

Создание эффективных веб-страниц

6-е издание



HTML и XHTML

Подробное руководство



O'REILLY®

Чак Муссиано и Билл Кеннеди

По договору между издательством «Символ-Плюс» и Интернет-магазином «Books.Ru – Книги России» единственный легальный способ получения данного файла с книгой ISBN 5-93286-104-5, название «HTML и XHTML. Подробное руководство», 6-е издание – покупка в Интернет-магазине «Books.Ru – Книги России». Если Вы получили данный файл каким-либо другим образом, Вы нарушили международное законодательство и законодательство Российской Федерации об охране авторского права. Вам необходимо удалить данный файл, а также сообщить издательству «Символ-Плюс» (piracy@symbol.ru), где именно Вы получили данный файл.

HTML & XHTML

The Definitive Guide

Sixth Edition

Chuck Musciano and Bill Kennedy

O'REILLY®

HTML и XHTML

Подробное руководство

Шестое издание

Чак Муссиано и Билл Кеннеди



Санкт-Петербург — Москва
2008

Чак Муссиано и Билл Кеннеди
**HTML и XHTML. Подробное руководство,
6-е издание**

Перевод С. Иноземцева

Главный редактор	<i>А. Галунов</i>
Зав. редакцией	<i>Н. Макарова</i>
Выпускающий редактор	<i>В. Овчинников</i>
Научный редактор	<i>О. Циллюрик</i>
Редактор	<i>Е. Орлова</i>
Корректоры	<i>О. Макарова, С. Минин</i>
Верстка	<i>Д. Орлова</i>

Муссиано Ч., Кеннеди Б.

HTML и XHTML. Подробное руководство, 6-е издание. – Пер. с англ. – СПб: Символ-Плюс, 2008. – 752 с., ил.

ISBN-10: 5-93286-104-5

ISBN-13: 978-5-93286-104-2

Шестое издание «HTML и XHTML» – самая полная и современная книга по языкам HTML и XHTML, разъясняющая работу и взаимодействие каждого их элемента. Она удачно сочетает в себе лучшие качества понятного учебного пособия, адресованного начинающим, и всеобъемлющего справочника, который всегда под рукой даже у опытных веб-программистов. Этот труд, ставший классическим, содержит все от базового описания синтаксиса и семантики до практических советов, поможет вам найти свой неповторимый стиль и в совершенстве овладеть языком веб-дизайна.

Описаны стандарты HTML 4.01, XHTML 1.0 и CSS2, приведен обзор еще не вступивших в силу стандартов XHTML 2 и CSS3. Уделено внимание и новейшим инициативам разработчиков XHTML (XForms, XFrames и модуляризации), а также основам XML. Рассмотрены: управление внешним видом документа с помощью таблиц стилей; работа с HTML-кодом, сгенерированным автоматически; работа с фреймами, интерактивными формами, динамическими документами; интеграция HTML-кода с мультимедийными данными, сценариями JavaScript и Java-апплетами.

ISBN-10: 5-93286-104-5

ISBN-13: 978-5-93286-104-2

ISBN 0-596-52732-2 (англ)

© Издательство Символ-Плюс, 2008

Authorized translation of the English edition © 2006 O'Reilly Media, Inc. This translation is published and sold by permission of O'Reilly Media, Inc., the owner of all rights to publish and sell the same.

Все права на данное издание защищены Законодательством РФ, включая право на полное или частичное воспроизведение в любой форме. Все товарные знаки или зарегистрированные товарные знаки, упоминаемые в настоящем издании, являются собственностью соответствующих фирм.

Издательство «Символ-Плюс». 199034, Санкт-Петербург, 16 линия, 7, тел. (812) 324-5353, www.symbol.ru. Лицензия ЛП N 000054 от 25.12.98.

Налоговая льгота – общероссийский классификатор продукции ОК 005-93, том 2; 953000 – книги и брошюры.

Подписано в печать 21.02.2008. Формат 70×100^{1/16}. Печать офсетная.

Объем 47 печ. л. Тираж 2000 экз. Заказ N

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

*Эту книгу мы посвящаем нашим женам
Синди, Кортни и Коул
и детям Жанне, Еве и Этан.*

*Если бы не их любовь и терпение,
то у нас не было бы ни сил ни времени,
чтобы написать ее.*

Оглавление

Предисловие	11
1. HTML, XHTML и World Wide Web	22
1.1. Интернет	22
1.2. Основные понятия Интернета	26
1.3. HTML и XHTML – что они собой представляют	32
1.4. HTML и XHTML – чем они не являются	33
1.5. Стандарты и расширения	34
1.6. Инструменты веб-дизайнера	36
2. Быстрый старт	39
2.1. На чем писать	39
2.2. Первый HTML-документ	40
2.3. Встраиваемые теги	42
2.4. Костяк HTML	43
2.5. Плоть HTML- и XHTML-документов	43
2.6. Текст	45
2.7. Гиперссылки	50
2.8. Изображения – это особая статья	54
2.9. Списки, строка поиска, формы	57
2.10. Таблицы	59
2.11. Фреймы	60
2.12. Таблицы стилей и JavaScript	61
2.13. Вперед!	62
3. Анатомия HTML-документа	63
3.1. Внешность обманчива	63
3.2. Структура HTML-документа	65
3.3. Теги и атрибуты	65
3.4. Корректные документы и XHTML	70
3.5. Содержимое документа	71
3.6. Элементы HTML/XHTML-документа	73
3.7. Заголовок документа	76
3.8. Тело документа	80
3.9. Редакторская разметка	81
3.10. Тег <bdo>	84

4. Текст	86
4.1. Разделы и абзацы	86
4.2. Заголовки	95
4.3. Управление внешним видом текста	101
4.4. Теги логической разметки	102
4.5. Теги физической разметки	110
4.6. Точные интервалы и макет	115
4.7. Блоки цитат	128
4.8. Адреса	132
4.9. Кодирование специальных символов	134
4.10. Использование расширенной HTML-модели шрифтов (нежелательно)	135
5. Линейки, изображения и мультимедийные элементы	143
5.1. Горизонтальные линейки	143
5.2. Изображения в документе	151
5.3. Цвета документа и фоновые изображения	181
5.4. Звуковой фон	189
5.5. Анимация текста	191
5.6. Другое мультимедийное содержимое	195
6. Гиперссылки и Сети	198
6.1. Основы гипертекста	198
6.2. Ссылки на документы: URL	199
6.3. Создание гиперссылок	219
6.4. Эффективное применение гиперссылок	228
6.5. Изображения, реагирующие на мышшь	233
6.6. Создание поисковых документов	245
6.7. Отношения	249
6.8. Поддержка автоматической обработки и создания документов	253
7. Форматированные списки	257
7.1. Неупорядоченные списки	257
7.2. Упорядоченные списки	261
7.3. Тег 	264
7.4. Вложенные списки	267
7.5. Списки определений	269
7.6. Как использовать списки	273
7.7. Директории	274
7.8. Меню	275
8. Каскадные таблицы стилей	277
8.1. Элементы стилей	278
8.2. Синтаксис стилей	288

8.3. Стиливые классы	295
8.4. Стиливые свойства	301
8.5. Бестеговые стили – тег 	363
8.6. Применение стилей в документах	364
9. Формы	368
9.1. Формы – основные понятия	369
9.2. Тег <form>	370
9.3. Простой пример формы	378
9.4. Получение данных из форм при помощи электронной почты	379
9.5. Тег <input>	382
9.6. Тег <button>	394
9.7. Многострочные области ввода текста	396
9.8. Элементы множественного выбора	398
9.9. Атрибуты формы общего назначения	402
9.10. Группировка элементов формы и обеспечение их надписями	406
9.11. Эффективные формы	410
9.12. Программирование форм	414
10. Таблицы	421
10.1. Стандартная модель таблиц	421
10.2. Основные теги таблицы	423
10.3. Новейшие теги таблицы	442
10.4. За пределами обычных таблиц	453
11. Фреймы	455
11.1. Обзор фреймов	455
11.2. Теги фреймов	456
11.3. Макетирование фреймов	458
11.4. Содержимое фреймов	464
11.5. Тег <noframes>	467
11.6. Встроенные фреймы	469
11.7. Окна и фреймы в качестве цели	471
11.8. Модель XFrames	476
12. Исполняемое содержимое	479
12.1. Апплеты и объекты	480
12.2. Вложенное содержимое	484
12.3. JavaScript	501
12.4. Таблицы стилей JavaScript (устарели)	510
13. Динамические документы	517
13.1. Обзор динамических документов	517

13.2. Client-pull-документы	518
13.3. SP-документы	524
14. Мобильные устройства	529
14.1. Мобильный Интернет	529
14.2. Фактор устройства	532
14.3. Язык XHTML Basic	535
14.4. Эффективный веб-дизайн мобильного содержимого	539
15. XML	546
15.1. Языки и метаязыки	547
15.2. Документы и DTD	550
15.3. Как читать XML DTD	551
15.4. Грамматика элементов	556
15.5. Атрибуты элементов	561
15.6. Условные разделы	564
15.7. Построение XML DTD	565
15.8. Использование XML	566
16. XHTML	570
16.1. Зачем нужен XHTML?	571
16.2. Создание документов XHTML	573
16.3. HTML и XHTML	576
16.4. XHTML 1.1	582
16.5. Использовать ли XHTML?	583
17. Трюки, штуки и советы	587
17.1. Главный совет	587
17.2. Доработка документа после HTML-редактора	589
17.3. Трюки с таблицами	594
17.4. Фокусы с окнами и фреймами	600
A. Грамматика HTML	604
B. Краткий справочник по тегам HTML/XHTML	615
C. Краткий справочник по свойствам каскадных таблиц стилей	649
D. HTML 4.01 DTD	660
E. XHTML 1.0 DTD	679
F. Коды символов	699
G. Названия и коды цветов	705
H. Расширенные средства макетирования для Netscape	709
Алфавитный указатель	733

Предисловие

Изучение языка разметки гипертекста (Hypertext Markup Language, HTML) и расширяемого языка разметки гипертекста (Extensible Hypertext Markup Language, XHTML) подобно изучению любого другого языка – естественного или компьютерного. Многие учащиеся первым делом углубляются в примеры. Изучение чужих работ – это естественный, легкий и приятный способ обучения. Поэтому мы советуем всякому, кто хочет изучить HTML и XHTML, подключиться к Сети и с помощью какого-либо браузера своими глазами увидеть, что выглядит хорошо, что эффективно работает и что может оказаться полезным. Изучайте чужое творчество и размышляйте над открывающимися возможностями. Именно подражая работам друг друга, большинство современных веб-мастеров выучило язык.

Подражание, однако, полезно только до известного предела. Образцы могут оказаться и удачными, и не очень. На пути подражания можно познакомиться с языком только в «первом приближении». Чтобы овладеть языком в полной мере, нужно узнать, как использовать его во множестве различных ситуаций. Всему можно научиться на примерах, если жить достаточно долго.

Компьютерные языки отличаются от естественных строгостью синтаксических конструкций, хотя языки разметки намного более снисходительны к пишущему, чем языки программирования. Достаточно одной ошибки в записи – и ничего не будет работать. Кроме того, существует проблема «стандартов». Академические комитеты и промышленные эксперты определяют грамматику и правила употребления компьютерных языков, подобных HTML. Проблема состоит в том, что браузерные технологии, используемые вами и вашей аудиторией, не всегда придерживаются стандартов. Некоторым, например тем, что реализованы в браузерах получающих все более широкое распространение мобильных устройствах, это просто не под силу, а разработчики других создают собственные версии языка, пренебрегая стандартами.

