

Джон Дакетт



**Все, что нужно знать
для создания первоклассных сайтов**

HTML и CSS

Разработка и создание веб-сайтов





**МИРОВОЙ
КОМПЬЮТЕРНЫЙ
БЕСТСЕЛЛЕР**

John Duckett



HTML & CSS

Design and Build Websites

WILEY

Джон Дакетт



HTML и CSS

Разработка и создание веб-сайтов



ЭКСМО

МОСКВА

2013

УДК 004.42
ББК 32.973.26
Д 14

Jon Duckett

HTML and CSS: Design and Build Websites

© 2011 by John Wiley & Sons, Inc., Indianapolis, Indiana

ISBN: 978-1-118-00818-8

All rights reserved. This translation published under licence

Дакетт Д.
Д 14 HTML и CSS. Разработка и дизайн веб-сайтов / Джон Дакетт ; [пер. с англ. М. А. Райтмана]. – М. : Эксмо, 2013. – 480 с. : ил. + 1 CD. – (Мировой компьютерный бестселлер).

ISBN 978-5-699-64193-2

Эта книга – самый простой и интересный способ изучить HTML и CSS. Независимо от стоящей перед вами задачи: спроектировать и разработать веб-сайт с нуля или получить больше контроля над уже существующим сайтом, эта книга поможет вам создать привлекательный, дружелюбный к пользователю веб-контент. Простой визуальный способ подачи информации с понятными примерами и небольшим фрагментом кода знакомит с новой темой на каждой странице. Вы найдете практические советы о том, как организовать и спроектировать страницы вашего сайта, и после прочтения книги сможете разработать свой веб-сайт профессионального вида и удобный в использовании.

**УДК 004.42
ББК 32.973.26**

Все названия программных продуктов являются зарегистрированными торговыми марками соответствующих фирм.

Никакая часть настоящего издания ни в каких целях не может быть воспроизведена в какой бы то ни было форме и какими бы то ни было средствами, будь то электронные или механические, включая фотокопирование и запись на магнитный носитель, если на это нет письменного разрешения ООО «Издательство «Эксмо».

ISBN 978-5-699-64193-2

© Райтман М.А., перевод на русский язык, 2013
© Оформление. ООО «Издательство «Эксмо», 2013

ОГЛАВЛЕНИЕ

Введение	7
Глава 1. Структура	17
Глава 2. Текст	43
Глава 3. Списки	63
Глава 4. Ссылки	73
Глава 5. Изображения	91
Глава 6. Таблицы	121
Глава 7. Формы	137
Глава 8. Дополнительная разметка	167
Глава 9. Flash, видео- и аудиоконтент	189
Глава 10. Введение в CSS	213
Глава 11. Цвет	231
Глава 12. Текст	247
Глава 13. Блоки	281
Глава 14. Списки, таблицы и формы	309
Глава 15. Макет	335
Глава 16. Изображения	381
Глава 17. Макет в HTML5	401
Глава 18. Процесс разработки	423
Глава 19. Практическая информация	445
Предметный указатель	461

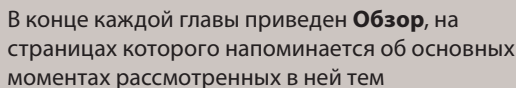
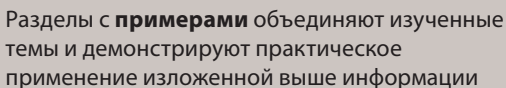
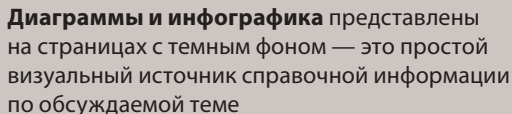
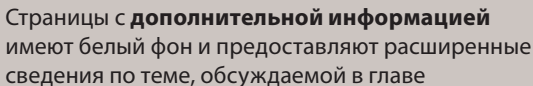
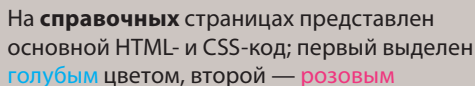
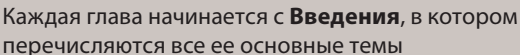
ВВЕДЕНИЕ

- ▶ Об этой книге
- ▶ Принципы Всемирной паутины
- ▶ Обучение на примере уже созданных страниц

В первую очередь, выражаем
благодарность за выбор данной книги.
Она была написана для тех, кто:

- хочет обучиться дизайну и верстке сайтов с нуля;
- владеет собственным сайтом (который, возможно, был создан с использованием системы управления контентом (CMS), блог-платформы, либо платформы электронной коммерции) и желает получить больший контроль над внешним видом его страниц.

Все, что вам понадобится, — это компьютер с установленными веб-браузером и текстовым редактором — таким как Блокнот (Notepad) в операционной системе Windows или TextEdit в OS X.



СЛОЖНО ЛИ НАУЧИТЬСЯ СОЗДАВАТЬ САЙТЫ?

Многие книги, посвященные HTML и CSS, напоминают скучные руководства пользователя. Чтобы вам было легче учиться, мы решили отказаться от традиционных шаблонов, используемых издателями, и в корне пересмотрели оформление этой книги.

На работе коллеги, взглянув на экран моего компьютера и увидев большое количество строк программного кода, говорят мне, что он выглядит очень сложным — и как, должно быть, я невероятно умен, раз понимаю его. Правда состоит в том, что научиться писать и читать код веб-страниц отнюдь не так сложно, как может показаться на первый взгляд: для этого даже не нужно быть «программером».

Понимание языков HTML и CSS полезно всем, чья работа связана со Всемирной паутиной: дизайнеры смогут создавать более привлекательные и удобные в использовании сайты, редакторы — наполнять

сайты более качественным контентом, специалисты по маркетингу — более эффективно общаться с целевой аудиторией и, наконец, менеджеры разного уровня будут более четко формулировать требования при создании сайтов и получать лучшие результаты от своих сотрудников.

При написании книги мы сконцентрировали внимание на коде, который вы будете использовать в 90% случаев, и опустили те его фрагменты, с которыми вам вряд ли доведется иметь дело, даже если создание сайтов — ваша основная работа. Однако, прочитав книгу до конца, вы будете знать, — если в реальной

жизни столкнетесь с кодом, попадающим в эти 10%, — как найти нужную информацию с помощью поисковых систем.

Кроме того, в данную книгу мы также включили дополнительные материалы о подготовке изображений, а также аудио- и видеофайлов для использования во Всемирной паутине, о достижении поставленных задач по дизайну и верстке веб-страниц сайта, об улучшении его показателей в результатах выдачи поисковых систем (то, что называется «поисковой оптимизацией», или SEO*) и о том, как использовать сервис Google Analytics, чтобы получать сведения о посетителях вашего сайта.

* Search Engine optimization (англ.) — поисковая оптимизация. — *Примеч. ред.*

СТРУКТУРА КНИГИ

Чтобы научить вас создавать веб-страницы, данная книга была разделена на три части.

HTML

На протяжении всей первой главы мы будем рассказывать, как можно использовать язык HTML для создания веб-страниц. Вы увидите, что для начала вам потребуется ввести текст, который вы хотите отобразить на создаваемой странице. После этого вам надо будет добавить к словам теги, или элементы, позволяющие браузеру понять, что из текста является заголовком, где начинается и заканчивается абзац и т.д. Оставшаяся часть раздела посвящена описанию этих тегов, используемых для верстки страниц. Все теги, находящиеся в вашем распоряжении, разделены по группам: текст, списки, ссылки, изображения, таблицы, формы, аудио-/видео- и Flash-объекты и пр. Мы также должны вас предупредить, что примеры, приводимые в первых девяти главах, вряд ли можно назвать интересными, но тем не менее это — основа любой веб-страницы. В последующих главах, посвященных CSS, мы покажем вам, как изменить внешний вид ваших веб-страниц, чтобы улучшить впечатление, которое создается от их просмотра у пользователей.

