera	303	6,80
Mozilla	251	5,63
Другие	118	2,64
Итого	4458	100,00

а

Браузер	Количество посещений	В % от общего количества посещений
Microsoft Internet Explorer	3786	84,93
Opera	303	6,80
Mozilla	251	5,63
Другие	118	2,64
Итого	4458	100,00

б

Рис. 5.14. Значения `border-collapse`: *а* — `separate` (по умолчанию), *б* — `collapse`

Примечание

Параметры рамок таблицы, как и все оформление, рекомендуется задавать именно в таблице стилей, а не в коде HTML. К сожалению, некоторые разработчики до сих пор пользуются устаревшими атрибутами HTML — такими, как `border`, `cellpadding`, `cellspacing` и т. д., например:

```
<table border="1" cellpadding="5" cellspacing="0"
width="100%" align="center">
```

С помощью CSS возможна гораздо более гибкая настройка внешнего вида как целой таблицы, так и отдельных ее строк, ячеек, групп строк и групп столбцов. Поэтому нет никаких оснований пользоваться табличными атрибутами HTML.

При оформлении таблиц следует учитывать ряд типографских правил:

- таблицы обычно набираются шрифтом меньшего кегля, чем основной текст. Рекомендуемый размер шрифта — от 0.8em до 0.9em;
- заголовки граф таблицы обычно набираются полужирным шрифтом и выравниваются по центру;
- если в ячейке таблицы находится текст, точка после него не ставится (так же, как в заголовках и других подобных надписях);
- текст, занимающий больше одной строки, обычно выключается влево. Числа и текст, умещающиеся в одну строку, выравниваются по центру. Оформление всех ячеек в одной графе должно быть одинаковым, так что если в одних ячейках длинный текст, а в других — короткий, то все ячейки придется выключать влево.

Примечание

Про составление и типографское оформление таблиц можно более подробно прочитать в [6], а про «табличные» свойства CSS — в официальной документации: **www.w3.org/TR/CSS2/tables.html**.

Чтобы создать отступ от рамок в ячейке таблицы, нужно использовать свойства `padding`, а не `margin`:

текст	текст

отступы по умолчанию	`td {`
	`  padding: 0.5em 1em;`
	`}`

Оформление перечней

Внимание!

`list-style-type` — тип маркера перечня

Использование:

- `list-style-type: тип;`

`list-style-position` — положение маркера перечня

Использование:

- `list-style-position: outside;` — снаружи
- `list-style-position: inside;` — внутри

CSS позволяет управлять внешним видом маркеров перечня. Для этого существует свойство `list-style-type`, которое применяется к элементам `li` и может принимать следующие значения:

disc	**circle**	**square**	**decimal**	**lower-roman**
• Красный	○ Красный	■ Красный	1. Красный	i. Красный
• Жёлтый	○ Жёлтый	■ Жёлтый	2. Жёлтый	ii. Жёлтый
• Зелёный	○ Зелёный	■ Зелёный	3. Зелёный	iii. Зелёный

upper-roman	**lower-latin**	**upper-latin**	**none**
I. Красный	a. Красный	A. Красный	Красный
II. Жёлтый	b. Жёлтый	B. Жёлтый	Жёлтый
III. Зелёный	c. Зелёный	C. Зелёный	Зелёный

Например, с помощью этого свойства можно оформить двухуровневый перечень из *разд. 5.1.4* «Перечни: *ul*, *ol*, *dl*» таким образом, чтобы внешний перечень был помечен незаполненными кружками, а внутренний — квадратиками (так же, как в этой книге):

```
ul li {
  list-style-type: circle;
}
ul li ul li {
  list-style-type: square;
}
```

(Обратите внимание на применение контекстного селектора в этом примере.)

Расположение маркера относительно перечня можно регулировать с помощью свойства `list-style-position`, которое также задается для элемента `li` и может иметь следующие значения:

`outside` (по умолчанию)

- При этом значении свойства `list-style-position` маркер расположен снаружи от перечня (как в этом пункте)

`inside`

- При этом значении свойства `list-style-position` маркер расположен в первой строке перечня (как в этом пункте)

Примечание

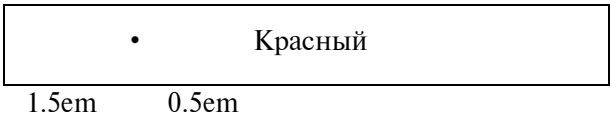
К сожалению, CSS не позволяет маркировать перечни буквами русского алфавита. Также современные браузеры еще не поддерживают свойства CSS, позволяющие организовать более гибкую нумерацию — например, начинать нумерацию пунктов перечня с произвольного числа или создавать вложенные нумерованные перечни с многоуровневой нумерацией (т. е. «1», «1.1», «1.1.1» и т. д.).

Чтобы регулировать отступ от левого края блока до маркера перечня и от маркера до текста, можно использовать свойства `margin-left` и `padding-left` для отдельных пунктов перечня (`li`):

- `margin-left` — отступ от левого края блока до маркера;
- `padding-left` — отступ от маркера до текста.

Например:

```
li {
  margin-left: 1.5em;
  padding-left: 0.5em;
}
```



Дополнительно

По умолчанию сами элементы `ul` и `ol` имеют отступ от левого края блока. Если вы хотите «прижать» перечень вплотную к левому краю, то нужно указать для него `margin-left: 0;` и `padding-left: 0;`.

На этом заканчивается обзор основных свойств CSS для оформления текстового содержания. Однако это не означает, что язык CSS ограничивается только перечисленными свойствами. Для того чтобы получить полное представление о таблицах стилей, автор рекомендует вам ознакомиться с официальной документацией CSS 2 (**www.w3.org/TR/CSS2**), а также с книгами, посвященными CSS (*см. список литературы в конце этой книги*).

```
<?xml version="1.0"
<!DOCTYPE html PUBLI
  "-//W3C//DTD XHTM
  "http://www.w3.or
<html xmlns="http:/
  lang="ru" xml:lan
<!-- Редактировал:
<head>
  <title>Евгений Он
  <link rel="styles
    href="index.css
</head>
```

Часть III

Все для человека

Как добиться, чтобы текстовое содержание сайта соответствовало интересам и задачам пользователей

— Ну, как? — поинтересовался Волков.
— В кайф.
— Вот видишь. А вы — менты, козлы, сволочи, уроды. Да наш главный принцип — все человеку, все для человека.

А. Кивинов. Улицы разбитых фонарей. Петербургский презент

```
<?xml version= 1.0
<!DOCTYPE html PUBL
  "-//W3C//DTD XHTM
  "http://www.w3.or
<html xmlns="http:/
  lang="ru" xml:lan
<!-- Редактировал:
<head>
  <title>Евгений Он
  <link rel="styles
    href="index.css
</head>
```

Глава 6

Ради красного словца…

Как составлять текст для размещения на сайте

Языки были изобретены людьми и для блага людей. В анналах компьютерных наук этот факт случайно оказался позабытым.

Л. Уолл, Т. Кристиансен, Дж. Орвант. Программирование на Perl

Язык мой — враг мой; все ему доступно,
Он обо всем болтать себе привык.
Фригийский раб, на рынке взяв язык,
Сварил его (у господина Копа
Коптят его). Эзоп его потом
Принес на стол...

А. С. Пушкин. Домик в Коломне

Умение составлять качественный текст не приходит само собой. Это настоящее искусство, которое требует и определенных талантов, и специального обучения, и немалого опыта. И в устной речи, и в печатной продукции, и на сайтах требуется такой текст, который наилучшим образом донесет до слушателя или читателя то, что вы хотите ему сказать.

Все знают, как высоко ценится умение составителей рекламных слоганов. Из двух-трех слов они составляют фразу, которая мгновенно запечатляется в памяти, которую вспоминают и цитируют на каждом шагу. Такие фразы входят в поговорки и анекдоты и остаются в народной памяти наравне с классическими цитатами из Пушкина или Грибоедова.

Такое же мастерство требуется от составителей текста для сайта. Просто высказать свою мысль по-русски сумеет любой грамотный человек, но этого мало. Чтобы эта мысль действительно «зацепила» пользователя, ее надо выразить, сформулировать оптимальным образом; нужно подобрать наиболее подходящие слова и расположить их в правильном порядке. Нужно выбрать определенный стиль, интонацию, эмоцию. Только в этом случае ваш текст действительно «сработает».

Многие разработчики и держатели крупных, профессиональных сайтов осознали это и привлекают к работе профессиональных технических писателей, редакторов и корректоров. Но, к сожалению, немало и таких сайтов, текст для которых пишут те же люди, которые кодируют HTML и CSS или разрабатывают серверные программы. Эти сайты иногда полны элементарных грамматических ошибок, а о качестве текста с точки зрения профессионального редактора вообще говорить не приходится. А ведь неудачный текст снижает эффективность и привлекательность сайта, уменьшает доверие пользователей и заставляет их переключаться на сайты конкурентов.

При составлении текста для сайтов также надо учитывать, что многие пользователи будут искать ваш сайт с помощью поисковых систем. Значит, текст должен быть таким, чтобы автоматические поисковые системы правильно восприняли его и находили сайт тогда и только тогда, когда пользователь ищет действительно его.

Эта глава не сделает вас сразу профессиональными техническими писателями и редакторами. Но, по крайней мере, из нее вы узнаете, как избежать наиболее распространенных ошибок, связанных с текстом, а также «помочь» поисковым системам обнаруживать ваш сайт.

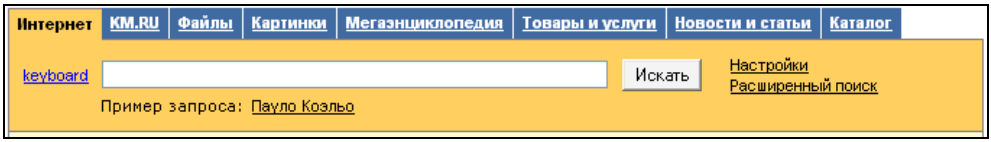
6.1. Доброе слово и кошке приятно
Как составить текст, производящий на пользователя нужное впечатление

> Стихи надо писать так, что если бросить стихотворение в окно, то стекло разобьется.
>
> *Д. Хармс*

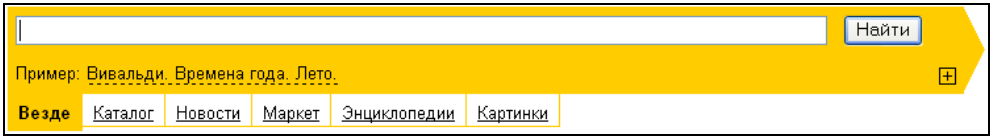
6.1.1. Совершенство малых форм
Самые короткие надписи

> И хочется петь и плясать,
> И справа налево писать,
> А после в затылке чесать,
> Не в силах исправить.
>
> *Р. Чебыкин. Левая песня*

Начнем разговор с самых коротких надписей, которые встречаются на веб-страницах. Это — надписи на кнопках для отправки формы (рис. 6.1).



а



б

Рис. 6.1. «Искать» (**www.km.ru**) или «Найти» (**www.yandex.ru**) — что лучше?

Примечание

Обратите внимание, что в этой книге не рассказывается о том, как кодировать формы на языке HTML или как писать серверные программы для обработки этих форм. Если вы хотите подробнее узнать о формах в HTML — обратитесь к официальной документации: **www.w3.org/TR/html4**. Также существует немало хорошей литературы о языках и технологиях серверного программирования — Perl, PHP, JSP и др.

По умолчанию браузер отображает на кнопке надпись вроде **Подача запроса**. Это, конечно, никуда не годится. Надпись на кнопке должна информировать пользователя о том, что именно делает эта кнопка, и чем точнее, тем лучше. Вместе с тем, надпись должна быть как можно короче. Если в заглавии страницы допустимо не более нескольких десятков символов, то здесь счет идет буквально на каждую букву.

Рассмотрим в качестве примера простую форму традиционной поисковой системы (такой, как Яндекс, Google или поисковой системы, встроенной в крупный сайт). Что писать на кнопке, расположенной рядом с текстовым полем?

Иногда встречается надпись **OK**. Это совершенно недопустимо. Дело тут даже не в борьбе за чистоту русского языка. (В конце концов, «OK» благополучно употребляют и немцы, и французы, и египтяне; возможно, рано или поздно это выражение будет заимствовано и русским языком, как «компьютер» или «принтер».) Дело в том, что «OK» не несет абсолютно никакой информации о назначении кнопки.

Более удачный вариант — надпись **Поиск**. Но и у нее есть свои недостатки. Обычно такая надпись ассоциируется не с кнопкой отправки формы, а со ссылкой на страницу, содержащую форму для поиска.

Для кнопки гораздо больше подходят слова, обозначающие конкретное действие, которое пользователь ожидает от нажатия на эту кнопку. Пользователь ожидает, что после нажатия на кнопку система будет искать то, что он запросил — значит, на кнопке можно так и написать: **Искать**.

Но еще лучше будет вариант **Найти** — ведь пользователь хочет от системы не только того, чтобы она что-то искала, но и того, чтобы она в конце концов нашла и предъявила ему результаты.

Наконец, надпись можно усилить восклицательным знаком: **Найти!** Это создаст у пользователя ощущение, что он отдает команду поисковой системе. Он будет чувствовать себя не униженным просителем (как в случае надписи **Подача запроса**), а командиром, хозяином положения. Всегда важно вызывать у пользователя сознание того, что сайт полностью в его распоряжении, а не наоборот.

	Подача запроса	Очень плохо
	OK	Очень плохо
	Поиск	Плохо
	Искать	Лучше
	Найти	Хорошо
	Найти!	Очень хорошо

Дополнительно

Где размещать кнопку для отправки простой формы поиска? Наилучший вариант — справа от строки, в которую пользователь ввел запрос. При этом пользователю не приходится «перескакивать» взглядом в другое место. В *главе 5* уже отмечалось, что наиболее удачно такое расположение элементов, которое отвечает естественному направлению внимания пользователя, а не заставляет его специально перемещать взгляд.

Конечно, если ваш сайт написан на языке, где читают справа налево, — кнопку, наоборот, следует помещать слева от строки ввода.

Другой пример — кнопка, предназначенная для отправки сообщения на форум. В этом случае пользователь обычно заполняет несколько полей: имя, пароль, тема сообщения, текст сообщения и, возможно, некоторые другие. Что писать на кнопке в этом случае? Текст **Отправить** — неплохой, но он не вполне определенный. Лучше **Отправить сообщение**. Так как сама форма в этом случае гораздо объемнее, чем простая строка поиска, то можно немного пожертвовать краткостью в пользу большей конкретности.

Примечание

Не пишите на кнопках «Послать», особенно если вы рассчитываете на молодежную аудиторию. Вообще, будьте осторожны с молодежью: если вы будете употреблять в тексте слова вроде «иметь» или «кончить» — это вызовет бурные эмоции, но совсем не те, на которые вы рассчитываете.

Дополнительно

Многие системы, в которых используются формы (например, форумы) заимствуются отечественными разработчиками с англоязычных сайтов. При этом надписи на кнопках не всегда переводятся; иногда так и остается — **Search**, **Send**, **Go!** и т. д. Не нужно следовать этим примерам. Если вы разрабатываете сайт для русскоязычных пользователей — переводите надписи на русский язык.

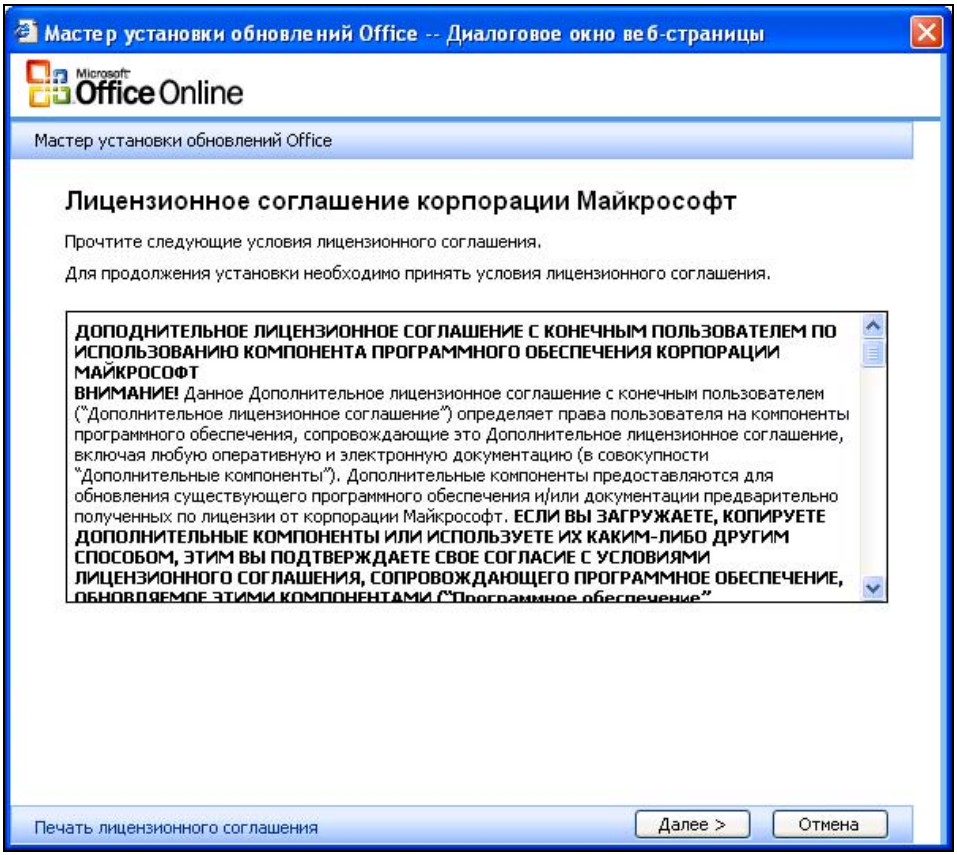


Рис. 6.2. Соглашение на сайте обновления Microsoft Office

Еще один пример — кнопки или ссылки под лицензионным соглашением или другим предупреждением (рис. 6.2). Пользователю требуется прочитать условия и согласиться с ними, чтобы получить доступ к нужному ресурсу.

В этом случае лучше всего подходит надпись **Я согласен**. (Если используется также кнопка с противоположным выбором, на ней должна быть надпись **Я не согласен**.) Надписи в повелительном наклонении (**Принять**, **Согласиться** и т. д.) не подходят: создается впечатление, что это не пользователь, а система должна с чем-то согласиться. В самом деле, ведь надпись **Найти!** означает, что искать будет именно поисковая система, а не пользователь. Аналогично, надпись **Принять** также подразумевает обращение пользователя к системе, а не наоборот.

Иногда требуется, чтобы пользователь ввел информацию о себе — например, при регистрации на форуме или для доступа к особым ресурсам. Для кнопки, сопровождающей форму с такими сведениями, в большинстве случаев подходит надпись **Зарегистрировать** или **Зарегистрировать меня** (но не **Зарегистрироваться**, **Сохранить** или тем более **Отправить**).

Дополнительно

Если пользователь может впоследствии внести изменения в ту информацию, которую он ввел, он должен видеть другую надпись на кнопке — **Сохранить изменения**. Даже если для первоначальной регистрации и последующих изменений используется форма одного и того же вида, надпись на кнопке должна отличаться.

Вы увидели всего несколько примеров того, из чего можно исходить при разработке надписей на кнопках. Общего рецепта «идеальной надписи» не существует: в каждом конкретном случае приходится учитывать смысл и назначение каждой конкретной кнопки. Для формулировки надписи нужно также представлять себе психологию пользователей, их цели и задачи, с которыми они приходят на сайт. Кроме того, необходимо хорошее знание русского языка. В самом деле, прежде чем выбирать нужное слово из нескольких подходящих, нужно взять откуда-то эти подходящие слова. А взять их можно только из собственной головы. И, наконец, требуется творческая фантазия и вдохновение — как, впрочем, и в любом другом занятии.

Все то, что сейчас было сказано про кнопки, актуально и для любого другого текста на сайте. И чем объемнее текст — тем больше требуется знаний, мастерства и таланта.

6.1.2. Туда-сюда
Особенности составления текста для навигации

И, наконец, он завопил отчаянным голосом:
— Ай-ай-ай, спасите-помогите! Не могу ни взад, ни вперед!

А. Милн. Винни-Пух и все-все-все

Однажды автору потребовалось разработать схему навигации для материала, состоящего из нескольких последовательных страниц. На каждой странице требовалось поместить ссылку на содержание, а также ссылки на предыдущую и следующую страницы.

Похожая навигация применяется в рекомендациях Консорциума W3C (см. рис. 6.3), но она обладает рядом недостатков. Главный из них — в том, что непонятно, куда именно ведут ссылки **Previous** (Предыдущая страница) и **Next** (Следующая страница). Пользователь должен знать, что именно находится на этих страницах.

Таким образом, участок навигации должен состоять из следующих ссылок:

- ссылка на страницу с содержанием;
- ссылка на предыдущую страницу (с заглавием этой страницы);
- ссылка на следующую страницу (с заглавием этой страницы).

Как разместить эти ссылки относительно друг друга? Первый вариант, который пришел в голову, выглядел примерно так:

Назад [Трезвучие с повышенной терцией] | Содержание |
Вперед [Особенности нотной записи]

Но слова «Назад» и «Вперед» были тут же забракованы. Ведь так называются стандартные кнопки во многих браузерах. Для пользователя слова «Вперед» и «Назад» означают страницы, которые записаны в журнале его браузера перед и после текущей. Если использовать эти слова в каком-нибудь другом смысле, пользователь может дезориентироваться. Например, он пришел на текущую страницу с какого-то другого сайта. Он хочет вернуться обратно, щелкает по ссылке **Назад** — и попадает вовсе не на тот сайт, где только что был, а на страницу «Трезвучие с повышенной терцией».

Примечание

Обратите внимание, что ссылка на предыдущую страницу расположена слева, а на следующую — справа. В самом деле, те, кто привык читать слева направо, обычно воспринимают движение слева направо как «Вперед», а справа налево — как «Назад». В браузерах тоже кнопка **Назад** находится левее, а **Вперед** — правее.

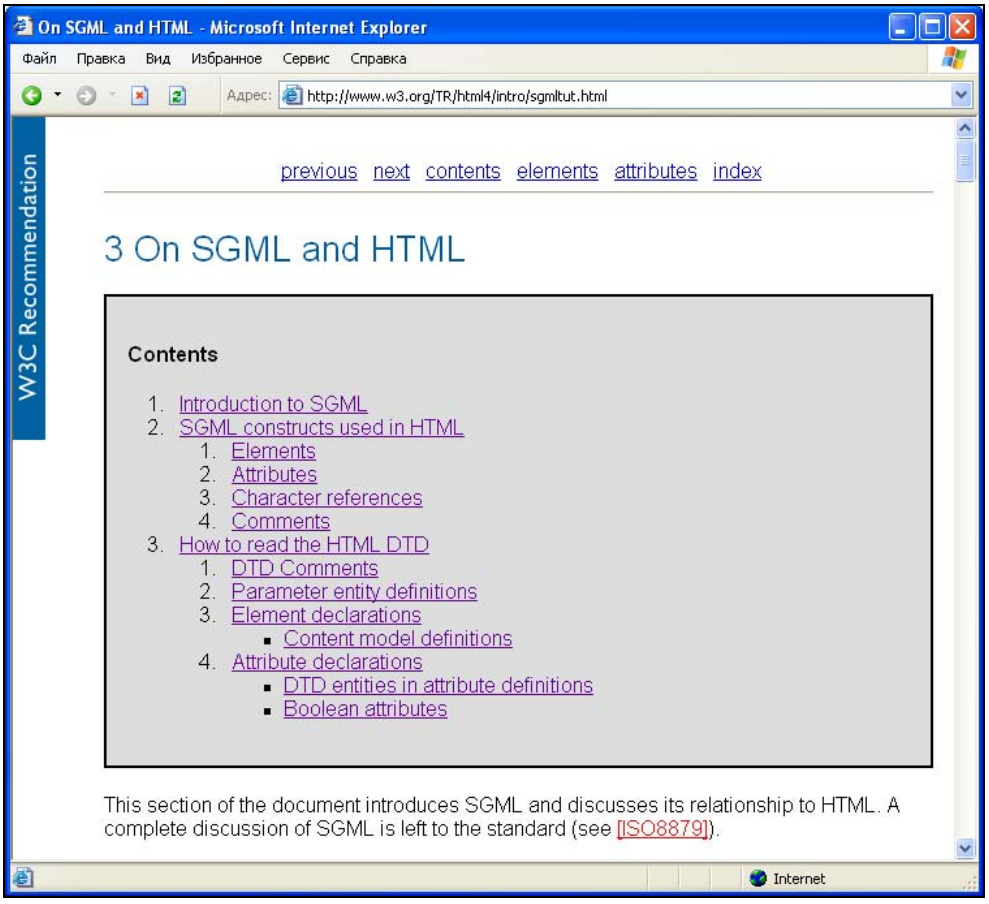


Рис. 6.3. Навигация по документации Консорциума W3C

纸 *Дополнительно*

Нередко на сайтах начинающих разработчиков попадаются ссылки вроде «Вернуться на главную страницу». Но если пользователь до этого не был на этой главной странице, если он пришел откуда-то еще — как же он может «вернуться»?

Было предложено заменить «Назад» и «Вперед» на «Предыдущая» и «Следующая». Но это было неприемлемо по соображениям грамотности русского языка: «Предыдущая...» — что? Значит, придется писать «Предыдущая страница» и «Следующая страница»:

Предыдущая страница [Трезвучие с повышенной терцией] | Содержание | Следующая страница [Особенности нотной записи]

При таком варианте строка получалась слишком длинной и не умещалась в окне браузера. Кроме того, она воспринималась скорее как ряд из пяти, а не из трех разных ссылок.

Тогда слова «Предыдущая страница» и «Следующая страница» перекочевали наверх:

Предыдущая страница: Трезвучие с повышенной терцией	Содержание	Следующая страница: Особенности нотной записи

Примечание

Рассматривались еще варианты «На предыдущую страницу» и «На следующую страницу», но они были признаны избыточными. Кроме того, слова в них стояли в дательном падеже, при том, что именительный воспринимается естественней.

Но две строки слева и справа все равно были похожи на две разные ссылки. Поэтому ссылками были сделаны только заглавия страниц, а надписи над ними уменьшились в размере:

Предыдущая страница:		Следующая страница:
Трезвучие с повышенной терцией	Содержание	Особенности нотной записи

На этом варианте было решено остановиться. Впрочем, автор не уверен, что это идеальное решение. Возможно, вам удастся придумать еще более удачную навигацию.

Этот пример приведен здесь для того, чтобы вы увидели, что даже составление простого фрагмента навигации требует размышлений и просмотра нескольких черновых вариантов. Не останавливайтесь на том, что первым пришло в голову. Пробуйте, экспериментируйте, сравнивайте, проверяйте разные решения в действии — только в этом случае вы достигнете успеха.

Еще одна интересная задача — поиск разделителей для ссылок. Если панель навигации горизонтальна, между ссылками должен находиться какой-нибудь разделитель (в вышеприведенном примере это была вертикальная черта). Но можно найти и другие символы, подходящие в этом качестве.

Примечание

В качестве разделителей ссылок крайне не рекомендуется применять изображения. Если у пользователя другое разрешение экрана или другой размер шрифта — панель навигации будет выглядеть некрасиво, т. к. разделители станут неестественно большими или неестественно маленькими по сравнению с текстом ссылок. Если же изображения вовсе отключены (напомним, так делают многие пользователи для экономии трафика) — то между ссылками будут отображаться уродливые квадратики.

纸 Дополнительно

Заметим, что в Unicode определено много вертикальных черточек (табл. 6.1).

Таблица 6.1. *Вертикальные черточки в Unicode*

Символ	Код	Название	Значение
	U+007C	VERTICAL LINE	Вертикальная черта ASCII; в Unicode также обозначает абсолютную величину (модуль числа)
	U+01C0	LATIN LETTER DENTAL CLICK	Символ для обозначения щелкающего зубного звука (у некоторых народов Африки и Кавказа)
	U+05C0	HEBREW PUNCTUATION PASEQ	Знак пунктуации в иврите
	U+2223	DIVIDES	Математический символ, обозначение делимости
	U+2502	BOX DRAWINGS LIGHT VERTICAL	Одинарная вертикальная линия (происходит из псевдографики)
	U+2503	BOX DRAWINGS HEAVY VERTICAL	Жирная одинарная вертикальная линия
	U+2758	LIGHT VERTICAL BAR	Вертикальная черта из набора значков (Dingbats)

Кроме того, вертикальную черту можно сделать с помощью CSS, указав свойство `border-left` или `border-right` для блока, в котором находится каждая отдельная ссылка.

Выбирайте тот способ, который лучше подходит к общему стилю вашей страницы. Не забывайте, что у пользователя ссылки могут отображаться шрифтом совсем не той гарнитуры, что вы указали. Поэтому употребляйте такую черту, которая поддерживается в разных шрифтах и хорошо сочетается с текстовыми ссылками.

Например, в качестве разделителя можно использовать какой-нибудь симметричный «вертикальный» символ — то есть значок, вытянутый в высоту. Вот несколько таких значков из Unicode:

¦	U+00A6	╫	U+2551
‖	U+2016	♦	U+2666

Менее удачные разделители получаются из значков, сопоставимых по высоте со строчными буквами. Но иногда и такие значки удачно смотрятся, например:

·	U+2022	□	U+25A1	◆	U+25C6	★	U+2605
■	U+25A0	▼	U+25BC	○	U+25CB	✦	U+2726

«Горизонтальные» символы (например, тире или дефис) практически не годятся для разделения ссылок. И, наконец, несимметричные знаки (✔, ◥, ¶ и т. д.) не подходят совсем.

Кроме разделения ссылок с помощью вышеперечисленных знаков, их можно обозначать другими способами. Например, можно заключить каждую ссылку в скобки или другие специальные ограничители:

(Адреса и телефоны) (Схема проезда) (Банковские реквизиты)

[Адреса и телефоны] [Схема проезда] [Банковские реквизиты]

〈 Адреса и телефоны 〉〈 Схема проезда 〉〈 Банковские реквизиты 〉

Другое дело — «хлебные крошки», обозначающие путь к текущей странице от главной страницы сайта (рис. 6.4).

Developers Home > Products & Technologies > Development Tools > Sun Java Studio Standard >

а

useit.com → Jakob Nielsen bio → Publication list

б

Рис. 6.4. «Хлебные крошки»: *а* — на сайте Sun и *б* — на сайте Якоба Нильсена

Примечание

Название «хлебные крошки» происходит из народных сказок. Злая ведьма уносила детей в лес, а они в пути крошили хлеб и высыпали на дорогу, чтобы потом по крошкам вернуться обратно. Но однажды птицы склевали все крошки, и дети не смогли вернуться. Постарайтесь, чтобы посетители вашего сайта не оказались в такой ситуации.

Для разделения «хлебных крошек» необходим именно несимметричный символ, отмечающий подчиненность одних ссылок другим. Таким символом может быть знак «>» (как на сайте Sun), косая черта и т. д. Знаменитый эксперт в области удобства сайтов Якоб Нильсен употребляет на своем сайте стрелочки (→).

Дополнительно

Ссылки в панелях навигации и «хлебных крошках» не должны переноситься по строкам. Если окно браузера очень узкое, сама панель навигации или цепочка «крошек» может размещаться на нескольких строках, но каждая отдельная ссылка при этом должна оставаться единой. Этого можно достичь, используя неразрывные пробелы:

```
<a href="../../">Компания ABC</a> &gt; <a
href="../">О компании</a> &gt; Адреса и телефоны
```

Компания ABC > О компании > Адреса и телефоны

```
<a href="../../">Компания ABC</a> &gt;
<a href="../">О компании</a> &gt;
Адреса и телефоны
```

Компания ABC > О компании > Адреса и телефоны

Обратите внимание, что разделители (в данном случае «>») тоже «привязаны» неразрывными пробелами к предшествующим ссылкам.

Русскоязычный сайт Creative — пример не очень удачно составленных «хлебных крошек» (рис. 6.5). Треугольнички, разделяющие «крошки», смотрятся невыразительно, и не всегда понятно, где кончается одна ссылка и начинается другая. Например, ощущается разрыв между **Sound Blaster ®** и **Внутреннее устройство**, хотя на самом деле это одна ссылка. Разрыв строки происходит посреди ссылки. Неразбериха усиливается тем, что ссылки не подчеркнуты. Кроме того, первая строка «хлебных крошек» имеет отступ от левого края, а вторая — нет, что выглядит неаккуратно. Обратите также внимание на слово «описания» в последнем пункте «крошек».

Рис. 6.5. Неудачные «хлебные крошки» на сайте **ru.europe.creative.com**

Примечание

Если цепочка «хлебных крошек» заканчивается заглавием текущей страницы, то это заглавие не должно быть ссылкой. Не забывайте: гиперссылка не должна вести в то же место, где сама находится.

6.1.3. Короче!
Соблюдайте краткость в тексте на сайтах

> Глядя на большинство веб-страниц, я поражаюсь тому, что большая часть слов на них просто занимает место, потому что вряд ли кто-то будет эти слова читать.
>
> *С. Круг. Веб-дизайн*

Пользователи обычно не читают текст на сайте так же внимательно, как книги или журналы. В самом деле, люди открывают книги именно затем, чтобы читать, — но на сайты заходят вовсе не за этим. Пользователь посещает сайт с какой-либо конкретной целью: например, узнать прогноз погоды, загрузить нужный файл, получить справку по интересующей его теме, купить товар в сетевом магазине и т. д. Он хочет как можно скорее и проще выполнить свою задачу, и чем больше ему придется для этого читать, тем меньше он будет доволен.

Конечно, бывают сайты, предназначенные для чтения. Например, электронные библиотеки, сборники статей, сетевые дневники — на эти ресурсы пользователи заходят как раз затем, чтобы почитать. Но даже тут важно, чтобы пользователь как можно легче получил доступ непосредственно к тому тексту, который хочет прочесть, и чтобы ему не пришлось читать еще что-то постороннее.

В Сети все еще попадаются сайты, на главной странице которых красуется длинный текст примерно такого содержания:

Добро пожаловать!

Мы приветствуем вас в интернет-магазине DOSTAL.RU. У нас вы сможете купить книги, компакт-диски, аудиокассеты, DVD, а также канцелярские принадлежности и компьютерные аксессуары. Богатый ассортимент нашего магазина позволит вам выбрать именно то, что вам надо, а умеренные цены станут приятным сюрпризом. Кроме того, в магазине DOSTAL.RU действует система гибких скидок...

В подобных текстах нет никакого смысла. Более того, обычно они раздражают пользователей, так как занимают место, на котором могли бы разместиться гораздо более полезные объекты.

Вместо обстоятельного рассказа о том, куда попал пользователь и что ему могут здесь предложить — дайте ему возможность сразу начать действовать!

Не нужно специально уведомлять пользователя о том, что он находится в магазине. Об этом должен говорить внешний вид и стиль сайта. Если пользователь видит корзину для покупок, изображения товаров с указанием цен, текст вроде «Специальное предложение!», «Скидки до 40 %» — он уж как-нибудь догадается, что находится не на сайте оперного театра и не в электронной библиотеке.

Название магазина вполне достаточно поместить на логотипе в левом верхнем углу страницы.

Также нет никакого смысла перечислять в тексте товары, которые продаются в магазине. Достаточно ограничиться панелью навигации:

Книги

Аудио

Компакт-диски

Аудиокассеты

Видео

DVD

Прочее

Канцелярские принадлежности

Компьютерные аксессуары

Если очень хочется — можно поместить над этой панелью заголовок «Купить». Этого будет более чем достаточно.

Фразы рекламного характера про богатый ассортимент и умеренные цены совершенно неуместны. Пользователь должен узнать об ассортименте и ценах, взглянув на сам этот ассортимент и сами эти цены, а вовсе не из ваших хвалебных речей. Поместите на главной странице краткую информацию о трех-четырех самых интересных из ваших товаров: их названия, изображения, цены, ссылки на страницы с подробностями. Пускай пользователи сразу же увидят воочию, что и почем вы продаете. Можно также поместить на главной странице более мелкие ссылки на 5—10 других интересных товаров. Но ни в коем случае не надо пространных текстов!

Не надо писать и про «гибкие скидки». Если вы действительно предлагаете скидки, которые могут заинтересовать покупателей, — поместите где-нибудь сбоку небольшой, но заметный блок с ярким заголовком вроде «Скидки 40 % на эти товары», перечислением названий двух-трех товаров и ссылками на страницы, где пользователь сможет подробнее узнать о товарах и скидках на них.

И, уж конечно, совершенно ни к чему приветствие «Добро пожаловать!» В нем нет абсолютно ничего полезного для посетителя. Оно только занимает место, на котором могло бы располагаться описание какого-нибудь интересного товара.

Все вышеприведенные рассуждения справедливы не только для интернет-магазинов, но и для любых других сайтов. «Вступительные речи», да еще на первой странице, — это печальное наследие эпохи любительских страничек, и от него следует отказаться раз и навсегда. Пользователь должен действовать, а не читать!

Внимание!

Не создавайте «вступительных речей». Никаких «добро пожаловать», «уважаемые посетители» и т. д. Дайте пользователю возможность сразу приступить к работе с сайтом, не читая никаких предварительных текстов.

Примечание

Предположим, вы хотите предложить посетителям своего сайта поучаствовать в благотворительной лотерее. Вы планируете разместить на сайте текст примерно такого содержания: «Предлагаем вам принять участие в благотворительной лотерее, организованной Фондом защиты домашних животных. Средства, вырученные в результате этой акции, будут направлены…» и т. д. Не надо всего этого. Отведите на странице блок подходящего размера и напишите в нем всего несколько слов. Во-первых, там должен быть броский, оригинальный заголовок. Нет, только не «Хотите выиграть миллион?» или «Беспроигрышная лотерея!» Пользователи бегут от таких заголовков, как черт от ладана. Лучше придумать действительно что-то оригинальное, можно даже с юмором — вроде «А ты подумал о хомячках?» или «Поддержите этого кота!» (и рядом маленькая фотография животного). Затем — небольшой текст, даже скорее слоган, поясняющий смысл акции. Например: «Живая лотерея — купите билет, и вы узнаете, на какое домашнее животное вы похожи!» (В самом деле, в лотерею играют скорее для развлечения, чем в надежде на выигрыш. Если вы убедите пользователя, что ему будет *интересно* принять участие в лотерее, — он купит билет, даже не особенно рассчитывая на выигрыш. И, наконец, поместите ссылку на страницу, где пользователь сможет купить билет и по необходимости прочесть о лотерее подробнее. В общем, что-то в этом роде:

А ты подумал
О ХОМЯЧКАХ?

Фотография хомячка

ЖИВАЯ ЛОТЕРЕЯ!

Хотите узнать, **на какое домашнее животное вы похожи**?

Купите билет!

Купить билет

Подробнее...

Таким образом, можно сформулировать ряд рекомендаций, позволяющих сократить текст и сделать его более пригодным для размещения на сайте.

- Прежде всего ответьте на вопрос: нужен ли пользователю вообще этот текст? Содержит ли этот текст что-то, что пользователь действительно *захочет* читать? Изменится ли что-нибудь, если убрать этот текст вообще?

  Попробуйте смоделировать конкретную ситуацию: кто именно, в каких условиях и зачем станет читать ваш текст? Представляйте себе не такого пользователя, который действует так, как вам бы хотелось, а реального, который пришел на сайт с конкретной задачей и хочет ее выполнить.

Примечание

Допустим, руководитель компании хочет разместить на главной странице сайта свое новогоднее обращение к клиентам. Он хочет, чтобы посетители сайта его прочитали. Возможно, оно даже содержит какую-то небесполезную информацию. Но стоит посмотреть правде в глаза: реальные пользователи *не будут* читать это обращение, если только их специально не заставят. Значит, ему не место на главной странице. Если руководитель компании все-таки хочет поместить обращение на сайте, лучше отвести под него отдельную страницу и дать ссылку на нее с главной.

纸 Дополнительно

Генеральный директор предлагает: «Давайте поместим этот текст на главную страницу». Специалист по разработке сайтов возражает: «Он займет стратегически важное место, где расположены полезные службы и ссылки. Лучше поместить ваш текст на отдельную страницу, а с главной сделать ссылку». «Но ведь тогда никто не будет его читать», — спорит генеральный директор. Тут возникает закономерный вопрос: а зачем вообще нужен такой текст, который никто не будет читать?

- Если текст нельзя выкинуть целиком (что бывает достаточно редко) — удалите из него все лишнее, то есть все, что не несет пользователю конкретной, полезной информации:
  - в первую очередь откажитесь от всего, что хотя бы отдаленно напоминает рекламу. Пользователи уже научились «отфильтровывать» рекламные материалы. «Спешите!», «Распродажа!», «Только у нас!», «Широкий выбор, доступные цены» — все это может произвести эффект в телевизионной рекламе, но в печатных изданиях или на сайте пользователь инстинктивно пропустит такой текст, даже в случае, если это на самом деле не реклама, а только нечто похожее на нее;
  - уберите все «общие слова», все неконкретное. Например, не нужно писать: «Среди наших клиентов — крупные банки и нефтяные компании». Если имена конкретных клиентов не являются коммерческой тайной — лучше поместить на странице (например, с правой стороны) небольшой блок примерно такого содержания:

Василий ПУПКИН, президент банка **«АВС-Банк»**:	Фотография В. Пупкина
«Сотрудничество с DOSTAL.RU оказалось для нас особенно плодотворным...» Подробнее...	

Если В. Пупкин — действительно известный и уважаемый человек, то такой блок подействует на пользователей гораздо сильнее, чем неконкретная реклама;

Примечание

Характерный пример «лишнего», бессмысленного текста: «Новая версия нашей программы обладает заметными преимуществами по сравнению с предыдущей версией». Читатель и так понимает, что новая версия программы отличается от старой в лучшую сторону, иначе не было бы смысла ее выпускать.

- уберите текст, который повторяет информацию, доступную из логотипа, навигации и других элементов страницы. Не нужно писать: «Приветствуем вас в интернет-магазине DOSTAL.RU». Название магазина уже и так присутствует в логотипе и в заглавии страницы. Не надо писать: «У нас вы можете купить...» Пользователь и так видит, что можно купить.

❒ Наконец, перепишите оставшийся текст так, чтобы он стал как можно короче и проще. Предположим, вы хотите разместить на странице такой текст:

> К сожалению, по техническим причинам отдел «DVD» временно не работает. Работа отдела «DVD» будет восстановлена в течение дня. Администрация интернет-магазина DOSTAL.RU приносит свои извинения.

Этот текст лучше сократить и упростить:

Отдел «DVD» сегодня не работает.

Приносим свои извинения

и ждем вас завтра!

Примечание

Некоторые редакторы предлагают такое правило: «Написав текст для сайта, выкиньте из него половину. Потом отдохните час-полтора, прогуляйтесь, перекусите; затем вернитесь к тексту и выкиньте половину того, что осталось. Наконец, покажите этот текст другому редактору, но не говорите ему о том, что вы уже сократили его. Пусть ваш коллега снова уменьшит его вдвое. То, что получилось, можно размещать на сайте. Впрочем, лучше сократить еще наполовину».

Краткость — сестра таланта. Не забывайте об этом, когда составляете текст для размещения на сайте. Чем длиннее текст, тем меньше вероятность того, что пользователь вообще будет его читать.

Дополнительно

Избавляйте текст от «слов-паразитов». Эти слова могут подстерегать в самых неожиданных местах — например, в предложении: «Щелкните значок „Сканеры и камеры“, находящийся на Панели управления». На первый взгляд, ничего лишнего здесь нет. Однако слово «находящийся» вместе с запятой можно выкинуть — предложение от этого только выиграет.

6.1.4. Проще простого

Пишите простой текст

> *I don't know half of you half as well as I should like; and I like less than half of you half as well as you deserve.* This was unexpected and rather difficult. There was some scattered clapping, but most of them were trying to work it out and see if it came to a compliment.
>
> *J. R. R. Tolkien. The Lord of the Rings*[1]

Широко распространенное среди значительного количества сайтов Всемирной Паутины явление, на которое обращают внимание как специалисты, так и обычные пользователи, заключается в том, что текстовые формулировки, представленные на них, несмотря на выраженность формально корректными языковыми конструкциями, представляют определенные затруднения для оперативного усвоения среднестатистическим пользователем. Другими словами — текст на многих сайтах написан слишком сложно.

Только что вы увидели пример неоправданно усложненного текста. Разве вам нравится читать такой текст? Не совершайте подобных ошибок, когда сами составляете текст для сайтов. Пишите проще.