Сами стандарты также меняются. HTML преобразовался в XHTML – HTML с использованием расширяемого языка разметки (Extensible Markup Language, XML). HTML и XHTML настолько похожи, что мы часто говорим о них как об одном языке. Есть между ними и коренные различия, о которых рассказывается далее в этом предисловии.

Полное, отражающее последние нововведения руководство по языку, которое охватывает синтаксис, семантику и детальное описание изменений и поможет отличить хороший текст от плохого, – вот надежный путь усвоения HTML и XHTML.

На пути к овладению языком нужно сделать еще один шаг. Чтобы стать настоящим мастером, нужно выработать свой собственный стиль. Это подразумевает умение выдать не просто работающий код, но код, который будет работать эффективно. «Порядок слов» имеет большое значение, так же как и структурирование материала в пределах документа, нескольких документов или собраний документов.

Наша цель при написании данной книги – помочь овладеть HTML и XHTML, дать вам исчерпывающие сведения о синтаксисе, семантике и правилах хорошего стиля. Мы избираем естественный подход к изучению – через примеры (хорошие, разумеется). Мы детально изучим все элементы современных версий этих языков (HTML 4.01 и XHTML 1.0), так же как и все расширения, поддерживаемые популярными браузерами. Мы объясним, как каждый из элементов работает и как он взаимодействует с другими элементами.

И со всем почтением к Странку и Уайту¹ на протяжении книги мы будем давать советы, касающиеся стиля и организации, чтобы помочь научиться эффективно использовать HTML и XHTML для выполнения различных задач – от создания простой онлайн-документации до сложных маркетинговых и торговых презентаций. Мы покажем вам, что работает, а что нет, расскажем, что существенно для тех, кто смотрит на ваши странички, и предупредим о возможных ошибках.

Короче говоря, эта книга – полное руководство по созданию документов с использованием HTML и XHTML, начиная с основ синтаксиса и семантики и заканчивая разнообразными рекомендациями в отношении стиля, призванное помочь в создании красивых, информативных и доступных документов, которые вы с гордостью сможете представить своим читателям.

Для кого эта книга

Мы написали эту книгу для всех тех, кто интересуется изучением и использованием языков, применяемых в Интернете, от самых обычных пользователей до профессиональных веб-дизайнеров. Мы не предполагаем у читателей никакого опыта в HTML и XHTML. На самом деле мы даже не ожидаем, что вы бывали во Всемирной паутине, хотя удивились бы, если бы вы до сих пор ни разу не попробовали познакомиться с ней. Нет необходимости иметь подключение к Интернету, чтобы чи-

¹ William Strunk Jr. и Elwyn Brooks White – авторы знаменитой книги «The Elements of Style», ставшей «классикой жанра», которую специалисты по стилю называют не иначе как «Strunk & White». – *Примеч. науч. ред.*

тать эту книгу; но если вы не подключены, эта книга будет для вас как путеводитель для домоседа.

Единственное, что вам нужно, – это компьютер, редактор для создания простых ASCII-текстовых файлов и браузеры. Примеры, приведенные в этой книге, были протестированы на последних версиях браузеров Internet Explorer, Mozilla Firefox, Netscape Navigator и Opera от Opera Software ASA. Поскольку HTML- и XHTML-документы хранятся в доступном для всех текстовом формате и поскольку эти языки аппаратно-независимы², мы не станем делать никаких предположений относительно того, каким компьютером вы пользуетесь. Браузеры, однако, различаются в зависимости от платформы и операционной системы, так что ваши HTML- и XHTML-документы могут выглядеть совершенно по-разному на разных компьютерах и при использовании различных версий браузеров. Мы объясним, как разные браузеры относятся к определенным элементам языка, уделяя особое внимание их различиям.

Если вы новичок в HTML, WWW или вообще в гипертекстовых документах, то вам следует начать с чтения главы 1. В этой главе мы рассказываем, как соединились все технологии WWW при создании сети взаимосвязанных документов.

Если вы уже знакомы с Интернетом, но не знаете ни HTML, ни XHTML, начните с главы 2. В ней дан краткий обзор наиболее важных особенностей языка, и она может служить схемой, демонстрирующей наши подходы к языку в остальной части этой книги.

Последующие главы касаются конкретных аспектов HTML и XHTML. Изложение в них ведется сверху вниз. Читайте их по порядку, чтобы познакомиться с языком в целом, или перескакивайте от раздела к разделу, чтобы найти то, что вас особенно интересует.

¹ ASCII (American Standard Code for Information Interchange) – американский стандартный код для обмена информацией. В данном случае имеется в виду текстовый файл, содержащий только символы в коде ASCII (в отличие, например, от документа Microsoft Word, содержащего кроме текста служебную информацию). Для пользователей Windows одним из таких редакторов (хотя и не самым удобным) является обычный «Блокнот». – *Примеч. науч. ред.*

² Под аппаратной независимостью подразумевается тот факт, что документы HTML и XHTML не требуют никакой доработки кода при их переносе на различные типы вычислительной техники (от интеллектуальных мобильных телефонов до суперкомпьютеров) под управлением практически любых операционных систем. Однако при переносе сохраняется только логическая структура и содержание документа, а визуальное представление (размещение и внешний вид элементов) может изменяться. – *Примеч. науч. ред.*

Как устроена эта книга

На протяжении книги мы используем моноширинные шрифты для выделения собственных элементов стандартов HTML/ХТМЛ, тегов и атрибутов. В тегах мы всегда используем строчные буквы.¹ Мы употребляем *курсив*, когда даем определение новых терминов, а также для элементов, необходимых при создании ваших собственных документов, таких как атрибуты тегов и определяемые пользователем строки.

Мы обсуждаем элементы языка на протяжении всей книги, но для каждого из подробно (может быть, даже слишком дотошно) изучаемых элементов можно найти краткое описание в справочном окне, которое выглядит примерно, как приведенное ниже.

<title>	
Функция:	Определяет заголовок документа
Атрибуты:	dirlang
Закрывающий тег:	</title>; присутствует обязательно
Содержит:	<i>plain_text</i>
Может содержаться в:	<i>head_content</i>

Первая строка в окне содержит название элемента, за которым следует краткое описание его функции. Затем перечислены атрибуты, если они имеются, которые можно или нужно указать в качестве составной части элемента.

Значок  идентифицирует теги и атрибуты, не представленные в стандартах HTML 4.01 и ХТМЛ 1.0, а также те, которые по-разному интерпретируются разными браузерами.

Краткое описание содержит также закрывающий HTML-тег элемента (если он требуется) вместе с указанием, может или нет закрывающий тег быть безопасно опущен при обычном использовании HTML. Для нескольких тегов, которые в HTML не имеют закрывающего тега, но имеют его в ХТМЛ, язык позволяет обозначить это завершение с помощью символа «прямой слэш» (/) в конце тега, например
. В этих случаях тег может также содержать атрибуты, обозначенные многоточием, например <br.../>.

В разделе «Содержит» перечисляются правила грамматики HTML, определяющие элементы, которые могут быть заключены в описываемом теге. Соответственно раздел «Содержится в» перечисляет те пра-

¹ HTML безразличен к регистру в том, что касается имен тегов и атрибутов, но в ХТМЛ строчные и заглавные буквы различаются. Также в некоторых элементах HTML, таких как имена файлов, регистр букв имеет значение, так что будьте осторожны.

вила, которые могут содержать этот тег. Данные правила определены в приложении А.

Наконец, HTML и XHTML в значительной степени пересекаются. В зависимости от контекста вы будете по-разному использовать элементы, но многие из них имеют общие атрибуты. Там, где это возможно, мы помещаем в тексте перекрестные ссылки, отсылающие вас к обсуждению родственных тем в других местах этой книги. Эти ссылки, как та, что находится в конце этого абзаца, являются грубой бумажной моделью гипертекстовой документации, которые могли бы превратиться в настоящие гипертекстовые ссылки, если бы эта книга была представлена в электронном виде. [синтаксис тегов, 3.3.1]

Советуем пользоваться этими ссылками при всякой возможности. Мы будем часто говорить об атрибутах вкратце, полагая, что вы последуете за ссылкой в поисках более подробных сведений. В других случаях ссылка приведет вас к описанию альтернативных вариантов использования обсуждаемого элемента, к советам по его применению или к описанию его стилистических особенностей.

Версии и семантика

Последний стандарт HTML – это версия 4.01, но большая часть изменений и нововведений языка присутствует и в версии 4.0. Поэтому в данной книге мы в основном упоминаем стандарт HTML как HTML 4, подразумевая версию 4.0 и более поздние. Мы точно указываем номер версии, только когда это существенно.

Стандарт XHTML в настоящее время сделал только первый шаг – версия 1.0. Консорциум W3C (World Wide Web Consortium) выпустил рабочий проект второй версии (XHTML 2.0), но этот стандарт еще только устанавливается. В основном XHTML 1.0 совпадает с HTML 4.01. Мы подробно опишем их различия в главе 16. На протяжении этой книги мы будем специально оговаривать случаи, когда XHTML рассматривает какой-либо аспект или элемент языка иначе, чем HTML.

В стандартах HTML и XHTML четко различаются «типы элементов» документа и «теги», размечающие эти элементы. Например, стандарт говорит об элементе типа «абзац», который вовсе не то же, что тег `<p>`. Элемент типа «абзац» состоит из принятого имени типа элемента в открывающем теге (`<p>`), содержимого элемента (в промежутке между тегами разметки) и тега завершения абзаца (`</p>`). Тег `<p>` – это открывающий тег для элемента типа «абзац», и его содержимое (атрибуты) воздействует на содержимое элемента типа «абзац».

Хотя это важные различия, на практике именно теги разметки применяются авторами в их документах с целью повлиять на содержимое элемента, если это необходимо. Соответственно и мы в этой книге нивелируем понятия «тип элемента» и «тег», говоря чаще всего о тегах

и их содержанием, не обязательно употребляя при этом термин «тип элемента», более подходящий в данном случае. Простите нам эту вольность, но мы допускаем ее для большей доходчивости изложения.

HTML и XHTML

HTML – не латинский язык, но он достиг преклонного возраста уже к версии 4.01. Консорциум W3C не планирует разработку следующей версии, и об этом объявлено официально. Напротив, HTML получил обоснование и внешнюю форму, став частью расширяемого языка разметки (Extensible Markup Language, XML). Его новое имя – XHTML (Extensible Hypertext Markup Language, расширяемый язык разметки гипертекста).

Появление XHTML – лишь еще одна глава в непростой истории HTML и Интернета, где путаница – это скорее норма, чем исключение. Хуже всего было, когда в период «броузерных войн» между Netscape и Microsoft старейшины W3C, ответственные за разработку и принятие приемлемых стандартов языка, потеряли контроль над языком. Стандарт HTML+ оказался не слишком удачным, а обсуждение HTML 3.0 настолько затянулось, что W3C просто положила весь проект на полку. Что бы ни утверждали в своей литературе предприимчивые деятели рынка, HTML 3.0 так и не появился. Вместо этого в конце 1996 года производители броузеров убедили W3C выпустить стандарт HTML версии 3.2, единственной целью которого было «узаконить» (включить в стандарт) большую часть расширений HTML для ведущего броузера (Netscape).

