Джон Дакетт

Все, что нужно знать
для создания первоклассных сайтов

HTML и CSS

Разработка и создание веб-сайтов

МИРОВОЙ
КОМПЬЮТЕРНЫЙ
БЕСТСЕЛЛЕР

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

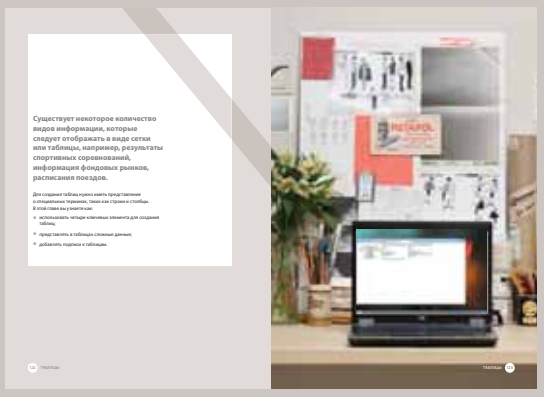
ВВЕДЕНИЕ

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

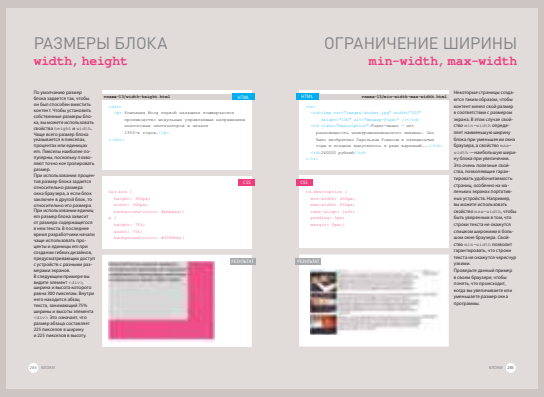
В первую очередь, выражаем благодарность за выбор данной книги. Она была написана для тех, кто:

- 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.

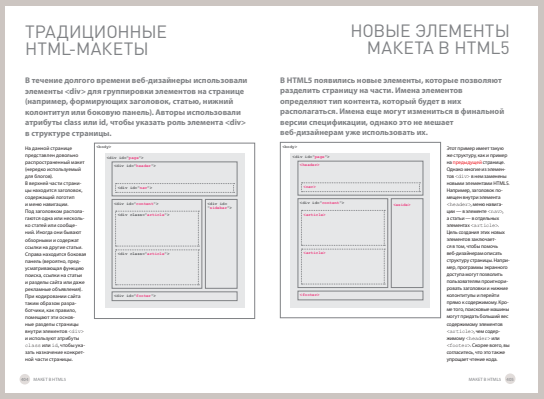
Все, что вам понадобится, — это компьютер с установленными веб-браузером и текстовым редактором — таким как Блокнот (Notepad) в операционной системе Windows или TextEdit в OS X.



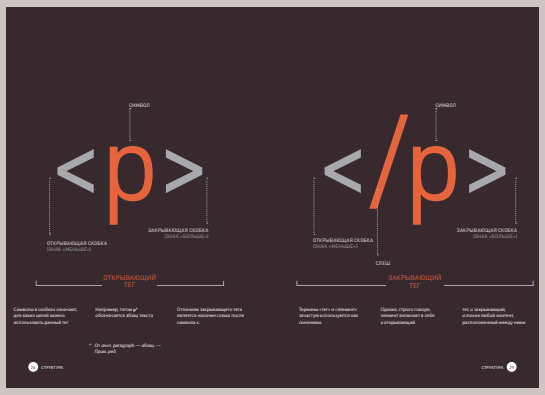
Каждая глава начинается с **Введения**, в котором перечисляются все ее основные темы



На **справочных** страницах представлен основной HTML- и CSS-код; первый выделен голубым цветом, второй — розовым

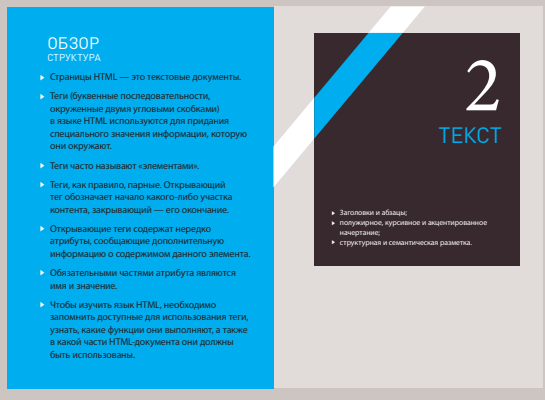


Страницы с **дополнительной информацией** имеют белый фон и предоставляют расширенные сведения по теме, обсуждаемой в главе



Диаграммы и инфографика представлены на страницах с темным фоном — это простой визуальный источник справочной информации по обсуждаемой теме

Разделы с **примерами** объединяют изученные темы и демонстрируют практическое применение изложенной выше информации



В конце каждой главы приведен **Обзор**, на страницах которого напоминается об основных моментах рассмотренных в ней тем

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 (чаще ежегодную) 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

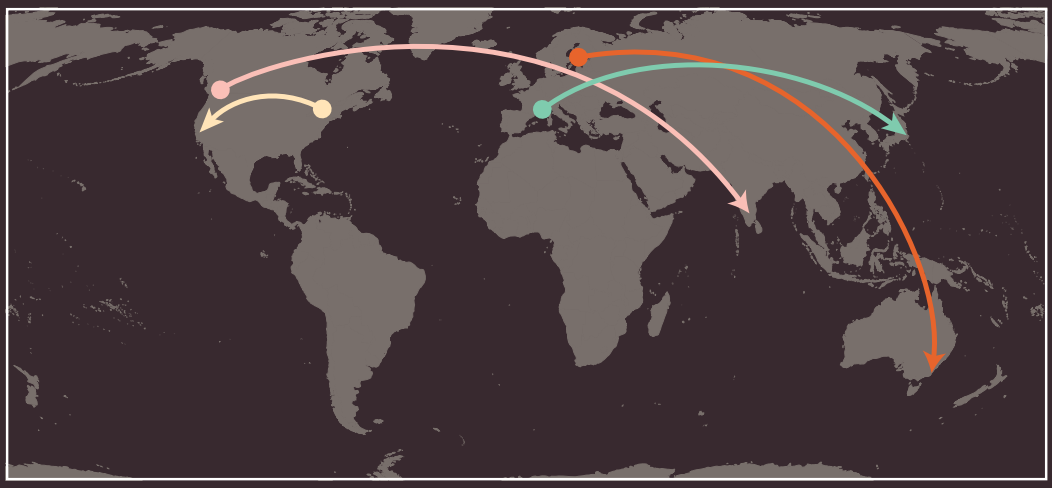
Небольшие сайты обычно верстаются только на HTML и CSS. Более крупные сайты, в частности те, что подвергаются регулярному обновлению и используют системы управления контентом*, а также платформы для блогов и электронной коммерции, зачастую создаются с применением более сложных технологий, функционирующих 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.

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.

Это основной заголовок
Еще один подзаголовок
MacBook Pro

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.

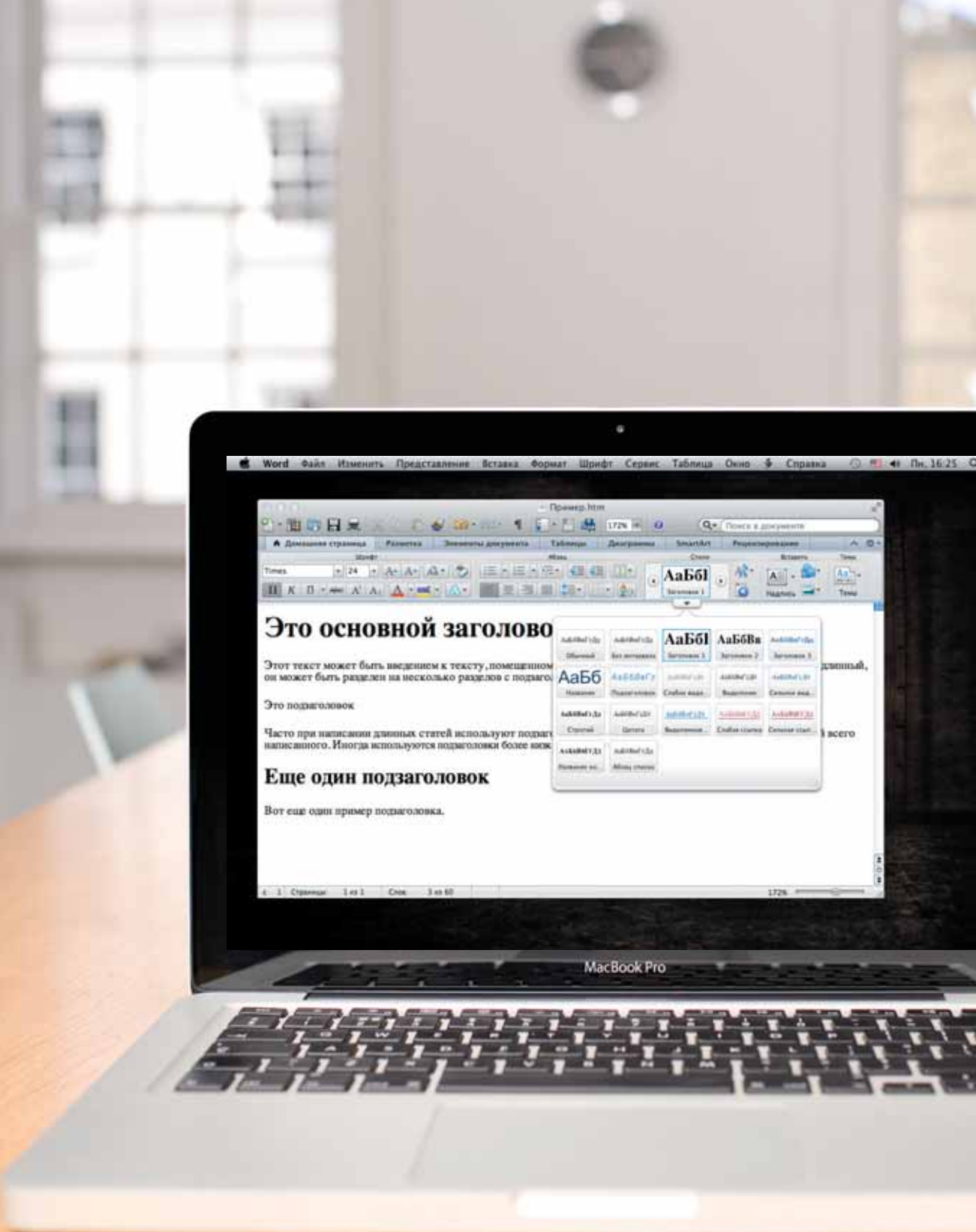
Шар в шаре
Прелесть Шанхая в том, что этот безусловно китайский город не определить как «китайский». То есть не «китайский Париж» и не «китайский Нью-Йорк»
Текст: Дмитрий Губин
Фото: Петер Балобрежский
AURORA
TOTO
GEO
Шар в шаре
Прелесть Шанхая в том, что этот безусловно китайский город не определить как «китайский». То есть не «китайский Париж» и не «китайский Нью-Йорк»
Текст: Дмитрий Губин
Фото: Peter Bialobrzeski / laif
ПОДПИСАТЬСЯ НА ЖУРНАЛ
СВЕЖИЙ НОМЕР
АРКТИКА

СТРУКТУРИРОВАНИЕ ДОКУМЕНТОВ MICROSOFT WORD

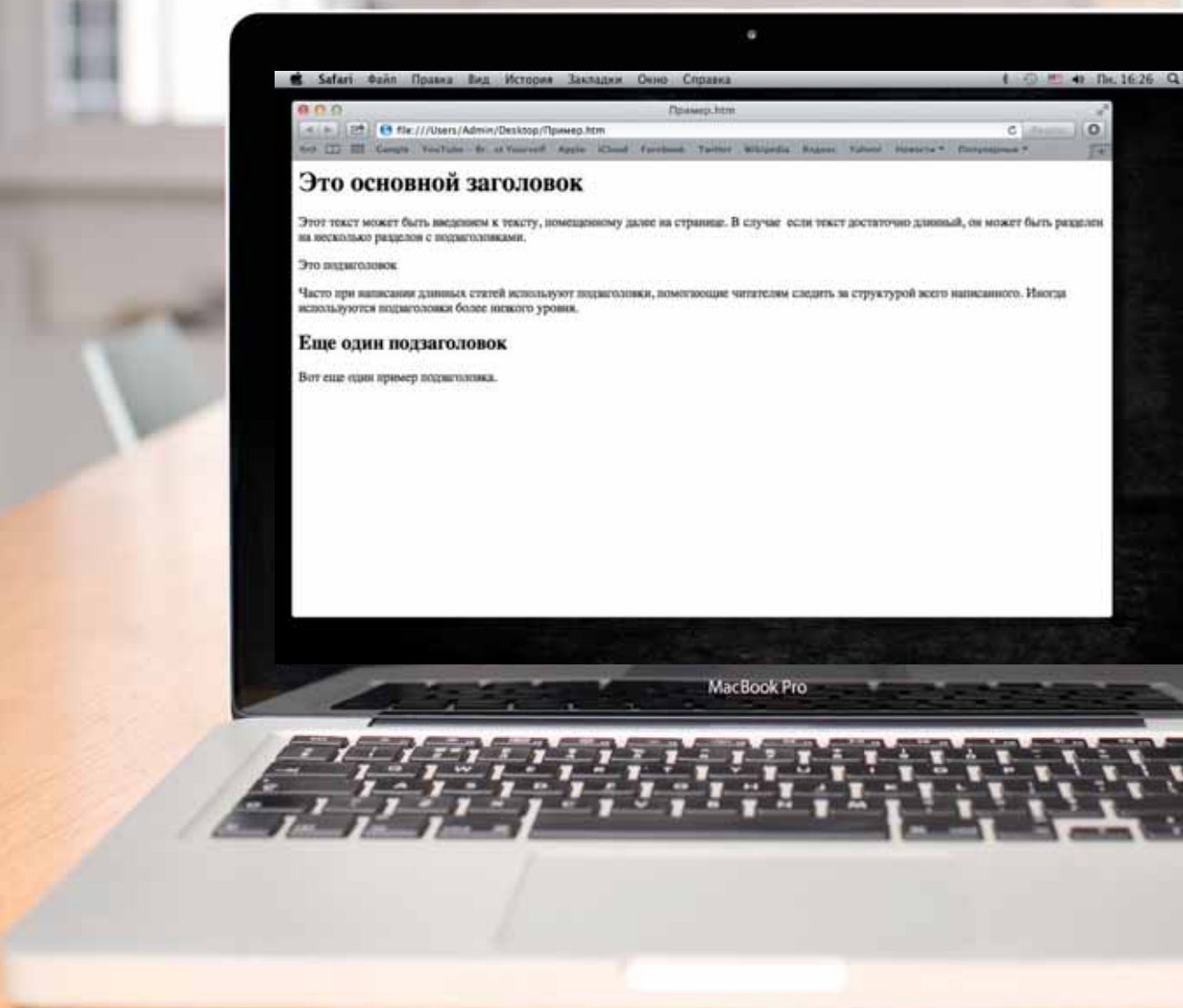
Использование заголовков и подзаголовков при составлении документов различных типов помогает отразить иерархию предоставляемых данных. Например любой документ должен начинаться с заголовка, написанного крупным шрифтом, после которого, как правило, следует введение или резюме с наиболее важной информацией.

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.


Word Файл Изменить Представление Вставка Формат Шрифт Сервис Таблица Окно Справка
Пример.htm
Это основной заголово
Это подзаголовок
Еще один подзаголовок
Вот еще один пример подзаголовка.
MacBook Pro

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



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

```
<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>
```

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 ** ** и ** **.

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

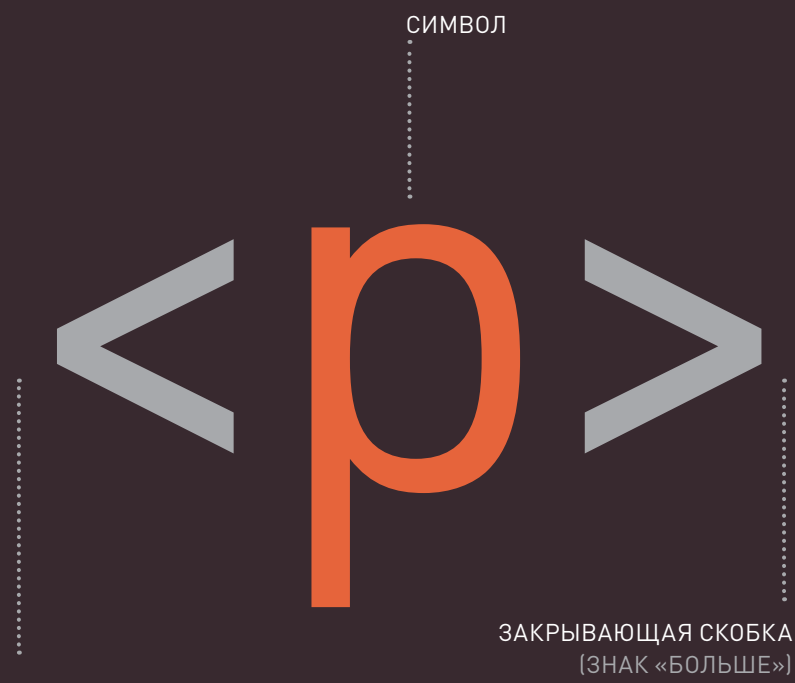
Это еще один абзац текста, помещенный между тегами ** ** и ** **.

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

Еще один абзац текста между тегами ** ** и ** **

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

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



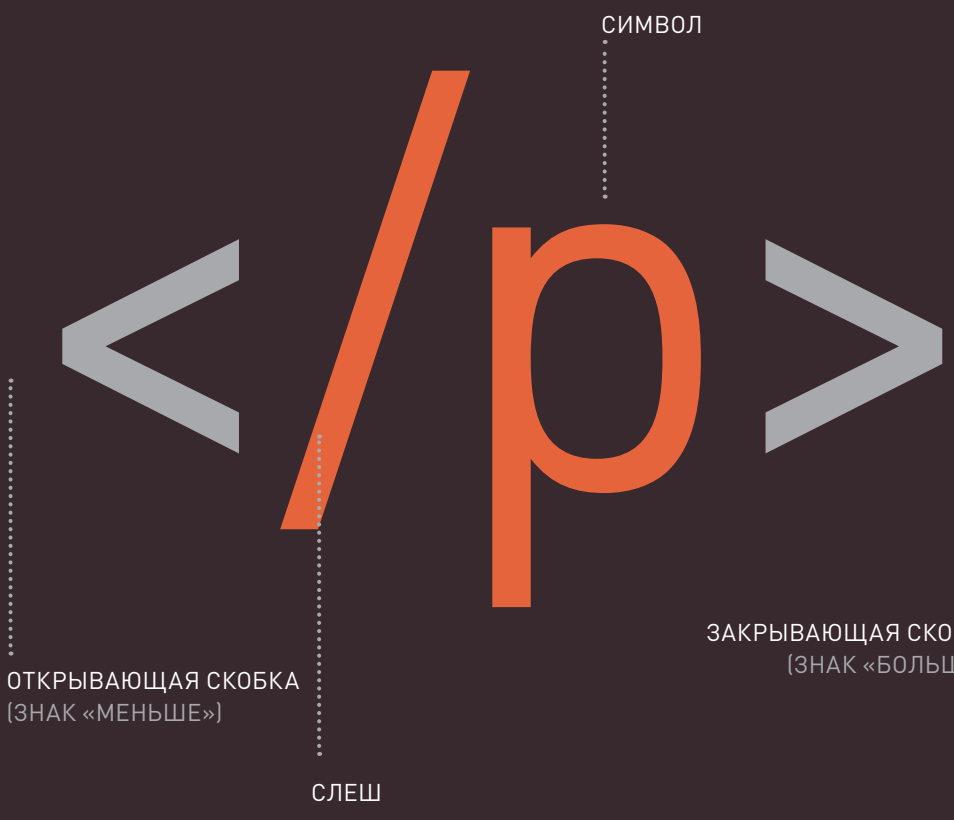
ОТКРЫВАЮЩИЙ ТЕГ

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

Например, тегом **p*** обозначается абзац текста

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

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

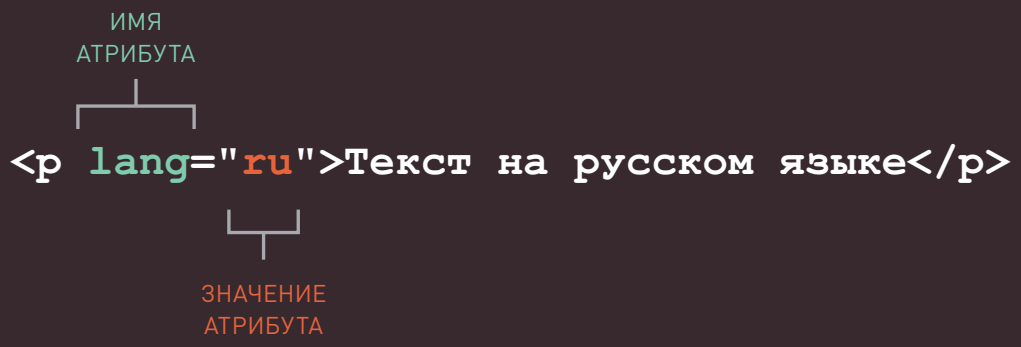


ЗАКРЫВАЮЩИЙ ТЕГ

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>.

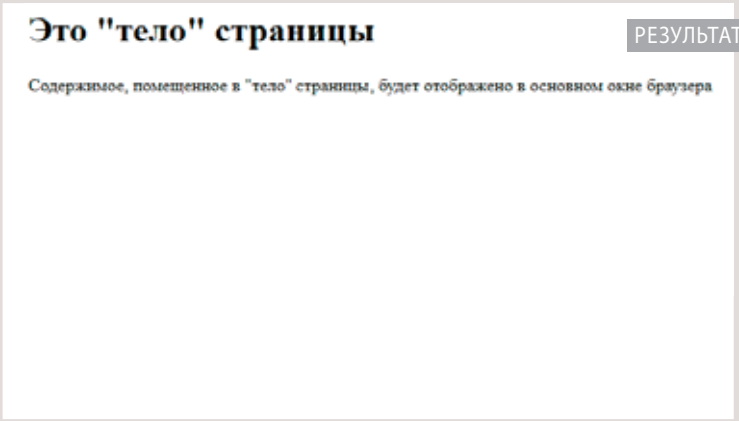
<title>

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

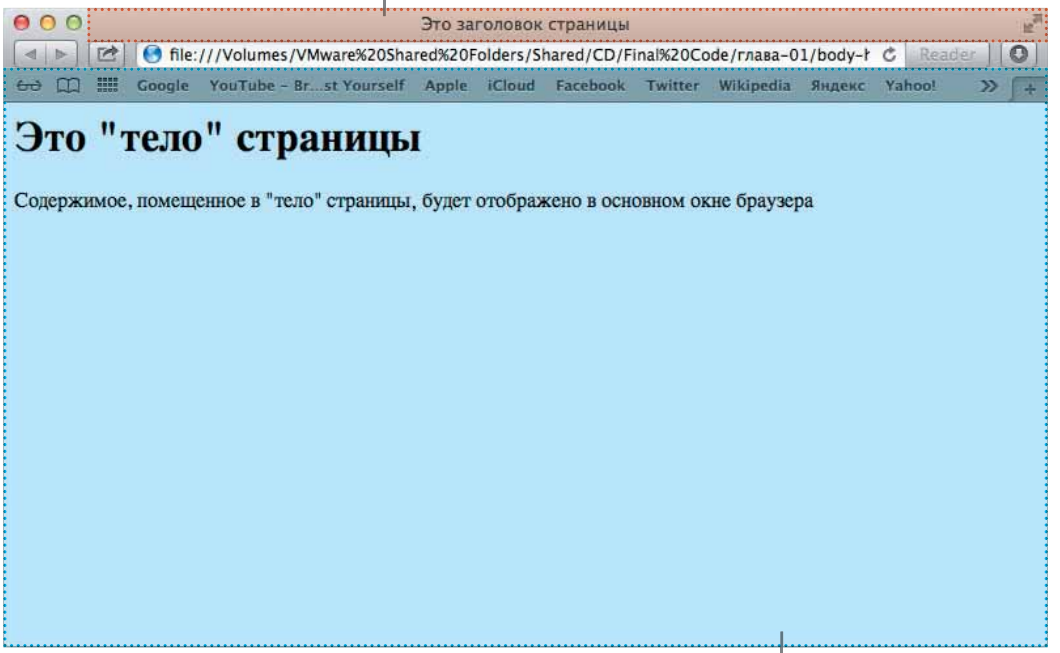
```
<html>
  <head>
    <title>Это заголовок страницы</title>
  </head>
  <body>
    <h1>Это "тело" страницы</h1>
    <p>Содержимое, помещенное в "тело" страницы,
      будет отображено в основном окне браузера.</p>
  </body>
</html>
```

РЕЗУЛЬТАТ



* 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

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).

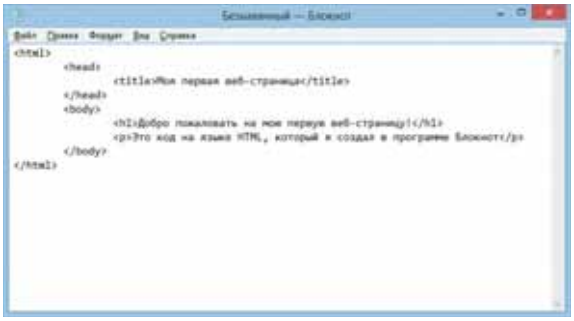
В предыдущих версиях операционной системы Windows, выберите команду меню **Пуск** ⇒ **Все программы** ⇒ **Стандартные** ⇒ **Блокнот** (Start ⇒ All Programs ⇒ Accessories ⇒ Notepad).

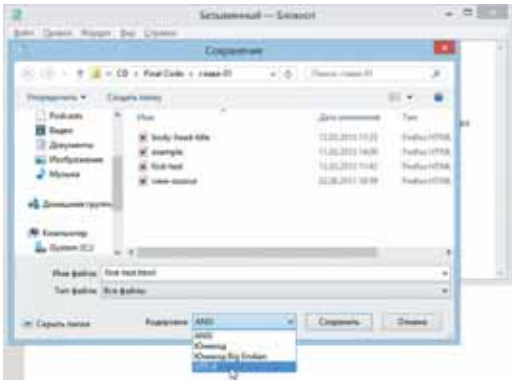
Вы также можете загрузить бесплатный текстовый редактор Notepad++ с сайта notepad-plus-plus.org.

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

1

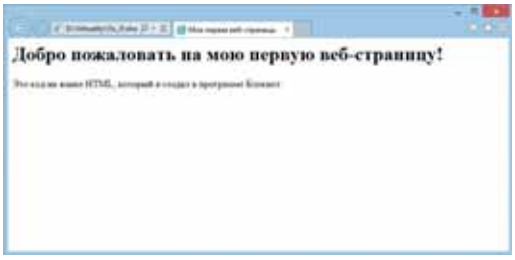
2





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

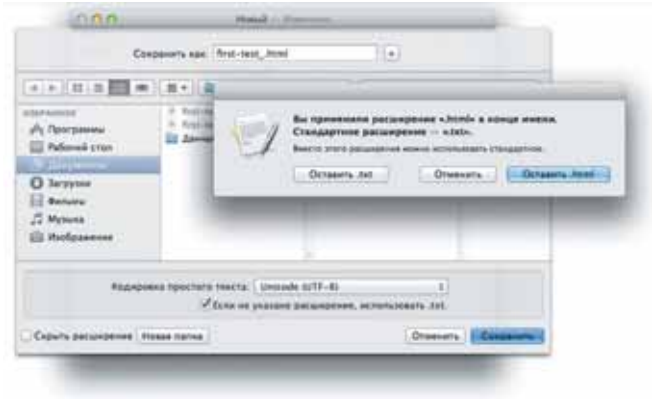
1

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**.

2

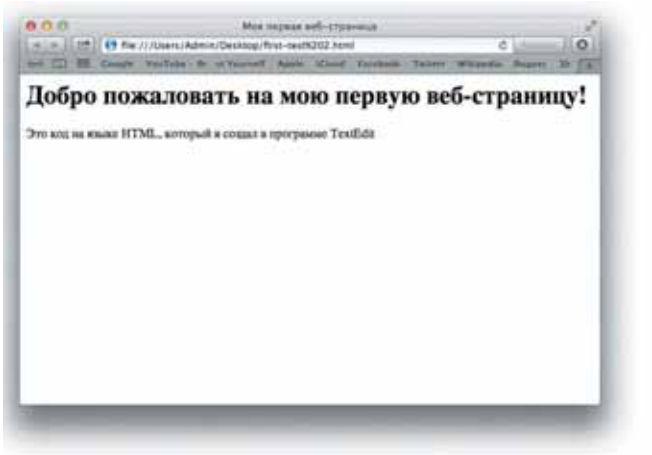
Программа TextEdit по умолчанию создает не простые текстовые, а RTF-документы. Для работы с HTML-кодом, выберите команду меню **Формат** ⇒ **Конвертировать в простой текст** (Format ⇒ Make Plain Text).

3



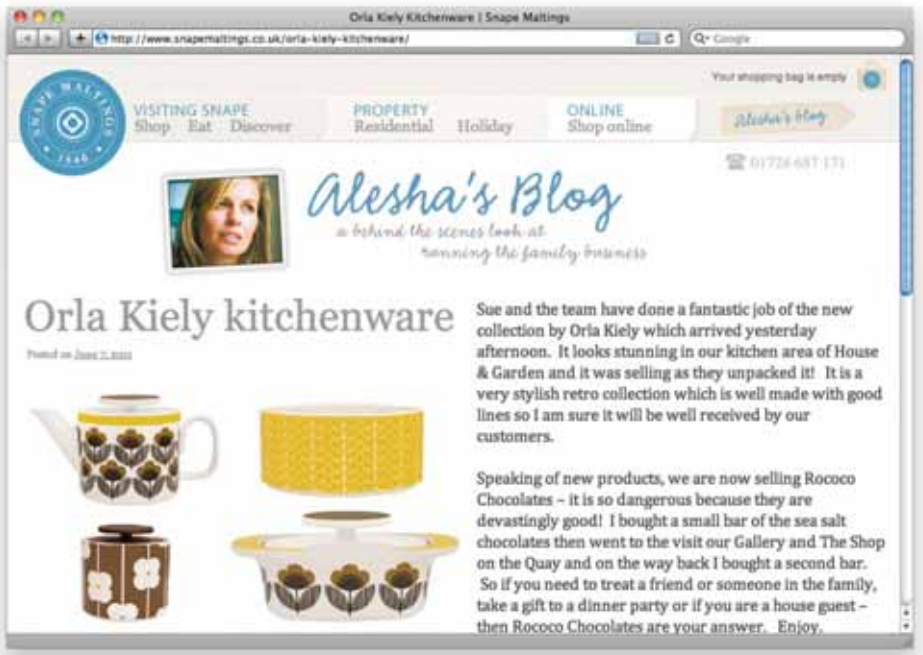
Теперь перейдите в меню **Файл** (File) и выберите команду **Сохранить** (Save). Сохраните файл и запомните его местоположение.
При желании вы можете создать отдельную папку для всех примеров из этой книги. Сохраните данный файл под именем *first-test.html*. Если на экране появится диалоговое окно с 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.

4



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



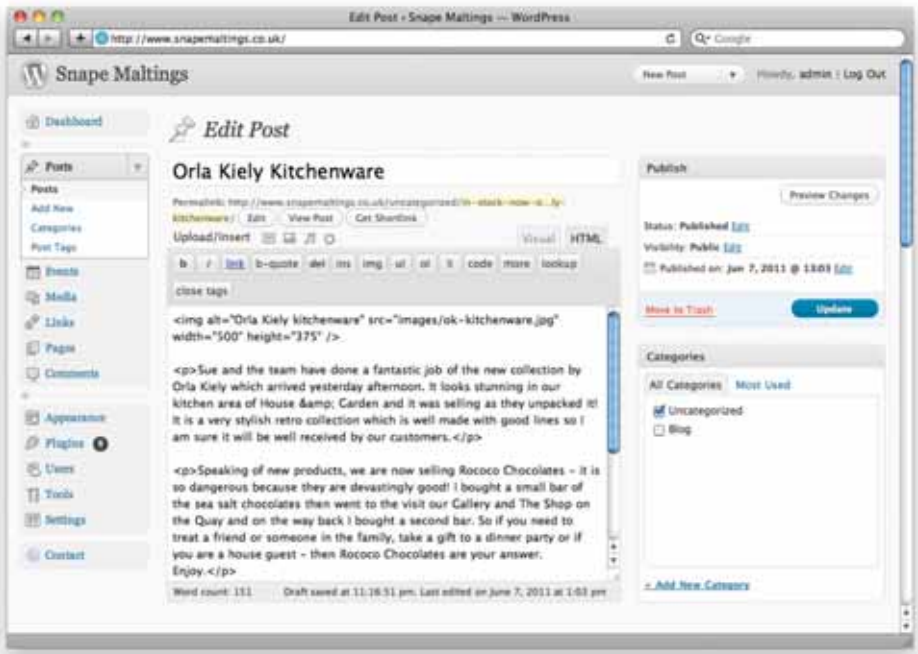
Обычно при работе с системами управления контентом, а также платформами для блогов или электронной коммерции для получения контроля над ними необходимо перейти в специальный административный раздел сайта. Инструменты, предоставляемые в таких разделах, как правило, позволяют редактировать отдельные участки 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 заголовок 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



отдельного продукта. В системах, работа с которыми подразумевает ввод и редактирование больших фрагментов текста, например статей, записей блога или описаний товаров, зачастую имеется встроенный текстовый редактор.

В таких текстовых редакторах, как правило, есть дополнительные элементы управления, наподобие тех, что присутствуют в программах типа Word, позволяющие изменять оформление текста, а также добавлять ссылки или вставлять изображения. Однако закадровая работа таких текстовых редакторов заключается в добавлении необходимого HTML-кода к вводимому вами тексту, наподобие того, что вы уже встречали ранее в этой главе. Многие из таких редакторов также позволяют просмотреть (и отредактировать) созданный ими код. Узнав, как читать и редактировать этот код, вы сможете получить больше контроля над подобными разделами вашего сайта.

В примере на рисунке, приведенном выше, вы можете видеть, что в текстовом редакторе есть закладки для отображения вводимого текста либо в формате HTML-кода, либо так, как его увидят посетители страницы (**Visual**). Другие системы могут иметь специальную кнопку (зачастую с изображением угловых скобок), нажатие на которую позволяет получить доступ к коду. Кроме того, некоторые из систем управления контентом также позволяют редактировать файлы шаблонов. Если вы хотите изменить шаблон, вам необходимо будет обратиться к документации по вашей системе управления контентом, так как шаблоны существенно отличаются от системы к системе. При редактировании файлов шаблонов нужно быть очень внимательным, так как удаление какого-нибудь важного фрагмента кода может привести к прекращению работы всего сайта.

КАК СОЗДАЮТСЯ ДРУГИЕ САЙТЫ

На заре появления веб-страниц одним из самых популярных методов изучения языка HTML был просмотр исходного кода уже существовавших страниц с целью получить подсказки или перенять какие-то технологии.

Сегодня существует огромное количество книг и онлайн-руководств по языку HTML, но вы все же можете при желании просматривать исходный код посещаемых страниц. Чтобы увидеть код, перейдите на желаемую страницу — к примеру, на расположенную по адресу **www.htmlandcssbook.com/code-samples/chapter-01/view-source.html**. Открыв ее, выберите команду меню **Разработка ⇒ Показать источник страницы** (Develop ⇒ Show Page Source). (В вашем браузере это может оказаться команда типа **Вид ⇒ Исходный код** (View ⇒ Source), название данного пункта меню будет зависеть от используемого браузера.)