Примечание

Иногда разработчикам кажется, что громоздкий, «наукообразный» текст придает сайту солидность и создает «серьезный» стиль. Это совсем не так. Даже серьезные ученые больше любят читать тексты, написанные простым языком, а простые пользователи — тем более.

---

[1] «Я не знаю половину из вас и вполовину так хорошо, как хотел бы, и я люблю почти половину из вас вполовину меньше того, как вы заслуживаете». Это прозвучало неожиданно и малопонятно. Раздалось несколько разрозненных аплодисментов, но большинство хоббитов решили пропустить фразу мимо ушей, полагая, что она приведет к очередному комплименту. *(Дж. Р. Р. Толкин. Властелин Колец)*

Это не значит, что к пользователям следует относиться как к неразумным новичкам, которым все надо «разжевывать». Простота, о которой здесь говорится, относится не к содержанию ваших текстов, а к их формулировкам. Рассмотрим, например, такой текст:

Чтобы подписаться на лист рассылки, вам следует отправить письмо произвольного содержания по адресу **subscribe@dostal.ru**, в теме которого содержится слово «Subscribe» (регистр символов не важен). После этого на электронный адрес, который вы указали в своем письме, придет запрос на подтверждение вашей подписки, на который вы должны ответить, чтобы подтвердить действительность своего адреса, после чего вы получите подтверждение того, что вы подписаны на лист рассылки.

Гораздо лучше упростить этот текст следующим образом:

Как подписаться на рассылку?

1. Отправьте письмо по адресу **subscribe@dostal.ru**. Напишите в теме письма слово «Subscribe». Что вы напишете в самом письме — значения не имеет.
2. Вам придет сообщение от нашего робота.
3. Отправьте ответ на это сообщение. Таким образом мы убедимся, что ваш электронный адрес — настоящий.
4. Вскоре вы начнете получать сообщения нашей рассылки.

Вовсе не требуется разъяснять пользователям, на какие именно кнопки им надо нажимать, чтобы отправить ответ на сообщение. Но длинных предложений и крупных участков сплошного текста лучше избегать.

Что именно можно сделать для того, чтобы упростить текст?

- Не надо сложных грамматических конструкций. Попробуйте там, где возможно, обойтись без слова «который», без причастных и деепричастных оборотов.

  Например, предложение

  Являясь одним из учредителей компании «Ординал», В. Пупкин возражает против ее реорганизации

  лучше перестроить:

  В. Пупкин — один из учредителей компании «Ординал» — возражает против ее реорганизации.

  Дополнительно

  Избегайте страдательного залога. Вместо «Деятельность совета директоров была раскритикована В. Пупкиным» лучше написать «В. Пупкин раскритиковал деятельность совета директоров».

- Лучше разбить длинное предложение на два или три коротких и простых.

  Например, такое предложение:

  Выйдя из перехода по указателю «К улице Пограничников», пройдите до арки вдоль дома, на котором висит вывеска «Кассы»

лучше разбить на три:

Выйдите из перехода по указателю «К улице Пограничников». Вы увидите дом с вывеской «Кассы». Пройдите вдоль этого дома до арки.

- Не пользуйтесь официальным, канцелярским стилем.

Например, предложение

Стороны не пришли к соглашению по вопросу урегулирования имущественных разногласий в отношении здания на улице Пограничников

следует упростить:

Стороны не смогли договориться о том, кому принадлежит дом на улице Пограничников.

Дополнительно

Одна из распространенных проблем — много слов подряд в родительном падеже. Например: «Этот город не случайно выбран в качестве места проведения встречи министров образования стран Содружества». При беглом чтении может показаться, что слово «образования» относится к «странам», а не к «министрам». Не злоупотребляйте подобными конструкциями.

- Избегайте специальных технических и научных терминов, если текст не рассчитан только на специалистов.

Например, вместо «Очистите кэш-память своего браузера и обновите страницу» лучше написать: «Нажмите <Ctrl>+<F5>».

Примечание

Разработчики сайтов иногда злоупотребляют специальными терминами вроде «кэширование», «скрипт», «аутентификация» и т. д. Однако рядовой пользователь Интернета может вовсе не знать о том, что такое «кэширование», — точно так же, как любитель музыки может не знать о том, что такое «вольта» и «аккомпанемент», а пользователь обычного телефона — что такое «селектор» и «коммутация каналов».

Обращайтесь к пользователю на том языке, который он понимает. Только в этом случае ваш сайт будет по-настоящему эффективным.

- Придерживайтесь правила, которое советуют многие специалисты: «Один абзац — одна законченная мысль». Разбивайте текст на абзацы таким образом, чтобы для понимания каждого отдельного абзаца не требовалось перечитывать предыдущий текст или заглядывать в последующий.

Кроме того, старайтесь, чтобы абзацы были как можно короче. Не забывайте: в Сети следует стремиться к такому тексту, который будет хорошо восприниматься не только при внимательном чтении, но и при беглом просмотре.

- Размещайте наиболее значимые слова ближе к началу предложения. Например, в предложении

  На состоявшейся в прошлый вторник конференции «Наука-2004» академик В. Пупкин представил новое издание Математической Энциклопедии

  части лучше поменять местами:

  Новое издание Математической энциклопедии представил академик В. Пупкин на конференции «Наука-2004», состоявшейся в прошлый вторник.

  Действительно, в этом предложении главное — именно новое издание энциклопедии, а не информация о том, кто его представил и когда состоялась конференция.

Примечание

Все главное должно стоять на первом месте. В этом заключается так называемый принцип «перевернутой пирамиды», полезный в журналистике и еще более полезный при составлении текста для сайтов. Каждое предложение должно быть построено таким образом, чтобы наиболее важные слова были в начале. Каждый отдельный абзац также должен начинаться с ключевой информации. Текст в целом тоже нужно составлять так, чтобы его основные мысли и выводы размещались в одном или нескольких первых предложениях.

В самом деле, читатель обычно бывает раздражен, если ему приходится прочесть всю статью до конца, только чтобы понять, что ничего интересного для него в ней нет.

Принцип «перевернутой пирамиды» справедлив и для сайтов в целом. Наиболее важное содержание и первостепенные службы должны быть доступны пользователю с самого начала, с главной страницы. Менее важное наполнение сайта пользователь должен получить, перейдя в один из основных разделов. А наименее значимые, вспомогательные материалы и службы можно расположить в подразделах более глубоких уровней.

Дополнительно

Не пишите предложения, которые можно понять неоднозначно. Например: «Найдите в файле httpd.conf строку, содержащую запись index.html.var, и удалите ее». Здесь непонятно, что надо удалить — запись или строку. Или: «Хирург провел операцию у меня на глазах» — это можно понять не только как «Я видел, как хирург провел операцию», но и как «Хирург сделал мне операцию».

Если вы пишете короткие предложения — это обычно спасает от неоднозначности.

Помните: все гениальное — просто. Сделайте так, чтобы пользователь легко получал информацию, а не «выковыривал» ее из громоздких грамматических конструкций. Не заставляйте пользователей задумываться там, где без этого можно обойтись.

Примечание

Перечитывайте текст через некоторое время после того, как вы его составили. Обычно это позволяет избавить его от значительной части недостатков.

Конечно, в этой книге невозможно рассказать обо всем, что должен знать профессиональный редактор. Более подробную информацию о подготовке текстов можно найти в учебниках и справочниках по редактированию (например, в [1] и [6]).

6.1.5. Сверим наши стили

Придерживайтесь единого стиля

> Текстовое содержание сайтов, как правило, менее формально и более дружественно, чем обычные корпоративные брошюры и документы. Напомните об этом клиенту, чтобы он не просто копировал подходящие материалы, а редактировал их.
>
> *К. Гото, Э. Котлер. Веб-редизайн*

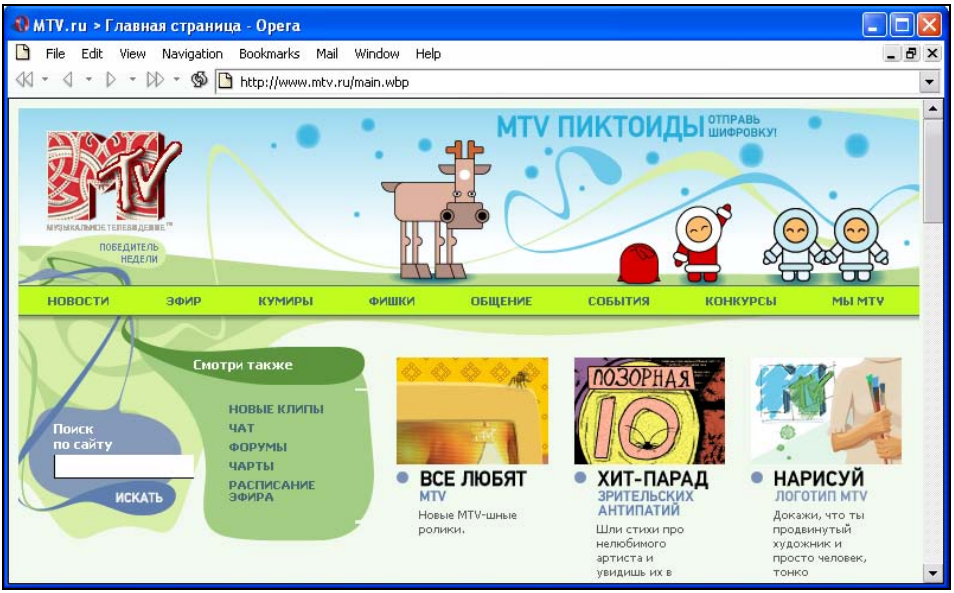
Стиль является важной составляющей сайта. Имеется в виду не только стиль текста, но и подбор материалов, их структура, оформление и все остальное, что придает сайту свое неповторимое «лицо». Сайт может быть строгим, академичным (например, сайт государственной организации или научного заведения), может быть ярким, кричащим (например, персональный сайт эстрадного исполнителя), а может производить теплое, «душевное» впечатление (например, сайт клуба садоводов).

Конечно, для воплощения стиля очень важен внешний вид, оформление сайта. Но это — не единственный компонент. То, что написано на сайте — его текстовое содержание — также должно соответствовать выбранному стилю. В противном случае даже самое изысканное оформление будет смотреться, как кружева на корове.

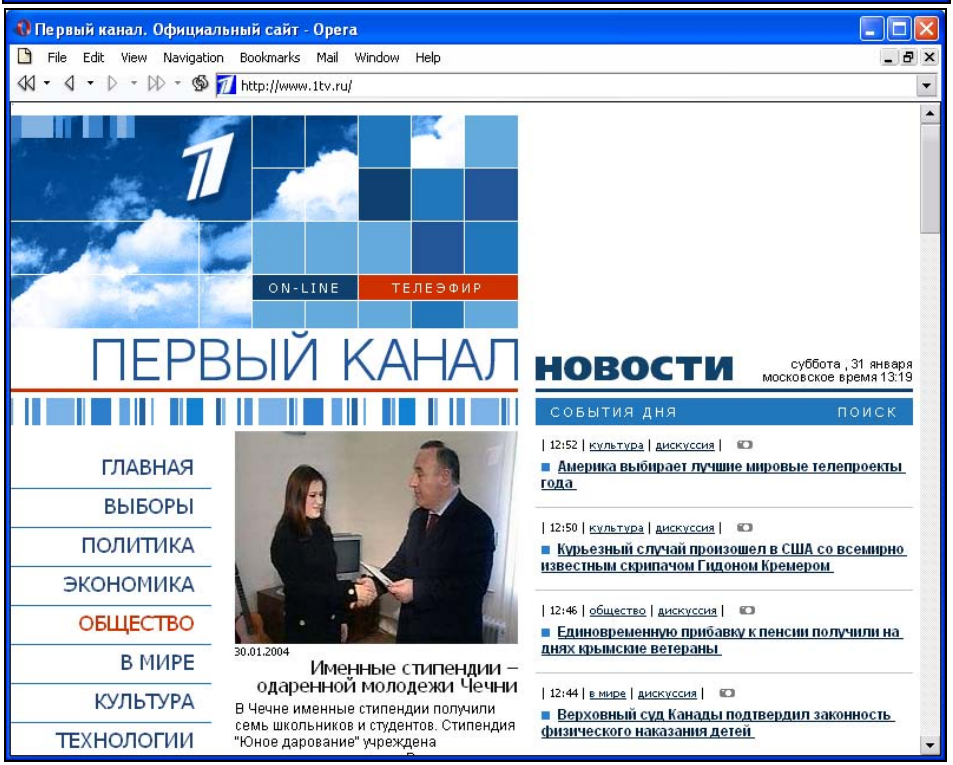
Внимание!

Придерживайтесь единого стиля для всех компонентов сайта. Пишите весь текст в одном и том же стиле — в стиле, которому соответствует концепция, содержание и оформление сайта.

На рис. 6.6 вы видите сайты двух телекомпаний. Сайт MTV ориентирован на молодежную аудиторию — и стиль текста на нем выбран соответствующим образом. Обратите внимание на постоянные обращения к посетителю: «Отправь шифровку», «Нарисуй логотип», «Шли стихи» и т. д. Напротив,



а



б

Рис. 6.6. Сравните стили написания текста на сайтах *а* — **www.mtv.ru** и *б* — **www.1tv.ru**

сайт Первого канала выдержан в строгом стиле: никаких «подзадориваний» пользователя, только аккуратный перечень ссылок и сообщения о событиях, написанные «официальным» языком.

Рекомендуется, чтобы все тексты, предназначенные для размещения на сайте, просматривал один и тот же литературный редактор. Все тексты — от надписей на кнопках до крупных статей — должны быть приведены в стилистическое соответствие друг с другом. Иначе сайт будет оставлять впечатление неаккуратности и беспорядка.

Употребляйте всюду на сайте одинаковую терминологию. Например, если в одном месте вы пишете «гостевая книга», а в другом называете эту же службу «доской объявлений» — это может дезориентировать пользователя. Если то, что на одной странице называется «электронным письмом», на другой странице фигурирует как «сообщение электронной почты» — пользователь не запутается, но у него возникнет ощущение некоторой неряшливости вашего сайта.

纸 Дополнительно

Стилистическое единообразие проявляется не только в единой терминологии, но и в лексике, в манере составления предложений. Если вы читаете книги, то наверняка убедились, что у каждого автора — свой характер изложения. Можно отличить стиль Толстого от стиля Тургенева. Для текста на сайтах единство стиля не менее важно, чем для художественных произведений. Поэтому необходимо, чтобы над сайтом работал специальный редактор.

6.1.6. И смех и грех

Не злоупотребляйте метафорами и юмором

> Метафора — это гвоздь в ботинке; ее надо выполоть.
>
> *Дж. Тригг. Грамматика*

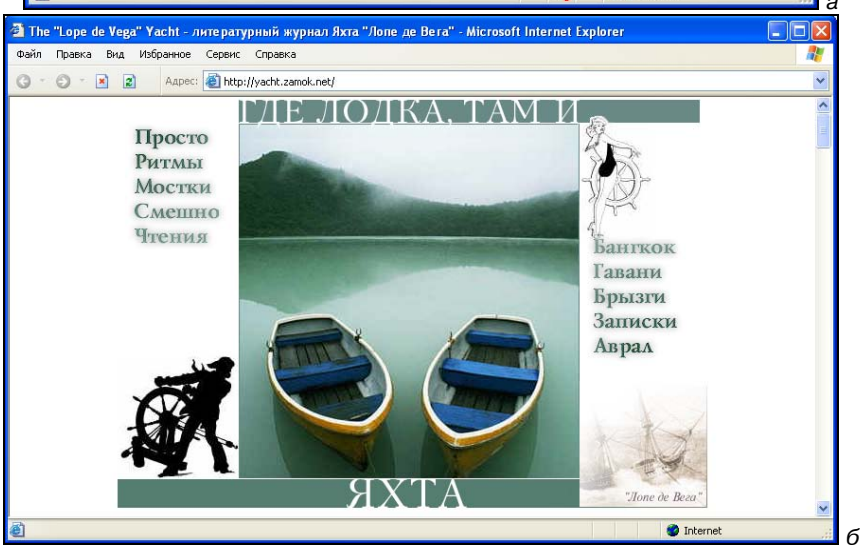
Метафора — это образное иносказание, употребление слова в переносном значении. Этот стилистический прием широко распространен в художественной литературе — например, у Лермонтова:

> Ночевала тучка золотая
> На груди утеса-великана...

Понятно, что у утеса не бывает никакой груди. Однако этот образный оборот позволяет передать художественный замысел автора.

Мы употребляем метафоры в повседневной жизни — например, когда называем глупого человека «ослом», а грубого — «медведем».

а



б

Рис. 6.7. Метафоры на сайтах: *а* — **arda.kulichki.com** и *б* — **yacht.zamok.net** не дают пользователю понять, что находится в тех или иных разделах

Однако в тексте на сайтах метафор лучше избегать. Слишком велика вероятность того, что пользователи поймут вас буквально (или не поймут вовсе — неизвестно, что хуже).

Посмотрите, например, на сайт «Арда-на-Куличках», созданный любителями творчества Толкина (рис. 6.7, *а*). Названия разделов представляют собой типичные метафоры — и из-за этого пользователь может легко запутаться. Например, что такое «Онтомолвище»? Или чем отличаются «Архивы Минас-Тирита» от «Подшивки Лэймара»? И правда ли, что информацию о романе «Властелин Колец» следует искать в разделе «Кольцо»? Не думайте, что эти названия смущают только тех, кто не увлекается творчеством Толкина. Для толкиниста, впервые посетившего этот сайт, они также выглядят крайне загадочно.

Подобные проблемы характерны для многих сайтов литературного и художественного направления (рис. 6.7, *б*). Но в этих случаях изощренные метафоры попросту мешают сайту выполнять ту задачу, ради которой он создавался, — т. е. предоставлять пользователям доступ к интересующим их художественным произведениям. Рассказы, стихотворения, эссе, полные ярких метафор и образов, несущие художественную ценность, — это просто прекрасно, но зачем «охудожествливать» *средства доступа* к этим произведениям?

В тексте на сайте также не надо злоупотреблять юмором. Текст вполне может передавать дружеское, доброжелательное отношение к пользователю — но шутки лучше оставить при себе. У разных людей разное чувство юмора: некоторые пользователи могут просто не понять, что вы хотели сказать, а некоторые могут понять неправильно и обидеться.

Примечание

Даже пользователи с хорошим чувством юмора нередко бывают дезориентированы, встретив юмор на сайте. Дело в том, что, посещая сайты, пользователь обычно *не ожидает* увидеть там шуток и иносказаний; он привык к обычному стилю содержания на сайтах и склонен воспринимать все буквально.

Если вы размещаете на сайте материал, не предназначенный специально для Сети (например, текст художественного произведения или подборку афоризмов) — уведомите пользователя об этом, чтобы он своевременно «переключил» свое восприятие и не начал читать литературный текст так же, как ленту новостей.

В Сети много сайтов, посвященных анекдотам, шуткам, розыгрышам, забавным историям. Но даже если вы делаете именно такой сайт, не превращайте его интерфейс в розыгрыш!

6.1.7. Честно говоря
Размещайте на сайте только достоверную информацию

Ферапонт. И тот же подрядчик сказывал — может, и врет, — будто поперек всей Москвы канат протянут.
Андрей. Для чего?
Ферапонт. Не могу знать. Подрядчик говорил.

А. П. Чехов. Три сестры

Для Всемирной Паутины характерна невиданная ранее свобода слова. Любой желающий может разместить в Сети свой сайт и написать там все что угодно. Эта свобода имеет и негативную сторону: на сайтах оказывается очень много непроверенной, недостоверной, некомпетентной или намеренно фальсифицированной информации.

В результате пользователи мало доверяют информации, полученной из Интернета, и предпочитают обращаться к более авторитетным источникам.

Примечание

Сказанное не относится к *официальным* представительствам фирм или проектов в Интернете. В самом деле, нет никаких сомнений в том, что на сайте **www.mozilla.org** находится грамотная и достоверная информация о браузере Mozilla, а на сайте **www.msu.ru** — официальная, проверенная информация о Московском государственном университете имени Ломоносова.

Но в Интернете очень сложно найти достоверную информацию о предмете, у которого нет собственного официального сайта.

Если вы размещаете на сайте данные, полученные из сторонних источников, — обязательно убедитесь в их достоверности. Если существует официальный источник информации на эту тему — сверьтесь с ним. Если нет — проверьте информацию хотя бы по двум независимым источникам.

Внимание!

Указывайте на сайте источники размещенной на нем информации. В противном случае доверие пользователей не гарантируется.

Дополнительно

Одно из «больных мест» Интернета — вопросы, связанные с авторством. С сайта на сайт кочуют популярные тексты, статьи, высказывания, литературные произведения — но информация об авторе при этом теряется.

Прежде чем размещать на сайте текст, написанный не вами, обязательно узнайте, кто его автор, и поинтересуйтесь, согласен ли он на публикацию его текста.

Не размножайте сплетни, слухи и домыслы. Их и так во Всемирной Паутине предостаточно. Если вы сомневаетесь в объективности той или иной информации, и нет возможности ее проверить — лучше не помещайте такую информацию на свой сайт.

Дополнительно

В Интернете размещено немало любительских «учебников» по HTML, CSS, JavaScript и другим технологиям. Они написаны дилетантами, обычно содержат множество ошибок, и обучаться по ним крайне не рекомендуется.

Это же касается и некоторых «переводов» официальных спецификаций HTML и CSS, которые можно найти в русскоязычной Сети. Эти переводы неточны, сумбурны, в них не соблюдается строгая терминология; кроме того, они отвратительно выполнены с точки зрения русского языка, полны элементарных грамматических ошибок, не говоря уж о стиле изложения.

Еще одна проблема — искажение информации ради того, чтобы произвести более сильное впечатление на пользователя. Например, сайт «5 баллов» открывается сенсационным заголовком: «Российские вузы возглавляют мошенники и бюрократы» (рис. 6.8, *а*). Перейдя по этой шокирующей ссылке, можно убедиться, что на самом деле речь идет всего лишь о том, что думают о руководителях вузов участники небольшого опроса (рис. 6.8, *б*). Такой прием широко употребляется в «желтой» прессе, но совершенно не годится для сайта, которому дорогá его репутация.

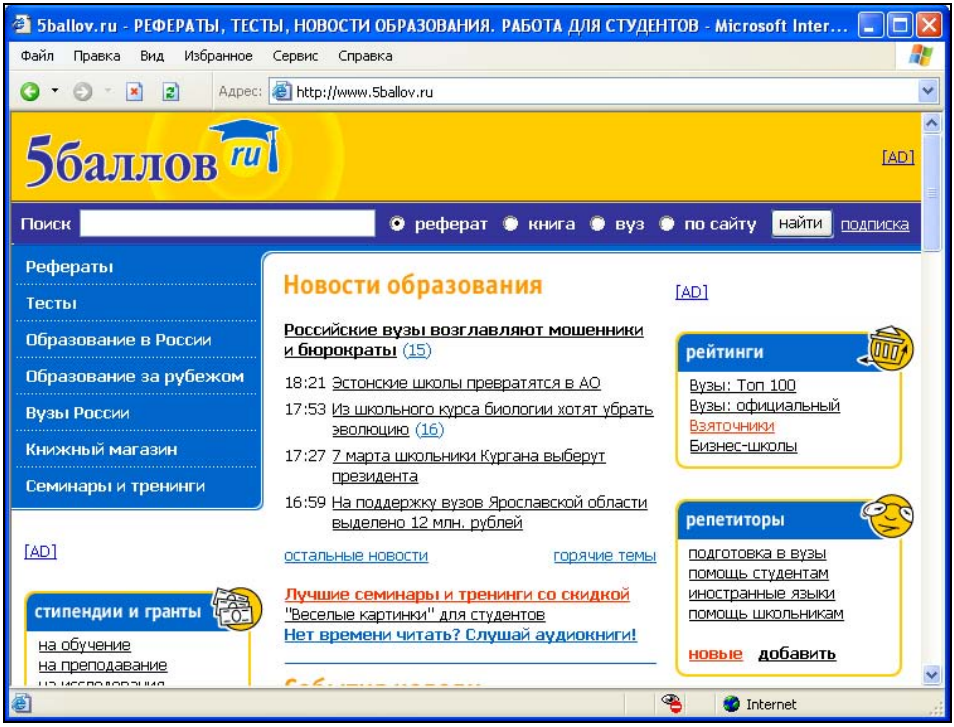
Заботьтесь не только о достоверности информации, но и о ее актуальности. Всегда указывайте временной период, в течение которого эта информация действительна. Например, не надо писать «Завтра состоится конференция». Может случиться так, что ваш текст сохранится в Сети и завтра, и послезавтра, и через год. Укажите конкретную дату и год события, чтобы не ввести пользователей в заблуждение.

Примечание

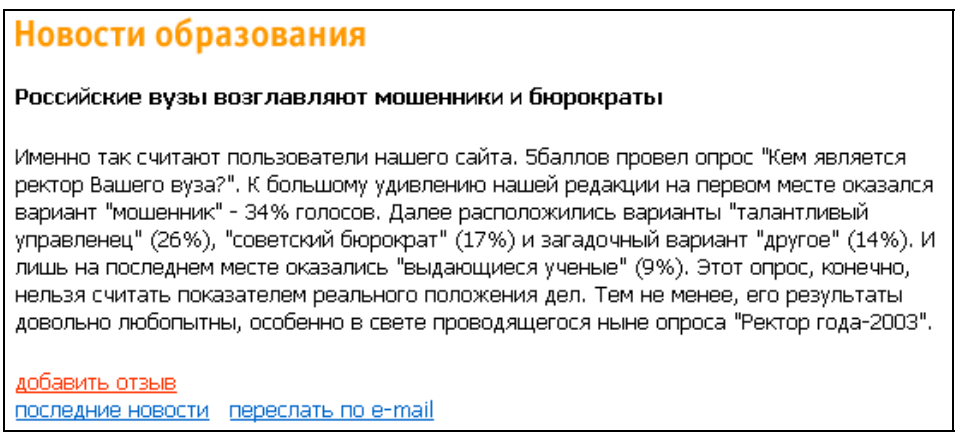
Автор встречал в Сети много статей на техническую тематику, в которых представлялись достоинства того или иного нового устройства — процессора, звуковой карты, сотового телефона и т. д. Но устройства со временем устаревают, вытесняются новыми моделями — а на сайте так и остается статья, говорящая о «новых, передовых технологиях», «последних достижениях» и «ультрасовременном дизайне». Это сбивает с толку пользователей-неспециалистов.

Размещая на сайте текст, учтите, что через некоторое время он устареет. Своевременно удаляйте или переписывайте устаревшие тексты. Если старый

текст должен сохраниться на сайте, сопроводите его комментарием о том, когда он был написан, и предупреждением, что сейчас он не является полностью актуальным.



а



Новости образования

Российские вузы возглавляют мошенники и бюрократы

Именно так считают пользователи нашего сайта. 5баллов провел опрос "Кем является ректор Вашего вуза?". К большому удивлению нашей редакции на первом месте оказался вариант "мошенник" - 34% голосов. Далее расположились варианты "талантливый управленец" (26%), "советский бюрократ" (17%) и загадочный вариант "другое" (14%). И лишь на последнем месте оказались "выдающиеся ученые" (9%). Этот опрос, конечно, нельзя считать показателем реального положения дел. Тем не менее, его результаты довольно любопытны, особенно в свете проводящегося ныне опроса "Ректор года-2003".

добавить отзыв
последние новости переслать по e-mail

б

Рис. 6.8. Шокирующий заголовок приводит ко вполне безобидному материалу

Дополнительно

На сайтах часто размещаются объявления о предстоящих событиях, реклама будущих акций, конференций, семинаров, концертов и т. д. Не оставляйте эту информацию на сайте после того, как рекламируемое событие уже состоялось! А на странице, которая раньше была посвящена предстоящему событию, поместите отчет об этом событии.

6.1.8. Внесите разнообразие! Не пренебрегайте средствами оформления

> За счет выделений текст может стать яснее, убедительней, занимательней и даже компактней, ибо типографические обозначения способны заменить собой многократно повторяющиеся слова.
>
> *В. Кричевский. Типографика в терминах и образах*

Однообразный текст читается очень плохо, особенно в Сети. Грамотное оформление позволяет «оживить» текст, сделать его легче для восприятия.

Предположим, мы хотим разместить на сайте биографию:

ПУПКИН Василий Эдуардович родился в 1964 году в Обнинске в семье служащих. В 1986 окончил с отличием Московский инженерно-физический институт по специальности «Прикладная математика и информатика». С 1986 по 1989 гг. работал научным сотрудником в НИИ им. Котовского. В 1989 основал кооператив «Василек», специализирующийся на продаже и ремонте бытовой техники. С 1991 по 1996 год работал в компании «Зеро-финанс», где прошел путь от руководителя отдела до вице-президента компании. В 1996 г. основал «АВС-Банк» и стал его руководителем. В 2000 г. возглавил международную финансовую группу «АВС», выросшую из «АВС-Банка». Женат, имеет двоих детей. Увлекается горнолыжным спортом.

Эта биография будет читаться гораздо лучше, если снабдить ее оформлением:

ПУПКИН Василий Эдуардович

Фотография В. Пупкина

- **Родился в 1964** в Обнинске в семье служащих
- **Окончил с отличием МИФИ** (1986). Специальность — прикладная математика и информатика
- **Работал научным сотрудником** в НИИ им. Котовского (1986–1989)
- **Основал кооператив «Василек»** (1989) — продажа и ремонт бытовой техники
- **Работал в компании «Зеро-финанс»** (1991–1996). Вице-президент компании (1995–1996)
- **Основал «АВС-Банк» (1996).** Председатель совета директоров «АВС-Банка» с момента основания и по сей день
- Глава международной финансовой группы «АВС» (с 2000)
- **Женат, двое детей.** Увлечение — горнолыжный спорт

Обратите внимание, что именно сделано для улучшения текста. Во-первых, предложения перестроены таким образом, чтобы начинаться с наиболее важных слов. Во-вторых, важные слова выделены полужирным шрифтом, так что даже при беглом взгляде пользователь обратит на них внимание. В-третьих, тексту придан вид маркированного перечня, что также облегчает восприятие. И, наконец, текст снабжен фотографией и ссылками на сайты упомянутых компаний.

Дополнительно

Напомним, что текстовое выделение, которое здесь показано полужирным шрифтом, лучше делать с помощью элемента `strong`, а не какими-либо другими средствами. Аналогично, то, что обычно отображается курсивом, — это элемент `em`.

Кроме этого, для оформления можно использовать текст или фон другого цвета, рамки и другие средства. Важно не переусердствовать и не применять к одному и тому же участку текста более одного (в редких случаях двух) средств выделения. Например, если текст выделен одновременно полужирным шрифтом и другим цветом — это явное излишество. Аналогично, не стоит дополнительно выделять ссылку: она и так выделена другим цветом и подчеркиванием. Наконец, никогда не оформляйте текст одновременно полужирным шрифтом и курсивом — это выглядит очень некрасиво.

Примечание

Напомним, что ссылки должны всегда оставаться подчеркнутыми, иначе пользователь не будет знать, какой текст является ссылкой, а какой — нет. Для ссылок внутри текста (как в этой биографии) это требование особенно важно. И наоборот, никогда не подчеркивайте текст, который не является ссылкой.

Дополнительно

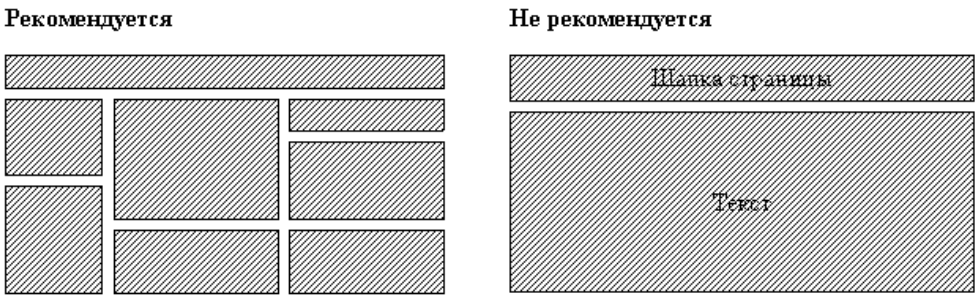
Выделения другим шрифтом в массиве сплошного текста обычно смотрятся неаккуратно. В редких случаях это уместно (например, в этой книге имена тэгов XHTML выделяются `моноширинным` шрифтом), но обычно такой способ выделения применять не следует.

Еще один неудачный способ выделения — когда при наведении указателя мыши на ссылку текст этой ссылки вдруг становится полужирным, курсивным, увеличивается в размерах или меняет гарнитуру. Это любительский прием, и пользоваться им не следует. При наведении указателя ссылку можно «подсвечивать», т. е. слегка изменять цвет фона. (Цвет шрифта лучше не изменять — это кажется пользователям чем-то навязчивым.)

Обычно веб-страница воспринимается не очень хорошо, если всю ее ширину занимает один блок текста. Исключением является случай, когда на

странице размещена статья или литературное произведение, ради которого пользователь пришел на сайт.

В большинстве случаев основную часть страницы лучше разбить на две или даже три колонки, а каждую колонку — на отдельные блоки:



Дополнительно

Пользователи сайтов обычно не читают текст на веб-страницах, а бегло просматривают его, «выхватывая» то, что кажется им ценным — ссылки, файлы для загрузки и т. д. Если пользователь зашел на страницу не специально для того, чтобы прочесть находящийся на ней текст, то его взгляд будет хаотично блуждать по странице. Если при этом всю страницу занимает один сплошной текст — скорее всего, пользователь не будет «переключаться» в режим внимательного чтения и покинет страницу. Если же страница состоит из отдельных блоков, каждый из которых в отдельности легко охватывается взглядом — пользователь обязательно «зацепится» хотя бы за один или два блока.

6.1.9. Заставьте текст работать!

Применяйте элементы, с которыми пользователь может взаимодействовать

> 85. Я не буду использовать план, в котором последний шаг будет очень сложным, например: «Соединить 12 Камней Силы на священном алтаре и активировать медальон в момент полного затмения». Скорее, это будет что-то вроде «Нажать на кнопку».
>
> *П. Анспрач. 100 вещей, которые я сделаю, когда стану Злым Властелином*

Пользователи обычно не любят просто читать текст на сайте. Одно из отличий веб-страниц от печатных изданий — это интерактивность, возможность взаимодействия. Не упускайте эту возможность.

- Обеспечивайте в тексте ссылки на связанные ресурсы. Например, если в тексте упоминается какое-то лицо или организация — сделайте имя или название ссылкой на официальный сайт:

  На конференции «Наука-2004» руководитель «АВС-Банка» В. Пупкин и ректор Народного университета Б. Собакин подписали соглашение о сотрудничестве.

  В расстановке ссылок, как и во всем остальном, нужно соблюдать меру. В частности, не рекомендуется делать разными ссылками соседние слова; между ссылками должно быть хотя бы несколько «обычных» слов. Плохо, если ссылок в тексте слишком много: в коротком тексте (например, в новостном сообщении) они должны занимать не более ¼ от всего текста, а в крупной статье лучше размещать не более 1—2 ссылок в пределах абзаца.

- Предоставляйте ссылки на другие ресурсы по теме. Посмотрите, например, на оформление статьи на CNews (рис. 6.9): после текста следуют ссылки на учебные курсы, относящиеся к теме сообщения, а также на другие сообщения по этой тематике.

  Например, если ваш сайт публикует научные статьи, на странице с каждой статьей можно привести список других статей на эту же тему и других статей этого же автора.

  Тактика «ссылок по теме» успешно работает в интернет-магазинах: на странице, посвященной какому-либо товару, приведены также ссылки на другие похожие товары, на принадлежности и аксессуары, подходящие к этому товару, а также на товары, которые пользователи обычно покупают вместе с данным.

- Предоставляйте пользователю возможность что-либо сделать с данным текстом: например, загрузить его в формате PDF, обсудить на форуме, проголосовать за него и т. д. Здесь, как и в случае со ссылками, надо знать меру; но две-три вспомогательных возможности обычно не помешают. Пользователь должен чувствовать, что он находится в интерактивной информационной среде и может активно работать в ней, а не просто воспринимать информацию, как при чтении печатного издания или просмотре телепрограммы.

Примечание

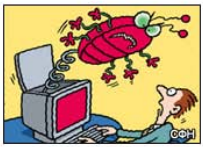
Организуйте обратную связь с пользователями. Не забывайте про еще одно важное отличие сайтов от печатных изданий: в вышедшей книге или журнале уже нельзя исправить допущенную ошибку или опечатку, а сайт можно исправить всегда. Дайте пользователям возможность сообщать вам об ошибках и оперативно исправляйте эти ошибки.

Для того чтобы такая служба работала, необходимо, чтобы пользователи убедились в ее действенности — т. е. чтобы они видели, что вы действительно исправляете ошибки и учитываете пожелания пользователей в работе над сайтом. Если пользователи поймут, что их отзывы действительно способствуют улучшению сайта — они будут еще активнее взаимодействовать с вами, облегчая вашу работу по тестированию и отладке сайта.

30.01.04, Пт, 11:05, Мск

в портфель обсудить

Голова автора вируса MyDoom оценена в $500 тыс.



Награда за информацию, которая могла бы помочь аресту и осуждению автора или группы лиц, участвовавших в создании вируса MyDoom, возросла вдвое. Примеру софтверной компании SCO Group, предложившей награду в 250 тыс. долларов, последовала корпорация Microsoft, официальный сайт которой - www.microsoft.com - должен стать объектом DDoS-атаки.

Червь Novarg/MyDooM распространяется в сетях 170 стран и по нанесенному ущербу входит в десятку самых опасных почтовых червей за все время существования интернета. Напомним, что мишенью первой версии вируса был сайт компании SCO Group – www.sco.com, и именно этот факт заставил ее обеспокоиться и назначить награду за информацию о вирусописателе. Однако впоследствии вирус мутировал и теперь зараженные компьютеры стали угрозой еще и для сайта Microsoft.

Червь опасен только для компьютеров на базе операционной системы Microsoft Windows. По мнению экспертов по сетевой безопасности, это позволяет считать, что автор вредоносной программы является сторонником ОС Linux. SCO Group пыталась заявить права на некоторые части Linux, и ее судебные притязания, хотя и окончились провалом, не понравились многим поклонникам программного обеспечения с открытым кодом. Судебные иски, поданные SCO Group, вызвали недовольство хакеров из разных стран, и веб-ресурсы SCO Group стали целью постоянных атак.

Microsoft, в свою очередь, готовится к защите от нападения на свой корпоративный сайт. «Наш сайт – второй или третий крупнейший в мире, и мы предпринимаем различные меры для его защиты, о которых не считаем нужным рассказывать публично», – заявил представитель Microsoft Шон Сандволл (Sean Sundwall).

Британская исследовательская компания mi2g считает, что экономический ущерб от вируса уже достиг $3 млрд. В оценку ущерба входят убытки компаний от потерь пропускной способности в каналах передачи данных, снижение производительности сетевых ресурсов и служб, увеличение задержек в управлении и стоимость восстановления работоспособности зараженных систем.

Источники: при подготовке материала использованы сообщения New York Posr и mi2g.

Учебные курсы по теме:

Применение межсетевых экранов для защиты корпоративных сетей

Безопасность сетей на базе Windows 2000/XP

Системы анализа защищенности Internet scanner и Database scanner

в портфель обсудить CNews.ru

см. также:

- Вышла публичная бета-версия Microsoft ISA Server 2004
- SCO предложила деньги за голову автора MyDoom
- Вирусные эпидемии приравняли к терроризму
- Mydoom мутировал, Сеть ожидает новой эпидемии
- Эпидемия Mydoom уже стоила бизнесу более $1 млрд.
- Microsoft устремился на рынок RFID
- Вирус парализовал почту меломанов
- IBM и Microsoft представили планы по миграции на Linux
- Вирус дискриминирует Microsoft
- Microsoft: грамотность для бедных

Рис. 6.9. www.cnews.ru: важен не только текст сообщения, но и аппарат ссылок, связанных с этим текстом

6.1.10. В особо крупных размерах
Что делать с крупными текстами

> Столпотворение не умещалось в сенях и на дворе и занимало пол-улицы перед окнами присутствия. К столам нельзя было протиснуться. За гулом сотни голосов никто ничего не понимал.
>
> *Б. Пастернак. Доктор Живаго*

Если на сайт все-таки требуется поместить крупный текст — его надо предварительно обработать. Речь не идет о литературных произведениях или других материалах, форма которых уже сложилась и должна оставаться неизменной. Но если вы сами составляете текст специально для того, чтобы разместить его на сайте, — пожалуйста, приведите его в порядок.

- Прежде всего подумайте, нельзя ли разбить этот текст на два или более независимых текстов. Три статьи, каждая из которых умещается в экран, — это всегда лучше, чем одна статья, простирающаяся на три экрана.

Дополнительно

Линейная структура, когда страницы выстраиваются в логическую последовательность (как в документации Консорциума W3C — см. рис. 6.3), в целом не подходит для Всемирной Паутины. Линейная структура характерна для печатных изданий, а веб-страницы лучше объединять в иерархическую структуру. Поэтому вместо того, чтобы разбивать одну большую статью на несколько страниц и связывать эти страницы в последовательность, лучше сделать несколько небольших статей, каждую из которых можно читать независимо от остальных.

Разумеется, это не всегда возможно: например, уже написанную книгу или техническую документацию нельзя переделать в набор независимых статей. Но когда текст в вашей власти, всегда старайтесь следовать этой рекомендации.

- Текст, который больше не делится на отдельные независимые тексты, разбейте на части по смыслу. Здесь можно разбивать гораздо мельче, чем в книгах: нет ничего плохого в том, чтобы часть занимала всего 2—3 абзаца. Придумайте для каждой части заголовок.

Примечание

Заголовки целого текста и каждой его части должны представлять собой нечто вроде очень краткого конспекта всего текста. Если вы пишете текст о выращивании фикуса в домашних условиях, так и назовите его — «Выращивание фикуса в домашних условиях». Заголовки, подходящие для журналов — «Зеленый друг», «Вот тебе и фикус» и т. д. — не годятся для веб-страницы. Не надо

называть сообщение о выставке произведений Фаберже «Дорого яичко...», а материал о выборах депутатов — «Утро стрелецкой казни». Из заголовка должно быть совершенно понятно, о чем, собственно, текст.

Дополнительно

Не забывайте, что заголовок целого текста должен быть размечен тэгами `<h1>...</h1>` (заголовок первого уровня), а заголовок каждой отдельной части — `<h2>...</h2>` (заголовок второго уровня). Если части текста, в свою очередь, сами делятся на еще более мелкие части, используйте заголовки третьего уровня и более.

- Разбейте текст на абзацы таким образом, чтобы каждый абзац представлял собой одну законченную мысль. В свою очередь, каждая часть текста, объединяющая несколько абзацев, также должна выражать цельную, обособленную совокупность мыслей. Отдельные абзацы и части не должны дополнять и пояснять друг друга. Каждый абзац должен говорить о чем-то своем.
- Используйте врезки (например, как «Примечание» и «Дополнительно» в этой книге). Во врезках можно высказывать наиболее важные предложения (выделяя их увеличенным шрифтом, заключая в рамку и т. д.), или, наоборот, вспомогательные сведения (также выделяя их соответствующим образом). Для текстов на сайте достаточно врезки какого-нибудь одного типа. Будет вполне достаточно, если на 4—5 обычных абзацев приходится одна врезка.

Дополнительно

Напомним, что наиболее подходящим элементом XHTML для врезки является `blockquote`. Этот элемент не может содержать внутри себя непосредственно текст, но может содержать блочные элементы:

Правильно	Неправильно

`<blockquote>`
`<p>Напомним, что наиболее подходящим...</p>`
`</blockquote>`
	`<blockquote>`
	`Напомним, что наиболее подходящим...`
	`</blockquote>`

Разумно применяя врезки, вы разнообразите внешний вид текста и сделаете его более привлекательным для пользователей.

- Создайте оглавление вашего текста — т. е. список заголовков всех его частей. Разместите это оглавление сразу после основного заголовка (в виде

нумерованного или маркированного перечня) и сделайте его пункты ссылками на соответствующие части в тексте, например:

Выращивание фикуса в домашних условиях

1. Приобретение семян
2. Проращивание
3. Высаживание в грунт
4. Освещение
5. Поливка

и т. д.