Доминирование Netscape в качестве ведущего броузера и лидера веб-технологий практически сошло на нет в конце XX века. К этому времени корпорация Microsoft эффективно встроила Internet Explorer в операционную систему Windows не только в качестве очередного приложения, но и как главную функциональную возможность графического пользовательского интерфейса. Кроме того, броузер Internet Explorer обладал функциональностью (нестандартной на тот момент), которая была принципиально ориентирована на растущий бизнес и маркетинг в Интернете.

К удовольствию тех из нас, кто ценит и твердо поддерживает стандарты, W3C вернул себе ведущую роль, выпустив стандарт HTML 4.0, который в декабре 1999 года сменился на HTML 4.01. Этот стандарт, вобравший в себя многие новшества, предложенные броузерами Netscape и Internet Explorer, более прост и понятен, чем все предыдущие. Он устанавливает четкие модели отображения документов, обеспечивающие переносимость кода между броузерами и платформами, хорошо интегрирован с дружественным стандартом каскадных таблиц стилей (Cascading Style Sheets, CSS). Стандарт CSS предназначен для задания параметров оформления содержимого HTML-документов и дает сред-

ства для альтернативных (невизуальных) клиентских программ, а также для поддержки различных универсальных языков.

В процессе разработки стандарта W3C осознал, что HTML никогда не сможет удовлетворить потребности сетевого сообщества в многообразии способов распространения, обработки и отображения документов. HTML предлагает лишь ограниченный набор примитивов для конструирования документов и решительно не способен справиться с таким нетрадиционным материалом, как химические формулы, музыкальная нотация или математические выражения. К тому же он не способен удовлетворительно поддерживать альтернативные средства отображения, такие как карманные компьютеры или интеллектуальные мобильные телефоны.

Отвечая на эти запросы, W3C разработал стандарт расширяемого языка разметки XML. В стандарте XML описываются принципы построения собственных языков разметки.¹ Причем для реализации таких языков уже не требуется никаких дополнительных действий со стороны W3C. XML-совместимые языки несут информацию, которая в дальнейшем может быть проанализирована, обработана, отображена (и что там еще делают с информацией) посредством разнообразных технологий, появившихся в результате революции в средствах цифровой связи, перевернувшей мир десять лет назад с появлением Интернета. XHTML – это HTML, переработанный с тем, чтобы отвечать стандарту XML. Это язык, на котором основано будущее Сети.

Почему бы просто не отказаться от HTML в пользу XHTML? По многим соображениям. Первое и главное – стандарт XHTML не взял Всемирную паутину штурмом. В создание основанной на HTML документации и в обучение языку вложено уже столько сил и средств, что вряд ли это может произойти в скором времени. Кроме того, XHTML – это переработанный с учетом XML язык HTML 4.01. Выучите HTML 4 – и вы готовы к будущему.²

Функциональные возможности, объявленные нежелательными

Одной из непопулярных мер, которую вынуждены предпринимать разработчики стандартов, является отказ от элементов, уже получивших широкое распространение, в пользу элементов, которые разработчики

¹ XML, в отличие от HTML, не определяет разметку для какого-либо конкретного применения (в данном случае для формирования WWW-документов), а определяет формальные правила разметки для *любых* целей: в этом и его сила, и его слабость. XHTML, конкретизирующий XML применительно к сфере WWW-документов, и восполняет избыточную общность XML. – *Примеч. науч. ред.*

² Мы погрузимся в глубины XML и XHTML в главе 15 «XML» и главе 16 «XHTML».

считают более совершенными. Авторы стандартов HTML и XHTML реализуют эту меру, объявляя «нежелательными»¹ те функциональные возможности, которые не вписываются в генеральную схему развития языка.

Например, тег `<center>` предписывает браузеру выводить обозначенный им текст в центре окна. В то же время стандарт CSS предоставляет свои средства для отцентровки текста. Консорциум W3C сделал выбор в пользу поддержки CSS и не рекомендует веб-авторам использовать тег `<center>`, объявив его нежелательным. Возможно, в одной из следующих версий стандарта тег `<center>` и другие нежелательные элементы и атрибуты будут удалены из языка.

На протяжении всей книги мы специально отмечаем нежелательные HTML-теги и другие элементы, а также постоянно напоминаем об их «нежелательности». Должны ли вы перестать пользоваться ими? И да, и нет.

Да, потому что существует более предпочтительный и, вероятно, лучший способ добиться того же эффекта. Применяя его, вы обеспечиваете долгую жизнь своих документов во Всемирной паутине. Кроме того, инструменты, которыми вы будете подготавливать HTML/XHTML-документы, будут, скорее всего, придерживаться рекомендуемых стандартов. У вас просто не будет выбора, если вы сознательно не откажетесь от этих инструментов. В любом случае, если вы не создаете все документы вручную, вы должны понимать, как работает предпочтительный способ, хотя бы для того, чтобы уметь найти его в сгенерированном коде и исправить.

И все-таки, какими бы убедительными ни были аргументы в пользу отказа от нежелательных элементов и атрибутов, они все еще являются стандартными. Они хорошо поддерживаются большинством браузеров, и их исчезновение в ближайшем будущем остается под вопросом. А поскольку стандарт HTML изменяться не будет, навешивание ярлыков «нежелательности» вводит в заблуждение.

Так что вам не следует излишне беспокоиться о нежелательных функциональных возможностях HTML, и уж тем более нет повода для паники. Просто мы рекомендуем вам сделать шаг вперед по направлению к новым стандартам.

¹ Дословно deprecated – нежелательный, устаревший, неподдерживаемый; практика постепенного «перевода» терминов и понятий в категорию deprecated сложилась и в других языковых технологиях, например (раньше, чем в других) в стандартах C/C++ компилятора GCC – «открытого» проекта GNU, и оказалась в высшей степени плодотворной. Состоит она в постепенном (от версии к версии) выведении из обращения терминов и понятий, не нарушающем цельности языка и не порождающем несовместимости версий. – *Примеч. науч. ред.*

Подробное руководство

Парадокс заключается в том, что стандарт HTML 4.01 не является полным и окончательным. Значительная часть возможностей, включенных в последний стандарт HTML, широко используется и поддерживается популярными браузерами. С другой стороны, многие части стандарта игнорируются. Уверяем вас, что вы можете окончательно запутаться.

Мы уже разложили все по полочкам, так что вам не придется потеть, выясняя, что в каком-то браузере работает, а что нет. Таким образом, эта книга является полным и окончательным руководством по HTML и XHTML. Мы сообщаем подробности обо всех элементах стандартов HTML 4.01 и XHTML 1.0, а также о множестве интересных и полезных расширений языка. Мы также включили в книгу подробное обсуждение стандарта CSS, поскольку он неразрывно связан с разработкой веб-страниц.

Кроме того, есть технологии, тесно связанные с HTML, но не являющиеся его частью. Так, например, мы лишь вскользь упоминаем о CGI¹ и Java. CGI- и Java-программы действуют в тесной связи с HTML-документами и работают под управлением браузеров или параллельно с ними, но все же не являются частью языка как такового, так что мы в них не углубляемся. Кроме того, эти темы столь обширны, что заслуживают отдельных книг, таких как David Flanagan «JavaScript: The Definitive Guide», Scott Guelich, Shishir Gundavaram, Gunther Birznieks «CGI Programming with Perl»², Eric Meyer «Cascading Style Sheets: The Definitive Guide» и Pat Niemeyer и Jonathan Knudsen «Learning Java», выпущенных издательством O'Reilly.

Эта книга – полное руководство по HTML и XHTML, рассказывающее, что они собой представляют и как ими следует пользоваться, включая все расширения, о которых нам известно. Некоторые расширения нигде не документированы, даже в многочисленных руководствах, встречающихся в Интернете. Если мы все-таки пропустили что-нибудь, дайте нам знать, и мы заполним пробел в следующем издании.

Использование фрагментов кода

Эта книга предназначена для оказания вам помощи в работе. Вообще говоря, вы можете использовать код, приведенный в этой книге, в своих программах и документах. Вы не обязаны получать наше разрешение

¹ CGI (Common Gateway Interface) – общий шлюзовой интерфейс. В данном случае подразумеваются программы и сценарии, использующие принципы CGI. – *Примеч. науч. ред.*

² С. Гулич, Ш. Гундаварама, Г. Бирзнекс «CGI программирование на Perl», издательство «Символ-Плюс», 2001.

за исключением тех случаев, когда воспроизводите значительные фрагменты кода. Например, для написания программы, в состав которой входят отрывки кода, приведенные в этой книге, разрешение не требуется. Продажа или распространение CD-ROM с примерами из книг O'Reilly *требует* разрешения. Когда вы отвечаете на чей-то вопрос, цитируя эту книгу и фрагменты кода из нее, вы не нуждаетесь в разрешении. Если вы включаете значительный объем кода из этой книги в документацию к своему продукту, вы *должны* получить разрешение.

Мы не требуем обязательных ссылок на книгу при цитировании, но будем благодарны, если вы их приведете. Ссылка обычно содержит название книги, автора, издательство и ISBN. Например, «HTML & XHTML: The Definitive Guide, Sixth Edition, by Chuck Musciano and Bill Kennedy. Copyright 2007 O'Reilly Media, Inc., 978-0-596-52732-7».

Safari® Enabled



Если на обложке технической книги есть пиктограмма «Safari® Enabled», это означает, что книга доступна в Сети через O'Reilly Network Safari Bookshelf.

Safari предлагает намного лучшее решение, чем электронные книги. Это виртуальная библиотека, позволяющая без труда находить тысячи лучших технических книг, вырезать и вставлять примеры кода, загружать главы и находить быстрые ответы, когда требуется наиболее верная и свежая информация. Она свободно доступна по адресу <http://safari.oreilly.com>.

Обратная связь

Вопросы и пожелания относительно этой книги направляйте по адресу:

O'Reilly Media, Inc.
1005 Gravenstein Highway North
Sebastopol, CA 95472
800-998-9938 (in the United States or Canada)
707-829-0515 (international/local)
707-829-0104 (fax)

На сайте O'Reilly поддерживается веб-страница, содержащая списки подтвержденных и неподтвержденных ошибок, примеры и дополнительную информацию:

<http://www.oreilly.com/catalog/html6>

Комментарии и вопросы по сути материала направляйте электронной почтой по адресу:

bookquestions@oreilly.com

Дополнительные сведения о книгах, конференциях, информационных центрах и сети информационных сайтов O'Reilly Network можно найти на сайте:

<http://www.oreilly.com>

Благодарности

Мы бы не написали и не могли бы написать ни эту книгу, ни любое из предыдущих изданий без великодушной помощи многих людей. Наши жены, Джоанна и Синди, наши дети, Эва, Этан, Кортни и Коул (они появились *раньше*, чем мы начали писать эту книгу) обеспечили первую линию поддержки. Множество друзей, соседей и коллег помогли нам, делясь идеями, тестируя браузеры и предоставляя свое оборудование для исследования HTML. Вы знаете, что мы говорим о вас, и мы вас всех благодарим.

Также мы благодарим наших технических редакторов Чата Классмана (Chat Clussman), Патрика Крекелберга (Patrick Krekelberg), Сэма Маршалла (Sam Marshall) и Шломи Фиш (Shlomi Fish) за тщательное изучение нашей работы. Мы приняли большинство ваших остроумных замечаний. Мы благодарим наших редакторов Майка Лукидеса (Mike Loukides) и Татьяну Апанди (Tatiana Arandi) за терпение. И особая благодарность Татьяне за доведение этого, шестого, издания до логического завершения.