После этого на экране появится новое окно, содержащее исходный код, с помощью которого была создана страница.

Результат выполнения этой операции вы можете видеть ниже. Страница, открытая в браузере, на рисунке показана сверху, а код — ниже.

Поначалу код может показаться сложным, но не стоит отчаиваться: к окончанию чтения следующей главы вы уже станете понимать его.

Все примеры из книги размещены на прилагающемся диске, и вы можете пользоваться этим нехитрым методом, чтобы просмотреть исходный код любого из файлов примеров.

Safari Файл Правка Вид История Закладки Разработка Окно Справка
View the Source of This Page
www.htmlandcssbook.com/code-samples/chapter-01/v
How to View the Source of This Page
If you go to your web browser's View menu, you will see an option that allows you to see the source code behind a web page.
When the web was first taking off this was a very popular way for people to learn how web pages had been built.
Today, people still often view the source of pages to learn how a web page has been built.
Веб-инспектор — www.htmlandcssbook.com — view-source.html
view-source.html
Исходный код
MacBook Pro

ОБЗОР
СТРУКТУРА

- Страницы HTML — это текстовые документы.
- Теги (буквенные последовательности, окруженные двумя угловыми скобками) в языке HTML используются для придания специального значения информации, которую они окружают.
- Теги часто называют «элементами».
- Теги, как правило, парные. Открывающий тег обозначает начало какого-либо участка контента, закрывающий — его окончание.
- Открывающие теги содержат нередко атрибуты, сообщающие дополнительную информацию о содержимом данного элемента.
- Обязательными частями атрибута являются имя и значение.
- Чтобы изучить язык HTML, необходимо запомнить доступные для использования теги, узнать, какие функции они выполняют, а также в какой части HTML-документа они должны быть использованы.

2

ТЕКСТ

- Заголовки и абзацы
- Полужирное, курсивное и акцентированное начертание
- Структурная и семантическая разметка

При верстке веб-страницы текст размечают тегами. Теги добавляют тексту дополнительное значение и позволяют браузерам правильно отображать структуру страницы.

В этой главе мы заострим внимание на том, как добавить разметку к тексту, выводимому на страницы. Вы изучите:

- **структурную разметку** — элементы, которые вы можете использовать для описания и заголовков и абзацев;
- **семантическую разметку**, предоставляющую дополнительную информацию, например на какое слово или словосочетание в предложении делается логическое ударение, является ли какой-то написанный текст цитатой, какова расшифровка акронима и т.д.

POUDRE D'IRIS
G. DUPONT
SAVON SURFIN
VIOLETTE
HELLO! ISN'T IT A LOVELY DAY?
DISTILLERIE DE MONTMORILLON
AMER LÉON
AU CITRON – SANS ALCOOL
GIRAUD-DUQUESNE
Роман без названия
Глава 1
MacBook Pro
Copperplate

ЗАГОЛОВКИ

<h1>
<h2>
<h3>
<h4>
<h5>
<h6>

В языке HTML существует шесть уровней заголовков:

`<h1>` используется для основных заголовков;

`с <h2> по <h6>` — для подзаголовков.

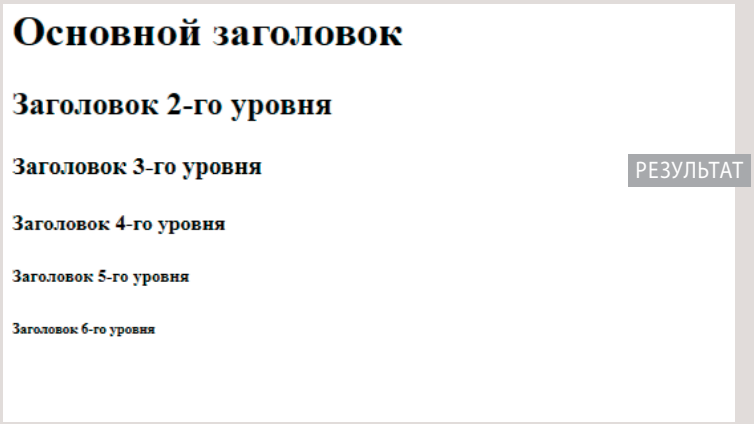
Если текст разделен на фрагменты более глубокого уровня, то для них используются заголовки `<h3>` и ниже.

Браузеры отображают содержимое элементов-заголовков шрифтом разного размера. Текст заголовка **<h1>** выводится самым крупным шрифтом, текст заголовка **<h6>** — самым мелким. Кегль шрифта, используемый разными браузерами для отображения заголовков разного уровня, незначительно отличается. Кроме того, пользователи также могут отрегулировать кегль шрифта отображаемого текста. Вы узнаете, как устанавливать кегль, цвет и гарнитуру текста, когда мы перейдем к изучению каскадных таблиц стилей (CSS).

глава-02/headings.html HTML

```
<h1>Основной заголовок</h1>
<h2>Заголовок 2-го уровня</h2>
<h3>Заголовок 3-го уровня</h3>
<h4>Заголовок 4-го уровня</h4>
<h5>Заголовок 5-го уровня</h5>
<h6>Заголовок 6-го уровня</h6>
```

РЕЗУЛЬТАТ



АБЗАЦЫ

HTML **глава-02/paragraphs.html**

```
<p> Абзац состоит из одного или нескольких
    предложений, формирующих самодостаточную часть
    какого-либо высказывания. Абзац начинается
    с красной строки.</p>
<p> Текст, разделенный на части, легче понимать.
    Например текст книги может быть разделен на
    главы. В главах могут быть подзаголовки. Под
    каждым из подзаголовков будет следовать текст,
    состоящий из одного или нескольких абзацев.</p>
```

РЕЗУЛЬТАТ

Абзац состоит из одного или нескольких предложений, формирующих самодостаточную часть какого-либо высказывания. Абзац начинается с красной строки.

Текст, разделенный на части,легче понимать. Например текст книги может быть разделен на главы. В главах могут быть подзаголовки. Под каждым из подзаголовков будет следовать текст, состоящий из одного или нескольких абзацев.

Для создания абзаца заключите текст в открывающий и закрывающий теги.

По умолчанию браузер автоматически отобразит каждый абзац с новой строки, при этом абзацы будут несколько отстоять друг от друга.

ПОЛУЖИРНОЕ И КУРСИВНОЕ НАЧЕРТАНИЕ

Чтобы выделить текст полужирным начертанием шрифта, нужно поместить его между тегами и . Элемент также обозначает фрагмент текста (например, ключевые слова), внешний вид которого будет отличаться от основного текста абзаца. При этом использование данного элемента не добавляет дополнительных значений к тексту.

глава-02/bold.html HTML

```
<p>Вот так можно вывести текст с <b>полужирным</b>
  начертанием шрифта.</p>
<p>В аннотации основные характеристики какого-либо
  продукта могут быть выделены <b>полужирным</b>
  начертанием шрифта.
```

РЕЗУЛЬТАТ

Вот так можно вывести текст с **полужирным** начертанием шрифта.

В аннотации основные характеристики какого-либо продукта могут быть выделены **полужирным** начертанием шрифта.

Чтобы отобразить текст с курсивным начертанием шрифта, его следует поместить между тегами и .
Элемент определяет фрагмент текста, отличающийся от основного текста, например технические термины, иностранные слова и прочие виды включений в текст, которые принято выделять курсивом.

глава-02/italic.html HTML

```
<p>Вот так можно отобразить текст <i>курсивом</i>.
  <p>Это сорт картофеля <i>Solanum tuberosum</i>.</p>
<p>Капитан Кук приплыл в Австралию на корабле
  <i>Индевор</i>.</p>
```

РЕЗУЛЬТАТ

Вот так можно отобразить текст *курсивом*.

Это сорт картофеля *Solanum tuberosum*.

Капитан Кук приплыл в Австралию на корабле *Индевор*.

ПОДСТРОЧНЫЕ И НАДСТРОЧНЫЕ ЗНАКИ

HTML **глава-02/superscript-and-subscript.html**

```
<p>Сейчас мы изучим формулу E=mc<sup>2</sup>.</p>
  <p>В 2009 году концентрация CO<sub>2</sub>
  в атмосферном воздухе возросла на 2 ppm.</p>
```

РЕЗУЛЬТАТ

Сейчас мы изучим формулу $E=mc^2$.

В 2009 году концентрация CO_2 в атмосферном воздухе возросла на 2 ppm.

<sup>

Элемент <sup> используется для выделения символов, которые должны быть отображены как надстрочные, к примеру, ряд математических понятий, таких как возведение в степень (2^2 и т.п.).

<sub>

Элемент <sub> используется для отображения символов как подстрочных. Подстрочные символы, как правило, используются в химических формулах, таких как H_2O.

ПРОБЕЛЫ

Верстальщики веб-страниц часто добавляют дополнительные пробелы в код или даже помещают некоторые элементы на новые строки с целью повышения удобочитаемости кода.

Когда браузер встречается с двумя или более пробелами, идущими друг за другом, он отображает только один. Аналогично он поступает и с переносом строки, воспринимая его как единичный символ пробела. Данное свойство браузеров известно как **«сворачивание пробелов»**.

Довольно часто можно столкнуться с тем, что создатели веб-страниц пользуются сворачиванием пробелов, чтобы расставить необходимые отступы в исходном коде страниц для повышения удобочитаемости.

глава-02/white-space.html HTML

```
<p>Луна понемногу отдаляется от Земли.</p>
<p>Луна          понемногу отдаляется от Земли.</p>
<p>Луна понемногу отдаляется от

      Земли.</p>
```

РЕЗУЛЬТАТ

Луна понемногу отдаляется от Земли.

Луна понемногу отдаляется от Земли.

Луна понемногу отдаляется от Земли.

ПЕРЕВОД СТРОК И ГОРИЗОНТАЛЬНЫЕ ЛИНИИ

HTML **глава-02/line-breaks.html**

```
<p>Земля<br />становится тяжелее
день ото дня<br />из-за падающей на нее космической
пыли.</p>
```

РЕЗУЛЬТАТ

Земля
становится тяжелее день ото дня
из-за падающей на нее космической пыли.

Как вы уже видели раньше, браузер автоматически начинает каждый абзац с новой строки, однако если вы хотите добавить перевод строки в пределах какого-либо абзаца, то вам понадобится вставить специальный тег — `<br />`.

HTML **глава-02/horizontal-rules.html**

```
<p> Венера — единственная планета, вращающаяся
    по часовой стрелке.</p>
<hr />
<p> Юпитер больше, чем все остальные планеты вместе
    взятые.</p>
```

РЕЗУЛЬТАТ

Венера — единственная планета, вращающаяся по часовой стрелке.

Юпитер больше, чем все остальные планеты вместе взятые.

<hr />

Чтобы визуально разделить темы книги или сцены пьесы, с помощью тега `<hr />` вы можете добавить в текст горизонтальную линию.

В языке HTML существует несколько элементов, не содержащих текста между открывающим и закрывающим тегами. Их принято называть **пустыми элементами.** Написание таких тегов отличается от стандартного. Обычно пустой элемент состоит только из одного тега, перед правой, закрывающей скобкой которого вставляются пробел и слеш. Некоторые верстальщики веб-страниц опускают пробел и слеш, хотя их указание и рекомендуется для следования спецификации.

ВИЗУАЛЬНЫЕ РЕДАКТОРЫ И РЕЖИМ ПРОСМОТРА КОДА

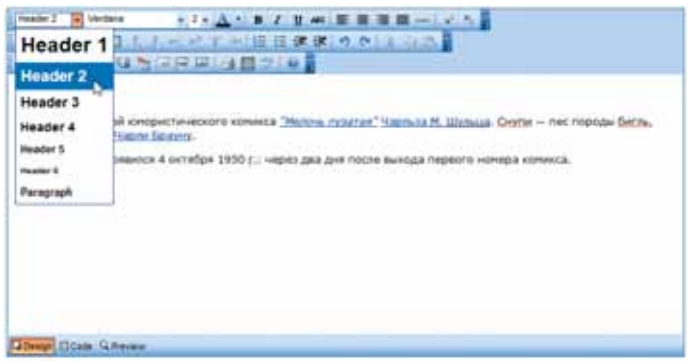
Системы управления контентом и редакторы HTML, такие как Adobe Dreamweaver, обычно предоставляют два способа отображения создаваемой страницы: режим визуального редактора и режим просмотра кода.

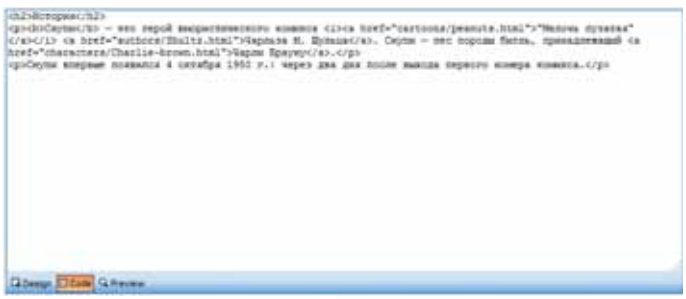
Визуальные редакторы часто напоминают по виду текстовые редакторы. Несмотря на то что редакторы могут незначительно отличаться друг от друга, все они имеют несколько общих черт, позволяющих управлять внешним видом и представлением текста.

- Заголовки создаются посредством выделения строки текста и выбора нужного уровня заголовка из раскрывающегося списка.
- Акцентирование курсивом или полужирным шрифтом создается выделением нужного текста и щелчком мышью по кнопке **Ж** (B) или **К** (I).
- Новые абзацы создаются нажатием клавиши **Enter/Return**.
- Перевод строки можно создать нажатием сочетания клавиш **Shift+Return**.
- Горизонтальные линии создаются с помощью кнопки с нарисованной на ней чертой.

При копировании текста из программ, позволяющих выполнять его форматирование (таких как Microsoft Word), в визуальный редактор компьютер может вставить дополнительную разметку. Чтобы избежать этого, всегда сначала вставляйте скопированный текст в простой текстовый редактор (такой как Блокнот (Notepad) в операционной системе Windows или TextEdit в OS X) и лишь затем заново копируйте его и вставляйте в визуальный редактор.

Режим просмотра кода позволяет отобразить HTML-теги, созданные визуальным редактором, с тем чтобы вы могли вручную отредактировать его либо вставить собственный код. Как правило, для перехода в этот режим нужно нажать кнопку с надписью HTML или угловыми скобками. Для повышения удобочитаемости кода визуальный редактор также может добавлять в него пробелы и отступы.





СЕМАНТИЧЕСКАЯ РАЗМЕТКА

В языке HTML существует ряд текстовых элементов, не вносящих изменения в структуру веб-страницы, но добавляющих дополнительную информацию. Эти теги принято называть семантической (то есть смысловой) разметкой.

В последней части этой главы вы познакомитесь еще с несколькими элементами, которые помогут вам с оформлением текста на веб-странице, — например с элементом `<em>`, позволяющим акцентировать внимание на словах, на которые падает логическое ударение, и с элементом `<blockquote>`, обозначающим, что включенный в него текст является цитатой. Зачастую браузеры используют особые стили отображения текста, помещенного в такие элементы. Например содержимое тега `<em>` отображается курсивом, тогда как текст в пределах элемента `<blockquote>` выделяется отступами. Впрочем, не стоит использовать эти элементы для изменения оформления отображаемого текста: они используются лишь для более точного описания контента веб-страниц.

Вы можете использовать элементы этой группы, чтобы позволить дополнительному программному обеспечению, например программам экранного доступа, пользоваться этой информацией. К примеру, голос программы экранного доступа может быть усилен на фрагменте текста внутри элемента `<em>` или поисковая система может сделать пометку о том, что на вашей странице опубликована цитата, если вы используете элемент `<blockquote>`.

ВАЖНАЯ ИНФОРМАЦИЯ И ЛОГИЧЕСКОЕ УДАРЕНИЕ

Элемент используется для обозначения того, что текст, помещенный в него, имеет высокую степень важности. Например текст этого элемента может быть интонационно выделен при чтении программой экранного доступа.
По умолчанию браузеры отображают содержимое элемента шрифтом с полужирным начертанием.

глава-02/strong.html HTML

```
<p><strong>Будьте осторожны:</strong>
    В данной местности орудуют воры-карманники.</p>
<p> Данная игрушка содержит большое количество мелких
    деталей и <strong>не предназначена для детей
    младше пяти лет.</strong></p>
```

РЕЗУЛЬТАТ

Будьте осторожны: В данной местности орудуют воры-карманники.

Данная игрушка содержит большое количество мелких деталей и **не предназначена для детей младше пяти лет.**

Элемент используется для обозначения логического ударения, которое несколько изменяет значение всего предложения.
По умолчанию браузеры отображают содержимое элемента шрифтом с курсивным начертанием.

глава-02/emphasis.html HTML

```
<p>Я <em>думаю</em>, Анна была первой.</p>
<p>Я думаю, <em>Анна</em> была первой.</p>
<p>Я думаю, Анна была <em>первой</em>.</p>
```

РЕЗУЛЬТАТ

Я *думаю*, Анна была первой.

Я думаю, *Анна* была первой.

Я думаю, Анна была *первой*.

ЦИТАТЫ

HTML **глава-02/quotations.html**

```
<blockquote cite="http://ru.wikipedia.org/
wiki/Винни-Пух">
<p>— Интересно, что это так бумкнуло? Не
мог же я один наделать столько шума. И где,
интересно знать, мой воздушный шарик?
И откуда, интересно, взялась эта тряпочка?
</p>
</blockquote>
<p>Как сказал А. А. Милн, <q> некоторые
люди говорят с животными. И лишь немногие
слушают. Вот в чем проблема</q>.</p>
```

РЕЗУЛЬТАТ

— Интересно, что это так бумкнуло? Не мог же я один наделать столько шума. И где, интересно знать, мой воздушный шарик? И откуда, интересно, взялась эта тряпочка?

Как сказал А. А. Милн, "некоторые люди говорят с животными. И лишь немногие слушают. Вот в чем проблема."

Для создания цитат в языке HTML используются два элемента:

<blockquote>

Элемент <blockquote> используется для отображения длинных цитат, занимающих целый абзац. Обратите внимание, что внутри элемента <blockquote> необходимо использовать и элемент <p>. В большинстве своем браузеры склонны к выделению содержимого элемента <blockquote> отступами, однако этот элемент не стоит использовать только лишь для того, чтобы создать фрагмент текста с отступом: лучше реализовать такое с помощью каскадных таблиц стилей (CSS).

<q>

Элемент <q> принято использовать для более коротких цитат, помещающихся внутри абзаца текста. Изначально предполагалось, что браузеры будут заключать содержимое элемента <q> в кавычки, однако Internet Explorer этого не делает, потому многие избегают использовать данный элемент.

Для обоих элементов может быть указан атрибут **cite***, указывающий, с какого сайта позаимствована та или иная цитата. Значение этого атрибута — URL-адрес сайта, содержащего более подробную информацию об источнике цитирования.

* В пер. с англ. — *ссылаться, цитировать. — Примеч. ред.*

АББРЕВИАТУРЫ И АКРОНИМЫ

<abbr>

При употреблении в тексте какого-либо акронима или аббревиатуры следует воспользоваться тегом `<abbr>`. В открывающем теге можно использовать атрибут `title` для указания расшифровки сокращения.

В языке HTML4 для описания акронимов был реализован специальный элемент `<acronym>`. Для отображения расшифровки акронима использовался атрибут `title` (аналогичный атрибуту элемента `<abbr>`). В языке HTML5 и для аббревиатур, и для акронимов используется тег `<abbr>`.

глава-02/abbreviations.html HTML

```
<p><abbr title="Профессор">Пр.</abbr>
    Стивен Хокинг — физик-теоретик и космолог.</p>
<p><acronym title="Национальное агентство
    по аэронавтике и исследованию космического
    пространства">NASA</acronym> проводит несколько
    невероятных космических экспериментов.</p>
```

РЕЗУЛЬТАТ

Пр. Стивен Хокинг — физик-теоретик и космолог.

NASA проводит несколько невероятных космических экспериментов.

Национальное агентство по аэронавтике и исследованию космического пространства

ИСТОЧНИКИ И ОПРЕДЕЛЕНИЯ

HTML **глава-02/citations.html**

```
<p> По всему миру было продано более десяти миллионов
    копий книги <cite>Краткая история времени</cite>
    Стивена Хокинга.</p>
```

РЕЗУЛЬТАТ

По всему миру было продано более десяти миллионов копий книги *Краткая история времени* Стивена Хокинга.

<cite>

Элемент <cite> может быть использован при оформлении ссылки на какой-то первоисточник, например на книгу, кинофильм или исследовательскую работу, для обозначения упоминаемого источника.
В спецификации HTML5 не рекомендуется использовать элемент <cite> с именами персоналий, однако это вполне допускается в HTML4, потому, скорей всего, большинство веб-дизайнеров так и продолжат использовать данный элемент в этих целях.
Браузеры отображают содержимое элемента <cite> шрифтом с курсивным начертанием.

HTML **глава-02/definitions.html**

```
<p><dfn>Черная дыра</dfn> — это область
    в пространстве-времени, гравитационное притяжение
    которой настолько велико, что покинуть её не могут
    даже объекты, движущиеся со скоростью света (в том
    числе и кванты самого света).</p>
```

РЕЗУЛЬТАТ

Черная дыра — это область в пространстве-времени, гравитационное притяжение которой настолько велико, что покинуть её не могут даже объекты, движущиеся со скоростью света (в том числе и кванты самого света).

<dfn>

Объяснение нового термина (научного концепта или профессионального жаргонизма) в тексте называется «определением».
Элемент <dfn> используется для обозначения определения какого-либо нового термина.
Некоторые браузеры отображают содержимое элемента <dfn> шрифтом с курсивным начертанием. В браузерах Safari и Chrome какие-либо изменения шрифта отсутствуют.

ИНФОРМАЦИЯ О ВЕБ-ДИЗАЙНЕРЕ

<address>

Элемент <address> имеет довольно специфическое назначение: он содержит контактные данные создателя веб-страницы.

Он может (но не должен) содержать почтовый адрес, а также адрес электронной почты или номер телефона.

Большинство браузеров отображают содержимое элемента <address> шрифтом с курсивным начертанием.

Возможно, вас также заинтересует так называемый микроформат hCard, позволяющий добавлять на веб-страницу контактную информацию.

ПРИМЕЧАНИЕ

На странице **ru.wikipedia.org/wiki/HCard** вы можете получить дополнительную информацию о микроформате hCard.

глава-02/address.html HTML

```
<address>
  <p> <a href="mailto:email@mywebsite.ru">
      email@mywebsite.ru</a></p>
  <p> г. Москва, ул. Долгоруковская, д. 25.</p>
</address>
```

РЕЗУЛЬТАТ

email@mywebsite.ru

г. Москва, ул. Долгоруковская, д. 25.

ИСПРАВЛЕНИЯ КОНТЕНТА

HTML **глава-02/insert-and-delete.html**

```
<p> Это была <del>худшая</del> <ins>лучшая</ins>
    из ее идей</p>
```

РЕЗУЛЬТАТ

Это была ~~худшая~~ лучшая из ее идей

<ins>
<del>

Элемент <ins> может быть использован для отображения контента, добавленного в документ, в то время как <del> — для отображения удаленного контента.
Содержимое элемента <ins>, как правило, выделяется подчеркиванием, тогда как элемента <del> — зачеркиванием.

HTML **глава-02/strikethrough.html**

```
<p>Ноутбук Samsung:</p>
<p><s>Старая цена: 31 900 руб.</s></p>
<p>Новая цена: всего 11 999 руб.</p>
```

РЕЗУЛЬТАТ

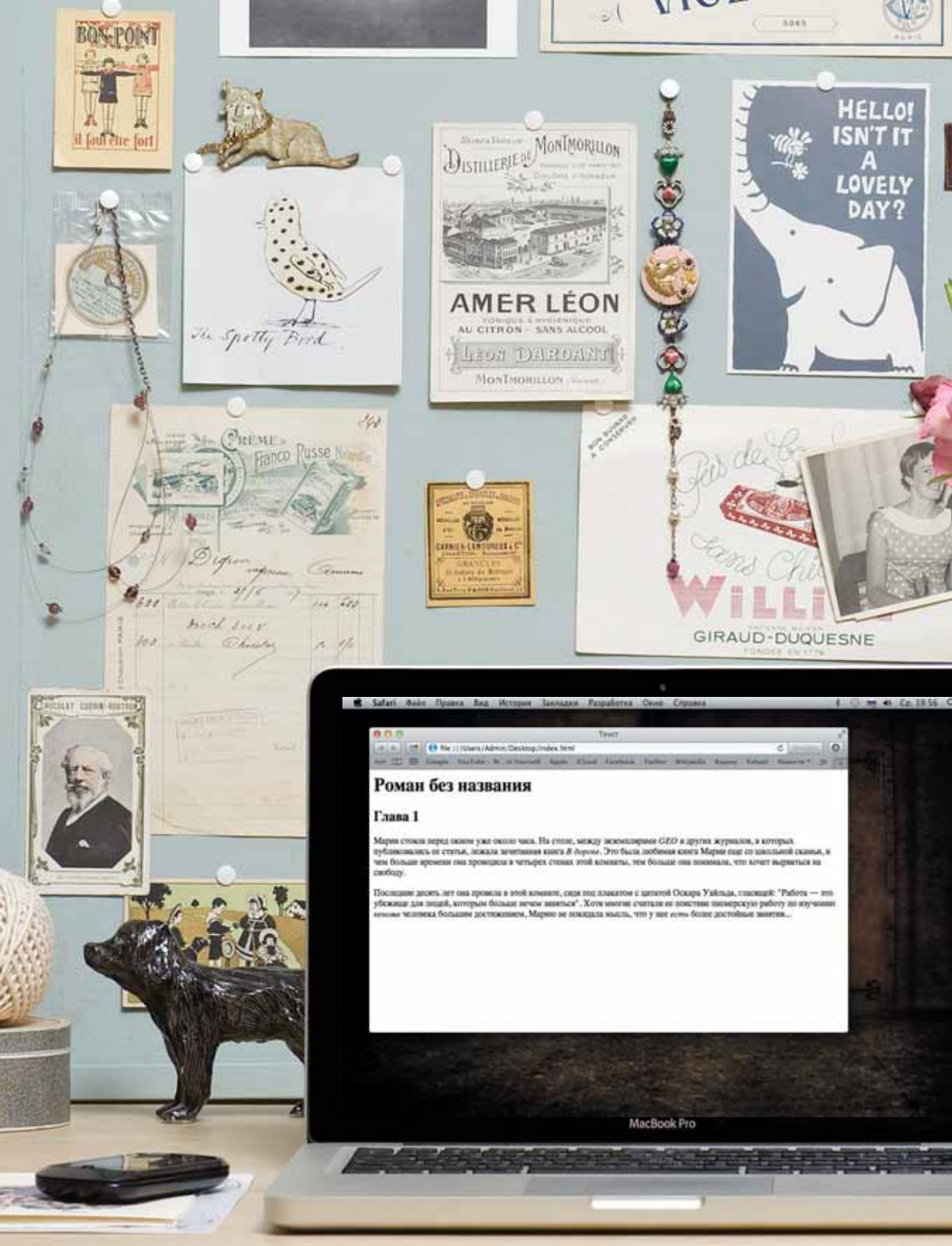
Ноутбук Samsung:

~~Старая цена: 31900 руб.~~

Новая цена: всего 11999 руб.

<s>

Элемент <s> позволяет обозначить, что какая-то информация ошибочна или устарела (но при этом ее не следует убирать со страницы).
Как правило, содержимое элемента <s> будет выведено на экран перечеркнутым.
В старых версиях языка HTML использовался элемент <u> для вывода текста с подчеркиванием, однако он постепенно уходит в прошлое.


BON POINT
il faut être fort
The Spotty Bird
DISTILLERIE DE MONTMORILLON
AMER LÉON
AU CITRON - SANS ALCOOL
LÉON DARDANT
MONTMORILLON
HELLO! ISN'T IT A LOVELY DAY?
CRÈME Franco Russe
GIRAUD-DUQUESNE
Текст
file:///Users/Admin/Desktop/Index.html
Роман без названия
Глава 1
Мария стояла перед окном уже около часа. На столе, между экземпляром GEO и других журналов, в которых публиковались ее статьи, лежала зачитанная книга В дороге. Это была любимая книга Марии еще со школьной скамьи, и чем больше времени она проводила в четырех стенах этой комнаты, тем больше она понимала, что хочет вырваться на свободу.
Последние десять лет она провела в этой комнате, сидя под плакатом с цитатой Оскара Уайльда, гласящей: "Работа — это убежище для людей, которым больше нечем заняться". Хотя многие считали ее довольно плодотворную работу по изучению человека большим достижением, Марию не покидала мысль, что у нее есть более достойные занятия...
MacBook Pro

ПРИМЕР
ТЕКСТ

Это очень простая HTML-страница, демонстрирующая разметку текста.

Структурная разметка текста включает такие элементы как <h1>, <h2> и <p>.

Семантическая информация представлена в таких элементах, как <cite> и

```
<html>
  <head>
    <title> Текст</title>
  </head>
  <body>
    <h1>  Роман без названия</h1>
    <h2>  Глава 1</h2>
    <p> Мария стояла перед окном уже около часа.
        На столе, между экземплярами <i>"GEO"</i>
        и других журналов, в которых публиковались
        ее статьи, лежала зачитанная книга
        <cite>"В дороге"</cite>. Это была любимая книга
        Марии еще со школьной скамьи, и чем больше
        времени она проводила в четырех стенах этой
        комнаты, тем больше она понимала, что хочет
        вырваться на свободу.</p>
    <p> Последние десять лет она провела в этой комнате,
        сидя под плакатом с цитатой Оскара Уайльда,
        гласящей: <q>Работа — это убежище для людей,
        которым больше нечем заняться</q>. Хотя многие
        считали ее поистине пионерскую работу по изучению
        <dfn>генома</dfn> человека большим достижением,
        Марию не покидала мысль, что у нее <em>есть</em>
        более достойные занятия...</p>
  </body>
</html>
```

ОБЗОР
ТЕКСТ

- HTML-элементы используются для описания структуры страницы (например, заголовки, подзаголовки и абзацы).
- Они также могут содержать семантическую (смысловую) информацию (например, в какой позиции нужно поставить логическое ударение, какова расшифровка использованных в тексте акронимов, что данный фрагмент текста является цитатой и т.д.).

3

СПИСКИ

- Упорядоченные списки
- Неупорядоченные списки
- Списки определений

Существует множество ситуаций, когда необходимо пользоваться списками. В языке HTML существует три их разновидности.

- **Упорядоченные списки** — это списки, каждый элемент которых имеет порядковый номер. Например в качестве такого списка можно представить рецепт какого-либо блюда, шаги которого необходимо выполнять в обозначенной последовательности, либо договор, каждое положение которого необходимо обозначить порядковым номером.
- **Неупорядоченные списки** — это списки, рядом с каждым элементом которых помещается маркер (а не цифровые или буквенные символы, обозначающие порядковый номер).
- **Списки определений** состоят из набора понятий и сопровождающих их определений.

Омлет
Продукты
Приготовление
MacBook Pro

УПОРЯДОЧЕННЫЕ СПИСКИ

Упорядоченные списки создаются при помощи элемента <ol>.

Каждый элемент списка помещается между открывающим тегом <li> и закрывающим тегом .

По умолчанию браузеры автоматически расставляют нужные отступы для элементов списков.

Иногда при просмотре существующих страниц вы будете встречать аргумент type в элементе <ol>, который используется для указания типа нумерации (буквы, римские и арабские цифры и т.п.). Однако лучше использовать свойство list-type-style каскадных таблиц стилей (CSS), описанное в главе 14.

глава-03/ordered-lists.html HTML

```
<ol>
  <li>Порежьте картофель кубиками.</li>
  <li>Варите картофель в соленой воде на медленном
      огне в течение 15-20 минут до размягчения.</li>
  <li>Подогрейте и смешайте молоко, сливочное масло
      и мускатный орех.</li>
  <li>Слейте воду из кастрюли и потолките
      картофель.</li>
  <li>Перемешайте картофель со смесью из молока,
      масла и мускатного ореха.</li>
</ol>
```

РЕЗУЛЬТАТ

1. Порежьте картофель кубиками.
2. Варите картофель в соленой воде на медленном огне в течение 15-20 минут до размягчения.
3. Подогрейте и смешайте молоко, сливочное масло и мускатный орех.
4. Слейте воду из кастрюли и потолките картофель.
5. Перемешайте картофель со смесью из молока, масла и мускатного ореха.

НЕУПОРЯДОЧЕННЫЕ СПИСКИ

HTML **глава-03/unordered-lists.html**

```
<ul>
  <li>1 кг картофеля</li>
  <li>100 мл молока</li>
  <li>50 г соленого масла</li>
  <li>Тертый мускатный орех</li>
  <li>Соль и перец по вкусу</li>
</ul>
```

РЕЗУЛЬТАТ

- 1 кг картофеля
- 100 мл молока
- 50 г соленого масла
- Тертый мускатный орех
- Соль и перец по вкусу

Неупорядоченные списки создаются при помощи элемента `<ul>`.

Каждый элемент списка помещается между открывающим тегом `<li>` и закрывающим тегом `</li>`.
По умолчанию браузеры автоматически расставляют нужные отступы для элементов списков.
Иногда при просмотре существующих страниц вы будете встречать аргумент `type` в элементе `<ol>`, который используется для указания типа маркера (круги, квадраты, ромбы). Однако лучше использовать свойство `list-type-style` каскадных таблиц стилей (CSS), описанное в главе 14.

СПИСКИ ОПРЕДЕЛЕНИЙ

<dl>

Список определений создается с помощью элемента <dl> и, как правило, содержит набор понятий и их определений.

Внутри элемента <dl> вы зачастую сможете увидеть пары элементов <dt> и <dd>.

<dt>

Используется для обозначения понятия (термина).

<dd>

Используется для определения понятия.

Иногда вы можете столкнуться со списками, в которых два понятия имеют одно определение или где для одного понятия приводится несколько определений.

глава-03/definition-lists.html HTML

```
<dl>
  <dt>  Сашими</dt>
  <dd>  Сырая рыба, нарезанная тонкими кусочками;
        подается с такими приправами, как
        нашинкованная японская редька (дайкон),
        корень имбиря, васаби или соевый соус.</dd>
  <dt>  Весы</dt>
  <dd>  Прибор, используемый для определения точного
        веса ингредиентов.</dd>
  <dd>  Зодиакальный знак, распространяющийся на
        календаре с 23 сентября по 22 октября,
        единственный в зодиаке неодушевленный
        символ.</dd>
  <dt>  Ноу-хау</dt>
  <dt>  Ноухау</dt>
  <dd>  Мастерство, умение, практическая сметка,
        высококвалифицированная работа; компетенция
        и опыт, приобретаемые длительной
        практикой.</dd>
</dl>
```

РЕЗУЛЬТАТ

Сашими
: Сырая рыба, нарезанная тонкими кусочками; подается с такими приправами, как нашинкованная японская редька (дайкон), корень имбиря, васаби или соевый соус.

Весы
: Прибор, используемый для определения точного веса ингредиентов.
: Зодиакальный знак, распространяющийся на календаре с 23 сентября по 22 октября, единственный в зодиаке неодушевленный символ.

Ноу-хау
Ноухау
: Мастерство, умение, практическая сметка, высококвалифицированная работа; компетенция и опыт, приобретаемые длительной практикой.

ВЛОЖЕННЫЕ СПИСКИ

HTML **глава-03/nested-lists.html**

```
<ul>
  <li>Муссы</li>
  <li>Пирожные
    <ul>
      <li>Круассаны</li>
      <li>Наполеон</li>
      <li>Чизкейки</li>
      <li>Профитроли</li>
    </ul>
  </li>
  <li>Торты</li>
</ul>
```

РЕЗУЛЬТАТ

- Муссы
- Пирожные
  - Круассаны
  - Наполеон
  - Чизкейки
  - Профитроли
- Торты

Внутри элемента `<li>` допустимо создание вложенного списка, или списка второго уровня.