CSS

Мы начнем этот раздел с объяснения того, как используются правила CSS, чтобы предоставить вам возможность управлять стилями и макетом создаваемых веб-страниц. Далее мы продолжим изучение CSS и рассмотрим большое количество свойств, которые вы сможете использовать при создании собственных правил. Обычно все имеющиеся свойства принято разделять на две следующие категории. **Оформление:** Свойства данной категории позволяют управлять такими настройками, как цвет текста, гарнитура и кегль используемых шрифтов, а также добавлять фоновый цвет или изображение на страницы (или части страниц). **Макет:** Эти свойства используются для позиционирования на экране различных элементов страниц. Также вы изучите несколько профессиональных приемов, позволяющих делать создаваемые страницы более привлекательными.

ПРАКТИКУМ

Мы завершим эту книгу полезной информацией, которая поможет вам создавать сайты, выглядящие профессионально. Мы рассмотрим несколько новых тегов, которые появятся в HTML5, они позволяют упростить описание структуры страницы. HTML5 — это новейшая версия языка гипертекстовой разметки (на момент написания книги она еще находилась в процессе разработки). Однако чтобы изучать эти новые элементы, вы уже должны быть хорошо знакомы с тем, как пользоваться средствами CSS для управления дизайном веб-страниц. В данном разделе мы также рассмотрим алгоритм создания страницы, которым вы, вероятно, захотите воспользоваться в будущем при создании собственного сайта. Наконец, мы поговорим о том, что поможет вам пользоваться уже созданным сайтом, например о размещении его на просторах Всемирной паутины, о поисковой оптимизации, а также об использовании аналитического программного обеспечения для отслеживания информации о посетителях и о том, какие разделы сайта они просматривают.

КАК ЛЮДИ ПОЛУЧАЮТ ДОСТУП К ВСЕМИРНОЙ ПАУТИНЕ

Прежде чем приступить к объяснению кода, используемого для верстки сайтов, мы считаем предельно важным сказать пару слов о том, какие способы доступа ко Всемирной паутине существуют, и разъяснить терминологию.

БРАУЗЕРЫ

Для доступа к сайтам пользователи используют специальные программы, называемые **веб-браузерами**. Среди наиболее популярных браузеров можно назвать Firefox, Internet Explorer, Safari, Chrome и Opera.

Чтобы просмотреть какую-либо страницу, пользователи могут ввести ее адрес в специальное поле браузера, перейти по ссылке, размещенной на другом сайте, либо воспользоваться закладкой в разделе «Избранное».

Производители программного обеспечения регулярно выпускают свежие версии браузеров с новыми функциями и поддержкой новых языковых дополнений. Однако важно помнить, что очень многие пользователи не обновляют версии браузеров. Поэтому нельзя полагаться на то, что абсолютно все посетители вашего сайта смогут воспользоваться новейшими функциями, реализованными только в последних версиях программ.

В главе 19 вы узнаете, как определить, с помощью какого браузера посетитель зашел на ваш сайт.

ВЕБ-СЕРВЕРЫ

Когда вы отдаете команду браузеру открыть тот или иной сайт, тот с помощью Интернета посылает запрос специальному компьютеру, называемому **веб-сервером**, на котором расположены файлы нужного вам сайта. Веб-серверы — это специальные компьютеры, постоянно подключенные к Интернету, работа которых оптимизирована для отправки веб-страниц запрашивающим их людям. Некоторые крупные компании используют собственные веб-серверы, но гораздо чаще владельцы сайтов пользуются услугами **хостинговых компаний**, которые берут определенную плату за возможность размещения сайта у них на сервере.

УСТРОЙСТВА

Разновидностей устройств, с помощью которых люди посещают сайты — настольных компьютеров, ноутбуков, планшетов, мобильных телефонов, — становится все больше и больше. Важно помнить, что разные устройства имеют различные размеры экрана и что скорость подключения к Интернету одних выше, чем у других.

ПРОГРАММЫ ЭКРАННОГО ДОСТУПА

Программы экранного доступа* — это специальное ПО, которое читает вслух текст с экрана компьютера. Обычно они используются людьми с нарушением зрения.

Аналогично существующим во многих странах законам, обязывающим общественные заведения быть доступными для инвалидов, в настоящее время принимаются акты, предписывающие создавать сайты, удобные для использования в том числе и людьми с ограничениями. В тексте книги вы несколько раз встретитесь со ссылками на программы экранного доступа. Они помогут вам проверить, доступен ли ваш сайт для пользователей таких программ. Интересно отметить тот факт, что технологии наподобие используемых при разработке программ экранного доступа также применяются в различных сферах деятельности, когда люди оказываются неспособными прочитать текст самостоятельно, например, при вождении машины или занятиях бегом.

* Screen readers (англ.). —
Примеч. ред.

КАК СОЗДАЮТСЯ САЙТЫ

Языки HTML и CSS используются для создания любых сайтов, однако системы управления контентом и платформы для блогов и электронной коммерции зачастую добавляют в этот коктейль еще некоторые технологии.

ЧТО ВЫ ВИДИТЕ

Когда вы просматриваете веб-страницу, ваш браузер, скорее всего, получает HTML- и CSS-код с сервера, на котором расположен данный сайт. Чтобы создать страницу, пригодную для просмотра, браузер интерпретирует код. В большинство веб-страниц обычно бывает включен дополнительный контент, например изображения, аудио- и видеоролики или анимация, и эта книга научит вас подготавливать подобные материалы для использования во Всемирной паутине, а также добавлять их на страницы. Некоторые страницы также посылают вашему браузеру код JavaScript и Flash — и вы также узнаете, как добавлять подобные интерактивные объекты на свой сайт. JavaScript и Flash — это более сложные темы, за изучение которых вы при желании можете взяться после того, как освоите верстку страниц с помощью HTML и CSS.

КАК СОЗДАЮТСЯ САЙТЫ

Небольшие сайты обычно верстаются только на HTML и CSS. Более крупные

сайты, в частности те, что подвергаются регулярному обновлению и используют системы управления контентом*, а также платформы для блогов и электронной коммерции, зачастую создаются с применением более сложных технологий, функционирующих на стороне веб-сервера. Однако на самом деле все эти технологии используются для создания необходимого HTML- и CSS-кода, впоследствии посылаемого браузеру. Потому, если имеющийся у вас сайт использует эти технологии, вы сможете применить новые знания по HTML и CSS, чтобы получить куда больший контроль над его оформлением.

При создании более крупных, сложных сайтов часто используются базы данных для хранения информации и различные языки программирования, такие как PHP, ASP.Net, Java или Ruby, работающие на сервере, но, чтобы управлять внешним видом вашего сайта, вам не потребуется знание этих технологий. Навыков, полученных вами по прочтении данной книги, должно хватить, чтобы помочь вам разобраться в создании сайтов.

HTML5 И CSS3

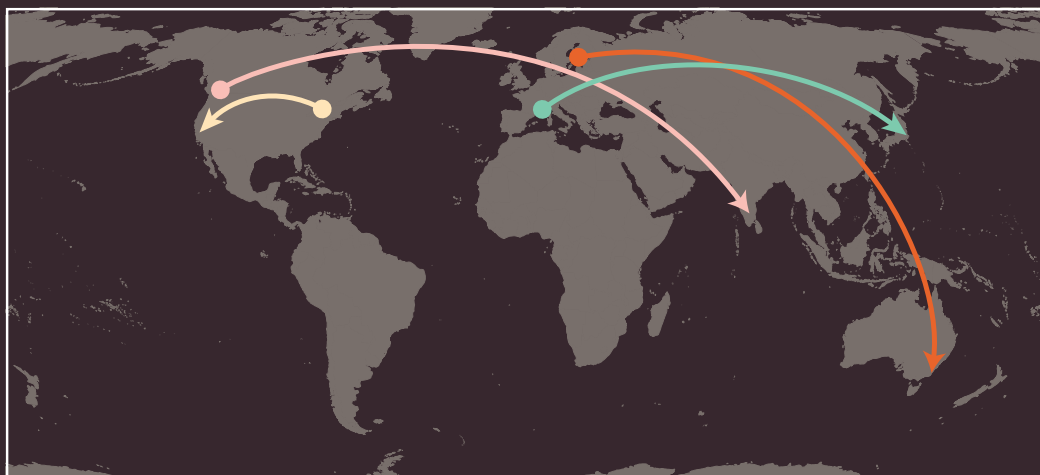
С момента создания Всемирной паутины последовательно разрабатывалось несколько версий HTML и CSS, и целью каждой было внести очередные усовершенствования в технологию.