- На чем писать
- Первый HTML-документ
- Встраиваемые теги
- Костяк HTML
- Плоть HTML- и XHTML-документов
- Текст
- Гиперссылки
- Изображения – это особая статья
- Списки, строка поиска, формы
- Таблицы
- Фреймы
- Таблицы стилей и JavaScript
- Вперед!

2

Быстрый старт

Мы не изучали часами какие-либо справочники, перед тем как написать наш первый HTML-документ. Вы, вероятно, тоже не станете. HTML легко читать и понимать, да и писать на нем не трудно. А ваш первый опус, созданный в HTML, будет почти готовым XHTML-документом. Так что приступим, не тратя времени на изучение множества трудных правил.

Чтобы помочь вам быстро и с удовольствием сделать первый шаг, мы выполнили эту главу в качестве краткой сводки по элементам HTML и его потомка, XHTML. Разумеется, было опущено много деталей и ряд уловок, которые следует знать. Читайте следующие главы для полного овладения HTML и XHTML.

Даже если эти языки вам не в новинку, мы советуем проработать эту главу, прежде чем переходить к оставшейся части книги. Она позволит не только усвоить основы HTML/XHTML и их жаргон, но и сделает дальнейшую работу эффективной, ободрив вас успехами в создании таких привлекательных документов за столь короткое время.

2.1. На чем писать

Для создания HTML- или XHTML-документов используйте любой редактор, способный сохранить на диске текстовый файл. Даже если документы имеют сложную планировку и включают в себя изображения, они тем не менее остаются простыми текстовыми файлами. Продвинутый WYSIWYG-редактор или конвертер, входящий в комплект вашего любимого текстового процессора, тоже годится, хотя они могут не поддерживать все возможности, которые будут обсуждаться в этой книге. Вам все равно придется внести последние штрихи в выданный ими

текст; к тому же не стоит надеяться, что компоновка страницы будет такой же, как при использовании специализированного приложения.

Вам также понадобится установить на компьютере популярный браузер, который хотя и не нужен для создания документов, но необходим, чтобы видеть результаты работы. Причина проста: исходный текст, составленный с помощью текстового редактора, выглядит совсем непохоже на то, что покажет браузер, хотя это один и тот же документ. Убедитесь, что ваши читатели действительно увидят то, что вы хотели им показать, просмотрев документ до них с помощью браузера. Кстати, популярные браузеры можно скачать бесплатно. На сегодняшний день мы можем рекомендовать Microsoft Internet Explorer, Mozilla Firefox, Apple Safari, Netscape Navigator и Opera.

Заметьте также, что нет необходимости подключаться к Интернету, чтобы писать или просматривать HTML- или XHTML-документы, – их можно сохранять на жестком диске или на дискете. Можно даже путешествовать по локальным документам, применяя гиперссылки, при этом не имея никакой связи с Интернетом или какой-либо другой сетью. На наш взгляд, следует работать локально, проектируя и тщательно тестируя документы, прежде чем делиться ими с окружающими.

Тем не менее мы настоятельно рекомендуем *обязательно иметь* выход в Интернет, если вы всерьез настроены создавать собственные документы. Можно загружать и просматривать чьи-то интересные страницы и выяснять, как на них применяются интересные возможности языка – хорошо или плохо. К тому же учиться на примерах нескучно. (С другой стороны, использование чужих находок часто проблематично, если даже не противозаконно.) Подключение к Интернету существенно, если в своей работе вы обращаетесь к ссылкам на другие документы в глобальной сети.

2.2. Первый HTML-документ

Похоже, что всякий когда-либо написанный учебник по языкам программирования начинается с простого примера: как отобразить сообщение «Hello, World!». Что же, в этой книге никакого «Hello, World!» вы не увидите. В конце концов, перед вами руководство по стилю, предназначенное для нового тысячелетия. Вместо этого мы пошлем приветствие World Wide Web:

```
<html>
<head>
<title>My first HTML document</title>
</head>
<body>
<h2>My first HTML document</h2>
Hello, <i>World Wide Web!</i>
  <!-- Обойдемся без "Hello, World" -->
<p>
```

```
        Greetings from<br>
<a href="http://www.ora.com">O'Reilly</a>
<p>
Composed with care by:
<cite>(insert your name here)</cite>
<br>&copy;2000 and beyond
</body>
</html>
```

Пойдем дальше. Наберите исходный текст примера на новой странице редактора и сохраните его на локальном диске как *myfirst.html*. Убедитесь, что он записан в простом текстовом формате. Другие форматы, такие как *.doc*, присущий Microsoft Word, добавляют к тексту специальные (непечатаемые) символы, которые могут запутать браузер и нарушить вывод HTML-документа.

Сохранив на диске файл *myfirst.html* (или *myfirst.htm*, если вы пользуетесь архаичным стандартом¹ образования имен файлов, поддерживаемым DOS или Windows 3.11), запустите браузер, обратитесь к его меню и с помощью команды File (Файл) найдите и откройте документ. Ваш экран должен выглядеть примерно так (рис. 2.1). Хотя некоторые элементы интерфейса, например меню и панели инструментов, могут выглядеть по-разному в разных браузерах, содержимое окна будет таким же.



Рис. 2.1. Очень простой HTML-документ

¹ Здесь подразумевается устаревшее так называемое «правило 8.3», по которому обозначение файла должно состоять из двух частей: имени, содержащем не более восьми символов, и расширения, включающего точку и следующие за ней от 1 до 3 символов. — *Примеч. науч. ред.*

2.3. Встраиваемые теги

Вы, наверное, уже заметили, и возможно с удивлением, что браузер отображает меньше половины исходного текста примера. Ближайшее рассмотрение источника показывает: пропало все, что заключено в скобки из символов «меньше» (<) и «больше» (>). [синтаксис тегов, 3.3.1]

HTML и XHTML – это встраиваемые языки: их директивы, или *теги*, вставляются в тот самый документ, который авторы и читатели загружают в браузер для просмотра. Браузер использует информацию, находящуюся внутри этих тегов, чтобы решить, как отобразить (или иначе обработать) следующее за тегом содержимое документа.

В частности, тег <i>, стоящий в примере за словом «Hello», предписывает браузеру отобразить следующий за ним текст курсивом.¹ [теги физической разметки, 4.5]

Первое слово в теге – это его официальное имя, которое обычно описывает присущие ему функции. Любые добавочные слова в теге являются специальными *атрибутами*, иногда с присвоенным им значением, стоящим после знака равенства (=), которые определяют или модифицируют действие тега.

2.3.1. Открывающий и закрывающий теги

Большая часть тегов определяет и оказывает влияние на ограниченную область документа. Эта область начинается там, где тег со своими атрибутами появляется впервые (его называют иногда *открывающим тегом*), и продолжается до соответствующего *закрывающего тега*, в котором его имени предшествует прямой слэш (/). Например, тегу «начало курсива» <i> соответствует закрывающий тег </i>.

В закрывающем теге не бывает атрибутов. В HTML большинство тегов, хотя и не все, имеют закрывающий тег. Но чтобы сделать жизнь пишущих на HTML легче, синтаксический анализатор браузера часто по контексту может однозначно определить место, где должен стоять закрывающий тег, поэтому в ряде случаев его присутствие в исходном HTML-тексте не обязательно. В последующих главах мы скажем, какие из закрывающих тегов можно опускать, а какие – нет. В нашем простом примере нет закрывающего тега </p>, местоположение которого обычно легко определяется автоматически, и, следовательно, он

¹ Отображение текста курсивом – это очень простой пример, поддерживаемый большинством браузеров, за исключением разве что текстовых, таких как Lynx. Как правило, браузер пытается сделать то, что ему сказано, но его настройки, как будет показано в следующих главах, могут отличаться в зависимости от компьютера, операционной системы и желания пользователя. Поэтому набор допустимых или отобранных шрифтов может быть непредсказуем. Имейте в виду, что не все могут или хотят отображать ваш HTML-документ в точности так, как вы видите его на своем экране.

так часто не включается в исходный текст, что один опытный HTML-разработчик даже не знал, что этот тег существует. Каково?

Стандарт XHTML гораздо более строг и настаивает на том, чтобы все теги имели соответствующие закрывающие. [закрывающие теги, 16.3.2] [пустые элементы, 16.3.3]

2.4. Костяк HTML

Обратите внимание на то, что в простом примере исходного текста HTML-документ начинается и заканчивается тегами `<html>` и соответственно `</html>`. Они сообщают браузеру, что весь документ написан на HTML.¹ Стандарты HTML и XHTML требуют присутствия тега `<html>`, но большинство браузеров способны определить и адекватно отобразить HTML-разметку текстового документа, в котором не хватает этого самого внешнего тега. [`<html>`, 3.6.1]

Как и в примере, все HTML- и XHTML-документы (за исключением специальных фреймовых документов) содержат две главные структуры: *заголовок* (head) и *тело* (body), заключенные в тексте каждый в соответствующую пару открывающего и закрывающего тегов. Информация о документе помещается в заголовок, а содержание, которое должно быть отображено в окне браузера, – в тело. За исключением редких случаев большую часть времени занимает работа над содержимым тела документа. [тег `<html>`, 3.6.1] [тег `<body>`, 3.8.1]

Есть несколько различных тегов заголовков, которые можно использовать, чтобы задать стиль отображения данного документа, подходящий для какой-либо подборки или большой схемы, размещенной в сети. Кроме того, существуют нестандартные теги заголовка, позволяющие вносить в документ элементы анимации.

Однако в большинстве случаев важнейшим заголовочным элементом является название документа. Стандарты требуют, чтобы каждый HTML- или XHTML-документ имел название, несмотря на то, что популярные в настоящее время браузеры не настаивают на этом правиле. Выберите осмысленное имя, которое тотчас скажет читателю, о чем документ. Закрывайте это название, как в примере, в теги `<title>` и `</title>` в заголовке документа. Популярные браузеры обычно отображают название в верхней части окна документа. [тег `<title>`, 3.7.2]

2.5. Плоть HTML- и XHTML-документов

Помимо тегов `<html>`, `<head>`, `<body>` и `<title>` стандарты HTML и XHTML включают еще несколько необходимых структурных элементов. Вы мо-

¹ XHTML-документы тоже начинаются с тега `<html>`, но с дополнительной информацией, чтобы можно было отличить их от обычных HTML-документов. Подробное описание см. в главе 16 «XHTML».

жете свободно вставить в содержимое документа несметное количество разного материала. (Веб-серферы, в числе которых состоят и авторы книги, власть попользовались этой свободой.) Наверное, удивительно, но существует всего три главных типа содержимого HTML/ХHTML-документов – теги (о которых уже шла речь), комментарии и текст.

2.5.1. Комментарии

Черновой документ со всеми вставленными в него тегами может быстро стать таким же неудобочитаемым, как исходные коды программ. Мы настоятельно рекомендуем использовать комментарии, чтобы самим себе помочь разобраться в создаваемых материалах.

Хотя комментарии и являются частью документа, ничего из их содержимого, находящегося между специальным открывающим тегом `<!--` и закрывающим `-->`, не отображается браузером. Посмотрите на комментарии в исходном тексте примера и убедитесь, взглянув на рис. 2.1, что браузер их не отобразил. Всякий тем не менее может скачать исходный текст вашего документа и прочитать комментарии, поэтому следите за тем, что вы пишете.