Браузеры отображают вложенные списки с большим отступом по сравнению с основным, или родительским, списком. Кроме того, браузер, как правило, также изменяет вид маркеров вложенного неупорядоченного списка.

Safari Файл Правка Вид История Закладки Разработка Окно Справка
Списки
file:///Users/Admin/Desktop/index.html
Omлет
Блюда из яиц – мои любимые. Ниже я помещаю рецепт приготовления омлета.
Продукты
2 яйца
70 мл. молока
1/2 ч.л. соли
1 ч.л. растительного масла
Приготовление
1. Яйца перемешать с солью.
2. Влить молоко, слегка перемешивая.
3. Форму для запекания смазать растительным маслом.
4. Вылить омлетную массу в смазанную форму. Поставить в предварительно разогретую до 250°C духовку и готовить омлет в духовке 20-30 минут.
5. Во время запекания на поверхности омлета образуется слегка румяная корочка, по степени зажаривания которой определите готовность омлета.
MacBook Pro

ПРИМЕР
СПИСКИ

В данном примере вы увидите основной заголовок, после которого идет вводный абзац. Используются неупорядоченный список для перечисления ингредиентов и упорядоченный — для описания шагов приготовления блюда.

```
<html>
  <head>
    <title>Списки</title>
  </head>
  <body>
    <h1>Омлет</h1>
    <p> Блюда из яиц — мои любимые. Ниже я помещаю рецепт
        приготовления омлета.</p>
    <h2>Продукты</h2>
    <ul>
      <li>2 яйца</li>
      <li>70 мл. молока</li>
      <li>1/2 ч.л. соли</li>
      <li>1 ч.л. растительного масла</li>
    </ul>
    <h2>Приготовление</h2>
    <ol>
      <li>Яйца перемешать с солью.</li>
      <li>Влить молоко, слегка перемешивая.</li>
      <li>Форму для запекания смазать растительным
          маслом.</li>
      <li>Вылить омлетную массу в смазанную форму.
          Поставить в предварительно разогретую до 250 °C
          духовку и готовить омлет в духовке 20-30
          минут.</li>
      <li>Во время запекания на поверхности омлета
          образуется слегка румяная корочка, по степени
          зажаривания которой определите готовность
          омлета.</li>
    </ol>
  </body>
</html>
```

ОБЗОР
СПИСКИ

- В языке HTML существует три вида списков: упорядоченные, неупорядоченные и списки определений.
- В упорядоченных списках используются цифры.
- В неупорядоченных списках используются маркеры.
- Списки определений применяются для расшифровки терминов.
- Списки могут быть вложены в другие списки.

4

ССЫЛКИ

- Создание ссылок между страницами
- Ссылки на другие сайты
- Ссылки на электронную почту

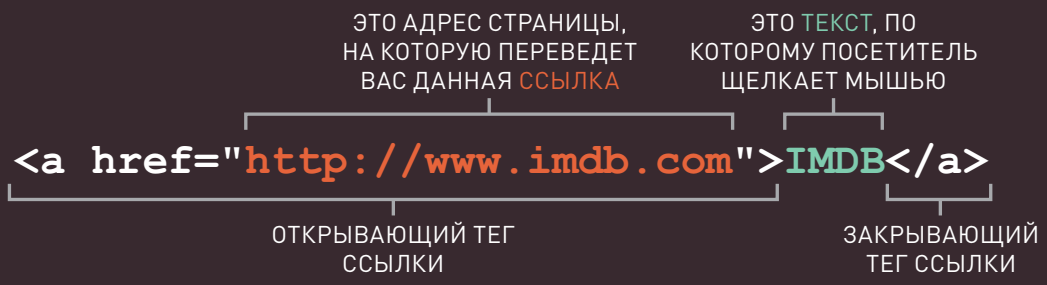
Ссылки — определяющая черта Всемирной паутины, так как они позволяют перемещаться с одного сайта на другой, иными словами, *путешествовать* **по Всемирной паутине.**

Чаще всего вы будете встречаться со ссылками следующих типов:

- ссылки с одного сайта на другой;
- ссылки с одной страницы сайта на другую страницу этого же сайта;
- ссылки с одной части страницы на другую часть этой же страницы;
- ссылки, открывающие новое окно браузера;
- ссылки, запускающие программу для отправки электронных писем, установленную на вашем компьютере, и тем самым позволяющие отправить письмо на определенный адрес электронной почты.

PICCADILLY
EXPRESS TRAINS
EALING COMMON
PARK ROYAL
SUDBURY TOWN
SOUTH HARROW
FROM 4 JULY
DUNLOP
Фестивали
Кинофестивали
Февраль
Май
Июнь
Октябрь
ASP.NET 2.0
XML
2nd Edition
Web Programming with HTML, XHTML, and CSS
HTML5
LANDING PAGE OPTIMIZATION
AMBROSE / HARRIS

Ссылки создаются с помощью элемента <a>. Пользователи могут щелкнуть по любому слову текста, помещенного между открывающим <a> и закрывающим тегами элемента. Используя атрибут href, вы можете указать, на какую страницу будет перенаправлен пользователь при щелчке мышью по данной ссылке.



Текст, помещенный между открывающим <a> и закрывающим тегами, называется «текстом ссылки». По мере возможности он должен пояснять посетителям, куда они будут переведены в результате щелчка мышью (а не просто гласить «щелкните здесь»). На рисунке ниже вы можете видеть ссылку на сайт «Кинопоиск».

Многие люди переходят между страницами, находя ссылки в тексте. Ясный и доходчивый текст способен помочь посетителям вашего сайта отыскать именно то, что нужно им. Это оставит у них больше приятных впечатлений о вашем сайте, что, в свою очередь, поощрит их посещать его чаще. (Кроме того, это поможет людям, использующим программы экранного доступа.) Чтобы создать хороший текст ссылки, подумайте над тем, какие запросы станут использовать люди при поиске страницы с информацией, на которую вы ссылаетесь. Например, вместо текста наподобие «места, где можно переночевать», напишите что-то более конкретное, например «гостиницы Москвы».

ССЫЛКИ НА ДРУГИЕ САЙТЫ

<a>

Ссылки создаются с помощью элемента <a> и атрибута href. Значением атрибута href является страница, на которую вы собираетесь переадресовать посетителя, когда он щелкнет мышью по ссылке.

Можно щелкнуть по любому слову текста, помещенного между открывающим <a> и закрывающим тегами элемента, после чего произойдет переход на страницу, адрес которой указан в значении атрибута href.

При ссылке на другой сайт значением атрибута href должен быть полный, или **абсолютный**, URL-адрес этого сайта.

По умолчанию браузеры отображают текст ссылки синим цветом с подчеркиванием.

глава-04/linking-to-other-sites.html HTML

```
<p>Обзоры кинофильмов:
  <ul>
    <li><a href="http://www.kinopoisk.ru">
        Кинопоиск</a></li>
    <li><a href="http://www.megacritic.ru">
        Мегакритик</a></li>
    <li><a href="http://kinoblog.org">
        Киноблог</a></li>
    <li><a href="http://televesti.ru">
        Телевести.Ру</a></li>
  </ul>
</p>
```

РЕЗУЛЬТАТ

Обзоры кинофильмов:

- Кинопоиск
- Мегакритик
- Киноблог
- Телевести.Ру

АБСОЛЮТНЫЕ URL-АДРЕСА

Аббревиатура URL расшифровывается как Uniform resource locator (единый указатель ресурсов). У каждой веб-страницы есть свой собственный URL, представляющий собой веб-адрес, вводимый в браузере.

Абсолютный URL-адрес начинается с доменного имени сайта, после которого также может быть указан путь к конкретной странице. Если адрес страницы не указан, то по умолчанию откроется главная страница.

ССЫЛКИ НА ДРУГИЕ СТРАНИЦЫ ВАШЕГО САЙТА

HTML **глава-04/linking-to-other-pages.html**

```
<p>
  <ul>
    <li><a href="index.html">На главную</a></li>
    <li><a href="about-us.html">О нас</a></li>
    <li><a href="movies.html">Фильмы</a></li>
    <li><a href="contact.html">Обратная связь</a></li>
  </ul>
</p>
```

РЕЗУЛЬТАТ

- На главную
- О нас
- Фильмы
- Обратная связь

ОТНОСИТЕЛЬНЫЕ URL-АДРЕСА

Для создания ссылок на страницы вашего сайта вы можете использовать относительные URL-адреса. Они представляют собой сокращенную версию абсолютных: вам нет необходимости указывать доменное имя сайта. Далее в этой главе вы поближе познакомитесь с относительными URL-адресами, так как существует несколько полезных сокращений, с помощью которых вы сможете создавать ссылки на страницы вашего сайта. Использование относительных URL-адресов бывает особенно полезным при создании сайта на компьютере, так как при этом вам не придется устанавливать доменное имя сайта и заключать договор на хостинг.

<a>

При создании ссылок на другие страницы вашего сайта нет необходимости указывать в URL-адресе его доменное имя. Вместо этого вы можете воспользоваться сокращенным вариантом, называемым также **относительным** URL-адресом.

Если все страницы сайта находятся в одной и той же папке, то значением атрибута `href` должно быть лишь имя файла страницы.

Если страницы вашего сайта распределены по разным папкам, то вам придется воспользоваться несколько более сложным синтаксисом для указания местоположения нужной страницы относительно текущей. Вы прочитаете об этом более подробно далее в главе.

Если вы взглянете на код для загрузки примеров страниц для каждой из глав, то увидите, что на странице *index.html* помещены ссылки с указанием относительных URL-адресов.

СТРУКТУРА ДИРЕКТОРИЙ

При создании больших сайтов полезной является такая организация, когда страницы каждого раздела сайта помещаются в собственную папку. Папки сайтов также иногда называют директориями.

СТРУКТУРА

Схема справа демонстрирует структуру директорий вымышленного сайта развлечений ExampleArts. Верхняя папка называется **корневым** каталогом. (В этом примере корневой каталог носит имя *examplearts*.) В корневом каталоге хранятся все файлы и папки сайта.
Для упрощения организации хранения файлов каждый раздел сайта имеет свою собственную папку.

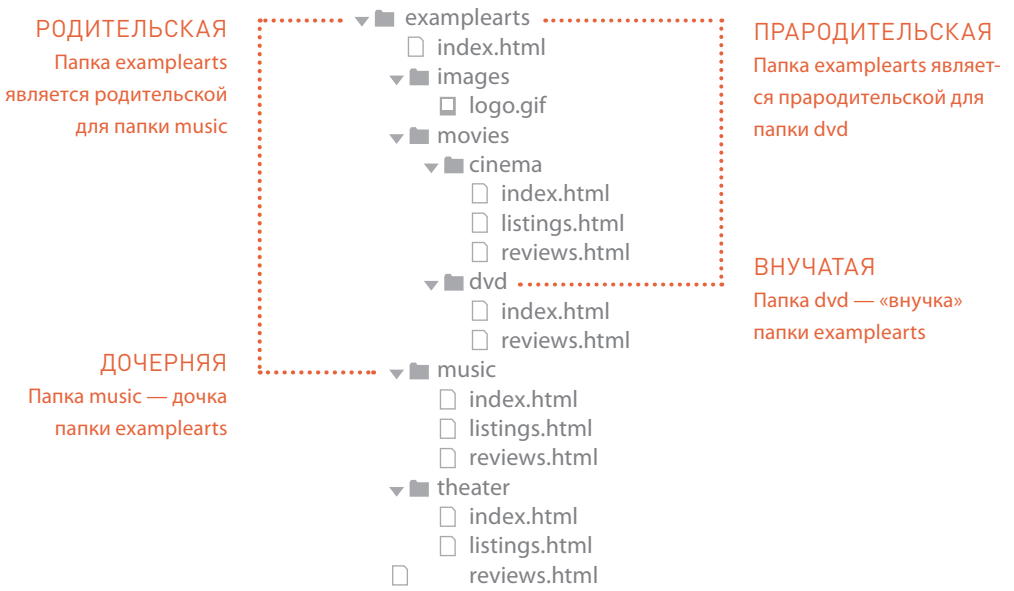
ОТНОШЕНИЯ

Отношения между папками и файлами сайта очень напоминает генеалогическое древо человека.
На схеме справа указаны некоторые отношения. Так, папка *examplearts* является родительской для папок *movies*, *music* и *theater*. В то же время папки *movies*, *music* и *theater* являются дочерними по отношению к папке *examplearts*.

ГЛАВНЫЕ СТРАНИЦЫ

Главная страница сайта, написанная на языке HTML (и главные страницы всех разделов сайта), имеет имя файла *index.html*.
Как правило, веб-серверы настроены таким образом, чтобы возвращать посетителю страницу с именем *index.html*, если иное не указано во введенном адресе. Поэтому, если вы введете адрес **examplearts.com**, то получите страницу *examplearts.com/index.html*; аналогично страница *examplearts.com/music/index.html* будет возвращена при вводе адреса **examplearts.com/music**.

При работе с системой управления контентом (CMS), блог-платформой или платформой электронной коммерции, возможно, для каждой страницы вашего сайта будет и не предусмотрен отдельный файл. Вместо этого в таких системах зачастую используется единый шаблон для страниц каждого типа (например, для новых статей, записей блогов или описаний товаров). Изменение файла шаблона приведет к изменению внешнего вида всех страниц, использующих его. Не изменяйте код шаблона, написанный не на языке HTML, в противном случае страница может перестать функционировать.



Каждая страница и каждое изображение, отображаемое на сайте, имеет собственный **URL-адрес**. Он состоит из доменного имени и **пути** к файлу страницы или изображения.

Путь к главной странице нашего вымышленного сайта выглядит следующим образом: **www.examplearts.com/index.html**. Путь к файлу с логотипом сайта — **www.examplearts/images/logo.gif**.

URL-адреса используются при создании ссылок на другие страницы и при включении изображений. На следующей странице вы познакомитесь с более быстрым способом привязки файлов к сайту.

Корневой каталог содержит:

- файл с именем *index.html*, являющийся главной страницей всего сайта;
- отдельные папки для разделов сайта, посвященных фильмам, музыке, и театру.

Каждая подпапка содержит:

- файл с именем *index.html*, являющийся главной страницей для данного раздела сайта;
- страницу с обзорами, сохраненную с именем *reviews.html*;
- страницу с перечнем названий *listings.html* (кроме раздела, посвященного DVD-дискам).

Раздел, посвященный фильмам, содержит:

- папку *cinema*;
- папку *dvd*.

ОТНОСИТЕЛЬНЫЕ URL-АДРЕСА

Относительные URL-адреса можно использовать при создании ссылок на другие страницы вашего сайта. Они представляют собой короткий вариант указания браузеру местоположения необходимых файлов.

При создании ссылок на страницы вашего сайта нет необходимости указывать доменное имя. Вместо этого вы можете воспользоваться **относительными URL-адресами** — сокращенным способом указания браузеру того места, где находится файл искомой страницы по отношению к текущей.

Это особенно полезно при создании нового сайта или при изучении языка HTML, так как вы можете создавать ссылки между страницами, когда они находятся на вашем компьютере (то есть у вас нет доменного имени и вы еще не загрузили их во Всемирную паутину).

В силу того, что при создании ссылок с использованием относительных адресов вам нет необходимости всякий раз повторять доменное имя сайта, вы сможете сэкономить много времени.

Если все файлы вашего сайта находятся в одной папке, все что вам нужно — это лишь указать имя файла нужной страницы.

Если файлы вашего сайта распределены по папкам (директориям, каталогам), то вам понадобится указать браузеру, как «добраться» с *текущей* страницы до *искомой*.

При создании ссылок на одну и ту же страницу из разных мест, возможно, вам придется указывать разные относительные URL-адреса этой страницы.

При описании ссылок такого рода используется та же самая терминология (позаимствованная из генеалогии), что и при описании структуры директорий сайта.

ВИД ОТНОСИТЕЛЬНОЙ ССЫЛКИ	ПРИМЕР (см. схему на предыдущей странице)
ТА ЖЕ ПАПКА Для создания ссылки на файл, расположенный в той же самой папке, укажите имя этого файла (больше ничего указывать не нужно)	Для создания ссылки на файл с обзорами на главной странице раздела о музыке: `<a href="reviews.html">Обзоры</a>`
ДОЧЕРНЯЯ ПАПКА Для создания ссылки на файл из дочерней папки укажите имя папки, слеш, а затем имя нужного файла	Для создания на главной сайта ссылки на список музыкальных композиций: `<a href="music/listings.html">Список</a>`
ВНУЧАТАЯ ПАПКА Укажите имена дочерней папки, внучатой папки и нужного файла, разделенные слешами	Для создания на главной странице раздела о музыке ссылки на файл с обзорами DVD-дисков: `<a href="movies/dvd/reviews.html">` Обзоры
РОДИТЕЛЬСКАЯ ПАПКА Введите `../` для указания, что файл находится в папке одним уровнем выше, после чего укажите имя нужного файла	Для создания ссылки на главную страницу со страницы обзоров музыкальных композиций: `<a href="../index.html">`На главную
ПРАРОДИТЕЛЬСКАЯ ПАПКА Повторите ввод символов `../` чтобы показать, что вы хотите подняться на два уровня вверх (а не на один), после чего укажите имя нужного файла	Ссылка на главную страницу со страницы обзоров DVD-дисков: `<a href="../../index.html">`На главную

Когда сайт загружен на веб-сервер, становятся доступны еще несколько приемов, не работающих, если он находится на локальном компьютере.

Например, можно указать имя дочерней папки без указания имени файла. В данной ситуации сервер, как правило, постарается отобразить главную страницу указанного раздела.

Слеш вернет главную страницу всего сайта, а если после него указано имя файла, то пользователю будет возвращена страница, сохраненная как файл с указанным именем в корневом каталоге.

ССЫЛКИ НА ЭЛЕКТРОННУЮ ПОЧТУ

mailto:

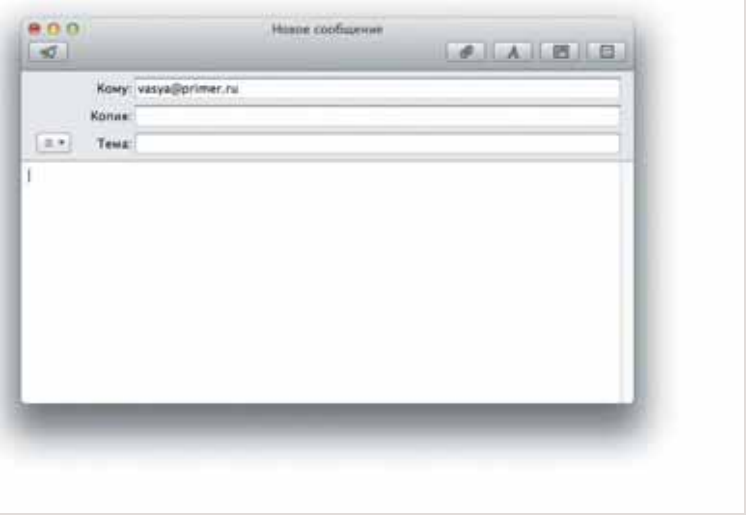
Для создания ссылки, которая запускала бы программу электронной почты, установленную на компьютере посетителя, и позволяла бы отправить электронное письмо на указанный вами адрес, вам нужно будет воспользоваться элементом `<a>`. Однако теперь значение атрибута `href` должно начинаться со слова `mailto:`, после которого необходимо указать адрес электронной почты, на который посетитель должен будет отправить свое письмо. На рисунке справа вы можете увидеть ссылку на электронную почту, которая выглядит как обычная ссылка, но при щелчке мышью по ней она запускает программу для отправки электронной почты и создает новое письмо, в адресной строке которого будет введен адрес, указанный вами при создании ссылки.

глава-04/email-links.html HTML

```
<a href="mailto:vasya@primer.ru">Напишите Васе</a>
```

РЕЗУЛЬТАТ

Напишите Васе



ОТКРЫТИЕ ССЫЛОК В НОВОМ ОКНЕ

HTML **глава-04/opening-links-in-a-new-window.html**

```
<a href="http://www.imdb.com" target="_blank">
База данных фильмов во Всемирной паутине</a>
(открывается в новом окне)
```

РЕЗУЛЬТАТ

target

Чтобы создать ссылку, открывающую страницу в новом окне браузера, вы можете воспользоваться атрибутом `target` открывающего тега элемента `<a>`. Значение этого атрибута должно быть установлено в `_blank`.

Чаще всего ссылки данного вида ставятся, когда щелчок по ним должен вести к переходу на сторонний сайт. В этом случае создатели сайтов надеются, что посетитель вернется на их сайт по завершении просмотра открывшейся страницы.

В обычной ситуации следует избегать открытия ссылок в новом окне, однако если вы используете этот прием, то будет не лишним заранее предупредить посетителя, что страница откроется в новом окне.

ССЫЛКА НА ОПРЕДЕЛЕННУЮ ЧАСТЬ СТРАНИЦЫ

В верхней части довольно длинной страницы можно поместить что-то наподобие оглавления: перечень ссылок, указывающих на соответствующие разделы, расположенные ниже. Кроме того, вы также можете поместить ссылку на верхнюю часть страницы где-нибудь в середине или в нижней части страницы, избавив посетителя от необходимости вращать колесико мышки, чтобы добраться до верха.

Прежде чем создавать ссылку на определенную часть страницы, сначала нужно отметить те участки страницы, на которые будут указывать ссылки. Это можно сделать с помощью атрибута `id`, который допускается использовать со всеми элементами разметки HTML. В этом примере вы можете видеть элементы `<h1>` и `<h2>`, которым был присвоен атрибут `id` для идентификации разделов страницы.

Значение атрибута `id` должно начинаться с латинской буквы или знака подчеркивания (но не с цифры или любого другого символа). Кроме того, на одной странице несколько элементов не могут иметь одинаковое значение атрибута `id`.

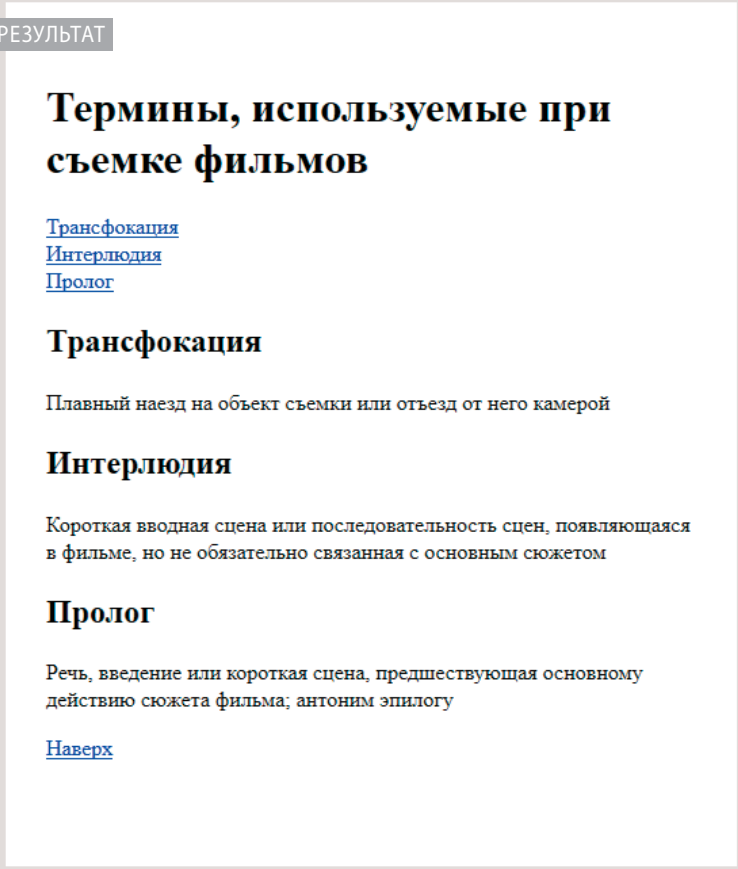
Для создания ссылки на элемент с установленным значением атрибута `id` вам нужно воспользоваться элементом `<a>`, значение атрибута `href` которого должно начинаться с символа «#», после которого следует указать значение атрибута `id` нужного вам элемента. В примере ссылка `<a href="#top">` ссылается на элемент `<h1>`, находящийся в верхней части страницы, атрибуту `id` которого присвоено значение `top`.

глава-04/linking-to-a-specific-part.html HTML

```
<h1 id="top">Термины, используемые при съемке
    фильмов</h1>
<a href="#zoom">Трансфокация</a><br />
<a href="#interlude">Интерлюдия</a><br />
<a href="#prologue">Пролог</a><br />
<h2 id="zoom">Трансфокация</h2>
<p>Плавный наезд на объект съемки или отъезд от него
    камерой.</p>
<h2 id="interlude">Интерлюдия</h2>
<p>Короткая вводная сцена или последовательность
    сцен, появляющаяся в фильме, но не обязательно
    связанная с основным сюжетом.</p>
<h2 id="prologue">Пролог</h2>
<p>Речь, введение или короткая сцена, предшествующая
    основному действию сюжета фильма; антоним
    эпилогу.</p>
<p><a href="#top">Наверх</a></p>
```

ССЫЛКА НА ОПРЕДЕЛЕННУЮ ЧАСТЬ ДРУГОЙ СТРАНИЦЫ

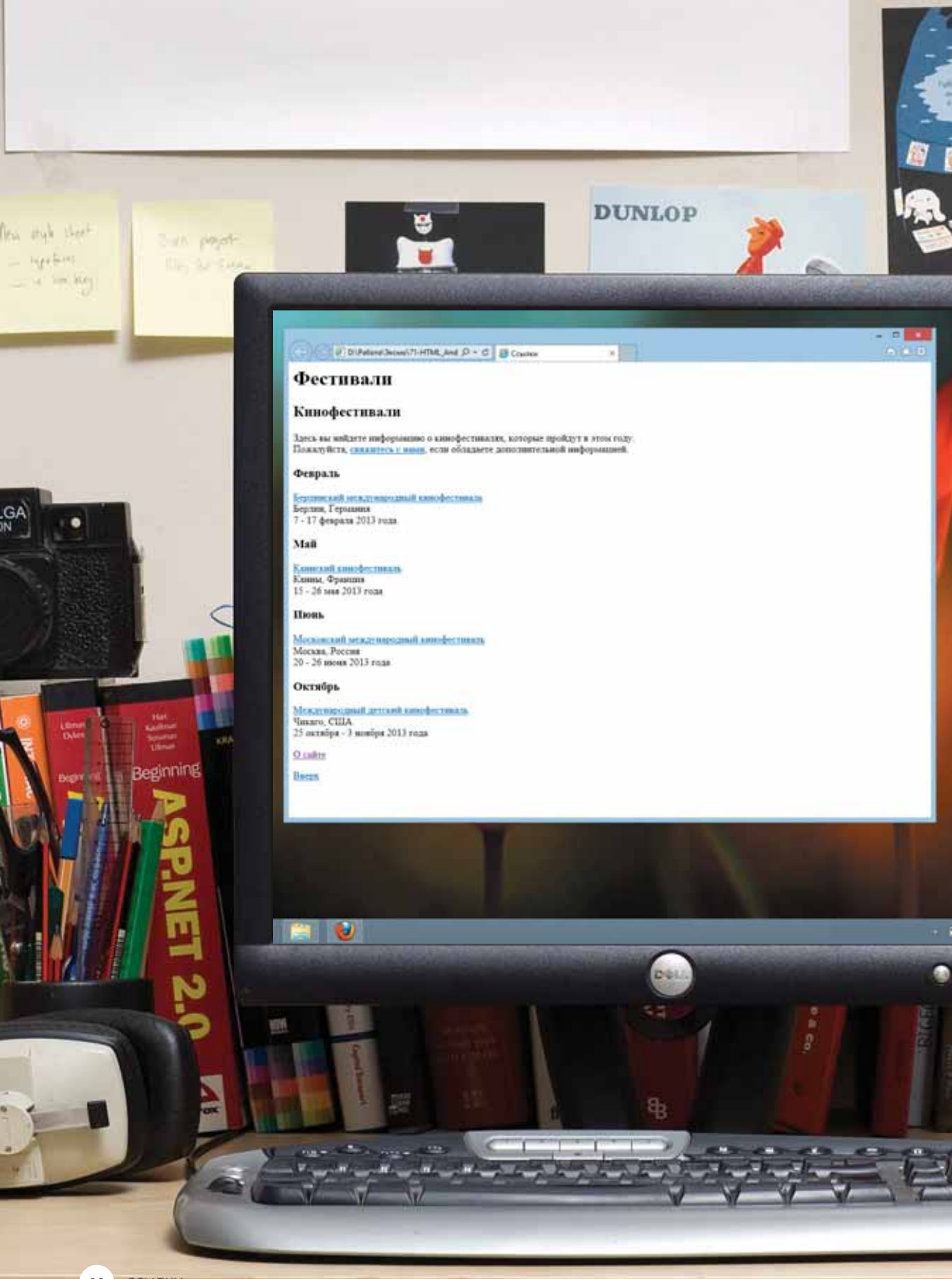
РЕЗУЛЬТАТ



Если вы хотите создать ссылку на определенную часть другой страницы (на вашем или стороннем сайте), можете воспользоваться приемом, аналогичным представленному в предыдущем разделе. В случае если элементы на странице, на которую вы ссылаетесь, имеют атрибуты `id`, обозначающие разделы этой страницы, то вы можете добавить в конец ссылки на нее тот же самый код, что был использован в предыдущем разделе. Следовательно, атрибут `href` должен содержать URL-адрес страницы (абсолютный или относительный) и значение атрибута `id` нужного элемента, разделенные символом #.

Например, для ссылки на нижнюю часть главной страницы сайта **www.htmlandcssbook.com**, вам потребуется написать следующую строку кода:

```
<a href="http://www.htmlandcssbook.com/#bottom">.
```


DUNLOP
Ссылки
Фестивали
Кинофестивали
Здесь вы найдете информацию о кинофестивалях, которые пройдут в этом году.
Пожалуйста, свяжитесь с нами, если обладаете дополнительной информацией.
Февраль
Берлинский международный кинофестиваль
Берлин, Германия
7 - 17 февраля 2013 года
Май
Канский кинофестиваль
Канны, Франция
15 - 26 мая 2013 года
Июнь
Московский международный кинофестиваль
Москва, Россия
20 - 26 июня 2013 года
Октябрь
Международный детский кинофестиваль
Чикаго, США
25 октября - 3 ноября 2013 года
О сайте
Вверх
Beginning
ASP.NET 2.0

ПРИМЕР
ССЫЛКИ

Данный пример — это веб-страница о расписании кинофестивалей.

Элементу `<h1>` присвоен атрибут `id`, чтобы можно было создать ссылку, щелчок мышью по которой переместил бы посетителя с нижней части страницы в верхнюю. На странице также размещена ссылка на электронную почту, позволяющая написать автору веб-страницы. Кроме того, на странице есть несколько ссылок на сайты, посвященные различным кинофестивалям. В нижней части страницы расположена относительная ссылка на страницу «О сайте», находящуюся в той же директории, что и страница-пример.

```
<html>
  <head>
    <title>  Ссылки</title>
  </head>
  <body>
    <h1 id="top">Фестивали</h1>
    <h2> Кинофестивали</h2>
    <p>  Здесь вы найдете информацию о кинофестивалях,
          которые пройдут в этом году.<br />Пожалуйста,
          <a href="mailto:filmfestival@example.org">
          свяжитесь с нами</a>, если обладаете
          дополнительной информацией.</p>
    <h3> Февраль</h3>
    <p>  <a href="http://www.berlinale.de/en/HomePage.html">
          Берлинский международный кинофестиваль</a><br />
          Берлин, Германия<br />7-17 февраля 2013 года</p>
    <h3> Май</h3>
    <p>  <a href="http://www.festival-cannes.com/ru.html">
          Каннский кинофестиваль</a><br />Канны, Франция<br />
          15-26 мая 2013 года</p>
    <!-- дополнительный контент -->
    <p><a href="about.html">О сайте</a></p>
    <p><a href="#top">Вверх</a></p>
  </body>
</html>
```

ОБЗОР
ССЫЛКИ

- Ссылки создаются с помощью элемента <a>.
- Для обозначения адреса страницы, на которую указывает ссылка, используется атрибут href элемента <a>.
- Для создания ссылок на страницы собственного сайта лучше всего использовать относительные URL-адреса.
- Вы можете создавать ссылки, открывающие программу для обмена электронной почтой и заполняющие адресное поле.
- Для обозначения целевых элементов страницы, на которые можно создавать ссылки, пользуйтесь атрибутом id.

5

ИЗОБРАЖЕНИЯ

- Как добавлять изображения на страницу
- Выбор правильного формата
- Оптимизация изображений для использования во Всемирной паутине

Существует достаточно много оснований, чтобы добавить на веб-страницу какое-либо изображение. Этим изображением может быть, например, логотип, фотография, иллюстрация, схема или диаграмма.

Прежде чем добавлять изображение на сайт, нужно задуматься о нескольких моментах, но время, затраченное на них, окупится тем, что ваш сайт будет выглядеть гораздо более профессионально и привлекательно. В этой главе вы узнаете, как:

- с помощью языка HTML добавлять изображения на веб-страницы;
- выбирать подходящий формат файла изображения;
- выводить изображение нужных размеров на экран;
- оптимизировать изображения для использования во Всемирной паутине, чтобы увеличить скорость загрузки ваших страниц.

Для добавления изображений на страницу вы также можете воспользоваться свойством `background-image` каскадных таблиц стилей (CSS), с которым познакомитесь в дальнейшем.

Safari Файл Правка Вид История Закладки Разработка Окно Справка
Чт, 17:39
Изображение
От A до Яблока
chocolate islands
Этот рецепт для приготовления уникальных шоколадных кексов так прост и так восхитителен!
Еще рецепты:

ВЫБОР ИЗОБРАЖЕНИЙ ДЛЯ САЙТА

Одно изображение стоит тысячи слов, кроме того, правильно подобранные изображения отличают обычные сайты от по-настоящему привлекательных.

Изображение несравнимо быстрее задает тон сайту, чем текстовое описание. Если у вас нет собственных фотографий для размещения на сайте, вы можете обратиться к **стоковым фотохостингам**, предлагающим миллионы изображений, доступных для использования за некоторое вознаграждение или вовсе бесплатно (чуть ниже вы найдете список таких ресурсов). Помните о том, что все изображения — это объекты, охраняемые законом об авторском праве, потому простое использование графики, скопированной с других сайтов, может повлечь серьезные проблемы.

Если вы создали страницу с несколькими изображениями (например, фотографиями товара или членов команды), поместите их на простой, но содержательный фон — это улучшит их восприятие как единой группы.

ИЗОБРАЖЕНИЯ ДОЛЖНЫ:

- быть к месту;
- нести информацию;
- создавать правильное настроение;
- быть легкоузнаваемыми;
- соответствовать цветовой палитре сайта.

СТОКОВЫЕ ФОТОХОСТИНГИ:

www.istockphoto.com
www.gettyimages.com
www.veer.com
www.sxc.hu
www.fotolia.com

ПРИМЕЧАНИЕ

На сайте www.Htmlandcssbook.com размещена онлайн-галерея, которая поможет вам выбрать правильное изображение для вашего сайта.