На момент написания данной книги спецификации HTML5 и CSS3 все еще находились в стадии разработки. Однако несмотря на то, что еще не было выпущено финальных версий обеих технологий, многие браузеры уже поддерживают некоторые из новых функций и многие создатели сайтов уже применяют новейшие элементы кода на своих страницах.

Поэтому мы решили обучить вас использованию именно их. По причине того, что HTML5 и CSS3 были созданы на основе предыдущих версий, вы, изучив их, сможете понимать любой HTML- и CSS-код. Мы также добавили пояснения, где именно используемый нами код является новым и где он с большой вероятностью не сработает в устаревших версиях браузеров.

* CMS, content management systems (англ.). — *Примеч. ред.*

Сервер, на котором размещен посещаемый вами сайт, может находиться в любой точке мира. Чтобы определить его местоположение, ваш браузер в первую очередь подключается к DNS-серверу*.



На приведенной странице вы можете видеть пример расположения сервера, на котором размещен посещаемый вами сайт, в любой точке планеты. DNS-серверы помогают браузеру найти нужный сайт.

- Пользователь из Барселоны запрашивает сайт **sony.jp**, находящийся в Токио.
- Пользователь из Нью-Йорка запрашивает сайт **google.com**, находящийся в Сан-Франциско.
- Пользователь из Стокгольма запрашивает сайт **quantas.com.au**, находящийся в Сиднее.
- Пользователь из Ванкувера запрашивает сайт **airindia.in**, находящийся в Бангалоре.

На следующей странице показано, что происходит, когда пользователь из Англии хочет просмотреть сайт Лувра, расположенный во Франции по адресу **www.louvre.fr**. В первую очередь браузер, находящийся в Кембридже, связывается с лондонским DNS-сервером. Тот сообщает браузеру местоположение сервера, на котором расположен запрашиваемый сайт (он находится в Париже).

* Domain name system (англ.), система доменных имен. — Примеч. ред.

1

Подключение к Интернету происходит с помощью провайдера. Чтобы перейти на какой-либо сайт, вам необходимо ввести в адресную строку браузера доменное имя или адрес интересующего вас сайта, например **google.ru**, **www.bbc.co.uk**, **microsoft.com**.

2

Ваш компьютер подключается к сети, состоящей из серверов, называемых DNS-серверами. По своему назначению сродни телефонной книге: они сообщают вашему компьютеру IP-адреса, соответствующие запрашиваемому доменному имени. IP-адрес — это число, состоящее из не более 12 цифр, разделенных точками на четыре группы. Каждое устройство, подключенное к Интернету, имеет свой уникальный IP-адрес, который напоминает телефонный номер, присвоенный тому или иному абоненту.

3

Уникальный номер, возвращаемый DNS-сервером на ваш компьютер, позволяет браузеру подключиться к серверу, на котором расположен запрошенный вами сайт.

Кембридж

ЛОНДОН

ПАРИЖ

4

После этого веб-сервер возвращает вашему браузеру запрошенную вами страницу.

1

СТРУКТУРА

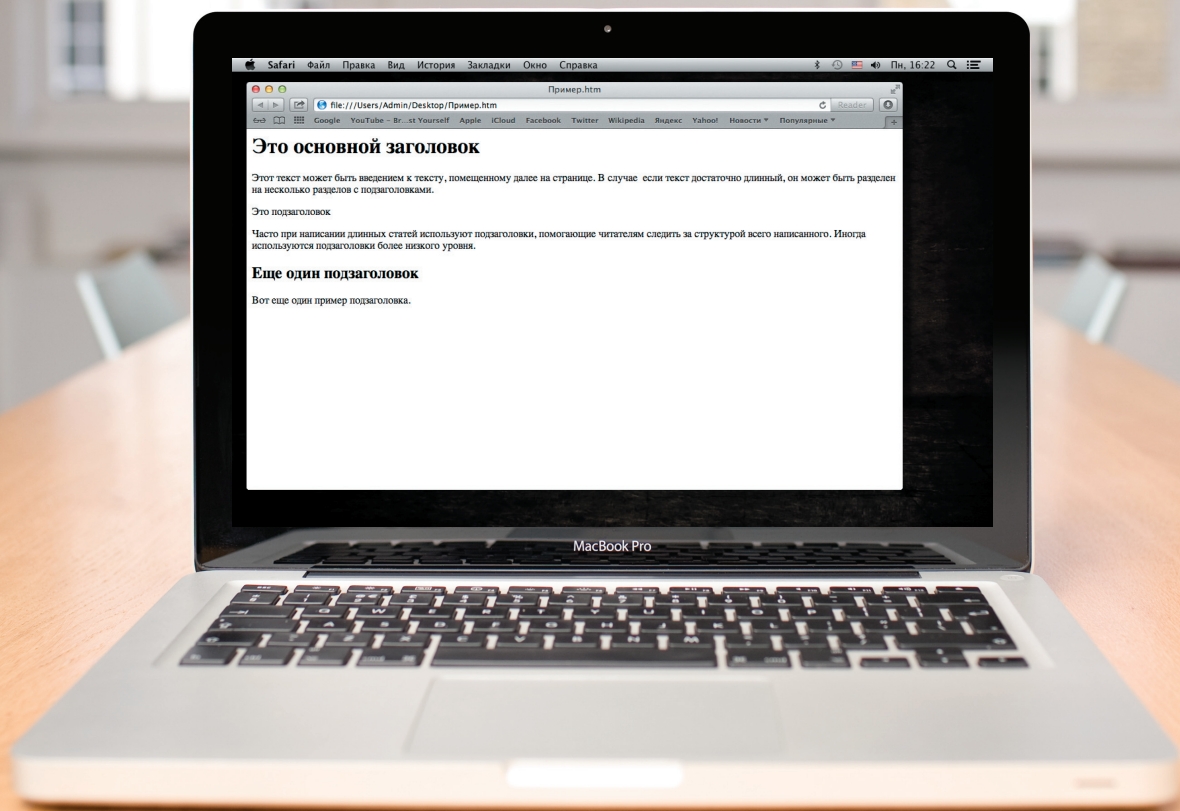
- ▶ Понимание структуры
- ▶ Изучение принципов разметки
- ▶ Теги и элементы

Ежедневно мы сталкиваемся с различными типами документов. Газеты, заявления на выдачу паспорта, каталоги — этот список можно продолжать до бесконечности.

Многие веб-страницы функционируют как электронные версии вышеперечисленных документов. Например газеты публикуют одни и те же репортажи как на бумаге, так и на собственных сайтах; с помощью Всемирной паутины вы также можете написать заявление на получение паспорта; интернет-магазины содержат онлайн-версии каталогов и средства электронной коммерции.

Структура важна для всех типов документов: она помогает читателям понять информацию, которую вы пытаетесь до них донести, а также способствует более простому переходу между частями документов. Таким образом, чтобы научиться создавать веб-страницы, важно понять, как правильно структурировать документы. В этой главе вы:

- увидите, как язык HTML описывает структуру веб-страницы;
- узнаете, как нужно вставлять в документ теги и элементы;
- сверстаете свою первую веб-страницу.



ПРИМЕРЫ СТРУКТУРИЗАЦИИ СТРАНИЦ

Давайте вспомним, как выглядят статьи, публикуемые в газетах: у каждой новостной заметки есть свой заголовок, текст и зачастую одно или несколько изображений. Если перед вами длинная статья, то она может быть разделена на несколько частей подзаголовками, в ней также иногда содержатся специально выделенные врезки-цитаты.

Если вы читаете статью во Всемирной паутине,

структура ее будет лишь незначительно отличаться от той, что вы видите на бумаге (к таким публикациям, например, иногда прикрепляют какие-либо аудио- или видеофайлы). Взглянув на следующую страницу, вы увидите отсканированную журнальную статью и ее электронную версию, опубликованную на сайте журнала.

А теперь давайте вспомним, как выглядит абсолютно

иной вид документа: заявление на выдачу паспорта. Как правило, у подобных форм есть несколько разделов, каждый из которых может иметь собственный заголовок. Во всех разделах имеются поля для заполнения и вопросы к ним, а также ячейки, чтобы ставить галочки. Структура этого документа, опять же, очень похожа на электронную версию формы, опубликованную на сайте Федеральной миграционной службы.