2.5.2. Текст

Если это не тег и не комментарий, значит, это текст. Основной объем содержимого HTML/ХHTML-документа (то, что читатели видят на экране) занимает текст. Специальные теги задают структуру текста – заголовки, списки, таблицы. Остальные теги указывают браузеру, как содержимое документа должно быть отформатировано и выведено на экран.

2.5.3. Мультимедийные элементы

А что же с изображениями и другими мультимедийными элементами, которые мы видим и слышим при воспроизведении страницы нашим браузером? Разве они не являются частью HTML-документа? Нет. Данные, содержащие цифровые изображения, видео, аудио и прочие мультимедийные элементы, которые может показывать и проигрывать браузер, хранятся отдельно от основного HTML/ХHTML-документа. В HTML-документ включаются ссылки на них при помощи специальных тегов. Браузер использует эти ссылки, чтобы загрузить и интегрировать документы других типов с вашим текстом.

Мы не включили никаких мультимедийных ссылок в предыдущий пример, потому что они являются отдельными нетекстовыми документами, которые вы не могли бы просто набрать в своем текстовом редакторе. В главе еще пойдет разговор и будет показано, как можно включить изображения и другие мультимедийные элементы в текст. В последующих главах этот вопрос обсуждается весьма детально.

2.6. Текст

Относящихся к тексту тегов разметки в стандартах HTML/ХHTML-языков больше всего. Дело в том, что изначально язык HTML появился как средство обогащения структуры и организации текста.

HTML вышел из академической среды. Для первых разработчиков языка было важно, да и сейчас остается важным, чтобы их, главным образом, научные материалы (как правило, текстовые) быстро просматривались и читались, не теряя при этом возможность распространяться в Интернете на разнообразные компьютерные платформы. (Unicode-текст – это единственный универсальный формат¹ в глобальном масштабе Интернета.) Возможность включения мультимедиа – только довесок к HTML и ХHTML, хотя и немаловажный.

Макет страницы тоже вторичен по отношению к структуре. Люди просматривают документ визуально и оценивают построение и отношения между частями текста, основываясь на его внешнем виде, тогда как машины могут только читать закодированную разметку. Поскольку теги, содержащиеся в тексте, соотнесены со смыслом документов, они хорошо подходят для компьютерного поиска, а также для автоматизированной обработки содержимого. Все это относится к средствам, очень важным для исследователей. Не так существенно то, *как* сказано, сколько то, *что* сказано.

Таким образом, ни HTML, ни ХHTML не являются языками, описывающими расположение текста на странице. Фактически при существующем разнообразии настраиваемых пользователями браузеров и не меньшем разнообразии платформ для отображения электронных документов все, к чему можно стремиться, – это *советовать*, но не предписывать, как должен выглядеть на экране представляемый материал. Невозможно заставить браузер показать документ каким-либо определенным образом. Вы повредите в уме, если будете настаивать на противном.

2.6.1. Внешний вид текста

В частности, нельзя предсказать, какой шрифт и какого абсолютного размера – 8 или 40 пунктов, Helvetica, Geneva, Subway или еще какой-то – будет применять конкретный пользователь. Постойте, но новейшие браузеры теперь поддерживают стандарт каскадных таблиц стилей и другие средства настольно-издательского типа, позволяющие

¹ Unicode – это, точнее, не формат, а универсальный стандарт кодирования символов, не зависящий от языка и кодировки; а вот формат представления Unicode может быть различный, например: UTF-16, UTF-8. В наши дни большие надежды возлагаются на применение формата Unicode. Многие современные операционные системы, программные пакеты, в частности браузеры, работают с кодировкой Unicode. – *Примеч. науч. ред.*

управлять компоновкой и внешним видом документов! Да, но пользователь может перенастроить характеристики своего браузера, отвечающие за отображение, и своевольно разрушить ваши старательно разработанные планы. Некоторые старые браузеры не поддерживают новых средств компоновки, а другие работают только с текстом без каких-либо красивых шрифтов. Что же делать? Сосредоточиться на содержании. Крутые странички – это пена на поверхности волн. Глубокое содержание – вот что заставит людей возвращаться раз за разом к вашему документу.

Тем не менее стиль облегчает чтение, поэтому там, где получается, его следует применять, если только это не вредит представлению содержания. Можно задать для своего текста обычные атрибуты стилей при помощи тегов *физических стилей*, таких как тег курсива `<i>` в примере. Более значимо и точно отражают идеологию языка теги *логических (контекстно-зависимых) стилей*, которые придают *смысловую нагрузку* различным фрагментам текста. Кроме того, можно изменять характеристики отображения текста, такие как стиль и размер шрифта, цвет и тому подобное, с помощью каскадных таблиц стилей.

Современные графические браузеры распознают теги физических и логических стилей и изменяют внешний вид относящихся к ним фрагментов текста, выражая визуально их значение и структуру. Но нельзя точно предсказать, как эти изменения будут выглядеть.

Стандарт HTML 4 (и тем более стандарт XHTML 1.0) подчеркивает, что в будущем браузеры не будут так тесно привязаны к визуальному отображению. Текстовое содержимое можно слушать или распознавать на ощупь, а не только читать глазами. Контекстная разметка в этих случаях подходит, очевидно, лучше, чем физические стили.

2.6.1.1. Логические стили

Теги логических стилей указывают браузеру, что некий фрагмент HTML/XHTML-текста имеет особое значение или употребляется особым образом. Тег `<cite>` в примере означает, что заключенный в нем текст является какой-то ссылкой¹ – в нашем случае на автора документа. Браузеры обычно, хотя и не всегда, отображают такие ссылки курсивом, в отличие от основного текста. [теги логической разметки, 4.4]

Ясно читателю или нет, что данный текст является ссылкой, но в один прекрасный день кто-нибудь может написать программу, прочесывающую различные документы в поисках тегов `<cite>` и составляющую по заключенному в них тексту специальный список ссылок. Схожие программы уже обшаривают Интернет, составляя списки обнаруживаемой информации, как это делает поисковый сервер Google.

¹ Тегом `<cite>` можно обозначить как ссылку, так и цитату. – *Примеч. науч. ред.*

К числу самых часто употребляемых логических стилей в настоящее время относится тег логического выделения, обозначаемый ``. Ну а если вы настроены особенно решительно, то можете употребить тег ``. Другие контекстно-зависимые стили включают в себя `<code>` – для обозначения фрагментов программного кода, `<kbd>` – для обозначения текста, вводимого пользователем с клавиатуры, `<samp>` – для выделения примеров, `<dfn>` – для определений, `<var>` – для имен переменных в программном коде. У всех этих тегов есть соответствующие закрывающие теги.

2.6.1.2. Физические стили

Даже самые простые редакторы поддерживают несколько традиционных стилей текста, таких как курсив и полужирный шрифт. Не являясь инструментами обработки текста в традиционном смысле, HTML и XHTML предоставляют теги, явно предписывающие браузеру отобразить (если он сумеет) символ, слово или фразу особенным физическим стилем.

Хотя по вышеизложенным соображениям вам следует пользоваться соответствующими логическими стилями, иногда форма важнее содержания. Тогда применяйте тег `<i>`, чтобы выделить текст курсивом, не приписывая ему какого-либо определенного смысла, тег `` – для выделения текста полужирным шрифтом, а тег `<tt>` применяйте, чтобы браузер, если сможет, отобразил текст моноширинным шрифтом. [теги физической разметки, 4.5]

Легко попасть в ловушку использования физических стилей, когда надо было бы применять логические. Стоит выбрать «контекстно-зависимую» манеру изложения, поскольку, как уже было сказано, она несет в себе смысл вместе со стилем, приспосабливая тем самым документ к автоматизации и сопровождению.

2.6.1.3. Специальные символы

Не все символы, отображаемые браузером, можно набрать с клавиатуры. К тому же, некоторые из них применяются как управляющие, например угловые скобки, в которые заключаются теги. Если их специально не выделить, то примененные в основном тексте, например знак «меньше» (`<`) в математическом неравенстве, они запутают браузер и превратят документ в электронную макулатуру. HTML и XHTML предоставляют способ *символьной замены* для включения с помощью ссылок любого Unicode-символа в произвольное место текста.

Так же как знак «авторского права» (copyright) в нашем примере, код символа начинается с амперсанда (&), за которым идет его условное имя, закрывающееся точкой с запятой. Альтернативный способ записи состоит в употреблении позиционного номера символа в Unicode-кодировке, который ставится вместо имени после знака «решетка» (#). Выводя документ, браузер показывает соответствующий символ, если он присутствует в шрифте, выбранном пользователем. [коды символов, 3.5.2]

По понятным причинам чаще всего кодируются знаки «больше» (>), «меньше» (<) и амперсанд (&). Загляните в приложение F «Коды символов», чтобы узнать, какой из них соответствует записи ¦. Вы будете приятно удивлены !

2.6.2. Элементы структуры

Из приведенного примера это не видно, но обычный символ возврата каретки¹, используемый для разделения абзацев в исходном документе, не имеет смысла в HTML или XHTML, за исключением специальных случаев. Документ, даже набранный в одну строку, все равно выглядел бы в окне браузера так, как на рис. 2.1.²

Вы бы скоро обнаружили, даже не прочитав об этом здесь, что за некоторым исключением браузер игнорирует начальные и конечные пробелы, а иногда и лишние в середине. (Если внимательно рассмотреть исходный текст примера, то видно, что строка «Greetings from» сдвинута вправо с помощью вставленных перед ней пробелов, что не заметно на рис. 2.1.)

2.6.2.1. Раздел, абзац и конец строки

Браузер принимает текст, находящийся в теле документа, и «размещает» его на экране компьютера, не обращая внимания на обычные символы возврата каретки и новой строки, расположенные в исходном тексте. Браузер заполняет каждую строку в своем окне, насколько это возможно: начинает от левого края и, только поместив справа последнее вошедшее слово, переходит на следующую строку. Если изменить размер окна, то текст перетечет, захватывая новое отведенное ему пространство, демонстрируя тем самым гибкость, присущую HTML.

Конечно, читатели возмутились бы, если текст тянулся бы и тянулся одной сплошной цепочкой. Чтобы этого не случилось, HTML и XHTML предоставляют как явные, так и неявные методы управления структурой документа. Самый элементарный и популярный способ состоит в использовании тегов раздела (<div>), абзаца (<p>) и конца строки (
). Все они прерывают поток текста, возобновляя его с новой стро-

¹ В Unix-системах символ возврата каретки (CR) используется в ASCII-файлах для обозначения конца строки. В DOS и Windows для этих же целей применяется комбинация из символов «перевод строки» (LF) и «возврат каретки» (CR). То, что сказано про символ CR, в полной мере относится и к комбинации LF + CR. – *Примеч. науч. ред.*

² Чтобы облегчить чтение HTML/XHTML-документов, мы используем разбиение на строки и организацию отступов так, как делается в языках программирования. Это не обязательно, и нет никаких формальных правил организации исходных HTML/XHTML-файлов. Тем не менее мы настоятельно рекомендуем выработать стиль оформления кода и придерживаться его, чтобы вам и окружающим было легко разбираться в исходных текстах.