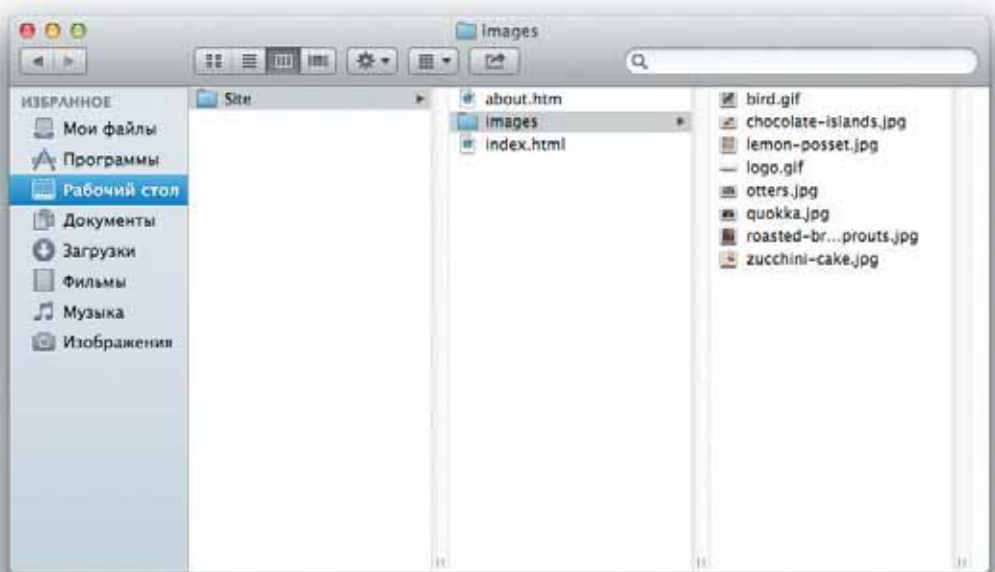
РАЗМЕЩЕНИЕ ИЗОБРАЖЕНИЙ НА САЙТЕ

Если вы создаете сайт с нуля, то полезно создать отдельную папку для хранения всех используемых на нем изображений.

Если ваш сайт будет расти и расширяться, размещение всех изображений в отдельной папке упростит управление файлами. Ниже вы можете видеть пример того, как нужно хранить изображения для сайта: все рисунки лежат в папке с именем *images*.

При создании крупного сайта вы также можете добавить в папку *images* несколько подпапок. Например такие изображения, как логотипы и кнопки, можно хранить в подпапке *interface*, фотографии товаров — в подпапке *products*, а изображения, связанные с новостями сайта, — в подпапке *news*.

Если вы используете систему управления контентом (CMS) или блог-платформу, то, как правило, там уже есть специальный инструментарий, позволяющий загружать изображения на сайт, и, возможно, отдельная папка для хранения графических и других пользовательских файлов.



ДОБАВЛЕНИЕ ИЗОБРАЖЕНИЙ

Для добавления изображения на страницу воспользуйтесь элементом <img>. Он является пустым (то есть без закрывающего тега). Для него должны быть указаны следующие два атрибута.

src

Данный атрибут сообщает браузеру расположение нужного изображения. Как правило, его значением будет относительный URL-адрес, указывающий на изображение, хранящееся в директориях вашего собственного сайта. (В данном примере все изображения расположены в подпапке *images*. Об относительных URL-адресах речь шла в предыдущей главе.)

alt

Данный атрибут предоставляет текстовое описание изображения, выводимое на экран в случае, если нет возможности показать само изображение.

title

С элементом <img> вы также можете использовать атрибут title для сообщения дополнительной информации об изображении. Большинство браузеров отобразят содержимое этого атрибута в качестве всплывающей подсказки при наведении указателя мыши на изображение.

глава-05/adding-images.html HTML

```
<img src="images/quokka.jpg" alt="Семейка
квокки" title="Квокка или короткохвостый
кенгуру — единственный представитель рода
Setonix семейства кенгуровых. Квокка внешне
напоминает кенгуру, но имеет короткий, даже
относительно размеров тела, хвост." />
```

РЕЗУЛЬТАТ

Текст, указываемый в значении атрибута alt, часто называют **замещающим текстом**. Он должен представлять собой точное описание изображения, чтобы оно могло быть распознано программным обеспечением экранного доступа (используемым людьми с нарушением зрения) и поисковыми системами.

Если вы используете изображение лишь как декорацию для улучшения оформления сайта (и оно не несет никакой смысловой нагрузки, как, например, графический разделитель — горизонтальная линия), используйте атрибут alt с пустым значением (между кавычек не должно быть никакого текста).

ВЫСОТА И ШИРИНА ИЗОБРАЖЕНИЙ

HTML **глава-05/height-and-width-of-images.html**

```
<img src="images/quokka.jpg" alt="Семейка
квокки" width="600" height="450" />
```

РЕЗУЛЬТАТ

Зачастую вы будете видеть элемент `<img>` с еще двумя атрибутами, определяющими его размер:

height

Данный атрибут устанавливает высоту изображения в пикселах.

width

Данный атрибут устанавливает ширину изображения в пикселах.

Как правило, время загрузки изображений значительно дольше, чем HTML-кода, составляющего всю остальную разметку страницы. Поэтому рекомендуется указывать размеры изображений, чтобы браузер мог продолжить обработку текста сайта, не дожидаясь окончания их загрузки и оставив необходимое количество пустого места.

Все чаще и чаще создатели сайтов устанавливают размеры изображений с помощью каскадных таблиц стилей (CSS), а не посредством языка HTML. Для получения дополнительной информации об этом см. главу 16.

РАЗМЕЩЕНИЕ ИЗОБРАЖЕНИЙ В КОДЕ

Расположение изображения в коде влияет на то, как оно будет выведено на экран. Ниже приведено три примера размещения изображения, приводящих к разным результатам.

ПЕРЕД АБЗАЦЕМ

Абзац начинается с новой строки после изображения.

В НАЧАЛЕ АБЗАЦА

Первая строка текста выравнивается относительно нижнего края изображения.

В СЕРЕДИНЕ АБЗАЦА

Изображение помещается в текст абзаца, в котором говорится о нем.

глава-05/where-to-place-images.html HTML

```
<img src="images/bird.gif" alt="Птица"
width="100" height="100" />
<p>В мире насчитывается более 10 тысяч видов птиц,
обитающих в различных экосистемах от Арктики до
Антарктики. Многие виды ежегодно совершают перелеты
на длинные расстояния, гораздо большее количество
совершают более короткие спонтанные путешествия.</p>
<hr />
<p><img src="images/bird.gif" alt="Птица"
width="100" height="100" /> В мире насчитывается
более 10 тысяч видов птиц, обитающих в различных
экосистемах от Арктики до Антарктики. Многие виды
ежегодно совершают перелеты на длинные расстояния,
гораздо большее количество совершают более короткие
спонтанные путешествия.</p>
<hr />
<p>В мире насчитывается более 10 тысяч видов птиц,
обитающих в различных экосистемах от Арктики до
Антарктики. <img src="images/bird.gif" alt="Птица"
width="100" height="100" />Многие виды ежегодно
совершают перелеты на длинные расстояния, гораздо
большее количество совершают более короткие
спонтанные путешествия.</p>
```

РЕЗУЛЬТАТ

В мире насчитывается более 10 тысяч видов птиц, обитающих в различных экосистемах от Арктики до Антарктики. Многие виды ежегодно совершают перелеты на длинные расстояния, гораздо большее количество совершают более короткие спонтанные путешествия.

В мире насчитывается более 10 тысяч видов птиц, обитающих в различных экосистемах от Арктики до Антарктики. Многие виды ежегодно совершают перелеты на длинные расстояния, гораздо большее количество совершают более короткие спонтанные путешествия.

В мире насчитывается более 10 тысяч видов птиц, обитающих в различных экосистемах от Арктики до Антарктики.

Многие виды ежегодно совершают перелеты на длинные расстояния, гораздо большее количество совершают более короткие спонтанные путешествия.

То, как вы размещаете изображение в коде, крайне важно, поскольку браузеры отображают HTML-элементы двумя следующими способами.

Блочные элементы выводятся с новой строки. Примерами блочных элементов могут служить `<h1>` и `<p>`.

Если после элемента `<img>` следует блочный элемент (такой, как новый абзац), то он будет перенесен на новую строку после изображения, как показано в первом примере на этой странице.

Встроенные элементы помещаются внутри блочного элемента и не начинают новую строку. Примерами таких элементов могут служить `<b>`, `<em>` и `<img>`.

Если элемент `<img>` находится внутри блочного элемента, то текст или другой встроенный элемент будет обтекать изображение, как показано во втором и третьем примере.

Блочные и встроенные элементы более подробно обсуждаются в главе 8.

УСТАРЕВШИЙ ПРИЕМ: ВЫРАВНИВАНИЕ ИЗОБРАЖЕНИЙ ПО ГОРИЗОНТАЛИ

align

Атрибут `align` раньше использовался для обозначения того, как остальные части страницы должны обтекать изображение. Этот атрибут был исключен из стандарта HTML5 — при создании новых сайтов для выравнивания изображений нужно использовать каскадные таблицы стилей CSS (о чем вы прочтете в главе 16).

Мы решили все же упомянуть об этом атрибуте по той причине, что при изучении исходного кода старых сайтов вы довольно часто будете с ним встречаться, а также потому, что некоторые визуальные редакторы используют именно его, когда вы устанавливаете выравнивание изображения.

Атрибут `align` может иметь одно из двух значений выравнивания по горизонтали.

left

Это значение выравнивает изображение по левому краю (позволяя тексту обтекать его справа).

right

Это значение выравнивает изображение по правому краю (позволяя тексту обтекать его слева).

глава-05/aligning-images-horizontally.html HTML

```
<p><img src="images/bird.gif" alt="Птица"
    width="100" height="100" align="left" /> В мире
    насчитывается более 10 тысяч видов птиц,
    обитающих в различных экосистемах от Арктики до
    Антарктики. Многие виды ежегодно совершают
    перелеты на длинные расстояния, гораздо большее
    количество совершают более короткие спонтанные
    путешествия.</p>
<hr />
<p><img src="images/bird.gif" alt="Птица"
    width="100" height="100" align="right" /> В мире
    насчитывается более 10 тысяч видов птиц,
    обитающих в различных экосистемах от Арктики до
    Антарктики. Многие виды ежегодно совершают
    перелеты на длинные расстояния, гораздо большее
    количество совершают более короткие спонтанные
    путешествия.</p>
```

РЕЗУЛЬТАТ

В мире насчитывается более 10 тысяч видов птиц, обитающих в различных экосистемах от Арктики до Антарктики. Многие виды ежегодно совершают перелеты на длинные расстояния, гораздо большее количество совершают более короткие спонтанные путешествия.

В мире насчитывается более 10 тысяч видов птиц, обитающих в различных экосистемах от Арктики до Антарктики. Многие виды ежегодно совершают перелеты на длинные расстояния, гораздо большее количество совершают более короткие спонтанные путешествия.

Благодаря такому выравниванию, страница выглядит гораздо опрятней, чем когда изображение просто вставлено на одну строку с текстом (как это было продемонстрировано в предыдущем примере). При присваивании атрибуту `align` значения `left` изображение помещается у левого края страницы, а текст обтекает его вокруг. При присваивании атрибуту `align` значения `right` изображение помещается у правого края страницы, а текст обтекает его вокруг. Если текст вплотную примыкает к краю изображения, это ухудшает его читаемость. О том, как вставить небольшой промежуток между изображением и текстом с помощью свойств `padding` и `margin` каскадных таблиц стилей (CSS), вы прочитаете в главе 13.

УСТАРЕВШИЙ ПРИЕМ: ВЫРАВНИВАНИЕ ИЗОБРАЖЕНИЙ ПО ВЕРТИКАЛИ

Как мы уже говорили ранее, атрибут `align` больше не используется в HTML5, но мы обсуждаем его в этой книге, так как вы можете встретить его при изучении исходного кода старых сайтов, а также он все еще используется при создании страниц с помощью некоторых визуальных редакторов.

Узнать о том, как пользоваться каскадными таблицами стилей (CSS) для достижения тех же результатов, вы можете в главе 12.

Атрибуту `align` допускается присваивать одно из трех значений, устанавливающих выравнивание изображения по вертикали относительно окружающего его текста.

глава-05/aligning-images-vertically.html HTML

```
<p><img src="images/bird.gif" alt="Птица"
  width="100" height="100" align="top" /> В мире
  насчитывается более 10 тысяч видов птиц, обитающих
  в различных экосистемах от Арктики до Антарктики.
  Многие виды ежегодно совершают перелеты на длинные
  расстояния, гораздо большее количество совершают
  более короткие спонтанные путешествия.</p>
<hr />
<p><img src="images/bird.gif" alt="Птица"
  width="100" height="100" align="middle" /> В мире
  насчитывается более 10 тысяч видов птиц, обитающих
  в различных экосистемах от Арктики до Антарктики.
  Многие виды ежегодно совершают перелеты на длинные
  расстояния, гораздо большее количество совершают
  более короткие спонтанные путешествия.</p>
<hr />
<p><img src="images/bird.gif" alt="Птица"
  width="100" height="100" align="bottom" /> В мире
  насчитывается более 10 тысяч видов птиц, обитающих
  в различных экосистемах от Арктики до Антарктики.
  Многие виды ежегодно совершают перелеты на длинные
  расстояния, гораздо большее количество совершают
  более короткие спонтанные путешествия.</p>
```

top

Это значение выравнивает первую строку текста с верхней границей изображения.

middle

Это значение устанавливает выравнивание первой строки текста относительно середины изображения.

bottom

Это значение выравнивает первую строку текста относительно нижней границы изображения.

РЕЗУЛЬТАТ

В мире насчитывается более 10 тысяч видов птиц, обитающих в различных экосистемах от Арктики до Антарктики. Многие виды ежегодно совершают перелеты на длинные расстояния, гораздо большее количество совершают более короткие спонтанные путешествия.

В мире насчитывается более 10 тысяч видов птиц, обитающих в различных экосистемах от Арктики до Антарктики. Многие виды ежегодно совершают перелеты на длинные расстояния, гораздо большее количество совершают более короткие спонтанные путешествия.

В мире насчитывается более 10 тысяч видов птиц, обитающих в различных экосистемах от Арктики до Антарктики. Многие виды ежегодно совершают перелеты на длинные расстояния, гораздо большее количество совершают более короткие спонтанные путешествия.

Значение атрибута `top` помещает первую строку текста на один уровень с верхним краем изображения, при этом остальные строки текста оказываются под изображением.

Значение `middle` помещает первую строку текста на один уровень с вертикальным центром изображения, остальные строки текста выводятся под изображением.

Значение `bottom` помещает первую строку текста чуть выше нижнего края изображения, при этом остальные строки текста оказываются под изображением.

Если текст вплотную примыкает к краю изображения, это ухудшает его читаемость. О том, как вставить небольшой промежуток между изображением и текстом с помощью свойств `padding` и `margin` каскадных таблиц стилей (CSS), вы прочитаете в главе 13.

Если вы хотите, чтобы не только первая строка, но и весь текст обтекал изображение, то вам нужно будет воспользоваться свойством `float` каскадных таблиц стилей, о котором вы можете прочитать в главе 15.

При изучении старого исходного кода вы увидите что подобный эффект достигался с помощью значений `left` или `right` атрибута `align` (как описано на предыдущей странице) — впрочем, сейчас их использование не рекомендуется.

ТРИ ПРАВИЛА СОЗДАНИЯ ИЗОБРАЖЕНИЙ

Существует три правила, которые необходимо помнить при создании изображений для сайтов, все они перечислены ниже. На протяжении следующих девяти **страниц мы детально обсудим каждое из них.**

1

СОХРАНЯЙТЕ ИЗОБРАЖЕНИЯ В ПРАВИЛЬНОМ ФОРМАТЕ.

В основном на сайтах используется графика трех файловых форматов: JPEG, GIF и PNG. При использовании других форматов изображение может потерять резкость, замедлить скорость загрузки страницы или вовсе не отобразиться на компьютере посетителя.

2

СОХРАНЯЙТЕ ИЗОБРАЖЕНИЯ С ПРАВИЛЬНЫМИ ЗНАЧЕНИЯМИ ШИРИНЫ И ВЫСОТЫ.

Сохраняйте изображения с теми же значениями ширины и высоты, с какими они должны быть отображены на сайте. Если значения ширины и/или высоты изображения меньше указанных вами в коде, это может привести к растягиванию и искажению рисунка. Если ширина и/или высота изображения больше, чем вы указали в коде, то для его отображения браузеру потребуется больше времени.

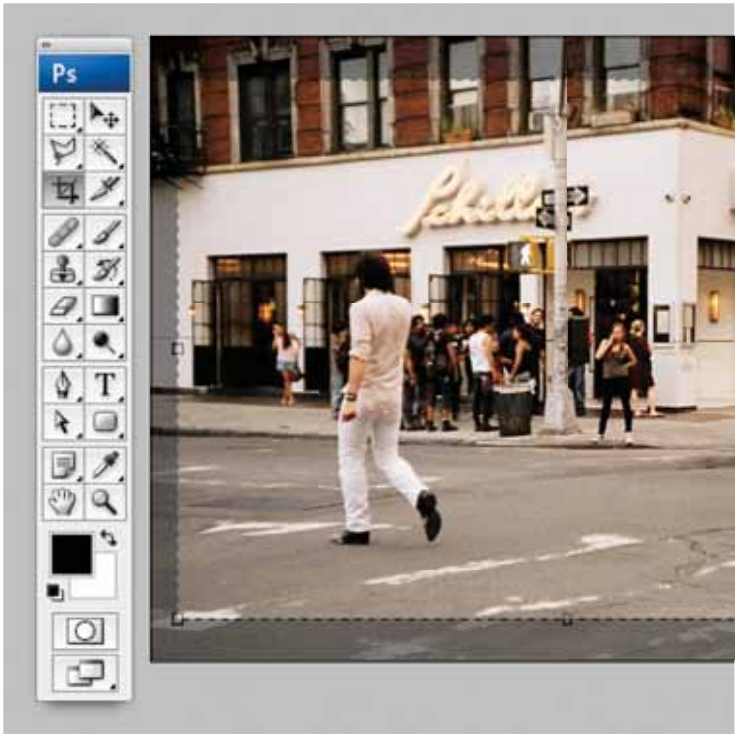
3

ИСПОЛЬЗУЙТЕ ПРАВИЛЬНОЕ РАЗРЕШЕНИЕ.

Экраны компьютеров состоят из точек, называемых пикселами. Изображения, опубликованные во Всемирной паутине, также состоят из очень маленьких точек. Разрешением называется количество точек, помещающихся на одном дюйме. Большинство мониторов отображают сайты с разрешением 72 пиксела на дюйм. Сохранение рисунка с большим разрешением приведет к увеличению его размеров, результатом чего, в свою очередь, будет более низкая скорость загрузки.

ИНСТРУМЕНТЫ ДЛЯ РЕДАКТИРОВАНИЯ И СОХРАНЕНИЯ ИЗОБРАЖЕНИЙ

Существует несколько инструментов, которыми вы можете пользоваться для редактирования и сохранения изображений с требуемыми размерами, разрешением и в нужном формате.



Среди профессиональных веб-дизайнеров самым популярным инструментом является программа Adobe Photoshop. (На самом деле профессионалы зачастую используют его для разработки графики всего сайта.) Полная версия программы Photoshop весьма дорогостоящая, однако также доступна более дешевая версия этой программы — Photoshop Elements, которая придется по вкусу большинству новичков.

ДРУГИЕ РЕДАКТОРЫ:
Adobe Fireworks
Pixelmator
PaintShop Pro
Paint.net

ОНЛАЙН-РЕДАКТОРЫ:
`www.photoshop.com`
`www.pixlr.com`
`www.splashup.com`
`www.ipiccy.com`

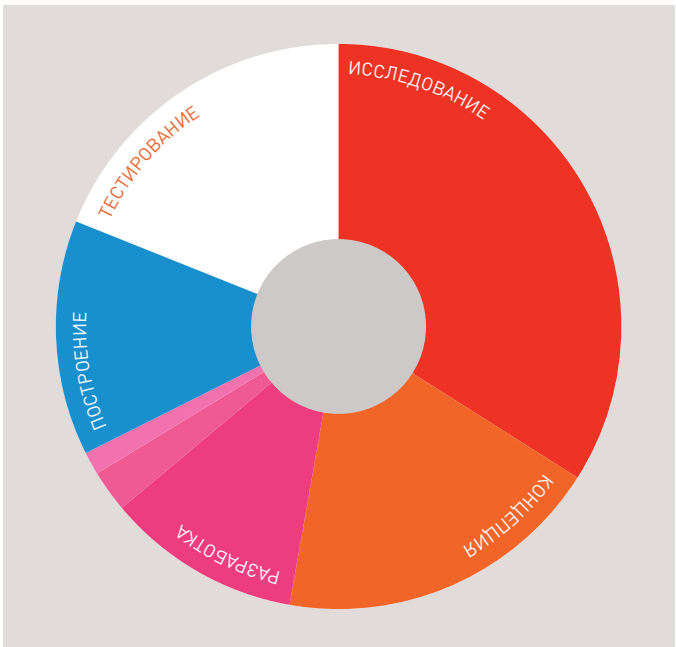
ПРИМЕЧАНИЕ
На сайте **www.htmlandcssbook.com** вы найдете видеокурс, демонстрирующий, как изменять размеры изображения и сохранять их в нужном формате с помощью программы Photoshop.

ФОРМАТ ИЗОБРАЖЕНИЙ: JPEG

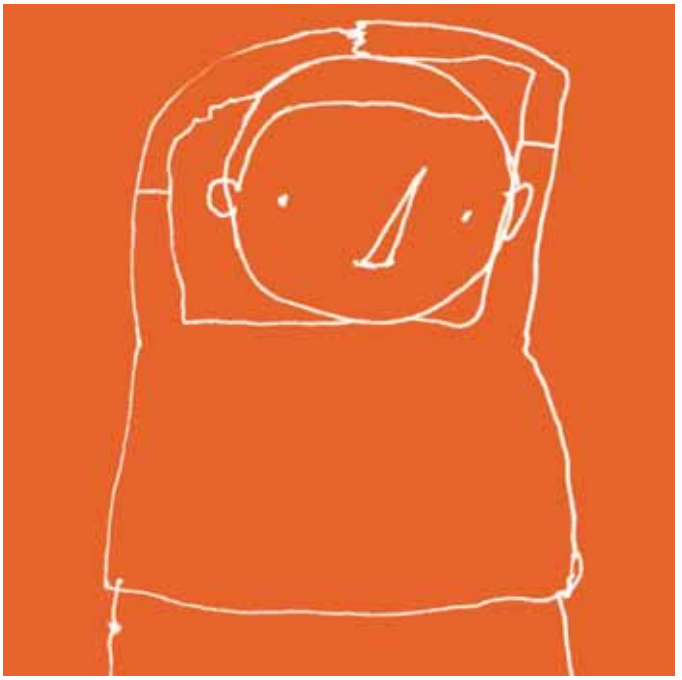
В случае если ваше изображение содержит большое количество оттенков, всегда используйте формат JPEG. Может показаться, что фотографии с изображением снега или пасмурного неба состоят из больших областей однотонного белого или серого цвета, но это не так: обычно фотографии содержат огромное число оттенков, едва различимых невооруженным глазом.

de
SOUK

ФОРМАТ ИЗОБРАЖЕНИЙ: GIF



Используйте формат GIF или PNG при сохранении изображений с малым количеством оттенков или имеющих большие однотонные области



Если на рисунке есть область, окрашенная одним цветом, она называется «однотонной». Логотипы, иллюстрации и схемы чаще всего имеют однотонные области. (Обратите внимание, что фотографии снега, неба или травы не однотонные: они состоят из большого количества оттенков, различия между которыми едва заметны человеческому глазу, и не подходят для сохранения в формате GIF или PNG.)

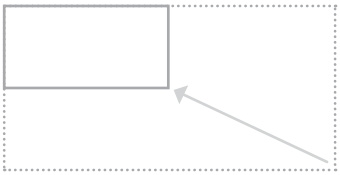
РАЗМЕРЫ ИЗОБРАЖЕНИЯ

Изображения, сохраняемые в папках сайта, должны обладать теми же самыми значениями высоты и ширины, что вы указали в коде страницы.

Например, если вы создали веб-страницу, которая включает изображение 150 пикселов в ширину и 300 в высоту, то вам нужно использовать изображение с размерами 150 × 300 пикселов. Для кадрирования или изменения размеров имеющегося изображения вы можете воспользоваться графическим редактором. При создании изображений для сайта очень важно понимать то, *как* можно изменять их размеры. Давайте представим, что вы создали веб-страницу, на которой нужно разместить изображение шириной 300 и высотой 150 пикселов.

УМЕНЬШЕНИЕ РАЗМЕРА ИЗОБРАЖЕНИЯ

Вы можете уменьшить размер исходного изображения для создания его миниатюры.



Пример. Если ширина вашего изображения 600 пикселов, а высота — 300, то вы можете уменьшить размер изображения на 50%.

Результат. Это приведет к созданию изображения, скорость загрузки которого выше, чем исходного.

УВЕЛИЧЕНИЕ РАЗМЕРА ИЗОБРАЖЕНИЯ

Значительное увеличение размера фотографии без потери качества рисунка невозможно.

Пример. Увеличение размеров изображения 100 пикселов в ширину и 50 в высоту на 300% приведет к ухудшению его качества.

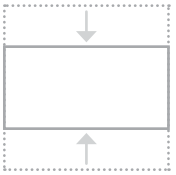
Результат. Изображение станет размытым или пикселизированным.

ПРИМЕЧАНИЕ

Посетите сайт **www.htmlandcssbook.com**, чтобы посмотреть видеоурок по изменению размеров изображений в программах Photoshop и GIMP.

КАДРИРОВАНИЕ ИЗОБРАЖЕНИЯ

Не все изображения можно кадрировать без потери важной графической информации (см. следующую страницу).



Пример. Если изображение представляет собой квадрат со стороной 300 пикселов, то вы можете обрезать какую-либо его часть, однако при этом вы рискуете потерять важную графическую информацию.

Результат. Не все изображения можно кадрировать так, чтобы они не потеряли своей значимости.

КАДРИРОВАНИЕ ИЗОБРАЖЕНИЙ

При кадрировании изображений очень важно не потерять ценную графическую информацию. На сайтах лучше всего использовать изображения нужной ширины и высоты.

КНИЖНАЯ ОРИЕНТАЦИЯ

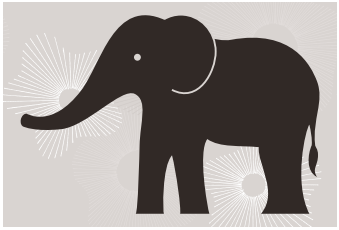
На этом рисунке вы видите картинку с жирафом, которая больше подходит для отображения в **книжной** ориентации.

АЛЬБОМНАЯ ОРИЕНТАЦИЯ

При **кадрировании** рисунка для отображения его в альбомной ориентации вы потеряете голову и ноги жирафа.

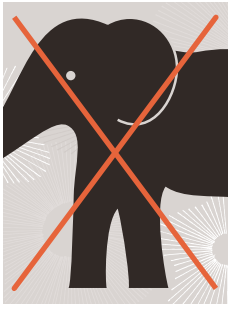
Если вы **добавите дополнительное пространство** слева и справа от рисунка, то непрерывного фона изображения у вас не получится.

АЛЬБОМНАЯ ОРИЕНТАЦИЯ

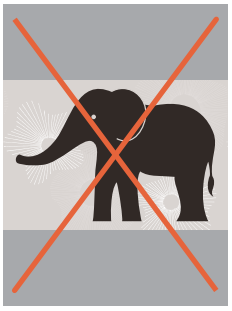


На этом рисунке вы видите картинку со слоном, которая лучше всего подходит для отображения в **альбомной** ориентации.

КНИЖНАЯ ОРИЕНТАЦИЯ



Если вы **кадрируете** это изображение так, чтобы вывести его в книжной ориентации, то потеряете хобот и заднюю часть слона.



Если вы **добавите дополнительное пространство** сверху и снизу от рисунка, то непрерывного фона изображения у вас не получится

РАЗРЕШЕНИЕ ИЗОБРАЖЕНИЯ

Изображение, сохраняемое для публикации во Всемирной паутине, должно иметь разрешение 72 ppi. Чем выше разрешение изображения, тем больше размер файла.

Форматы JPEG, GIF и PNG принадлежат к семейству форматов **растровых** изображений. Такие изображения состоят из большого числа миниатюрных квадратов. **Разрешением** изображения называется количество этих квадратов, умещающихся на площади в 1 квадратный дюйм. Изображения, выводимые на **экран** компьютера, состоят из маленьких квадратов, называемых **пикселами**. Для демонстрации того, как изображение составляется из пикселов, мы увеличили фрагмент изображения. На большинстве настольных компьютеров браузеры

отображают графику с разрешением 72 ppi*. Изображения в **печатных** изданиях, таких как книги и журналы, состоят из крохотных кружочков, называемых **точками**. Такие изображения обычно печатаются с разрешением 300 точек на дюйм, dpi**.

Данное изображение:
Формат JPEG 300 ppi = 1526 кбайт
Формат JPEG 72 ppi = 368 кбайт

По причине того, что разрешающая способность экранов компьютеров не превышает 72 ppi, использование на сайте графики с большим разрешением не приведет к улучшению качества отображения, результатом будет лишь увеличение размера файла, что, в свою очередь, увеличит время, необходимое для его загрузки, и замедлит скорость просмотра страниц.

* Pixels per inch (*англ.*) — пикселов на дюйм. — *Примеч. ред.*
** Dots per inch (*англ.*) — точек на дюйм. — *Примеч. ред.*

ВЕКТОРНАЯ ГРАФИКА

Векторные изображения отличаются от растровых и не зависят от разрешения. Создавать векторные изображения можно в таких программах, как Adobe Illustrator.

Когда изображение — это совокупность линий (как бывает в случае с логотипами, иллюстрациями или схемами), оно, как правило, сохраняется в векторном формате.

Векторные изображения создаются путем нанесения точек на координатную сетку и соединения их линиями. После этого к созданным линиям может быть добавлен цвет.

Главное преимущество сохранения подобных изображений в векторном формате заключается в том, что впоследствии вы сможете увеличивать их без потери качества.

Использование векторных изображений на сайте подразумевает создание растровой копии исходного векторного изображения с последующим размещением ее на странице.
Формат SVG* относительно новый. Он используется для непосредственного отображения векторной графики на сайтах (без необходимости создания растровых копий), однако еще не получил широкого распространения.

* Scalable Vector Graphics (*англ.*) — масштабируемая векторная графика. — *Примеч. ред.*

АНИМИРОВАННЫЙ GIF

Анимированный GIF-файл позволяет отображать последовательность из нескольких изображений и поэтому может быть использован для создания простой анимации.

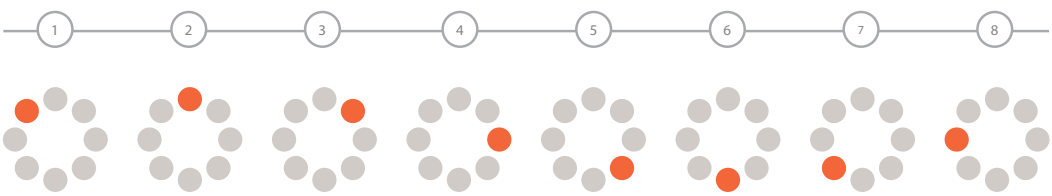
В нижней части страницы изображена последовательность кадров — отдельных изображений, — составляющих GIF-анимацию: оранжевая точка вращается по кругу. Подобную анимацию вы можете видеть при загрузке страницы.

С помощью некоторых программ для обработки изображений, таких как Adobe Photoshop, можно создавать GIF-анимацию. Во Всемирной паутине находится множество руководств, посвященных этому вопросу. Там же есть и сайты, позволяющие загрузить отдельные изображения для создания из них анимации.

ВАЖНО ПОМНИТЬ.
Каждый кадр увеличивает размер изображения, отчего скорость его загрузки становится медленнее (а пользователи Интернета не любят долго ждать).

Из-за того, что формат GIF — это не лучшее решение для отображения фотографий, он чаще используется для простых иллюстраций.

Некоторые дизайнеры избегают создания анимации в формате GIF, так как еще жива память о девяностых годах, когда начинающие сайтостроители битком набивали свои страницы такими «шевелящимися картинками».



ПРОЗРАЧНОСТЬ

Создание частично прозрачных изображений (то есть таких, сквозь которые видны объекты на заднем плане) для сайтов предполагает выбор одного из следующих двух форматов.

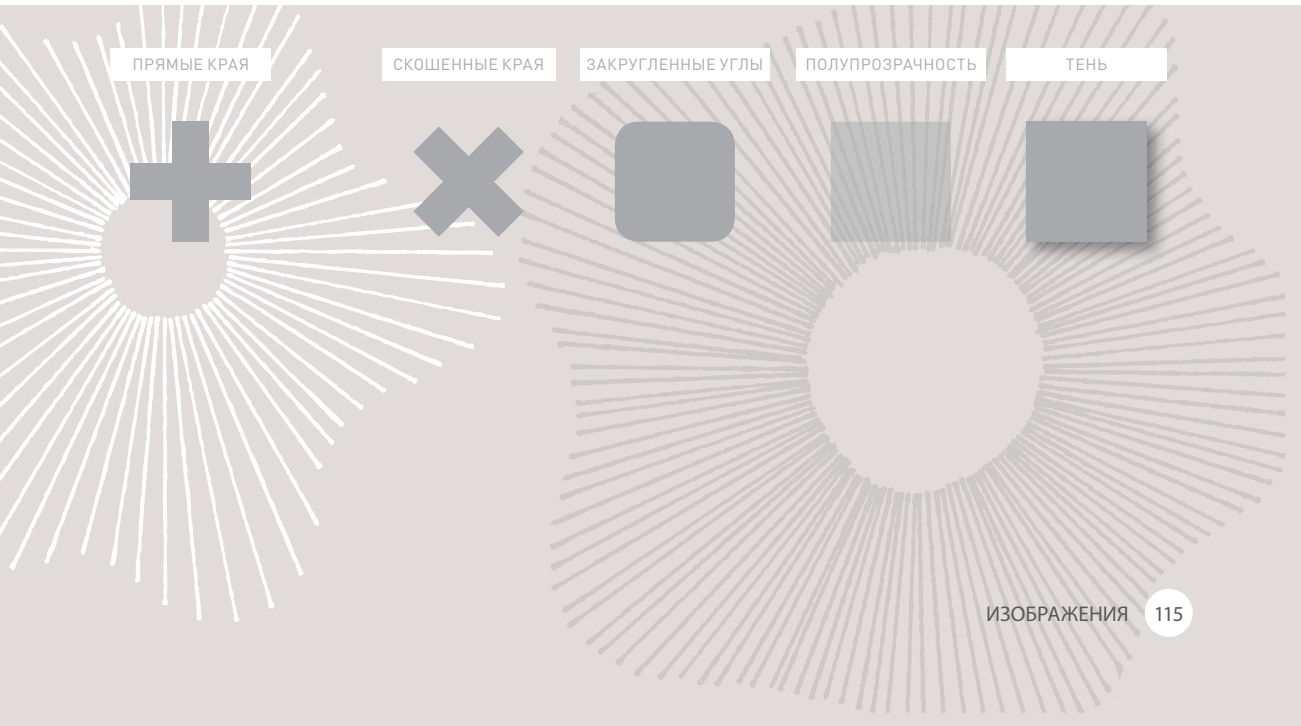
ПРОЗРАЧНЫЙ GIF

Если прозрачная область создаваемого вами изображения имеет четкие края и степень ее прозрачности равна 100%, то есть она не должна быть *полу*прозрачной), то вы можете сохранить такое изображение в формате GIF (установив флажок **Прозрачность** (Transparency).

PNG

Если изображения содержит области, прозрачные лишь частично, либо если у таких областей скошенные или округлые очертания, либо же если вы хотите добавить эффект падающей тени — тогда вам подойдет формат PNG.

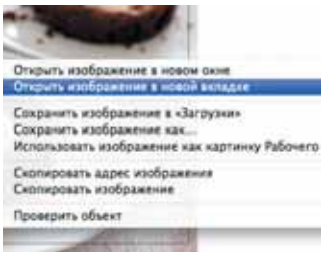
Формат PNG не полностью поддерживается устаревшими браузерами (особенно это заметно в Internet Explorer 6). Существует сценарий на языке JavaScript, позволяющий обойти эту проблему. Более подробно о нем вы можете прочесть на сайте **www.htmlandcssbook.com**.