Примечание

Иногда встречаются неправильные «содержания» вроде:

- Часть 1
- Часть 2
- Часть 3

Такое «содержание» бессмысленно, так как пользователь не получит из него никакой информации о страницах, на которые ведут ссылки.

- Составьте аннотацию текста, не превышающую 3—4 предложений. Поместите эту аннотацию перед текстом: возможно, некоторым пользователям будет достаточно прочитать только аннотацию, и это позволит им сэкономить свое время.

Примечание

Аннотация должна отражать действительное содержание статьи, а не носить рекламный, «завлекательный» характер:

Рекомендуется	Не рекомендуется
Как вырастить фикус в обычной городской квартире? Кандидат биологических наук В. Пупкин предлагает детально описанную процедуру выращивания, начиная от приобретения семян и заканчивая сбором и хранением плодов.	Фикус — это просто! Уникальный метод профессора В. Пупкина! Вырастите самый большой фикус всего за 2 месяца! Без капиталовложений!

Кроме того, разместите аннотацию на той странице, где находится основная ссылка, ведущая к вашему тексту. Посмотрите, как аннотации работают на сайте Я. Нильсена. На главной странице помещается не только заголовок новой статьи, но и аннотация к ней (рис. 6.10, *а*). Эта же аннотация (Summary) повторяется перед текстом самой статьи (рис. 6.10, *б*).

useit.com: Jakob Nielsen on Usability and Web Design - Microsoft Internet Explorer

Файл Правка Вид Избранное Сервис Справка

Адрес: http://www.useit.com/

useit.com: usable information technology Search

useit.com: Jakob Nielsen's Website

Permanent Content

Alertbox
Jakob's column on Web usability

How Big is the Difference Between Websites? (Jan. 19)
The average difference in measured usability between competing websites is 68%. This is smaller than expected, but makes sense given the dynamics of design within individual industries.

Information Pollution (Jan. 5)
Web Design Mistakes (Dec. 22)
Confirmation Email (Dec. 8)

All Alertbox columns from 1995 to 2004

News

Usability Week 2004 Conference
- USA: Orlando, San Francisco
- Europe: London, Copenhagen
- Australia: Melbourne

Intensive usability immersion in my 3-day camp on usability in practice.
First presentation of new research on what design guidelines are most important for Web users' behavior in 2004.

Design Competition

Best government intranets: call for entries.

Готово Internet

а

Competitive Testing of Website Usability (Jakob Nielsen's Alertbox) - Microsoft Internet Explorer

Файл Правка Вид Избранное Сервис Справка

Адрес: http://www.useit.com/alertbox/20040119.html

useit.com → Alertbox → Jan. 2004 Difference between websites | Search

Jakob Nielsen's Alertbox, January 19, 2004:

How Big is the Difference Between Websites?

Summary:
The average difference in measured usability between competing websites is 68%. This is smaller than expected, but makes sense given the dynamics of design within individual industries.

I recently read through a huge pile of reports on usability studies. My main goal was to compute statistics about usability project outcomes for my three-day camp on usability in practice to help frame participants' expectations for the studies they'll run when they get back home. One of those statistics is likely of wider interest, however, even for readers who don't run usability projects.

Competitive testing is a special type of usability study that compares designs from multiple companies in the same industry. Across the studies I analyzed, the average **difference in measured usability was 68%** when comparing two competing companies.

(Because the raw data was in the form of ratios, I computed the average using the geometric mean. One data point, for example, might represent a comparison of two online brokers, where buying shares took users two minutes on one site, and three minutes on the other. The ratio between the two sites would be 3/2 = 1.50, corresponding to a difference of 50% in measured

Готово Internet

б

Рис. 6.10. www.useit.com: аннотации в действии

Дополнительно

Аннотации особенно полезны на странице, где находится список ссылок на разные материалы по схожей тематике. По заголовку не всегда можно догадаться, чем похожие статьи отличаются друг от друга. Хорошо составленные аннотации помогут пользователю сделать правильный выбор и перейти именно к тому материалу, который ему нужен.

- Текст можно дополнить иллюстрациями, схемами, графиками, диаграммами и другими графическими элементами. Но делать это стоит только в том случае, если эти изображения действительно несут полезную для пользователя информацию, а не являются просто красивыми картинками.

  Например, статью о выступлении ректора института на научной конференции можно снабдить фотографией, изображающей ректора именно во время этого выступления. Но не нужно сопровождать такую статью «просто подходящей картинкой» — символическим изображением атома, знаком интеграла и т. д.

Примечание

Не забудьте про средства оформления, о которых говорилось ранее. По необходимости применяйте гиперссылки внутри текста, выделения `strong` и `em`, выделения цветом и т. д. Разумеется, здесь важно не переборщить: однообразный текст скучен, но чересчур разнообразный тоже вряд ли будут читать. Хороший вкус и здравый смысл помогут вам найти золотую середину.

6.1.11. Вы ничего не забыли? Еще несколько распространенных ошибок

> Наступать на грабли — больно, а на детские грабли — еще и очень обидно.
>
> *Анекдот*

Существует несколько мелких текстовых элементов, ошибки в размещении которых стали практически традиционными.

Время и дата

Внимание!

Не помещайте на веб-страницах текущую дату и текущее время. Помещайте вместо этого дату последнего обновления страницы.

Текущая дата и особенно точное время — это совершенно лишняя информация, размещение которой на веб-странице не приносит пользователю никакой пользы. В самом деле, представьте, что кассир в магазине, прежде чем выдать вам чек, сообщает, что сегодня 7 февраля.

Пользователь идет на ваш сайт не затем, чтобы узнать, какое сегодня число и сколько сейчас времени. Пожалуйста, не забывайте об этом.

Вместе с тем, на странице рекомендуется указывать другую информацию о времени, а именно — *дату последнего обновления* этой страницы. Если страница обновляется редко или вообще не обновляется — достаточно указать день, в который она изменялась последний раз. Эта информация, в отличие от текущей даты, действительно может оказаться важной для пользователя.

Если сайт обновляется очень часто (например, сайт новостей) — указывайте для каждого материала не только день, но и время его размещения на сайте с точностью до минуты.

纸 Дополнительно

Некоторые разработчики считают, что текущая дата полезна, если пользователь сохраняет страницу на жестком диске, распечатывает или делает снимок экрана. В результате он всегда сможет сказать, в какой день была «зафиксирована» эта страница.

Это неверный довод. Если страница не обновлялась, например, в течение недели — то какая разница, в какой из дней этой недели она сохранена? Дата последнего обновления играет гораздо более важную роль.

Впрочем, на сайтах новостей и другой оперативной информации дата последнего обновления обычно совпадает с текущей. Но при этом важно не забывать, что размещать эту дату стоит не потому, что она сегодняшняя, а потому, что она соответствует последнему обновлению страницы.

Контактная информация

Иногда пользователям важно знать, кто разработал сайт и отвечает за его поддержку и наполнение. Не держите это в тайне. Размещайте на каждой странице знак охраны авторского права, в состав которого входит ссылка на ваш электронный адрес или на страницу с контактной информацией, например:

© Студия «ABC», разработка сайта, оформление, 2004

Бывает так, что из-за недостаточно ясной контактной информации к разработчику сайта приходят письма от пользователей, связанные с деятельностью компании, для которой сделан сайт. И наоборот, на электронный адрес компании приходят жалобы на технические недостатки в работе сайта.

Сделайте так, чтобы пользователи ясно понимали, кто за что отвечает. Создайте на сайте отдельную страницу «Контакты». Напишите прямо на этой

странице, к кому и по каким вопросам могут обращаться пользователи. Предусмотрите наиболее частые вопросы и поместите их на первом месте, например:

Ошибки и задержки в доставке заказов

Служба доставки: delivery@dostal.ru

Проблемы, связанные с оплатой заказа

Бухгалтерия: accounts@dostal.ru

У вас коммерческие предложения?

Зам. коммерческого директора В. Пупкин: pupkin@dostal.ru

Проблемы в работе этого сайта?

Веб-мастер: webmaster@dostal.ru

По всем остальным вопросам

Информационный отдел: info@dostal.ru

Дополнительно

Не скрывайте свой электронный адрес. Начинающие разработчики, опасаясь спама, «прячут» свой адрес — например, заключая его в изображение или применяя сложные сценарии JavaScript. Это любительские решения; они только затрудняют связь с пользователями сайта и очень незначительно влияют на объемы нежелательных сообщений. В настоящее время существуют хорошие программы-фильтры, которые устанавливаются на сервере электронной почты и борются со спамом гораздо эффективнее.

Under construction

Еще одна любительская традиция — сообщения о том, что сайт находится в процессе разработки. Это дурной тон, и таких сообщений следует избегать. Хороший сайт *постоянно* находится в процессе разработки и переработки, и это не мешает ему предоставлять пользователям нужную информацию и службы.

Сообщения типа «Under construction» фактически говорят пользователю: «Посмотрите, чего нет у нас на сайте». Но пользователь-то хочет получить то, что на сайте есть!

В самом деле, не надо сообщать пользователю, сколько всего у вас еще не готово. Наоборот, предоставляйте ему все то, что уже готово.

Например, на сайте «Озеро Селигер» (рис. 6.11) ссылка **Линкография** ведет на страницу, где пользователя извещают: «Извините, данная страница находится в разработке». Но зачем тогда вообще нужна эта ссылка? Гораздо лучше вообще убрать ее из панели навигации до тех пор, пока соответствующий раздел не заработает.

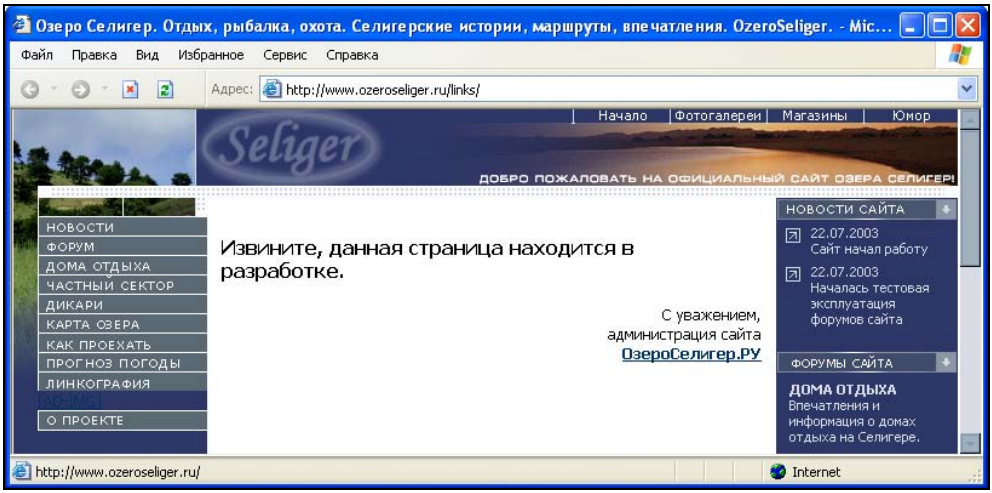


Рис. 6.11. www.ozeroseliger.ru: пользователь обманывается, перейдя по ссылке **Линкография**

6.2. Не забывайте про роботов

Как оптимизировать текстовое содержание сайта для поисковых систем

Бороться и искать, найти и не сдаваться.

В. Каверин. Два капитана

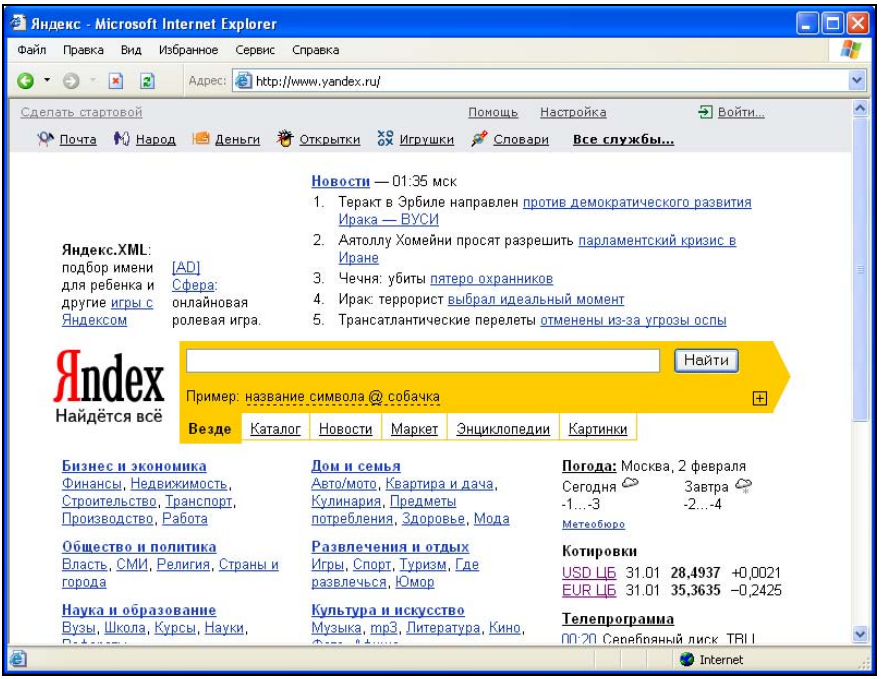
6.2.1. Облик Искателей

Общая информация о поисковых системах

— Ваше правительство приобретает облик нашего, это замечательно, — добавил я, — но наше начинает походить на ваше, вот что ужасно!
— Лучше уж нам походить друг на друга, чем уничтожать, — сказал Иван.

Р. Бах. Единственная

Поисковые системы играют огромную роль в современной Сети. Широко известны Google, AltaVista, Yahoo, Яндекс и десятки других систем. Их интерфейс со временем становится все ближе друг к другу, и можно выделить несколько стандартных элементов поисковой системы (рис. 6.12).



а

б

Рис. 6.12. Интерфейс различных поисковых систем очень похож

- **Стандартная форма поиска** — состоит из текстовой строки ввода и кнопки, по нажатию которой запрос пользователя отправляется к поисковой системе.
- **Страница расширенного поиска**, где пользователь может указать особые параметры запроса, например:
  - искать ли просто документ, содержащий перечисленные слова, или точно то сочетание слов, которое было введено в строке поиска;
  - искать страницы, изменившиеся с определенной даты;
  - искать только на определенных сайтах или, наоборот, исключить из поиска определенные сайты.
- **Категории поиска** — например, у русскоязычной службы Google четыре категории: **Веб** (Web), **Картинки** (Images), **Группы** (Groups), **Каталог** (Directory). У Яндекса на момент написания этой книги пять категорий: **Везде**, **Каталог**, **Новости**, **Маркет**, **Энциклопедии**, **Картинки**. По умолчанию поисковые системы ищут во всей Всемирной Паутине, но благодаря категориям пользователь может искать, например, только среди размещенных в Сети изображений или в телеконференциях Usenet.
- **Тематические каталоги** — системы классифицируют сайты по тематике и предоставляют пользователю возможность ограничить свой поиск определенной тематикой. Например, если вы ищете с помощью Яндекса страницы, посвященные поэту Александру Блоку, вам имеет смысл искать только в каталоге «Культура и искусство» или даже в его подкаталоге «Литература». В противном случае вам обеспечено много посторонней информации о строительных и прочих блоках, а также о блок-схемах.

Примечание

Кроме того, многие поисковые системы позволяют настроить свой интерфейс для конкретного пользователя, а также предоставляют сопутствующие службы — например, новости, прогноз погоды, биржевые котировки, ящик электронной почты.

Когда вы вводите запрос и нажимаете на кнопку отправки, поисковая система, конечно, не начинает спешно обыскивать все сайты Всемирной Паутины. Она обращается к своим собственным базам данных, где уже собрана информация о различных сайтах.

Базы данных поисковых систем обычно пополняются двумя независимыми способами:

- специальные программы-роботы постоянно «ходят» по Интернету и просматривают все обнаруженные страницы. Содержание этих страниц анализируется, и результаты этого анализа попадают в базу данных. Этот процесс называется *индексацией*;

- владелец сайта может зарегистрировать свой сайт в поисковой системе. При этом поисковая система специально проиндексирует сайт, и владелец обычно может добавить в базу описание сайта и данные, которые сама система может определить неточно (например, отнести сайт к тому или иному тематическому каталогу).

Со временем поисковые системы становятся все совершеннее, но все равно нередки случаи, когда пользователь не может найти то, что ему нужно, даже если эта информация есть в Интернете. Далее будет рассказано о том, что может сделать разработчик сайта для того, чтобы улучшить «находимость» своего сайта.

6.2.2. Разметка превыше всего

Грамотно употребляйте структурную разметку

> При необоснованно суженном заголовке в тексте окажется много такого, чего, судя по заголовку, в нем быть не должно. При неосновательно расширенном заголовке нельзя будет найти в тексте того, что заголовок обещает. В первом случае можно пропустить при выборочном чтении ценный материал, во втором — зря потратить время.
>
> *А. Мильчин, Л. Чельцова. Справочник издателя и автора*

Вы хотите, чтобы ваш сайт попадал в результаты поиска всякий раз, когда пользователь ищет нечто, что ваш сайт может ему предоставить? Вы хотите, чтобы ваш сайт оказывался не просто среди результатов поиска, а на одной из первых позиций?

Сразу оговоримся: абсолютно верных рецептов добиться этого не существует. Но рекомендации, приведенные в этой главе, в большинстве случаев действительно способствуют лучшему «взаимопониманию» между вашим сайтом и поисковыми системами.

Прежде всего, грамотно употребляйте структурную разметку. Многие поисковые системы придают большее значение тому, что содержится в заглавии страницы (элемент `title`) и в ее заголовках (элементы `h2` — `h6`), а также в логических выделениях (в первую очередь `strong`). Пишите в заглавии и заголовках именно то, что находится на странице.

Посмотрите, например, на результаты запроса `Cascading Style Sheets` в Google (рис. 6.13). В первую очередь показана страница, заглавие которой в точности совпадает с запросом. Затем идут страницы, в заглавии которых есть это словосочетание вместе с другими словами. Затем — страницы, где

оно встречается в заголовках. Наконец, страницы, где `Cascading Style Sheets` просто встречается в тексте, находятся далеко не на первой странице поиска.

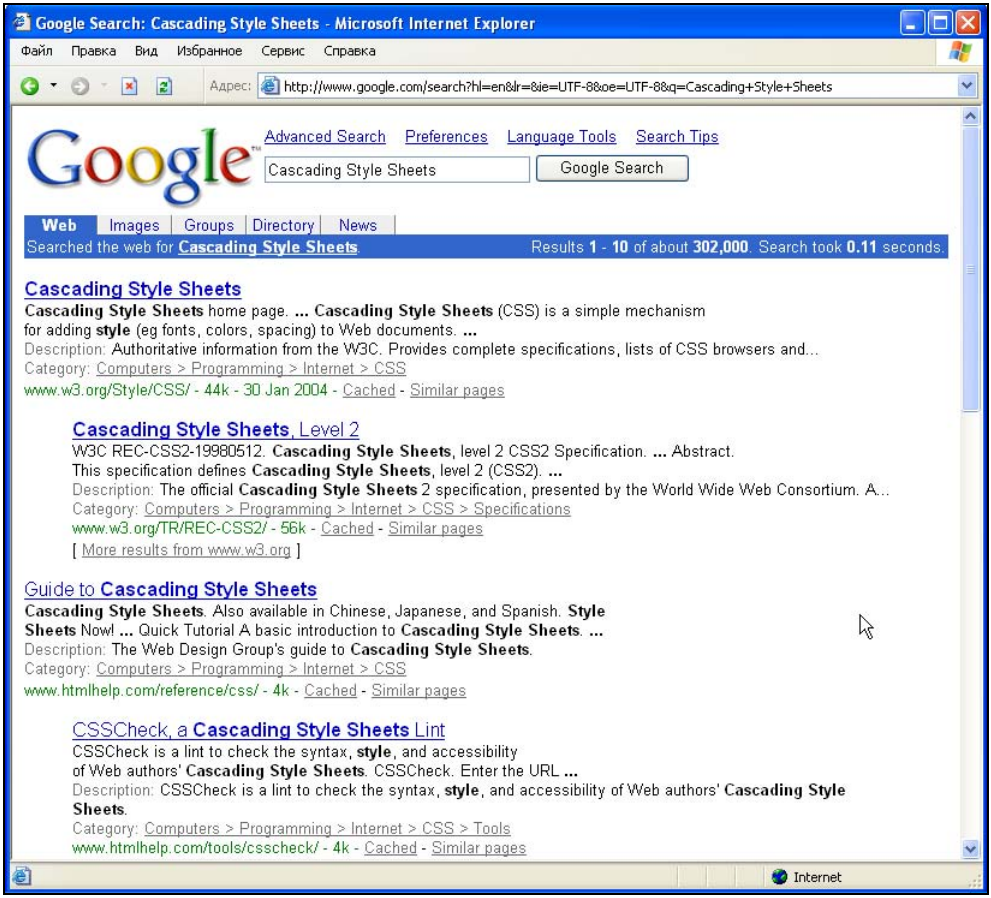


Рис. 6.13. По запросу `Cascading Style Sheets` показаны в первую очередь страницы, в заглавии которых есть это словосочетание

Таким образом, если запрос пользователя точно или почти точно совпадает с заглавием одной из ваших страниц — скорее всего, эта страница будет одной из первых в результатах поиска.

Примечание

Впрочем, не пишите в заглавии слишком много слов в надежде на то, что какое-нибудь из них совпадет с запросом. Поисковые системы обычно обрабатывают

только такое количество символов, которое помещается в верхней строке окна браузера в наиболее распространенных видеорежимах — т. е. не более 100 букв и других знаков.

Подумайте, на какие запросы должен «откликаться» ваш сайт в первую очередь. Предположим, вы разрабатываете сайт для фирмы, торгующей недвижимостью в Москве. Скорее всего, среди наиболее близких запросов будут следующие:

- купить квартиру в Москве;
- цены на жилье;
- продажа недвижимости;
- жилой комплекс;
- квартира в пределах Садового кольца;
- новая квартира в спальном районе.

Это далеко не исчерпывающий список. Разработчик должен тщательно проконсультироваться со специалистом по недвижимости, в результате чего возникнет гораздо более длинный список «подходящих» запросов.

Конечно, оптимизировать сайт абсолютно под все эти запросы не получится. Надо расположить запросы в порядке приоритета и обработать в первую очередь наиболее важные из них. Допустим, наиболее важными оказались именно вышеперечисленные запросы. Это надо учесть при разработке заглавия и заголовков, а также текстового содержания страницы — например, следующим образом:

- включить в заглавие (`title`) страницы текст: «Недвижимость, квартиры в Москве»;
- сделать словосочетания «Купить квартиру в Москве» и «Цены на жилье» заголовками отдельных блоков на главной странице;
- разместить на главной странице «рекламный» блок, анонсирующий новое специальное предложение фирмы. Дать этому блоку заголовок: «Новая квартира в Москве!»;
- в панели ссылок или в тексте страницы упомянуть словосочетания «в пределах Садового кольца» и «в спальном районе»;
- несколько раз на странице упомянуть словосочетание «жилой комплекс».

Обратите внимание: это только приблизительный вариант. Автор не утверждает, что все фирмы, торгующие недвижимостью, должны разрабатывать свои сайты именно таким образом. Это всего лишь пример, иллюстрирующий принцип оптимизации текстового содержания сайта для поисковых систем.

Вообще, поиск «подходящих» (или, как еще говорят, «релевантных») запросов и составление текста в соответствии с этими запросами — серьезная творческая задача. Чтобы справиться с ней на должном уровне, необходимо

сотрудничество специалистов по информационной архитектуре и редакторов, хорошо владеющих русским языком.

Примечание

Некоторые разработчики считают так: если очень много раз повторить на странице одно и то же слово или словосочетание, поисковые системы будут выдавать эту страницу среди первых результатов поиска, когда запрос пользователя совпадает с этим словом или словосочетанием.

В целом это справедливо. Но при этом страница получается «заточенной» под один конкретный запрос и не «откликается» на похожие по смыслу запросы, сформулированные по-другому. Кроме того, пользователю вряд ли доставит удовольствие видеть страницу, половина слов на которой — одно и то же слово.

Оптимизируя страницу для поисковой системы, не забывайте о том, что конечная цель — не поисковая система, а живой пользователь. Делайте страницы прежде всего такими, чтобы они удовлетворяли пользователя. Оптимизация для поисковых систем — важный, но всего лишь вспомогательный шаг в этом процессе.

6.2.3. Чтоб язык не заплетался

Учитывайте особенности русского языка

> Понимаете ли, я живу в такое время, когда библиотек уже нет, одна машинная память. Это, конечно, гораздо удобнее, но если нужно откопать что-нибудь древнее, начинаются всякие казусы. Я запрашиваю о Пастернаке, а мне выдают какую-то чушь про укроп, сельдерей, словом, полный набор для супа. С Блоком еще хуже. Миллионы всяких схем электронных блоков.
>
> *И. Варшавский. Любовь и время*

Поисковые системы пока еще плохо справляются с задачами, для которых требуется знание русского языка. Их несовершенство проявляется прежде всего в следующих направлениях:

- **морфология** — слова в русском языке склоняются, спрягаются, изменяются по лицам и числам и т. д. К сожалению, очень немногие поисковые системы учитывают эту особенность. Например, если ввести запрос `Пушкин`, система не предъявит сайты, на которых нет слова «Пушкин», но есть слова «Пушкину» или «о Пушкине»;

Дополнительно

Яндекс умеет искать страницы с учетом русской морфологии. Но это достоинство влечет за собой ряд недостатков: например, если в строке поиска присут-

ствует слово «ворóна», то Яндекс найдет и те страницы, на которых есть слово «вóрон» или словосочетание «воронóй конь». А запрос «сахар» приведет вас к множеству страниц о пустыне Сахаре.

- **омонимы** — одно и то же слово может иметь разные значения. Ранее уже приводился пример со словом «блок»; можно также вспомнить слова «коса» и «ключ». Когда пользователь вводит в строку запроса слово `блок`, поисковая система действительно никак не может догадаться, имеется ли в виду русский поэт или строительный элемент. Поэтому пользователю приходится уточнять запрос, дополнять его, ограничивать поиск — и все равно он получает множество ненужных результатов;

Примечание

Отдельная проблема — с буквой «Ё». Поисковые системы не отличают «лён» от «лен» (феодальное владение), «съем» от «съём» и т. д. Историки, разыскивающие в Сети информацию о средневековых ленах, видят в списке результатов поиска гораздо больше страниц о льняных изделиях.

- **синонимы** — иногда, наоборот, пользователь не находит нужной страницы потому, что запрашивает не те слова, которые на ней находятся, а их синонимы. Например, запрос `Фауна Африки` не приведет к странице, озаглавленной «Животные Сахары» или «Млекопитающие Черного континента». Пользователю приходится перебирать разные варианты запросов, чтобы найти страницы на одну и ту же тему;
- **семантический анализ** — если вы введете в строку запроса слово `Пушкин` — поисковая система выдаст страницы, на которых встречается это слово. Но если вы захотите найти стихотворения Пушкина — вы можете потерпеть неудачу. Запрос `стихотворения Пушкина` приводит всего лишь к страницам, на которых есть слова «стихотворение» и «Пушкин», при том, что самих стихотворений там может и не быть. И, наоборот, если на странице действительно находится стихотворение Пушкина, но нет самого слова «стихотворение» — эта страница не будет включена в результаты запроса.

Дополнительно

Поисковые системы анализируют текст, но они не понимают смысла этого текста. Поисковая система не сможет предъявить вам, например, те и только те страницы, на которых находятся прозаические литературные произведения. Или, если вы ищете все, что относится к первому космонавту Ю. Гагарину, поисковая система не выдаст вам знаменитой песни «Знаете, каким он парнем был?». (Именно из этой песни строчка «Он сказал: „Поехали!“, он взмахнул рукой».) Песня написана о Гагарине, однако в тексте нет ни его имени, ни фамилии. Поэтому, если на странице содержится только текст этой песни, без специальных комментариев, поисковая система его проигнорирует.

Аналогично, если вы ищете стихи Пушкина, посвященные его няне Арине Родионовне — ни одна поисковая система не предоставит вам «Зимний вечер» («…Что же ты, моя старушка, приумолкла у окна?») — если, конечно, на странице с текстом этого стихотворения нет особого упоминания о пушкинской няне.

Полное преодоление этих проблем — дело пока еще отдаленного будущего, но разработчик сайта может их значительно уменьшить.

- Используйте наиболее важные, ключевые слова и словосочетания в нейтральной форме (существительные — в именительном падеже, глаголы — в неопределенной форме и т. д.). Например, не пишите в заглавии: «Продажа компьютеров», лучше напишите «Компьютеры — Продажа». Пишите не «Рефераты по молекулярной биологии», а «Молекулярная биология. Рефераты».

  Дополнительно

  Из этого правила есть исключения. Например, пользователи обычно ищут не «Пушкин. Сказки», а «сказки Пушкина». Если вы хотите, чтобы ваша страница «откликалась» на такие запросы, разместите на ней текст «сказки Пушкина» — но не вместо нейтральной формы, а вместе с ней. Например, в заглавии (`title`): «Сказки А. С. Пушкина», а в заголовке первого уровня (`h1`) — «А. С. Пушкин. Сказки».

- Указывайте в тексте страницы, какой именно тип текстового наполнения там присутствует. Например, если на странице помещен текст стихотворения, найдите возможность поместить на ней также слова «Стихи», «Стихотворение», «Поэзия». (Это можно сделать в основной панели навигации, в «хлебных крошках», в подзаголовке, в комментарии к стихотворению или в специальном блоке.) Если на странице статья или обзор — позаботьтесь о том, чтобы где-нибудь на этой же странице находилось само слово «статья» или «обзор».

  Дополнительно

  Когда-то разработчики сайтов широко использовали элемент `meta`, в котором можно было представить краткое описание страницы и перечислить ключевые слова. К сожалению, некоторые создатели страниц злоупотребляли этим тэгом и включали в него слова, не имеющие отношения к содержанию страницы, но «откликающиеся» на наиболее популярные запросы. Поэтому поисковые системы перестали ориентироваться на `meta` и придают значение в первую очередь действительному содержанию страницы.

- Сделайте так, чтобы вместе с ключевыми словами на странице располагались также их распространенные синонимы. Например, упоминайте не только «сотовые телефоны», но и «мобильные телефоны», не только

«фауну», но и «животных», не только «квартиры», но и «жилье», и «недвижимость». Чтобы справиться с этой задачей, потребуется словарь синонимов и богатый опыт литературного редактирования.

Хороший способ обеспечить «находимость» ресурса — это его уникальность и определенность тематики. Например, если пользователь ищет информацию об Олимпийских играх 2004 года — поисковая система первым делом предъявит ему сайты, посвященные именно этим конкретным Играм, затем — сайты про Олимпийские игры вообще, и только потом — сайты на общую спортивную тематику, в содержании которых также есть данные про Олимпийские игры.

Примечание

Уникальное название фирмы или организации, которой посвящен сайт, — еще один фактор, способствующий его «находимости». Если организация называется «Алые паруса» или «Лабиринт» — скорее всего, даже если пользователь ищет сайт именно этой организации, ему придется пробираться сквозь тысячи ссылок, ведущих на совершенно посторонние страницы. Наоборот, хорошие примеры — компания «ABBYY» или музыкальный коллектив «Башня Rowan». Если название не похоже вообще ни на что — пользователь не получит посторонних результатов.

6.2.4. Хитрости полезные и вредные

Как еще повысить «находимость» вашего сайта

> И будут его искать — и, конечно, никогда не найдут. Даже если планету заселят, множество людей будет жить на ней — все равно никто никогда не найдет доктора Павлыша, шатена, рост 183, глаза голубые...
>
> *К. Булычев. Великий дух и беглецы*

Многие страницы на крупных, профессиональных сайтах не хранятся на сервере постоянно, а генерируются серверной программой после того, как пользователь ввел адрес в строке браузера или щелкнул по ссылке. Такое поведение особенно характерно для страниц, которые часто обновляются.

Однако поисковые системы не могут следить за каждым обновлением каждой страницы. Из-за этого происходят распространенные ошибки: система выдает адрес страницы в результатах поиска, пользователь переходит на эту страницу, но не обнаруживает на ней искомого содержания. С тех пор как поисковая система последний раз индексировала эту страницу, страница изменилась, и старое содержание пропало (рис. 6.14).

Помощь Настройка

Яndex
Найдётся всё

удобство и доступность сайтов Найти

в найденном расширенный поиск

Везде Каталог Новости Маркет Энциклопедии Картинки

Результат поиска: страниц — **4840**, сайтов — не менее **1835**
Статистика слов: удобство — 2074749, и — 1944521412, доступность — 1028550, сайтов — 177218400
Запросов за месяц: удобство — 542, доступность — 677, сайтов — 1418065

1. Молот ведьм. Краткий курс
удобстве и **доступности сайта**.
www.eholot.net/news/fido7.ru.web.design/18794/ (11 КБ) 13.12.2003 — совпадение фразы
Найденные слова · Похожие документы · Еще с сайта (2)

2. Версия статьи для печати / Доступность текста в сети - определяем размер шрифта ...
Так значит ли это, что вам, как дизайнеру, озабоченному проблемой **удобства** и **доступности сайта**, необходимо использовать шрифт большего размера?
... например, пунктов) для задания размеров, т. к. это одна из основных проблем **удобства** использования **сайтов**, с которой я сталкиваюсь при тестировании ...
computerlibrary.info/view/article43/print/ (47 КБ) 08.06.2004 — совпадение фразы
Найденные слова · Похожие документы · Еще с сайта (4)

3. Информационный сервер города Невинномысска
... одна из возможных схем, в соответствии с которой пользователь может конструктивно сообщать разработчику о недостатках в **удобстве** и **доступности сайта**

а

Главная> Форумы> Дизайн WEB сайтов> [] English

eholot.net

Искать

[Список сообщений]

Re: Yandex - почта **Dima Sadakov** "Aleksey Sosnovikov"
Re: safari **Serhiy Prosyanyk** "Aleksey Sosnovikov"
http://nick3d.nm.ru **All** Nick Bogdanov
Re: safari **Vitaly Harisov** Ruslan Kosolapov
Хостинг **All** Vladimir Zolotarev
Safari **Vitaly Harisov** "Rostislav Chebykin"
Re: safari **Sergey A. Shashkov** Vitaly Harisov
сайт федерального гос. учреждения **All** Alexander Shulgin
Re: сайт федерального гос. учреждения **Alexander Shulgin** "Alex Vader"
Re: Yandex - почта **Vasily Tchekalkin** Victor Kropp
Re: Yandex - почта **Victor Kropp** Victor Kropp
Re: Test **Nick Loboda** Andy Antonov
Re: Yandex - почта **Igor Turashev** Victor Kropp
Yandex - почта **Victor Kropp** Igor Turashev
Re: Yandex - почта **Victor Kropp** Vasily Tchekalkin
Re: Test **Victor Novoselov** Andy Antonov
Test **Andy Antonov** Victor Novoselov

б

Рис. 6.14. По запросу «удобство и доступность сайтов» поисковая система «Яндекс» первым номером предлагает страницу, где размещена статья «Молот ведьм», написанная автором этой книги (*а*). Однако если перейти по ссылке — оказывается, что статьи по этому адресу уже давно нет (*б*)

Если какое-то содержание находится на вашем сайте постоянно — не меняйте адрес страницы с этим содержанием. Например, на сайте новостей не стоит сегодняшние новости располагать по адресу **www.mynews.ru/today**, а вчерашние — по адресу **www.mynews.ru/yesterday**. Гораздо лучше придерживаться такой схемы: **www.mynews.ru/2004/02/01**, **www.mynews.ru/2004/02/02** и т. д. При этом новости за 1 февраля 2004 года будут доступны по одному и тому же адресу и в этот самый день, и через год, и через 10 лет.

Если же какое-то содержание меняется настолько часто, что никакая поисковая система за ним не уследит (например, главная страница на сайте новостей) — настройте веб-сервер таким образом, чтобы вообще запретить поисковым системам индексировать непостоянные элементы на этой странице. (За консультацией в этом вопросе обратитесь к администратору веб-сервера, где размещается ваш сайт.)

Примечание

Если заключить текст в изображение, то, кроме уже перечисленных проблем с удобством для пользователя, это приведет также к тому, что поисковая система не сможет проиндексировать такой «текст».

«Находимость» сайта также повысится, если на этот сайт будет указывать много ссылок с других сайтов. Такие ссылки накапливаются всегда, если сайт предоставляет действительно нужное и полезное содержание. В конце концов сайт становится настолько известным, что для его обнаружения даже не требуется поисковой системы — его адрес и так все знают, как, например, **www.yandex.ru** или **www.anekdot.ru**.

Дополнительно

В самом деле, многие поисковые системы часто ориентируются на так называемый «индекс цитирования», показывающий, сколько ссылок с других сайтов ведет на данный ресурс. Чем больше ссылок — тем раньше в списке результатов поиска отображается сайт.

Позаботьтесь о том, чтобы ваш сайт был занесен в различные каталоги ресурсов и списки ссылок на сайтах по дружественной тематике. Предложите дружественным сайтам обменяться ссылками. В этом случае, если пользователь попал хотя бы на один из сайтов по вашей тематике, то весьма вероятно, что он доберется и до вашего.

Хороший пример подают сами поисковые системы: если результаты поиска не устроили пользователя, они предлагают ему поискать также в других системах (рис. 6.15).

Примечание

Иногда разработчики применяют «запрещенные приемы»: например, помещают на страницу много текста, который не имеет отношения к ее тематике, но отвечает популярным запросам. Бывает, что часть страницы занимает текст, который не отображается на экране (например, он скрыт с помощью CSS) или отображается белым цветом на белом фоне.

Это любительские приемы, которые на самом деле нисколько не способствуют продвижению страницы. Наоборот, они только отпугивают пользователей, которые все-таки попали на эту страницу.

Предоставляйте нужное, полезное и интересное содержание, грамотно его размечайте и оформляйте — это наилучший способ добиться первых мест в результатах поисковых систем.

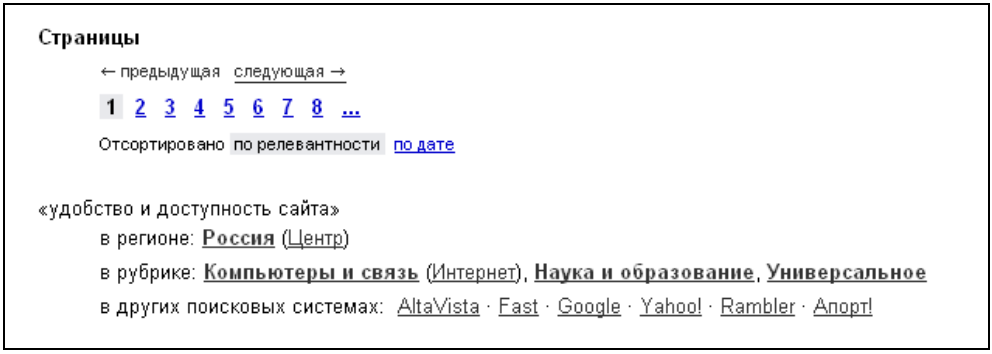


Рис. 6.15. Яндекс предлагает переадресовать запрос шести другим популярным поисковым системам

```
<?xml version= 1.0
<!DOCTYPE html PUBL
  "-//W3C//DTD XHTM
  "http://www.w3.or
<html xmlns="http://
  lang="ru" xml:lan
<!-- Редактировал:
<head>
  <title>Евгений Он
  <link rel="styles
    href="index.css
</head>
```

Глава 7

Пользователь всегда прав

Удобство, доступность, эффективность и привлекательность текстового содержания сайта

Ранним утром было любо-дорого посмотреть на покос Тита Горбатого, на котором старик управлялся своею одною семьей.

Д. Н. Мамин-Сибиряк. Три конца

Подчас работа была тяжела: орудия были приспособлены для людей, а не для животных, и было очень неудобно, что ни одно животное не могло пользоваться орудиями, которые требовали стояния на задних ногах.

Дж. Оруэлл. Скотный двор

Считается, что хороший сайт должен удовлетворять по крайней мере четырем критериям: быть удобным, доступным, эффективным и привлекательным для пользователей. В то же время, смысл каждого из этих понятий и разница между ними не всегда определяется достаточно ясно.

В действительности между этими четырьмя критериями нет абсолютно четких границ. Отдельные аспекты разработки сайта могут в равной мере относиться и к удобству, и к доступности; и к эффективности, и к привлекательности. Это не четыре разные свойства сайта, а скорее четыре разные позиции, с которых можно рассматривать сайт.

Опишем каждую из этих позиций более подробно:

- **удобство (usability)** — это, другими словами, простота и легкость, с которой пользователь работает с сайтом. Сайт удобен, если пользователю легко выполнять на нем те задачи, ради которых он посетил его. Сайт удобен, если пользователь не встречает на нем того, что было бы для него сложно или непонятно. На удобном сайте пользователь быстро находит нужные ресурсы, свободно владеет интерфейсом, без затруднений заполняет нужные формы и т. д.

Примечание

С одной стороны, удобство — субъективная категория. Вполне возможны ситуации, когда что-то удобно для одних пользователей и неудобно для других. Но, с другой стороны, за время существования Всемирной Паутины выработано немало общих принципов, относящихся к удобству сайтов (см., например, [13] и [14]).

Даже если пользователь не чувствует затруднений при работе с сайтом, это еще не значит, что удобство сайта находится на идеальном уровне. Даже популярные сайты, сделанные настоящими профессионалами, иногда перерабатываются для того, чтобы стать еще удобнее. Удобство сайта — не абсолютная величина, а относительная; ее можно оценивать, сравнивая разные технические решения;

Дополнительно

Существуют и несколько объективных параметров измерения удобства. Прежде всего это время, за которое пользователь выполняет свою задачу, и количество операций, которые он при этом осуществляет. Например, если в одном интернет-магазине пользователь оформил покупку товара за 15 минут, совершив при этом 12 переходов по ссылкам, а в другом ему потребовалось 10 минут и 7 переходов — значит, скорее всего, для данного пользователя и данной задачи второй сайт более удобен, чем первый.

Это не абсолютные критерии; кроме времени и количества действий, приходится учитывать множество факторов, которые не поддаются точному измерению. Тем не менее, определение времени и количества действий можно положить в основу исследования того, удобен ли сайт.

- **доступность (accessibility)** — это показатель того, насколько сайт универсален, насколько хорошо он подходит для различных категорий пользователей, применяющих разные устройства для просмотра сайтов. Существуют разные операционные системы и разные браузеры, у пользователей разные настройки видеорежима; кроме того, они могут просматривать сайты с помощью портативных устройств или экзотических приспособлений (вроде текстовых или речевых браузеров). Если последние — действительно экзотика, то портативные компьютеры приобретают в нашей стране все большее распространение, и накопилось уже немало откликов от недовольных пользователей «наладонников», которые не смогли работать с любимыми сайтами так же легко, как на обычном компьютере;

Примечание

Разумеется, невозможно сделать сайт, который будет идеально работать на любом устройстве и в любых условиях, в которых пользователь пожелает его

просмотреть. Но это не значит, что разработчикам вовсе не следует заботиться о доступности. Нужно установить определенный уровень доступности, исходя из того, кто и зачем заходит на сайт, и твердо придерживаться этого уровня.

Дополнительно

На сайте Консорциума W3C есть раздел, посвященный различным аспектам доступности сайтов: **www.w3.org/WAI**. Кроме того, Консорциум выпустил сборник рекомендаций по доступности содержания сайтов, который находится по адресу **www.w3.org/TR/WCAG10**. К этим рекомендациям прилагается простой в использовании контрольный лист, позволяющий легко проверить, какому уровню доступности удовлетворяют ваши страницы.

- **эффективность** — это показатель того, насколько успешно выполняются задачи сайта. Иногда об эффективности говорят, как о «подразделе» удобства, иногда вообще считают, что удобство и эффективность — это одно и то же. С точки зрения некоторых специалистов, к эффективности следует отнести все то, что поддается однозначному измерению — например, время, за которое пользователь выполняет свои задачи.

  Но, в конце концов, важно не столько само понятие, сколько его смысл. В этой книге «эффективность» рассматривается как одна из составляющих частей удобства, хотя в других источниках вы можете встретить иное разделение;

- **привлекательность** — характеристика, показывающая отношение пользователей к сайту. С одной стороны, ее можно рассматривать как субъективную — нравится или не нравится сайт, хочется ли пользователю его посещать. Но, с другой стороны, привлекательность можно оценивать и объективно: например, из двух конкурирующих сайтов более привлекателен тот, который пользователи чаще выбирают для посещения. (Речь идет только о таких пользователях, которые посещают один из сайтов не потому, что не знают о существовании другого, а потому, что этот сайт устраивает их больше.)