ки. Разница в том, что теги `<div>` и `<p>` выделяют как блок фрагмент документа и соответственно текста, содержимое которого можно специальным образом выровнять в окне браузера, выделить стилем или изменить с помощью других средств, применимых к блоку текста.

Когда теги `<div>` и `
` не сопровождаются специальными атрибутами выравнивания, они просто обрывают текущую строку и помещают следующие символы на новую строку. У тега `<p>` вертикальный отступ имеет больший размер, чем у тегов `<div>` и `
`. [тег `<div>`, 4.1.1] [тег `<p>`, 4.1.2] [тег `
`, 4.6.1]

Естественно, стандарт HTML определяет закрывающие теги для тегов абзаца и раздела, в то время как «конец строки» им не обладает.¹ Многие авторы используют в своих документах закрывающий тег абзаца. Браузер обычно может разобраться сам, где один абзац заканчивается и начинается другой.² Вручите себе орден, если вы знали, что существует тег `</p>`.

2.6.2.2. Заголовки

Помимо деления текста на разделы и абзацы, можно формировать документ, применяя озаглавленные разделы. Так же как и на всех страницах этой книги, заголовки не только разбивают текст и именуют его различные разделы, но и несут некий визуальный смысл. При этом заголовки могут оказаться полезными при автоматической обработке документа.

Существует шесть тегов заголовков, от `<h1>` до `<h6>`, с соответствующими закрывающими тегами. Обычно браузер отображает их содержимое шрифтами от очень большого до соответственно очень маленького размера, как правило применяя и полужирное написание. Текст в теге `<h4>` имеет обычно тот же размер, что и основной документ. [теги заголовков, 4.2.1]

Теги заголовков также прерывают поток текста, занимая целые выделенные из него строки, даже если перед заголовком или после него нет явно выписанных тегов абзаца или конца строки.

2.6.2.3. Горизонтальные линейки

Помимо заголовков HTML и XHTML предоставляют горизонтальные линейки, помогающие отделить друг от друга разделы документа.

¹ В XHTML начало и конец тега `
` заключается в одни и те же скобки: `
`. Браузеры склонны ко всепрощению и часто игнорируют все посторонние вещи, как прямой слэш (/) в данном случае, так что стоит привыкнуть представлять такой признак конца.

² Завершающий тег абзаца теперь используют чаще, так как популярные браузеры поддерживают атрибуты выравнивания, применяемые к абзацу.

Когда браузер встречает тег `<hr>`, он прерывает поток текста и в следующей строке рисует через все окно горизонтальную прямую черту. Поток текста возобновляется сразу под линейкой.¹ [тег `<hr>`, 5.1.1]

2.6.2.4. Преформатированный текст

Бывает, что хочется видеть блок текста на экране таким, как он был написан, скажем с отступами и вертикально выровненными буквами, цифрами, и чтобы вид его сохранялся прежним, даже если окно браузера изменяется в размере. На этот случай есть тег `<pre>`. Весь текст вплоть до закрывающего тега `</pre>` представлен на экране точно так, как вы его набрали, включая возвраты каретки, концы строк и начальные, конечные и промежуточные пробелы. Весьма полезный для таблиц и форм, `<pre>`-текст выглядит тем не менее довольно скучно, так как популярные браузеры отображают его моноширинными шрифтами. [тег `<pre>`, 4.6.5]

2.7. Гиперссылки

Если сравнить текст с костяком и плотью HTML или XHTML-документа, то гипертекст будет его сердцем. Гипертекст дает возможность отыскать и отобразить любой документ из произвольной подборки (размещенной где угодно в сети), просто щелкнув кнопкой мыши по связующим словам или фразе (*гиперссылке*), расположенной на текущей странице. Используйте эти интерактивные гиперссылки, чтобы помочь читателям путешествовать в поисках нужной информации по вашим или чужим собраниям документов, которые иначе оставались бы просто ворохом отдельных файлов разных форматов, включая мультимедийные, HTML-, XHTML-, прочие XML- и, наконец, обычные текстовые элементы. Гиперссылки, образно говоря, буквально сводят богатства знаний всего Интернета к кончику указателя мыши.

Чтобы вставить ссылку на какой-либо документ, лежащий в собственной коллекции или на сервере в Тимбукту, нужно знать только уникальный адрес документа и способ, как «бросить» *якорь* в вашу разработку.

2.7.1. URL

Как бы ни было трудно в это поверить, но каждый из миллиардов документов и ресурсов Интернета имеет уникальный адрес, именуемый *универсальным указателем ресурса* (Uniform Resource Locator, URL). URL состоит из имени документа, которому предшествует иерархия имен каталогов, содержащих файл (*путь к документу*), *доменного имени* сервера, где он находится, и названия способа (*протокола*), поддерживающего связь и обмен данными между сервером и браузером:

¹ Подобно тегу `
`, в XHTML тег горизонтальной линейки формально записывается `<hr/>`.

протокол://доменное_имя_сервера /путь_к_документу

Вот несколько примеров URL:

- *http://www.kumquat.com/docs/catalog/price_list.html*
- *price_list.html*
- *../figs/my_photo.png*
- *ftp://ftp.netcom.com/pub/*

Первый пример – это *абсолютный*, или полный, URL. Он содержит все части формата URL: протокол, имя сервера и путь к документу. Хотя абсолютные URL однозначно определены и точны, их употребление может прибавить вам головной боли, если вы перемещаете документы в другой каталог или на новый сервер. К счастью, браузеры позволяют работать с *относительными* URL и автоматически заполняют всякий пропущенный фрагмент соответствующей частью *базового* URL текущего документа. Второй пример дает нам простейший из возможных относительный адрес. Рассматривая его, браузер будет предполагать, что *price_list.html* находится на том же сервере, в том же каталоге, что исходный документ, и применяет тот же сетевой протокол (*http*). Аналогичным образом третий пример является относительным URL, который отсылает браузер наверх и в каталог */figs*, где хранится файл с картинкой.

Хотя внешний вид обманчив, но последний пример URL для FTP является полным. Он указывает прямо на содержимое каталога */pub*. Более того, указание протокола *ftp* в этом примере является обращением к совсем иной программе на сервере, чем та, что работает с протоколом *http* в других примерах.

2.7.2. Якоря

Тег якоря `<a>` – это средство HTML/ХHTML для определения как исходной точки, так и пункта назначения гиперссылки.¹ Чаще всего тег `<a>` используется с атрибутом `href` для создания исходной точки ссылки. Значение атрибута `href` – это URL пункта назначения.

Содержимое тега `<a>`, определяющего источник ссылки, – это слова и/или изображение между ним и его закрывающим тегом ``. Они представляют собой особым образом активизированную браузером часть документа, с помощью которой пользователь выбирает гиперссылки. Это *якорное* содержимое обычно отличается по внешнему виду от своего окружения (подчеркнутый или выделенный цветом текст, изобраа-

¹ Принятые здесь обозначения несколько неудачны: тег якоря должен был бы отмечать только конечный пункт ссылки, но не то место, с которого она «прыгает». Мы «бросаем якорь», но не прыгаем с него. Не стоит даже упоминать о страшно запутанной терминологии, которую W3C использует для обозначения различных частей гиперссылки. Заметим только, что кое-кто находит положение дел здесь чертовски неуклюжим.

жения в рамке особого цвета), кроме того, экранный образ указателя мыши меняется, когда он попадает на якорь. Содержимое тега `<a>`, таким образом, должно быть текстом или изображением (пиктограммам здесь самое место), которое явно или намеком говорит пользователю, куда приведет его гиперссылка. [тег `<a>`, 6.3.1]

Так, в следующем примере браузер специальным образом выводит текст «Кумкват¹-архив» и изменяет изображение попавшего на него указателя мыши:

```
Дополнительную информацию о кумкватах можно узнать, посетив
<a href="http://www.kumquat.com/archive.html">
Кумкват-архив</a>
```

Если пользователь щелкнет кнопкой мыши по этой ссылке, браузер автоматически запросит с сервера *www.kumquat.com* веб-страницу (с помощью протокола *http:*) под названием *archive.html* и затем отобразит ее на экране.

2.7.3. Имена гиперссылок и навигация

Возможность указывать на документ, находящийся на другом конце света, – это не только круто, но и полезно для ваших собственных веб-страниц. Еще одна важная задача гиперссылок – помогать читателям перемещаться по вашим подборкам в поисках необходимой информации. Естественным образом возникает понятие о домашней странице и сопровождающих материалах.

Ни один из документов не должен тянуться бесконечно. Во-первых, это неудобно для пользователя. Ценность вашей работы, как бы насыщена она ни была, страдает, если нужна вечность, чтобы загрузить документ, а затем, с трудом получив, без конца прокручивать его вверх и вниз, чтобы найти нужный раздел.

Лучше проектируйте свою работу в виде собрания нескольких компактных, сжатых страниц, подобных главам этой книги, каждая из которых посвящена определенной теме так, чтобы ее легко мог выбрать и просмотреть пользователь. Затем применяйте гиперссылки для организации вашей коллекции.

К примеру, используйте домашнюю страницу – ведущий документ подборки – как оглавление с краткими описаниями и соответствующими гиперссылками на остальные части коллекции.

Чтобы особым образом поименовать все разделы документа, вы можете применять тег `<a>` с атрибутами `name` или `id` (последний годится практически для всех тегов). Идентификаторы разделов будут слу-

¹ Кумкват (кинкан, японский апельсин) – разновидность миниатюрных цитрусовых. Кроме того, так же называется направление в современной музыке и дизайне. – *Примеч. науч. ред.*

жить целями внутренних ссылок в коллекции и помогут пользователю ориентироваться в одном документе или перепрыгивать к определенному разделу другого. Ссылайтесь на поименованные разделы, приписывая знак «решетка» (#) и имя раздела после URL документа.

Чтобы сослаться на определенную тему в архиве, скажем, «Рецепты тушеных кумкватов» в кумкват-архиве нашего примера, пометим сначала имя раздела с помощью id:

```
... предыдущее содержимое...  
<h3 id="Stews">Рецепты тушеных кумкватов</h3>
```

Затем в этом же или другом документе приготовим исходную гиперссылку, которая указывает прямо на данные рецепты, так как включает название раздела в виде суффикса к URL, отделенного от него знаком «решетка»:

```
Дополнительную информацию о кумкватах можно узнать, посетив  
<a href="http://www.kumquat.com/archive.html">  
  Кумкват-архив</a>,  
и, возможно, попробовав наши  
<a href="http://www.kumquat.com/archive.html#Stews">  
  Рецепты тушеных кумкватов</a>.
```

Последняя гиперссылка, выбранная пользователем, побудит браузер загрузить документ *archive.html* и начать его отображение с раздела «Stews».¹

2.7.4. Что не под силу якорю

Употребление гиперссылок не ограничено только связью с HTML- и XHTML-документами. Якоря позволяют указывать на практически любые материалы, доступные в глобальной сети, включая другие сервисы Интернета.

Однако «позволить» и «сделать доступным» – это не одно и то же. Браузеры могут работать с различными сервисами, подобными FTP и Gopher, так что удается загружать не только HTML-документы. Но пока они не вполне справляются и не слишком ладят с мультимедийными данными.