Шар в шаре

Прелесть Шанхая в том, что этот безусловно китайский город не определить как «китайский». То есть не «китайский Париж» и не «китайский Нью-Йорк»

Текст: Дмитрий Губин
Фото: Петер Блобрезский

Ш
а

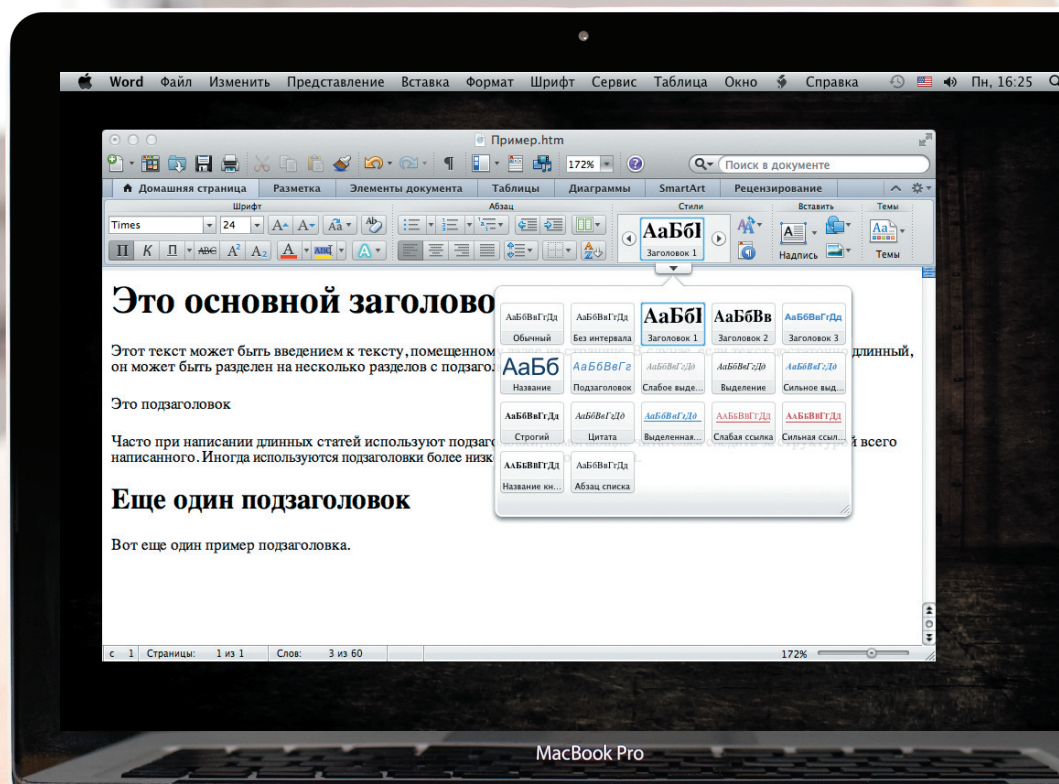


СТРУКТУРИРОВАНИЕ ДОКУМЕНТОВ MICROSOFT WORD

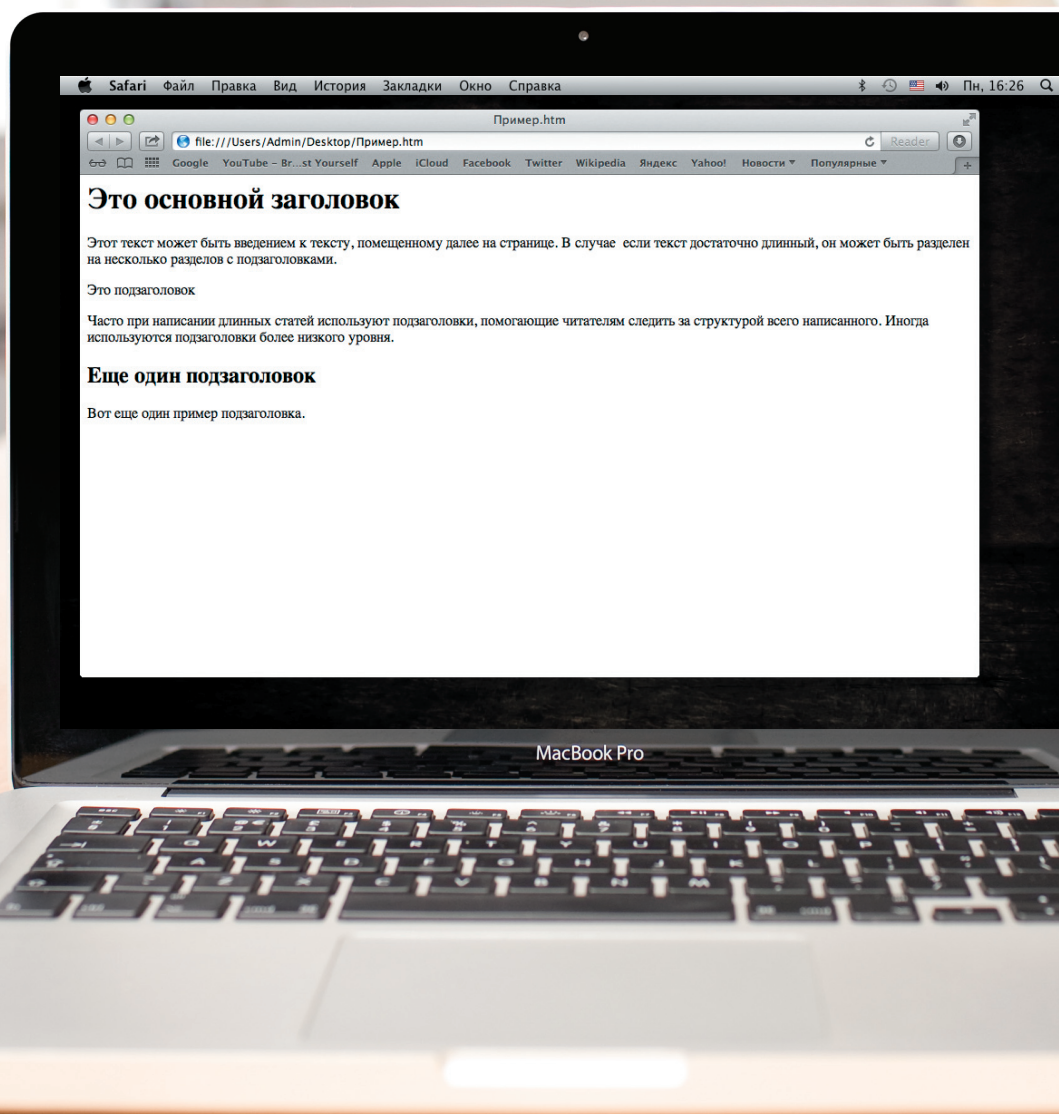
Использование заголовков и подзаголовков при составлении документов различных типов помогает отразить иерархию предоставляемых данных. Например любой документ должен начинаться с заголовка, написанного крупным шрифтом, после которого, как правило, следует введение или резюме с наиболее важной информацией.

Такая структура может быть расширена с помощью подзаголовков, помещаемых в тексте ниже на странице. При составлении документа с помощью текстового редактора принято разделять фрагменты текста, чтобы структурировать их. Каждая подтема может выделяться в отдельный абзац, кроме того, каждый подраздел также может иметь собственный заголовок, описывающий то, о чем говорится в следующем за ним тексте.

На следующей странице представлен простой документ, составленный в программе Microsoft Word. В раскрывающемся списке показаны различные стили текста, которые можно применить в документе, например для создания заголовков разного уровня. Если вы часто работаете с программой Word, то вам, вероятно, уже приходилось использовать инструменты форматирования или палитру цветов для выделения заголовков и подзаголовков.



На предыдущей странице был приведен пример структурирования документа Microsoft Word, используемого для упрощения восприятия текста. Подобное использование структуры принято и при верстке веб-страниц.



Это основной заголовок

Этот текст может быть введением к тексту, помещенному далее на странице. В случае если текст достаточно длинный, он может быть разделен на несколько разделов с подзаголовками.

Это подзаголовок

Часто при написании длинных статей используют подзаголовки, помогающие читателям следить за структурой всего написанного. Иногда используются подзаголовки более низкого уровня.

Еще один подзаголовок

Вот еще один пример подзаголовка.