ИСПОЛЬЗОВАНИЕ ИЗОБРАЖЕНИЙ ВО ВСЕМИРНОЙ ПАУТИНЕ

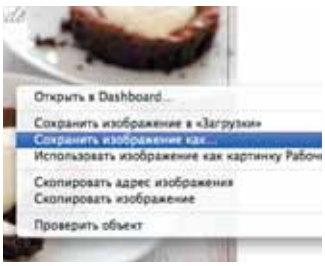
ПРОВЕРКА РАЗМЕРА ИЗОБРАЖЕНИЙ

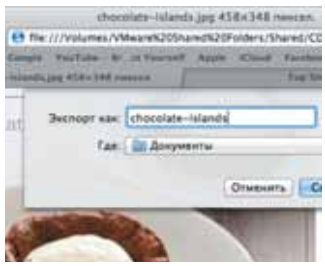
При обновлении контента сайта иногда бывает нужно узнать размеры имеющегося на нем изображения, прежде чем заменить его новым. Это можно сделать, щелкнув правой кнопкой мыши и выбрав соответствующий пункт в контекстном меню. (Пользователи компьютеров Mac могут щелкнуть мышью по изображению при нажатой клавише **Ctrl**.)



ЗАГРУЗКА ИЗОБРАЖЕНИЙ НА КОМПЬЮТЕР

Если вы хотите сохранить изображения с сайта на свой компьютер, можете сделать это, выбрав соответствующий пункт того же контекстного меню. (Однако не забывайте, что все изображения — объекты авторского права и для их использования требуется согласие их владельцев.)





На рисунках слева показано, как узнать размер изображения, а также как загрузить его на компьютер с помощью браузера Safari. Ниже мы поместили названия команд меню для пользователей разных браузеров.

CHROME

Размер: **Открыть картинку в новой вкладке** (Open Image in New Tab).
Размер изображения появится в заголовке новой вкладки.
Загрузить: **Сохранить картинку как** (Save Image As).

FIREFOX

Размер: **Информация об изображении** (View Image Info)
Размер будет указан в открывшемся диалоговом окне
Загрузить: **Сохранить изображение как** (Save Image As)

INTERNET EXPLORER

Размер: **Свойства** (Properties).
Размер будет указан в открывшемся диалоговом окне.
Загрузить: **Сохранить изображение как** (Save Image As).

SAFARI

Размер: **Открыть изображение в новой вкладке** (Open Image in New Tab).
Размер изображения появится в строке заголовка.
Загрузить: **Сохранить изображение как** (Save Image As).

HTML5: ИЛЛЮСТРАЦИИ И ПОДПИСИ

HTML **глава-05/figure-and-figure-caption.html**

```
<figure>
  <img src="images/otters.jpg"
    alt="Фотография двух выдр, держащихся за лапки">
    <img />
  <br />
      <figcaption>Во время сна выдры держат друг
      друга за лапки, чтобы их не разнесло
      течением.</figcaption>
</figure>
```

РЕЗУЛЬТАТ

Во время сна выдры держат друг друга за лапки, чтобы их не разнесло течением.

<figure>

Зачастую изображения сопровождаются подписями.
В спецификации HTML5 появился новый элемент `<figure>`, объединяющий изображение и подпись к нему, таким образом текст и рисунок оказываются связанными друг с другом.
В один элемент `<figure>` можно включить несколько изображений, в случае если для них используется одна подпись.

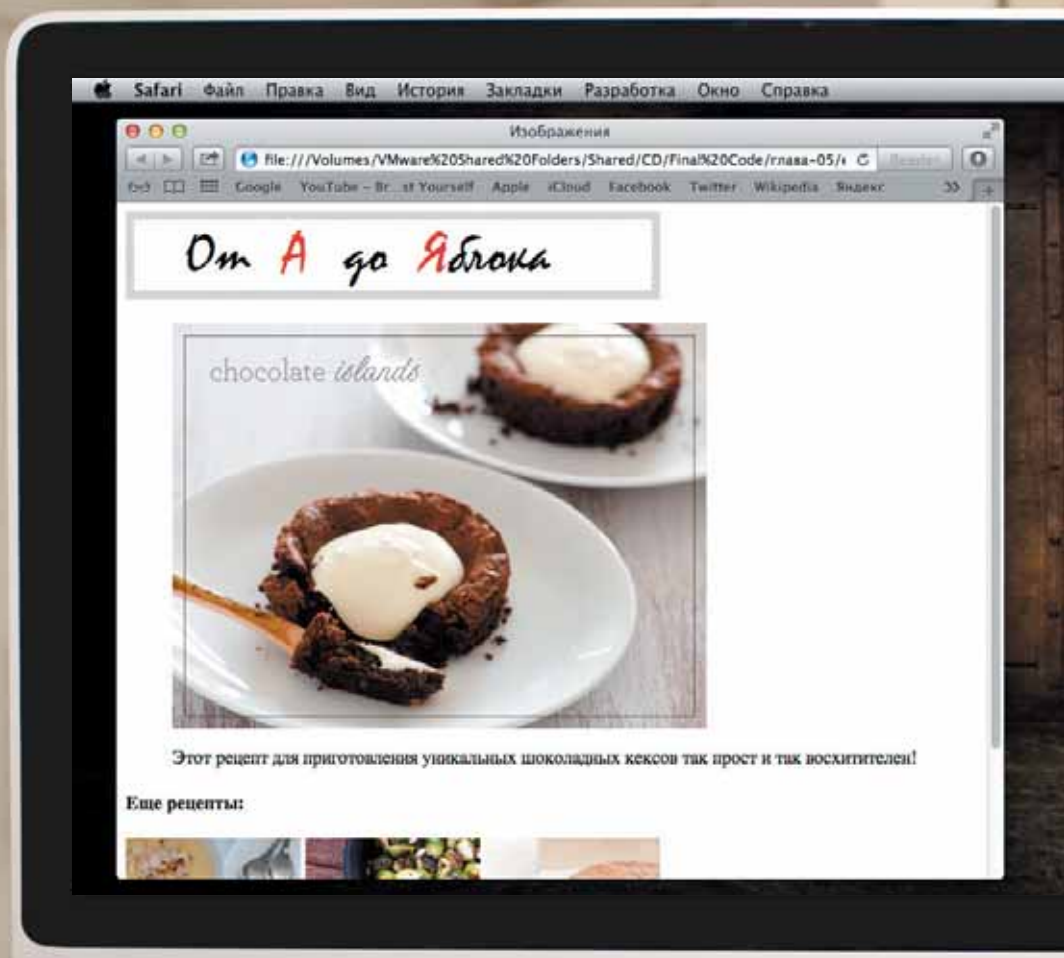
<figcaption>

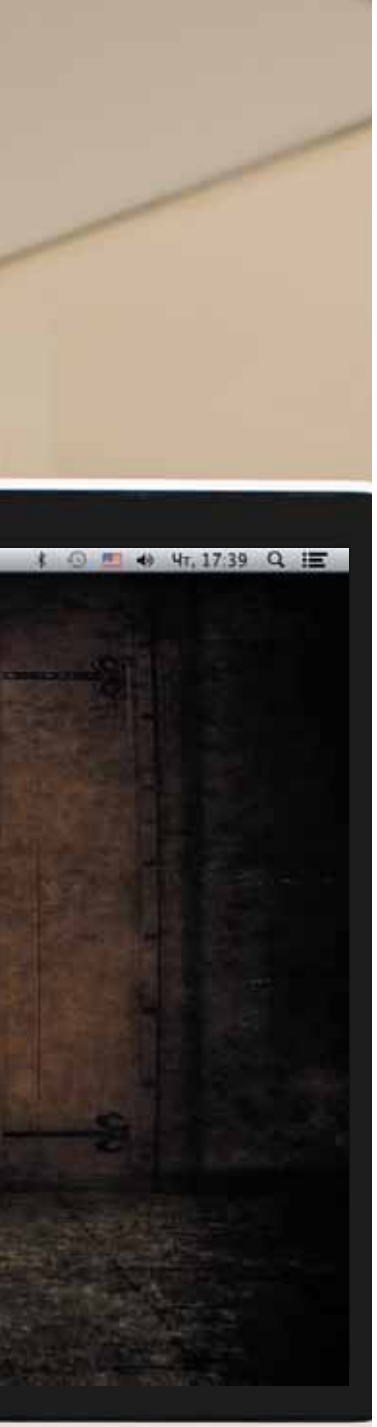
Элемент `<figcaption>` был включен в язык HTML5, чтобы позволить веб-дизайнерам добавлять подписи к публикуемым изображениям.
До введения этих двух элементов связать изображение `<img>` с текстом было невозможно.
Устаревшие версии браузеров не распознают элементы языка HTML5 и просто отображают их содержимое на странице.

В приведенном примере логотип — это изображение в формате GIF (оно содержит однотонные участки), а фотографии сохранены в формате JPEG. Основная фотография помещена в элемент `<figure>` языка HTML5 и у нее есть подпись.

Атрибут `alt` каждого из изображений предоставляет описание того, что можно видеть на рисунке, для посетителей сайта, использующих программы экранного доступа. Атрибут `title` содержит дополнительную информацию, выводимую во всплывающей подсказке.

В этом примере вы не увидите таких атрибутов, как `height`, `width` и `align`, так как на сегодняшний день они считаются устаревшими и вместо них рекомендуется использовать соответствующие свойства каскадных таблиц стилей (CSS).





ПРИМЕР
ИЗОБРАЖЕНИЯ

```
<html>
  <head>
    <title>Изображения</title>
  </head>
  <body>
    <h1>
      <img src="images/logo.gif" alt="От А до Яблока" />
    </h1>
    <figure>
      <img src="images/chocolate-islands.jpg"
        alt="Шоколадные острова" title="Уникальные кексы
        Шоколадных островов" />
      <p>
        <figcaption>
          Этот рецепт для приготовления уникальных
          шоколадных кексов так прост и так восхитителен!
        </figcaption>
      </p>
    </figure>
    <h4>Еще рецепты:</h4>
    <p>
      <img src="images/lemon-posset.jpg"
        alt="Лимонный поссет" title="Десерт - лимонный
        поссет" />
      <img src="images/roasted-brussel-sprouts.jpg"
        alt="Жареная брюссельская капуста" title="Гарнир
        из жареной брюссельской капусты" />
      <img src="images/zucchini-cake.jpg" alt="Яблочный
        пирог" title="Яблочный пирог без охлаждения" />
    </p>
  </body>
</html>
```

ОБЗОР
ИЗОБРАЖЕНИЯ

- Для добавления изображений на веб-страницу используется элемент <img>.
- Всегда прописывайте атрибут src для указания расположения изображения и атрибут alt для описания того, что на нем можно видеть.
- Перед размещением, запаситесь изображениями нужного размера и формата.
- Фотографии лучше всего сохранять в формате JPEG, иллюстрации и логотипы с однотонными областями — в формате GIF.

6

ТАБЛИЦЫ

- Как создавать таблицы
- Какая информация подходит для таблиц
- Как представить в таблицах сложные данные

Существует некоторое количество видов информации, которые следует отображать в виде сетки или таблицы, например результаты спортивных соревнований, информация фондовых рынков, расписания поездов.

Для создания таблиц нужно иметь представление о специальных терминах, таких как строки и столбцы. В этой главе вы узнаете как:

- использовать четыре ключевых элемента для создания таблиц;
- представлять в таблицах сложные данные;
- добавлять подписи к таблицам.

AVEC
METAPOL
Pâte ou liquide
FER·FONTE·ACIER
retrouvent
le Poli du neuf
LION NOIR
Letratone
LT1000
ART SHEET
HIGH TEA

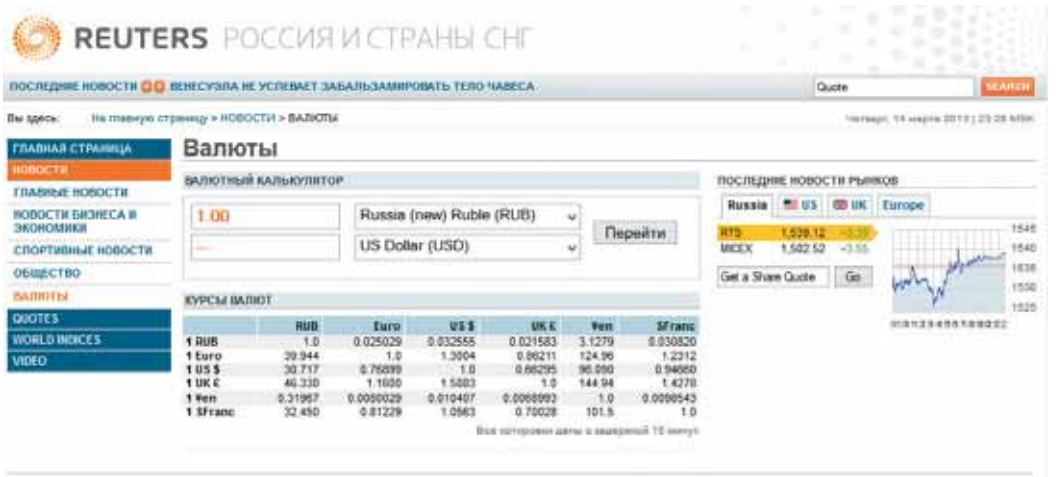
АЙ 2012
САМЫЕ КРУПНЫЕ РАЗЛИВЫ НЕФТИ В ИСТОРИ
ИСТОЧНИК НЕФТИ
МЕСТО
ПРИЧИНА
Нефтяные месторождения Кувейта
Кувейт, Персидский залив
Месторождение Лайквью-Гашер
США, штат Калифорния
ошибка при бурении
Нефтяная платформа Deepwater Horizon
Мексиканский залив, США
Нефтяная платформа ixTOC1
Мексиканский залив, Мексика
Танкеры Atlantic Empress и Aegean Captain
Карибское море, у побережья Тринидада-и-Тобаго
Месторождение Мингбулак
Узбекистан
Нефтепровод Харьяга — Усинск
Россия, Республика Коми, бассейн реки Колва
Нефтяная платформа Новруз
Персидский залив, Иран
Танкер ABT Summer
Атлантический океан, 700 миль от побережья Анголы
Танкер Castillo De Bellver
у побережья Южной Африки
Танкер Amoco Cadiz
у побережья Бретани, Франц
Танкер Haven
7 миль от Генуи, Италия
Танкер Odyssey
Атлантический океан, 70 от побережья Канады
Танкер Torrey Canyon
острова Силли, Велик
Танкер Sea Star
Оманский залив
Салат «Мон»
Салат «Гобойский»
Салат «Грезский»
Салат «Цезарь»
Салат «Цезарь с беконом»
Салат «Цезарь с креветками»
Салат «Арлекино»
Салат «Океан»
Салат «Гоги»
Салат «Нина»
Салат «Обаяний»
Шоколадный брил
Наполеон
Жареное морое
Жареный бой
тел.: 8-9
ПИЦЦА
ДЕСЕРТЫ
САЛАТЫ
ВАЗАНЫЯ МЯСНАЯ ГРИБНОЙ ЖУЛЬЕН
СЭНДВИЧИ В АССОРТИМЕНТЕ
ПАСТА
СУПЫ
НАПИТКИ
КОФЕ
ЧАЙ
МОРОЖЕНОЕ
МОРОЖЕНОЕ BR В АССОРТИМЕНТЕ

ЧТО ТАКОЕ ТАБЛИЦА?

Таблица представляет информацию в виде сетки. Примерами таблиц могут служить финансовые отчеты, телепрограммы и результаты спортивных соревнований.

Сетка помогает воспринимать большие объемы данных путем соотнесения их с подписями по обеим осям.

Отдельный блок сетки называется **ячейкой таблицы**. С помощью языка HTML таблицы пишутся построчно.



БАЗОВАЯ СТРУКТУРА ТАБЛИЦЫ

<table>

Для создания таблиц используется элемент <table>. Контент таблицы описывается построчно.

<tr>

Открывающий тег <tr>* обозначает начало новой строки таблицы.
После него помещаются элементы <td>, каждый из которых соответствует отдельной ячейке в этой строке.
Конец строки обозначается закрывающим тегом </tr>.

<td>

Каждая ячейка таблицы должна быть представлена элементом <td>**.
Конец ячейки обозначается закрывающим тегом </td>.
Некоторые браузеры автоматически создают линии вокруг таблиц и/или ячеек. Вы узнаете, как управлять границами таблиц с помощью каскадных таблиц стилей (CSS) в главах 13 и 14.

глава-06/basic-table-structure.html HTML

```
<table>
  <tr>
    <td>15</td>
    <td>15</td>
    <td>30</td>
  </tr>
  <tr>
    <td>45</td>
    <td>60</td>
    <td>45</td>
  </tr>
  <tr>
    <td>60</td>
    <td>90</td>
    <td>90</td>
  </tr>
</table>
```

РЕЗУЛЬТАТ

15 15 30
45 60 45
60 90 90

* Table row (*англ.*) — строка таблицы. — *Примеч. ред.*
** Table data (*англ.*) — данные таблицы. — *Примеч. ред.*

ЗАГОЛОВКИ ТАБЛИЦ

HTML глава-06/table-headings.html

```
<table>
  <tr>
    <th></th>
    <th scope="col">Суббота</th>
    <th scope="col">Воскресенье</th>
  </tr>
  <tr>
    <th scope="row">Продано билетов</th>
    <td>120</td>
    <td>135</td>
  </tr>
  <tr>
    <th scope="row">Выручка</th>
    <td>18 000 руб.</td>
    <td>20 250 руб.</td>
  </tr>
</table>
```

РЕЗУЛЬТАТ

	Суббота	Воскресенье
Продано билетов	120	135
Выручка	18000 руб.	20250 руб.

<th>

Элемент `<th>`* используется точно так же, как и элемент `<td>`, однако его назначение — создание заголовка строки или столбца.

Помните, что даже если ячейка не содержит данных, то вы все равно должны представить ее пустым элементом `<td>` или `<th>`, в противном случае таблица будет некорректно отображена в браузере. (В примере первая ячейка первой строки пустая.)

Употребление элемента `<th>` помогает людям, использующим программы экранного доступа, улучшает результативность индексирования таблиц поисковыми системами, а еще этот элемент помогает управлять оформлением таблиц при использовании CSS.

Для обозначения того, относится ли данный заголовок к строке или столбцу, вы можете воспользоваться атрибутом `scope` элемента `<th>`. Этот атрибут способен принимать одно из двух значений: `row`** или `col`*** для обозначения, к чему именно (к строке или столбцу) относится данный заголовок.

Как правило, браузеры выделяют полужирным шрифтом содержимое элемента `<th>` и выравнивают его по центру ячейки.

---

* Table heading (*англ.*) — заголовок таблицы. — *Примеч. ред.*

** Row (*англ.*) — строка. — *Примеч. ред.*

*** Column (*англ.*) — столбец. — *Примеч. ред.*

ОБЪЕДИНЕНИЕ СТОЛБЦОВ

Иногда вам может понадобиться, чтобы ячейка таблицы занимала не один, а несколько столбцов. Атрибут `colspan` может быть использован с элементами `<th>` и `<td>` для обозначения, сколько столбцов должна занимать ячейка.

В примере справа вы можете видеть расписание, состоящее из пяти столбцов: первый содержит заголовок строки (день недели), оставшиеся четыре представляют собой часы.

Если вы посмотрите на код ячейки с текстом «География», то увидите, что значение атрибута `colspan` равно 2, это значит, что ячейка должна занимать два столбца. Ячейка «Физкультура», расположенная на третьей строке, занимает три столбца.

Вы также можете видеть, что вторая и третья строки состоят из меньшего количества элементов `<td>`. Дело в том, что если какая-либо ячейка занимает несколько столбцов, то занятые ею элементы `<td>` в коде не прописываются.

Для иллюстрации этого примера мы использовали несколько стилей CSS с тем, чтобы показать вам, как одна ячейка может занимать сразу несколько столбцов. Вы узнаете, как это было сделано, в главах 11 и 14.

глава-06/spanning-columns.html HTML

```
<table>
  <tr>
    <th></th>
    <th>9:00</th>
    <th>10:00</th>
    <th>11:00</th>
    <th>12:00</th>
  </tr>
  <tr>
    <th>Понедельник</th>
    <td colspan="2">География</td>
    <td>Музыка</td>
    <td>ИЗО</td>
  </tr>
  <tr>
    <th>Вторник</th>
    <td colspan="3">Физкультура</td>
    <td>География</td>
  </tr>
</table>
```

РЕЗУЛЬТАТ

	9am	10am	11am	12am
Понедельник	География		Музыка	ИЗО
Вторник	Физкультура			География

ОБЪЕДИНЕНИЕ СТРОК

HTML **глава-06/spanning-rows.html**

```
<table>
  <tr>
    <th></th>
    <th>TV1000</th>
    <th>THT</th>
    <th>CTC</th>
  </tr>
  <tr>
    <th>20:00-21:00</th>
    <td rowspan="2">Фильм</td>
    <td>Ток-шоу</td>
    <td>Юмор</td>
  </tr>
  <tr>
    <th>21:00-22:00</th>
    <td>Ток-шоу</td>
    <td>Мультфильм</td>
  </tr>
</table>
```

РЕЗУЛЬТАТ

	TV1000	THT	CTC
20:00-21:00	Фильм	Ток-шоу	Юмор
21:00-22:00		Ток-шоу	Мультфильм

Кроме того, вам может понадобиться сделать так, чтобы одна ячейка занимала несколько строк.

Атрибут `rowspan` может быть использован с элементами `<th>` и `<td>` для обозначения, на сколько строк должна простираться ячейка.

В примере слева вы можете видеть, что телеканал TV1000 показывает фильм с 18:00 до 20:00, а в это время на каналах THT и CTC транслируются по две развлекательные телепередачи (каждая из которых длится один час).

Если вы посмотрите на последний элемент `<tr>`, то увидите, что он содержит только три элемента, несмотря на то что в таблице на рисунке ниже четыре столбца. Это произошло потому, что в элементе `<tr>`, содержащем текст «Фильм», указан атрибут `rowspan`, благодаря которому он «поглотил» расположенную ниже ячейку.

Для иллюстрации этого примера мы использовали несколько параметров стиля CSS с тем, чтобы показать вам, как одна ячейка может занять сразу несколько строк. Вы узнаете, как это было сделано, в главах 11 и 14.

ДЛИННЫЕ ТАБЛИЦЫ

В языке HTML существуют три элемента, помогающих отличить основной контент страницы от первой и последней ее строк (которые иногда содержат принципиально отличающуюся информацию). Эти элементы полезны людям, использующим программы экранного доступа, а кроме того, они помогают вам выделять данные разделы таблицы особым стилем (что вы научитесь делать, начав изучать CSS).

<thead>

Заголовки таблицы должны быть помещены в элемент <thead>.

<tbody>

Основное содержимое (тело) таблицы должно находиться внутри элемента <tbody>.

<tfoot>

Последняя, завершающая, строка таблицы должна быть указана внутри элемента <tfoot>.
По умолчанию очень немногие браузеры хоть как-либо выделяют содержимое этих трех элементов, однако веб-дизайнеры изменяют их внешний вид с помощью каскадных таблиц стилей (CSS).

глава-06/long-tables.html HTML

```
<table>
  <thead>
    <tr>
      <th>Дата</th>
      <th>Прибыль</th>
      <th>Расход</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <th>1-е января</th>
      <th>250</th>
      <th>36</th>
    </tr>
    <tr>
      <th>2-е января</th>
      <th>285</th>
      <th>48</th>
    </tr>
    <!-- прочие строки как предыдущие две -->
    <tr>
      <th>31-е января</th>
      <th>129</th>
      <th>64</th>
    </tr>
      </tbody>
      <tfoot>
    <tr>
      <td></td>
      <th>7824</th>
      <th>1241</th>
    </tr>
  </tfoot>
</table>
```

РЕЗУЛЬТАТ

Дата	Прибыль	Расход
1-е января	250	36
2-е января	285	48
3-е января	260	42
4-е января	290	38
5-е января	310	115
6-е января	168	14
7-е января	226	20
8-е января	253	37
9-е января	294	33
10-е января	216	46
11-е января	244	29
12-е января	297	32
13-е января	328	86
14-е января	215	38
15-е января	254	30
16-е января	256	27
17-е января	311	68
18-е января	212	39
19-е января	234	36
20-е января	221	43
21-е января	259	38
22-е января	246	31
23-е января	248	17
24-е января	229	45
25-е января	263	34
26-е января	258	41
27-е января	283	22
28-е января	256	30
29-е января	278	47
30-е января	251	15
31-е января	129	64
	7824	1241

Часть HTML-редакторов, встроенных в системы управления контентом (CMS), обладает инструментами, которые позволяют создавать таблицы. Если первая строка вашей таблицы содержит только элементы `<th>`, то вы можете увидеть, как редактор автоматически вставит элемент `<thead>`.

Одна из причин использования элементов `<thead>` и `<tfoot>` заключается в том, что если ваша таблица слишком длинная для единовременного отображения на экране (или для печати), то браузер будет отображать и заголовок (`<thead>`) и последнюю строку (`<tfoot>`), когда пользователь станет прокручивать вашу таблицу. Так делается, чтобы пользователи могли видеть, в каких столбцах находятся отображаемые данные. Однако эта функция присутствует не во всех современных браузерах.

Для иллюстрации этого примера мы использовали несколько параметров стиля CSS с тем, чтобы показать вам, что содержимое элементов `<thead>` и `<tfoot>` может быть выведено на экран с собственным стилем. Вы узнаете, как использовать эти стили CSS, в главах 13 и 14.

СТАРЫЙ КОД: ШИРИНА И ПРОМЕЖУТОК

Существует несколько устаревших атрибутов, вытесненных каскадными таблицами стилей (CSS). Использование этих атрибутов не рекомендуется, однако вы можете встретить их при просмотре исходного кода старых сайтов, потому мы решили все же коротко рассказать о них.

Атрибут `width*` используется в открывающем теге `<table>` для указания ширины таблицы. Кроме того, его можно ставить в открывающих тегах элементов `<th>` и `<td>` для задания ширины отдельных ячеек таблицы. Значение этого атрибута — ширина таблицы или ячейки, указанная в пикселах.

Так как столбцы таблицы должны быть ровными, то чаще всего вы встретите данный атрибут только в первой строке таблицы (все последующие строки наследуют это значение).

В открывающем теге элемента `<table>` также может быть указан атрибут `cellpadding`, добавляющий промежуток между границами ячейки и ее содержимым, а также атрибут `cellspacing`, задающий величину промежутка между ячейками таблицы. Значения этих атрибутов устанавливаются в пикселах.

Для иллюстрации этого примера мы использовали несколько параметров стиля CSS с тем, чтобы вы могли более четко видеть ширину ячеек таблицы. Если вы хотите изменять ширину ячеек или устанавливать промежутки между ними или внутри них, то вы должны использовать соответствующие правила CSS, описанные в главах 13 и 14.

глава-06/width-and-spacing.html HTML

```
<table width="400" cellpadding="10"
cellspacing="5">
  <tr>
    <th width="150"></th>
    <th>Снято с карты</th>
    <th>Кредит</th>
    <th width="150">Баланс</th>
  </tr>
  <tr>
    <th>Январь</th>
    <td>2500,00</td>
    <td>6600,50</td>
    <td>4100,50</td>
  </tr>
  <tr>
    <th>Февраль</th>
    <td>1350,50</td>
    <td>8950,20</td>
    <td>11700,10</td>
  </tr>
</table>
```

РЕЗУЛЬТАТ

	Снято с карты	Кредит	Баланс
Январь	2500,00	6600,50	4100,50
Февраль	1350,50	8950,20	11700,10

* Width (*англ.*) — ширина. — *Примеч. ред.*

СТАРЫЙ КОД: ГРАНИЦЫ И ЗАЛИВКА

HTML **глава-06/border-and-background.html**

```
<table broder="2" bgcolor="#efefef">
  <tr>
    <th width="150"></th>
    <th>Снято с карты</th>
    <th>Кредит</th>
    <th width="150" bgcolor="#cccccc">Баланс</th>
  </tr>
  <tr>
    <th>Январь</th>
    <td>2500,00</td>
    <td>6600,50</td>
    <td bgcolor="#cccccc">4100,50</td>
  </tr>
  <tr>
    <th>Февраль</th>
    <td>1350,50</td>
    <td>8950,20</td>
    <td bgcolor="#cccccc">11700,10</td>
  </tr>
</table>
```

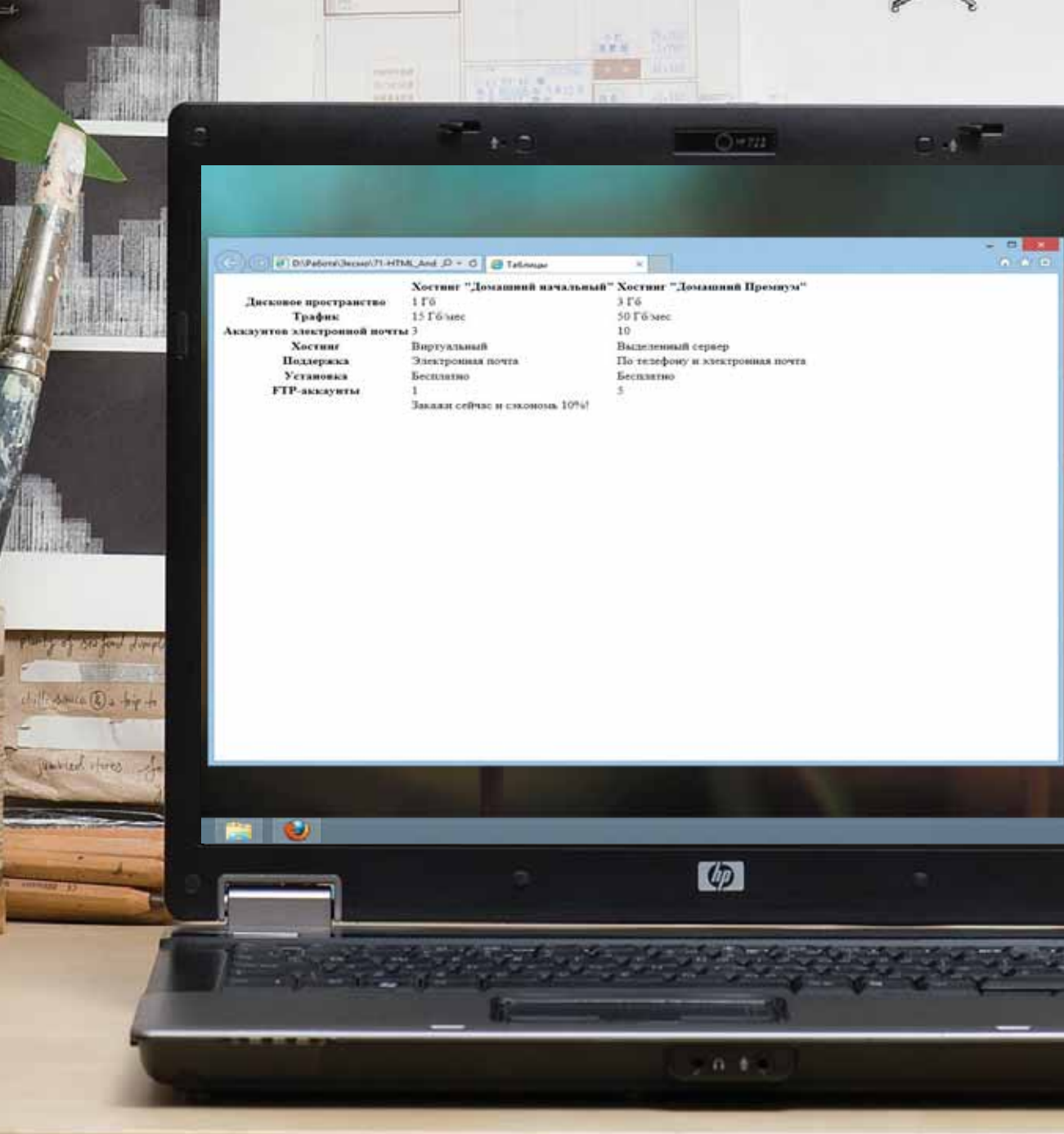
РЕЗУЛЬТАТ

	Снято с карты	Кредит	Баланс
Январь	2500,00	6600,50	4100,50
Февраль	1350,50	8950,20	11700,10

Атрибут `border` используется с элементами `<table>` и `<th>` для установки ширины границы в пикселах.

Атрибут `bgcolor` используется для установки цвета всей таблицы либо отдельных ячеек. Его значением, как правило, является шестнадцатеричный код цвета (который мы обсудим в главе 11).

В этом примере мы не использовали CSS, только атрибуты `border` и `bgcolor` языка HTML. При создании нового сайта для настройки оформления таблиц рекомендуется использовать CSS, а не эти атрибуты. Мы говорим о них только лишь потому, что вы можете встретить их, просматривая исходный код старых сайтов.



В этом примере мы создадим таблицу, демонстрирующую клиентам разные тарифные планы на услуги размещения сайта на сервере (хостинг). Первая строка и первый столбец таблицы представляют собой заголовки. Несмотря на то, что верхняя левая ячейка не содержит информации, для нее все равно создан отдельный элемент `<th>`. Каждая ячейка таблицы должна быть представлена либо элементом `<th>`, либо `<td>`. Для элемента `<th>` можно установить атрибут `scope`, чтобы указать к чему относится данный заголовок: к строке или столбцу. В последней строке таблицы используется атрибут `colspan` для объединения всех ее ячеек в одну.

ПРИМЕР
ТАБЛИЦЫ

```
<html>
  <head>
    <title>Таблицы</title>
  </head>
    <body>
      <table>
        <thead>
          <tr>
            <th></th>
            <th scope="col">Хостинг "Домашний начальный"</th>
            <th scope="col">Хостинг "Домашний Премиум"</th>
            </tr>
        </thead>
        <tbody>
          <tr>
            <th scope="row">Дисковое пространство</th>
            <td>1 Гб</td>
            <td>3 Гб</td>
          </tr>
          <tr>
            <th scope="row">Трафик</th>
            <td>15 Гб/мес</td>
            <td>50 Гб/мес</td>
          </tr>
          <!-- еще несколько строк, подобных первым двум -->
        </tbody>
        <tfoot>
          <tr>
            <td></td>
            <td colspan="2">Закажи сейчас и сэкономь 10%!</td>
          </tr>
        </tfoot>
      </table>
    </body>
</html>
```

ОБЗОР
ТАБЛИЦЫ

- Для добавления таблицы на веб-страницу используется элемент <table>.
- Таблица размещается на странице построчно. Каждая строка создается с помощью элемента <tr>.
- Внутри каждой строки есть несколько элементов <td>, представляющих ячейки (или элементов <th>, если данная строка — это шапка таблицы).
- С помощью атрибутов rowspan и colspan можно объединить несколько ячеек по вертикали или по горизонтали, соответственно.
- При создании длинной таблицы вы можете разбить ее содержимое на заголовок, тело и последнюю строку с помощью элементов <thead>, <tbody> и <tfoot>.

7

ФОРМЫ

- Как собирать информацию с посетителей
- Различные элементы формы
- Новые элементы формы в языке HTML5

Исходно термин «форма» обозначал документ с полями для заполнения от руки.

Разработчиками языка HTML концепция формы была позаимствована для обозначения набора элементов, позволяющих собирать данные, вводимые посетителями сайтов.

HTML-формы предоставляют весь необходимый набор элементов для сбора пользовательских данных, независимо от того, хотите ли вы добавить простое поле ввода поискового запроса или же создаете сложное приложение для заполнения заявления на получение паспорта. В этой главе вы узнаете, как:

- создавать формы на сайте;
- пользоваться различными инструментами для сбора данных;
- применять новые элементы формы языка HTML5.

Данные пользователя:
Имя:
Email:
Ваше мнение:
Как вы узнали о нас?
Посетите ли вы наш сайт снова?
Да Нет Возможно
Комментарий
Я хочу получать рассылку с новостями сайта
MacBook Pro

ПОЧЕМУ ФОРМЫ?

Наверное, самая известная веб-форма — это поле ввода поискового запроса, находящаяся в центре главной страницы сайта поисковой системы Google.

Однако помимо поиска формы позволяют пользователям совершать в Интернете множество других действий. Вы видите формы при регистрации на каком-либо сайте, при совершении покупок в интернет-магазинах, а также когда подписываетесь на рассылку с новостями.