Сейчас есть несколько стандартов для многих типов и форматов мультимедийных данных. Компьютерные системы, подключенные к сети, чрезвычайно различаются своими возможностями воспроизведения аудио- и видеоформатов. За исключением некоторых графических изображений, стандарт HTML/XHTML не предлагает никаких специаль-

¹ Здесь подразумевается, что запрошенный документ в окне браузера будет прокручен таким образом, что начало раздела, отмеченное именованным тегом, будет располагаться (если это возможно) вверху видимой части окна. – *Примеч. науч. ред.*

ных средств для представления мультимедийных документов помимо возможности ссылаться на них. Броузер, получивший мультимедийный документ, должен запустить *вспомогательное приложение*, загрузить и исполнить *апплет* или иметь дополнительно устанавливаемый *плагин* (встраиваемый модуль), позволяющий декодировать и показать такой материал прямо внутри демонстрационного окна.

Хотя сегодня HTML и большая часть браузеров избегают подобной неразберихи, просто обходя ее стороной, это не значит, что невозможно или запрещено использовать мультимедийные данные в своих документах. Помните только об имеющихся ограничениях.

2.8. Изображения – это особая статья

Файлы изображений – это мультимедийные элементы, на которые можно ссылаться с помощью якорей в документе для их загрузки и отображения посредством браузера. В отличие от других мультимедийных данных, стандарты HTML и XHTML имеют явные средства, обеспечивающие вывод картинок «в строку» с текстом, при этом на картинках можно создавать сложные карты гиперссылок. Это связано с тем, что существует корпоративный стандарт, касающийся применения только определенных форматов файлов изображений, а именно форматов GIF, PNG и JPEG, и графические браузеры имеют встроенные декодеры, которые включают изображения этих типов в документы.¹

2.8.1. Встроенные изображения

HTML/XHTML-тег для встроенных изображений – это ``; его обязательным атрибутом `src` является URL файла, который нужно вставить в документ. [тег ``, 5.2.6]

Броузер загружает изображения и вставляет их в поток текста, представляя как бы специальными, возможно очень большими, символами. Обычно браузер выравнивает изображения и текущую строку по нижней границе. Такое расположение можно изменить с помощью специального атрибута `align` тега ``, значение которого вы устанавливаете равным `top`, `middle` или `bottom`, чтобы изображение позиционировалось по верхней границе, середине или нижней границе примыкающего текста. Проверьте, какое выравнивание вам больше нравится (рис. 2.2–2.4).

¹ Некоторые браузеры поддерживают вывод в виде внедренного объекта не только GIF и JPEG, но и других мультимедийных файлов. Internet Explorer, например, поддерживает тег, который проигрывает аудио в фоновом режиме. Кроме того, стандарты HTML 4 и XHTML предоставляют способ отображения других мультимедийных данных, встраивая их в текст документа с помощью общего для них тега.

Конечно, широкая картинка может занять всю строку и таким образом прервать поток текста. Впрочем, вы можете разместить изображение отдельно, включив в документ предшествующие и последующие теги раздела, абзаца или конца строки.

Опытные HTML-авторы не только используют рисунки как иллюстрации, но также добавляют маленькие встроенные символы или глифы (glyphs), чтобы помочь читателю при просмотре и подчеркнуть разделы документа, имеющие особое значение. Ветераны HTML¹ также применяют изображения в качестве маркеров элементов списка и более четких разделителей, чем стандартные горизонтальные линейки. Изображения также могут быть включены в гиперссылку, так что пользователи смогут выбирать вставленную в строку маленькую пиктограмму, чтобы загрузить полноэкранное изображение. Нет числа способам применения встроенных изображений.

Мы также должны упомянуть атрибут `<alt>`. Укажите для него какое-нибудь текстовое значение, чтобы описать изображение тем, кто отключил показ картинок, или чтобы браузеры, способные озвучивать текст, могли прочесть его незрячему пользователю.



Рис. 2.2. Встроенное изображение, выровненное по нижнему краю (bottom) текста (по умолчанию)

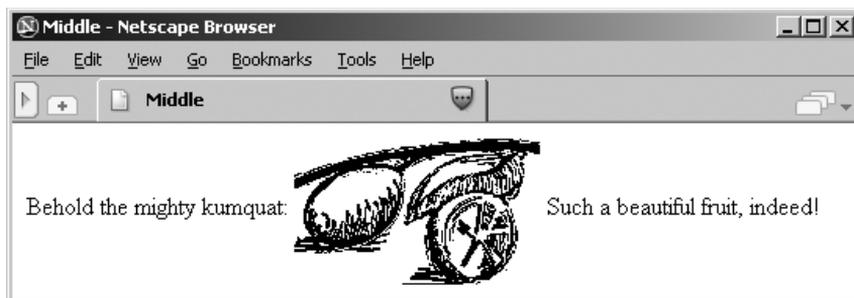


Рис. 2.3. Встроенное изображение, выровненное по центру (middle) текста

¹ XHTML слишком молод, чтобы называть кого-либо ветераном или опытным XHTML-автором.



Рис. 2.4. Встроенное изображение, выровненное по верхнему краю (*top*) текста

2.8.2. Карты

Карты¹ – это изображения в теге якоря со специальными атрибутами. Они могут содержать несколько гиперссылок.

Один из способов создания карт состоит в добавлении атрибута `ismap` к тегу ``, помещенному внутри тега `<a>`. Когда пользователь щелкает мышью где-то на рисунке, графический браузер посылает координаты положения мыши серверу, который также указан в якоря. Специальная программа на сервере преобразует эти координаты в некоторое действие, такое как загрузка нового документа. [что происходит на сервере, 6.5.1.1]

Хорошим примером употребления карты может быть выбор отеля по маршруту путешествия. Пользователь щелкает на карте местности, которую он намеревается посетить, и сервер карты возвращает название, адрес и номер телефона местной гостиницы.

Такие мощные и внешне привлекательные карты *со стороны сервера* подразумевают, что автор имеет доступ к программе, обрабатывающей координаты на карте. Тем временем, многие авторы не имеют даже доступа к серверу, не говоря уже о программах на нем. Лучшее решение – воспользоваться картой *со стороны клиента*, или *клиентской*.

Чтобы не зависеть от веб-сервера, можно применять атрибут `usemap` для тега `` вместе с тегами `<map>` и `<area>`, которые позволяют автору вставить всю информацию, нужную браузеру для обработки карты, в тот же документ, где находится и ссылка на ее изображение. Независимость от сервера и меньший объем обмена по сети сделали клиентские карты популярными среди авторов документов и системных администраторов. [клиентские карты, 6.5.2]

¹ Более точно, навигационные карты – это разделение объекта (не только изображения) на замкнутые области, с каждой из которых может быть связано некоторое действие: переход по гиперссылке, запуск сценария и т. п. Назначенное действие выполняется, когда соответствующая область активизируется пользователем. – *Примеч. науч. ред.*

2.9. Списки, строка поиска, формы

Полагаете, мы исчерпали все текстовые элементы? Заголовки, абзацы и концы строк – это всего лишь рудиментарные составляющие для организации текста в документе. Кроме них HTML и XHTML предлагают несколько продвинутых текстовых структур, включая три вида списков, строку поиска и формы. Строки поиска и формы находятся за пределами задач форматирования; они служат взаимодействию с читателями документа. Формы дают возможность пользователю вводить текст, ставить флажки в окошках, нажимать на переключатели, делая свой выбор, и затем возвращать информацию серверу. Специальная программа, получив сообщение от формы, обрабатывает его и соответственно отвечает, либо заполняя бланк заказа, либо подбирая данные, интересующие пользователя.¹

Синтаксис этих специальных средств языка и их разнообразные атрибуты, по-видимому, чрезмерно сложны для быстрого старта. Так что здесь мы их только упоминаем и предлагаем детально с ними познакомиться в следующих главах.

2.9.1. Неупорядоченные, упорядоченные списки и списки определений

Эти три типа списков – как раз те, с которыми мы более всего знакомы. Неупорядоченный список, в котором порядок элементов не имеет значения, такой как перечень бакалейных лавок или прачечных, заключается в теги `` и ``. Каждый элемент списка, чаще всего слово или короткая фраза, помечается тегом `` (list item, пункт списка) и – в случае XHTML – закрывающим тегом ``. Обычно список отображается с отступом от левого края. Как правило, браузер снабжает его элементы маркерами. [тег ``, 7.1.1] [тег ``, 7.3]

Упорядоченные списки, заключенные между тегами `` и ``, совпадают по формату с неупорядоченными, включая теги `` (и ``), когда речь идет о XHTML), помечающими элементы списка. Однако порядок составляющих списка здесь важен, например последовательность сборки оборудования. Браузер отображает каждый элемент списка, ставя перед ним его номер, и располагает их в возрастающем порядке. [тег ``, 7.2.1]

Список определений несколько сложнее предыдущих двух. В списке определений, заключаемом в теги `<dl>` и `</dl>`, каждый элемент состоит из двух частей: имени или заголовка в теге `<dt>` и значения или оп-

¹ Разработка, установка и использование серверных сценариев и программ, которые необходимы для обслуживания форм, находятся за границами нашего повествования. Мы даем некоторые вводные советы в подходящих главах, но за подробной информацией обращайтесь к документации и к администратору сервера.

ределения в теге `<dd>` (XHTML требует соответствующего закрывающего тега). При отображении браузер обычно помещает имя элемента на отдельной строке без отступа, а определение, которое может состоять из нескольких абзацев – с отступом под именем. [тег `<dl>`, 7.5.1]

Различные виды списков содержат практически все, что может входить в тело документа. Так, к примеру, вы можете создать упорядоченный список, включающий коллекцию семейных фотографий в оцифрованном виде, или поместить их в список определений, дополнив аннотацией. Стандарты языков разметки позволяют даже помещать списки внутри списков (вложенные списки), открывая интереснейшие способы различных комбинаций.

2.9.2. Строка поиска

Простейшее интерактивное средство, появившееся в ранних версиях HTML и все еще доступное сегодня, хотя и не рекомендуемое стандартами, – это *строка поиска*, в основе которой лежит тег `<isindex>`. Браузер автоматически предоставляет пользователю какой-либо способ ввести одно или несколько ключевых слов в поле ввода и послать их соответствующему обрабатывающему приложению на сервере.¹

Ясно, что строки поиска крайне ограничены – один и только один элемент пользовательского ввода на документ. К счастью, HTML и XHTML обеспечивают лучшее, более мощное средство для сбора информации, вводимой пользователем, а именно *формы*. [тег `<isindex>` (нежелателен), 6.6.1] [тег `<form>`, 9.2]

Вы создаете один или несколько специальных разделов формы в документе при помощи тегов `<form>` и `</form>`, внутри которых можно разместить как жестко определенные, так и настраиваемые окна ввода, допускающие однострочный или многострочный ввод. Туда можно вставить переключатели выбора между двумя и большим числом вариантов, кнопки специального назначения для приведения содержимого формы в начальное состояние и для отправки его на сервер. Пользователи заполняют форму по своему усмотрению, возможно прочитав перед этим остальные части документа, и нажимают специальную кнопку «отправить», которая приказывает браузеру послать данные формы серверу. Специальная программа со стороны сервера обрабатывает данные и соответствующим образом отвечает браузеру, например запрашивая дополнительную информацию или модифицируя документы, отсылаемые затем пользователю, и т. д. [тег `<form>`, 9.2]

HTML/XHTML-формы предоставляют все средства, которые встречаются у автоматической ее разновидности, включая добавление меток

¹ Этот тег использовался немногими авторами. Тег `<isindex>` был признан «нежелательным» в стандарте HTML 4.0; отправлен, как говорится, в прошлое, но еще не умер.