ИСПОЛЬЗОВАНИЕ ЯЗЫКА HTML ДЛЯ ОПИСАНИЯ СТРУКТУРЫ ВЕБ-СТРАНИЦ

В окне браузера вы видите документ с абсолютно таким же содержимым, как и на скриншоте программы Word на странице 18. Для описания структуры веб-документа необходимо добавить немного кода к тексту, который вы хотите отобразить на экране.

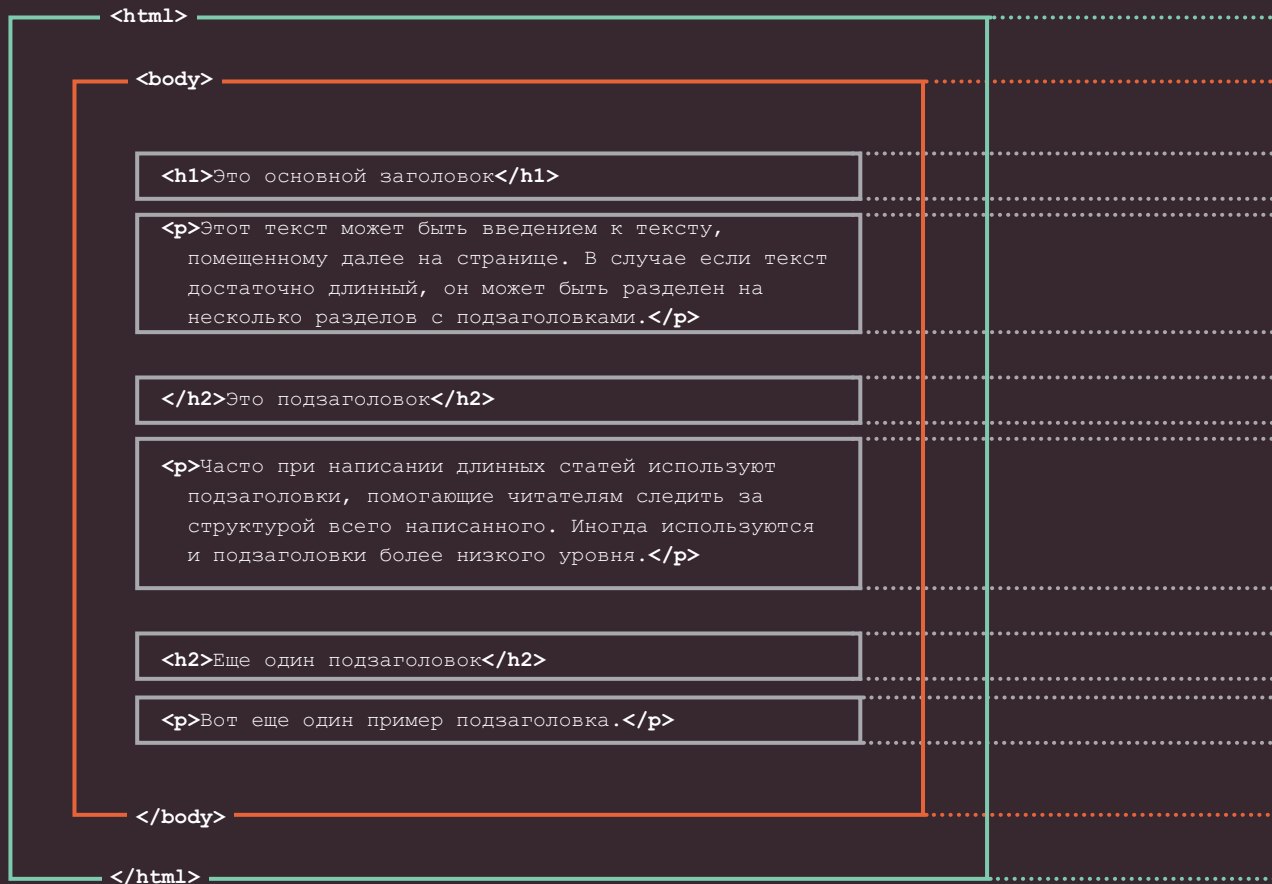
Ниже на странице приведен код на языке HTML. Не волнуйтесь по поводу того, что вы не понимаете его назначения. Уже на следующей странице мы начнем подробно знакомить вас с ним. Обратите внимание, что HTML-код напечатан голубым цветом, а текст, выводимый на экран, — черным.

```
<html>
  <body>
    <h1>Это основной заголовок</h1>
    <p>Этот текст может быть введением к тексту,
помещенному
    далее на странице. В случае если текст достаточно
    длинный, он может быть разделен на несколько
разделов
    с подзаголовками.</p>
    </h2>Это подзаголовок</h2>
    <p>Часто при написании длинных статей используют
    подзаголовки, помогающие читателям следить за
структурой
    всего написанного. Иногда используются
и подзаголовки
    более низкого уровня.</p>
    <h2>Еще один подзаголовок</h2>
    <p>Вот еще один пример подзаголовка.</p>
  </body>
</html>
```

HTML-код (выделен голубым цветом) состоит из символов, помещенных между двух угловых скобок. Все это называется **HTML-элементами**. Элементы, как правило, состоят из двух **тегов**: открывающего и закрывающего. (Внутри закрывающего тега также имеется косая черта, иначе называемая **слеш**.) Каждый HTML-элемент сообщает браузеру какую-либо информацию о тексте, помещенном между открывающим и закрывающим тегами.

Теперь давайте более детально рассмотрим код, с которым вы познакомились на предыдущей странице. Этот код содержит несколько различных элементов. У каждого из них имеются открывающий и закрывающий теги.

КОД



**Элементы аналогичны контейнерам.
Они сообщают какую-либо информацию
о тексте, расположенном между
открывающим и закрывающим тегами.**

ОПИСАНИЕ

Открывающий тег `<html>` означает, что все, помещенное между ним и закрывающим тегом `</html>`, является HTML-кодом.

Тег `<body>` сообщает, что все, помещенное между ним и закрывающим тегом `</body>`, должно быть отображено в основном окне браузера.

Текст между тегами `<h1>` и `</h1>` представляет основной заголовок.

Абзац текста должен быть помещен между тегами `<p>` и `</p>`.

Текст между тегами `<h2>` и `</h2>` представляет подзаголовок.

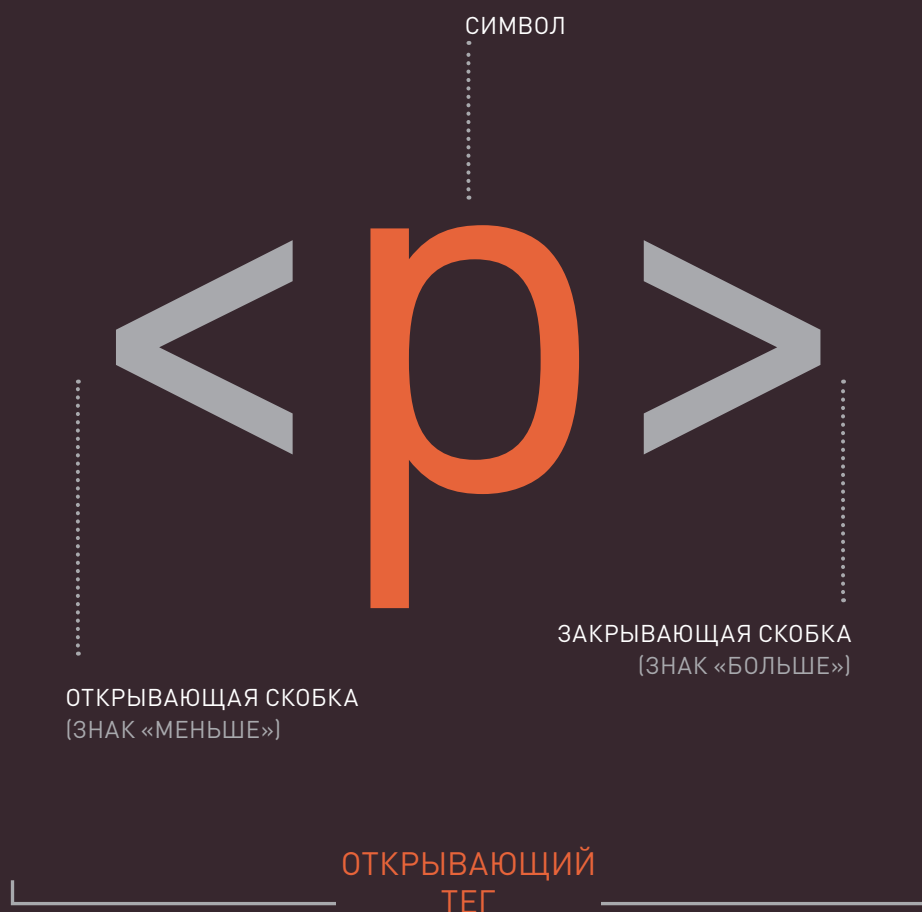
Это еще один абзац текста, помещенный между тегами `<p>` и `</p>`.