  Привлекательность сайта во многом зависит от его удобства и доступности. Пользователи более охотно будут посещать тот сайт, с которым легче и удобнее работать. Но привлекательность в немалой степени зависит и от содержания сайта. Разумеется, пользователи будут посещать сайт только в том случае, если он предоставляет важное, интересное для них содержание и полезные службы.

В этой главе рассказывается об удобстве, доступности, эффективности и привлекательности текстового содержания сайта. Для того чтобы не только текст, но и сайт в целом обладал этими качествами, требуется учитывать много вещей, выходящих за рамки этой главы. Многое из того, чего вы не найдете в этой книге, можно прочесть в [13], [14] и [15].

7.1. Чтоб никто не ушел обиженным

Основные принципы удобства текстового содержания

> — Дай вам Б-г никогда не оказаться на моем месте, — ответил Моше. — А книги Маймонида трудно назвать глупыми. Хотите — принесу почитать.
> — Спасибо, — равнодушно произнес врач, — но у каждого в мире своя задача. Кому Учение, а кому...
>
> *Я. Шехтер. Обманщик*

Пользователь заходит на сайт для того, чтобы выполнить какую-то конкретную практическую задачу: получить информацию, загрузить определенный файл, найти нужный ресурс, купить товар в интернет-магазине, оставить сообщение на форуме и т. д. Если что-то мешает пользователю выполнить эту задачу или отвлекает его — он остается недовольным. Значит, разрабатывая сайт, вы должны иметь в виду, как именно пользователь будет с ним обращаться, какие задачи он будет выполнять. Отсюда следуют основные принципы удобства, которые перечисляются далее.

7.1.1. Привычка — вторая натура

Учитывайте естественное поведение и привычки пользователей

> Напротив, меня всего пронизывает радостная мысль, что ныне я вполне сроднился с этой сладостной привычкой и не имею ни малейшего желания когда-либо расставаться с нею.
>
> *Э. Т. Гофман. Житейские воззрения кота Мурра*

Внимание!

Учитывайте естественное поведение пользователей. Вместо того чтобы навязывать пользователям какие-то определенные действия, сами подстраивайтесь под их режим работы.

Некоторые примеры уже приводились в *главе 6*. Скажем, кнопку **Найти!** лучше расположить справа от строки ввода, потому что взгляд пользователя движется слева направо. Пользователь ввел запрос — и автоматически утыкается взглядом в кнопку, которую ему следует нажать.

Другой пример связан с тем, что люди привыкли читать печатные издания, начиная с левого верхнего угла страницы. Поэтому левый верхний угол веб-страницы обычно воспринимается как ее «начало», именно в нем чаще всего располагается логотип или текстовое обозначение сайта (этот логотип или обозначение традиционно является ссылкой на главную страницу). «Хлебные крошки» также обычно размещаются вблизи от левого верхнего угла.

Примечание

На сайте Звуки.Ru логотип расположен справа (рис. 7.1). Это не вызывает никаких затруднений у постоянных посетителей сайта, но может ввести в заблуждение тех, кто попадает сюда впервые. Пользователь по привычке смотрит в левый верхний угол — и видит там «ДДТ». Может быть, это сайт, посвященный группе «ДДТ»? Посмотрев на панель навигации, которая находится непосредственно под изображением, пользователь убеждается в своем мнении: в самом деле, названия разделов вполне соответствуют сайту музыкальной группы. Но тогда остальное содержание страницы вызывает недоумение, и пользователь не сразу догадывается, что это музыкальный «портал», а не сайт одной группы.

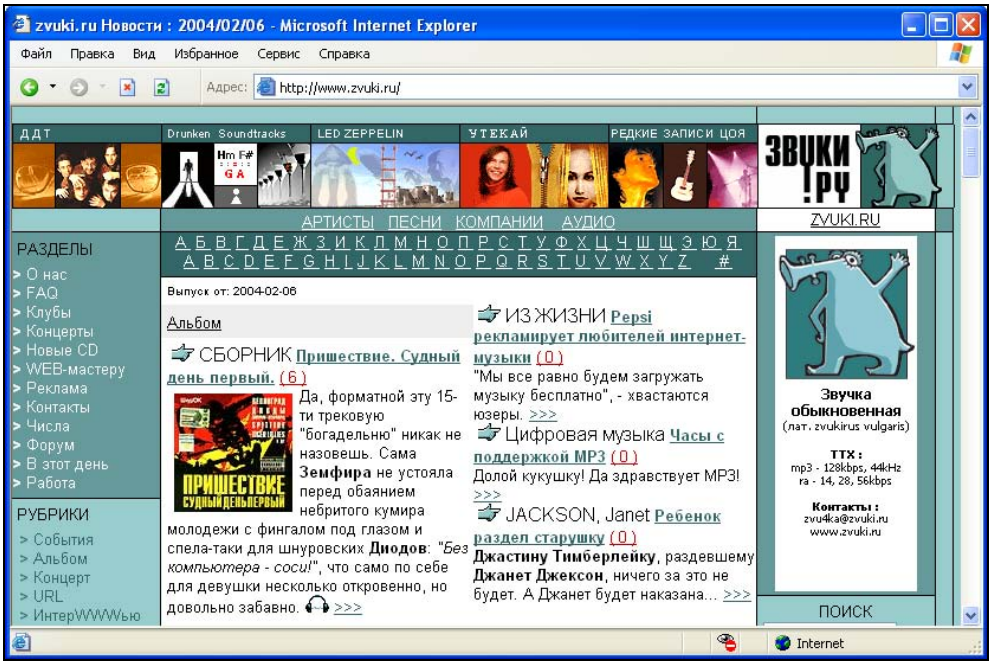


Рис. 7.1. www.zvuki.ru: главный логотип — справа. Тем не менее, легко принять за главный логотип иллюстрацию в верхнем левом углу

Внимание!

Учитывайте сложившиеся привычки и ожидания пользователей.

При разработке не только сайтов, но и любых других продуктов, с которыми пользователь должен взаимодействовать, важно учитывать, что у пользователей уже накоплен опыт работы с подобными вещами, сформировались определенные привычки и ожидания.

Дополнительно

Многие пользователи программного обеспечения часто предпочитают пользоваться более старыми и менее эффективными версиями программ — только потому, что они к ним привыкли. Они не торопятся осваивать новую, усовершенствованную программу до тех пор, пока не столкнутся с непреодолимыми трудностями.

Автор еще в 2003 году встречал пользователей, просматривающих веб-страницы с помощью Netscape 4. Это были обычные пользователи, не специалисты в области компьютерных технологий, они принадлежали в основном к старшему поколению и попросту не видели смысла обновлять свой браузер.

Пользователю комфортнее работать с тем, что для него привычно, — даже если непривычными средствами можно добиться более эффективных результатов.

Примечание

Даже разработчики сайтов, редактируя код в текстовом редакторе, нередко пользуются мышью там, где быстрее и эффективнее было бы обойтись без нее. Также разработчики знают, что слепой десятипальцевый метод набора позволяет работать существенно быстрее и продуктивнее — но немногие действительно осваивают этот метод.

В частности, пользователи привыкли к «стандартным» названиям определенных разделов сайта. Страница с контактной информацией чаще всего называется «Контакты», а схема иерархической структуры сайта — «Карта сайта». Если вы назовете страницу с контактной информацией «Наши координаты» — весьма вероятно, что пользователь задумается: «Это то же самое, что на других сайтах называется „Контакты“, или нет? Если то же самое, то почему оно называется по-другому? А если не то же самое, то чем оно отличается от обычного раздела „Контакты“, и стоит ли мне заходить на эту страницу?»

Если в тексте гиперссылок или просто в тексте на промежуточных страницах сайта упоминается главная страница, то лучше всего так и называть ее — «Главная страница», «На главную страницу» и т. д. Названия «Первая страница», «Начало», «Домашняя страница» и другие являются гораздо менее привычными.

Дополнительно

В английском языке главная страница сайта чаще всего называется «homepage». Дословно это переводится как «домашняя страница», однако такой термин так и не получил в русском языке точного значения. Поэтому лучше пользоваться словосочетанием «Главная страница».

Основная панель навигации чаще всего находится либо в левой части окна, под логотипом (в этом случае ссылки расположены друг под другом), либо в верхней части окна (ссылки расположены в одну строчку, которая чаще всего выключена в левый край или в красную строку). Пользователи привыкли обнаруживать панель навигации именно в этих местах. Поэтому, если для того нет каких-то специальных причин, не стоит размещать навигацию, например, в правой или центральной из трех колонок.

Примечание

Единообразие сайта, о котором упоминалось в *главе 6*, играет немалую роль в его удобстве. Пользователь ожидает единообразного внешнего вида, интерфейса и поведения веб-страниц. Например, плохо, если кнопка с надписью **Зарегистрироваться** на одних страницах будет обеспечивать регистрацию нового пользователя, а на других — вход на сайт для уже зарегистрированных пользователей.

Пользователю проще работать с сайтом, если этот сайт похож на большинство других, которые он посещал раньше. Именно поэтому популярные поисковые системы так похожи друг на друга по внешнему виду и поведению. То же самое можно сказать и про интернет-магазины.

Например, магазины и многие другие сайты предлагают пользователям зарегистрироваться. Однако форма регистрации на них составляется по-разному (рис. 7.2). В результате невнимательный пользователь может ввести свою фамилию в строку, предназначенную для имени, и наоборот. Еще печальнее, если он перепутает поля для телефона и кредитной карты.

Для новых пользователей

Для регистрации юридического лица **нажмите здесь**

* Имя

* Фамилия

* E-mail

* Пароль

* Повтор пароля

☑ Подписаться на новости о специальных предложениях и скидках Озона (2-3 письма в месяц)

☑ Подписаться на новинки основных разделов Озона (ежедневно)

Зарегистрироваться

а

Вход для новых пользователей

Если Вы еще не зарегистрированы в нашем магазине, то Вы можете это сделать сейчас: Обязательно **правильно** заполните поле **e-mail**, в противном случае мы не сможем с Вами связаться и выполнить Ваш заказ.

Фамилия

Имя

Отчество

e-mail

◉ муж. ○ жен.

Пароль

Повторите пароль

Страна

Россия

Зарегистрироваться

б

Рис. 7.2. В интернет-магазинах **www.ozon.ru** (*а*) и **www.bolero.ru** (*б*) поля в форме регистрации расположены в разном порядке

Дополнительно

Некоторым разработчикам может показаться странным, что пользователь способен перепутать поля, рядом с которыми недвусмысленно написано «Имя» и «Фамилия». Тем не менее, пользователи очень часто бывают невнимательны. Когда их просят объяснить, почему они поступили неправильно, они чаще всего говорят: «А я думал, что надо было так».

В самом деле, не ждите, что пользователи будут обращаться с вашим сайтом именно так, как с ним надо обращаться. Они будут действовать так, как, *по их мнению*, будет правильно. Чтобы это не приводило к затруднениям — не заставляйте пользователей осваивать новые способы работы с сайтом. Наоборот, исследуйте, как они уже работают с другими сайтами, и сделайте так, чтобы с вашим сайтом можно было работать таким же образом.

Не требуйте от пользователей переучиваться или, что еще хуже, переключаться в разные режимы работы. Последняя проблема проявляется не только на сайтах, но и в традиционном программном обеспечении. Например, чтобы ввести неразрывный пробел в Microsoft Word, требуется нажать сочетание клавиш <Ctrl>+<Shift>+<Пробел>, а в издательской системе Corel Ventura — <Ctrl>+<Пробел>. Тот, кто много работает с обеими программами, часто ошибается, нажимая в одной из них то сочетание клавиш, к которому привык в другой.

То же самое верно и для сайтов: если для выполнения одной и той же задачи на разных сайтах требуется совершать разные операции, пользователь будет часто путаться.

Примечание

Иногда опасаются, что, если соблюдать перечисленные рекомендации, все сайты станут одинаковыми, скучными и однообразными. Это напрасные опасения. Например, все автомобили устроены одинаково: сидение водителя — впереди, багажник — сзади, мотор — внизу, двери — по бокам и т. д. Тем не менее, это не мешает конструкторам создавать разные модели автомобилей, которые отличаются друг от друга не только техническими показателями, но и стилем, внешним видом. Или, например, при строительстве зданий должны соблюдаться все соответствующие ГОСТы и другие нормативные инструкции, обеспечивающие удобство проживания в этих зданиях — но при этом дома вовсе не получаются одинаковыми. Разработка внешнего вида автомобиля или здания — интересная творческая задача. Необходимость соблюдать общие стандарты не мешает появляться машинам и домам, которые не только успешно выполняют свои прямые функции, но и производят прекрасное эстетическое впечатление.

Это касается и сайтов: соблюдение требований удобства и доступности не мешает создавать весьма красивые и художественно самостоятельные сайты. Разумеется, художественные замыслы не должны противоречить удобству — но ведь и в автомобилестроении никто не размещает руль на крыше, даже если кажется, что это очень красиво и оригинально.

Возвращаясь к текстовому содержанию сайта, можно отметить, что наиболее привычен — а следовательно, наиболее удобен — тот текст, который отображается шрифтом, установленным в браузере по умолчанию. Поэтому не следует специально регулировать гарнитуру, кегль, цвет и другие параметры шрифта в основном тексте. Вы можете устанавливать особые шрифты для заголовков, панелей навигации, вспомогательных текстов и т. д. Но если вы размещаете на сайте сплошной текст, который пользователь должен читать (например, сообщение на ленте новостей, статью, обзор продукта) — позвольте этому тексту отображаться таким шрифтом, к которому пользователь привык.

7.1.2. Лень — двигатель прогресса

Пользователь не должен прилагать лишних усилий

Я вас хотел избавить от хлопот,
Но если вы находите в них радость,
Я умолкаю.

В. Шекспир. Двенадцатая ночь

Человеку свойственно оптимизировать свою деятельность. Из нескольких способов решения одной и той же задачи мы выбираем тот, на который требуется затратить меньше времени, усилий, средств и других ресурсов.

Пользователи Интернета выбирают те сайты, с помощью которых они выполняют свои задачи с наименьшими затратами. Например, если один сайт предлагает перед загрузкой файла заполнить определенную форму, а на другом сайте этот же файл можно загрузить без всяких форм, — пользователь предпочтет второй сайт.

Внимание!

Постарайтесь уменьшить затраты времени и труда пользователей всеми возможными средствами.

- **Пользователи не любят ждать.** Они хотят, чтобы страницы загружались как можно быстрее. Поэтому следует добиваться того, чтобы объем страницы был как можно меньше. В частности, для этого рекомендуется сокращать тексты и разбивать длинные тексты на несколько страниц.

Примечание

Для пользователей выделенных линий, количество которых в нашей стране постоянно увеличивается, проблемы времени не так актуальны, как для пользователей модемов. Однако эти пользователи платят за трафик, а следовательно, для них тоже выгоднее, если объем страниц меньше.

Конечно, автор не призывает доходить до абсурда и выкидывать со страниц все, кроме «голого» текста со ссылками. Однако страницы все же имеет смысл «пропалывать», удаляя с них то, что в действительности не нужно пользователям и никак не помогает им выполнять задачи, ради которых они пришли на сайт.

- **Пользователи любят определенность.** Прежде всего, попав на какую-либо страницу, пользователь должен ясно понимать:
  - где он находится (на каком сайте, в каком разделе этого сайта);
  - что он может здесь сделать, что он может получить от данной страницы;
  - куда он может отсюда перейти.

  Немалую роль при этом играет тот текст, который пользователь видит на странице. Информацию о том, где находится пользователь, ему предоставят грамотно составленные «хлебные крошки», которые должны начинаться с названия сайта:

Рекомендуется	Не рекомендуется
Компания «ABC» → Филиалы → Санкт-Петербург	Главная страница → Филиалы → Санкт-Петербург

Что пользователь может сделать на странице — он увидит благодаря информативным заголовкам и грамотному расположению блоков на странице. Куда пользователь может перейти — он узнает из находящихся на странице гиперссылок, текст которых должен соответствовать ресурсу, на который они ведут.

Кроме решения трех основных вопросов, требуется определенность и во второстепенных вопросах, например:

- размещайте на каждой странице дату ее последнего обновления. Пользователь не должен гадать, читает ли он свежий материал или давно устаревший;
- сообщайте полную и точную информацию об обновлениях и изменениях на сайте;

Примечание

Рассмотрим электронную библиотеку на сайте Московского центра непрерывного математического образования (рис. 7.3). Эта библиотека часто пополняется, и в верхней части страницы размещается дата последнего обновления. Однако нет никакой информации о том, в чем именно заключается это обновление. Приходится просматривать длинный список книг, чтобы выяснить, какая же новая книга в нем появилась.

МЦНМО

МОСКОВСКИЙ ЦЕНТР НЕПРЕРЫВНОГО МАТЕМАТИЧЕСКОГО ОБРАЗОВАНИЯ

Свободно распространяемые издания

Последнее обновление 21 января 2004 года

На этой странице мы предлагаем размещать материалы (полные тексты) свободно распространяемых книг, предоставленные авторами и издательствами (по возможности в форме оригинал-макетов с исходными текстами), а также записи лекций, сборники задач, программы курсов и т.п.

Смотрите также страницы, посвященные сравнительно дешёвым распространяемым книгам.

О кодировках и форматах

В качестве стандартного формата для печатных материалов мы используем postscript. Такие файлы можно печатать на специальных принтерах, а также печатать и просматривать с помощью программы ghostscript.

Многие тексты подготовлены с помощью программы TeX (автор — Дональд Кнут). Её варианты для определенных систем DOS-, Windows- и UNIX-based (с русскими добавлениями) см., например, в ftp://ftp.mccme.ru/pub/tex/ или на сайте СТАН ftp://ftp.dante.de/tex-archive. Можно также воспользоваться русификацией Александра Виноградова. Имейте в виду, что версия существенно зависит от используемой русификации.

Материалы

Значения знаков копирайта, например, (c2), смотрите в конце этой страницы.

1. (c3) "Библиотека "Математические олимпиады" (серия Бронзы)
2. (c3) Введение в криптографию. Под общей редакцией *А.Б.Ашрова*.
   - Сведения о книге М. МЦНМО, 2000, 272 с.
   - Оглавление
3. (c1) *Н.К. Верещагин, А.Шень*. Лекции по математической логике и теории алгоритмов.
   - Часть 1. Начала теории множеств
     - Сведения о книге М. МЦНМО, 1999, 128 с.
       (В электронной версии исправлены несколько опечаток, замечанных в отзывах)
     - Оглавление
     - полный текст (zipped postscript, 250 Кб), (zipped pdf, 430 Кб)
     - полный исход. (TeX, tar, zipped 6, 460 Кб)
   - Часть 2. Языки и исчисления
     - Сведения о книге М. МЦНМО, 2000, 288 с.
     - полный текст (zipped postscript, 540 Кб)
     - полный исход. (TeX), 600 Кб
   - Часть 3. Вычислимые функции.
     - Сведения о книге М. МЦНМО, 1999, 176 с.
     - оглавление
     - полный текст (zipped postscript, 320 Кб), (zipped pdf, 450 Кб)
     - полный исход. (TeX), 140 Кб
4. (c2) *С.Г.Гиндикин*. Рассказы о физиках и математиках.
5. (c2) *И.М.Гельфанд*. Лекции по линейной алгебре.
   - Сведения о книге М. Добросвет, МЦНМО, 1998. Издание пятое, исправленное, 320 с.
   - Оглавление и полный текст (zipped postscript, zipped pdf-postscript).
6. (c2) *И.М.Гельфанд, С.Г.Гиндикин, М.И.Граев*. Избранные задачи интегральной геометрии.
7. (c2) *А.В.Гладкий*. Введение в современную логику.
   - Сведения о книге М. МЦНМО, 2001, 200 с.
   - Полный текст (zipped postscript, 1.10Мб)
   - Полный текст (zipped postscript, 1.10Мб)
8. (c2) *Р.Е.Горбун*. Задачи по геометрии.
9. (c2) *А.Громыко*. Рассказы о числах.
10. (c1) Задачи математического класса (57 школа, выпуск 2000 года, класс Б)
    - Сведения о книге Задачи по математике под редакцией *А.Шень*. М. МЦНМО, 2000, 272 с.
    - Предисловие (PS-файл)
    - 5-МЕС 5 класс, 9 класс, 10 класс (PS-файлы)
    - Олимпиадные лекции по математике
    - Полный PS-файл, gzip (551 Кб)
    - исходные TeX-файлы (701 Кб)
11. (c2) *М.А.Кочарян*. Курс дифференциальной геометрии (2001-2002).
    - Сведения о книге М. МЦНМО, 2002, 42 с.
    - Полный текст (zipped postscript, 201Кб)
    - Полный текст (zipped postscript, 201Кб)
12. (c1) *А.Лупанов, А.Шень, М.Вялый*. Классические и квантовые вычисления.
    - Сведения о книге М. МЦНМО, 1999, 192 с.
    - оглавление
    - полный текст (zipped postscript, 540 Кб), (TeX, zipped, 290 Кб)
    - Русский перевод (zipped postscript, 470 Кб), (TeX, zipped, 1.1 Мб)
13. (c1) *С.М.Лоповский*. Лекции по комплексному анализу
    - Сведения о книге М. МЦНМО, ВМУ, 136 с.
    - Оглавление и полный текст (zipped postscript, zipped postscript).
14. (c1) *С.М.Лоповский*. Набор и верстка в пакете LaTeX. (3-е издание, испр. и доп.)
    - Сведения о книге М. МЦНМО, 2003, 448 с.
    - полный текст Postscript (zipped, 1.3 Мб),
    - полный текст PDF (zipped, 4.4 Мб),
    - исходные тексты (TeX, zipped, 450 Кб),
    - полный текст PDF (с возможностью поиска в Acrobat'ыне, шрифты соответствуют печатному изданию: Файл подготовлен И.М.Минковским) (PDF, 5.5 Мб)
15. (c1) Математические олимпиады
16. (c2) *И.М.Парамонова, О.К.Шебзухова*. Задачи семинара "Алгебры Ли и их приложения"
    - Сведения о книге М. МЦНМО, 2004, 48 с.
    - Полный текст (zipped postscript, 120Б)
    - Полный текст (zipped postscript, 120Б)
17. (c2) *Б.Б.Прасолов*. Задачи по планиметрии.
    - Сведения о книге М. МЦНМО, 2001, 584 с.
    - Вид текста. Читаются лучше под Ghostview
    - Архив этой Ghostview (2.5 Мб) 7z: все без картинок (0.3 Мб)
    - Исходные тексты книги (TeX), EPS и PDF версии. В этих файлах исправлены замеченные в четвертом издании опечатки.
18. (c1) Программирование: входной курс. Под редакцией *Д. Шенгольца*.
    - Сведения о книге М. МЦНМО, 1995, 96 с.
    - Файлы с дискеты, распространявшейся с книгой. (zipped, 820 Кб).
      Содержит архив самой книги на языке TeX и программы, сопровождающие курс.
19. (c2) *Б.А.Розенфельд*. Аполлоний Пергский.
    - Сведения о книге М. МЦНМО, 2004, 176 с.
    - полный текст (PDF, 1.2 Мб)
20. (c2) *А.Б.Сосинский*. Математический кружок. 7 класс.
    - Сведения о книге М. Мех.-мат. МГУ, 2001, 72 с.
    - (PDF, 663 Кб)
21. (c0) Турниры им. Ломоносова (задачи, решения и др. информация)
    - 25 турнир (2002) (ISBN 5-94057-066-4)
    - 24 турнир (2001) (ISBN 5-94057-020-8)
    - 23 турнир (2000) (ISBN 5-900916-74-X)
    - 22 турнир (1999) (ISBN 5-900916-44-8)
22. (c1) *В.А.Успенский*. Труды по нелинейному анализу.
    - Сведения о книге М.: ОГИ, 2002, 1408 с. в 2-х томах.
    - Файлы книги
23. (c2) *А.Г.Хованский*. Комплексный анализ.
    - Сведения о книге М. МЦНМО, 2004, 48 с.
    - Полный текст (zipped postscript, 130Б)
    - Полный текст (zipped postscript, 130Б)
24. (c1) *А.Шень*. Математическая индукция.
    - Сведения о книге: не опубликована, 32 с.
    - Полный текст (zipped postscript, 140Б)
    - Полный текст (zipped postscript, 140Б)
25. (c1) *А.Шень*. Программирование: теоремы и задачи.
    - Сведения о книге М. МЦНМО, 1995, 264 с.
    - полный текст (zipped postscript, 450 Кб)
    - полный исходный текст (с такими (TeX, AMS, EPS)
26. (c2) *М.А.Шубин*. Псевдодифференциальные операторы и спектральная теория.
    - Сведения о книге М. Добросвет, 2003, 312 с., 2-е изд., испр. и доп.
    - полный по главам PS (zipped and PS zipped)
27. *И.А.Яценко*. Прикладная из математической лекции.

О копирайте (сокращения)

(C0) распространение без ограничений

(C1) распространение без ограничений (в электронном виде) можно распространять в электронном и печатном виде, но только без изменений и с сохранением копирайта

(C2) некоммерческое использование материалы можно просматривать и печатать, но коммерческое использование (например, издание и продажа) требует специального разрешения

(C3) только для просмотра на экране можно просматривать на экране, но не печатать

Возможны и другие варианты копирайта, если они явно указаны на соответствующей странице

Разное

- Ответы авторам
- Ответы редакции

Контакты

- Виталий Арнольд (vitar@mccme.ru),
- Михаил Вялый,
- Сергей Лоповский,
- Вадим Гиндикин (vadim@mccme.ru),
- Максим Умнов (umnov@mccme.ru),
- Дмитрий Шварц (dvsvarts@mccme.ru),
- Виктор Шуванов (shuvanov@mccme.ru).

Рис. 7.3. www.mccme.ru: есть дата последнего обновления, но пользователь вынужден сам разыскивать, что именно обновилось

- если ссылка ведет на загружаемый файл, указывайте формат и объем этого файла, чтобы пользователь мог заранее предугадать, сколько времени уйдет на загрузку.

Текст нужно составлять ясно и просто, чтобы пользователям не пришлось гадать, что вы имели в виду. Пользователь вряд ли будет сосредоточивать все свое внимание именно на вашем сайте. Скорее всего, одновременно с вашим сайтом у него на компьютере открыто еще несколько других, а также приложения, с которыми он работает в данный момент.

Часто возникают ситуации, когда пользователь бегло просмотрел страницу, и ему показалось, что он понял все, что хотел. Однако на самом деле какую-то важную информацию он пропустил, а какую-то понял неправильно. Сделайте все возможное, чтобы предотвратить такие ситуации.

❒ **Пользователи не хотят совершать лишних действий.** Например, на многих сайтах, предлагающих файлы для загрузки, пользователю сначала требуется зарегистрироваться или заполнить еще какую-нибудь форму. Кроме того, нередки случаи, когда пользователь щелкает по ссылке, которая, как ему кажется, должна вести прямо к нужному файлу, — а попадает на еще одну страницу, где и находится настоящая ссылка на файл.

Хороший пример здесь подает сайт браузера Mozilla (рис. 7.4). Многие пользователи заходят туда специально для того, чтобы загрузить наиболее свежую версию браузера. Поэтому прямо на главной странице находится ссылка, ведущая непосредственно к файлу.

Примечание

Составляйте достаточно ясные и определенные названия разделов сайта. Посмотрите, например, на сайт популярного исполнителя (рис. 7.5). В панели навигации есть три ссылки: **Альбомы**, **Музыка** и **Песни**, разделение между которыми совершенно непонятно. Автор предлагал нескольким пользователям, не переходя по этим ссылкам, предположить, чем разделы отличаются друг от друга. Ни один пользователь, который раньше не посещал этот сайт, не дал правильного ответа.

Постарайтесь, чтобы пользователям вашего сайта не пришлось играть в угадайку.

Еще один пример, когда пользователь вынужден совершать лишние действия — когда в тексте на веб-странице упоминается какой-нибудь сайт или адрес электронной почты, но это упоминание не является ссылкой:

Рекомендуется	**Не рекомендуется**
На сайте www.rbc.ru сообщили, что...	На сайте www.rbc.ru сообщили, что...
В. Пупкин (pupkin@dostal.ru)	В. Пупкин (pupkin@dostal.ru)

Рис. 7.4. www.mozilla.org: пользователь сразу может загрузить файл, не заходя на другие страницы

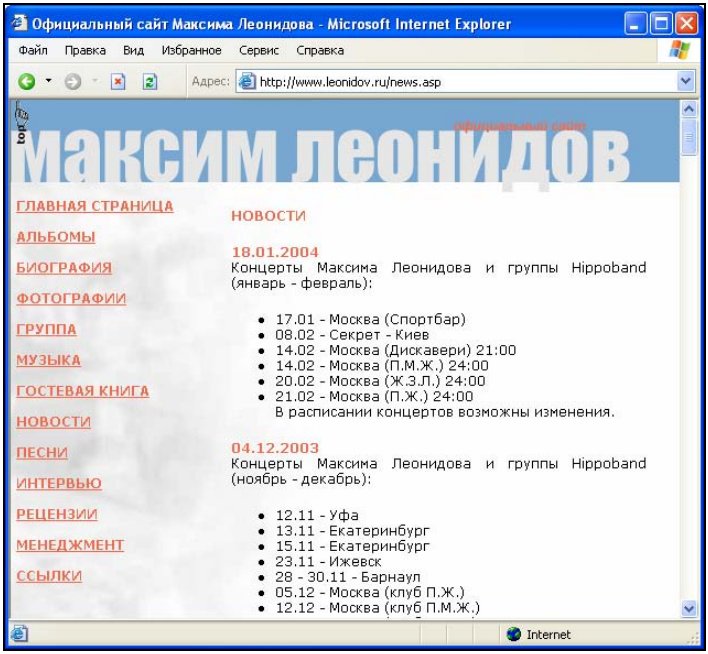


Рис. 7.5. www.leonidov.ru: не вполне понятно, куда ведут ссылки с этой страницы

Если упоминания ресурсов не являются ссылками, пользователю придется самостоятельно набирать их в адресной строке или копировать через буфер обмена, а это совершенно неоправданные действия.

- **Пользователи не любят напрягать память.** В частности, пользователь не утруждает себя запоминанием, на каких страницах он уже был, а на каких — еще нет. Поэтому требуется по-разному выделять посещенные и непосещенные ссылки.

Дополнительно

Вообще, с неграмотным оформлением ссылок связано много проблем, о которых уже упоминалось. Если не делать ссылки подчеркнутыми — пользователю придется угадывать, что на странице является ссылкой, а что — нет. Чтобы исследовать страницу, ему придется подводить указатель мыши к каждому подозрительному элементу и смотреть, превращается ли указатель в «указующий перст». Но даже если пользователь выполнил это исследование — он достаточно скоро забудет, оказался ли тот или иной элемент ссылкой. И при следующем посещении страницы ему приходится проводить изыскания заново.

Аналогично, если подчеркнут текст, не являющийся ссылкой, при первом посещении страницы пользователь легко может обмануться и щелкнуть по нему. Но к следующему посещению он вполне может забыть о своей неудаче и снова попытаться щелкнуть по этому тексту.

- **Пользователи скорее просматривают страницы, а не читают их.** Пользователь внимательно читает текст на веб-странице только в одном случае — если этот текст и есть именно то, ради чего он посетил сайт. Попав на ту или иную веб-страницу, пользователь думает только об одном: «Что мне надо здесь сделать, чтобы выполнить ту задачу, ради которой я пришел?» Например, если он хочет загрузить файл, он будет осматривать страницу на предмет ссылки **Загрузить**, **Скачать**, **Файлы**, **Download** и т. д. Остальное содержание страницы его не интересует, он не будет его читать.

Поэтому содержание страницы надо грамотно структурировать, разбивать на смысловые блоки, по необходимости использовать средства оформления. Пользователь должен сразу «схватить» общую картину страницы и быстро выделить в ней именно то, что ему надо.

Примечание

Что думает пользователь, посещая сайт на рис. 7.6? Скорее всего, этот сайт посвящен какой-то акции (или возможно, организации) под названием «Эстафета олимпийского огня». Внизу — рекламные логотипы спонсоров этой акции... Конечно, если присмотреться внимательнее, легко понять истинное предназначение сайта; но все дело в том, что большинство пользователей просто не будут присматриваться внимательнее!

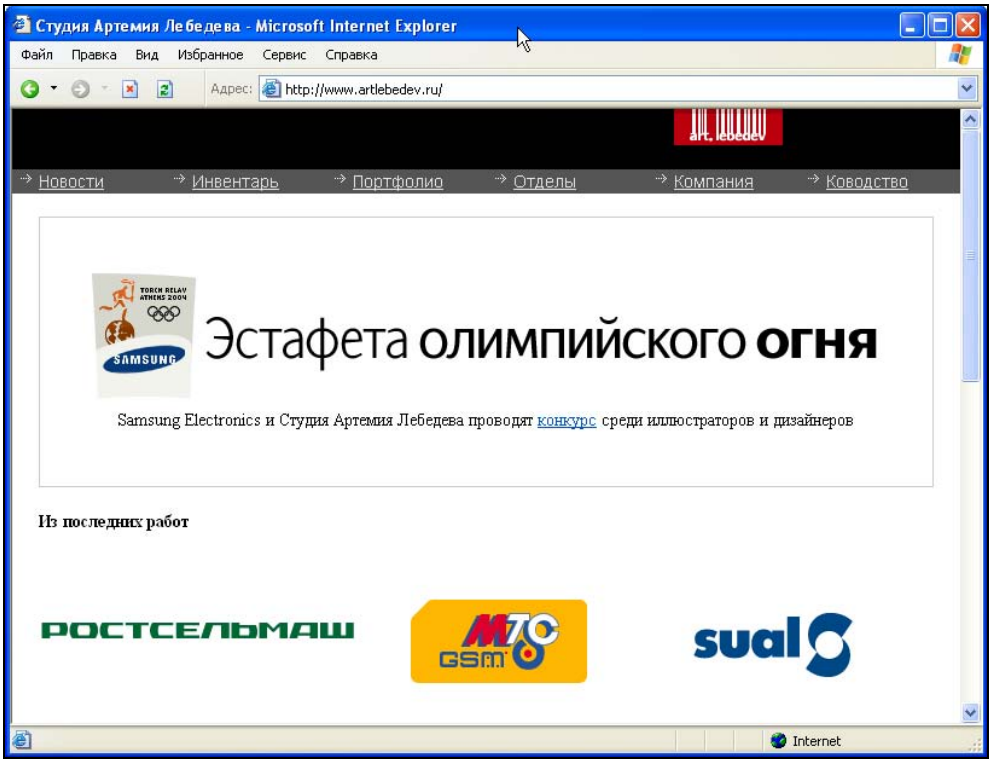


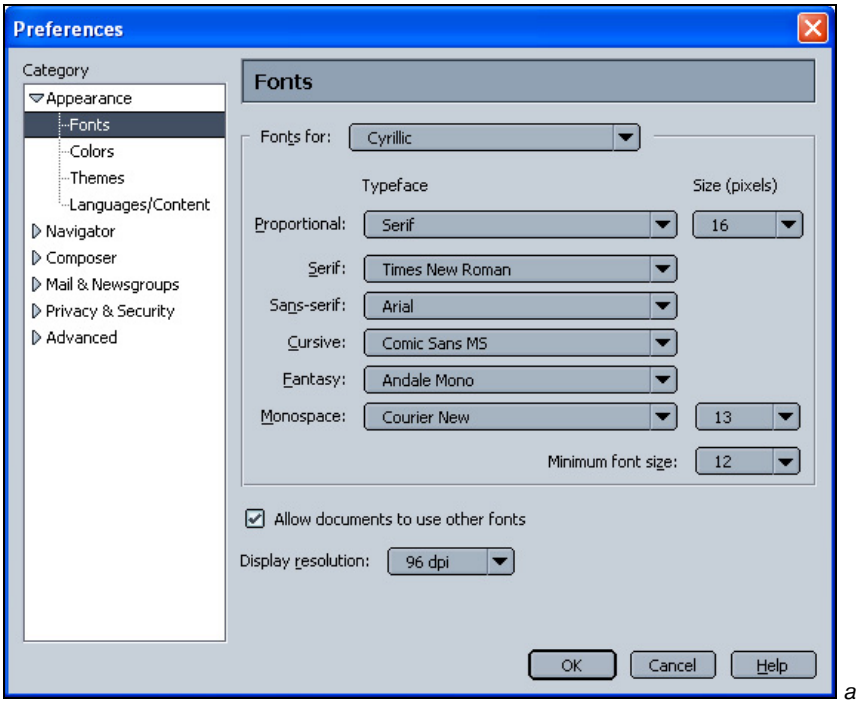
Рис. 7.6. www.artlebedev.ru: назначение сайта «удачно» замаскировано

- **Пользователь хочет контролировать свое взаимодействие с сайтом.** Сайт не должен навязывать пользователю определенного режима работы и определенных действий. Например, у пользователя должна быть возможность изменить размер шрифта, если он кажется ему слишком мелким или слишком крупным. Если размер шрифта указан в пикселах или других абсолютных единицах, то в некоторых браузерах пользователь не сможет изменить его размер.

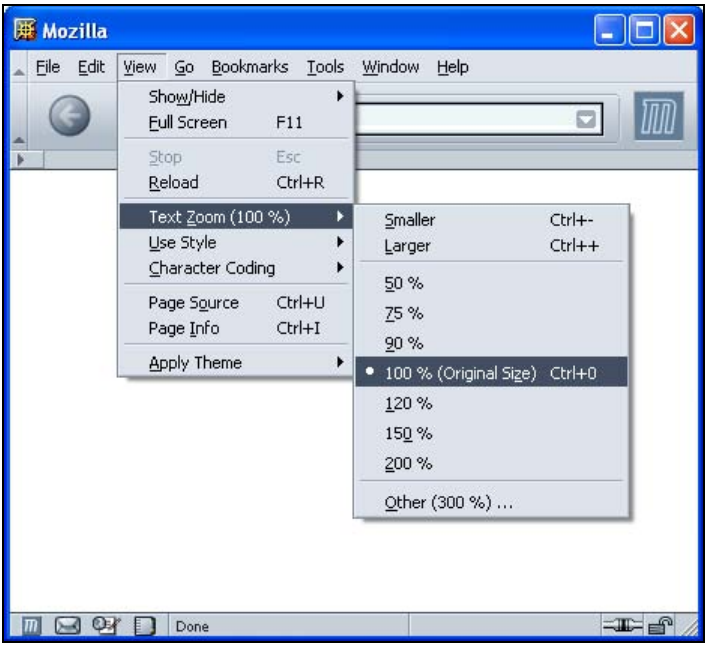
纸 *Дополнительно*

Браузер Mozilla предоставляет пользователям широкие возможности по настройке текста на веб-страницах (рис. 7.7). Например, поле **Minimum font size** позволяет установить минимальный размер шрифта, так что даже если на странице есть более мелкий шрифт, Mozilla увеличит его до этого размера.

Также Mozilla позволяет пользователю масштабировать шрифт, даже если разработчик задал его кегль в фиксированных единицах.



а



б

Рис. 7.7. Браузер Mozilla позволяет установить минимальный размер шрифта для просмотра (*а*) и масштабировать текст на веб-странице как угодно (*б*)

Некоторые страницы автоматически обновляются через определенные промежутки времени. Это совершенно недопустимо: представьте себя на месте пользователя, который спокойно просматривает страницу... и вдруг она исчезает и начинает загружаться заново! Такое поведение страницы — верный способ потерять пользователя.

Дополнительно

Пользователь должен контролировать также поведение ссылок. Он может открыть ссылку в текущем окне (так браузеры поступают по умолчанию), открыть ее в новом окне или в другой панели текущего окна (такая возможность есть, например, в браузерах Mozilla и Opera).

Если разработчик насильно заставляет ссылки открываться в новом окне, это уменьшает контроль со стороны пользователя. Предоставляйте пользователю право самому решать, где он хочет открывать каждую ссылку.

Многие аспекты контроля пользователя над сайтом не связаны непосредственно с текстовым содержанием, поэтому упомянем их вкратце:

- пользователь может изменять размер окна браузера, поэтому композиция страницы должна быть «резиновой», чтобы автоматически подстраиваться под текущий размер окна;
- пользователь хочет видеть, куда ведет ссылка, еще до того, как щелкнет по ней. Обычно адрес назначения ссылки отображается в строке состояния браузера, когда на ссылку наведен указатель мыши. JavaScript позволяет разработчику поместить в строку состояния любой другой текст, но пользоваться этой возможностью не следует;
- пользователя раздражают всплывающие окна (pop-up windows), выскакивающие без его ведома. К счастью, современные браузеры позволяют отключать эти окна.

❒ **Пользователь склонен уставать.** В самом деле, пользователь — не машина, он может испытывать нетерпение, скуку и усталость, спешить, раздражаться или лениться. Пользователь не будет просматривать 100 однотипных ссылок подряд, чтобы выбрать из них подходящую. Если у пользователя не получилось с первой попытки найти на сайте то, что ему нужно, он редко совершает вторую попытку и практически никогда — третью.

Пользователь может отвлечься от работы над сайтом, отойти куда-нибудь, переключиться в другое приложение на своем компьютере и т. д. Это необходимо учитывать при разработке сайтов.

Некоторые разработчики рассчитывают сайты на «сферических пользователей в вакууме», которые не устают и не расслабляются, которые направляют все свое внимание только на данный сайт и взаимодействуют с ним в точности так, как хотелось бы разработчикам. Таких пользовате-

лей не бывает. Вы разрабатываете сайты для живых людей, и не стоит забывать об этом.

Копирование, сохранение и печать текста

> Король повернулся к Тому и ласково сказал:
> — Мой бедный мальчик, как ты мог вспомнить, куда я спрятал печать, если я и сам не мог вспомнить?
>
> *М. Твен. Принц и нищий*

Пользователь может выделить фрагмент текста на странице и скопировать его в свой документ или сохранить еще каким-нибудь способом. Эта операция не должна приводить к затруднениям.

Прежде всего, такие затруднения возникают, если текст заключен в изображение. Чтобы процитировать этот текст в своем документе, пользователь вынужден перепечатывать его вручную.

Но встречаются и другие, менее очевидные проблемы при копировании текста. Например, если на странице размещен перечень (`ul` или `ol`), и пользователь выделяет его и копирует в буфер обмена, — то во многих браузерах маркеры перечня при этом не копируются:

На веб-странице	В буфере обмена	
	Internet Explorer	**Mozilla**
1. Красный	Красный	1. Красный
2. Жёлтый	Жёлтый	2. Жёлтый
3. Зелёный	Зелёный	3. Зелёный

Таким образом, при копировании текста нумерация может пропасть. Это особенно нежелательно, если она сама по себе является важной частью содержания.

纸 *Дополнительно*

Еще один пример, когда текст портится при копировании, связан с верхними и нижними индексами. Например, чтобы поместить на веб-страницу конструкцию «C_n^k», можно составить такой код HTML и CSS:

HTML:
```
<var>C</var><sub><var>n</var></sub><sup><var>k</var></sup>
```

CSS:
```
sup {
  position: relative;
  left: -0.4em;
}
```

Однако, если скопировать результат в буфер обмена и вставить в текстовый файл, получится обычный текст: `Cnk`. Индексы потеряются, и смысл выражения исказится.

Бывают проблемы и со стихотворным текстом, если отступы и пробелы в нем создаются с помощью CSS, например:

На веб-странице	После копирования в буфер обмена
Я	Я
теперь	теперь
свободен	свободен
от любви	от любви
и от плакатов,	и от плакатов,
Шкурой	Шкурой
ревности медведь	ревности медведь
лежит когтист.	лежит когтист.
Можно	Можно
убедиться,	убедиться,
что земля поката,—	что земля поката,-
сядь	сядь
на собственные ягодицы	на собственные ягодицы
и катись!	и катись!

Примечание

Эта же неприятность может случиться с текстом песни, над которым помещены аккорды для гитары:

На веб-странице	После копирования в буфер обмена
Am Dm E_7 Am G_7	Am Dm E7 Am G7
Не слышны в саду даже шорохи,	Не слышны в саду даже шорохи,
C G_7 C	C G7 C
Всё здесь замерло до утра.	Всё здесь замерло до утра.

Чтобы избежать такой проблемы со стихотворным текстом, следует помещать его в элемент `pre` — тогда и при отображении на веб-странице, и при копировании все пробелы сохранятся.