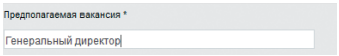
ЭЛЕМЕНТЫ ФОРМЫ

Существует несколько разновидностей элементов формы, с помощью которых вы можете собирать информацию от посетителей вашего сайта.

ВВОД ТЕКСТА

Поле ввода текста (одна строка)

Позволяет ввести строку текста, например адрес электронной почты или имя пользователя.



Поле ввода пароля

Оно аналогично полю ввода текста с той лишь разницей, что поле ввода пароля скрывает вводимые символы от посторонних глаз.

Текстовая область (многострочное поле ввода текста)

Данная разновидность поля используется для отправки более длинного текста, например сообщения или комментария.

ВЫБОР ВАРИАНТОВ

Переключатели

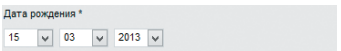
Используются, когда пользователь должен выбрать только один вариант из нескольких представленных.

Флажки

Используется, когда пользователь может выбрать несколько вариантов из предложенных либо не выбрать ни одного.

Раскрывающиеся списки

Используются, когда пользователь должен выбрать один вариант из списка



ОТПРАВКА ФОРМ НА СЕРВЕР

Кнопка подтверждения

Кнопка подтверждения используется для отправки введенных данных формы на другую веб-страницу.

Графические кнопки

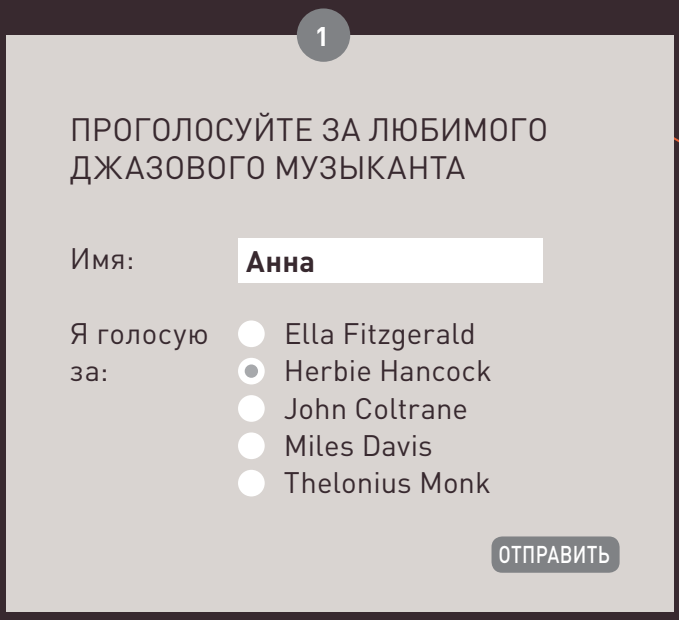
Они аналогичны обычным кнопкам подтверждения, но вместо стандартного изображения можно использовать собственное.

ЗАГРУЗКА ФАЙЛОВ НА САЙТ

Загрузка файлов

Данный элемент формы позволяет пользователям загружать на сайт файлы, например изображения.

Посетитель сайта заполняет поля формы, после чего щелкает мышью по кнопке подтверждения для отправки введенных данных на сервер.



На сервер отправляется имя каждого элемента формы и значение, введенное или выбранное посетителем

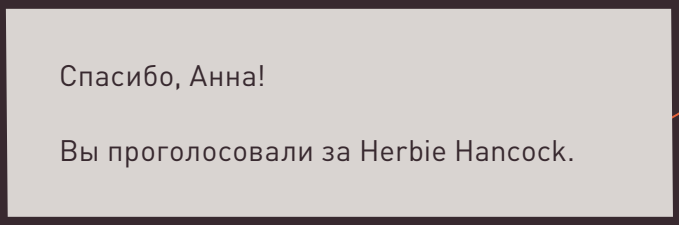
2

3

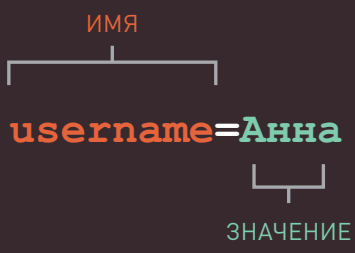
Сервер обрабатывает полученную информацию с помощью языков программирования, таких как PHP, C#, VB.net или Java. Кроме того, он может сохранить информацию в базе данных

4

Основываясь на полученной информации, сервер создает новую веб-страницу и возвращает ее браузеру



В форме может быть размещено несколько элементов для сбора различной информации. Серверу необходимо знать, какому элементу формы соответствует каждое из введенных значений .



Для разделения введенных данных вся информация посылается на сервер в виде пар «имя/значение». В приведенном выше примере форма запрашивает ввод имени пользователя, а также выбор любимого джазового музыканта. В данном случае сервер получает следующие пары «имя/значение».

username=Анна

Если элемент формы позволяет посетителю ввести какой-либо текст, то он и будет являться его значением.

vote=Herbie

Если элемент формы позволяет выбрать один пункт из представленного списка ответов (например, с помощью группы переключателей, флажков или раскрывающегося списка), то веб-дизайнеру необходимо ввести код, автоматически присваивающий значение каждому пункту.

Никогда не изменяйте имя элемента формы, если не уверены в том, что веб-сервер распознает новый вариант.

СТРУКТУРА ФОРМЫ

<form>

Все элементы формы помещаются в тег <form>. Для него всегда должен быть указан атрибут action, кроме того, для этого элемента часто устанавливают атрибуты method и id.

глава-07/form-structure.html HTML

```
<form action="http://www.primer.ru/subscribe.php"
  method="get">
  <p>Здесь будут отображены элементы формы.</p>
</form>
```

РЕЗУЛЬТАТ

Здесь будут отображены элементы формы.

action

Каждый элемент <form> требует указания атрибута action, значением которого является URL-адрес страницы-получателя введенных данных при отправке формы на сервер.

method

Формы можно отправить одним из двух методов: get или post.

При использовании метода get введенные данные будут добавлены в конец URL-адреса, указанного в атрибуте action. Метод get идеален:

- для коротких форм (таких как поля для ввода поискового запроса);
- при простом получении данных с веб-сервера (без отправки сведений, добавляемых в базу данных).

При использовании метода post данные отправляются на сервер с помощью так называемых HTML-заголовков. Хорошей практикой будет использование метода post в случае, если ваша форма:

- позволяет пользователям загружать файлы на сайт;
- очень длинная;
- содержит конфиденциальные данные (например, пароли);
- добавляет сведения в базу данных или удаляет их оттуда.

Если атрибут method не указан, то форма будет отправлена на сервер с помощью метода get.

id

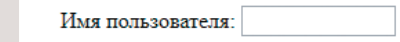
Мы обсудим атрибут id в главе 8. Сейчас лишь скажем, что его значение используется для различения формы и других элементов разметки страницы (зачастую этот атрибут используется сценариями, например, для проверки заполнения всех обязательных полей формы).

ВВОД ТЕКСТА

HTML **глава-07/text-input.html**

```
<form action="http://www.primer.ru/login.php">
  <p>Имя пользователя:
    <input type="text" name="username" size="15"
    maxlength="30" />
  </p>
</form>
```

РЕЗУЛЬТАТ



maxlength

Вы можете использовать атрибут `maxlength` для ограничения количества символов, которое посетителю разрешается ввести в данное поле. Например при запросе года вы можете ограничить количество символов, установив значение атрибута `maxlength` равным `4`.

size

При создании новых форм не следует указывать атрибут `size`. Он использовался ранее для установки ширины поля ввода текста (в качестве единицы измерения выступало количество видимых символов).

Так, например, значение `3` создавало поле ввода текста, ширина которого была достаточной для отображения трех введенных символов (хотя пользователи при желании могли ввести и большее количество).

Для управления шириной элементов новых форм рекомендуется использовать средства каскадных таблиц стилей (CSS). Мы упомянули атрибут `size` только потому, что вы можете встретиться с ним при просмотре исходного кода старых сайтов.

<input>

Элемент `<input>` используется для создания компонентов формы нескольких типов. Значение атрибута `type` указывает этот тип.

type="text"

Когда атрибуту `type` присваивается значение `text`, это значит, что будет создано однострочное поле ввода текста.

name

При введении пользователем данных в веб-форму сервер должен знать, какие сведения относятся к какому элементу формы (например, при обработке формы авторизации на сайте сервер должен знать, что было введено в качестве имени пользователя, а что — в качестве пароля). Поэтому для каждого элемента формы следует указывать атрибут `name`*. Значение этого атрибута служит для идентификации элемента формы и отправляется на сервер вместе с введенными данными.

* Name (*англ.*) — имя. — *Примеч. ред.*

ВВОД ПАРОЛЯ

<input>

type="password"

Когда атрибуту `type` присваивается значение `password`*, на странице создается однострочное поле ввода текста, которое функционирует как обычное поле ввода текста с той лишь разницей, что вводимые в него символы маскируются. Это сделано, чтобы не допустить утечки таких важных данных, как пароль, в случае, если кто-то подглядывает из-за спины посетителя сайта.

name

Атрибут `name` устанавливает имя поля ввода пароля, отправляемое на сервер вместе с паролем, введенным пользователем.

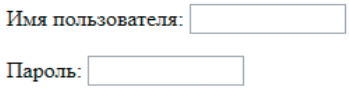
size, maxlength

Аналогично однострочному полю ввода текста, для поля ввода пароля также можно указать атрибуты `size` и `maxlength`.

глава-07/password-input.html HTML

```
<form action="http://www.primer.ru/login.php">
  <p>Имя пользователя:
    <input type="text" name="username" size="15"
    maxlength="30" />
  </p>
  <p>Пароль:
    <input type="password" name="password" size="15"
    maxlength="30" />
  </p>
</form>
```

РЕЗУЛЬТАТ



Несмотря на то, что вводимый пароль скрыт от чужих глаз на экране, это вовсе не означает, что он будет безопасно передан на сервер. Никогда не используйте данный элемент формы для отправки таких конфиденциальных данных, как номера банковских карт.

Для обеспечения полной безопасности сервер должен быть настроен таким образом, чтобы обмениваться данными с пользовательскими компьютерами по криптографическому протоколу SSL* (Secure Sockets Layer — Уровень защищенных сокетов). Обсуждение протокола SSL выходит за рамки этой книги.

* Password (*англ.*) — пароль. — *Примеч. ред.*

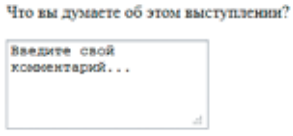
* Secure sockets layer (*англ.*) — уровень защищенных сокетов. — *Примеч. ред.*

ТЕКСТОВАЯ ОБЛАСТЬ

HTML **глава-07/textarea.html**

```
<form action="http://www.primer.ru/profile.php">
  <p>Что вы думаете об этом выступлении?</p>
  <textarea name="comments" cols="20"
    rows="4">Введите свой комментарий...</textarea>
</form>
```

РЕЗУЛЬТАТ



При создании новой формы используйте правила CSS для установки ширины и высоты текстовой области. Однако при просмотре исходного кода старых сайтов вы можете встретить атрибуты `cols` и `rows`.

Атрибут `cols` устанавливает ширину текстовой области (измеряется в количестве символов в одной строке). Атрибут `rows` задает высоту поля в строках.

<textarea>

Элемент `<textarea>`* используется для создания многострочных областей ввода текста. В отличие от других элементов формы, он не является пустым, а следовательно, для него должны следует указать открывающий и закрывающий теги.

Текст, помещенный между ними, будет выведен в поле при загрузке страницы.

Если посетитель не удалит этот текст, то он отправится на сервер вместе с введенными им данными. (На некоторых сайтах реализован сценарий на языке JavaScript, очищающий текстовую область при щелчке мышью по ней.)

* Text area (*англ.*) — область текста. — *Примеч. ред.*

ПЕРЕКЛЮЧАТЕЛИ

<input>

type="radio"

Переключатели позволяют посетителям сайта выбрать только один вариант из предложенных.

name

Атрибут `name` отправляется серверу вместе со значением выбранного переключателя. Когда ответ на вопрос подразумевает выбор одного из нескольких вариантов, все переключатели, формирующие список ответов, должны иметь одинаковое значение атрибута `name`.

value

Атрибут `value` устанавливает значение выбранного переключателя, отправляемое серверу. Значение каждого переключателя должно быть уникальным внутри группы (чтобы сервер знал, какой вариант ответа выбрал посетитель).

checked

Атрибут `checked` указывает, какой из предлагаемых вариантов должен быть выбран по умолчанию при загрузке страницы (если это необходимо). Атрибут принимает единственное значение — `checked`*. Данный атрибут должен быть установлен только у одного переключателя из группы.

глава-07/radio-button.html HTML

```
<form action="http://www.primer.ru/profile.php">
  <p>Пожалуйста, выберите любимый жанр музыки:
    <br />
    <input type="radio" name="genre" value="rock"
checked="checked" /> Рок
    <input type="radio" name="genre" value="pop" />
Поп
    <input type="radio" name="genre" value="techno"
/> Электроника
  </p>
</form>
```

РЕЗУЛЬТАТ

Пожалуйста, выберите любимый жанр музыки:
◉ Рок ○ Поп ○ Электроника

Примечание. Если переключатель был установлен, сбросить его уже нельзя. Единственное, что может сделать пользователь, — выбрать другой вариант ответа. Если ваш вопрос содержит один-единственный ответ и вы хотите предоставить посетителю возможность сбросить переключатель (например, для обозначения того, что пользователь не согласен с условиями договора), вместо переключателей используйте флажки.

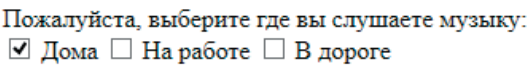
* Checked (*англ.*) — установлен. — *Примеч. ред.*

ФЛАЖКИ

HTML **глава-07/checkbox.html**

```
<form action="http://www.example.com/profile.php">
  <p>Пожалуйста, выберите, где вы слушаете музыку:
    <br />
    <input type="checkbox" name="service"
    value="house" checked="checked" /> Дома
    <input type="checkbox" name="service"
    value="work" /> На работе
    <input type="checkbox" name="service"
    value="inroad" /> В дороге
  </p>
</form>
```

РЕЗУЛЬТАТ



<input>

type="checkbox"

С помощью флажков посетители сайта могут выбирать из нескольких вариантов ответа, а также отменять выбор.

name

Атрибут `name` отправляется на сервер вместе со значением выбранных посетителем вариантов ответа.
В данном случае значение атрибута `name` должно быть одинаковым для всех флажков группы.

value

Атрибут `value` указывает значение, отправляемое серверу при установке данного флажка.

checked

Атрибут `checked` указывает, что данный флажок должен быть установлен по умолчанию при загрузке страницы.
Атрибут принимает единственное значение — `checked`.

РАСКРЫВАЮЩИЙСЯ СПИСОК

<select>

Раскрывающийся список появляется при щелчке мышью по элементу формы и позволяет посетителю сайта выбрать один вариант.
Для создания раскрывающегося списка используется элемент <select>. Он может содержать два и более элементов <option>.

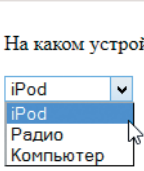
name

Атрибут name определяет имя элемента формы, отправляемое на сервер вместе со значением, выбранным посетителем.

глава-07/drop-down-list-box.html HTML

```
<form action="http://www.example.com/profile.php">
  <p> На каком устройстве вы слушаете музыку чаще
      всего?</p>
  <select name="ustroistva">
    <option value="ipod">iPod</option>
    <option value="radio">Радио</option>
    <option value="computer">Компьютер</option>
  </select>
</form>
```

РЕЗУЛЬТАТ



<option>

Элемент <option> используется для указания вариантов ответа, которые может выбрать посетитель сайта. Текст, помещенный между тегами <option> и </option>, будет выведен на экран как пункт раскрывающегося списка.

value

Атрибут value используется в элементе <option> для установки значения, отправляемого на сервер вместе с именем выбранного элемента.

selected

Атрибут selected устанавливает вариант ответа, который должен быть выбран по умолчанию во время загрузки страницы. Данный атрибут принимает единственное значение — selected.
Если атрибут selected не установлен, то при загрузке страницы будет отображен первый вариант ответа из списка. Если пользователь не выберет другого, то на сервер будет отправлен именно он в качестве значения данного элемента формы.

Функционирование раскрывающегося списка аналогично переключателям (может быть выбран только один вариант). Однако у каждого элемента есть свое преимущество:

- если необходимо, чтобы посетитель сайта видел сразу все варианты ответа, то лучше использовать переключатели;
- если список вариантов достаточно длинный (как, например, список стран), то лучше подходит раскрывающийся список.

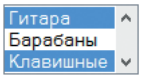
СПИСОК МНОЖЕСТВЕННОГО ВЫБОРА

HTML **глава-07/multiple-select-box.html**

```
<form action="http://www.primer.ru/profile.php">
  <p> Играете ли вы на одном из инструментов,
      перечисленных ниже? (Вы можете выбрать сразу
      несколько вариантов ответа, нажав и удерживая
      клавишу <b>Ctrl</b> (Windows) или <b>Cmd</b>
      (OS X)).</p>
  <select name="instrumenty" size="3"
      multiple="multiple">
    <option value="gitara"
      selected="selected">Гитара</option>
    <option value="barabany">Барабаны</option>
    <option value="klavishnye"
      selected="selected">Клавишные</option>
    <option value="bas">Бас-гитара</option>
  </select>
</form>
```

РЕЗУЛЬТАТ

Играете ли вы на одном из инструментов, перечисленных ниже? (Вы можете выбрать сразу несколько вариантов ответа, нажав и удерживая клавишу **Ctrl** (Windows) или **Cmd** (OS X)).



<select>

size

Добавив в элемент `<select>` атрибут `size`, вы превратите раскрывающийся список в поле, отображающее сразу несколько вариантов выбора. Значением этого атрибута должно быть количество пунктов списка, отображаемых за раз. В примере одновременно видны три из четырех вариантов ответа.

К сожалению, не все браузеры правильно отображают списки с атрибутом `size`, потому работоспособность сайта нужно тщательно проверять (особенно это касается браузеров Firefox и Safari на компьютерах с операционной системой OS X).

multiple

Добавив атрибут `multiple` со значением `multiple`, вы позволите посетителям вашего сайта выбирать сразу несколько вариантов ответа (не забудьте написать предупреждение об этой возможности).

Также полезно будет упомянуть о том, что пользователи компьютеров под управлением операционной системы Windows могут выбрать несколько пунктов списка, нажав и удерживая клавишу **Ctrl**, а пользователи компьютеров Mac — с помощью клавиши **Cmd**.

ЗАГРУЗКА ФАЙЛОВ НА СЕРВЕР

<input>

Если вы хотите позволить посетителям размещать файлы на вашем сайте, например изображения, видеоролики, MP3-записи или документы в формате PDF, вам понадобится специальное поле для загрузки файлов.

type="file"

Данное значение атрибута `type` создает поле, по внешнему виду напоминающее поле ввода текста, после которого помещается кнопка **Обзор** (Browse). Когда посетитель щелкает по ней мышью, браузер открывает диалоговое окно, позволяющее выбрать нужный файл на компьютере.

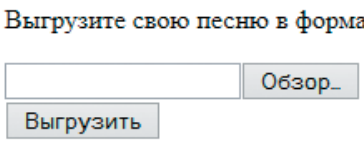
При создании формы, позволяющей загружать файлы на сервер, всегда указывайте метод `post` (значение атрибута `method` элемента `<form>`). Передача файлов на сервер невозможна при использовании метода `get`.

Внешний вид диалогового окна, выводимого при щелчке мышью по кнопке **Обзор** (Browse) и позволяющего посетителю выбрать нужный файл для загрузки, определяется установленной на компьютере посетителя операционной системой. Вы не можете изменить интерфейс этого диалогового окна.

глава-07/file-input-box.html HTML

```
<form action="http://www.primer.ru/zagruzka.php"
    method="post">
    <p>Выгрузите свою песню в формате MP3:</p>
    <input type="file" name="pesnya" /><br />
    <input type="submit" value="Загрузить" />
</form>
```

РЕЗУЛЬТАТ

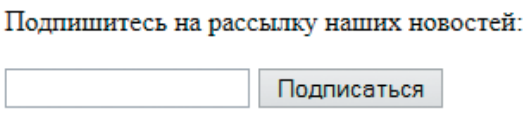


КНОПКА ПОДТВЕРЖДЕНИЯ

HTML **глава-07/submit-button.html**

```
<form action="http://www.primer.ru/podpiska.php">
  <p>Подпишитесь на рассылку наших новостей:</p>
  <input type="text" name="email" />
  <input type="submit" name="podpisatsya"
    value="Подписаться" />
</form>
```

РЕЗУЛЬТАТ



<input>

type="submit"

Кнопка подтверждения используется для отправки формы с данными на сервер.

name

Данному элементу формы может быть присвоен атрибут `name`, но его использование необязательно.

value

Атрибут `value` используется для указания текста надписи на кнопке. Устанавливать текст надписи самостоятельно нужно потому, что по умолчанию в некоторых браузерах на кнопке пишется **Отправить** (Submit), что подходит не для всех форм.

В разных браузерах стили кнопок подтверждения могут отличаться, чаще всего соответствуя интерфейсу самого браузера. Если вы хотите самостоятельно настроить стиль кнопки, вы можете воспользоваться средствами CSS (о которых вы прочитаете в главе 14), либо использовать графические кнопки.

ГРАФИЧЕСКАЯ КНОПКА

<input>

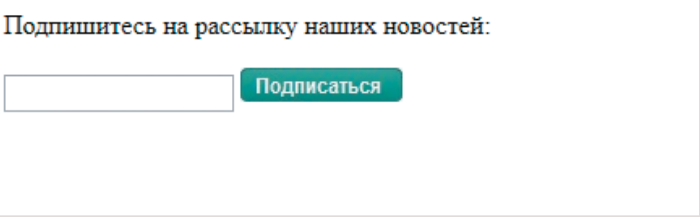
type="image"

Если вы хотите использовать собственное изображение для кнопки подтверждения, то вам потребуется присвоить атрибуту `type` значение `image`. При этом атрибуты `src`, `width`, `height` и `alt` будут работать точно так же, как и для элемента `<img>` (речь о котором шла в главе 5).

глава-07/image-button.html HTML

```
<form action="http://www.primer.ru/podpiska.php">
  <p>Подпишитесь на рассылку наших новостей:</p>
    <input type="text" name="email" />
    <input type="image" src="images/subscribe.jpg"
      width="100" height="20" />
</form>
```

РЕЗУЛЬТАТ

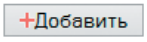


КНОПКИ И СКРЫТЫЕ ЭЛЕМЕНТЫ ФОРМЫ

HTML **глава-07/button-and-hidden-controls.html**

```
<form action="http://www.example.com/add.php">
  <button><img src="images/add.gif" alt="добавить"
    width="10" height="10" />Добавить</button>
  <input type="hidden" name="zakladka"
    value="tekst_pesni" />
</form>
```

РЕЗУЛЬТАТ



<button>

Элемент `<button>` был добавлен в язык HTML, чтобы позволить разработчикам сайтов получить более полный контроль над внешним видом кнопок, а также сделать возможным появление внутри кнопки других элементов.
Это значит, что теперь вы можете сочетать на кнопке текст и изображение, располагая их между открывающим `<button>` и закрывающим `</button>` тегами.

<input>

type="hidden"

В этом примере мы также продемонстрируем вам элемент формы `hidden`*. Скрытые элементы не отображаются, хотя вы и можете увидеть их при просмотре исходного кода страницы. Они позволяют веб-дизайнерам размещать в форме данные, невидимые посетителям. Например разработчик может сохранить в скрытом поле информацию о том, с какой страницы посетитель отправил форму на сервер.

*Hidden (*англ.*) — скрытый. — *Примеч. ред.*

МЕТКИ ЭЛЕМЕНТОВ ФОРМЫ

<label>

При рассказе об элементах формы мы намеренно упрощали код, поясняя назначение каждого элемента формы с помощью помещаемого рядом текста. Однако у каждого элемента формы должна быть своя надпись, или метка, создаваемая при помощи элемента `<label>` и упрощающая пользование сайтом людям с нарушениями зрения.

Элемент `<label>` можно использовать двумя способами.

Окружить им и метку и сам элемент формы, как показано справа на примере поля ввода текста.

Указать отдельно от элемента формы, но при этом использовать атрибут `for`* для соотнесения с нужным элементом, как показано на примере переключателей.

for

Данный атрибут устанавливает связь между `<label>` и элементом формы. Обратите внимание, что для группы переключателей указан атрибут `id`.

глава-07/labelling-form-controls.html HTML

```
<label>Возраст: <input type="text" name="vozrast" /></label>
<br />
Пол:
<input id="female" type="radio" name="pol" value="f">
<label for="female">Женский</label>
<input id="male" type="radio" name="pol" value="m">
<label for="male">Мужской</label>
```

РЕЗУЛЬТАТ

Возраст:
Пол: ○ Женский ○ Мужской

Значение атрибута `id` служит для отличия данного элемента разметки от всех прочих на странице. Для каждого элемента на странице значение этого атрибута должно быть уникальным. (Более подробно см. в главе 8.)

Значение атрибута `for` — это значение атрибута `id` требуемого элемента формы. Данный прием присвоения меток может быть использован со всеми элементами формы. Если элемент `<label>` используется с переключателем или флажком, то для выделения элемента формы посетитель может щелкнуть мышью либо по самому элементу формы, либо по элементу `<label>`. Местоположение метки принципиально важно: если посетители не знают, куда следует ввести запрашиваемую информацию или что именно нужно вводить, то велик шанс того, что форма окажется заполнена неправильно.

За годы создания веб-страниц были выделены следующие наилучшие позиции меток:

СВЕРХУ ИЛИ СЛЕВА:
- однострочное текстовое поле;
- текстовая область;
- списки;
- поля для загрузки файлов.

СПРАВА:
- отдельные флажки;
- отдельные переключатели.

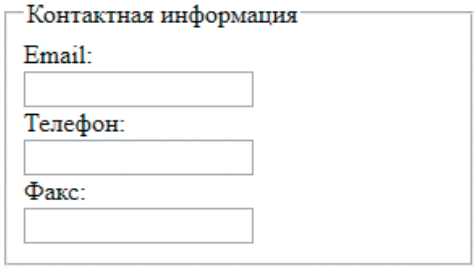
* For (*англ.*) — для. — *Примеч. ред.*

ГРУППИРОВКА ЭЛЕМЕНТОВ ФОРМЫ

HTML **глава-07/grouping-form-elements.html**

```
<fieldset>
  <legend>Контактная информация</legend>
  <label>Email:<br />
  <input type="text" name="email" /></label><br />
  <label>Телефон:<br />
  <input type="text" name="phone" /></label><br />
  <label>Факс:<br />
  <input type="text" name="fax" /></label>
</fieldset>
```

РЕЗУЛЬТАТ



<fieldset>

Элементы формы, связанные по смыслу, можно сгруппировать между тегами <fieldset> и </fieldset>. Это бывает особенно полезно при создании длинных форм. Большинство браузеров отобразят <fileldset> в виде рамки вокруг группы элементов формы. Внешний вид рамки может быть изменен с помощью каскадных таблиц стилей (CSS).

<legend>

Элемент <legend>, содержащий надпись для обозначения группы элементов формы, как правило, помещают сразу после открывающего тега элемента <fieldset>.

HTML5: ВАЛИДАЦИЯ ДАННЫХ ФОРМЫ

Возможно, вы встречали во Всемирной паутине формы, выводящие предупреждения в случае, если не все поля были правильно заполнены. Эта реакция называется **валидацией данных формы**.

Традиционно валидация введенных данных осуществляется с помощью специальных сценариев JavaScript (обсуждение которых не входит в рамки нашей книги).

В языке HTML5 появилась собственная технология валидации введенных данных: теперь эта работа перекладывается на веб-браузер.

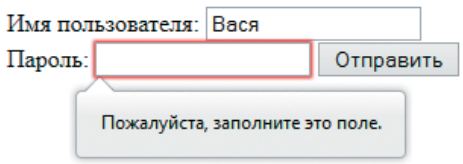
Валидация позволяет гарантировать, что пользователь введет данные в формате, с которым сможет работать сервер. Она позволяет:

- уменьшить количество операций, выполняемых сервером;
- увидеть пользователям ошибки в заполнении формы быстрее, чем если бы подобная проверка осуществлялась на сервере.

глава-07/html5-form-validation.html HTML

```
<form action="http://www.primer.ru/login/"
method="post">
  <label for="username">Имя пользователя:</label>
  <input type="text" name="imyapolz"
required="required" /><br />
  <label for="password">Пароль:</label>
  <input type="password" name="parol"
required="required" />
  <input type="submit" value="Отправить" />
</form>
```

РЕЗУЛЬТАТ



Для поддержки совместимости со старыми версиями браузеров (не поддерживающими HTML5) большинство разработчиков продолжают использовать сценарии JavaScript.

Примером встроенной системы проверки введенных данных HTML5 может служить атрибут `required`*, используемый с любым элементом формы, принимающим пользовательский ввод. Данному атрибуту HTML5 не нужно значение, однако в HTML4 всем атрибутам должно быть присвоено значение, поэтому веб-дизайнеры используют значение `required`.

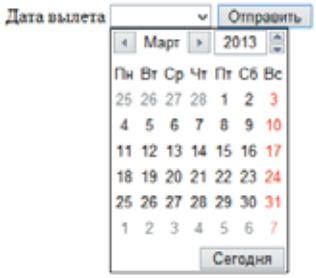
* Required (*англ.*) — обязательный. — *Примеч. ред.*

HTML5: ВВОД ДАТЫ

HTML **глава-07/html5-date-input.html**

```
<form action="http://www.primer.ru/bookings/"
    method="post"
  <label for="depart">Дата вылета</label>
  <input type="date" name="depart" />
  <input type="submit" value="Отправить">
</form>
```

РЕЗУЛЬТАТ



<input>

Во многие формы требуется вводить такую информацию, как даты, адреса электронной почты или URL. Обычно это реализуется с помощью полей ввода текста.

В языке HTML5 появились новые элементы формы, позволяющие стандартизировать ввод данных определенного типа. Более старые версии браузеров, не поддерживающие HTML5, будут визуализировать такие элементы формы просто как однострочные поля ввода текста.

type="date"

Для запроса ввода даты вы можете использовать элемент `<input>` со значением `date` атрибута `type`. Таким образом, в браузерах, поддерживающих элементы формы спецификации HTML5, будет выведено поле для ввода даты.

В примере показан внешний вид поля ввода данных в браузере Opera. Внешний вид этого элемента может меняться в зависимости от используемого браузера.

HTML5: ВВОД URL-АДРЕСА И АДРЕСА ЭЛЕКТРОННОЙ ПОЧТЫ

<input>

В спецификации HTML5 также появились специальные элементы формы, позволяющие посетителям сайта вводить адреса электронной почты и URL. Браузеры, не поддерживающие эти элементы формы, отобразят их как обычные однострочные поля ввода текста.

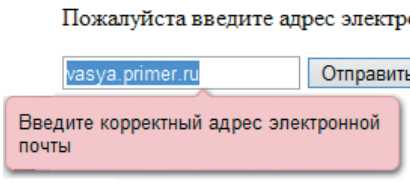
type="email"

Для запроса адреса электронной почты вы можете воспользоваться специализированным полем ввода. Браузеры, поддерживающие язык HTML5, проверят, соответствует ли введенный посетителем адрес принятому стандарту для данного типа адресов. Экранная клавиатура некоторых смартфонов также будет оптимизирована для отображения символов, наиболее часто встречающихся в адресах электронной почты (например, символа @).

глава-07/html5-email-input.html HTML

```
<form action="http://www.primer.ru/profile.php"
method="post"
  <p>Пожалуйста введите адрес электронной почты:</p>
  <input type="email" name="email" />
  <input type="submit" value="Отправить">
</form>
```

РЕЗУЛЬТАТ



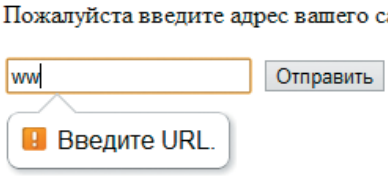
type="url"

Поле ввода URL-адресов может быть использовано для запроса ввода адреса какой-либо страницы во Всемирной паутине. Браузеры, поддерживающие валидацию данных HTML5, удостоверятся в том, что введенный текст соответствует стандарту для URL-адресов, прежде чем отправить форму на сервер. Экранная клавиатура некоторых смартфонов также будет оптимизирована для отображения символов, наиболее часто встречающихся в URL-адресах.

глава-07/html5-url-input.html HTML

```
<form action="http://www.primer.ru/profile.php"
method="post"
  <p>Пожалуйста введите адрес вашего сайта:</p>
  <input type="url" name="sait" />
  <input type="submit" value="Отправить">
</form>
```

РЕЗУЛЬТАТ



HTML5: ПОЛЕ ВВОДА ПОИСКОВОГО ЗАПРОСА

HTML **глава-07/html5-search-input.html**

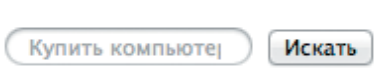
```
<form action="http://www.example.org/search.php">
  <p>Поиск:</p>
  <input type="search" name="poisk" />
  <input type="submit" value="Искать" />
</form>
```

РЕЗУЛЬТАТ

HTML **глава-07/html5-placeholder.html**

```
<form action="http://www.example.org/search.php">
  <p>Поиск:</p>
  <input type="search" name="poisk"
placeholder="Купить компьютер"/>
  <input type="submit" value="Искать" />
</form>
```

РЕЗУЛЬТАТ



<input>

В языке HTML5 существует специальный тип элементов формы, предоставляющий возможность создать поле ввода поискового запроса.

type="search"

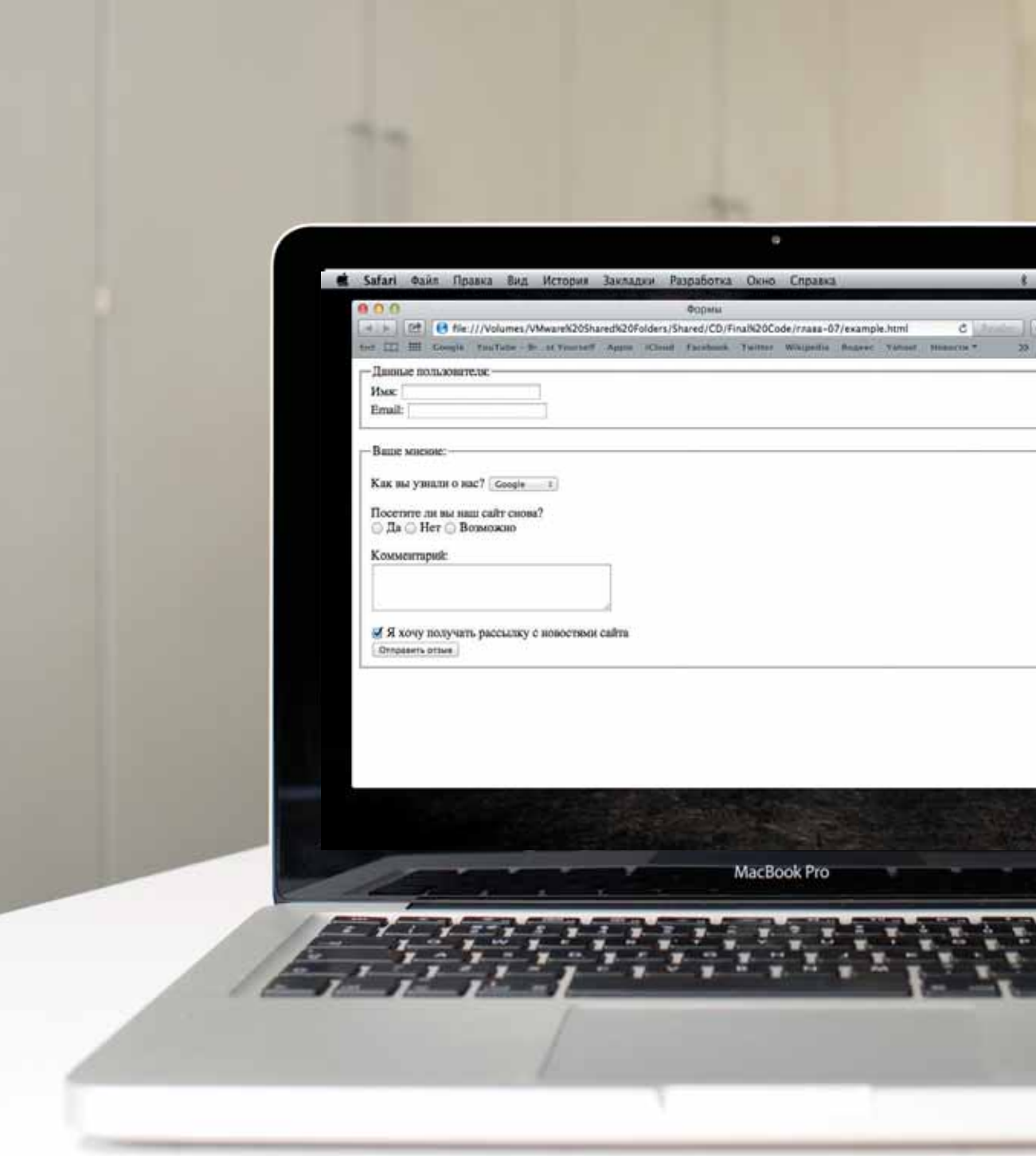
Для создания поля поискового запроса на языке HTML5 присвойте атрибуту `type` элемента `<input>` значение `search`. Старые версии браузеров отобразят этот элемент формы как простое поле ввода текста.