Еще один подзаголовок между тегами `<h2>` и `</h2>`

Еще один абзац текста между тегами `<p>` и `</p>`

Закрывающий тег `</body>` означает окончание контента, который должен быть отображен в основном окне браузера

Закрывающий тег `</html>` означает конец HTML-кода.

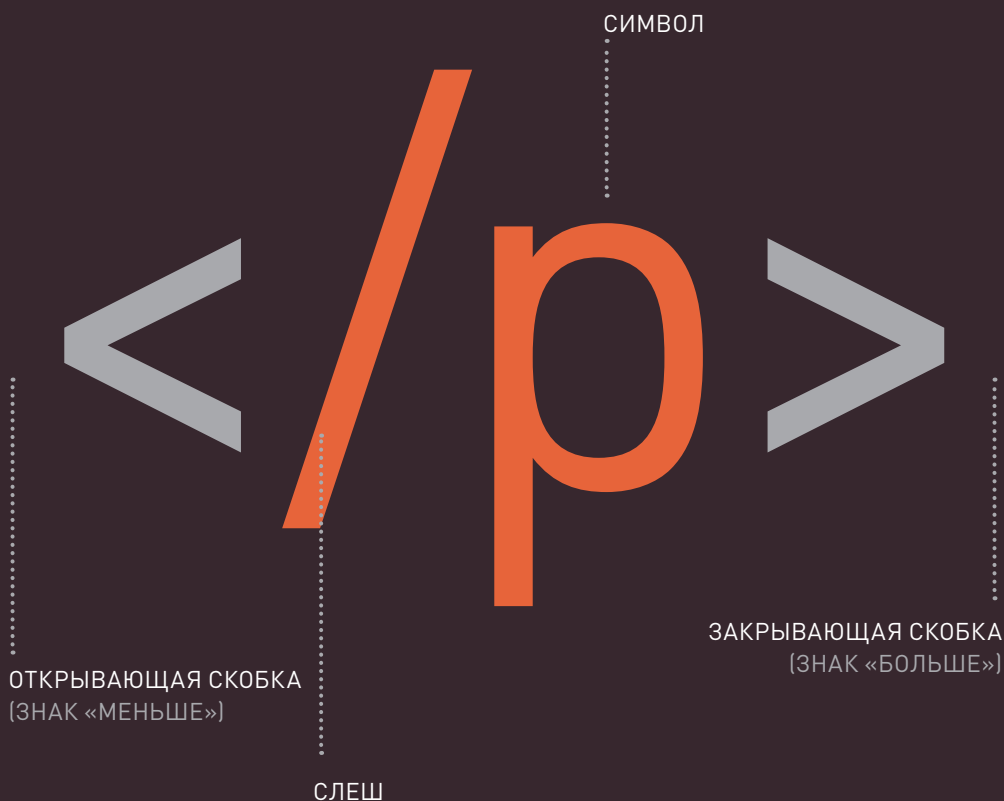


Символы в скобках означают, для каких целей можно использовать данный тег

Например, тегом **р*** обозначается абзац текста

Отличием закрывающего тега является наличие слеша после символа **<**.

* От *англ.* paragraph — абзац. —
Примеч. ред.

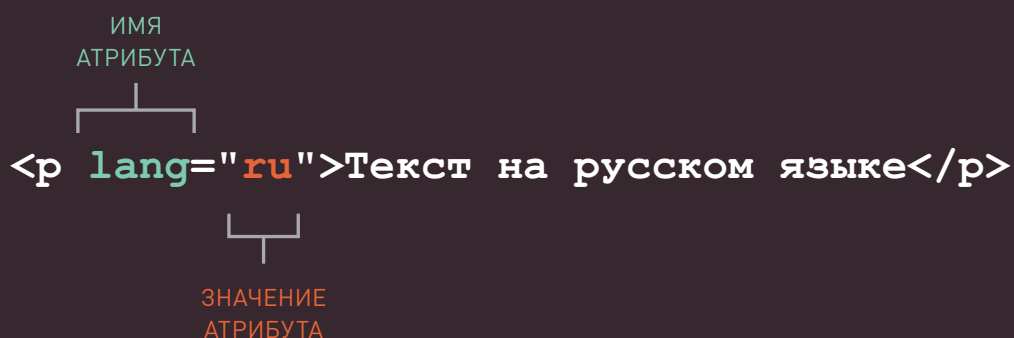


Термины «тег» и «элемент» зачастую используются как синонимы.

Однако, строго говоря, элемент включает в себя и открывающий тег,

и закрывающий, а также любой контент, расположенный между ними.

Атрибуты предоставляют дополнительную информацию о содержимом HTML-элементов. Они располагаются в открывающем теге элемента и состоят из двух частей: имени и значения, разделенных знаком «равно».



Имя атрибута определяет, какого рода дополнительную информацию о содержимом элемента вы намереваетесь сообщить, в то время как **значение** — это

собственно информация или настройка элемента, изменяемая атрибутом. Значение следует заключить в двойные кавычки. У разных атрибутов могут быть различные значения.

Выше в качестве примера приводился атрибут `lang`, используемый, для обозначения языка текста данного элемента. Значение атрибута из примера устанавливает русский язык.

Язык HTML5 допускает написание значений атрибутов заглавными буквами без кавычек, но делать это не рекомендуется.



Большинство атрибутов могут быть использованы только с определенными элементами. Несмотря на это некоторые атрибуты (например, `lang`) допускается ставить в любом элементе.

Большинство значений атрибутов могут быть либо предопределенными, либо установленными в соответствии с принятыми стандартами. По мере изучения атрибутов вы также узнаете и о доступных

для них значениях. Так, значением атрибута `lang` может быть сокращенное обозначение языка содержимого элемента. Браузеры должны распознавать это сокращение.

ЭЛЕМЕНТЫ BODY, HEAD И TITLE

<body>

С тегом `<body>*` вы уже встречались в первом примере. Все, помещаемое внутрь этого элемента, отображается в основном окне браузера.

<head>

Вам часто придется видеть элемент `<head>**`, предшествующий элементу `<body>`. Он содержит информацию о самой странице, а не ту, которая будет выведена в основную часть окна браузера (выделенную голубым цветом на следующей странице). В элемент `<head>` также часто включается элемент `<title>`.

<title>

Содержимое элемента `<title>***` выводится либо в заголовке окна браузера (выше текстового поля, в которое вы обычно вводите адрес сайта), либо в качестве названия вкладки страницы (если ваш браузер использует вкладки).