С перечнями и более сложными текстами дело обстоит сложнее. Если вы ожидаете, что пользователь захочет скопировать фрагменты текста с веб-страницы, предложи́те ему ссылку на файл в формате «обычный текст» (plain text), где представлено содержание страницы в наиболее простом виде, но с учетом перечней и других подобных элементов.

纸 *Дополнительно*

При копировании текста с веб-страницы могут возникнуть проблемы со специальными символами — например, тире, кавычками, буквами с диакритическими знаками. Если пользователь редактирует собственный документ (например, электронное письмо) в одной из расширенных кодировок ASCII (например, в KOI8-R) — то в тексте, скопированном с веб-страницы через буфер обмена, специальные символы могут превратиться в посторонние знаки:

На веб-странице	**После копирования через буфер обмена**
— Отчаянное положение, — проговорил он. — Ведь кое-что помню! «Ха-ха-ха! Будет чем полакомиться: конь — на обед, молодец — на ужин...» Откуда бы это? А Иван — сами понимаете — дурак, отвечает: «Эх ты, поганое чудище, не уловивши бела лебедя, да кушаешь!» Потом, естественно — калёная стрела, все три головы долой, Иван вынимает три сердца и привозит, кретин, домой матери... Каков подарочек! — Кот сардонически засмеялся, потом вздохнул. — Есть ещё такая болезнь — склероз, — сообщил он.	≈ Отчаянное положение,≈ проговорил он.≈ Ведь кое-что помню! ╚Ха-ха-ха! Будет чем полакомиться: конь ≈ на обед, молодец ≈ на ужин╝ ╩ Откуда бы это? А Иван ≈ сами понимаете ≈ дурак, отвечает: ╚Эх ты, поганое чудище, не уловивши бела лебедя, да кушаешь!╩ Потом, естественно ≈ калёная стрела, все три головы долой, Иван вынимает три сердца и привозит, кретин, домой матери╝ Каков подарочек! ≈ Кот сардонически засмеялся, потом вздохнул.≈ Есть ещё такая болезнь ≈ склероз,≈ сообщил он.

Альтернативная версия страницы в формате «обычный текст» позволяет избежать этой проблемы, т. к. вы можете проконтролировать, чтобы в этой версии употреблялись только те символы, которые доступны в выбранной кодировке. Если вы хотите обеспечить грамотную типографику (тире, кавычки, многоточия и др.) — сохраняйте текст в кодировке Windows-1251, а еще лучше — в UTF-8.

Пользователь может пожелать сохранить веб-страницу на своем компьютере — например, чтобы дочитать ее позже, без подключения к Интернету, или чтобы обращаться к ней впоследствии. Ему также может потребоваться распечатать страницу. Для обеих этих задач хорошо подходит специальная версия страницы, предназначенная для печати.

Быстрая помощь
Выберите...

SUPER GSM

Отправь SMS

Мой Би Лайн | Новости | Тарифы | Услуги | Телефоны | Территория | Где купить | О компании | Контакты

Где купить /

Где купить
- Офисы Би Лайн
- Салоны «Мобайл Центра»
- Дилеры Би Лайн
- Где оплатить

Поиск по сайту

Офисы Би Лайн

Где купить
Офисы Би Лайн

Офисы Би Лайн работают ежедневно:
- По рабочим дням — с 9:00 до 20:00
- По субботам и воскресеньям — с 10:00 до 17:00

Офис «Третьяковская»

Адрес: 3-й Кадашевский пер.,д 8

Тел.: 755-0031
Факс: 755-0031

Офис «8 марта»

Адрес: ул. 8 Марта,д 10, стр. 14

Тел.: 212-8283
Факс: 212-8225

Офис «Маяковская»

Адрес: ул. 1-я Тверская-Ямская, д. 2, стр. 1

Тел.: 251-7000
Факс: 251-0584

Версия для печати

Мобильный кошелёк
Новое качество мобильного телефона - покупка товаров и услуг с помощью сотового!
подробнее

beeoffice | beeonline

© ОАО «ВымпелКом», 2003.
Все права защищены.
Нелицензированное использование материалов данного сайта запрещено.

Карта сайта | Информация о сайте | English version

designed by ActisSystems

Рис. 7.8, *а*. www.beelinegsm.ru: основная версия страницы

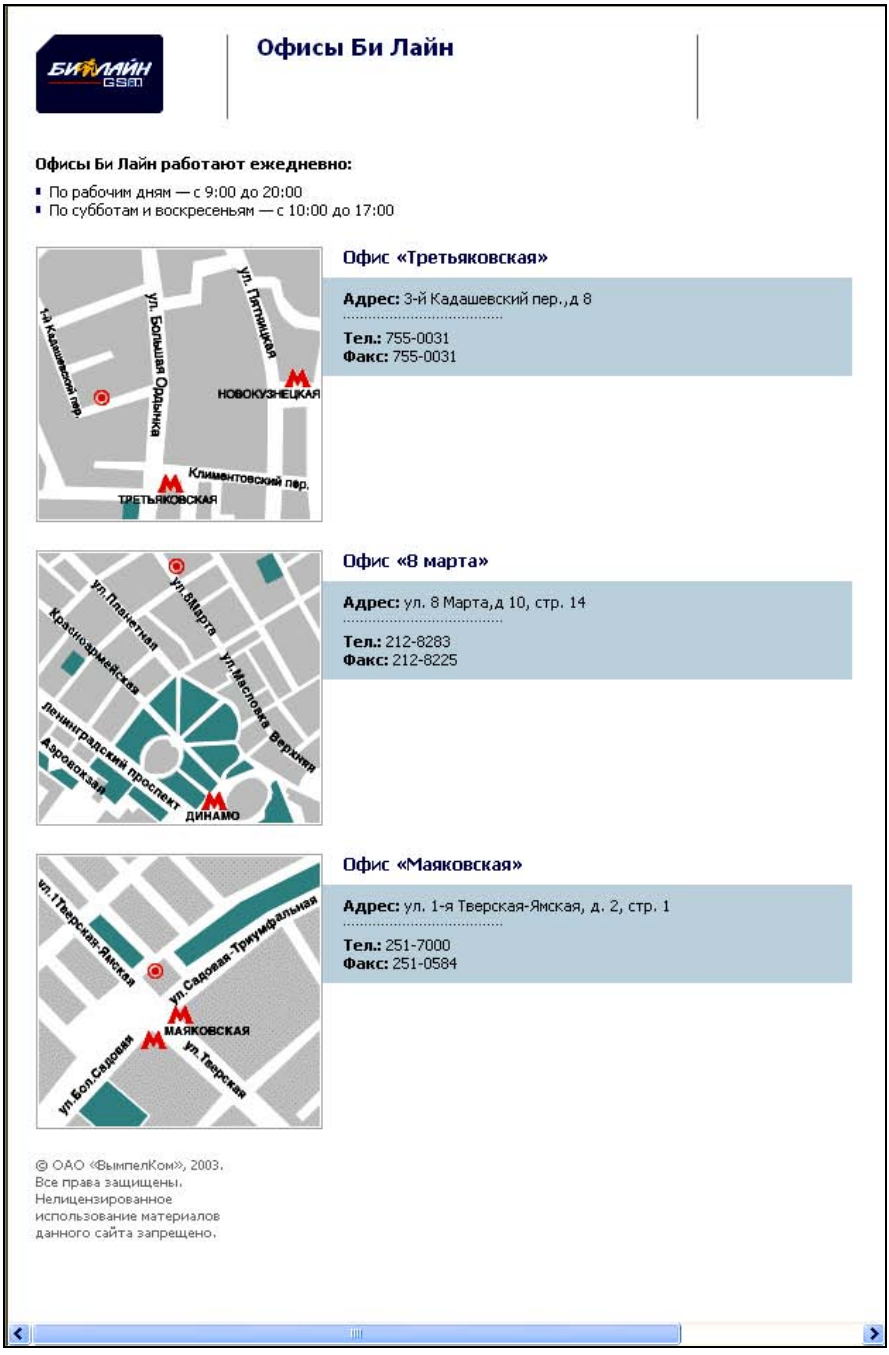


Рис. 7.8, *б*. www.beelinegsm.ru: в версии для печати убрано все, кроме полезной информации

При подготовке версии для печати придерживайтесь следующих рекомендаций:

- предоставляйте версию для печати в формате PDF. Это наиболее универсальный формат для документов, предназначенных для постоянного хранения и печати;
- помещайте в версии для печати только содержательную часть страницы. Не надо копировать с веб-страницы навигацию, оформление, вспомогательный аппарат (рис. 7.8);
- в версии для печати, кроме содержания, должны быть следующие элементы:
  - название (и, возможно, логотип) сайта, с которого загружена эта версия, адрес сайта;
  - постоянный адрес этой страницы в Сети;
  - дата (и, возможно, время), когда была получена эта версия для печати;
  - адрес электронной почты или другой способ связи с лицом, ответственным за подготовку версии для печати.

7.1.3. Доверяй, но проверяй
О тестировании удобства сайта

> Король в сопровождении свиты сам пришел к ним одеваться. Обманщики поднимали кверху руки, будто держали что-то, приговаривая:
> — Вот панталоны, вот камзол, вот кафтан! Чудесный наряд! Легок, как паутина, и не почувствуешь его на теле! Но в этом-то и вся прелесть!
>
> *Г.-Х. Андерсен. Новое платье короля*

Тестирование удобства (usability testing) — пожалуй, один из самых сложных этапов в разработке сайта. Здесь приходится иметь дело с субъективными впечатлениями пользователей и другими факторами, практически не поддающимися точному измерению. Но, с другой стороны, это и самый важный этап: ведь сайт разрабатывается именно для того, чтобы им пользовались. Значит, надо сделать его таким, чтобы он производил на пользователей хорошее впечатление, чтобы они считали его полезным и нужным, а также удобным в использовании.

Многие книги и другие источники (в том числе [13] и [14]) рассказывают об общих принципах удобства сайта. Эти принципы сформулированы на основании практического опыта и подтверждены многочисленными исследованиями. Но это вовсе не незыблемые истины. Их можно взять за основу при

оценке удобства сайта, но окончательные выводы можно сделать только после тестирования удобства с участием конкретных пользователей.

Все методы тестирования удобства основываются на главном принципе:

Внимание!

Основным критерием качества сайта являются результаты практического взаимодействия с ним реальных пользователей.

Пока вы разрабатываете сайт, вы можете придерживаться того или иного мнения о том, что понравится пользователям, а что — нет, что они сочтут удобным, а что будет вызывать у них затруднения. Но все ваши предположения — это только гипотезы, и они нуждаются в обязательной проверке, которую можно осуществить только с помощью настоящих пользователей сайта.

Примечание

Разумеется, все пользователи разные. У всех разные вкусы и привычки; одним нравится одно, другим — другое. Именно поэтому нужно проводить тестирование удобства, которое поможет выявить проблемы *разных* пользователей и сделать сайт удобным не для какого-то абстрактного «среднего» пользователя, а для большого количества различных *конкретных* посетителей.

В самом деле, иногда пользователи встречаются с такими затруднениями, о которых разработчики даже не подозревали.

Как же провести тестирование удобства? Существует множество методик — от простых и практически беззатратных, которые годятся для небольших сайтов, до дорогостоящих, требующих привлечения профессиональных экспертов и подходящих для крупных проектов.

Но прежде — несколько слов о том, как *не надо* тестировать удобство:

- не надо привлекать в качестве тестовой группы для своих сайтов собственных родственников и знакомых. Они редко бывают объективными и рассматривают сайт не как нечто, чем они будут пользоваться на практике, а как работу своего друга. Ищите таких пользователей, которые действительно будут или по крайней мере могут работать с сайтом по его прямому назначению;

Дополнительно

Знакомые, которые хорошо вас знают, увидят в вашей работе не то, что вы действительно сделали, а то, что вы хотели сделать. Они знают, как вы мыслите, как ведете себя, — значит, они видят на сайте много такого, что вовсе

не бросится в глаза обычным пользователям. Действительно, они скорее угадывают, чего вы хотели достичь тем или иным решением, и оценивают сайт так, как будто вы действительно этого достигли.

- не спрашивайте пользователей: «Вам больше нравится синий цвет заголовков или черный?» «Вы считаете, что этот блок лучше поместить справа или слева?» «Хорошо ли смотрится здесь шрифт Verdana?» Представьте себе, обычные пользователи на самом деле *не знают*, что им удобнее. Они могут высказывать те или иные мнения о том, что, *как они считают*, будет для них лучше. Но когда дело доходит до реального использования сайта — чаще всего оказывается, что они ошибались в своих оценках.

  Не забывайте, что пользователи — не специалисты по разработке сайтов. Пользователи не должны советовать вам, что вы должны делать, — в какой цвет раскрашивать те или иные элементы, в каком порядке размещать блоки. Наоборот, вы должны узнавать, что хотят делать пользователи, посещая сайт, — и, исходя из этого, решать, как вы им это обеспечите.

Примечание

Пользователь, которому задали вопрос о том, нравится ли ему шрифт Verdana, без колебаний ответил, что нравится. В самом деле, с его точки зрения, страница с этим шрифтом выглядела весьма красиво. Но когда пользователь решил прочесть крупный материал, расположенный на этой странице, он убедился, что его глаза быстро устают, и ему сложнее читать, чем обычно.

Наоборот, когда другого потенциального пользователя спросили о том, что его не устраивает на сайте, он ответил, что ссылки следовало бы сделать потемнее. Вместе с тем, опрос двух десятков реальных пользователей показал, что никто из них, практически работая с сайтом и выполняя с его помощью реальные задачи, ни разу не обратил внимания на цвет ссылок и не испытал из-за этого никаких затруднений.

То, что пользователи думают о сайте, и то, как они действительно работают с этим сайтом, — совершенно разные вещи. Ориентируйтесь на реальную практику, а не на субъективные ощущения.

Напомним, что эта книга посвящена текстовому содержанию сайтов. Сайт не ограничивается только этим содержанием, и книга не раскрывает всех вопросов, касающихся удобства сайта. Автор надеется, что для получения полного представления об удобстве использования сайтов вы уже прочитали или в скором времени прочтете и такие книги, как [13] или [14].

Любая методика грамотного исследования удобства опирается на наблюдение за поведением конкретных пользователей. Исследователь смотрит, как реальный пользователь заходит на сайт и какие действия он там выполняет. Он не руководит действиями пользователя, не вмешивается в его работу. Он только фиксирует процесс взаимодействия пользователя и сайта, чтобы потом проанализировать его.

Простая методика тестирования удобства приводится в книге [15]. Реальному пользователю предлагают выполнить на сайте какую-нибудь конкретную, привычную ему задачу, и внимательно отмечают все затруднения, с которыми пользователь сталкивается в процессе выполнения. Это тестирование практически беззатратно; для него нужен только пользователь и его обычный компьютер.

В более серьезных исследованиях задействуется больше технических средств, привлекается больше пользователей. Процесс работы с сайтом записывается с помощью видеокамер, измеряется время, которое пользователи тратят на каждое элементарное действие. Специальные приборы следят за движением глаз пользователей; дело может даже доходить до измерения частоты пульса. Такие исследования проводятся в специальных лабораториях и стоят очень дорого.

纸 *Дополнительно*

Nielsen Norman Group (**www.nngroup.com**) — одна из известных компаний, занимающихся тестированием удобства сайтов. Полноценное исследование удобства стоит несколько десятков тысяч долларов. Однако находится немало владельцев сайтов, которые сотрудничают с Nielsen Norman Group. Они убедились на собственном опыте, что, если их сайт неудобен, они недополучают гораздо больше прибыли, чем та сумма, в которую им обойдется профессиональное тестирование удобства.

Впрочем, пора переходить от общих слов к конкретным примерам. Рассмотрим два реальных случая тестирования удобства сайтов. В обоих случаях тестирование не было организовано специально — и оттого его результаты особенно ценны. В самом деле, наблюдение за реальной работой пользователей всегда дает более объективную информацию.

Редактор моей мечты

Поиск подходящего текстового редактора на сайте *www.listsoft.com*

— А у вас нет такого же, но с перламутровыми пуговицами?
— К сожалению, нет.
— Будем искать...

Л. Гайдай, Я. Костюковский, М. Слободской. Бриллиантовая рука

Пользователь — **Вадим**, студент технического вуза. Увлекается компьютерной графикой и разработкой сайтов. Работает, оформляя выставочные стенды в небольшой научно-производственной фирме. Хочет стать профессиональным промышленным дизайнером.

Автор: ...Я бы еще поспорил, что скорее включат в Unicode — тенгвар или дорожные знаки...

Вадим: О, кстати! Unicode! Ты мне напомнил — я как раз хотел найти текстовый редактор с поддержкой UTF-8!

Поворачивается к компьютеру, набирает в адресной строке браузера: `www.listsoft.ru`.

А.: А почему именно ListSoft?

В.: Ну... кто-то, кажется, посоветовал.

Тем временем в браузере открывается страница (рис. 7.9).

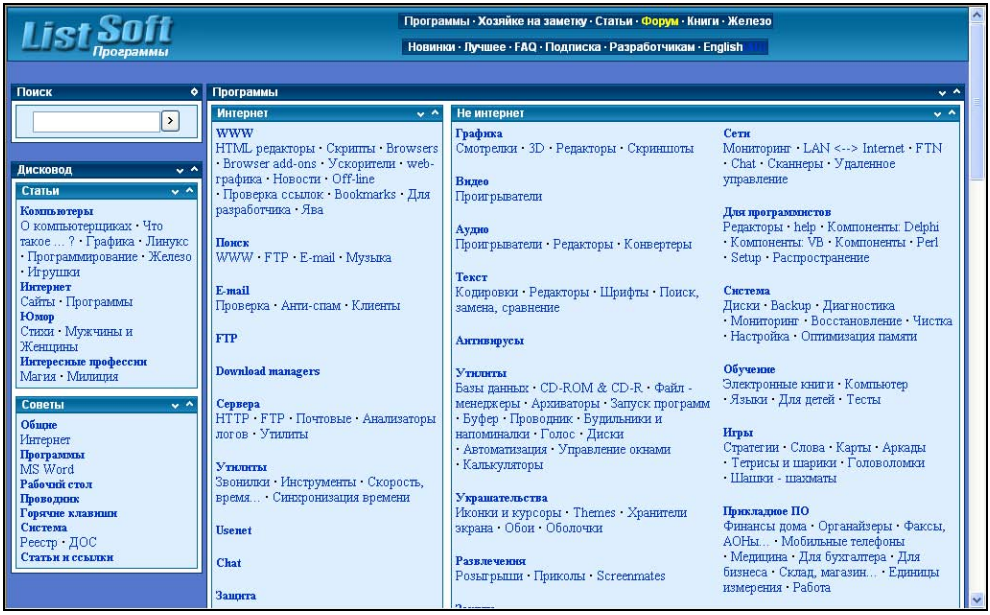


Рис. 7.9. www.listsoft.ru — главная страница

А.: А Блокнот в Windows тебя не устраивает? Он прекрасно поддерживает UTF.

В.: Шутишь? Что мне UTF без подсветки, автоматических отступов?.. Да! И чтобы сохранял по F2!

А.: То есть как твой Aditor, только с UTF.

Примечание

В этот момент сделан первый шаг в тестировании удобства: определены задачи пользователя, с которыми он посещает сайт. Сайт не может быть удобен или неудобен сам по себе; он может быть удобен или неудобен для тех или иных задач, которые пользователи хотят с его помощью выполнить.

А.: Что, какие-то затруднения?

В.: Не совсем понятно... То, что я ищу, — относится к «HTML-редакторам» или просто к «Редакторам»?

Примечание

Вот и первое затруднение. Мы видим, что содержание сайта разделено на два блока: **Интернет** и **Не Интернет**. В обоих блоках есть категории, подходящие по смыслу: в первой — **WWW | HTML-редакторы**, во второй — **Текст | Редакторы**.

В.: Ладно, начну просто с «Редакторов».

А.: Почему?

В.: Я думаю, что «HTML-редакторы» — это, скорее, всякие специализированные программы вроде Dreamweaver, а мне нужен простой текстовый редактор.

Примечание

Обратите внимание: пользователю приходится гадать, что скрывается за теми или иными ссылками. Стоит сказать, что в данном случае он угадал не совсем правильно — а это промах разработчиков сайта.

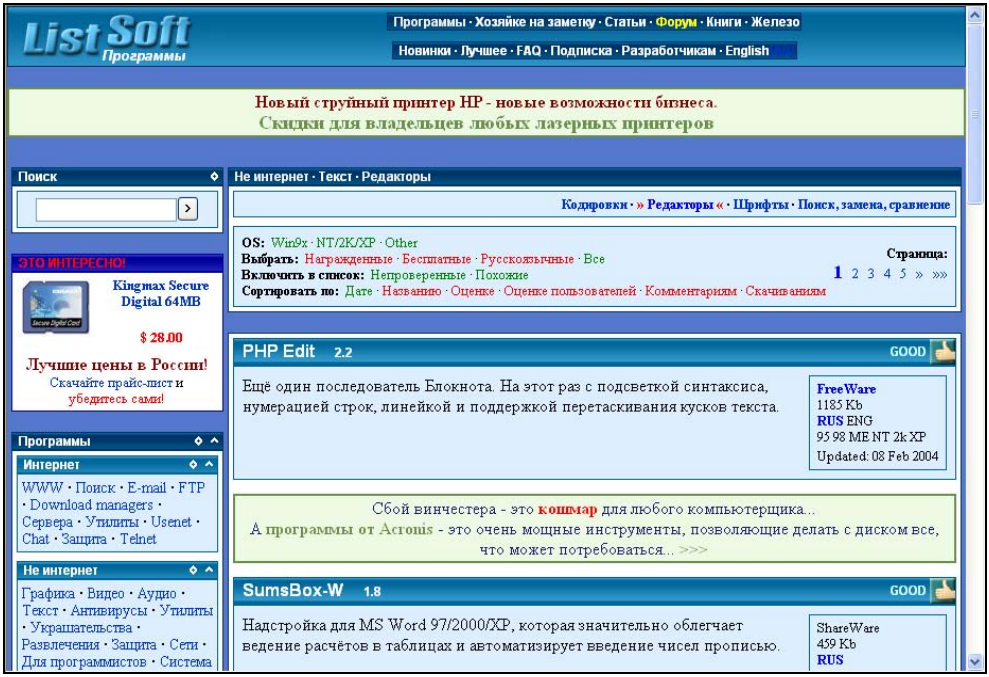


Рис. 7.10. www.listsoft.ru: раздел **Редакторы**

Открывается новая страница (рис. 7.10).

В.: Да, их тут немало... Целых пять страниц одних редакторов... Погоди, или не пять? Непонятно — тут имеется в виду, что всего пять страниц, или их больше?

Примечание

На самом деле там 15 страниц, но единственный способ узнать об этом — пролистать все страницы до последней.

В.: Тут какая-то ерунда... PHP Edit — судя по описанию, действительно редактор. Но дальше — какой-то SumsBox-W — надстройка для Office; потом FastNoter — «программа для упорядоченного хранения записей», потом какой-то «просмотрщик каталогов»...

А.: Да, из 10 записей на первой странице только 4 — действительно текстовые редакторы.

Примечание

Можно отметить, что заголовок раздела (**Редакторы**) не соответствует содержанию. Пример Вадима показывает, что это совсем не нравится пользователю.

В.: И ни слова про то, поддерживают ли они UTF. Про последний редактор в списке сказано, что он «с поддержкой различных кодировок текста (KOI8-R, DOS-866 и др.)», но ни слова про UTF-8. Значит, наверное, не поддерживает. Посмотрим вторую страницу...

Вадим просматривает страницы 2–4 и не находит ничего, что его бы заинтересовало.

В.: Нет, так мы будем до завтра искать...

А.: Но ведь ты пропустил несколько многообещающих ссылок. Вот, например, на второй странице — EmEditor: «Есть регулируемая подсветка синтаксиса, поддержка макросов, Unicode, регулярные выражения для поиска и замены». Раз Unicode — значит, скорее всего, поддерживает UTF-8...

В.: Ты предлагаешь мне скачать и проверить? А если он меня не устроит? А если он поддерживает UTF, но не поддерживает русские кодировки? Получится, что я скачал целый мегабайт совершенно зря. Ты хочешь, чтобы я скачивал, устанавливал и проверял все редакторы подряд?

Примечание

В самом деле — пользователь не будет загружать и тем более устанавливать все похожее на то, что он ищет. Не забывайте: пользователи любят определенность.

Рис. 7.11. www.listsoft.ru: результат поиска

А.: Давай тогда посмотрим еще одну страницу.

В.: Нет, мне уже лениво. Я целых пять страниц пролистал и ничего не нашел — где гарантия, что дальше будет лучше?

Примечание

Здесь можно отметить сразу две особенности, свойственные многим пользователям. Во-первых, пользователи стараются избегать однообразной, монотонной деятельности. В самом деле, редкий пользователь будет щелкать по всем ссылкам подряд и читать описание каждой «подозрительной» программы, не говоря уж о том, чтобы загружать ее. Во-вторых, многие пользователи быстро устают и разочаровываются — возможно, гораздо быстрее, чем стоило бы. Вадим не добрался даже до шестой страницы из пятнадцати.

В.: Может, воспользоваться поиском?

Вадим вводит `UTF` и запускает поиск (рис. 7.11).

А.: Нашлось 12 программ, среди них на первой странице — только 6 редакторов.

В.: А что это справа? Похоже, оценки программ... Слушай, а что лучше — «Good» или «Cool»?

А.: Понятия не имею.

Примечание

Еще одно затруднение. Зачем нужны оценки, если пользователь не может догадаться, какая из них выше?

В.: Ладно, сперва посмотрю еще следующую страницу... Упс! (рис. 7.12)

А.: Действительно, «упс». Что, подождем пять минут?

В.: Да ну их, знаю я эти «пять минут». Неделю назад один знакомый сайт тоже закрылся на пять минут — так до сих пор не работает.

А.: А как же твой редактор?

В.: Ладно, потом еще где-нибудь поищу.

Примечание

Пользователь ушел с сайта неудовлетворенным, его задача не выполнена. В то же время на сайте **www.listsoft.ru** есть несколько ссылок на текстовые редакторы, которые могли бы подойти Вадиму. Вадим не смог найти ни одного из этих редакторов, а значит, удобство сайта не на высоте.

Например, один из редакторов, подходящих по заявленным критериям — MiBEditor. Но в его описании (см. рис. 7.11) не сказано ни слова ни про Unicode, ни про UTF-8, поэтому Вадим не обратил на него внимания.

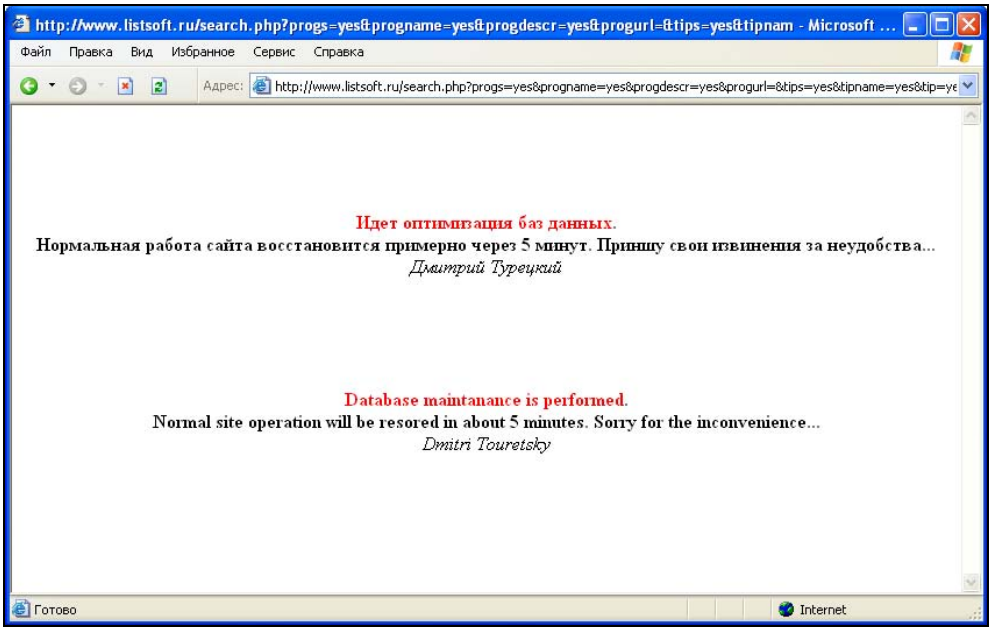


Рис. 7.12. Упс!

Этот неформальный диалог — пример самого настоящего тестирования удобства сайта. Самое главное в этом тестировании — то, что удалось пронаблюдать, как реальный пользователь действительно взаимодействует с сайтом и какие затруднения он при этом испытывает.

Конечно, по реакции одного пользователя нельзя делать окончательные выводы. Для полноценного тестирования нужно больше пользователей и более широкий спектр их задач. Однако результаты единственного пользователя по крайней мере дают пищу для размышления и позволяют выдвинуть ряд гипотез.

Одна из таких гипотез заключается в том, что описания программ составлены недостаточно удачно, они не отражают существенной информации. Посмотрим, например, как описан упомянутый MiBEditor:

Перспективный текстовый редактор. Поддерживает подсветку синтаксиса, умеет работать с кодировками, может быстро вставлять HTML-теги, шаблоны, дату и специальные символы. Имеет множество дополнительных возможностей и приятный интерфейс.

Первое предложение — совершенно неинформативно. Что значит «перспективный»? Это субъективная оценка, а не описание действительных свойств редактора. Во фразе «Поддерживает подсветку синтаксиса» — лишнее слово «поддерживает». Можно написать просто: «Подсветка синтаксиса». Далее, «умеет работать с кодировками» — еще одна бессмысленная фраза. Разумнее было бы написать: «Кодировки: WIN, DOS, KOI, ISO, транслит, экспорт в Unicode и UTF-8». Наконец, последнее предложение также лишнее.

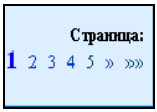
Не стоит писать о «множестве дополнительных возможностей», лучше указать сами эти возможности.

Исправить описание программы можно было бы, например, следующим образом:

Текстовый редактор. Кодировки: WIN, DOS, KOI, ISO, транслит, экспорт в Unicode и UTF-8. Синтаксическая подсветка, быстрая вставка тэгов HTML и специальных символов, HEX, регулярные выражения, простые расчеты.

Дополнительно

Еще один заметный недостаток сайта — употребление выражений, которые понятны разработчику, но могут быть непонятны пользователю. Прежде всего, это оценки (в самом деле, что лучше — «Good» или «Cool»?). Также взгляните на ссылки, позволяющие перемещаться между отдельными страницами:



При наведении указателя мыши на последнюю ссылку (две правые кавычки) появляется всплывающая подсказка: **Следующий блок**. Что это означает? Опытным путем выясняется, что эта ссылка перемещает на 5 страниц вперед. Но заранее об этом никак нельзя догадаться.

Вместе с тем, нельзя не отметить ряд бесспорных достоинств ListSoft. Во-первых, рядом с описанием каждой программы указан ее статус, объем загружаемого файла, доступные языки, операционные системы и дата последнего обновления данных об этой программе на ListSoft. Во-вторых, для каждой программы доступно расширенное описание вместе с отзывами пользователей и ссылкой на страницу программы. И, наконец, пользователь может сам выбрать цветовую схему сайта из нескольких вариантов.

Belle!

Поиск нот популярного трио из «Notre-Dame de Paris»

Où est-elle,
Notre Esmeralda?
Les rues de Paris
Sont tristes sans elle.

L. Plamondon. Notre-Dame de Paris[1]

Пользователь — **Светлана Николаевна**, немолодая учительница музыкальной школы. Она недавно освоила основы компьютерной грамотности, слушает музыку с помощью компьютера, набирает документы в Microsoft Word,

[1] Где она — где Эсмеральда?
Пусты без нее проспекты Парижа. (*фр.*)

пользуется электронной почтой и посещает некоторые сайты.

С. Н.: Мне кто-то говорил, что на Musicals.Ru есть ноты «Belle» и других номеров из «Notre-Dame».

А.: Вы можете это легко проверить.

С. Н.: В самом деле... Сейчас, я надену очки... Значит, набираю `www.musicals.ru` ... или нужно просто `musicals.ru`?

А.: Если точно не уверены — лучше набирайте с «www». В конце концов, если не сработает, всегда можно исправиться.

С. Н.: Действительно... Я-то привыкла к печатной машинке, а там, знаете, ошибаться нельзя — чуть что не так, придется всю страницу перепечатывать.

Тем временем открывается главная страница сайта (рис. 7.13, *а*).

С. Н. (видит фотографию в верхнем левом углу): А что, тут только бродвейские мюзиклы? Может быть, здесь нет «Notre-Dame»? Ох, как мелко написано, не разберу... Сейчас переменю очки...

А.: Шрифт можно сделать крупнее — вот в этом меню.

С. Н.: Сейчас... (Рис. 7.13, *б*.) Ой, тут что-то справа вверху вылезло... Я что-то не так сделала? Сайт испортился?

А.: Ничего страшного. Если это вам не мешает — не обращайте внимания.

Примечание

При изменении размера шрифта верстка сайта немного «поползла». Это, безусловно, промах разработчиков. Еще один промах — фиксированная ширина страницы. Гораздо лучше делать «резиновые» страницы, ширина которых подстраивается под текущий размер окна.

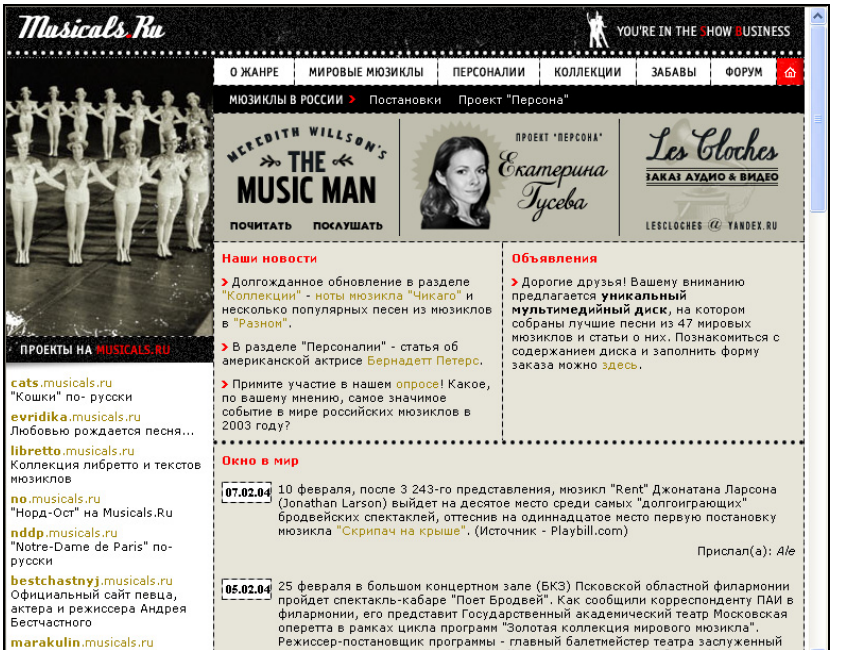
С. Н.: Ой, а тут сверху размер не изменился... Что тут написано? Как-то оно еще и размыто... «О жанре», да? Это ссылка? (Подводит указатель мыши.) Да, ссылка. Нет, это мне не нужно...

Примечание

Сразу две неприятности. Во-первых, текст **О жанре** заключен в изображение, и его размер не увеличился при масштабировании. Во-вторых, пользователь понял, что это ссылка, только после того, как навел указатель мыши.

С. Н.: «Мировые мюзиклы», «Персоналии», «Коллекции»... Что такое «Коллекции»? Коллекции чего? И что такое «Забавы»?

А.: Щелкните, посмотрите. Мне самому интересно.



а

б

Рис. 7.13. www.musicals.ru: главная страница в обычном режиме (*а*) и после масштабирования шрифта (*б*)

С. Н.: Ой, тут какие-то игры... Нет, мне надо назад. Как тут назад?

Светлана Николаевна пытается щелкнуть по логотипу Musicals.Ru в левом верхнем углу, но он не является ссылкой, и ничего не происходит.

С. Н.: Как же назад? Я не знаю, как назад!

А.: Нажмите <Backspace>. Эта клавиша возвращает на предыдущую страницу.

С. Н.: Спасибо. Но где же тут ноты? «Мюзиклы в России»... (Пытается щелкнуть по надписи **Мюзиклы в России** под панелью навигации.) Ой, это опять не работает... Я что-то не так делаю?

А.: Нет, все хорошо. Это действительно не ссылка, она и не должна работать.

С. Н.: Да? А почему тогда «Забавы» — это ссылка, а «Мюзиклы в России» — не ссылка?

Примечание

Автор признается, что этот вопрос поставил его в тупик. Разработчикам сайта действительно следовало бы задуматься над ним.

С. Н.: Ладно, но где же ноты? Может быть, их здесь нет? Вообще, как попасть на «Notre-Dame»? Наверное, все-таки «Мировые мюзиклы»... (Щелкает по ссылке.) Точно — вот и «Notre-Dame». (Снова щелкает по ссылке.) Нашла! (См. рис. 7.14.)

С. Н. (читает): Так, «Мюзиклы не всегда были популярны во Франции»... «Гораздо охотнее посмотрели бы музыкальный спектакль»... Нет, это что-то не то. Это рассказ о мюзикле, нот здесь нет. (Нажимает <Backspace>.)

А.: Стойте! Может быть, ту страницу стоило просмотреть до конца.

Примечание

Здесь автор нарушает важное правило тестирования удобства: тестирующий не должен вмешиваться в работу пользователя с сайтом. Светлане Николаевне не пришло в голову посмотреть страницу до конца — и, скорее всего, на этом ее поиски были бы закончены. Автору стало жалко Светлану Николаевну — в результате «чистые» условия тестирования были нарушены, зато учительница смогла приблизиться к своей цели.

С. Н.: Да, и правда — тут «Ноты», «Миди-файлы», «MP3». Ой, спасибо! Вот они, ноты! (См. рис. 7.15.)

С. Н.: Вот тут и «Belle», и не только... Да, «Cathedrales» я бы тоже у себя сохранила... Вообще, очень приятная музыка. Знаете, можно даже устроить у нас в школе вечер «Notre-Dame de Paris». Разучить несколько номеров... Что там

Musicals.Ru YOU'RE IN THE SHOW BUSINESS

О ЖАНРЕ | МИРОВЫЕ МЮЗИКЛЫ | ПЕРСОНАЛИИ | КОЛЛЕКЦИИ | ЗАБАВЫ | ФОРУМ

МЮЗИКЛЫ В РОССИИ > Постановки Проект "Персона"

ЗАМЕТКИ НА ПОЛЯХ

"Собор Парижской Богоматери" - первое значительное произведение Виктора Гюго. Издатель Гюго хотел получить книжку в духе романов Вальтера Скотта и установил жесткие сроки. Роман был написан в 1831 году всего за четыре с половиной месяца, в отличие от последовавших за ним "Отверженных", над которыми писатель трудился целых семнадцать лет.

Публикация романа вызвала у современников автора всплеск интереса к средневековой архитектуре.

Собор Парижской Богоматери - одно из самых грандиозных сооружений в мире. Его длина - 130 м, ширина - 108 м, высота башен - 69 м, высота сводов - 39 м, здание вмещает 10,000 человек. Строительство собора, являющегося одним из самых знаменитых шедевров мирового искусства, началось в 1163 году и закончилось в 1345. Имя первого архитектора

Мировые мюзиклы

42nd Street
Aspects of Love
By Jeeves
Cats
Chess
Chicago
Dance of The Vampires
Elisabeth
Evita
Fiddler on the Roof
Grease
Into The Woods
Jekyll & Hyde
Miserables, Les
Miss Saigon
Mozart!
Music Man, The
My Fair Lady
Notre-Dame de Paris
Oklahoma!
Oliver!
Phantom of the Opera, The
Scarlet Pimpernel, The
Sound of Music, The
Starlight Express
Sunset Boulevard
Sweeney Todd, the Demon Barber of Fleet Street
Tommy
Whistle Down The Wind
Witches of Eastwick, The
Орфей и Эвридика
Фламандская легенда

Notre-Dame de Paris / Собор Парижской Богоматери (1998)
музыка: Richard Cocciante (Риккардо Коччианте)
либретто: Luc Plamondon (Люк Пламондон)

Мюзиклы не всегда были популярны во Франции. Еще несколько лет назад даже знаменитые шоу Эндрю Ллойд-Уэббера, идущие по всему миру, принимались местной публикой сдержанно. Возможно, это было еще одним проявлением "великофранцузского шовинизма" - французы гораздо охотнее помотрели бы музыкальный спектакль на родную, близкую им тему. Шонберг и Бублиль учли эту особенность французского зрителя, и их работы - "Французская революция" и "Отверженные" сразу же завоевали любовь соотечественников. Более того, эти мюзиклы были хорошо приняты и за рубежом. Правда, "Французскую революцию" поставили за пределами страны лишь однажды - в соседней Германии, а вот "Отверженные" стали настоящей мировой сенсацией и успешно конкурировали с уэбберовскими блокбастерами. В 1998 году все изменилось.

Собор Парижской Богоматери - самый знаменитый символ Франции и ее столицы, не считая Эйфелевой башни. А если вспомнить о крупнейшем французском писателе - Викторе Гюго, который прославил Собор в своем одноименном романе, то мюзикл "Notre-Dame de Paris" был обречен на успех, по крайней мере, на родине Гюго. В конце концов, в этой истории есть все, что нужно зрителю, то есть, говоря словами стоппардовского Актера, "кровь, любовь и риторика".

Рис. 7.14. www.musicals.ru: страница «Notre-Dame»

Картина восемнадцатая (373 МБ)
Заключительный выход (1,24 МБ)
Содержание (113 МБ)

Notre-Dame de Paris

Ouverture (71 КБ)
Le temps des cathedrales (527 КБ)
Les sans-papiers (750 КБ)
Intervention de Frollo (110 КБ)
Bohemienne (986 КБ)
Esmeralda tu sais (85 КБ)
Ces Diamants-la (303 КБ)
Dechire (622 КБ)
Anarkia (212 КБ)
Belle (817 КБ)
Ma maison c'est ta maison (370 КБ)
Ave Maria paien (267 КБ)
Vivre (657 КБ)

Ноты любезно предоставлены Полиной Клейменовой. Остальные номера из мюзикла "Notre-Dame de Paris" можно скачать на сайте nddp.musicals.ru.

Oklahoma! (рус)

Увертюра (1,02 МБ)
О, что за день благодатный! (750 КБ)
Выход Лори - часть первая (895 КБ)
Выход Лори - часть вторая (914 КБ)
Канзас-Сити (1,43 МБ)
Реминисценция (278 КБ)
Я не могу сказать - нет! (709 КБ)

Рис. 7.15. www.musicals.ru: ноты найдены

еще такого было? Вот, вспомнила — заключительная ария Квазимодо... (Напевает.) Да, очень мелодично. Она здесь есть? Так, вот, вижу: «Остальные номера из мюзикла можно скачать на сайте»... А почему все сюда не поместили? Почему одни номера здесь, другие — там?

Примечание

Отметим, что автор не смог разъяснить и этого недоумения. Кроме того, ссылка на «остальные номера» вела вовсе не на ту страницу, с которой можно загрузить ноты, а на главную страницу сайта **nddp.musicals.ru**.

С. Н.: Ой, спасибо, вы мне так помогли! Без вас я бы ничего на этом сайте не нашла!

Это тестирование показывает ряд традиционных проблем, возникающих у пользователей. Заголовки разделов составлены невнятно — в самом деле, что значит «Коллекции»? Текст заключен в изображение, что мешает его масштабировать. Логотип в верхнем левом углу не является ссылкой на главную страницу.

Проблема, из-за которой Светлана Николаевна едва не прекратила поиски, — в том, что в верхней части страницы нет никакой информации о содержании этой страницы. Между тем, «над сгибом» (то есть в той части страницы, которая доступна без прокрутки) следовало бы поместить ссылки на ее отдельные элементы. Например, так:

Notre-Dame de Paris

О мюзикле | Либретто | Ноты | MIDI | MP3 | Ссылки

Дополнительно

Еще одна проблема обнаружилась, когда ноты были загружены. Они были сжаты архиватором RAR, в то время как «общепринятым» архивным форматом (по крайней мере, для Windows) уже давно является ZIP. Светлана Николаевна не смогла «распаковать» файл привычным ей способом, и ей снова пришлось обращаться к помощи автора.

В самом деле, если вы предлагаете пользователям загружать заархивированные файлы — путь это будут файлы в ZIP, а не в каком-либо другом формате.