В новых версиях браузеров реализованы дополнительные возможности для работы с такими полями. Например, Safari для компьютеров Mac выводит дополнительный значок с крестиком, позволяющий очистить поле от введенной информации. Кроме того, в этом браузере поля ввода поискового запроса автоматически отображаются с закругленными углами.

placeholder

С любым полем ввода вы можете использовать атрибут `placeholder`*, значением которого является текст, отображаемый в поле до того, как пользователь введет свой запрос. Старые версии браузеров игнорируют этот атрибут.

* Placeholder (*англ.*) — заполнитель. — *Примеч. ред.*


Safari Файл Правка Вид История Закладки Разработка Окно Справка
Формы
file:///Volumes/VMware%20Shared%20Folders/Shared/CD/Final%20Code/глава-07/example.html
Данные пользователя:
Имя:
Email:
Ваше мнение:
Как вы узнали о нас? Google
Посетите ли вы наш сайт снова?
Да Нет Возможно
Комментарий:
Я хочу получать рассылку с новостями сайта
Отправить отзыв
MacBook Pro

ПРИМЕР
ФОРМЫ

Это пример демонстрирует создание формы для сбора мнений посетителей сайта и оформления подписки на новостную рассылку.

В элементе `<form>` используется атрибут `action`, указывающий адрес веб-страницы, на которую отправляются введенные данные. Все элементы формы находятся между тегами `<form>` и `</form>`. В форме используется несколько видов элементов, созданых для сбора различных данных; `<fieldset>` используется для группировки связанных по смыслу элементов формы; `<label>` указывает назначение каждого из элементов.

ПРИМЕР
ФОРМЫ

```
<html>
  <head>
    <title>Формы</title>
  </head>
  <body>
    <form action="http://www.example.com/review.php" method="get">
      <fieldset>
        <legend>
          Данные пользователя:
        </legend>
        <label>
          Имя:
          <input type="text" name="name" size="30" maxlength="100">
        </label>
        <br />
        <label>
          Email:
          <input type="email" name="email" size="30" maxlength="100">
        </label>
        <br />
      </fieldset>
      <br />
      <fieldset>
        <legend>
          Ваше мнение:
        </legend>
        <p>
          <label for="hear-about">
            Как вы узнали о нас?
          </label>
          <select name="referrer" id="hear-about">
            <option value="google">Google</option>
            <option value="friend">От друзей</option>
            <option value="advert">Реклама</option>
            <option value="other">Другое</option>
          </select>
        </p>
        <p>
```

ПРИМЕР
ФОРМЫ

```
        Посетите ли вы наш сайт снова?
        <br />
        <label>
          <input type="radio" name="rating" value="yes" />
          Да
        </label>
        <label>
          <input type="radio" name="rating" value="no" />
          Нет
        </label>
        <label>
          <input type="radio" name="rating" value="maybe" />
          Возможно
        </label>
      </p>
      <p>
        <label for="comments">
          Комментарий:
        </label>
        <br />
        <textarea rows="4" cols="40" id="comments">
        </textarea>
      </p>
      <label>
        <input type="checkbox" name="subscribe" checked="checked" />
        Я хочу получать рассылку с новостями сайта
      </label>
      <br />
      <input type="submit" value="Submit review" />
    </fieldset>
  </form>
 </body>
</html>
```

ОБЗОР
ФОРМЫ

- Для сбора данных с посетителей вашего сайта вам понадобится создать веб-форму, все элементы которой должны находиться между тегами <form> и </form>.
- Информация, введенная с помощью элементов формы, отправляется на сервер в виде пар «имя/значение».
- Каждому элементу формы присваивается имя. Текст, вводимый посетителями, или варианты ответа, выбираемые ими, отправляются на сервер.
- В языке HTML5 реализованы новые элементы формы, упрощающие ее заполнение.

8

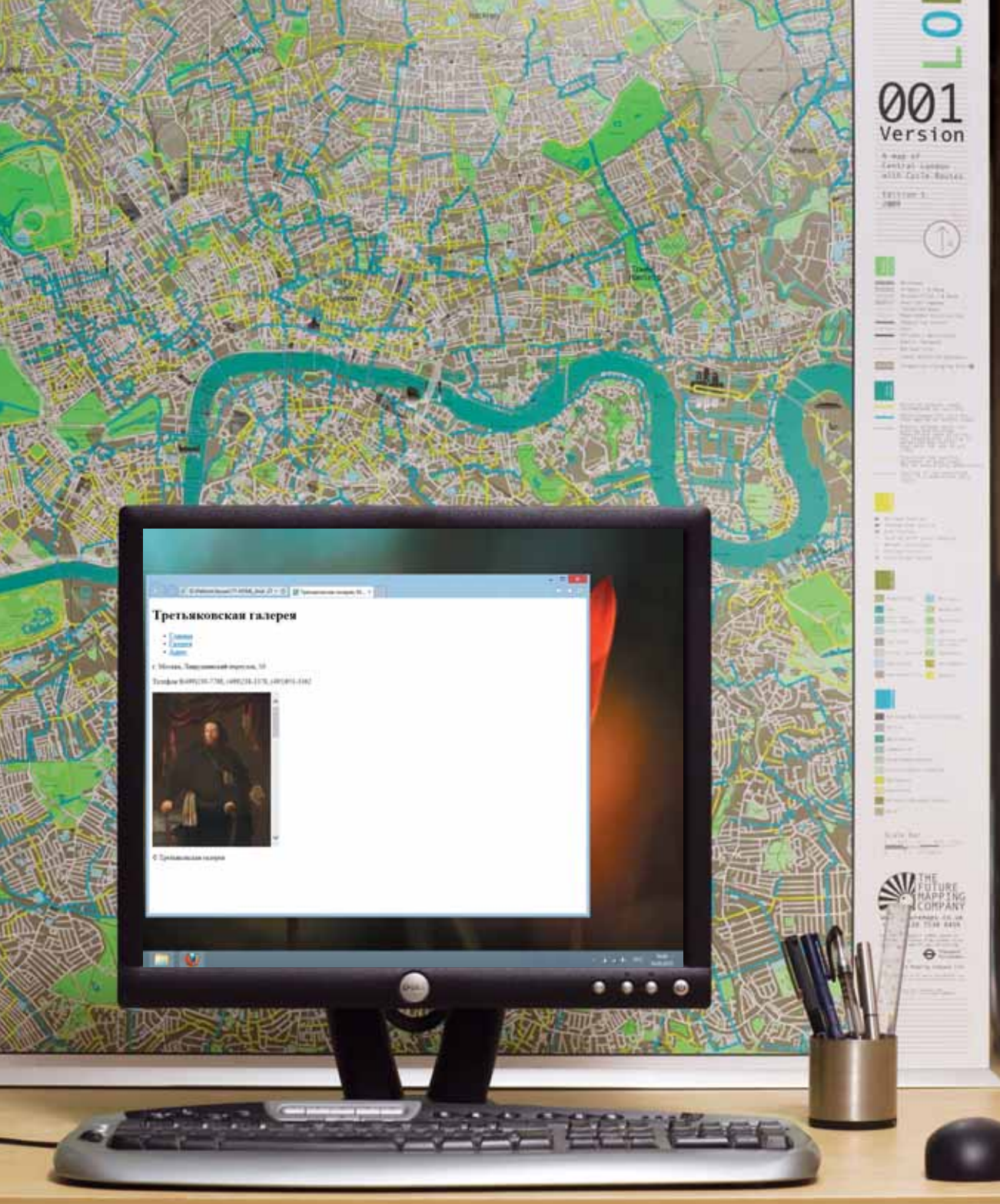
ДОПОЛНИТЕЛЬНАЯ РАЗМЕТКА

- Обозначение используемой версии языка HTML
- Идентификация и группировка элементов
- Комментарии, метаданные и плавающие фреймы

К этому моменту вы уже изучили основные теги, которые можно без труда разбить на группы.

В данной главе мы познакомим вас с несколькими полезными темами, которые не так легко классифицировать. Вы узнаете:

- о разных версиях языках HTML и о том, как обозначить, какую именно версию вы использовали при создании страницы;
- о том, как добавлять комментарии в код;
- о глобальных атрибутах, которые можно использовать со всеми элементами HTML, например `id` или `class`;
- об элементах, используемых для объединения различных частей страницы в группы, когда никакие другие элементы для этого не подходят;
- о том, как вставить одну страницу в другую с помощью элемента `<iframe>`;
- о том, как добавить информацию о странице с помощью элемента `<meta>`;
- о том, как вывести на экран специальные символы, такие как угловые скобки и знак защиты авторских прав.


001
Version
Третьяковская галерея

ЭВОЛЮЦИЯ ЯЗЫКА HTML

С момента появления Всемирной паутины было разработано несколько версий языка HTML.

Каждая следующая версия создавалась с целью улучшения предыдущей (появлялись новые элементы и атрибуты, уходили в прошлое устаревшие). Кроме того, для просмотра сайтов выпускались различные версии браузеров, каждая из которых получала возможность распознавать новый код. Впрочем, стоит заметить, что не у всех посетителей сайтов установлены последние версии браузеров, и это значит, что не у всех получится воспользоваться преимуществами новейших элементов разметки.
В случае, когда вам следует помнить о несовместимости возможностей HTML и старых версий браузеров, мы делаем соответствующие заметки (как при обсуждении новых элементов HTML5 в главе 7 и как будет в главах, посвященных CSS).

HTML 4

Версия реализована в 1997 году

За исключением некоторых, появившихся в HTML5, основная часть элементов, о которых шла речь в этой книге, была доступна в HTML4.
Несмотря на то что в версии HTML4 представлена группа элементов для управления оформлением страницы, их дальнейшее использование не рекомендуется. Примерами таких элементов могут быть `<center>`, позволяющий выровнять содержимое страницы по центру, `<font>`, использующийся для установки параметров шрифта, и `<strike>`, позволяющий перечеркнуть строку текста. Теперь все эти эффекты достигаются с помощью каскадных таблиц стилей (CSS).

XHTML 1.0

Версия реализована в 2000 году

В 1998 году появился новый язык под названием XML. Его целью было дать возможность создавать новые языки разметки. По причине того, что HTML считался самым популярным языком разметки гипертекста, разработчики решили, что HTML4 следует переосмыслить так, чтобы он соответствовал правилам XML, потому он был переименован в XHTML. Это означало, что веб-дизайнерам теперь пришлось следовать новым, более строгим правилам разметки — таким, как приведенные ниже.

- Каждый элемент должен иметь закрывающий тег (за исключением пустых элементов наподобие `<img />`).
- Имена атрибутов следует писать строчными буквами.
- Все атрибуты должны иметь значение, заключенное в двойные кавычки.
- Устаревшие элементы использовать нельзя.
- Любой элемент, открытый внутри другого элемента, должен быть закрыт внутри него же.

Все примеры из нашей книги полностью соответствуют этим жестким требованиям языка XML.
Одним из основных преимуществ столь значительных изменений стало то, что язык XHTML мог безошибочно работать с разными программами, написанными для создания и обработки XML-документов.
Этот язык также мог быть использован и с другими форматами данных, такими как SVG (язык для создания графики, разработанный на основе XML), MathML (язык для создания математических формул) и CML (язык для создания разметки химических формул).
Чтобы помочь веб-разработчикам перейти на новый синтаксис, были созданы две версии языка XHTML 1.0:

- **Strict XHTML 1.0** — строгая версия языка, при использовании которой разработчики должны были следовать всем новым правилам создания разметки;
- **Transitional XHTML 1.0** — переходная версия, в которой разработчики все еще могли использовать элементы управления оформлением страницы (такими как `<center>` или `<font>`).

Переходная версия языка XHTML была создана, чтобы позволить разработчикам продолжать использовать старые методы (с менее строгим синтаксисом), а также элементы и атрибуты, которые планировалось убрать из дальнейших версий языка HTML.
Кроме того, была создана еще и третья версия XHTML 1.0, получившая название **XHTML 1.0 Frameset**, позволявшая разделять окно браузера на несколько окон меньшего размера с использованием так называемых фреймов (что переводится с английского языка как «рамка»). Внутри каждого такого фрейма загружалась отдельная HTML-страница. Сегодня фреймы используются крайне редко и постепенно уходят в прошлое.

HTML5

Версия планируется к реализации в 2014 году

По стандартам языка HTML5, веб-дизайнерам не обязательно указывать закрывающий тег для всех элементов. Кроме того, в этой версии ожидются новые элементы и атрибуты. На момент написания книги спецификация HTML5 еще не была завершена, но разработчики программного обеспечения уже начали включать в свои браузеры возможности этого языка, кроме того, все большее и большее количество веб-дизайнеров начинает пользоваться новыми инструментами разметки.
Несмотря на то что создание языка HTML5 еще не завершено, вы можете без опаски использовать его нововведения — но только в том случае, если уверены, что посетители, использующие старые версии браузеров, сумеют увидеть контент вашего сайта (даже несмотря на то, что некоторые дополнительные возможности окажутся недоступными для них).

ТИПЫ ДОКУМЕНТОВ

Из-за существования нескольких версий языка HTML каждая веб-страница должна начинаться с объявления используемой версии языка при помощи элемента DOCTYPE. (Впрочем, большинство браузеров отобразит страницу и без этого элемента.) Потому указание используемой версии языка HTML будет включаться во все последующие примеры из этой книги.

Как вы увидите позже, при изучении каскадных таблиц стилей (CSS) и используемой ими модели разметки, использование объявления DOCTYPE способствует правильной обработке кода страницы браузером.

Так как язык XHTML создавался на основе XML, вы иногда можете встретить страницы с двойным объявлением DOCTYPE: там будет указан строгий XHTML и дополнительное объявление версии языка XML. В таком случае объявление версии XML должно идти первой строкой исходного кода страницы. Перед объявлением не должно быть ничего, даже пробела.

HTML5 HTML

```
<!DOCTYPE html>
```

HTML 4

```
<!DOCTYPE html PUBLIC
  "-//W3C//DTD HTML 4.01 Transitional//EN"
  "http://www.w3c.org/TR/html4/loose.dtd">
```

Переходный XHTML 1.0

```
<!DOCTYPE html PUBLIC
  "-//W3C//DTD XHTML 1.0 Transitional//EN"
  "http://www.w3c.org/TR/xhtml1/DTD/xhtml1-
    transitional.dtd ">
```

Строгий XHTML 1.0

```
<!DOCTYPE html PUBLIC
  "-//W3C//DTD XHTML 1.0 Strict//EN"
  "http://www.w3c.org/TR/xhtml1/DTD/
    xhtml1-strict.dtd ">
```

Объявление XML

```
<?xml version="1.0" ?>
```

КОММЕНТАРИИ В HTML

HTML **глава-08/comments-in-html.html**

```
<!-- начало введения -->
<h1>Текущие выставки</h1>
<h2>Олафур Элиассон</h2>
<!-- конец введения -->
<!-- начало основного текста -->
<p>Олафур Элиассон родился в Копенгагене в семье
  переселенцев из Исландии.</p>
<p>Широкую известность он получил благодаря своим
  скульптурам и крупномасштабным инсталляциям
  с использованием таких элементарных материалов, как
  свет, вода и температура воздуха для воздействия
  на зрительское восприятие.</p>
<!-- конец основного текста -->
<!-- <a href="mailto:info@primer.ru">Связаться</a>
-->
```

РЕЗУЛЬТАТ

Текущие выставки

Олафур Элиассон

Олафур Элиассон родился в Копенгагене в семье переселенцев из Исландии.

Широкую известность он получил благодаря своим скульптурам и крупномасштабным инсталляциям с использованием таких элементарных материалов, как свет, вода и температура воздуха для воздействия на зрительское восприятие.

<!-- -->

Если вы хотите включить в код какое-либо пояснение, которого не будет видно в браузерах посетителей, обозначьте нужный текст следующим образом:

```
<!-- текст комментария -->
```

Добавление комментариев рекомендуется, так как сколь хорошо бы вы ни знали код страницы на момент его написания, если вам придется вернуться к нему спустя какое-то время (или кому-либо еще понадобится просмотреть его), именно комментарии помогут разобраться, что есть что.

Хотя комментарии не видны посетителям в основном окне браузера, любой желающий сможет увидеть их, просмотрев исходный код. В объемных страницах вы часто встретите комментарии, поясняющие, где начинается или заканчивается тот или иной раздел, а также помогающие правильно интерпретировать код страницы.

Кроме того, можно комментировать части кода для предотвращения их отображения. В следующем примере вы увидите закомментированную гиперссылку.

АТРИБУТ ID

Каждый HTML-элемент может иметь атрибут `id`. Он используется для идентификации, поэтому его значение должно быть уникальным для каждого элемента на странице. Значение атрибута может начинаться с латинской буквы или со знака подчеркивания (но не с цифры или другого символа). Очень важно, чтобы значения атрибута `id` не повторялись на странице (в противном случае они не будут уникальными).

Как вы увидите при изучении каскадных таблиц стилей, присвоение элементу уникального значения атрибута `id` позволяет задавать параметры внешнего вида, отличающиеся от таковых у остальных элементов на странице. Так, вы можете выделить особым стилем один из абзацев текста (например, содержащий цитату). В примере справа абзац со значением `pullquote` атрибута `id` с помощью CSS выделен прописными буквами.

Если вы продолжите обучение знакомством с языком JavaScript (позволяющим сделать созданную вами страницу интерактивной), то увидите, что атрибуты `id` используются, чтобы написанный сценарий работал с конкретными элементами разметки.

Атрибут `id` называют **глобальным**, поскольку он может быть использован с любыми HTML-элементами.

глава-08/id-attribute.html HTML

```
<p>Вода и воздух. Как обыденны и банальны эти две
  субстанции: они едва ли привлекают к себе чье-то
  внимание, но, тем не менее, они обеспечивают наше
  существование.</p>
<p id="pullquote">Каждый раз, смотря на море, во
  мне рождается успокаивающее чувство защищенности,
  подобно тому, что появляется во время посещения
  родительского дома; я словно отправляюсь в морское
  путешествие, а это путешествие есть
  наблюдение.</p>
<p>Самая тайная тайна: вода и воздух здесь, рядом
  с нами: в море.</p>
```

РЕЗУЛЬТАТ

Вода и воздух. Как обыденны и банальны эти две субстанции: они едва ли привлекают к себе чье-то внимание, но, тем не менее, они обеспечивают наше существование.

КАЖДЫЙ РАЗ, СМОТРЯ НА МОРЕ, ВО МНЕ РОЖДАЕТСЯ УСПОКАИВАЮЩЕЕ ЧУВСТВО ЗАЩИЩЕННОСТИ, ПОДОБНО ТОМУ, ЧТО ПОЯВЛЯЕТСЯ ВО ВРЕМЯ ПОСЕЩЕНИЯ РОДИТЕЛЬСКОГО ДОМА; Я СЛОВНО ОТПРАВЛЯЮСЬ В МОРСКОЕ ПУТЕШЕСТВИЕ, А ЭТО ПУТЕШЕСТВИЕ ЕСТЬ НАБЛЮДЕНИЕ.

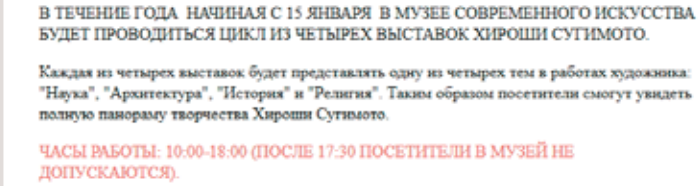
Самая тайная тайна: вода и воздух здесь, рядом с нами: в море.

АТРИБУТ CLASS

HTML **глава-08/class-attribute.html**

```
<p class="vazhno">В течение года начиная с 15
  января в Музее современного искусства будет
  проводиться цикл из четырех выставок Хироши
  Сугимото.</p>
<p>Каждая из четырех выставок будет представлять
  одну из четырех тем в работах художника: "Наука",
  "Архитектура", "История" и "Религия". Таким
  образом посетители смогут увидеть полную панораму
  творчества Хироши Сугимото.</p>
<p class="vazhno chasy_raboty">Часы работы:
  10:00-18:00 (после 17:30 посетители в музей не
  допускаются).</p>
```

РЕЗУЛЬТАТ



В языке HTML элементу также может быть присвоен атрибут `class`. Иногда вместо идентификации каждого элемента разметки на странице гораздо удобнее оказывается выделить группу элементов. Например введенный вами текст может содержать несколько абзацев, более важных, чем другие, или вы вдруг захотите выделить особым стилем ссылки, ведущие на страницы других сайтов и тем самым дать возможность посетителям отличать их от внутренних ссылок. Чтобы сделать это, следует воспользоваться атрибутом `class`. Его значение должно описывать, к какому классу принадлежит элемент. В примере слева ключевым абзацам текста был присвоен атрибут `class` со значением `vazhno`. Нескольким элементам на странице допустимо иметь одно и то же значение элемента `class`. Поэтому в данном примере значение `vazhno` может быть использовано и для заголовков, и для ссылок.

По умолчанию использование этого атрибута не влияет на внешний вид элемента. Он будет изменен только в том случае, если вы создадите специальное правило CSS.

В данном примере мы использовали средства CSS, чтобы отобразить элементы класса `vazhno` прописными буквами, а элементы со значением атрибута `class="chasy _ raboty"` — красным цветом.

Если вы желаете указать, что какой-либо элемент относится к нескольким классам, то вы можете перечислить имена классов через пробел, как мы сделали это с третьим абзацем в примере.

БЛОЧНЫЕ ЭЛЕМЕНТЫ

Некоторые элементы всегда отображаются в браузерах с новой строки. Это так называемые **блочные элементы**.

Примерами блочных элементов могут быть `<h1>`, `<p>`, `<ul>` и `<li>`.

глава-08/block-elements.html HTML

```
<h1>Хироши Сугимото</h1>
<p> Ниже приведены даты проведения выставки
    ИСТОКИ ИСКУССТВА:</p>
<ul>
  <li>Наука: 15.01.2013 — 20.02.2013</li>
  <li>Архитектура: 21.02.2013 — 15.05.2013</li>
  <li>История: 16.05.2013 — 27.08.2013</li>
  <li>Религия: 28.08.2013 — 30.12.2013</li>
</ul>
```

РЕЗУЛЬТАТ

Хироши Сугимото

Ниже приведены даты проведения выставки ИСТОКИ ИСКУССТВА:

- Наука: 15.01.2013 — 20.02.2013
- Архитектура: 21.02.2013 — 15.05.2013
- История: 16.05.2013 — 27.08.2013
- Религия: 28.08.2013 — 30.12.2013

ВСТРОЕННЫЕ ЭЛЕМЕНТЫ

HTML **глава-08/inline-elements.html**

```
Рассчитанная по времени одного полного оборота
Земли вокруг Солнца с уклоном 23,4 градуса,
играющего ритм смены сезонов, выставка
<em>Истоки искусства</em> состоит из четырех тем:
<b>наука, архитектура, история</b> и <b>религия</b>.
```

РЕЗУЛЬТАТ

Рассчитанная по времени одного полного оборота Земли вокруг Солнца с уклоном 23,4 градуса, играющего ритм смены сезонов, выставка *Истоки искусства* состоит из четырех тем: **наука, архитектура, история** и **религия**.

Некоторые элементы отображаются на той же строке, что и соседние — они называются **встроенными**.

Примерами таких элементов могут быть `<a>`, `<b>`, `<em>` и `<img>`.

ГРУППИРОВКА ТЕКСТА И ЭЛЕМЕНТОВ В БЛОК

Элемент позволяет сгруппировать несколько элементов в единый блок. Например, вы можете создать , объединяющий в себе все элементы заголовка сайта (логотип и панель навигации). Вы также можете создать для объединения комментариев посетителей сайта. В браузере содержимое элемента будет отображено с новой строки, никаких других изменений внешнего вида не произойдет. Однако с помощью атрибутов id и class вы можете создать правило CSS, определяющее, сколько места будет занимать элемент на экране, а также изменить внешний вид всех элементов, содержащихся в нем.

глава-08/grouping-block-elements.html HTML

```
<div id="header">
  <img src="images/logo.gif" alt="Electric Universe"
/>
  <ul>
    <li><a href="index.html">Главная</a></li>
    <li><a href="biography.html">Биография</a></li>
    <li><a href="works.html">Дискография</a></li>
    <li><a href="contact.html">Контакты</a></li>
  </ul>
</div> <!-- конец заголовка -->
```

РЕЗУЛЬТАТ

- Главная
- Биография
- Дискография
- Контакты

Кроме того, разобраться в вашем коде будет гораздо проще, если каждый раздел страницы вы заключите в отдельный элемент .

Так как внутри может находиться несколько других элементов, комментарии после закрывающего тега бывают очень полезны.

Это позволит вам четко видеть, к чему относится открывающий тег, как показано в последней строке примера.

ГРУППИРОВКА ТЕКСТА И ЭЛЕМЕНТОВ В СТРОКУ

HTML **глава-08/grouping-inline-elements.html**

```
<p>Electric Universe — проект в жанре
  <span class="psy">психоделический транс
  </span>, основанный в Германии в 1991 году.</p>
```

РЕЗУЛЬТАТ

Electric Universe — проект в жанре ПСИХОДЕЛИЧЕСКИЙ ТРАНС, основанный в Германии в 1991 году.

Элемент — встроенный эквивалент элемента . Он используется для:

- окружения части текста, когда никакой другой элемент для этого не подходит;
- окружения нескольких встроенных элементов.

Наиболее важная причина использования — это возможность управлять внешним видом его содержимого с помощью средств CSS.

Довольно часто можно встретить использование атрибутов id и class с элементом :

- для объяснения назначения элемента ;
- для последующего применения стилей CSS к элементам.

ПЛАВАЮЩИЕ ФРЕЙМЫ

<iframe>

Плавающие фреймы напоминают маленькие окна, словно «врезанные» в вашу страницу, через которые вы можете видеть контент других страниц. Название элемента `<iframe>` — это сокращение английского термина «плавающий фрейм».

Содержимым плавающего фрейма может быть любая HTML-страница, как ваша, так и с другого сайта.

Как вы уже поняли, плавающие фреймы создаются с помощью элемента `<iframe>`. У него есть несколько атрибутов, которые следует знать.

src

Атрибут `src` используется для указания URL-адреса страницы, отображаемой во фрейме.

height

Атрибут `height` устанавливает высоту окна плавающего фрейма в пикселах.

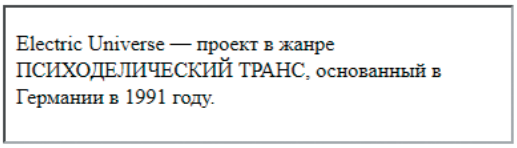
width

Атрибут `width` устанавливает ширину окна плавающего фрейма в пикселах.

глава-08/iframes.html HTML

```
<iframe
  width="380"
  height="100"
  src="grouping-inline-elements.html">
</iframe>
```

РЕЗУЛЬТАТ



HTML **глава-08/iframes2.html**

```
<iframe src="grouping-inline-elements.html"
  width="380" height="100" frameborder="0"
  scrolling="no">
</iframe>
```

РЕЗУЛЬТАТ

Electric Universe — проект в жанре ПСИХОДЕЛИЧЕСКИЙ ТРАНС, основанный в Германии в 1991 году.

scrolling

В языке HTML5 атрибут `scrolling` поддерживаться не будет. В HTML4 и XHTML он определяет, нужно ли отображать полосы прокрутки в окне фрейма. Это важно, если страница внутри окна плавающего фрейма больше, чем его размеры (заданные с помощью атрибутов `height` и `width`). Полосы прокрутки позволяют посетителям увидеть большее количество контента такой страницы. Данному атрибуту можно присвоить три значения: `yes` (для отображения полос прокрутки), `no` (для их сокрытия) или `auto` (для их автоматического отображения в случае необходимости).

frameborder

Атрибут `frameborder` не будет поддерживаться в HTML5. В HTML4 и XHTML он определяет наличие границ у окна плавающего фрейма. Значение `0` устанавливает, что рамка не должна отображаться, значение `1` — что должна.

seamless

В HTML5 реализован новый атрибут `seamless`*, который может быть установлен для плавающего фрейма в случае, когда полосы прокрутки на экране не нужны. Как и многим другим атрибутам HTML5, для `seamless` не требуется значение, однако многие разработчики присваивают ему значение `seamless`. Старые версии браузеров этот атрибут не поддерживают.

* Seamless (*англ.*) — бесшовный, непрерывный. — *Примеч. ред.*

СВЕДЕНИЯ О СТРАНИЦЕ

<meta>

Элемент <meta> помещается внутрь элемента <head> и содержит информацию о текущей странице.
Его содержимое не видно посетителям, однако он выполняет множество функций, например сообщает поисковым системам информацию о странице, о том, кто ее создал, а также установлен ли для нее срок действия (если срок действия установлен, то страница может быть помечена как неактуальная).
Элемент <meta> пустой, поэтому ему не требуется закрывающий тег. Для передачи информации в <meta> используются атрибуты.
Наиболее часто используемые атрибуты — это name и content, которые обычно используются вместе. Они устанавливают свойства всей страницы. Значение атрибута name — это имя свойства страницы, которое вы хотите настроить, а значение атрибута content — новое значение, которое вы собираетесь присвоить данному свойству.
В первой строке исходного кода на следующей странице вы можете видеть элемент <meta>, в котором атрибут name используется для изменения описания страницы, а атрибут content — это то место, где указывается ее новое описание.

Атрибуту name можно присвоить любое значение. Ниже перечислены несколько наиболее употребительных.

description

Значение этого атрибута является описанием страницы, оно чаще всего используется поисковыми системами для определения, чему та посвящена. Длина описания не должна превышать 155 символов. Иногда описание может быть отображено в выдаче поисковой системы.

keywords

Этот параметр должен содержать список ключевых слов, разделенных запятой, которые посетитель может использовать для поиска нужной информации. На практике это свойство уже давно не влияет на индексацию страницы поисковой системой.

robots

Это свойство показывает, должны ли поисковые системы включать данную страницу в результаты поиска. Значение noindex может быть использовано для запрета добавления страницы в результаты. Значение nofollow устанавливает, что поисковые системы могут включать данную страницу в результаты поиска, но не должны показывать страницы, на которые ведут ссылки с нее.

HTML **глава-08/meta.html**

```
<!DOCTYPE html>
<html>
  <head>
    <title>Сведения о ваших веб-страницах</title>
    <meta name="description" content="Эссе об
      искусстве инсталляций" />
    <meta name="keywords" content="инсталляция,
      искусство, мнение" />
    <meta name="robots" content="nofollow" />
    <meta http-equiv="author"
      content="Джон Даккет" />
    <meta http-equiv="pragma" content="no-cache" />
    <meta http-equiv="expires" content="Fri,
      04 Apr 2014 23:59:59 GMT" />
  </head>
  <body>
  </body>
</html>
```

Параметр `http-equiv` элемента `<meta>` также часто используется совместно с атрибутом `content`. В примере атрибут `http-equiv` содержится в трех элементах `<meta>`, при этом каждое его использование имеет свое назначение.

author

Используется для указания имени автора веб-страницы.

pragma

Данный параметр предотвращает кэширование страницы браузером. (Кэширование — это сохранение страницы на жестком диске компьютера посетителя для более быстрого ее открытия при повторном посещении.)

expires

Так как браузеры зачастую кэшируют содержимое страниц, параметр `expires` может быть использован для указания того, когда у страницы должен истечь срок актуальности (и она должна быть удалена из кэша). Обратите внимание, что дату нужно указывать строго в формате, который вы можете видеть в исходном коде примера.

СПЕЦИАЛЬНЫЕ СИМВОЛЫ

В языке HTML существует набор зарезервированных символов, использующихся для написания кода (к ним относятся, например, левая и правая угловые скобки).

Поэтому, если вы хотите отобразить данные символы на странице, вам необходимо обратиться к так называемым **символам-мнемоникам**. Иногда символы-мнемоники также называют HTML-сущностями. Например для отображения левой угловой скобки вы можете воспользоваться кодом `<` или `<`, а для отображения символа амперсанда (&) — кодом `&` или `&`.

Существует набор символов-мнемоник, таких как символ защиты авторских прав (©), торговой марки (™), обозначения национальных валют (¢, €, ¥), математических операторов и некоторых знаков препинания. Например для отображения на странице символа защиты авторских прав воспользуйтесь символом-мнемоником `©` или `©`.

При использовании символов-мнемоник очень важно проверить создаваемую страницу в браузере, чтобы убедиться, что отображается именно то, что нужно. Это следует делать потому, что не все шрифты поддерживают полный набор специальных символов, потому вам может потребоваться установить другой шрифт с помощью каскадных таблиц стилей.

ПРИМЕЧАНИЕ

На странице **www.htmlandcssbook.com/extras/html-escape-codes/** вы найдете полный список символов-мнемоник.