глава-01/body-head-title.html

HTML

```
<html>
  <head>
    <title>Это заголовок страницы</title>
  </head>
  <body>
    <h1>Это "тело" страницы</h1>
    <p>Содержимое, помещенное в "тело" страницы,
      будет отображено в основном окне браузера.</p>
  </body>
</html>
```

Это "тело" страницы

РЕЗУЛЬТАТ

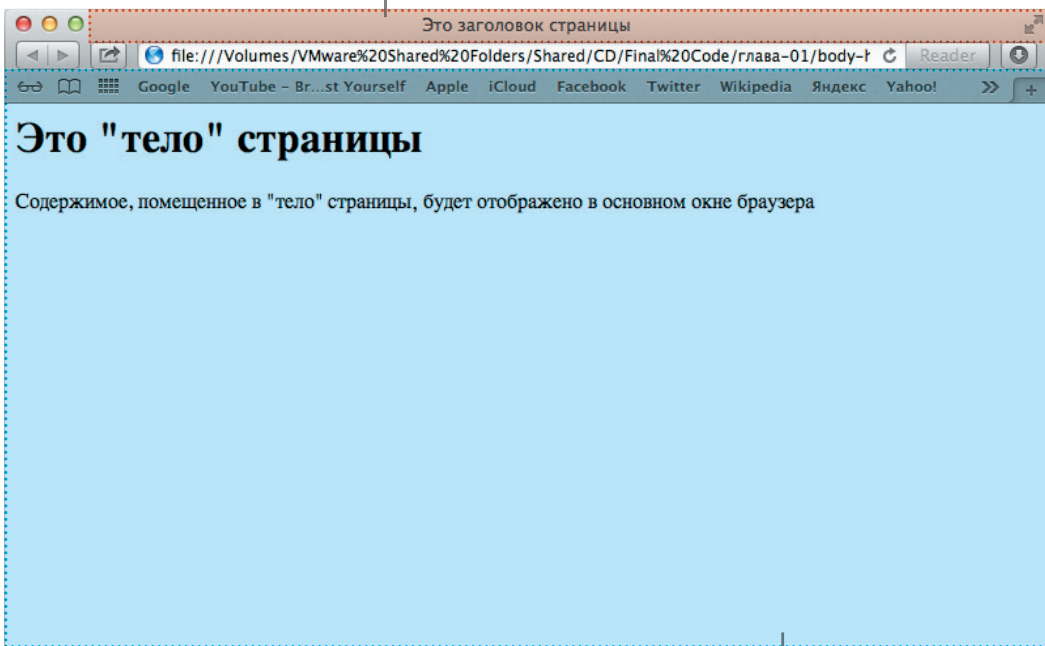
Содержимое, помещенное в "тело" страницы, будет отображено в основном окне браузера

* Тело (англ.). — Примеч. ред.

** Голова (англ.). — Примеч. ред.

*** Заголовок (англ.). — Примеч. ред.

Весь текст, находящийся между тегов `<title>`, выводится в заголовок окна браузера или вкладки страницы (на рисунке выделен **оранжевым** цветом)



Все, находящееся между тегами `<body>`, выводится в основное окно браузера (на рисунке выделено голубым цветом)

Возможно, вы уже знаете, что аббревиатура HTML расшифровывается как HyperText Markup Language — язык разметки гипертекста. В данном случае слово «гипертекст» означает, что с помощью языка HTML вы можете создавать ссылки, позволяющие

посетителям быстро и легко перемещаться с одной страницы на другую. Язык разметки позволяет вам создать примечания к тексту, а эти примечания, в свою очередь, придают дополнительное значение содержимому документа. Представьте себе

веб-страницу: к исходному тексту, который вы хотите отобразить в окне браузера, вы добавляете какой-либо код, используемый браузером для корректного отображения страницы. Поэтому теги, добавляемые вами, называются «элементами разметки».

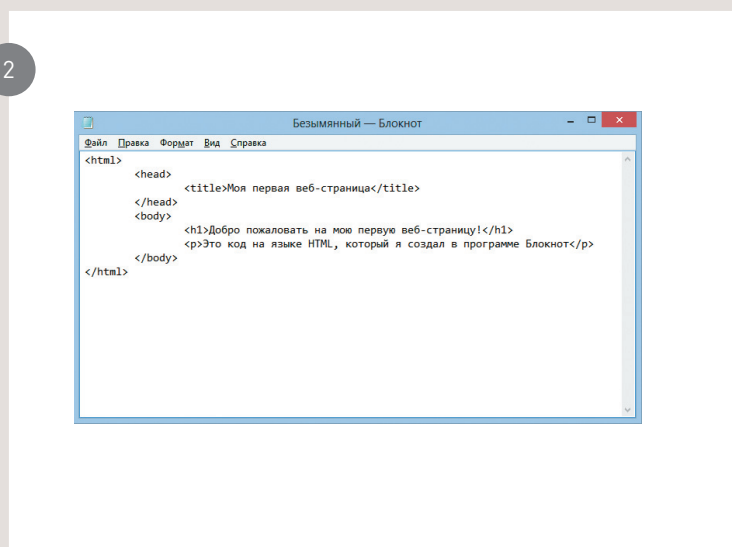
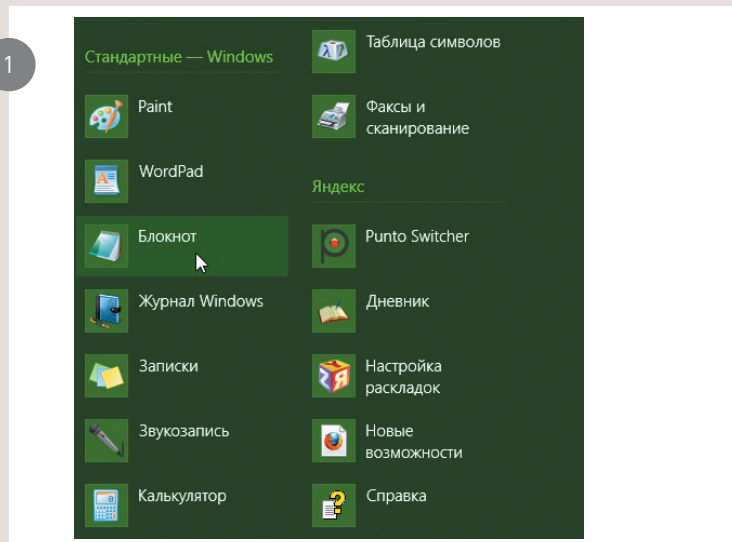
СОЗДАНИЕ ВЕБ-СТРАНИЦЫ В ОПЕРАЦИОННОЙ СИСТЕМЕ WINDOWS

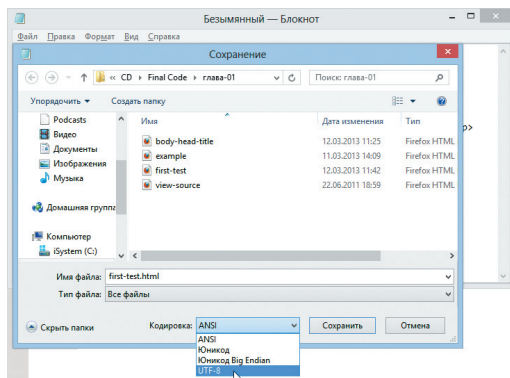
1 Чтобы создать вашу первую веб-страницу на компьютере под управлением операционной системы Windows, запустите программу Блокнот (Notepad). Это можно сделать следующим образом: на экране **Пуск** (Start) в операционной системе Windows 8 щелкните правой кнопкой мыши и выберите пункт **Все приложения** (All Programs) в правом нижнем углу экрана. В списке приложений в группе **Стандартные — Windows** (Accessories — Windows) щелкните мышью по ярлыку приложения Блокнот (Notepad).

В предыдущих версиях операционной системы Windows, выберите команду меню **Пуск** ⇒ **Все программы** ⇒ **Стандартные** ⇒ **Блокнот** (Start ⇒ All Programs ⇒ Accessories ⇒ Notepad).

Вы также можете загрузить бесплатный текстовый редактор Notepad++ с сайта notepad-plus-plus.org.

Если вы пользуетесь более ранней версией Windows, выполните команду меню **Пуск** ⇒ Введите в окне программы код, показанный на рисунке справа.



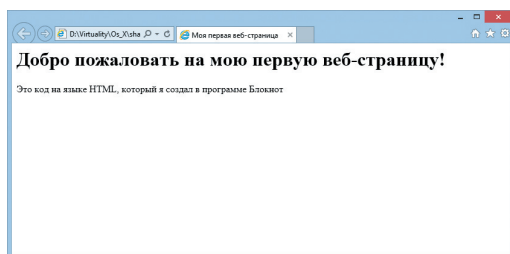


3

В меню **Файл** (File) выберите команду **Сохранить как** (Save As). Сохраните файл и запомните, где он находится. При желании вы сможете создать отдельную папку для всех примеров из этой книги.

Сохраните этот файл под именем *first-test.html*. Не забудьте выбрать пункт **Все файлы** (All Files) в раскрывающемся списке **Тип файла** (Save as type), чтобы сохранить документ как веб-страницу, а не как простой текст.

Для веб-страниц с кириллическими буквами **обязательно** выберите пункт **UTF-8** в раскрывающемся списке **Кодировка** (Encoding). Это необходимо для правильного отображения русского текста в окне браузера.