С другой стороны, положительное качество Musicals.Ru заключается в том, что пользователь может выполнить одну и ту же задачу разными способами. Например, он может добраться до нот «Belle» не только так, как это сделала Светлана Николаевна, но и перейдя с главной страницы сразу в раздел **Коллекции**. (Конечно, при условии, что пользователь догадается об этом!)

Примечание

Если пользователь не смог найти на сайте что-то, что там есть, не догадался о какой-то возможности сайта или запутался в его структуре — это ошибки не пользователя, а разработчиков.

Таким образом, суть тестирования удобства — в том, чтобы наблюдать, как реальные пользователи действительно работают с сайтом и какие затруднения у них при этом возникают.

Дополнительно

У многих пользователей, которых приглашают принять участие в тестировании, возникает ощущение «студента на экзамене» — как будто тестируют не сайт, а их самих. Они очень боятся ошибиться, не справиться с задачей; они склонны спрашивать: «Правильно ли я делаю?» В результате их работа с сайтом происходит совсем не так, как в «естественных» условиях, где за ними никто не наблюдает.

Поэтому важно правильно настроить пользователей, убедить их в том, что их никто не экзаменует, и что трудности, с которыми они сталкиваются, — свидетельство несовершенства сайта, а не их «неправильных» действий.

Наиболее полезные результаты тестирования получаются тогда, когда пользователей удается и в самом деле застать в «естественных условиях», как в вышеописанных примерах, когда они не подозревают о том, что участвуют в тестировании сайта. К сожалению, такое тестирование удается очень редко, и его практически невозможно организовать специально.

Примечание

Разработчики тоже являются пользователями. Попробуйте следить за тем, как вы сами посещаете сайты и с какими затруднениями вы при этом сталкиваетесь. Бывают ли случаи, когда вам не удается решить задачу, ради которой вы зашли на сайт? Разберитесь, почему так произошло. Не получится ли так, что пользователь вашего сайта испытает те же проблемы, которые вы сами испытывали на других сайтах?

Но тестирование удобства не ограничивается отслеживанием явных затруднений. Занимаясь тестированием, вы также должны подумать об оптимизации работы пользователя. Даже если пользователь не чувствует проблем — нельзя ли сделать так, чтобы он выполнял свои задачи еще быстрее и проще?

- **Избавьте пользователя от прокрутки.** Посмотрите на сайт строительной компании (рис. 7.16). Шапка в верхней части страницы занимает почти треть окна, огромные отступы сверху и снизу от заголовков **Новости** и **Пресса о нас**, также большие отступы между новостями. Из-за этого

большая часть страницы ушла «под сгиб». Чтобы прочесть ее, требуется пользоваться полосой прокрутки. В то же время, если уменьшить шапку и избавиться от отступов, то можно уместить в окне практически всю полезную информацию: весь блок **Пресса о нас** и не меньше четырех последних новостей.

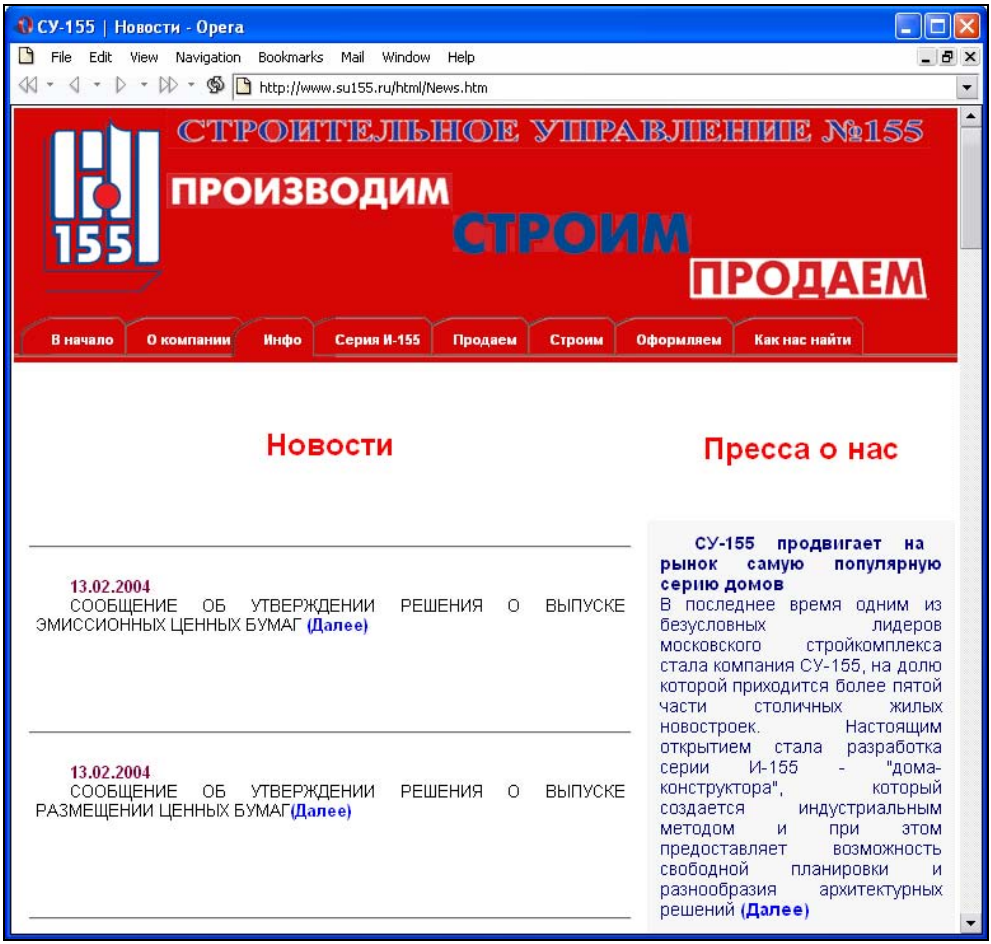


Рис. 7.16. www.su155.ru:
нерациональное использование площади окна

- **Позвольте пользователю меньше ждать.** Посмотрите, как устроен форум на сайте любителей ролевых игр «Александр VI» (рис. 7.17). При просмотре в режиме ленты вверху отображаются наиболее старые сообщения, а внизу — наиболее новые.

Рис. 7.17. Форум на сайте **www.alexander6.ru**

В результате первыми загружаются старые сообщения, которые пользователь уже видел, и ему приходится ждать, пока лента загрузится до конца. Если в обсуждении накопилось несколько десятков сообщений, ожидание затягивается надолго. Кроме того, чтобы добраться до наиболее свежих сообщений, приходится прокручивать страницу.

Разумнее было бы сортировать сообщения в обратном порядке — чтобы пользователь сразу видел самые новые реплики. Впрочем, еще лучше было бы предоставить пользователю возможность настроить интерфейс форума — в том числе указав предпочтительный порядок сортировки.

- **Избавьте пользователя от лишних переходов по ссылкам.** Например, если в тексте на сайте употребляются специальные термины, то недостаточно компетентные пользователи захотят узнать значение этих терминов и станут искать определения на этом же сайте или в сетевых энциклопедиях. Вы можете облегчить им задачу, если прямо от специальных терминов сделаете ссылки на страницы с определениями, например:

Ян Чихольд: В основе совершенного художественного оформления лежит абсолютная гармония всех элементов. Суть гармонии — соотношения пропорций. Пропорции встречаются во всем: в размере полей, во взаимных соотношениях четырех полей книжной полосы, в соотношении пробелов между строками и в том, насколько удалена колонцифра от занятой текстом полосы, в размере разрядки набранных прописными буквами строк относительно сплошного набора и, не в последнюю очередь, в выключке слов. Только благодаря непрерывным упражнениям и строжайшей самокритике, путем постоянной учебы мы сможем сформировать у себя вкус к совершенной работе и понимание ее.

纸 *Дополнительно*

В литературных текстах лучше смотрятся более скромные ссылки на комментарии — например, в виде звездочки после специального слова:

…При государыне служил Екатерине,
А в те поры все важны! В сорок пуд…
Раскланяйся — тупеем* не кивнут.
Вельможа, в случае* — тем паче;
Не как другой, и пил, и ел иначе.
А дядя! Что твой князь? что граф?
Сурьезный взгляд, надменный нрав.
Когда же надо подслужиться,
И он сгибался вперегиб:
На ку́ртаге* ему случилось обступиться;
Упал да так, что чуть затылка не пришиб…

Обратите внимание, что здесь нет необходимости обозначать ссылки разными пометками, как в книгах (*, **, *** и т. д.).

Примечание

Если одни и те же термины повторяются в тексте много раз, не имеет смысла делать каждый из них ссылкой; можно поместить рядом с текстом их список, например:

Я. Нильсен

10 главных ошибок при разработке сайтов в 2003

...

5. Чрезмерно подробное альтернативное описание (alt)

Многие сайты начали обращать внимание на пользователей с ограниченными возможностями и следовать принципам доступности, таким, как указание альтернативного описания (alt) для изображений. К сожалению, разработчики некоторых сайтов не понимают, что альтернативное описание — это элемент пользовательского интерфейса, а не просто проявление политкорректности. Это описание должно помочь слепым людям (и тем, кто по каким-то причинам не видит изображений) перемещаться по сайту и взаимодействовать с ним. Текст должен описывать суть изображения в той мере, в какой это необходимо для эффективной работы пользователей с сайтом. Нет необходимости описывать второстепенные детали изображения.

См. также:

- Альтернативное текстовое описание изображений (alt)
- Устройство озвучивания экрана
- Политкорректность

Например, мне попадался сайт, где в качестве альтернативного текстового описания изображения был следующий текст: «Ссылка на главную страницу, оформленная как логотип IDEAS: два облачка пара вокруг идеи и восходящее солнце на заднем плане». Возможно, было бы целесообразно поместить текстовое описание логотипа где-то на сайте, чтобы его смогли прочесть невидящие люди, желающие узнать, как он выглядит. Но нет оснований заставлять каждое устройство озвучивания экрана озвучивать количество облачков пара всякий раз, когда открывается новая страница.

Устройству озвучивания требуется много времени, чтобы прочесть целых девятнадцать слов, и такое избыточное описание мешает пользователям усвоить первостепенную информацию: это графическая ссылка на главную страницу.

Краткость похвальна, когда речь идет о тексте для Сети. Она еще более критична, если иметь в виду устройства озвучивания экрана.

- **Предоставьте пользователю разные способы достижения одной и той же цели.** Возможно, разные пользователи будут искать одну и ту же страницу в разных разделах сайта. Так поместите ее в каждый из этих разделов! В отличие от печатных изданий, на сайте это легко осуществимо. Предоставьте на каждой странице побольше ссылок на страницы, связанные с

ней по тематике, или просто на те страницы, на которые пользователи предпочитают переходить с данной. Обеспечьте средства поиска, карту сайта и другие элементы вспомогательного аппарата.

В самом деле, вместо того, чтобы навязывать пользователю определенный способ решения задачи, ради которой он пришел на сайт, дайте ему возможность решить эту задачу наиболее удобным для него способом.

7.1.4. Как вам будет угодно
Персонализация сайтов

> Может быть, только теперь астролетчики полностью, всем существом поняли, что самое важное во всех поисках, стремлениях, мечтах и борьбе — это человек. Для любой цивилизации, любой звезды, целой галактики и всей бесконечной Вселенной главное — это человек, его ум, чувства, сила, красота, его жизнь!
>
> *И. Ефремов. Сердце змеи*

Большое распространение приобрела персонализация сайтов, то есть настройка их интерфейса под конкретного пользователя. Существует два вида персонализации:

- **по запросу пользователя** — сайт предлагает пользователю установить определенные настройки, которые обычно сохраняются с помощью cookies («ключиков») в браузере;

  Дополнительно

  На самом деле, настройки пользователя сохраняются не у него в браузере, а на веб-сервере. А «ключик» нужен только для того, чтобы при следующем входе система опознала этого пользователя и загрузила сайт с сохраненными настройками.

- **автоматическая** — программное обеспечение сайта анализирует поведение пользователя и автоматически «подстраивается» под него.

Ранее приводился пример персонализации по запросу пользователя на ListSoft: пользователь может выбрать цветовую гамму сайта из нескольких возможных. Многие другие сайты предоставляют разные возможности настройки, исходя из задач, которые выполняют пользователи. Например, в интернет-магазине ведется учет заказов пользователя, так что он в любой момент может увидеть, какие товары уже просмотрел, какие из них заказал и какие покупки совершил. Пользователь может «отложить» корзину с товарами

и покинуть интернет-магазин, а при следующем посещении этого сайта он сможет вернуться к своей корзине и продолжить покупки.

Примечание

На многих зарубежных сайтах можно увидеть ссылку **My ...** (например, **My Yahoo**, **My Netscape**). Эта ссылка ведет на страницу или даже в целый раздел, сконфигурированный в соответствии с запросами и предпочтениями конкретного пользователя. Разумеется, для разных пользователей раздел **My ...** будет разным: сайт узнает пользователя по «ключику» или предлагает ему ввести регистрационное имя и пароль.

На отечественных сайтах также начинают распространяться подобные средства — например, поисковая система Яндекс предлагает службу **Мой Яндекс**.

Если ваш сайт обеспечивает персонализацию по запросу, то необходимо уведомлять пользователя о том, что он видит специально настроенную страницу, а не то, что видит «посторонний» посетитель. Во многих случаях в качестве такого уведомления выступает имя пользователя, размещенное на странице; иногда оно включается в приветственную надпись вроде **Добрый вечер, Василий Пупкин!** (рис. 7.18).

Рис. 7.18. Мой Яндекс приветствует пользователя по имени

Примечание

Выше говорилось о том, что «Добро пожаловать» и другие «приветственные речи» крайне нежелательны. Может показаться, что автор противоречит сам себе. Однако это не так. На странице, предназначенной *для всех* пользователей, специальные приветствия действительно нежелательны, т. к. не предоставляют пользователю ничего полезного и только занимают место, где могли бы располагаться более нужные элементы. А на *персонализованной* странице главное — не приветствие, а собственно имя пользователя: именно оно говорит о том, что это персонализованная страница. Можно обойтись без приветствия, например:

- Мой Яндекс: Василий Пупкин;
- Василий Пупкин — Мой Яндекс.

Дополнительно

Здесь, кстати, всплывает проблема падежей. Пока имя пользователя употребляется в контексте именительного падежа (как в приведенных примерах), все нормально. Но если попробовать поставить его в другой контекст, может получиться неприятность: «Почта для Василий Пупкин» или «Персональная информация о Василий Пупкин». Если вы разрабатываете систему, где имя пользователя подставляется автоматически, убедитесь, что оно везде требуется в именительном падеже. В противном случае вам придется неоправданно усложнять систему, занося в нее правила склонения русских существительных.

При автоматической персонализации, наоборот, пользователя не надо специально предупреждать о том, что сайт «подстраивается» под него. Если эта персонализация проходит удачно, пользователь будет доволен, но даже не заметит того, что сайт изменился лично для него. Но если предупредить пользователя — он скорее будет обеспокоен тем, что «за ним следят».

Простой пример автоматической персонализации — интернет-магазины. Во многих из них пользователю, выбравшему ряд товаров, предлагают также посмотреть на дополнительные товары, имеющие отношение к выбранным.

Можно представить себе и сайт новостей, на котором есть автоматическая персонализация. Этот сайт следит за тем, на какую тематику пользователь просматривает больше новостей, и в дальнейшем выдвигает такие новости на первый план — возможно, даже добавляя к ним дополнительные материалы, до которых другому пользователю придется специально добираться по ссылкам.

Грамотная персонализация помогает сделать сайт удобнее для каждого конкретного пользователя. Но, обеспечивая персонализацию, разработчики не должны забывать и об общих принципах удобства, справедливых для всех пользователей: требуется сократить время ожидания, избежать прокрутки и т. д.

Примечание

Неудачно организована регистрация и персонализация на сайте «Коммунивер». Пользователь, желающий зарегистрироваться, вводит свои данные — и видит текст (рис. 7.19):

Ошибка: Нарушение ограничения OPEN_INFO_MUST_BE

Что это значит? Какую конкретно ошибку допустил пользователь и что ему теперь делать, чтобы исправить ее? Узнать об этом невозможно.

Если вы хотите сообщить пользователю об ошибке — сделайте это так, чтобы он вас понял и разобрался, что ему делать дальше. Не злоупотребляйте техническими терминами и сложными объяснениями, иначе вы потеряете пользователей.

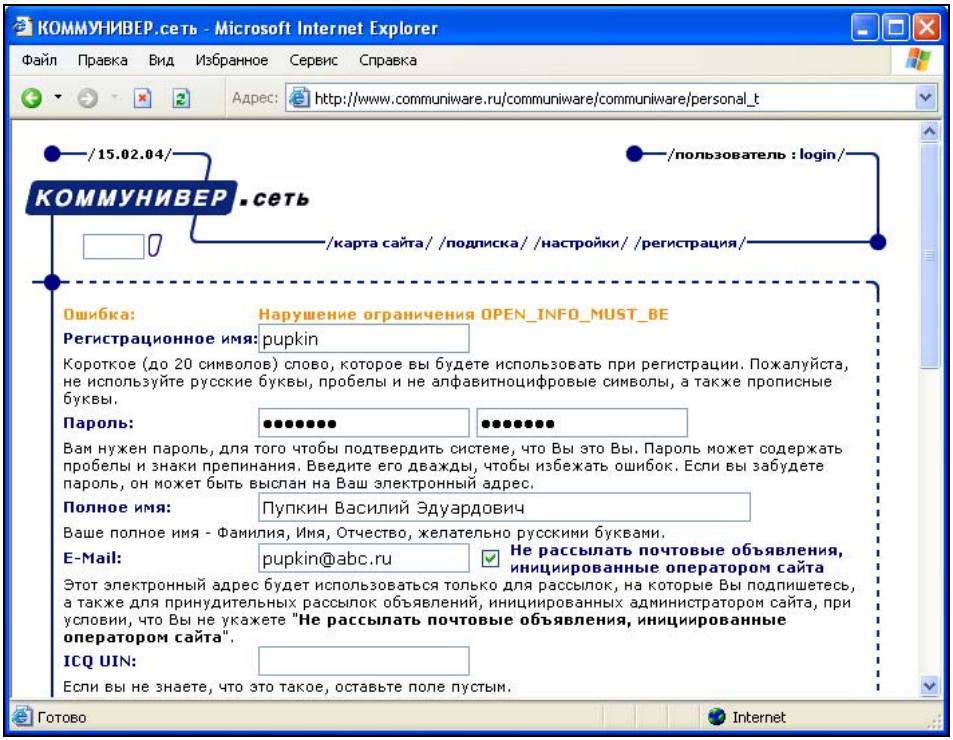


Рис. 7.19. www.communiware.ru: что делать пользователю?

Сочетайте общие принципы удобства с хорошо продуманными средствами персонализации — и пользователи останутся довольны вашим сайтом.

7.2. Это доступно?

Критерии доступности текстового содержания сайтов

> «Что, если, — думал он на другой день вечером, бродя по Петербургу, — что, если старая графиня откроет мне свою тайну! — или назначит мне эти три верные карты!»
>
> *А. С. Пушкин. Пиковая дама*

Все пользователи разные, и они применяют разные устройства для взаимодействия с сайтами. Разработчик должен это учитывать. Конечно же, невозможно сделать сайт, с которым сможет работать абсолютно любой пользователь с абсолютно любым устройством, но можно обеспечить работоспособность сайта для достаточно широкого круга пользователей.

Существуют три основных условия, при соблюдении которых доступность сайта можно считать удовлетворительной:

Внимание!

Три основные условия доступности сайта:

- пользователь может прочесть весь текст, имеющийся на сайте;
- пользователь может пользоваться навигацией по сайту;
- пользователь может заполнять и отправлять формы, имеющиеся на сайте.

Доступность не ограничивается соблюдением только этих трех условий. Далее в этой главе будут перечислены другие требования доступности. Но без выполнения трех основных условий о доступности сайта вообще не имеет смысла говорить.

На доступность сайта влияют две группы свойств:

- устройства просмотра сайтов, аппаратное и программное обеспечение, настройки браузера. Это «машинный фактор» доступности;
- свойства самого пользователя, его физические возможности и особенности восприятия. Это «человеческий фактор».

7.2.1. Найдите десять отличий
«Машинный фактор» доступности

> Итак, в разных концах мира человечество занято мыслями не о самоусовершенствовании, но об усовершенствовании механических чудовищ, которые должны заменять людей во многих работах.
>
> *Н. Рерих. Роботы*

Существует множество устройств, с помощью которых пользователи посещают сайты. Мы рассмотрим здесь основные различия между этими устройствами, влияющие на отображение текстового содержания.

Устройства просмотра

Неуклонно увеличивается доля пользователей, просматривающих сайты не на обычном персональном компьютере, а с помощью особых устройств — наладонных компьютеров, ноутбуков, сотовых телефонов и т. д. У многих пользователей есть по несколько устройств, с помощью которых они просматривают сайты. Один и тот же пользователь в разное время может посещать сайт со своего домашнего компьютера, с ноутбука, с карманного компьютера.

Кроме того, в нашей стране возникает все больше мест, из которых можно получить доступ к Интернету. Выход в Сеть появляется на предприятиях, в учреждениях, в учебных заведениях и библиотеках. Распространяются интернет-кафе и другие службы общедоступного доступа в Сеть. В любом крупном городе легко можно найти доступ к Всемирной Паутине, и люди все чаще пользуются этой возможностью.

Если у себя дома пользователь успешно работает с сайтом, а потом, оказавшись в интернет-кафе или в гостях, не может выполнить на сайте своих задач, — скорее всего, доверие к сайту будет подорвано.

Поэтому для разработчиков становится важно обеспечить эффективную работу сайта для разных устройств просмотра. Всемирная Паутина становится все больше похожа на телефонную сеть, где, чтобы позвонить нужному абоненту, обычно не требуется какой-то конкретный телефонный аппарат или конкретный телефонный узел. Звонок на любой номер можно совершить с любого аппарата, подключенного к телефонной линии.

В идеале разработчики сайтов должны стремиться к такому же положению дел. Нужно, чтобы пользователь мог эффективно работать с нужными ему сайтами на любом устройстве. Это не значит, что сайт должен выглядеть совершенно одинаково на всех устройствах; это просто невозможно. Но возможно сделать так, чтобы на всех устройствах выполнялись три основных условия доступности.

Бывает, что разработчики сайта применяют решения, затрудняющие выполнение этих задач. К таким решениям, в частности, относятся:

- заключение текста в изображение;
- фиксированный размер шрифта, фиксированная ширина страницы;
- применение Flash и JavaScript для навигации и работы с формами.

Примечание

Один из способов добиться универсальной совместимости — разрабатывать отдельную версию страницы для каждого типа родственных устройств. Например, не имеет смысла просматривать обычные веб-страницы с помощью сотовых телефонов с монохромным дисплеем, поэтому для таких устройств разрабатываются страницы, совместимые с WAP (Wireless Application Protocol — протокол передачи данных, предназначенный в первую очередь для беспроводных устройств; этим термином часто обозначают всю совокупность технологий, применяющихся для просмотра страниц на этих устройствах).

Но к сотовым телефонам добавилось множество других устройств, технически вполне пригодных для отображения полноценных страниц. Разрабатывать специальную версию страницы для каждого из таких устройств — нерационально.

К счастью, одновременно с развитием и распространением «альтернативных» технологий просмотра сайтов происходит и другой процесс: устройства

просмотра все лучше придерживаются стандартов HTML, CSS и других технологий, разработанных Консорциумом W3C. Поэтому первый шаг к универсальности веб-страниц — это их разработка в соответствии со стандартами. Это, конечно, не избавит от всех проблем с совместимостью, но позволит существенно уменьшить их. Кроме того, если страницы соответствуют стандартам — они будут еще лучше совместимы с браузерами и устройствами просмотра следующих поколений.

Дополнительно

Для доступа к сайтам могут использоваться и более специализированные устройства, например:

- текстовые браузеры;
- речевые браузеры;
- браузеры Брайля.

Такие устройства — большая редкость, ими пользуется ничтожно малое количество посетителей. Однако, если позаботиться о совместимости сайта с этими устройствами, автоматически улучшится доступность сайта и для пользователей обычных браузеров.

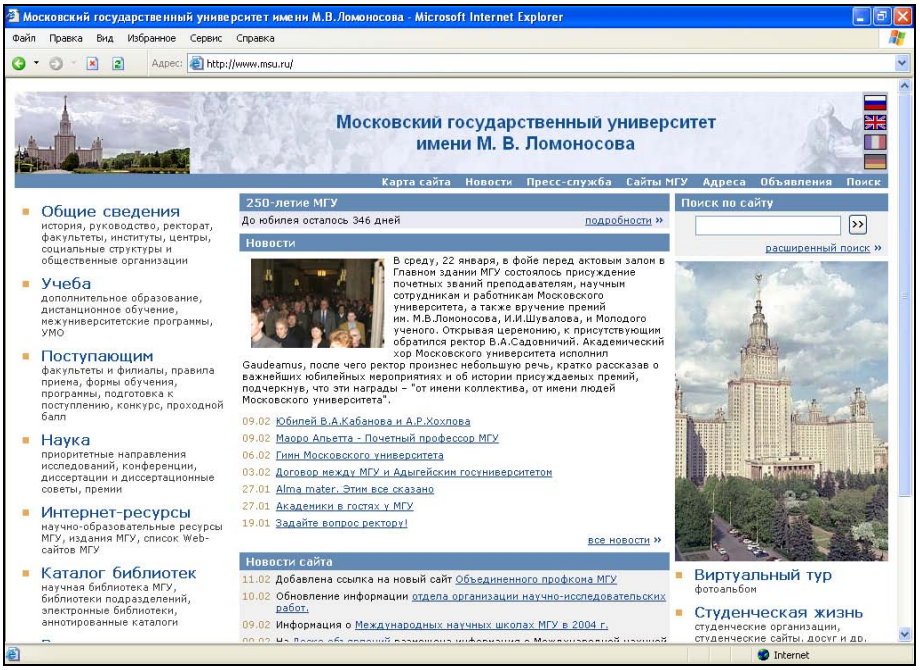
Операционная система

От операционной системы в первую очередь зависит, какие браузеры можно установить под ее управлением. Например, браузер Safari выпускается специально для Mac OS, а Konqueror — для систем класса UNIX. Internet Explorer существует и для Windows, и для Mac OS, однако эти версии заметно отличаются друг от друга — прежде всего по степени поддержки тех или иных элементов HTML и свойств CSS.

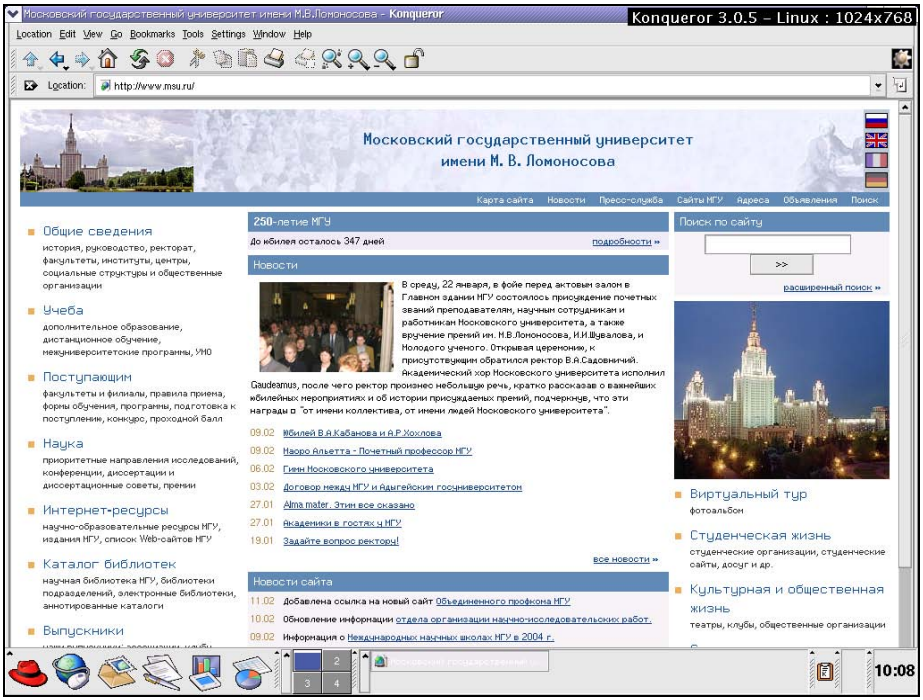
Про браузеры как таковые будет более подробный разговор далее, а сейчас хотелось бы обратить внимание еще на один аспект, связанный с операционной системой. От нее зависит, какие шрифты могут использоваться при отображении веб-страниц. Принципы работы со шрифтами и ассортимент этих шрифтов различаются в разных системах.

Посмотрите на сайт Московского государственного университета (рис. 7.20). По всей видимости, разработчики тестировали его только в Windows, не позаботившись о совместимости с другими операционными системами.

Разница особенно заметна, если посмотреть на верхний заголовок. Например, в браузерах под Linux он стал чрезмерно мелким, и его гарнитура изменилась. Остальной текст претерпел похожие изменения: скажем, в Mozilla под Linux основной текст просто невозможно прочесть, не прибегая к масштабированию. Браузеры под Mac OS отображают шрифты одних и тех же надписей по-разному. Хотя здесь, в отличие от браузеров под Linux, все читается хорошо и выглядит достаточно красиво.

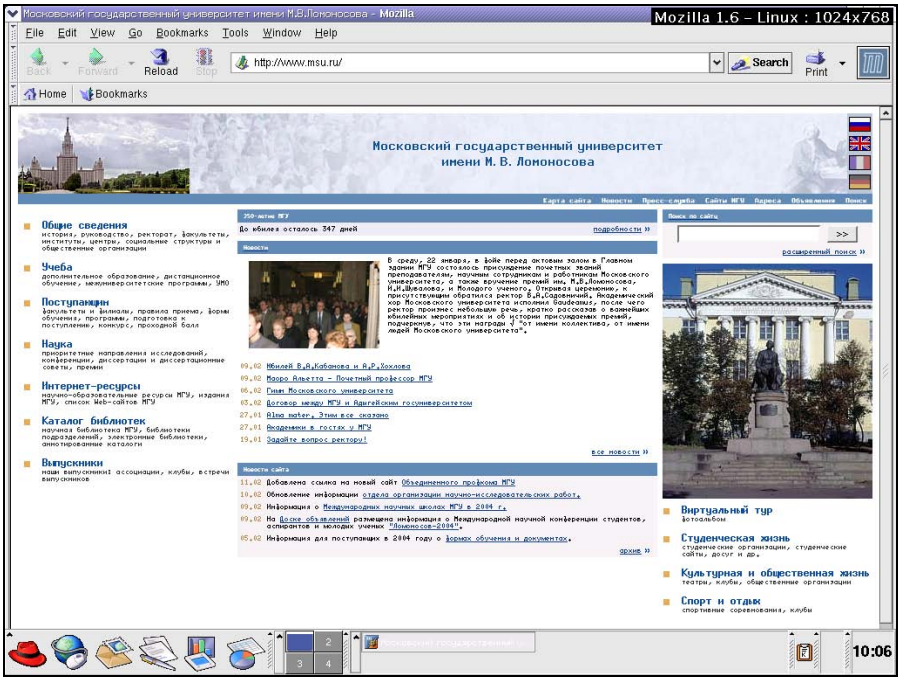


а

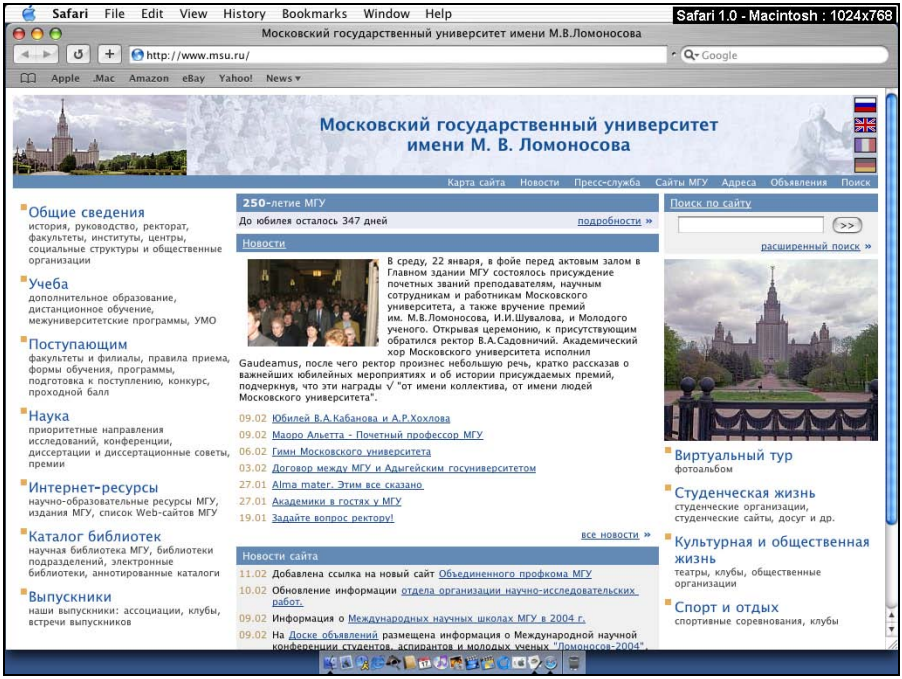


б

Рис. 7.20, *а* и *б*. www.msu.ru: *а* — Internet Explorer, Windows; *б* — Konqueror, Linux



в



г

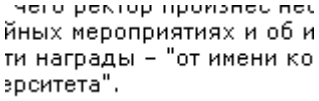
Рис. 7.20, *в* и *г*. www.msu.ru: *в* — Mozilla, Linux; *г* — Safari, Mac

Рис. 7.20. www.msu.ru: *д* — Opera, Mac

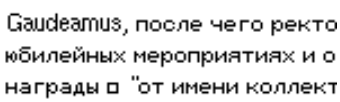
Дополнительно

Обратите внимание на текст главной новости. Короткое тире (после слова «награды») не везде отображается корректно:

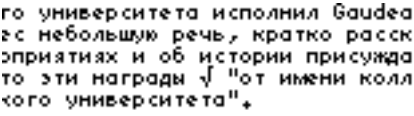
Internet Explorer, Windows



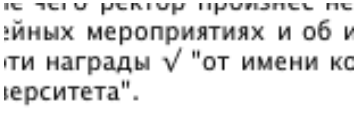
Konqueror, Linux



Mozilla, Linux



Safari, MacOS



Чтобы не происходило таких неприятностей, тире и другие специальные символы нужно вставлять в документ с помощью подстановок (например, короткое тире — `–`, длинное тире — `—`).

Кстати, в данном случае короткое тире еще и неуместно — согласно типографским правилам, здесь должно быть длинное тире.

Примечание

Конечно, у сайта МГУ есть и неоспоримые достоинства. Стройная структура сайта позволяет быстро найти на нем необходимую информацию. В левой части страницы перечислены основные разделы сайта: имя каждого из них не только само по себе информативно, но и снабжено вспомогательной аннотацией. Пользователь, посетивший этот сайт впервые, легко сориентируется на нем. Остальные компоненты главной страницы (новости, форма поиска, меню аппарата и т. д.) также расположены и оформлены достаточно эффективно.

Внимание!

Тестируйте сайт на разных платформах. При тестировании текстового содержания обращайте особое внимание на правильное отображение отдельных символов, на гарнитуру и кегль шрифта.

У начинающих разработчиков нередко возникают следующие вопросы:

- **как сделать так, чтобы текст на всех браузерах и во всех системах отображался одинаково?**

  Это в принципе невозможно, так как разные системы используют разные форматы шрифтов, несовместимые друг с другом. Кроме того, это не нужно: лучше дать каждому пользователю возможность просматривать текст тем шрифтом, который для него наиболее удобен и привычен. Это особенно важно для основного текста — напомним, что его вообще не рекомендуется специально оформлять, следует предоставить браузеру право отобразить его с параметрами по умолчанию. Заголовки и другие надписи можно оформлять, но при этом нет ничего плохого в том, что в разных системах они будут выглядеть немного по-разному. Нужно только следить за тем, чтобы их размер вдруг не становился чрезмерно мелким (как на сайте МГУ в некоторых браузерах);

  Дополнительно

  Указывайте в значении свойства `font-family` список альтернативных шрифтов для разных систем, например:

  ```
  font-family: Verdana, Arial, Helvetica, sans-serif;
  ```

  Не забывайте заканчивать этот список названием стандартного семейства шрифтов.

- **как загрузить мой шрифт на компьютер пользователя?**

  Это также невозможно. Например, если ваш шрифт имеет формат TrueType (что характерно для шрифтов в Windows), его технически не полу-

чится установить в некоторых системах класса UNIX. Максимум, что вы можете сделать, — это разместить на своем веб-сервере установочный файл нужного шрифта и предложить желающим пользователям самостоятельно загрузить и установить его. Впрочем, не нужно рассчитывать на то, что все посетители вашего сайта именно так и поступят.

Дополнительно

В CSS 2 предложен механизм подключения нестандартных шрифтов с помощью конструкции `@font-face`. Система, обрабатывающая такую конструкцию, должна уметь подбирать шрифт по заданным параметрам и даже синтезировать шрифт «с нуля». Современные браузеры пока не умеют эффективно работать с `@font-face`.

Видеорежим

Видеорежим складывается из нескольких взаимосвязанных составляющих:

- длина диагонали экрана;
- тип монитора (например, ЭЛТ или ЖК);
- разрешение экрана;
- частота развертки;
- цветовые характеристики монитора (количество оттенков, цветовая температура, настройки яркости и контраста и т. д.).

Кроме того, на просмотр страницы влияет внешнее освещение.

Разнообразие этих параметров приводит к тому, что одну и ту же страницу разные пользователи видят по-разному. Один и тот же шрифт может показаться на мониторе с электронно-лучевой трубкой слишком мелким и неразборчивым, а на жидкокристаллическом мониторе — чересчур крупным. Пользователь, у которого старый монитор с невысокой частотой развертки, предпочтет скорее распечатать длинный текст, чем читать его с экрана. А обладатель современного монитора без затруднений прочитает текст любого объема.

Разработчик может применять близкие друг к другу цвета, чтобы добиться определенного художественного эффекта. Но на недостаточно ярком мониторе разница между этими цветами не будет заметна. Если различие цветов использовалось не только в оформительских целях, но и должно было нести какую-то информацию, эта информация останется недоступной для пользователя.

Дополнительно

На мониторах компьютеров Macintosh цвета обычно выглядят ярче, чем на мониторах PC. Иногда возникает такая проблема: если сделать цвет того или иного

элемента темнее — на PC он будет слишком мрачным, а если сделать его светлее — на Mac он будет «резать глаза». Лучше не использовать цветов, с которыми случаются такие сложности.

Чтобы обеспечить удовлетворительную работу сайта в разных видеорежимах, необходимо соблюдать рекомендации, многие из которых уже назывались ранее:

- не указывайте размер шрифта в абсолютных единицах (например, в пикселах). Такой текст будет выглядеть слишком крупным на экранах с низким разрешением и слишком мелким — с высоким. Кроме того, в некоторых старых браузерах (например, в Internet Explorer) пользователь не сможет увеличить или уменьшить размер шрифта;
- не заключайте текст в изображение. Растровое изображение имеет фиксированные размеры в пикселах и поэтому тоже выглядит крупным в одних видеорежимах и мелким в других; его также нельзя масштабировать;

Дополнительно

Некоторые браузеры (например, Opera 7) могут масштабировать не только текст, но и изображения. Однако растровые изображения при масштабировании искажаются, и заключенный в них текст может все равно оказаться нечитаемым.

- используйте для текста и фона контрастные цвета, которые гармонично сочетаются друг с другом и не режут глаза. Например, для основного текста неплохо подходит черный цвет на белом фоне. Впрочем, еще лучше сохранить те цвета, которые заданы у пользователя по умолчанию.

  Употребляйте только такие цвета, которые относятся к безопасной палитре;
- разрабатывайте «резиновые» страницы, которые не имеют фиксированной ширины и подстраиваются под ширину окна браузера. Иначе на экранах с маленьким разрешением страница будет вылезать за пределы окна, и пользователю придется прибегать к горизонтальной прокрутке, а на экранах с большим разрешением будет оставаться много пустого пространства.

Внимание!

Тестируйте сайт в разных видеорежимах, на мониторах разных типов, при разных разрешениях и цветовых настройках. Убедитесь, что в распространенных видеорежимах сайт отображается корректно, а в устаревших и экзотических режимах по крайней мере можно прочитать текст, воспользоваться навигацией и отправить формы.

Примечание

Учтите, что окно браузера у пользователя может быть не развернуто на полный экран. Например, если пользователь сравнивает два продукта, один из которых он хочет выбрать для покупки, он может открыть одновременно две страницы с описанием этих продуктов и просматривать их вместе, разместив рядом на экране.

Убедитесь, что при разумном изменении размеров окна функциональность вашего сайта не нарушается.

Браузер

Разные браузеры также по-разному отображают страницы. В *разд. «Операционная система»* уже отмечались некоторые различия в отображении одной и той же страницы разными браузерами. Но даже в одной операционной системе разные браузеры могут вести себя по-разному.

Например, рассмотрим простой маркированный перечень:

```
<ul>
  <li>Красный</li>
  <li>Жёлтый</li>
  <li>Зелёный</li>
</ul>
```

Это обыкновенный, «чистый» HTML. И, тем не менее, разница в поведении браузеров налицо:

Internet Explorer 6	Mozilla 1.6	Opera 6
• Красный • Жёлтый • Зелёный	◆ Красный ◆ Жёлтый ◆ Зелёный	◆ Красный ◆ Жёлтый ◆ Зелёный

Несовместимость проявляется еще больше, если включить в документ специальные символы. Например, здесь в «хлебных крошках» использован символ (U+25BA):

Internet Explorer, Mozilla	Opera
ABC ► Контакты	ABC ► Контакты

Наконец, множество различий наблюдается при оформлении текста с помощью CSS. Например, следующее оформление:

```
font: 600 2em Verdana, Arial, Helvetica, sans-serif;
letter-spacing: 0.15em;
```

отображается в браузерах таким образом:

Internet Explorer 6	Mozilla 1.6	Opera 7
Браузер	**Браузер**	Браузер

Что можно сделать, чтобы устранить различия? Прежде всего необходимо усвоить, что полностью устранить их невозможно. Не стоит даже пытаться добиваться того, чтобы страница во всех браузерах выглядела одинаково.

Дополнительно

Например, Mozilla поддерживает только два значения `font-weight` — `normal` (от `100` до `500`) и `bold` (от `600` до `900`). В двух других браузерах для некоторых шрифтов возможна и промежуточная насыщенность (`600`). В Mozilla отобразить шрифт этой насыщенности невозможно.

Также невозможно заставить Mozilla изображать в качестве маркеров перечня не ромбики, а кружочки. Даже если специально указать:

```
li {
  list-style-type: disc;
}
```

— Mozilla все равно будет помечать пункты ромбиками.

Тем не менее, во многих случаях можно существенно приблизить отображение страницы в разных браузерах к общему «эталону». Вот некоторые из способов сделать это:

- не использовать решений, не работающих в каком-либо из распространенных браузеров. Например, не пытаться отобразить шрифт промежуточной насыщенности между `normal` и `bold`;
- включать в код HTML и CSS дополнительные конструкции, которые не повлияют на внешний вид страницы в одних браузерах, но изменят его в других.

  Например, чтобы устранить увеличенные интервалы между пунктами перечня в Opera 6, можно записать в CSS:

  ```
  li {
    margin-top: 0;
    margin-bottom: 0;
  }
  ```

  Это не повлияет на отображение элементов в Internet Explorer и Mozilla (там эти свойства и так по умолчанию имеют значение 0), но в Opera 6 интервалы исчезнут;

Дополнительно

Internet Explorer, в отличие от Mozilla и Opera, плохо поддерживает многие специальные символы (например, математические знаки ∅ и ∉, значки ✔, ✘ и другие). Вместо таких символов отображаются пустые прямоугольники. Чтобы Internet Explorer тоже отображал эти символы, их надо заключить в какой-либо строчный элемент (например, `span` с определенным атрибутом `class`) и указать для этого элемента свойство `font-family`, в списке значений которого первым следует шрифт, содержащий нужные символы. Например, в Windows таким шрифтом является Arial Unicode MS. Конечно, чтобы это сработало, необходимо, чтобы на компьютере пользователя был нужный шрифт.