областей ввода, встраивание текста команд, установку значения полей по умолчанию и т. п., за исключением автоматической проверки ввода, такой как проверка количества цифр в почтовом индексе или в номере телефона. Эту функцию должна исполнять программа на стороне сервера или JavaScript-сценарий на стороне клиента.

2.10. Таблицы

Как и следовало ожидать от языка, возникшего в научном мире, переполненном цифрами, HTML и его потомок XHTML поддерживают набор тегов для таблиц данных, которые не только выравнивают числа, но и могут специальным образом форматировать текст.

Восемь тегов обеспечивают формирование таблицы, включая собственно тег `<table>` и тег `<caption>`, нужный для задания описания таблицы. Специальные атрибуты позволяют изменять размер и внешний вид таблицы. Вы создаете ее построчно, располагая между тегом строки таблицы (`<tr>`) и его закрывающим тегом (`</tr>`) теги данных (`<td>`) или теги заголовка (`<th>`) и их содержимое для каждой ячейки таблицы (а в случае XHTML еще и закрывающие теги). Заголовки и данные могут содержать практически все что угодно: текст, изображения, формы и даже другие таблицы. Поэтому разумно использовать таблицы для сложного форматирования текста, располагая его, скажем, в нескольких колонках, или для создания боковых заголовков (рис. 2.5). За дополнительной информацией обратитесь к главе 10.

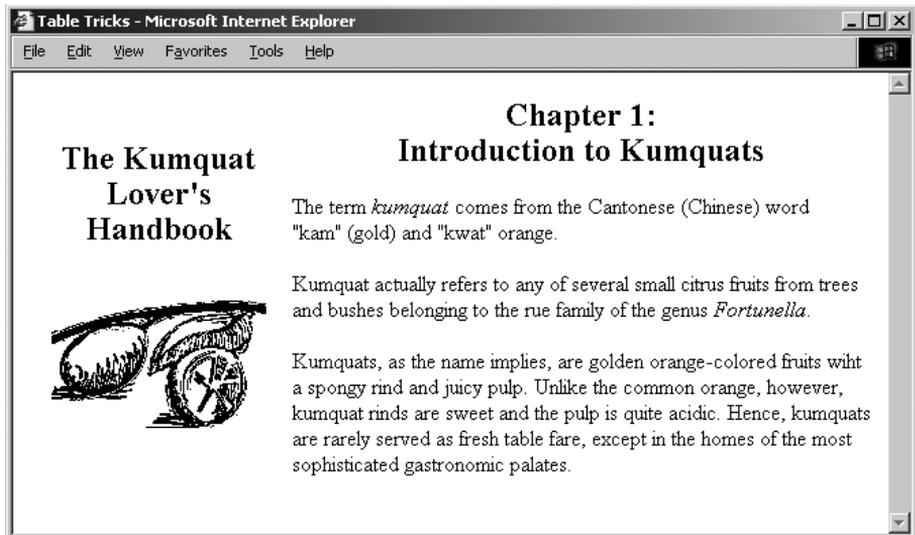


Рис. 2.5. Таблицы тоже позволяют совершать чудеса форматирования

2.11. Фреймы

Всякий, кто открывал на своем экране одновременно несколько окон приложений, немедленно оценит преимущества, которые дают фреймы.¹ Фреймы позволяют разделить окно браузера на несколько областей отображения, в каждой из которых содержится свой документ.

Вот пример отображения фрейма (рис. 2.6). Он показывает, как окно документа может быть разбито на независимые окна, разделенные простыми линейками и полосами прокрутки. А вот чего не видно на рисунке, так это того, что в каждом фрейме может отображаться отдельный документ, который вовсе не обязан иметь HTML/ХHTML-происхождение. Фрейм представляет любое значимое содержимое, которое умеет воспроизводить браузер, в том числе мультимедийное. Если содержимое фрейма включает в себя гиперссылку и пользователь ее выбирает, то новый документ или даже другой документ с фреймами может быть отображен либо во фрейме, с которого осуществлялся

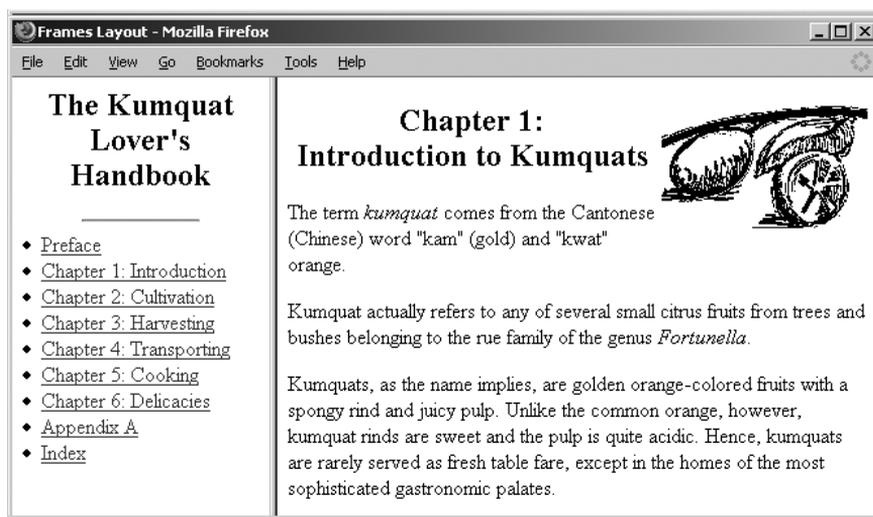


Рис. 2.6. Окно браузера разбито на фреймы, в каждом из которых отображается свой документ

¹ У фреймов есть не только достоинства. Они затрудняют навигацию и установку закладок на загруженные в них документы, содержимое фрейма, не поместившегося в окно браузера, нельзя просмотреть, а если документ не поместился во фрейм, появляются линейки прокрутки. Некоторые браузеры до сих пор не поддерживают фреймы. Поэтому отношение к фреймам предвзятое. Разработчик, решивший использовать фреймы, должен тщательно продумать навигацию и их размещение на странице. Самая распространенная ошибка новичков в веб-строительстве – необоснованное применение фреймов. – *Примеч. науч. ред.*

переход по ссылке, либо в другом фрейме, либо он будет загружен в новое или то же самое окно браузера.

Фреймы определяются в специальном документе, в котором на месте тега `<body>` находятся один или несколько тегов `<frameset>`; они и сообщают браузеру, как разделить его окно на несколько отдельных фреймов. Особые теги `<frame>` содержатся в теге `<frameset>` и указывают на документы, входящие во фреймы. [тег `<frameset>`, 11.3.1]

Документ с фреймами управляет окном в целом, хотя отдельные элементы, представленные в своем окне показа, действуют до известной степени независимо. Можно, тем не менее, распорядиться, чтобы документ, находящийся в одном из фреймов, загрузил и отобразил новое содержимое в смежном фрейме. Например, в окне на рис. 2.6 выбор гиперссылки Chapter (глава) во фрейме Table of Contents (содержание) заставит браузер загрузить эту главу и вывести ее в правом фрейме. При этом пользователю всегда доступно оглавление набора документов, который он просматривает в окне с фреймами. За дополнительной информацией обращайтесь к главе 11.

2.12. Таблицы стилей и JavaScript

Кроме того, браузеры включают поддержку двух мощных нововведений HTML – таблиц стилей и JavaScript. Подобно их близким родственникам в настольных издательских системах, таблицы стилей в HTML/ХHTML позволяют управлять внешним видом документов – стилями и размером шрифтов, цветом, фоном, выравниванием и т. д. Что более существенно, таблицы стилей дают возможность устанавливать характеристики отображения для документа в целом или даже для полного собрания документов.

JavaScript – это язык программирования с функциями и командами, которые позволяют управлять поведением браузера по отношению к пользователю. Эта книга посвящена не JavaScript, но мы довольно подробно обсудим в следующих главах этот язык и покажем, как вставлять программы, написанные на JavaScript, в документы для достижения очень мощных и привлекательных эффектов.

W3C – организация, устанавливающая стандарты, – рекомендует использовать для проектирования HTML/ХHTML-документов модель каскадных таблиц стилей (CSS). Все современные браузеры с графическим интерфейсом поддерживают CSS и JavaScript.¹ «Древний» Netscape 4

¹ Строго говоря, язык JavaScript в полной мере поддерживает только Netscape Navigator. Internet Explorer поддерживает язык Jscript, который во многом сходен с JavaScript, но все же имеет некоторые существенные различия. Тот факт, что интерпретатор Jscript, встроенный в Internet Explorer, воспринимает фрагменты кода JavaScript как код своего родного языка, вносит еще больше путаницы в теорию и практику веб-строительства. – *Примеч. науч. ред.*

работает также с моделью таблиц стилей языка JavaScript (JavaScript-based Style Sheets, JSS), которая описана в главе 12, но мы не рекомендуем ее использовать. Нет, мы скажем иначе: «Не тратьте свое время на JSS». CSS – это всеми одобренный и всеми поддерживаемый способ управления тем, как может (но не обязательно будет) отображаться ваш документ браузером пользователя.

Чтобы проиллюстрировать CSS, покажем способ представления заголовков самого верхнего уровня (h1) красным цветом:

```
<html>
<head>
<title>Пример использования CSS</title>
<!-- Помещаем описание свойств CSS внутрь комментария, чтобы старые браузеры,
которые не поддерживают CSS, не "споткнулись" или не вывели на экран
непонятное им содержимое тега style. -->
  <style type="text/css">
    <!--
      h1 {color: red}
    -->
  </style>
</head>
<body>
<h1>Если ваш браузер поддерживает CSS, то этот текст будет красным</h1>
Что-нибудь между.
<h1> И этот тоже красный!</h1>
</body>
</html>
```

Вы, разумеется, не сможете увидеть красного цвета в этой черно-белой книге, так что мы не станем показывать результат на рисунке. Поверьте нам или наберите и загрузите этот пример в браузер – заключенный в теги `<h1>` текст будет красным на цветном дисплее.

JavaScript – это объектно-ориентированный язык. Он рассматривает документ и отображающий его браузер как совокупность частей (объектов), обладающих определенными свойствами, которые вы можете изменять или вычислять. Это очень мощное средство, но вряд ли оно посылно большинству. Скорее многие из нас попытаются ухватить несколько крутых JavaScript-программ, распространенных в сети, чтобы украсить ими свои документы. Мы расскажем вам, как это сделать, в главе 12.

2.13. Вперед!

Конечно, данная глава – только верхушка айсберга. Если вы дочитали до этого места, наверное, интерес подстегивает вас и к дальнейшим исследованиям. Теперь вы уже имеете базовое представление об основных чертах и пределах возможностей HTML и XHTML. Переходите к следующим главам, чтобы расширить свои знания и узнать больше о каждой из возможностей языка.

По договору между издательством «Символ-Плюс» и Интернет-магазином «Books.Ru – Книги России» единственный легальный способ получения данного файла с книгой ISBN 5-93286-104-5, название «HTML и XHTML. Подробное руководство», 6-е издание – покупка в Интернет-магазине «Books.Ru – Книги России». Если Вы получили данный файл каким-либо другим образом, Вы нарушили международное законодательство и законодательство Российской Федерации об охране авторского права. Вам необходимо удалить данный файл, а также сообщить издательству «Символ-Плюс» (piracy@symbol.ru), где именно Вы получили данный файл.