4

Запустите браузер. В меню **Файл** (File) выберите команду **Открыть** (Open). Перейдите в папку, содержащую только что созданный файл, выделите его и щелкните по кнопке **Открыть** (Open). Появившаяся страница должна быть похожа на ту, что изображена на рисунке слева.

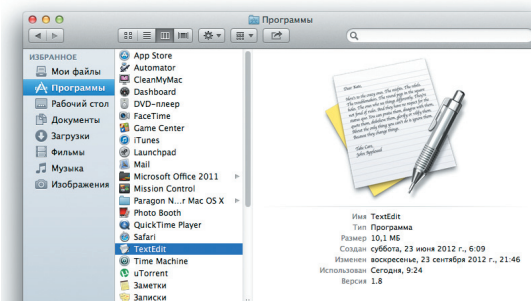
Если открытая вами страница отличается от изображенной на рисунке, найдите только что созданный файл и убедитесь, что он имеет расширение **.html** (если у файла расширение **.txt**, то вам придется вернуться в программу Блокнот (Notepad) и заново сохранить его, поместив имя файла **"first-test.html"** в кавычки).

СОЗДАНИЕ ВЕБ-СТРАНИЦЫ В ОПЕРАЦИОННОЙ СИСТЕМЕ OS X

Чтобы создать страницу на компьютере под управлением операционной системы OS X, запустите программу TextEdit, расположенную в папке **Программы** (Applications).

Вы также можете загрузить бесплатный текстовый редактор для создания веб-страниц TextWrangler с сайта www.barebones.com.

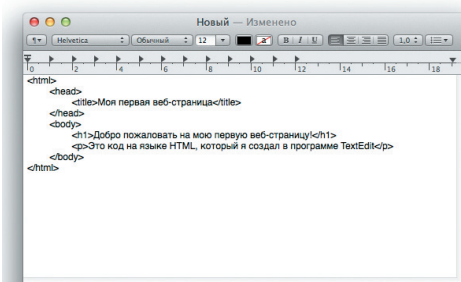
1



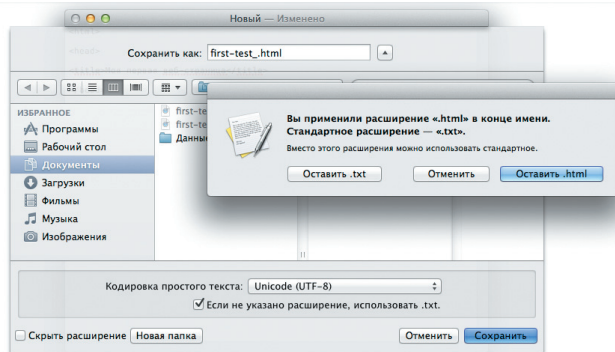
Программа TextEdit по умолчанию создает не простые текстовые, а RTF-документы. Для работы с HTML-кодом, выберите команду меню

Формат ⇒ Конвертировать в простой текст (Format ⇒ Make Plain Text).

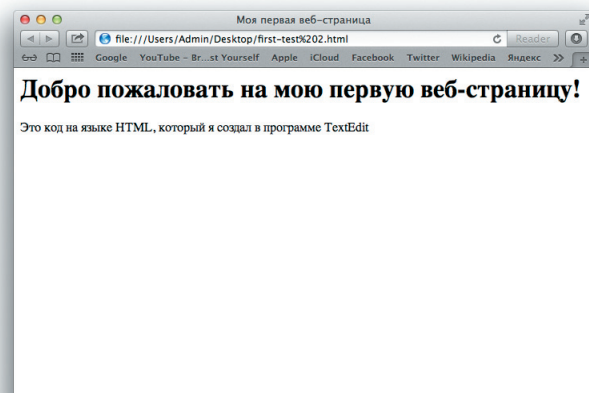
2



3



4

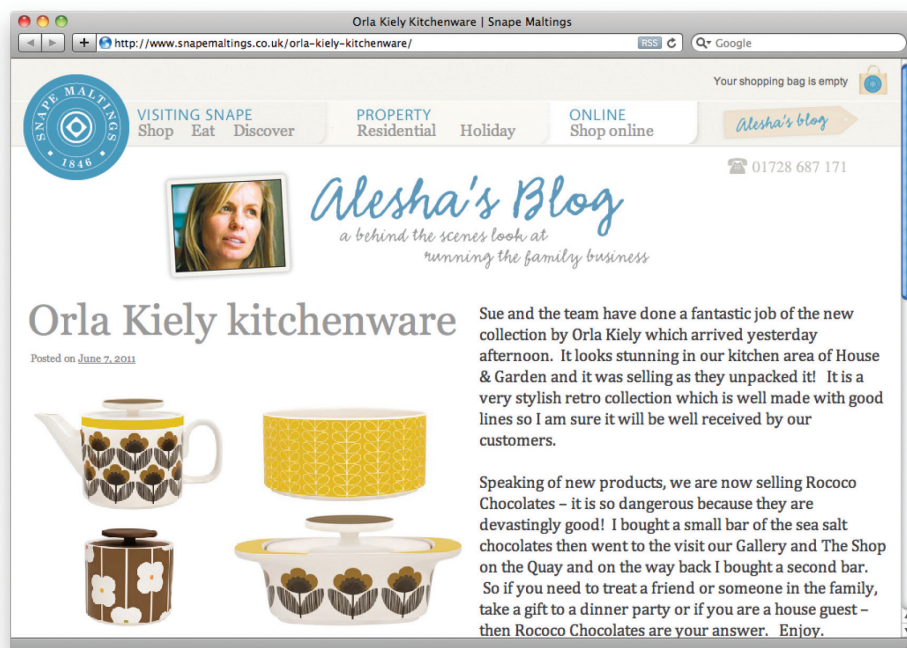


Теперь перейдите в меню **Файл** (File) и выберите команду **Сохранить** (Save). Сохраните файл и запомните его место-положение.

При желании вы можете создать отдельную папку для всех примеров из этой книги. Сохраните данный файл под именем *first-test.html*. Если на экране появится диалоговое окно с запросом, какое расширение использовать для файла, нажмите кнопку **Оставить .html** (Use .html). Далее запустите браузер. В меню **Файл** (File) выберите команду **Открыть файл** (Open file). Перейдите в папку, содержащую только что созданный вами файл, выделите его и щелкните по кнопке **Открыть** (Open). Страница должна быть похожа на ту, что изображена на рисунке слева.

Если на открытой вами странице кириллический текст отображается неправильно, вам следует выбрать кодировку вручную (как указывать ее в коде, вы узнаете позже). В меню браузера Safari выберите команду **Вид** ⇒ **Кодировка текста** ⇒ **Unicode (UTF-8)** (View ⇒ Encoding ⇒ Unicode (UTF-8)).

КОД В СИСТЕМЕ УПРАВЛЕНИЯ КОНТЕНТОМ



Обычно при работе с системами управления контентом, а также платформами для блогов или электронной коммерции для получения контроля над ними необходимо перейти в специальный административный раздел сайта. Инструменты, предоставляемые в таких разделах, как правило, позволяют редактировать отдельные участки страниц, но не страницу целиком, поэтому при работе с ними вы редко будете встречать такие элементы, как `<html>`, `<head>` или `<body>`. Взглянув на рисунок с изображением системы управления контентом на следующей странице, вы

увидите два текстовых поля, позволяющие вводить заголовки документа и основной текст. Также на этой странице вы сможете указать дату публикации и обозначить, в какой раздел сайта должна быть помещена статья. Если бы вы работали с сайтом интернет-магазина, то вы, возможно, встретили бы поля, позволяющие вводить название продукта, его описание, цену и количество, имеющееся на складе. Это возможно благодаря тому, что для управления всеми страницами раздела сайта используется один и тот же шаблон. (Например, на сайте интернет-магазина может использоваться один шаблон

для отображения информации обо всех имеющихся продуктах.) Предоставляемая вами информация о товаре вводится в шаблон. Преимущество данного метода в том, что он позволяет добавлять информацию на сайт даже людям, не умеющим верстать веб-страницы. Кроме того, вы можете изменить внешний вид отдельных блоков шаблона, и все страницы, созданные с его использованием, автоматически обновятся. Представив себе магазин, продающий 1000 наименований продукции, вы поймете, что отредактировать один шаблон гораздо проще, чем править страницу каждого