- разрабатывать разные версии страниц для разных браузеров. В первую очередь это касается таблиц стилей. Для одной и той же веб-страницы можно разработать несколько разных таблиц стилей, чтобы веб-сервер мог определять тип и версию браузера и передавать именно тот файл CSS, который предназначен для этого браузера.

Дополнительно

В редких случаях имеет смысл разрабатывать разные версии документов HTML. Например, если у вас в тексте используются мягкие переносы (``) — то браузерам, не поддерживающим этих переносов, придется передавать страницу, где они отсутствуют. Иначе в этих браузерах страница может отобразиться некорректно.

Примечание

На некоторых сайтах есть доступ к их текстовым версиям. Например, сайт Adobe предлагает альтернативную текстовую страницу, с помощью которой пользователь может загрузить Adobe Reader (рис. 7.21). Это позволяет получить доступ к файлам пользователям практически любого устройства и сколь угодно старых браузеров. А это, в свою очередь, способствует распространению программных продуктов Adobe.

Настройки пользователя. Во многих браузерах можно настроить режим просмотра страниц определенным образом. В большинстве современных браузеров пользователь имеет возможность:

- полностью или частично отключить CSS и другие средства оформления (рис. 7.22, *а* — для Internet Explorer); в этом режиме текст будет отображаться с оформлением по умолчанию;
- установить размер шрифта по умолчанию и минимальный размер шрифта (рис. 7.22, *б* — для Mozilla); если установлен минимальный размер

шрифта, то любой текст на веб-странице будет отображаться не мельче этого размера;

- задать определенный стиль отображения ссылок — например, заставить все ссылки отображаться синим цветом с подчеркиванием, независимо от воли разработчика (рис. 7.22, *в* — для Opera);

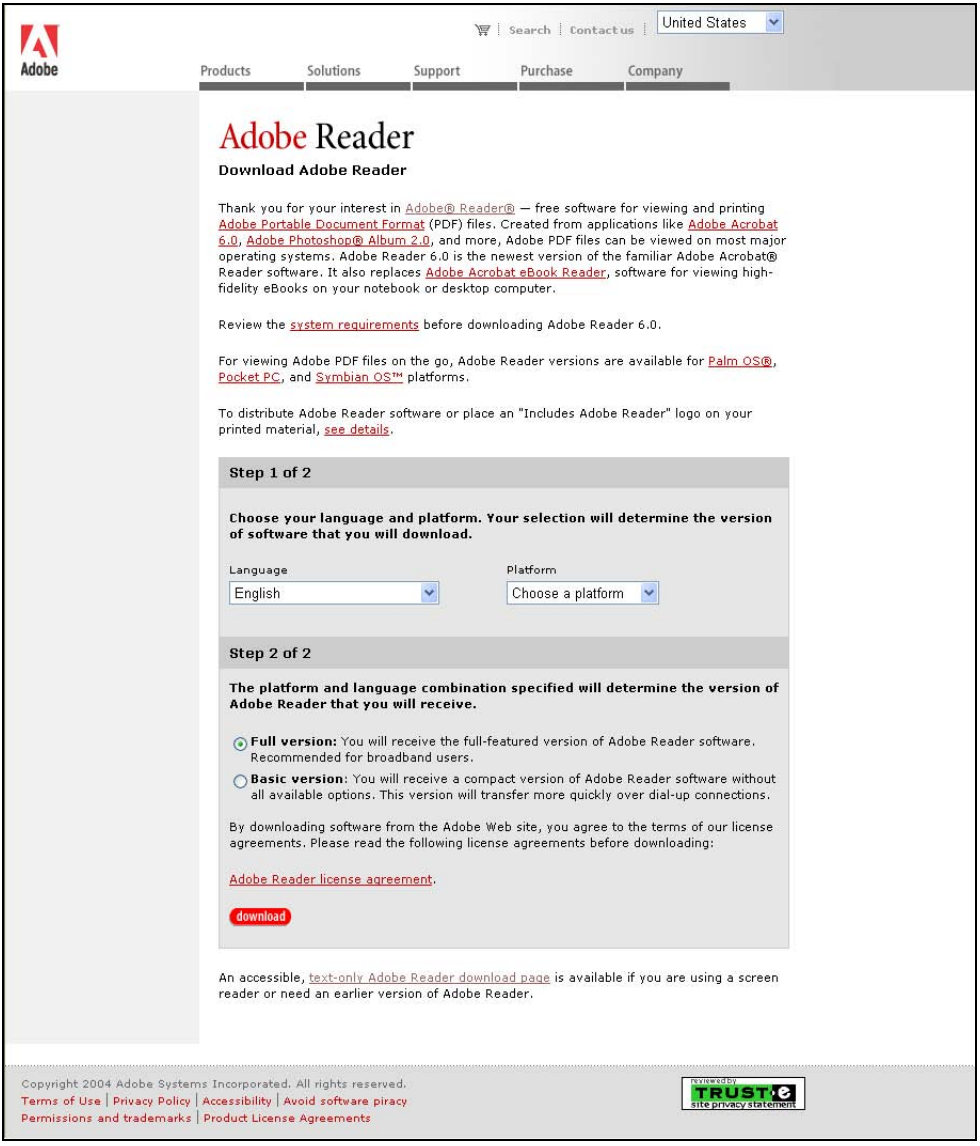


Рис. 7.21, *а*. www.adobe.com: кроме обычного интерфейса для загрузки файлов...

| Adobe home page | access.adobe.com |

Adobe Reader download page

Below are quick links to Adobe® Reader® software categorized by version. These links are followed by a more complete set of links to available versions of Adobe Reader in other languages. If you are distributing Adobe Reader 6.0.1, use the full version rather than the basic version. The full version includes accessibility, search, multimedia, and eBook support, as well as Adobe Photoshop® Album slide show and electronic card support. This will help ensure that recipients have access to all available functions for Adobe Reader. For more information on accessibility and Adobe Reader, including non-MSAA solutions, visit access.adobe.com.

All available versions of Adobe Reader, by version:

Adobe Reader 6.0.1, full version
Adobe Reader 6.0.1, basic version
Acrobat® Reader 5.1
Acrobat Reader 5.1 with Search and Accessibility
Acrobat Reader for Pocket PC 2002 (all models - ARM, X-Scale)
Acrobat Reader for Pocket PC 1.0
Adobe Reader for Palm OS® 3.0
Acrobat Reader for Palm OS 2.0
Acrobat Reader for Palm OS 1.1
Acrobat Reader for Symbian OS™ 1.0
Acrobat Reader 5.0.8
Acrobat Reader 5.0.5

Рис. 7.21, *б*. www.adobe.com: ...сайт предлагает текстовую страницу, доступную более широкому кругу пользователей

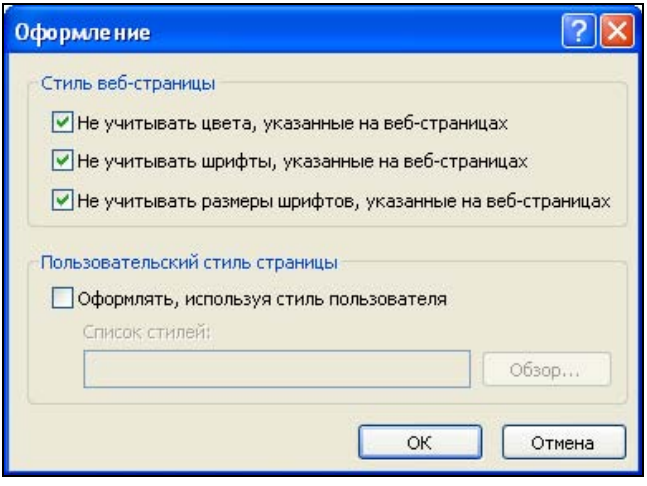
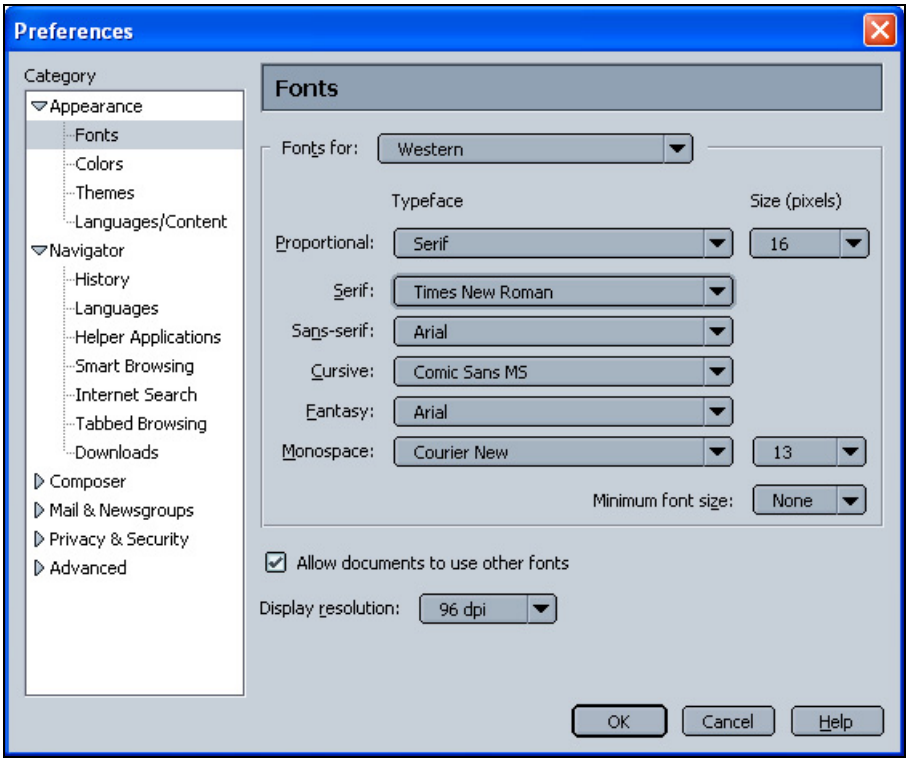
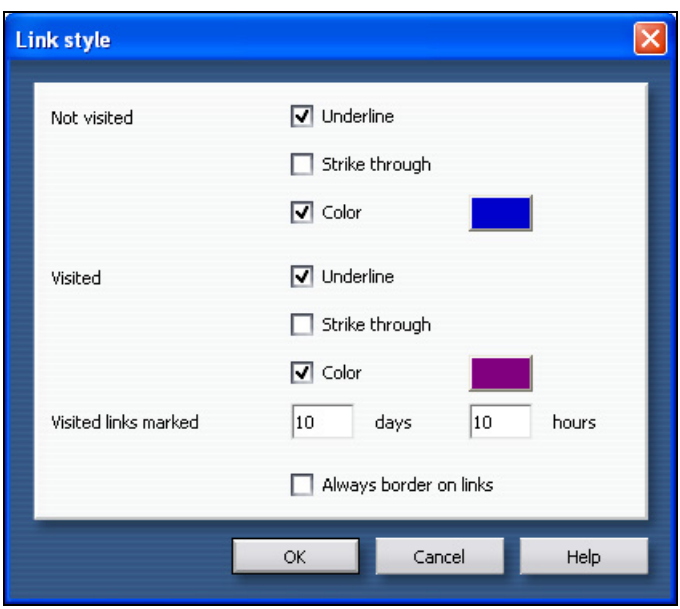


Рис. 7.22, *а*. Некоторые настройки пользователя: Internet Explorer



б



в

Рис. 7.22, *б* и *в*. Некоторые настройки пользователя: *б* — Mozilla, *в* — Opera

- отключить загрузку изображений — многие пользователи поступают именно так для экономии трафика. Учитывая это, не следует заключать текст в изображения (Напомним, что это далеко не единственный и не главный недостаток текста, заключенного в изображение.);
- установить кодировку по умолчанию — браузер, отправляя запрос HTTP, будет указывать эту кодировку в качестве предпочтительной. Если веб-сервер имеет такую возможность, он предоставит страницу именно в этой кодировке. Если не соблюдать рекомендаций, касающихся выбора кодировки (*см. главу 3*) и специальных символов, некоторые символы на странице могут отобразиться некорректно.

Дополнительно

Настройки пользователя не ограничиваются перечисленными. Например, большинство браузеров предоставляют пользователю возможность подключить к веб-странице собственную таблицу стилей; существуют и другие экзотические настройки. Правда, пользователи редко прибегают к особо тонкой настройке своего браузера; многие вообще не используют его специальных возможностей. Тем не менее, при разработке страниц необходимо учитывать по крайней мере пять перечисленных настроек.

Если отключить все оформление страницы, она будет отображаться в так называемом «академическом» режиме. Разрабатывая сайт, убедитесь, что в этом режиме пользователь может с ним взаимодействовать.

Сравните, например, сайт финансовой группы «Альфа-Страхование» в обычном виде и в «академическом» режиме (рис. 7.23). В академическом режиме он выглядит довольно невнятно: бóльшая часть страницы пустует, в то время как новости и часть ссылок из правого столбца ушли за нижнюю границу экрана. Может быть, такие неприятности возникают только в «академическом» режиме, а у обычных пользователей все просматривается нормально? Увы, взглянув на сайт в обычном режиме, мы все равно видим, что подавляющую часть страницы занимает изображение — в то время как пользователи приходят на сайт явно не за ним! Текст ссылок в правом столбце слишком мелкий, они не подчеркнуты; цветовая гамма неудачна: синие ссылки в верхней части страницы сливаются с синим фоном, серый текст снизу (**Новости** и дата) также плохо различим на сером фоне.

Это лишний раз доказывает, что если не заботиться о доступности сайта в «академическом» режиме, то он, скорее всего, будет плохо работать и в обычном.

Внимание!

Тестируйте сайт в «академическом» режиме. Это нужно не потому, что пользователи будут просматривать сайт в этом режиме (хотя такое тоже возможно),

а для того, чтобы предотвратить многие затруднения с доступностью, которые могут проявиться в обычном режиме просмотра.

В этой книге не описаны все возможные аспекты совместимости текстового содержания страниц с различными браузерами. Мало того, *все* случаи несовместимости описать просто невозможно, не говоря уже о том, чтобы указать решение проблемы в каждом случае.

Разработчику требуется хорошо знать стандарты XHTML и CSS, а также обладать опытом кодирования документов и таблиц стилей, знать особенности различных браузеров — тогда он сможет находить эффективные решения проблем совместимости по мере их возникновения.

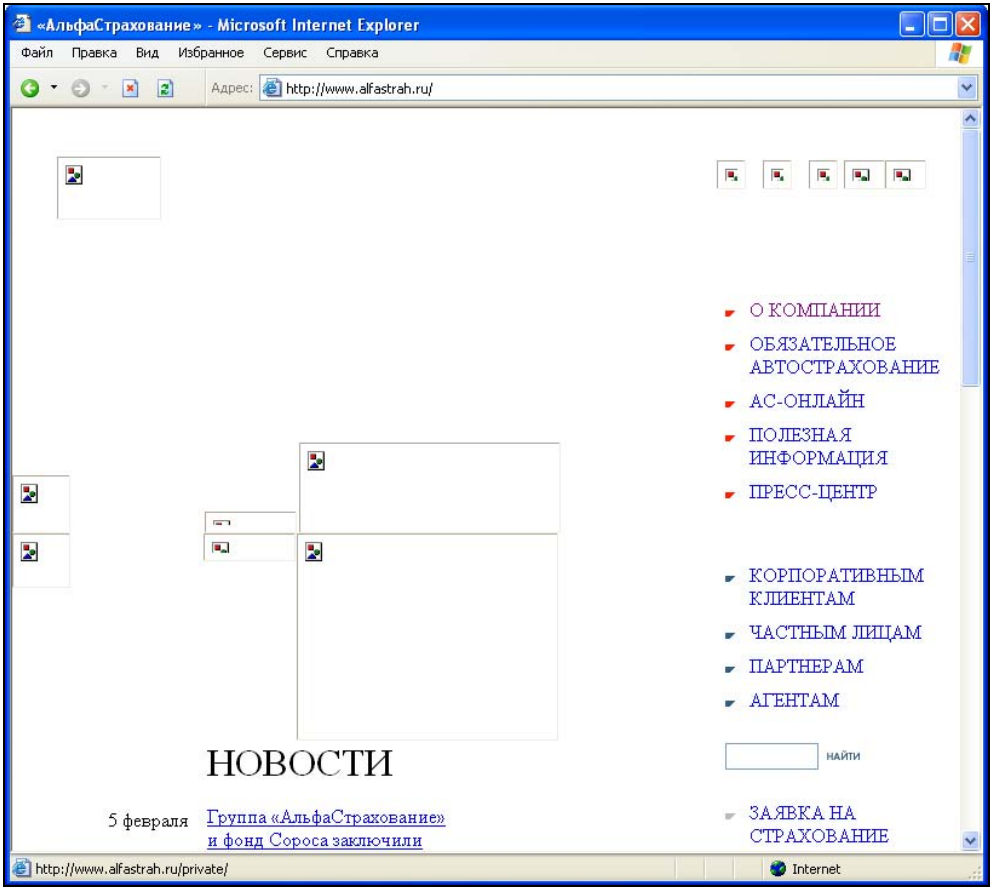


Рис. 7.23, *а*. www.alfastrah.ru: в «академическом» режиме сайт неудобен...

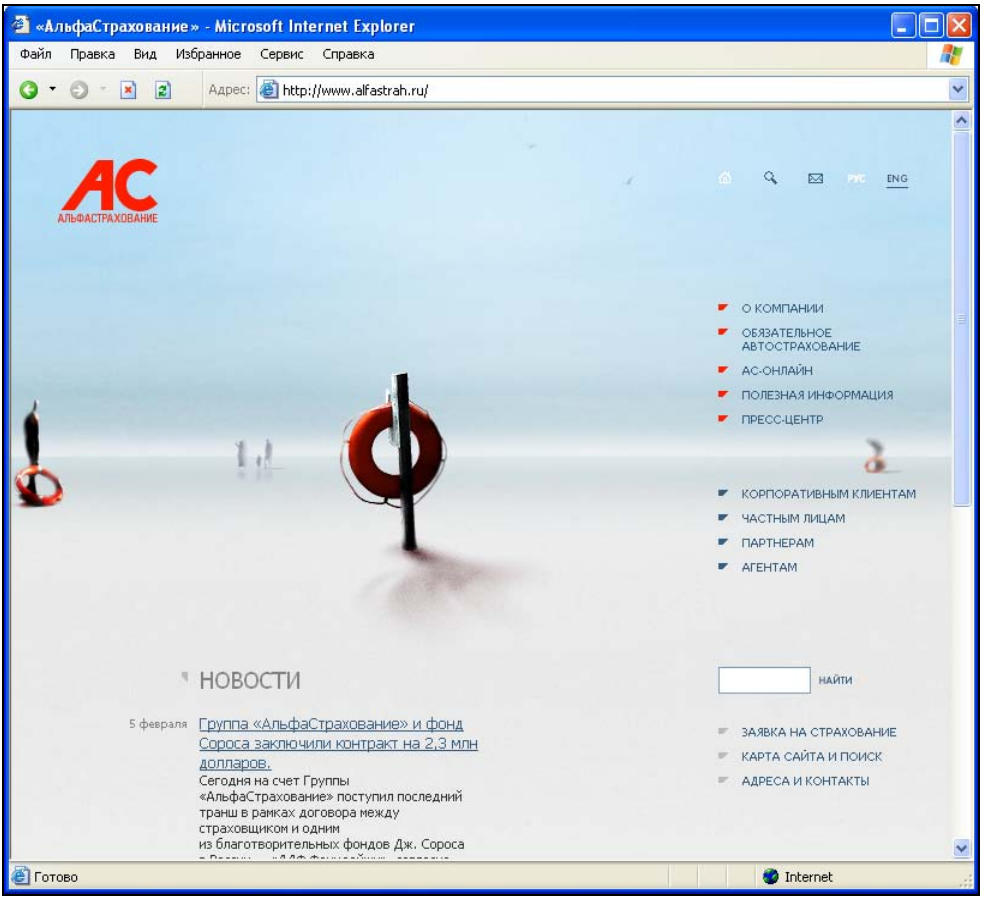


Рис. 7.23, *б*. www.alfastrah.ru: ...но и в обычном — не лучше

Примечание

Стоит ли поддерживать нестандартные и устаревшие браузеры, которыми пользуется всего несколько процентов посетителей? Все дело в том, что если разработчик заботится об удобстве и доступности сайта для основной массы посетителей — это автоматически приводит к тому, что и пользователи редких устройств и браузеров могут обращаться к сайту. Если же какие-то из трех основных условий доступности нарушаются в старых браузерах — значит, скорее всего, у сайта будут какие-то проблемы и в современных браузерах.

При грамотной разработке сайта разработчик не должен специально беспокоиться о его совместимости со старыми и экзотическими средствами просмотра. Если он обеспечивает настоящую доступность сайта для современных устройств, то совместимость с древними браузерами обеспечится сама собой.

7.2.2. Esse homo[2]
«Человеческий фактор» доступности

Какая кутерьма! О, мама мия!..
Убогие... безрукие... хромые...
На многих карнавалах я бывал,
Но это самый жуткий карнавал!

Л. Филатов. Любовь к трем апельсинам

Все люди — разные. Поэтому у одних пользователей могут возникать такие затруднения в работе с сайтом, до которых разработчики и другие пользователи могут вовсе не додуматься. Учитывая особенности различных пользователей, вы сделаете сайт более доступным — а следовательно, более востребованным.

Что же отличает пользователей друг от друга?

- **Зрение** — молодым разработчикам бывает трудно понять, что за проблемы со зрением возникают у пользователей. Тем не менее стоит отметить, что старшее поколение все активнее осваивает Всемирную Паутину. Именно к этому поколению относятся персоны, от которых многое зависит — директора крупных фирм и банков, наиболее состоятельные покупатели интернет-магазинов и т. д. Если вы хотите, чтобы представители этой категории оказались в числе частых посетителей вашего сайта, — не забывайте о том, что у многих из них трудности со зрением. Эти трудности разделяются на две группы:
  - **слабое зрение** — пользователи со слабым зрением предпочитают крупный четкий текст, они быстрее утомляются при чтении. Чтобы облегчить им взаимодействие с сайтом, нужно соблюдать те же рекомендации, о которых уже многократно говорилось: указывать размер шрифта в em, не заключать текст в изображение, не использовать нестандартных шрифтов, позволить браузеру оформлять основной текст по умолчанию;
  - **нарушения цветового восприятия** — в эту группу входят не только дальтоники, но и многие люди с небольшими отклонениями в восприятии цветов. Такие пользователи могут испытать затруднения, если, например, для выбора одного из двух вариантов им предложат нажать либо желтую, либо оранжевую кнопку. Они не всегда смогут угадать, какая из кнопок желтая, а какая — оранжевая.

  Чтобы не дезориентировать таких пользователей, нужно применять контрастные цвета. Кроме того, следует избегать ситуаций, когда какая-либо информация доступна только благодаря цвету. Например, не надо делать двух кнопок, которые отличаются только тем, что одна из них желтая,

[2] Се человек (*лат.*).

а другая — оранжевая. Если эти кнопки действительно обеспечивают разные функции, укажите это с помощью надписей на них и, возможно, вспомогательного текста.

Примечание

Не забывайте, что бывают и невидящие пользователи. Они взаимодействуют с сайтами с помощью звуковых или брайлевских устройств. Пока такие устройства весьма редки и несовершенны, и их применение носит скорее экспериментальный характер. Однако в недалеком будущем они станут естественными предметами для невидящих людей — так же, как сейчас для них естественны книги, напечатанные по системе Брайля.

- **Опыт** — среди пользователей есть и квалифицированные специалисты в области информационных технологий, которые подключены к Интернету постоянно, и случайные или начинающие пользователи, включая домохозяек и директоров предприятий.

Дополнительно

Пользователи склонны действовать интуитивно. Попав на незнакомый сайт, они взаимодействуют с ним таким образом, который, *как им кажется*, скорее приведет к успеху. Только испытав непреодолимые затруднения, они обращаются к документации, читают справочную информацию или повышают собственную квалификацию еще как-нибудь.

Пользователю наиболее легко и удобно работать с сайтом, когда интерфейс этого сайта отвечает его интуитивным представлениям, а не когда ему приходится специально обучаться. Поэтому нужно исследовать аудиторию пользователей, их поведение и предпочтения, и разрабатывать сайты, которые соответствуют этим предпочтениям в наибольшей степени.

Чтобы учесть многообразие опыта, интеллектуального и культурного уровня пользователей, следует придерживаться рекомендаций, многие из которых также высказывались ранее:

- пишите простым, ясным литературным языком. Не конструируйте слишком сложных предложений. Пишите с нейтральной, доброжелательной интонацией: не надо ни чрезмерного панибратства, ни официально-канцелярского стиля;
- ограничивайте метафоры и юмор. Чувство юмора у разных пользователей неодинаково; многие поймут вас неправильно. Не употребляйте слишком много цитат из известных книг, фильмов и других произведений: пользователи могут не догадаться, что это цитаты;
- предоставляйте пользователям помощь и справочную систему: поиск по сайту, карту сайта, ответы на часто задаваемые вопросы, указатели и другие элементы аппарата. Сделайте так, чтобы пользователь с лю-

бой страницы мог перейти к справочным разделам;

- подчеркивайте ссылки, выделяйте посещенные и непосещенные ссылки разными цветами;
- убедитесь, что по каждой отдельной странице сайта пользователь может легко понять, где он находится, что он может здесь сделать и куда можно отсюда перейти.

纸 Дополнительно

Если вы применяете формы, постарайтесь обойтись без выпадающих списков. В некоторых случаях эти списки незаменимы (например, если пользователю требуется выбрать страну своего проживания), но обычно их можно заменить другими элементами управления.

Выпадающий список плох тем, что, когда он закрыт, пользователь не видит всех его пунктов. Если на странице два и больше таких списка, пользователь может одновременно обозревать только один из них — при этом он с большой вероятностью забудет, что было в других. Кроме того, работа с выпадающими списками требует специальных навыков, особой выдержки и реакции: даже у опытных пользователей, стоит им случайно щелкнуть мышью или сдвинуть ее, список в самый неподходящий момент захлопывается, и страница в окне браузера начинает внезапно меняться.

Постарайтесь, чтобы пользователи ваших сайтов сталкивались с такими неожиданностями как можно меньше.

Примечание

Не очень удачный элемент интерфейса поисковой системы Яндекс — знак «+» в правом нижнем углу области поиска (рис. 7.24). Даже опытному пользователю сложно догадаться, что это ссылка на страницу расширенного поиска. Например, автор узнал об этом только после того, как случайно провел указателем мыши над этим значком и увидел всплывающую подсказку **Расширенный поиск**.

Не нужно предъявлять чрезмерных требований к сообразительности пользователей. Поисковая система — не школьная олимпиада знаний. Лучше вообще ограничить использование пиктограмм или по крайней мере сопровождать их очевидными надписями.

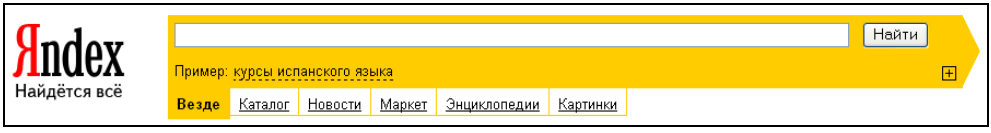


Рис. 7.24. www.yandex.ru: угадайте с трех раз, где тут расширенный поиск

- **Ограниченные возможности** — существуют пользователи, которым трудно пользоваться клавиатурой или мышью вследствие заболеваний или травм кистей рук. Таким пользователем, к сожалению, может стать каждый, поэтому особенности этой категории следует учитывать.

  Кроме того, бывают пользователи, которым удобнее пользоваться клавиатурой или, наоборот, мышью.

  Современные браузеры позволяют работать с сайтами с помощью одной только клавиатуры или одной только мыши. Например, перемещаться между ссылками можно нажатием клавиши <Tab>, а перейти по выбранной ссылке — нажатием клавиши <Enter>. Многие браузеры (например, Mozilla) предоставляют возможность перейти к определенному фрагменту текста на странице, просто набрав несколько первых символов этого фрагмента на клавиатуре. Не затрудняйте пользователям применение подобных средств. (Прежде всего, опять же, не заключайте текст в изображения.)

  Учтите, что пользователь с ограниченными возможностями — это не обязательно инвалид. Человек может стать таким пользователем временно — например, если просматривает сайт не на своем привычном компьютере, а в других условиях — скажем, на ноутбуке в дороге (где вместо мыши приходится пользоваться гораздо более неудобным тачпадом) или при плохом внешнем освещении.

Примечание

Мышь, клавиатура или видеокарта пользователя могут выйти из строя. Возможно, ему потребуется обратиться к Всемирной Паутине именно затем, чтобы решить эту проблему. Например, автор этой книги дважды попадал в такую ситуацию. Один раз потребовался драйвер видеокарты, без которого в системе класса UNIX не удавалось настроить графический режим. Пришлось воспользоваться текстовым браузером, чтобы загрузить драйвер с сайта производителя видеокарты. В другой раз внезапно отказала мышь, и потребовалось срочно узнать адреса магазинов, в которых продается нужная модель.

Помните: заботясь о пользователях с ограниченными возможностями, вы на самом деле заботитесь обо всех пользователях. Если с сайтом может эффективно работать инвалид — значит, здоровому человеку это удастся тем более. И наоборот: обычно, если сайт не годится для пользователей с ограниченными возможностями, он обязательно оказывается недостаточно доступен и для обычных посетителей.

7.2.3. Скрижали Закона
Принципы доступности сайтов от консорциума W3C

> Не дал ли вам Моисей закона? и никто из вас не поступает по закону. За что ищете убить Меня?
>
> *Евангелие от Иоанна, гл. 7, ст. 19*

В 1999 Консорциум W3C выпустил стандарт WCAG 1.0. (Аббревиатура «WCAG» расшифровывается «Web Content Accessibility Guidelines» и примерно переводится как «Принципы доступности содержания сайтов».) Этот стандарт содержит 65 рекомендаций, соблюдение которых улучшит доступность сайта для различных категорий пользователей и разных средств просмотра страниц.

Примечание

Стандарт WCAG 1.0 находится по адресу **www.w3.org/TR/WCAG10**. Кроме того, на странице **www.w3.org/WAI** можно найти информацию о разработках Консорциума W3C в области доступности сайтов и ссылки на другие нормативные и вспомогательные документы по этой теме.

Рассмотрим некоторые из наиболее важных требований WCAG, относящихся к текстовому содержанию сайтов.

Стандарты

- Соблюдайте стандарты Консорциума W3C. Проверяйте документы HTML и таблицы стилей с помощью валидатора.
- Применяйте наиболее свежие, рекомендованные W3C версии стандартов. Разрабатывайте документы на XHTML, а не HTML. Пользуйтесь CSS 2, а не CSS 1.

Дополнительно

Вместе с тем, нужно не «переборщить» с применением самых современных стандартов. Например, XHTML 1.1 — это формально более свежий стандарт, чем XHTML 1.0. Однако его несовместимость с более ранними версиями HTML приводит к тому, что документы становятся менее доступными в старых браузерах. Например, XHTML 1.1 запрещает пользоваться конструкцией `<a name="...">` в качестве якоря, предлагая вместо нее использовать атрибут `id` для соответствующего элемента. Однако не все старые браузеры поддерживают переход по `id`, поэтому в них окажутся недоступными ссылки, ведущие на какой-то конкретный участок веб-страницы.

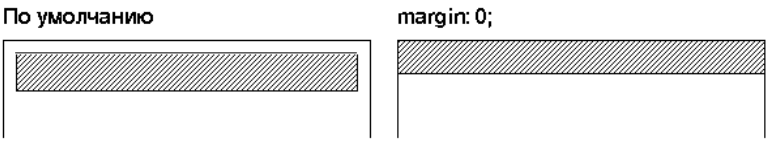
Чтобы избежать проблем со старыми системами и браузерами, следует применять только ту часть нового стандарта, которая хорошо поддерживается в устаревших, но еще использующихся устройствах. Также можно применять решения, которые просто не обрабатываются старыми браузерами — например, свойства CSS `letter-spacing` и `font-size-adjust`. В современных браузерах эти свойства приведут к нужному эффекту, а в устаревших не сработают, но и не вызовут каких-либо затруднений для пользователя.

Примечание

Соблюдение стандартов и совместимость с браузерами — это не противоречащие, а дополняющие друг друга требования. Пишите код, соответствующий стандартам, но одновременно с этим учитывайте особенности конкретных браузеров.

Например, чтобы шапка страницы оказалась плотно прижатой к границам окна, можно записать в CSS:

```
body {
  margin: 0;
}
```

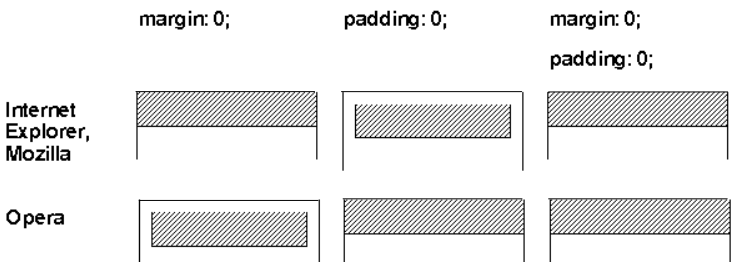


Это работает в браузерах Internet Explorer и Mozilla, однако в Opera отступ вокруг шапки остается. Для этого браузера требуется обнулить не `margin`, а `padding`. Но если вместо `margin: 0;` записать `padding: 0;` — шапка перестанет прижиматься к границам в двух других браузерах.

Значит, надо использовать и ту, и другую конструкцию:

```
body {
  margin: 0;
  padding: 0;
}
```

Эта запись соответствует стандарту CSS и при этом обеспечивает совместимость с основными браузерами:



- Не пользуйтесь конструкциями, отсутствующими в стандарте, — например, элементами `nobr` и `marquee`, атрибутами `leftmargin` и `topmargin` и т. д. Всегда можно написать код, обеспечивающий те же функции, но не противоречащий стандартам.
- Не пользуйтесь элементами и атрибутами, которые объявлены в стандарте *нежелательными* (*deprecated*). Например, элементы `font` и `center`, а также атрибуты `align` и `bgcolor`, хотя и присутствуют в стандарте HTML и XHTML, но относятся к нежелательным. Вместо подобных конструкций HTML всегда лучше пользоваться таблицами стилей.

Примечание

XHTML 1.0 Strict — это разновидность XHTML, в которой нежелательные элементы и атрибуты запрещены. Рекомендуется разрабатывать документы в соответствии с ней, а не с другими разновидностями XHTML.

- Используйте элементы HTML только в соответствии с их структурным назначением. Например, в большинстве браузеров элемент `blockquote` (блочная цитата) отображается с отступом от левого края. Поэтому некоторые разработчики употребляют этот элемент, чтобы создать отступ. Валидатор не сочтет это ошибкой: в самом деле, он же не знает, цитата или не цитата находится внутри `blockquote`! Однако такое употребление элемента неграмотно; `blockquote` следует применять только для настоящих цитат и других блоков, имеющих схожую структурную роль.

Структура информации

Все, что рекомендует Консорциум W3C по части организации информации, уже обсуждалось в этой книге. Перечислим такие рекомендации вкратце:

- разделяйте содержание сайта на небольшие блоки. Группируйте элементы навигации в специальные панели;
- составляйте информативные заголовки, дающие представление о содержании озаглавленного ими текста. Добивайтесь того, чтобы заголовки были понятны без контекста (например, когда они встречаются в списке результатов поиска). То же самое касается заглавий страниц и текстов гиперссылок. Не надо ссылок типа «Щелкните здесь»;

Примечание

Если у вас есть несколько похожих заголовков, то начинайте их с разных слов. Общую часть лучше поместить в конец заголовка.

Рекомендуется	Не рекомендуется
• Слои в Adobe Photoshop CS • Сохранение для Сети в Adobe Photoshop CS • Adobe Photoshop 7.0 и CS	• Adobe Photoshop CS: слои • Adobe Photoshop CS. Сохранение для Сети • Adobe Photoshop CS и 7.0

- пишите короткие предложения в простом стиле. Не стройте сложных грамматических конструкций, избегайте страдательного залога. Старайтесь употреблять синонимы; если все-таки нужно выбрать один синоним из нескольких, то выбирайте наиболее употребительный, например:

Лучше	Хуже
компьютер	ЭВМ
микроволновая печь	СВЧ-печь
елочные игрушки	елочные украшения
фильм	кинокартина

Дополнительно

Для многих зарубежных языков (например, для английского) существуют методики, позволяющие рассчитать численный показатель удобочитаемости. Этот показатель основан на среднем, минимальном и максимальном количестве слов в предложениях, на длине отдельных слов, на сочетаниях букв и других параметрах текста. Для русского языка пока не существует удачных методик такого расчета.

Может показаться, что стремление удовлетворять формальным критериям удобочитаемости сделает текст «мертвым», исключит из него всякую эмоциональную и художественную составляющую. Между тем, произведения Шекспира и Байрона при формальных измерениях обычно получают достаточно высокие показатели удобочитаемости; им уступают многие официальные документы и газетные статьи.

- создавайте эффективный аппарат сайта: карту сайта, средства поиска по сайту, содержания, указатели и т. д.;

Примечание

Структуру сайта можно дополнительно подчеркнуть с помощью средств браузерной навигации. (Эти средства доступны во многих современных браузерах — например, в Mozilla и Opera 7.) Используйте в шапке документа элемент `link` с атрибутами `rel`, `href` и `title`, например:

```
<link rel="top" href="http://www.abc.ru/" title="Главная страница" />
```

```
    <!-- переход на главную страницу -->
<link rel="next" href="chapter31" title="Трезвучие с повышенной терцией" />
    <!-- переход к следующей странице -->
<link rel="contents" href="../../toc.html" title="Содержание" />
    <!-- переход к содержанию -->
```

Некоторые значения атрибута `rel` (и их смысл) рекомендует Консорциум W3C в спецификации HTML 4 (**www.w3.org/TR/html4/types.html#type-links**). Многие браузеры поддерживают и другие значения — например, Mozilla поддерживает значения `up` (переход на один уровень выше) и `toc` (то же, что `contents`), а Opera — значения `first` и `last` (переход соответственно к первой и последней странице в последовательности из нескольких страниц). Более подробно узнать о возможностях браузерной навигации можно в документации к этим браузерам.

- обозначайте язык страницы с помощью атрибутов `lang` и `xml:lang` элемента `html`. Если на странице есть фрагменты на других языках, указывайте для них эти же атрибуты;
- применяйте атрибут `title` для аббревиатур и сокращений (элементы `abbr` и `acronym`), а также для гиперссылок, если это поможет пояснить их назначение.

Примечание

Не забывайте, что `title` — это *вспомогательный* текст гиперссылки. Основную информацию о назначении ссылки должен нести текст самой ссылки. Атрибут `title` должен всего лишь давать дополнительные пояснения пользователю, который уже понял, куда ведет ссылка.

Разделение структуры и представления

Разделение структуры и представления улучшает доступность сайта, особенно если условия просмотра сайта отличаются от тех, к которым привык разработчик. Разделяя структуру и представление, можно обеспечить выполнение трех основных условий доступности практически во всех возможных устройствах просмотра, включая наладонные компьютеры, а также такие редкие устройства, как голосовые или брайлевские браузеры. Итак:

- грамотно используйте структурную разметку: заголовки, гиперссылки, списки, внутритекстовые выделения и т. д. В этом случае, даже если оформление сайта окажется недоступно, пользователь все равно сможет эффективно им воспользоваться (см. рис. 7.25);

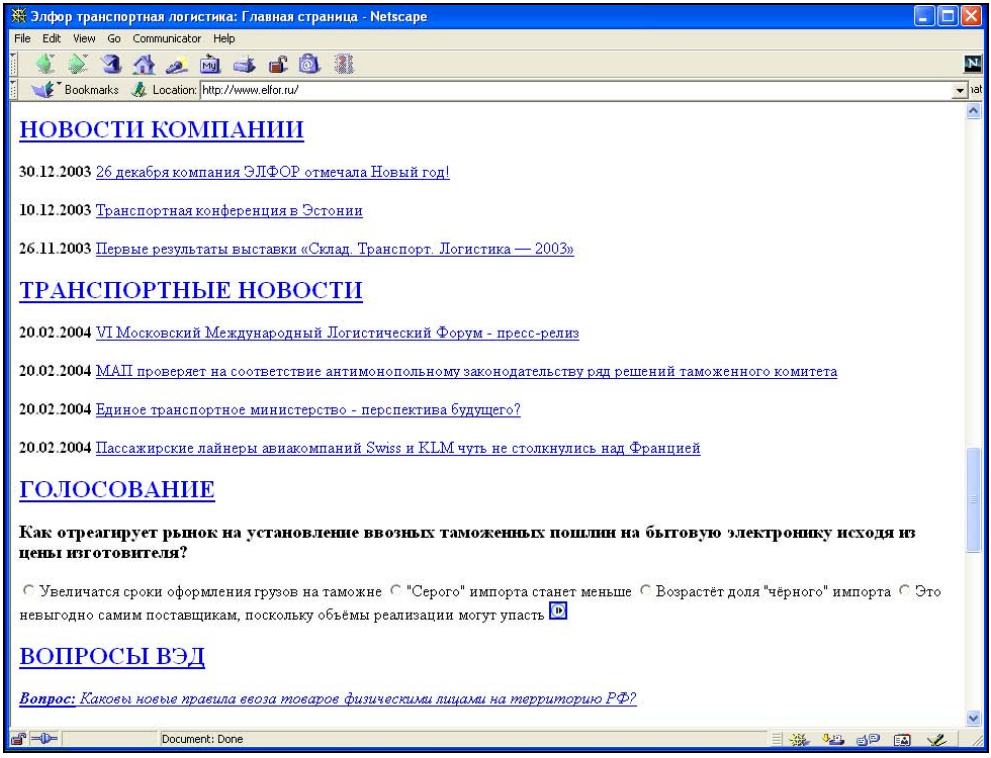


Рис. 7.25. www.elfor.ru: в устаревшем браузере оформление сайта недоступно (сравните с рис. 5.5), но грамотная структурная разметка позволяет посетителю получить всю необходимую информацию и воспользоваться всеми службами

纸 *Дополнительно*

Если вы посмотрите на сайт **www.elfor.ru** в Netscape 4 или еще каком-нибудь древнем браузере, вы увидите наверху ссылку: **Пропустить панель навигации**. Эта ссылка не отображается в обычном режиме, так как в таблице стилей для нее указано свойство `display: none;`. Однако в браузере, не поддерживающем CSS, панель навигации занимает слишком много места, поэтому разработчики предоставили пользователям возможность пропустить ее и сразу перейти к содержательной части страницы.

✓ *Примечание*

Консорциум W3C советует вам прочесть свои веб-страницы кому-нибудь по телефону. Скорее всего, ваш слушатель не получит полного представления о со-

держании и службах страницы, если вы просто прочтете ему подряд весь текст. Значит, вам потребуется сопровождать чтение специальными комментариями. А теперь соберите все эти комментарии и включите их в свою страницу (при этом их можно сделать недоступными для пользователей обычных, современных браузеров). Действительно, представьте, что кто-нибудь и в самом деле будет прослушивать вашу страницу с помощью голосового браузера. Вы же не будете стоять рядом и комментировать все, что необходимо!

- тестируйте сайт в текстовых браузерах и в обычном браузере в академическом режиме. *Академический режим* — это следующие настройки браузера:
  - отключена загрузка изображений;
  - отключены таблицы стилей;
  - отключен JavaScript, а также все остальные расширения браузера: Java, Flash и т. д.;
  - установлены настройки:
    1. Не учитывать шрифты, указанные на веб-страницах;
    2. Не учитывать цвета, указанные на веб-страницах.

  Убедитесь, что в академическом режиме выполняются три основных условия доступности;

Примечание

Напомним, что тестирование в «неполноценных» режимах нужно в первую очередь не для того, чтобы учесть интересы пользователей с этими режимами (таких пользователей может вовсе не быть), а для того, чтобы гарантировать доступность сайта для широкого круга реальных пользователей.

Действительно, вряд ли на ваш сайт придет хотя бы один слепоглухонемой безрукий посетитель. Но если вы сделаете сайт доступным для такого вымышленного посетителя — вы автоматически обеспечите его доступность для *всех* реальных посетителей. Если же вымышленный инвалид испытает затруднения в работе с сайтом — значит, кто-нибудь из реальных посетителей также будет хотя бы иногда сталкиваться с этими затруднениями.

- всегда указывайте альтернативное текстовое описание (атрибут `alt`) для изображений, а также описания для внедренных объектов (элемент `object` и другие). Не используйте нетекстовых объектов для представления текстовой информации. Не заключайте текст в изображения, ролики и т. д. Убедитесь, что любой текст на вашем сайте можно:
  - выделить мышью и скопировать в буфер обмена;
  - увеличить или уменьшить стандартными средствами браузера.

Оформление

- Убедитесь, что информация на сайте доступна при отсутствии цветового оформления.

  Например, если при заполнении формы названия обязательных полей выделены цветом — некоторые пользователи не смогут отличить этих полей от необязательных. Это не значит, что от выделения цветом нужно отказаться; можно просто дополнить его другими знаками различия — например, пометить обязательные поля звездочками (*).

- Используйте контрастную цветовую гамму. Цвет текста и цвет фона должны как можно сильнее отличаться друг от друга.

Примечание

Консорциум W3C рекомендует прочитать на эту тему статью А. Ардити, размещенную по адресу **www.lighthouse.org/color_contrast.htm**.

- Указывайте размеры шрифтов в относительных единицах (предпочтительнее всего в em), чтобы пользователь мог масштабировать текст средствами браузера. В списке шрифтов для свойства `font-family` всегда указывайте последним одно из стандартных семейств шрифтов (`serif`, `sans-serif`, `monospace`).
- Разрабатывайте «резиновые» страницы, ширина которых «подстраивается» под ширину окна браузера.
- Тестируйте сайт в разных браузерах, на разных платформах, в разных видеорежимах. Учтите, что пользователи приходят к вам не только с самыми свежими версиями браузеров, но и с браузерами на 1—2 версии ниже. Например, на момент написания этой книги наиболее популярным является браузер Internet Explorer 6-й версии, но немалая доля пользователей еще применяет версии 5.0 и 5.5.

Примечание

Не помещайте гиперссылки рядом друг с другом, иначе пользователь не всегда сможет отделить одну ссылку от другой.

Рекомендуется	Не рекомендуется
01.04.2004. В разделе «Классическая музыка» обновилась страница Людвига ван Бетховена: размещена Пятая симфония (1808).	**01.04.2004.** В разделе «Классическая музыка» размещена Пятая симфония Людвига ван Бетховена.

- Оформляйте страницы таким образом, чтобы максимально облегчить пользователям их быстрый просмотр. Разбивайте информацию на блоки, продумывайте их оптимальное расположение на странице, разумно пользуйтесь средствами выделения, рамками, изображениями и другими возможностями оформления. (Но не забывайте при этом, что страница должна быть доступна и без оформления!)

Персонализация

- Если вы опасаетесь, что страница может быть недоступна определенным категориям пользователей, предоставьте ссылку на альтернативную (например, текстовую) версию.

Дополнительно

Некоторые браузеры поддерживают ссылки на альтернативные страницы с помощью элемента `link`, например:

```
<link title="Текстовая версия" rel="alternate"
href="/txt/index.html" media="aural, braille, tty" />
```

Обеспечьте также ссылку с альтернативной страницы на основную и убедитесь, что пользователи ясно различают, какая из страниц — основная, а какая — альтернативная.

- Не делайте страниц, которые сами обновляются через какой-то промежуток времени. Пользователь вряд ли будет доволен, если страница, которую он читает в данный момент, вдруг начинает обновляться.

Примечание

Не делайте также перенаправляющих страниц, которые сами не несут содержания, а перенаправляют пользователя к другому ресурсу — например, с помощью JavaScript. Если требуется перенаправление, осуществляйте его на стороне сервера.

Кроме того, не допускайте всплывающих окон. Они раздражают и дезориентируют пользователей.

Дополнительно

Отправляя запрос серверу, браузер может включить в него сведения о предпочтениях пользователя, в частности:

- о языке, на котором пользователь хотел бы получить страницу;
- о желательной кодировке;

- о желательном типе содержания (Content-type — например, text/html, text/plain или text/xml).

Рекомендуется, чтобы ваш веб-сервер обрабатывал эти сведения и по возможности предоставлял страницу, наиболее удовлетворяющую требованиям пользователя. Если сервер готов отдавать страницу в разных кодировках, предусмотрите решение проблем, связанных с перекодировкой (*см. главу 3*), чтобы пользователь не получил страницу, полную посторонних символов.

Степени соответствия WCAG 1.0

Стандарт WCAG 1.0 определяет для каждого требования уровень приоритета. К первому уровню (Priority 1) относятся наиболее важные требования, несоблюдение которых приводит к серьезному ухудшению доступности сайта. Среди этих требований, например, — наличие атрибута `alt` у всех изображений и возможность пользоваться сайтом без подключенных таблиц стилей.

Ко второму уровню (Priority 2) относятся менее важные требования — например, запрет на использование нежелательных (deprecated) элементов и недопустимость автоматического обновления страниц.

Наконец, к третьему уровню (Priority 3) относятся второстепенные требования — например, чтобы все аббревиатуры и сокращения расшифровывались, а также чтобы схожие по содержанию заголовки начинались с разных слов.

Примечание

Если сайт не удовлетворяет требованиям уровня 1 — некоторые категории пользователей вообще не смогут с ним работать.

Если сайт не удовлетворяет требованиям уровня 2 — некоторые категории пользователей будут испытывать заметные трудности в работе с ним.

Если сайт не удовлетворяет требованиям уровня 3 — у пользователей не возникнет существенных затруднений, но выполнение этих требований позволит сделать сайт еще доступнее и легче.

Если сайт удовлетворяет всем требованиям первого уровня, ему присваивается степень «A» соответствия стандарту WCAG 1.0. Профессионально сделанный сайт обязан соответствовать этой степени.

Если, кроме требований первого уровня, сайт удовлетворяет требованиям второго уровня — ему присваивается степень соответствия «Double-A» («AA»). Наконец, если соблюдены также требования третьего уровня — сайт получает степень «Triple-A» («AAA»). Стремитесь к тому, чтобы ваши сайты соответствовали именно этой, самой строгой степени. Достичь этого совсем

не трудно, а в результате вы получите сайт, который заслужит одобрение пользователей и будет служить образцом доступности для ваших коллег.

Дополнительно

На момент написания этой книги в разработке находится следующая версия стандарта — WCAG 2.0. Черновой вариант этого документа находится по адресу **www.w3.org/TR/WCAG20**.

7.3. ...И люди к вам потянутся

Замечания о привлекательности сайта

— Может, круиз окажется не таким мерзопакостным, как я предполагал.
— Многие джентльмены находят его весьма привлекательным, сэр.

П. Г. Вудхаус. Кодекс чести Вустеров

Сайты разрабатываются для того, чтобы реальные пользователи посещали их и выполняли с их помощью свои задачи. Пользователи будут посещать ваш сайт только в том случае, если он будет им зачем-то нужен. Не забывайте об этом на всех этапах разработки сайта.

Чтобы сайт имел успех, он должен стать привлекательным для пользователей. Пользователи должны *хотеть* его посещать. В самом деле, если сайт никто не хочет посещать — зачем же он вообще нужен?

Чем же определяется привлекательность сайта?

7.3.1. Содержание

Итак, пусть или все мое содержание — безобразно, или каждый поступок, вытекающий из основного, красив.

А. Цветаева. Дым, дым и дым

Прежде всего, сайт должен предоставлять такое содержание, которое действительно нужно пользователям. Содержание — это, во-первых, информация, которую можно получить на сайте, а во-вторых, действующие службы (например, поисковая система или система заказа товаров). Содержание — основа сайта, самая главная его составляющая.

Внимание!

Пользователи будут посещать сайт только в том случае, если на этом сайте действительно есть то, что им нужно. Не забывайте об этом никогда.

Итак:

- убедитесь, что содержание вашего сайта действительно нужно пользователям, что к нему действительно будут обращаться. К сожалению, в Интернете немало сайтов, которые никто не посещает. Вряд ли вам стоит разрабатывать еще один такой сайт;

Примечание

Одним из лучших доказательств того, что содержание — это главное, служит электронная библиотека **www.lib.ru**. Оформление этого сайта оставляет желать лучшего, и тем не менее это один из самых посещаемых русскоязычных сайтов. Он уступает многим другим электронным библиотекам не только по внешнему виду, но и по организации информации — и, тем не менее, пользователи обращаются к нему намного чаще, чем к его конкурентам.

Автор не призывает вас делать такие же страховидные сайты, как **www.lib.ru**. Но пример этой электронной библиотеки показывает, что хорошее содержание уже само по себе способно обеспечить востребованность и привлекательность сайта, даже если оформление хромает.

- убедитесь, что информация, которую вы предоставляете, достоверна, полна, своевременна и актуальна. Если вы хотя бы раз обманете пользователей — больше вам не поверят;
- убедитесь, что службы, которые есть на вашем сайте, действительно нужны пользователям.

Примечание

На многих сайтах предлагается форма поиска, в которой пользователь может выбрать, желает ли он искать только в пределах этого сайта или по всему Интернету (рис. 7.26). Возможность искать по всему Интернету — совершенно бессмысленная. Посудите сами: пользователь не будет заходить на сайт **www.aquarium.ru**, чтобы с помощью этого сайта искать какую-то информацию по всему Интернету. Если он действительно захочет найти что-то в Сети — он обратится к специальной поисковой системе (например, Яндекс или Google).

Размещайте на сайте только такие службы, за которыми пользователи действительно будут ходить именно на этот сайт.

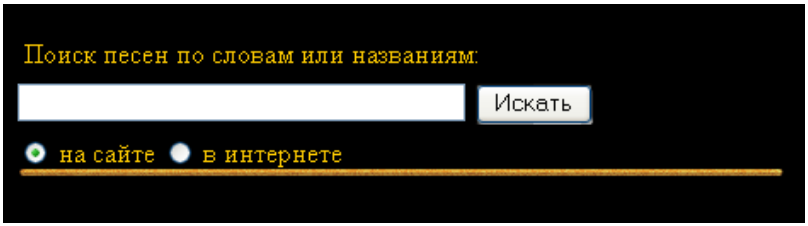


Рис. 7.26. Форма поиска на **www.aquarium.ru**

Текст — главнейшая часть содержания любого сайта. Даже если сайт представляет собой виртуальную картинную галерею, пользователь все равно будет ориентироваться на этом сайте именно по тексту. Если ему нужна какая-то определенная картина, он не будет просматривать все картины подряд, а обратится к текстовому перечню картин, чтобы найти в нем нужное название.

7.3.2. Структура и навигация

> А в чистом поле стоит столб, а на столбу написаны эти слова: «Кто поедет от столба сего прямо, тот будет голоден и холоден; поедет в правую сторону, тот будет здрав и жив, а конь его будет мертв; а кто поедет в левую сторону, тот будет убит, а конь его жив и здрав останется».
>
> *Сказка об Иване-царевиче, Жар-птице и о сером волке*

Содержание сайта должно быть организовано в ясную структуру. Пользователю будет легко ориентироваться на сайте, если он понимает, по какому принципу содержание сайта разбито на разделы и подразделы.

Примечание

Предположим, что пользователь ищет на сайте запись «Yesterday». Какая структура поможет ему легче найти ее?

Рекомендуется	Не рекомендуется
Классическая музыка	Легенды рока
Джаз	Зарубежная рок-музыка
Рок-музыка	Музыка 1960-х
Диско	Мировые хиты
...	...

В первом случае пользователь, не сомневаясь, выберет «Рок-музыку» и отправится дальше, в подразделы. (Если они организованы так же ясно и однозначно — он легко доберется до нужной записи.) Во втором случае пользователь затруднится: то, что он ищет, так или иначе подходит к любому разделу. Так куда же ему идти?

Дополнительно

Иногда бывает полезна параллельная структура. Например, на сайте кулинарных рецептов вполне уместен примерно такой состав разделов:

…
- Мясные блюда
  - ◊ Горячие
  - ◊ Холодные
    …
- Рыбные блюда
  - ◊ Горячие
  - ◊ Холодные
    …
- …
- Горячие блюда
  - ◊ Мясные
  - ◊ Рыбные
  - …
    …

Одни пользователи будут искать котлету по-киевски прежде всего в разделе «Мясные блюда», а для других это прежде всего горячее блюдо, а потом уже мясное. Параллельная структура позволит и тем, и другим пользователям найти нужную страницу наиболее просто.

Основные средства, благодаря которым пользователь узнает о структуре, — это главная панель навигации и «хлебные крошки». Трудно представить себе эти средства, выраженные чем-то, кроме текста. Следовательно, именно текст позволяет пользователю ориентироваться на сайте и пользоваться навигацией.

Примечание

Графические элементы навигации (например, пиктограммы в виде домика или конверта) иногда уместны в качестве вспомогательных средств, но совершенно невозможно вообразить эффективную навигацию, целиком основанную на подобных пиктограммах.

Итак:

- убедитесь, что структура сайта ясна для пользователей, что они хорошо представляют себе:
  - какие задачи они могут выполнить на этом сайте, а какие — нет;
  - по какому принципу содержание сайта разбито на разделы и подразделы;
  - в каком разделе следует искать нужное им содержание;
- убедитесь, что структура и навигация сайта удобны для разных категорий пользователей, что они могут добраться до нужного содержания разными способами. Убедитесь, что на сайте есть вспомогательные средства ориентирования и поиска — карта сайта, форма поиска, указатели и т. д.;
- убедитесь, что навигация сайта доступна в академическом режиме и в нестандартных устройствах просмотра.

7.3.3. Оформление

> То, что иные превозносят как индивидуальный стиль, есть малые, ничтожные, порой даже вредные черты, часто выдаваемые за новаторство, как, например, использование лишь одного определенного вида шрифта, будь то гротесковые шрифты, или странные шрифтовые формы XIX столетия, или пристрастие к определенным смесям шрифтов, или использование мнимо смелых правил, как, скажем, применение лишь одного кегля во всей, даже сложной, работе и т. п... Только начинающие и неумные люди могут идти по такому пути.
>
> *Я. Чихольд. Как глина в руках гончара...*

Оформление — немаловажная составляющая сайта. Оно выполняет две основные функции:

- помогает прояснить содержание, структуру и навигацию сайта, выделить ключевые элементы, направить внимание пользователя на то, что ему нужно;
- передает определенный стиль и настроение, создает эмоциональный фон.

Не забывайте, что первая из этих функций — ключевая, в то время как вторая — всего лишь вспомогательная. Начинающие разработчики нередко увлекаются художественными изысками, забывая о том, для чего вообще на сайте нужно оформление.

Итак:

- убедитесь, что содержание вашего сайта доступно и без оформления. В первую очередь это касается текстового содержания;

Примечание

Размечая текстовое содержание, обеспечьте прежде всего грамотную структурную разметку. Внешний вид текста не так важен, как структурная разметка. Думайте не о том, как должен выглядеть тот или иной текстовый элемент, а о том, какую структурную роль он играет и какой элемент XHTML соответствует этой роли.

- убедитесь, что оформление не мешает пользоваться сайтом.

ProDTP :: Дизайн, шрифты, DTP! - Microsoft Internet Explorer

Файл Правка Вид Избранное Сервис Справка

Адрес: http://www.prodtp.ru/

ProDTP & VEDI

объединенный портал проектов

HOME | ACCOUNT | TOPICS | DOWNLOADS | FORUMS | SUBMIT | CONTACT US

Сейчас на сайте

AndeiMo
Designboy
Ivan
Kassian
Zappa
hotello
ious
lisica
mrh
nik7777
plam13
querulus
severmoroz
snorkel
vorchun

Вы анонимный пользователь. Можете зарегистрироваться здесь

У нас 80 гостей и 15 пользователей онлайн.

Меню

- В начало
- Мой аккаунт
- Выйти

Welcome!

Друзья, коллеги!

Обновленный "**проект ВЕДИ**" теперь обитает на этом сервере.
Не поленитесь -- зарегистрируйтесь и обживайте сайт. Зарегистрированные пользователи будут видеть скрытые от других посетителей сообщения, смогут пользоваться внутрисайтовой системой приватных сообщений и проч.
Еще не все готово и будет доделываться "на ходу". Будьте терпеливы!

Господа! Самая большая просьба -- не ломайте сайт -- здесь нет скрытых разделов с коммерческими шрифтами, программами и прочим. Все, что есть -- все на виду!

15 последних сообщений в форуме:

архив шрифтов	Kassian	25 Фев, 2004 г. - 13:00	1 ответ
Странные ассоциации	vorchun	25 Фев, 2004 г. - 12:53	0 ответов
Bulka died!!!!	ious	25 Фев, 2004 г. - 12:43	5 ответов
Посоветуйте шрифт под карандашный рисунок	Ivan	25 Фев, 2004 г. - 12:42	3 ответов
Помогите распознать	nik7777	25 Фев, 2004 г. - 12:32	6 ответов
Построение символа. Вопросы хинтинга.	nik7777	25 Фев, 2004 г. - 11:53	13 ответов
Как расслабиться после верстки?	Kassian	25 Фев, 2004 г. - 09:59	28 ответов
Текст по подъему	Mefisto	25 Фев, 2004	2 ответов

User's Login

Имя пользователя

Пароль

Запомнить меня

Вход

Все еще не зарегистрировались? Регистрация сделает комфортными Ваши посещения этого сайта, предоставит доступ ко многим дополнительным сервисам и настройкам, которые для анонимного пользователя недоступны.

Статьи, документы

О ШРИФТАХ

Практика NEW!

5 последних страниц

- Сочетаемость гарни...
- FontLab и размышле...
- ГОСТ Р 51506-99

http://www.prodtp.ru/modules.php?op=modload&name=Downloads&file=index

Интернет

Рис. 7.27. www.prodtp.ru: сайт дискредитирует отечественное издательское дело

Взгляните на оформление сайта ProDTP (рис. 7.27). Этот сайт посвящен типографскому делу и ориентирован на русскоязычную аудиторию. Тем не менее не только название сайта, но и верхняя панель навигации написаны по-английски! По-английски написана и половина заголовков («Welcome!», «User's Login»). При этом настоящее главное меню находится в нижней части страницы, за пределами первого экрана. Форма поиска по сайту расположена еще ниже.

Обратите внимание: на этом сайте, где размещены ГОСТы по издательскому делу и технические правила набора и верстки — в тексте самого сайта никакие правила и стандарты не соблюдаются! Текст набран безобразно, с опечатками, с суррогатами вместо тире и кавычек, без какого-либо типографского оформления. Статьи и выдержки из книг размещены сплошными кусками текста, без вразумительного разделения на абзацы и структурных выделений. Зато сайт обильно усыпан украшениями вроде «::» или «.:», раскрашена полоса прокрутки, применяются перекатывающиеся изображения-ссылки. Без сомнения, такое «художественное» оформление — позор для разработчиков.

Разработчикам ProDTP и многих других сайтов следовало бы прислушаться к мнению знаменитого немецкого типографа Яна Чихольда (которое, заметим, процитировано на самом сайте **www.prodtp.ru**):

> Полиграфическое исполнение решительно отстает. Его бесформенность разрушает все зачатки хорошего вкуса, серьезно препятствует его формированию. Так как многие люди из лености ума больше читают газеты, чем книги, то не удивительно, что они плохо воспринимают оформление прочей типографской продукции, включая и книги. Да и откуда наборщику, если он больше читает газеты, чем что-либо иное, получить знания о хорошем типографском вкусе? И как привыкают к плохой еде, когда не имеют лучшей, а потому и бывают лишены возможности для сравнения, так и многие из теперешних читателей привыкли к некрасивым изданиям, потому что они читают больше газеты, нежели книги, и тем самым убивают, как справедливо говорят, свое свободное время. Они не знакомы с лучшим оформлением, а потому не могут и предъявлять к нему более высоких требований. У других не достает голоса, так как и они не могут сказать, как сделать лучше.

На сайте, как и в человеке, все должно быть прекрасно. Содержание, оформление, структура, навигация и все остальные элементы должны не мешать, а помогать друг другу. Ничто не мешает делать такие сайты, которые будут одновременно и содержательными, и хорошо структурированными, и красивыми, и удобными, и доступными. Перед разработчиком не должен вставать вопрос о том, чтобы пожертвовать доступностью в пользу художественного оформления, или, наоборот, пожертвовать красотой оформления в пользу удобства. Современные технологии позволяют делать такие сайты, где все компоненты удачно сочетаются между собой и где не

приходится ухудшать некоторые из них ради улучшения каких-нибудь других.

Единственной причиной, по которой какой-либо из компонентов сайта может быть сделан недостаточно хорошо, является низкая квалификация разработчика или его слабая профессиональная культура. Но и то, и другое всегда можно повысить. Автор надеется, что вы относитесь к таким разработчикам, которые всегда стремятся сделать свою работу наилучшим образом и заботятся о повышении своей квалификации, своего опыта и культурного уровня. (В противном случае, вы вряд ли вообще стали бы читать эту книгу.)

Пора и честь знать
Заключение

Ну что сказать вам, москвичи,
На прощанье?
Чем наградить мне
Вас за вниманье?

В. Масс, М. Червинский. Дорогие москвичи

— Хорош ли мой дизайн, о Учитель? — спросил ученик.
— Нет, — ответил Учитель.
— Хорош ли мой дизайн теперь? — спросил ученик, переделав работу.
— Нет, — снова ответил Учитель.
— Хорош ли он, наконец, теперь? — спросил ученик, переделав работу еще раз.
— Нет, — ответил Учитель.
— Но что же мне сделать, чтобы мой дизайн был хорош? — спросил ученик.
— Ничего! — ответил Учитель. — Вот тебе истина, запомни же ее: дизайн не может быть хорошим! Всякий дизайн в чем-то да плох!

Р. Чебыкин. Дзен-дизайн

Разрабатывая текстовое содержание сайта, не следует забывать и о других его составляющих. Автор этой книги не хотел бы, чтобы у вас сложилось впечатление, будто сайт должен состоять только из одного текста. Для получения полной картины прочтите, кроме этой книги, также литературу о графике для Всемирной Паутины, о разработке интерфейсов пользователя, о кодировании HTML и CSS, о программировании клиентских и серверных приложений, а также о других технологиях, имеющих отношение к разработке сайтов.

Что такое хороший сайт? Это сайт, удовлетворяющий двум главным условиям:

- во-первых, *этот сайт нужен*, полезен, востребован. Пользователи заинтересованы в том, чтобы часто посещать этот сайт, так как он предоставляет нужные им ресурсы, помогает выполнять их задачи;

Примечание

В Сети очень много ненужных, бесполезных сайтов, которые не посещает никто, кроме разработчика и двух-трех его ближайших знакомых. Хотите ли вы, чтобы ваши сайты тоже вошли в эту категорию?

- во-вторых, *пользователи не чувствуют затруднений*, работая с этим сайтом. Они не путаются в гиперссылках, не совершают лишних щелчков мышью, не встречаются с неприятными сюрпризами и недоразумениями и т. д.

Примечание

Даже если сайт предоставляет нужные и полезные ресурсы, это еще не значит, что он автоматически удобен и привлекателен для пользователей. Добивайтесь того, чтобы пользователи не только находили на вашем сайте то, за чем они туда пришли, но и не чувствовали бы при этом, что им что-то неудобно или непонятно.

Профессионализм разработчика состоит именно в том, чтобы делать сайты, в наибольшей степени удовлетворяющие этим условиям. Эта книга не превратит вас моментально в профессионалов, но автор надеется, что она хотя бы немного поможет вам на пути к мастерству.

Чаще всего, чтобы сайт получился хорошим, разработчику сто́ит позаботиться о следующих аспектах:

- **содержание** — это именно то, ради чего пользователи посещают сайт. Содержание должно быть полезным, интересным, компетентным, актуальным, грамотно подобранным. Нередко в группу поддержки профессиональных сайтов входит отдельный специалист по подбору и организации содержания (по-английски эта должность называется content manager);

Примечание

Содержание сайта вовсе не обязано быть текстовым. Содержанием виртуальной картинной галереи являются электронные репродукции картин; содержанием архива загружаемого программного обеспечения являются файлы — установочные комплекты программ. Содержанием электронного магазина является не что иное, как служба, обеспечивающая работу этого магазина, и т. д.

Эта книга посвящена только одному из многочисленных типов содержания, которые могут быть на сайте.

- **структура** — чтобы пользователь легко работал с содержанием сайта, оно должно быть организовано в логичную и удобную структуру. Сайты чаще всего имеют иерархическую структуру: все содержание делится на отдельные разделы, те — на подразделы и т. д. Разрабатывайте для сайтов

такую структуру, чтобы она помогала пользователю, а не вводила его в заблуждение;

- **навигация** — ответственно подходите к разработке навигации, тщательно продумывайте систему гиперссылок. Исследуйте маршруты, по которым пользователи ходят по вашему сайту. Повторяйте на каждой странице привычные элементы навигации — ссылку на главную страницу, основную панель навигации и «хлебные крошки». Добивайтесь того, чтобы для перемещения от любой страницы вашего сайта к любой другой пользователю требовалось совершать как можно меньше переходов по ссылкам;

纸 Дополнительно

Просмотрев протоколы веб-сервера, вы убедились, что подавляющее большинство пользователей посещает одни и те же 3–4 страницы вашего сайта? Так помогите этим пользователям: сделайте ссылки на эти 3–4 страницы непосредственно с главной страницы сайта!

- **оформление** — не впадайте в крайности. Бывают сайты, ошеломляющие пользователей непревзойденными художественными достоинствами — но при этом не предоставляющие ничего, что было бы хоть кому-то действительно нужно. И, наоборот, бывают сайты, где нет вообще никакого или почти никакого оформления; они выглядят скучно и пресно. Постарайтесь достичь разумного баланса. Разрабатывайте именно такое оформление, которое поможет пользователям воспринимать содержание сайта и оставит у них приятное, положительное впечатление;
- **доступность** — не забывайте, что сайты посещают самые разные пользователи с самыми разными устройствами просмотра. Это и персональные компьютеры множества разновидностей, с разнообразными видеорежимами, а также карманные компьютеры (наладонники), коммуникаторы, сотовые телефоны, речевые браузеры... Всевозможные портативные устройства распространяются все больше, и плох тот разработчик, который не задумывается о совместимости сайтов с этими устройствами;
- **технологии** — какую редакцию HTML использовать для кодирования страниц — HTML 4.0, XHTML 1.0 или еще какую-нибудь? Какую разновидность применять — Strict, Transitional, Frameset? На каком сервере размещать сайт? Какую систему управления базами данных выбрать — MySQL, PostgreSQL, Oracle? На каком языке писать серверные приложения — на PHP, Perl, Java? Разработчику требуется определиться с этими вопросами, выбрав именно те технологии, которые лучше всего отвечают целям и задачам сайта;

Примечание

Существует немало сайтов, сделанных на PHP, для которых, тем не менее, гораздо лучше подошла бы другая технология — скажем, Perl или JSP. Почему же эти сайты разработаны на PHP? Ответ простой: потому что разработчики знали PHP и не знали ни одного из тех языков, которые в данной ситуации подходят лучше.

Профессиональные сетевые проекты осуществляются под руководством квалифицированных технических директоров, которые умеют правильно выбрать технологию и найти разработчиков, владеющих этой технологией. Но даже если вы не работаете в группе под управлением такого директора, будьте техническим директором сами для себя: если вы видите, что та или иная технология не годится для разрабатываемого сайта — не используйте ее, даже если вы очень хорошо ее знаете.

- **безопасность** — для всякого сайта важна защищенность от различного рода неприятностей. Вопреки распространенному мнению, атаки злоумышленников — не главная из этих неприятностей. Гораздо чаще проблемы возникают из-за недостаточно продуманного программирования «движка» сайта, из-за плохо настроенного веб-сервера или из-за того, что пользователи случайно совершают на сайте действия, не предусмотренные разработчиками.

Примечание

Если сайт размещен на «слабом» сервере, то слишком большое количество посещений может его погубить. Убедитесь, что аппаратное и программное обеспечение сервера соответствует реальной нагрузке вашего сайта.

Дополнительно

Позаботьтесь о защите персональной информации пользователей. Если для доступа к некоторым ресурсам вашего сайта требуется авторизация — убедитесь, что посторонние не могут получить доступ к паролям пользователей. Если пользователь заполняет какую-либо форму, где указывает персональные данные (скажем, дату рождения и адрес электронной почты), — также убедитесь, что эта информация не может попасть в чужие руки. И, конечно, еще важнее заботиться о безопасности, если через ваш сайт осуществляются финансовые операции или удаленное управление компьютерными системами.

Привлеките специалистов, которые смогут протестировать ваш сайт и помогут укрепить его безопасность.

Все компоненты сайта должны хорошо сочетаться друг с другом. Это касается не только элементов оформления, но и всего остального: оформление должно соответствовать содержанию, навигация должна подходить к структуре, структура должна отражаться в оформлении и т. д.

Примечание

Возвращаясь к основной теме этой книги, хочется выразить недоумение по поводу некоторых сайтов, разработчики которых уделяют слишком много внимания графическому оформлению. На таких сайтах обычно бросается в глаза разительное отличие между прекрасно подобранными изображениями и совершенно безобразным текстом. Если вы хотите, чтобы ваш сайт действительно выглядел красиво — не забывайте, что для этого текст тоже должен быть красивым. Он должен быть грамотно составлен, набран с соблюдением типографских стандартов, правильно сверстан и оформлен. Иначе графическое оформление будет смотреться на вашем сайте, как кружева на корове.

Автор желает всем читателям этой книги профессиональных и творческих успехов. Однако не забывайте, что здесь содержится лишь малая часть тех знаний, которые необходимы настоящему разработчику сайтов. Повышайте свое мастерство, набирайтесь опыта, старайтесь, чтобы каждая последующая ваша работа была хоть в чем-то лучше предыдущей.

Последнее пожелание автор этой книги будет применять и к себе. Все ошибки и недостатки, выявленные в этой книге, не должны повториться в следующих. Вы можете помочь, если сообщите автору свое мнение о книге, проинформируете о замеченных ошибках и выскажете полезные замечания и комментарии. Адрес электронной почты Ростислава Чебыкина — **philigon@master.elserv.msk.su**.

Всего вам доброго!

А теперь — перечитайте эту книгу еще раз.

Смотрит в книгу — видит ...

Рекомендуемая литература

Ваши книги дали мне мудрость. Все то, что веками создавала неутомимая человеческая мысль, сдавлено в моем черепе в небольшой ком. Я знаю, что я умнее всех вас.
И я презираю ваши книги, презираю все блага мира и мудрость. Все ничтожно, бренно, призрачно и обманчиво, как мираж.

А. П. Чехов. Пари

Орфография, пунктуация и стилистика русского языка

Не ставьте точек над «У» — оно в этом не нуждается.

В. Маяковский

1. Д. Э. Розенталь. Справочник по правописанию и литературной правке. М., Айрис-пресс, 2003.

   Пожалуй, это лучший справочник, предназначенный для широкого круга читателей. В нем рассматриваются все вопросы, с которыми сталкивается обычный «пользователь» русского языка. Автор этой книги часто обращается к справочнику Розенталя, когда пишет собственные тексты.

   Конечно, если вы уже являетесь или собираетесь быть профессиональным редактором, корректором или техническим писателем — этого справочника вам будет недостаточно. Обратитесь к специальной литературе, а еще лучше — поступите в профильный вуз.

2. Орфографический словарь русского языка. Под ред. В. В. Лопатина. М., Дрофа, Русский язык, 1999.

   Если вы имеете хоть какое-то отношение к разработке текстов — обязательно заведите себе орфографический словарь. Нет ничего постыдного в том, чтобы иногда сверять по нему правописание того или иного слова. Гораздо более постыдно — если вы не знаете точно, как пишется слово, и поэтому пишете его, как вам в голову взбредет.

Издательское дело

Кто поверит теперь, что в годы блокады, голодовки и бумажного кризиса, когда население Совроссии читало газеты только на стенах домов, в Шкидской маленькой республике с населением в шестьдесят человек выходило 60 (шестьдесят) периодических изданий — всех сортов, типов и направлений?

Г. Белых, А. Пантелеев. Республика ШКИД

1. П. Г. Гиленсон. Справочник художественного и технического редакторов. М., Книга, 1988.

   Это по-настоящему культовый справочник, выпуск которого совпал с окончанием эпохи «бескомпьютерного» издательского дела в нашей стране. Достать его сейчас сложно, а те, кому это удалось, берегут этот справочник, как зеницу ока, и сверяют по нему каждый свой шаг в работе над книгой.

   Справочник Гиленсона посвящен двум вопросам: какой должна быть готовая книга и как технически этого добиться. Все, что относится ко второму вопросу, конечно, устарело — но все оставшееся актуально и сейчас и, по всей видимости, останется таковым еще много десятилетий.

2. М. В. Шульмейстер. Ручной набор. М., Книга, 1967.

   Это квинтэссенция, вершина отечественного издательского дела XX века. Эта книга закрепила нормы и традиции, сложившиеся в советском книгопечатании, и определила эти нормы на несколько последующих десятилетий. Как и в случае со справочником Гиленсона, «Ручной набор» устарел в той части, где говорится о конкретных технологиях. Но все остальные принципы и рекомендации применимы и в эпоху компьютерных издательских технологий.

   纸 ***Дополнительно***

   Например, вы знаете, почему один из наиболее предпочтительных размеров абзацного отступа — 1,5 em? Ответ на этот и многие подобные вопросы можно найти в книге Шульмейстера.

   Остается добавить, что сама книга набрана и сверстана превосходно; ее можно, даже не читая, применять как образец для набора и верстки любых текстов.

3. Я. Чихольд. Облик книги. Избранные статьи о книжном оформлении. М., Книга, 1980.

   Ян Чихольд в издательском деле — примерно то же самое, что Шекспир в литературе или Нильсен в разработке сайтов. На его статьях и книгах, переведенных на десятки языков, учились специалисты во всех странах мира.

Вместе с тем, нужно учитывать, что Чихольд рассказывает о западном опыте издательского дела, и некоторые его рекомендации требуют поправки на отечественную специфику.

纸 ***Дополнительно***

Например, в английском и многих других языках, основанных на латинском алфавите, оптимальный междусловный пробел составляет ⅓ кегельной, а в русском языке — ½ кегельной. Это связано с тем, что в тексте на английском языке преобладают буквы с округлыми формами, а на русском — с прямыми вертикальными штрихами. Автор книги, которую вы сейчас читаете, даже не задумывался о таких тонкостях, пока не познакомился с трудами Чихольда.

4. А. Э. Мильчин, Л. К. Чельцова. Справочник издателя и автора. Редакционно-издательское оформление издания. М., Олма-Пресс, 2003.

   В этом справочнике рассматриваются все принципы, незнание которых приводит к наиболее распространенным ошибкам в печатном тексте. Ставить ли точку после заголовка, в какие кавычки заключать цитату, как составлять сокращения и аббревиатуры, в каких случаях записывать числительные цифрами, а в каких — прописью — все эти вопросы здесь подробно рассматриваются.

   Немалая часть справочника посвящена грамотному оформлению списков и таблиц. Если вы используете списки и таблицы на своем сайте — обязательно прочтите справочник Мильчина и Чельцовой, иначе вы рискуете допустить любительские ошибки.

5. А. Э. Мильчин. Издательский словарь-справочник. М., Олма-Пресс, 2003.

   Вы забыли, что такое «кегль», «интерлиньяж», «красная строка»? Вы не понимаете разницу между жирным и полужирным шрифтом? Вы называете «выравниванием» то, что на самом деле называется выключкой? Тогда вам обязательно нужно иметь под рукой этот словарь.

   Словарь-справочник основывается на ГОСТах и других стандартах, действующих на момент издания — то есть на 2003 год. Это большое преимущество по сравнению со справочником Гиленсона и руководством Шульмейстера.

6. В. Кричевский. Типографика в терминах и образах. М., Слово, 2000.

   Этот двухтомник сверстан и издан очень оригинально; в книжном магазине он сразу притягивает к себе внимание и уже не отпускает его. Второй том целиком занят разнообразными иллюстрациями, примерами типографского оформления текста в самых разных ситуациях — от этикеток на коробках до научных трудов.

7. Г. П. Вильберг, Ф. Форсман. Азбука книжного дизайна. СПб, изд-во СПбГПУ, 2003.

   Эта книга — действительно азбука, она предназначена для тех, кто только начал учиться издательскому делу. В ней нет длинных текстов, а есть множество иллюстраций практического применения тех или иных принципов и рекомендаций.

В разделе, где обсуждаются разные типы бумаги для печати, даже вставлены несколько страниц разной бумаги!

Правда, как и в случае с трудом Чихольда, эта книга отражает западные (прежде всего немецкие) нормы издательского дела.

8. Дж. Феличи. Типографика: шрифт, верстка, дизайн. СПб, БХВ-Петербург, 2004.

Эта книга, в отличие от всех предыдущих, — плод эпохи компьютерных издательских технологий. И, тем не менее, ее главная мысль — в том, что несмотря на эти технологии, следует знать и уважать традиции издательского дела, складывавшиеся на протяжении столетий.

По большей части книга «Типографика» — о том, как применять современные компьютерные технологии таким образом, чтобы выпускать не любительские поделки, а настоящие шедевры издательского искусства.

9. П. Каров. Шрифтовые технологии: описание и инструментарий. М., Мир, 2001.

Из этой книги можно узнать основные принципы работы компьютерных издательских систем со шрифтами, способы описания и построения компьютерных шрифтов. В книге интересна только теоретическая часть; все, касающееся конкретных технологий, устарело лет на двадцать. (Это, к сожалению, особенность многих переводных книг.)

Отдельного внимания заслуживает включенная в книгу статья Н. Ефимова об истории кириллических шрифтов.

10. Е. В. Малышкин, А. Э. Мильчин, А. А. Павлов, А. Е. Шадрин. Настольная книга издателя. М., АСТ, Олимп, 2003.

Выход этой книги показывает, что отечественное книгопечатание преодолело засилие «компьютерно грамотных» дилетантов, и издатели начинают вспоминать о том, что в их профессии есть определенные стандарты, нормы, принципы и традиции. «Настольная книга издателя» показывает, что современные компьютерные технологии не только не отменяют опыт предыдущих веков, но и, наоборот, позволяют еще точнее приблизить печатные издания к идеалу, о котором мечтали печатники со времен Гутенберга и Ивана Федорова.

В книге много рассказывается о настольной издательской системе Corel Ventura, но важно помнить, что это все-таки не руководство по этой программе, а прекрасное руководство по издательскому процессу в условиях современного технического обеспечения.

Разработка сайтов

«Чтоб музыкантом быть, так надобно уменье
И уши ваших понежней», —
Им отвечает Соловей.

И. А. Крылов. Квартет

1. Я. Нильсен. Веб-дизайн. СПб, Символ-Плюс, 2002.

   Это самое известное в мире руководство по удобству и доступности сайтов. Оно переведено на десятки языков, и на некоторых языках претерпело уже не одно переиздание. Вы можете соглашаться или не соглашаться с мнением Нильсена, но прочитать эту книгу просто обязаны — если, конечно, хотите быть профессиональным разработчиком сайтов.

2. Я. Нильсен, М. Тахир. Дизайн веб-страниц. Анализ удобства и простоты использования 50 узлов. М., Вильямс, 2002.

   В этой книге очень много примеров. Авторы выбрали главные страницы 50 сайтов и по косточкам разобрали эти страницы вплоть до мельчайших элементов оформления. К сожалению, в русском издании допущено много опечаток — особенно в иллюстрациях. Возможно, при следующем издании этой книги опечатки будут устранены, и вы сможете наслаждаться прекрасным практическим руководством по улучшению сайтов.

3. С. Круг. Веб-дизайн, или Не заставляйте меня думать! СПб, Символ-Плюс, 2001.

   Текст увлекательно изложен и легко читается, в книге множество цветных иллюстраций. Основная идея книги — делайте сайты как можно проще и понятнее обычному посетителю. И люди к вам потянутся.

   Один из наиболее интересных фрагментов книги — описание практически беззатратной, но вместе с тем эффективной методики тестирования удобства сайтов.

4. К. Гото, Э. Котлер. Веб-редизайн. СПб, Символ-Плюс, 2003.

   Основная идея книги: «Ваш сайт никуда не годится. Его надо переделать». Однако, если вы не захлопнете книгу после такого «откровения», вас ждет подробное пошаговое описание всего технического процесса разработки сайта — от первоначальных переговоров с заказчиком до следующей переработки сайта. В книге множество схем, диаграмм и контрольных таблиц, которые можно немедленно применять на практике.

   Книга хороша тем, что в ней скрупулезно рассматриваются все аспекты взаимоотношений разработчиков и заказчиков сайта — переговоры, документы, оплата и т. д.

5. К. Уодтке. Информационная архитектура. Чертежи для сайта. М., Кудиц-Образ, 2004.

   В нашей стране очень мало специалистов по информационной архитектуре. Мало того, многие разработчики сайтов вообще не знают о том, что это такое. А между тем, информационная архитектура — это дисциплина, без которой не может получиться хорошего сайта.

   Книга небольшая по объему, с множеством иллюстраций — включая фотографии Якоба Нильсена, корчащего забавные рожи.

6. Дж. Прайс, Л. Прайс. Текст для Web: доступность и привлекательность. М., Вильямс, 2003.

Тематика этой книги пересекается с той, которую вы сейчас читаете. Однако книга предназначена прежде всего для разработчиков сайтов на английском языке. В переводе некоторые моменты выглядят очень забавно — например, когда авторы углубляются в тонкости употребления артиклей или используют шутки и цитаты, понятные только носителям английского языка.

7. М. Дубаков. Веб-мастеринг средствами CSS. СПб, БХВ, 2002.

8. М. Дубаков. Создание Web-страниц. Искусство верстки. Минск, Новое знание, 2004.

Положительное качество обеих книг Дубакова — в том, что, кроме них, у нас вообще нет книг, посвященных CSS. Еще одно положительное качество — их автор является действующим разработчиком сайтов и излагает опыт, полученный в ходе практической работы. Если вы хотите знать, как настоящие профессионалы используют CSS — вы просто не можете пройти мимо этих книг.

Хотя у книг есть и недостатки. Во-первых, автор иногда увлекается оформлением сайтов в ущерб их удобству, доступности и содержательной ценности. Во-вторых, обе книги устарели года на два относительно даты своего выхода. Например, в них много внимания уделяется канувшему в лету браузеру Netscape 4 и нет ничего про современные версии Opera и Mozilla, которые как раз отличаются улучшенной поддержкой CSS.

9. К. Шмитт. Создание Web-страниц средствами CSS. М., Кудиц-Образ, 2003

Сравните опыт зарубежных кодировщиков CSS (он изложен в этой книге) с опытом отечественных специалистов. К сожалению, наши разработчики заимствуют слишком много решений у зарубежных коллег, не всегда задумываясь о том, применимы ли эти решения к сайтам на русском языке.

Ресурсы Всемирной Паутины

Меч Зари опускался! Он опускался медленно, и Хокмун заметил, что чудесный клинок висит на паутине из металлических нитей. И что в этой паутине находится что-то еще...

М. Муркок. Рунный посох

Стандарты и документация

1. **www.w3.org**

Консорциум W3C: официальные стандарты HTML, XHTML и CSS, стандарты и документация по многим другим технологиям Интернета.

2. **www.unicode.org**

   Консорциум Unicode: официальный стандарт Unicode, таблицы символов, справочная информация о Unicode и мировых системах письменности.

3. **www.rfc-editor.org**

   Сборник сетевых стандартов серии RFC — в том числе стандарты протоколов IP (RFC 791) и HTTP (RFC 1945 и RFC 2616).

纸 ***Дополнительно***

Пожалуйста, не пренебрегайте стандартами и официальной документацией. Не забывайте, что настоящий мастер — не тот, кто не нарушает никаких правил, а тот, кто точно знает, когда и какие правила можно нарушать, а когда — нет. Получается — для того, чтобы стать мастером, нужно как минимум знать правила и стандарты.

Другие ресурсы

1. **www.useit.com**

   Сайт Якоба Нильсена — всемирно известного эксперта в области удобства и доступности сайтов. Каждые две недели на этом сайте появляется новая статья Нильсена — хотя бы ради этого сайт стоит почаще посещать.

2. **www.w3schools.com**

   Множество учебной и справочной информации по всем аспектам разработки сайтов — от HTML и CSS до SQL и PHP. Интерактивные тесты, с помощью которых вы можете проверить уровень своих знаний.

Браузеры

1. **www.microsoft.com/windows/ie_intl/ru/** — Microsoft Internet Explorer
2. **www.mozilla.org** — Mozilla, Camino
3. **www.opera.com** — Opera
4. **www.apple.com/safari/** — Safari (популярный браузер для Mac OS)