< Знак меньше
<
<

> Знак больше
>
&

& Амперсанд
&
&

" Кавычки
"
"

¢ Цент
¢
¢

£ Британский фунт
£
£

¥ Японская йена
¥
¥

€ Евро
€
€

© Символ защиты авторских прав
©
©

® Зарегистрированная торговая марка
®
®

™ Торговая марка
™
™

‘ Открывающая одинарная кавычка
‘
‘

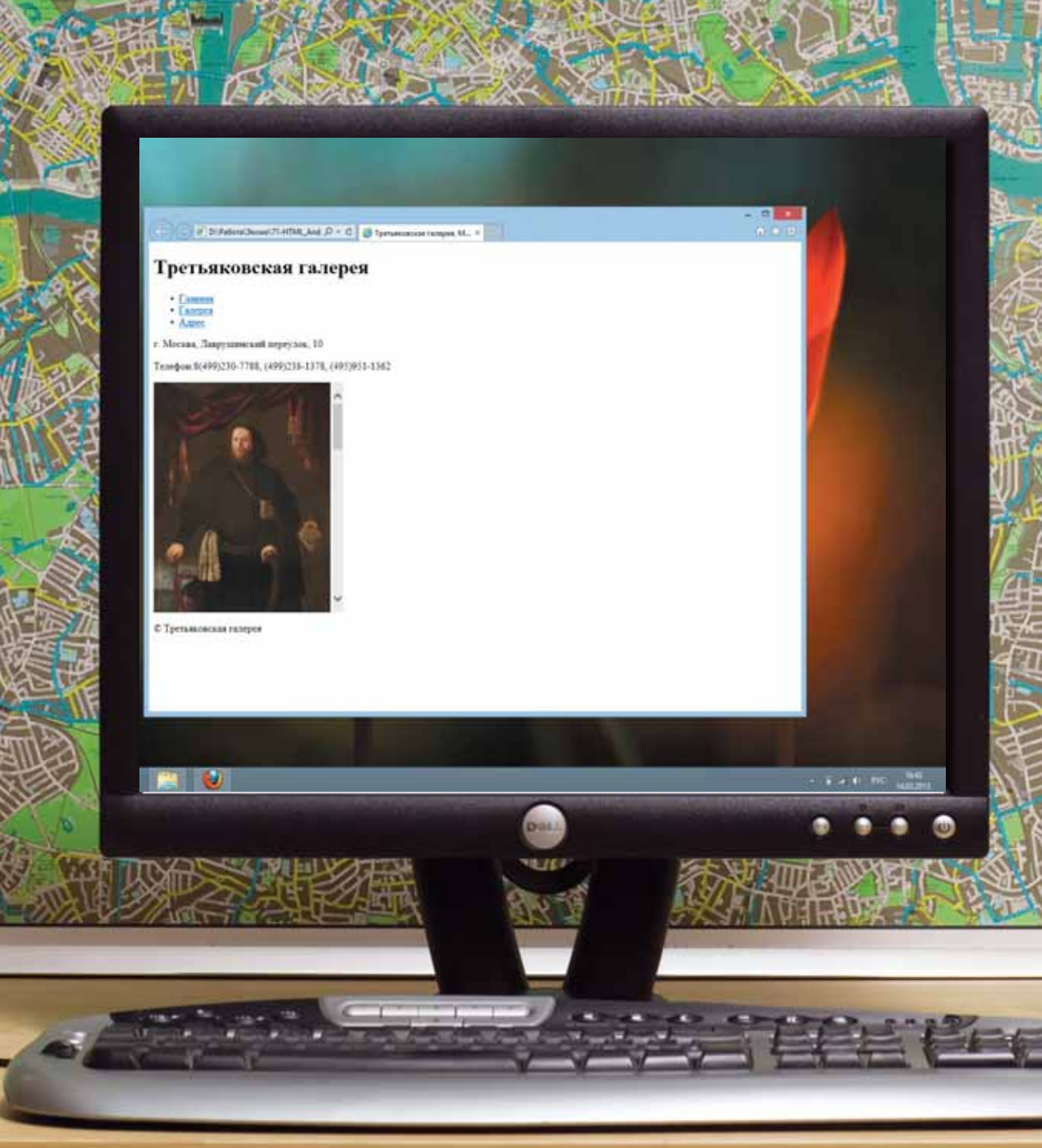
’ Закрывающая одинарная кавычка
’
’

“ Открывающая двойная кавычка
“
“

” Закрывающая двойная кавычка
”
”

× Знак умножения
×
×

÷ Знак деления
÷
÷



Код примера начинается с объявления `DOCTYPE` для указания, что страница была написана на языке HTML4. В `<head>` вы также можете видеть элементы `<meta>`, описывающие содержание страницы.

Для некоторых элементов установлены атрибуты `id` и `class`, указывающие их назначение. Знак защиты авторских прав вставлен в текст с помощью соответствующего символа-мнемоника. Отдельные части страницы сгруппированы с помощью элементов `<div>`, кроме того, к закрывающим тегам `</div>` добавлены комментарии, объясняющие, какую именно часть страницы они завершают.

ПРИМЕР
ДОПОЛНИТЕЛЬНАЯ РАЗМЕТКА

```
<!DOCTYPE html PUBLIC
  "-//W3C//DTD HTML 4.01 Transitional//EN"
  "http://www.w3.org/TR/html4/loose.dtd">
<html>
  <head>
    <meta name="description" content="Телефон и адрес
      Третьяковской галереи, Москва, Россия" />
    <title> Третьяковская галерея, Москва, Россия</title>
  </head>
  <body>
    <div id="header">
      <h1>Третьяковская галерея</h1>
      <ul>
        <li><a href="index.html">Главная</a></li>
        <li><a href="index.html">Галерея</a></li>
        <li class="current-page">
          <a href="index.html">Адрес</a></li>
      </ul>
    </div><!-- основная часть -->
    <div id="content">
      <p>г. Москва, Лаврушинский переулок, 10</p>
      <p><span class="contact">Телефон:
        </span>8(499)230-7788, (499)238-1378,
        (495)951-1362</p>
        <iframe width="290" height="360"
          frameborder="0" scrolling="yes"
          marginheight="0" marginwidth="0" src="images\
          gallery.htm">
        </iframe>
    </div>
    <!-- конец основной части -->
    <p>&copy; Третьяковская галерея</p>
  </body>
</html>
```

ОБЗОР
ДОПОЛНИТЕЛЬНАЯ РАЗМЕТКА

- Объявления DOCTYPE сообщают браузеру, какая версия языка HTML используется.
- Вы можете добавлять в код комментарии, помещая текст между символами <!-- и -->.
- Атрибуты id и class позволяют идентифицировать конкретные элементы разметки.
- Элементы и позволяют группировать блочные и встроенные элементы.
- Элемент <iframe> превращает страницу в окно, за которым могут быть видны другие страницы.
- Тег <meta> позволяет предоставить всю информацию о странице.
- Для отображения на странице специальных знаков, таких как <, > или ©, используются специальные символы-мнемоники.

9

FLASH, ВИДЕО- И АУДИОКОНТЕНТ

- Как добавлять на сайт Flash-ролики
- Как добавлять на сайт видео и аудиофайлы
- Элементы <video> и <audio> языка HTML5

Flash — популярная технология, позволяющая добавлять на сайты анимацию, видеоролики и аудиозаписи. Данная глава начинается с рассказа о том, как использовать эту технологию на вашем сайте.

После этого мы переключаемся на рассмотрение возможностей добавления аудиозаписей и видеороликов с помощью элементов `<video>` и `<audio>` языка HTML5 или сторонних сервисов (таких как YouTube или SoundCloud). В этой главе вы узнаете:

- как использовать технологию Flash на своем сайте;
- как использовать элементы `<video>` и `<audio>` языка HTML5;
- когда следует размещать аудио/видеофайлы на своем сайте, а когда следует воспользоваться сторонним сервисом, таким как YouTube.

MacBook Pro

КАК РАБОТАЕТ ТЕХНОЛОГИЯ FLASH

Технология Flash обрела популярность в 1990-е годы сначала как средство для создания анимации, впоследствии также и как технология для проигрывания аудиозаписей и видеороликов на сайтах.

Независимо от того, создаете вы анимацию или мультимедийный проигрыватель средствами Flash, файлы, загружаемые вами на сайт, называются **Flash-роликами**.

Для разработки собственных Flash-роликов необходимо приобрести специальное программное обеспечение, Adobe Flash Professional.

Однако существует несколько компаний, предлагающих готовые Flash-анимации, слайд-шоу, а также видео- и аудиопроигрыватели, которые можно использовать без необходимости приобретения вышеупомянутого программного обеспечения.

При создании проекта в программе Adobe Flash Professional он сохраняется с расширением **.**fla. Для использования на сайте также сохраняется файл в формате SWF (с расширением .swf).

При экспорте проекта в формате SWF программа Adobe Flash Professional создает код, который может быть использован для внедрения Flash-ролика на страницу. Раньше в подобном коде, как правило, использовались элементы `<object>` и `<embed>` языка HTML. Однако теперь чаще можно встретить сценарии на языке JavaScript.

Для просмотра Flash-роликов в браузере должно быть установлено дополнительное программное обеспечение (плагин), называемое Flash Player. Согласно статистическим данным, на 98% настольных компьютерах оно есть. (Однако процентный показатель для мобильных устройств и планшетных компьютеров с установленным плагином Flash Player существенно ниже.)

К сожалению, тема создания Flash-роликов выходит за рамки нашей книги (этому посвящено очень много книг, например «Adobe Flash Professional CS6. Официальный учебный курс» издательства Эксмо), а в настоящей главе мы покажем вам, как разместить Flash-ролики на сайте.

Flash-ролики создаются с помощью специального программного обеспечения

Файл FLA экспортируется в формат SWF для дальнейшего использования на веб-странице

Файл SWF подключается к странице с помощью сценария, написанного на языке JavaScript

ИСПОЛЬЗОВАНИЕ ТЕХНОЛОГИИ FLASH

Начиная с 2005 года из-за нескольких ключевых факторов количество сайтов, частично или полностью написанных с помощью технологии Flash, постепенно уменьшается.

Изначально технология Flash была предназначена для разработки анимации. Однако данная технология быстро эволюционировала, и люди начали использовать ее для создания проигрывателей мультимедийных файлов и даже сайтов целиком.

Несмотря на ее популярность в наши дни разработчики стали тщательнее выбирать, когда пользоваться ею, а когда нет (в результате сейчас редко можно встретить сайт, созданный полностью средствами Flash).

Для уменьшения количества сайтов, на которых используется данная технология, есть несколько причин.

В 2005–2006 годах увидел свет набор новых библиотек JavaScript (включая Prototype, script.aculo.us и jQuery), значительно упростивший разработчикам создание анимационных эффектов средствами языка JavaScript.

При выпуске смартфона iPhone в 2007 и планшетного компьютера iPad в 2010 году компания Apple приняла решение о прекращении поддержки технологии Flash.

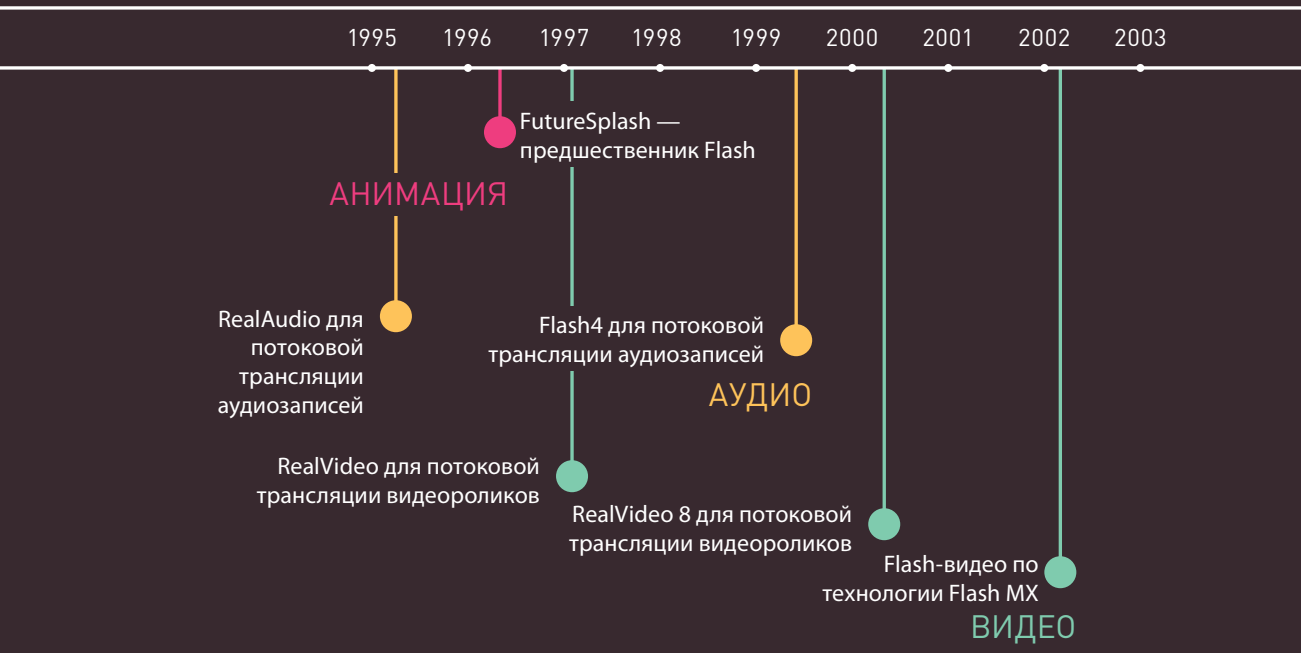
В мире появились законы, обязывающие создателей сайтов делать их доступными для людей с нарушением зрения, в результате чего технология Flash была раскритикована за неполноценную поддержку специальных возможностей, предоставляющих доступ к сайтам таким посетителям.

Начиная с 2008 года браузеры стали поддерживать теги `<video>` и `<audio>` языка HTML5. На момент написания книги технология Flash все еще продолжала оставаться популярным средством для проигрывания аудиозаписей и видеороликов во Всемирной паутине, однако все большее и большее количество разработчиков переключаются на язык HTML5.

(Как использовать эти элементы, вы узнаете чуть позже в настоящей главе.)

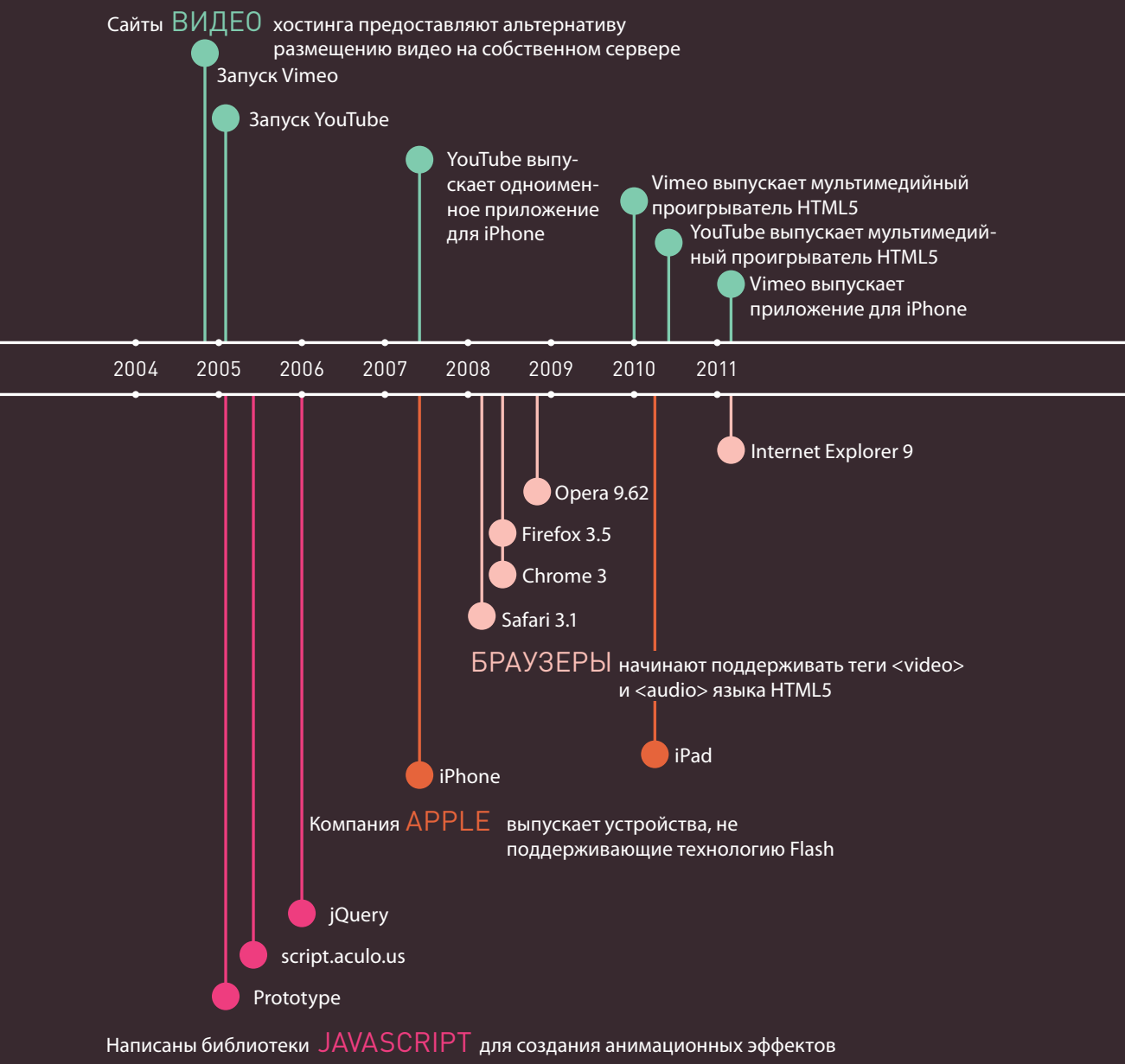
ЭКСКУРС В ИСТОРИЮ: FLASH, ВИДЕО И АУДИОКОНТЕНТ

Веб-технологии изменяются очень быстро. Ниже вы можете видеть несколько этапов изменения технологий создания анимации, проигрывания видеороликов и аудиозаписей во Всемирной паутине.



На этой странице вы можете видеть первые основные проигрыватели, позволявшие воспроизводить анимацию, аудиозаписи и видеоролики.

На следующей странице вы увидите некоторые технологии и события, способствовавшие их замещению.


Сайты ВИДЕО хостинга предоставляют альтернативу размещению видео на собственном сервере
Запуск Vimeo
Запуск YouTube
YouTube выпускает одноименное приложение для iPhone
Vimeo выпускает мультимедийный проигрыватель HTML5
YouTube выпускает мультимедийный проигрыватель HTML5
Vimeo выпускает приложение для iPhone
2004
2005
2006
2007
2008
2009
2010
2011
Internet Explorer 9
Opera 9.62
Firefox 3.5
Chrome 3
Safari 3.1
БРАУЗЕРЫ начинают поддерживать теги <video> и <audio> языка HTML5
iPad
iPhone
Компания APPLE выпускает устройства, не поддерживающие технологию Flash
jQuery
script.aculo.us
Prototype
Написаны библиотеки JAVASCRIPT для создания анимационных эффектов

ДОБАВЛЕНИЕ FLASH-КОНТЕНТА НА ВЕБ-СТРАНИЦУ

Наиболее популярный способ добавления Flash-ролика на веб-страницу заключается в использовании для этих целей языка JavaScript. Существуют несколько сценариев, позволяющих вам сделать такое без необходимости детального ознакомления с языком JavaScript. Сценарий, рассматриваемый нами в этой главе, называется SWFObject. Вы можете бесплатно найти его копию с помощью Google. Способ его применения мы рассмотрим на следующей странице.

Одно из преимуществ использования этого способа в том, что вы можете отобразить альтернативный контент в браузерах, которые не поддерживают работу с Flash.

При этом способе используется элемент `<div>`, создающий пространство, в которое будет помещен Flash-ролик. Для элемента `<div>` установлен атрибут `id`, значение которого используется сценарием SWFObject. В этом примере атрибуту `id` присвоено значение `bird`.

глава-09/adding-a-flash-movie.html HTML

```
<!DOCTYPE html>
<html>
  <head>
    <title>Добавление Flash-ролика</title>
    <script type="text/javascript"
      src="http://ajax.googleapis.com/ajax/libs/
      swfobject/2.2/swfobject.js"></script>
    <script type="text/javascript">
      swfobject.embedSWF("flash/bird.swf",
      "bird", "400", "300", "8.0.0");</script>
  </head>
  <body>
    <div id="bird"><p>Анимация: птичка,
      принимающая душ</p></div>
  </body>
</html>
```

В элемент `<div>` вы можете поместить альтернативный контент, отображаемый для пользователей, которые не способны воспроизвести Flash-ролик.

Сценарий SWFObject проведет проверку, поддерживает ли браузер посетителя воспроизведение Flash-роликов.

Если да, то браузер заменит содержимое элемента `<div>` файлом SWF.

Для пользователей, которые не способны просмотреть Flash-ролик, можно поместить некий кадр из него и/или текстовое описание Flash-файла.

При использовании текстового описания в качестве альтернативного контента, у вас появляется два преимущества.

Текст может быть прочитан посетителями с нарушением зрения и/или другими ограничениями, то есть теми людьми, которые не способны просмотреть Flash-ролик.

Текст может быть проиндексирован поисковыми системами (эффективность индексирования SWF-файлов не сравнится с эффективностью индексирования текста), и это увеличит вероятность того, что созданный вами контент будет найден посредством поиска.

РЕЗУЛЬТАТ

В этом примере сценарий SWFObject размещен на веб-сервере компании Google. Мы **встраиваем** данный сценарий в страницу с помощью первого из двух элементов `<script>`. Атрибут `type` используется с элементом `<script>` для указания того, что сценарий написан на языке JavaScript. Атрибут `src` указывает браузеру, где найти нужный файл сценария.

Второй элемент `<script>` используется, чтобы сообщить браузеру дополнительную информацию о Flash-ролике, а также и то, какой элемент должен быть им заменен. На самом деле данный элемент передает сценарию SWFObject **пять** фрагментов данных, заключенных в скобки:

Местоположение файла SWF: `flash/bird.swf`;

Имя элемента, **замещаемого** Flash-роликом, устанавливаемое значением атрибута `id` элемента `<div>`: `bird;`

Ширину встраиваемого Flash-ролика в пикселах: `400 px;`

Высоту встраиваемого Flash-ролика в пикселах: `300 px;`

Минимальный номер **версии** плагина Flash Player, необходимого для просмотра данного ролика: `Flash Player 8.`

ОБЗОР ВИДЕОФОРМАТОВ И ПРОИГРЫВАТЕЛЕЙ

Для добавления видеоролика на сайт необходимо понимать два ключевых момента: форматы файлов и проигрыватели/плагины для просмотра видео.

ФОРМАТЫ

Вы можете встретить видеофильмы в самых разных форматах (диски Blu-ray и DVD, кассеты VHS и т.п.). Во Всемирной паутине существует *гораздо большее* количество видеоформатов (в их числе AVI, Flash Video, H264, MPEG, Ogg Theora, QuickTime, WebM и Windows Media).

Аналогично тому, как ваш DVD-проигрыватель не воспроизводит видеокассеты VHS, браузеры отличаются по количеству поддерживаемых и неподдерживаемых форматов.

Чтобы пользователи смогли просмотреть ваш видеоролик во Всемирной паутине, вероятно, вам придется конвертировать его в другой формат. Процесс конвертирования видеофайла в другой формат иногда также называется «перекодированием» видео. Во Всемирной паутине доступно множество онлайн-сервисов и приложений, позволяющих осуществлять перекодирование видеофайлов например **www.mirovideoconverter.com**.

ПРОИГРЫВАТЕЛИ И ПЛАГИНЫ

С самого начала браузеры разрабатывались для отображения только текста и изображений. Поэтому для браузеров, выпущенных до 2010 года, как правило требуется дополнительное программное обеспечение, называемое проигрывателем, или плагином, и позволяющее воспроизводить видеоконтент сайтов.

Эти проигрыватели и плагины поддерживали только определенный набор форматов. В последние годы браузеры перешли на новую ступень эволюции и теперь поддерживают тег `<video>` языка HTML5, отправивший все эти программные дополнения в прошлое.

Однако, к сожалению, вы не можете рассчитывать на то, что каждый посетитель вашего сайта пользуется новейшей версией браузера с поддержкой тегов языка HTML5. Браузеры, не поддерживающие элемент `<video>`, требуют перекодирования видеофайла в другой формат.

МЕТОДЫ

Самый простой способ размещения видеоролика на странице заключается в использовании сайтов видеохостинга, таких как YouTube и Vimeo.

Однако, как вы увидите на следующей странице, существуют ситуации, когда размещение видео на сторонних серверах нежелательно, — и у вас может возникнуть потребность расположить видеоконтент на собственном сайте.

На момент написания книги самым лучшим методом размещения видеоконтента на сайте считалось использование элемента `<video>` языка HTML5 для браузеров, поддерживающих эту версию языка, а для прочих предлагалось использование технологии Flash. Все видеоролики, встраиваемые в страницу, должны быть загружены на сервер как минимум в двух форматах: WebM и MP4.

САЙТЫ ВИДЕОХОСТИНГА

Самый простой способ добавления видеоконтента на ваш сайт — это загрузка нужного файла на сервер сервиса видеохостинга, такого как YouTube, и дальнейшее встраивание видеоролика на веб-страницу с помощью инструментов данного сервиса.

ПРЕИМУЩЕСТВА

Сайты видеохостинга (такие как YouTube) предоставляют проигрыватели, работающие в большинстве браузеров.

Вам не стоит беспокоиться о перекодировании загружаемых видеофайлов, так как эти сайты позволяют использовать файлы в разных форматах. После загрузки файлы будут конвертированы в форматы, совместимые с разными браузерами.

Хостинг-провайдеры, как правило, повышают абонентскую плату, если трафик вашего сайта становится слишком большим, а видеофайлы занимают достаточно много пространства на жестком диске сервера. Поэтому размещение видеофайлов на собственном сайте может обойтись вам недешево. Однако если ваш видеоконтент размещен на таком сайте, как YouTube, вам не придется платить за трафик.

НЕДОСТАТКИ

Загруженное вами видео окажется доступно посетителям сайта видеохостинга, поэтому, если вы хотите, чтобы данный файл можно было просматривать только через ваш сайт (но не через какой-либо другой), то вам придется разместить его на своем сервере и добавить на страницу собственный проигрыватель.

Некоторые сайты видеохостинга жестко ограничивают содержимое загружаемых видеороликов. Например наиболее распространен запрет использования в загружаемом видеоконтенте рекламной информации (что предотвращает получение материальной выгоды от видеоконтента).

Некоторые сайты могут в первую очередь запускать проигрывание собственных рекламных роликов, и лишь потом — вашего видеоконтента. На других реклама перекрывает часть изображения. Кроме того, некоторые сайты ограничивают качество транслируемого видео.

АЛЬТЕРНАТИВА

Если же вы хотите разместить видеоконтент на собственном сервере, а не на специализированном сервисе видеохостинга, то вам потребуется приложить достаточное количество усилий, чтобы настроить ваш сайт на проигрывание видеороликов.

Мы рассмотрим два способа размещения видео на пользовательском сайте: с использованием технологии Flash и элемента `<video>` языка HTML5.

Чтобы быть уверенным, что большинство посетителей вашего сайта сможет просмотреть размещенный на нем видеоролик, вам придется пользоваться обеими этими технологиями одновременно.

ПОДГОТОВКА FLASH-КОНТЕНТА ДЛЯ РАЗМЕЩЕНИЯ НА САЙТЕ

Для добавления Flash-видео на веб-страницу необходимо пройти три этапа.

1

КОНВЕРТИРОВАНИЕ ВИДЕО В ФОРМАТ FLV.

Для воспроизведения Flash-видео загружаемый файл должен быть конвертирован в формат FLV. Начиная с шестой версии Flash программное обеспечение для работы с этой технологией дополняется инструментом Flash Video Encoder, позволяющим конвертировать видеофайлы в формат FLV.

Некоторые проигрыватели Flash-контента поддерживают формат H.264 (а некоторые видеоредакторы позволяют экспортировать в него файлы).

Поиск информации по запросу «конвертер FLV» и «конвертер H.264» позволит вам найти подходящее программное обеспечение.

Мы предоставляем вам в качестве примера файл FLV (он находится в отдельной папке, поскольку видеофайлы достаточно велики), который вы можете использовать совместно с исходным кодом.

2

ПОИСК FLV-ПРОИГРЫВАТЕЛЯ ДЛЯ ПРОСМОТРА ВИДЕОКОНТЕНТА.

Для воспроизведения файлов формата FLV вам понадобится специальный **проигрыватель**, сделанный по технологии Flash. Его назначение — создание виртуального экрана для проигрывания видео формата FLV с добавлением элементов управления, таких как кнопки **Воспроизведение** или **Пауза**.

Ниже приведено два сайта, предлагающих собственные FLV-проигрыватели:
www.osflv.com;
www.longtailvideo.com.

Для использования этих проигрывателей на своем сайте вам не понадобится специализированная программа для работы с Flash-проектами.

В следующем примере мы воспользуемся бесплатным проигрывателем с открытым кодом OS FLV. Он также включен в архив с исходным кодом. Данный проигрыватель поддерживает только формат видео FLV (но не H.264).

3

ВСТРАИВАНИЕ ПРОИГРЫВАТЕЛЯ И ВИДЕОКОНТЕНТА В ВЕБ-СТРАНИЦУ.

Вы можете встроить проигрыватель в свою страницу с помощью сценария JavaScript, например SWFObject, речь о котором шла ранее в этой главе.

Кроме того, вам потребуется указать проигрывателю местоположение файла, который вы желаете воспроизвести. (Некоторые проигрыватели обладают расширенным функционалом, например возможностью создавать списки воспроизведения из нескольких видеороликов или отображать статичный кадр из ролика перед началом его воспроизведения.)

Кроме того, в следующем примере мы воспользуемся SWFObject — сценарием на языке JavaScript, упомянутым ранее.

ДОБАВЛЕНИЕ FLASH-КОНТЕНТА НА СТРАНИЦУ

HTML **глава-09/adding-a-flash-video.html**

```
<!DOCTYPE html>
<html>
  <head>
    <title>Добавление Flash-ролика </title>
    <script type="text/javascript"
      src="http://ajax.googleapis.com/ajax/libs/
      swfobject/2.2/swfobject.js"></script>
    <script type="text/javascript">
      var flashvars = {};
      var params = {movie:"../video/puppy.flv"};
      swfobject.embedSWF("flash/splayer.swf", "snow",
      "400", "320", "8.0.0", flashvars, params);
      </script>
  </head>
  <body>
    <div id="snow"><p>Щенок, играющий в снегу
      </p></div>
  </body>
</html>
```

РЕЗУЛЬТАТ

В этом примере для воспроизведения видеоконтента используется проигрыватель OS FLV, а также видеофайл puppy.flv, предварительно конвертированный в формат FLV.
Вы уже видели пример использования сценария SWFObject для внедрения простой анимации на страницу, но иногда Flash-роликам для работы требуются дополнительные данные. В этом примере проигрывателю необходимо сообщить расположение нужного видеофайла, потому в сценарии SWFObject используются переменные JavaScript для передачи этой информации Flash-ролику. Переменные объявляются в двух строках кода, начинающихся с ключевого слова `var`. Данный проигрыватель не использует переменную `flashvars`, поэтому ее значение остается пустым.
Путь к файлу видеоролика указывается в переменной `params`.

```
var params = {movie:
"../videos/puppy.flv"};
```

Строка кода, следующая сразу после объявления переменных, передает сценарию команду заменить HTML-элемент видеопроигрывателем. Использованный метод очень похож на тот, что был представлен вместе со сценарием SWFObject.

Разным проигрывателям, как правило, требуется передавать путь к используемому видеофайлу в разных форматах. Впрочем, к каждому такому проигрывателю прилагается вся необходимая документация с примерами, которые помогут вам разобраться, как им пользоваться.

HTML5: ПОДГОТОВКА ВИДЕОКОНТЕНТА ДЛЯ ДОБАВЛЕНИЯ НА СТРАНИЦУ

Несмотря на то что элемент `<video>` относительно новый, он уже приобрел достаточную популярность среди разработчиков. Ниже перечислено несколько моментов, о которых следует помнить.

ПОДДЕРЖКА

Элемент `<video>` языка HTML5 поддерживается только новейшими версиями браузеров, поэтому, чтобы все посетители вашего сайта могли просмотреть ваш видеоконтент, необходимо продолжать сочетать технологии HTML5 и Flash.

АВТОРСКИЕ ПРАВА

На момент написания книги элемент `<video>` еще не обладал поддержкой технических средств защиты авторских прав*. Однако, как правило, при желании можно обойти любое средство защиты авторских прав.

* DRM, Digital Rights Management (*англ.*). Иногда эту систему также называют защитой от копирования. — *Примеч. ред.*

ФОРМАТЫ

Некоторые браузеры не поддерживают воспроизведение видеоконтента некоторых форматов. Поэтому вам необходимо предоставлять видеофайлы в разных форматах.

Чтобы ваше видео можно было воспроизводить в большинстве браузеров, предоставьте видеофайлы следующих форматов:

H264: Internet Explorer и Safari;

WebM: Android, Chrome, Firefox и Opera.

Разработчики браузеров Chrome, Firefox и Opera подтвердили, что их браузеры поддерживают формат WebM. (Некоторые проигрыватели Flash-контента также поддерживают оба формата — H264 и WebM — что поможет вам сэкономить время на перекодировании.)

ЭЛЕМЕНТЫ УПРАВЛЕНИЯ

Браузеры предоставляют собственные элементы управления мультимедийным контентом, которые различаются в разных браузерах. Вы можете управлять оформлением этих элементов управления с помощью сценариев JavaScript (данная тема выходит за рамки этой книги).

В БРАУЗЕРЕ

Одна из проблем плагина Flash Player заключается в нестабильном поведении при перекрывании окна воспроизведения раскрывшемся меню или при изменении размеров окна. HTML5 решает эти проблемы.

Далее вы узнаете, как совмещать технологию Flash со средствами HTML5 для достижения лучших результатов.

Мы предоставили видеофайл, доступный вместе с исходным кодом примеров, в двух форматах: H264 и WebM.

Если вы запустите следующий пример в браузерах Firefox и Opera, то сможете увидеть, что при наведении указателя мыши на область воспроизведения, элементы управления проигрывателя будут различны в этих браузерах.

HTML5: ДОБАВЛЕНИЕ ВИДЕОКОНТЕНТА НА СТРАНИЦУ

HTML **глава-09/adding-html5-video.html**

```
<!DOCTYPE html>
<html>
  <head>
    <title>HTML5: добавление видео на страницу</title>
  </head>
  <body>
    <video src="video/puppy.mp4"
      poster="images/puppy.jpg"
      width="400" height="300"
      preload
      controls
      loop>
      <p>Щенок, играющий в снегу</p>
    </video>
  </body>
</html>
```

В языке HTML5 не обязательно присваивать значения атрибутам `controls`, `autoplay` и `loop`, используемым с элементом `<video>`. Эти атрибуты подобны переключателям. Если атрибут указан, значит, данная функция считается включенной, если не указан — соответственно, отключенной.

Если браузер не поддерживает элемент `<video>` или формат используемого видеофайла, то на экран будет выведена информация, помещенная между открывающим `<video>` и закрывающим `</video>` тегами.

preloaded

Данный атрибут дает браузеру инструкцию, что следует сделать при загрузке страницы. Он может принимать одно из следующих трех значений.

none

Браузер не должен начинать загрузку видеоролика, пока пользователь не щелкнет мышью по кнопке **Воспроизведение**.

auto

Браузер должен загрузить видеоролик по окончании загрузки страницы

metadata

Браузер должен лишь собрать такую метаинформацию, как размер видео, первый кадр, список воспроизведения и длительность.

<video>

Элемент `<video>` обладает целым набором атрибутов, позволяющих управлять воспроизведением видеоролика.

src

Данный атрибут позволяет указать расположение нужного видеофайла. (Видеоролик для этого примера был сохранен в формате H.264 и потому может быть воспроизведен только в браузерах Internet Explorer и Safari.)

poster

Данный атрибут позволяет установить изображение, показываемое в процессе загрузки видеоконтента и до момента, пока посетитель не начнет воспроизведение ролика.

width, height

Эти атрибуты определяют размеры окна проигрывателя в пикселах.

controls

Этот атрибут используется, если требуется, чтобы браузер предоставил собственные элементы управления воспроизведением.

autoplay

Этот атрибут используется, если требуется, чтобы видеоролик автоматически начинал воспроизводиться после загрузки.

loop

Этот атрибут используется, если требуется, чтобы проигрыватель автоматически повторял воспроизведение видеоролика.

HTML5: НЕСКОЛЬКО ИСТОЧНИКОВ ВИДЕОКОНТЕНТА

<source>

Для указания расположения видеофайла вы можете воспользоваться элементом `<source>`, помещаемым внутрь элемента `<video>`. (Данный элемент призван заменить атрибут `src` тега `<video>`.)
Для обозначения того, что видеоролик доступен в нескольких форматах, вы можете использовать несколько элементов `<source>`.
(Из-за ошибки в программном обеспечении планшета iPad видеофайл может не воспроизводиться, если в качестве первого источника не указан видеоролик в формате MP4.)

глава-09/multiple-video-sources.html HTML

```
<!DOCTYPE html>
<html>
  <head>
    <title>HTML5: несколько источников видео</title>
  </head>
  <body>
    <video poster="images/puppy.jpg" width="400"
      height="320" preload controls loop>
      <source src="video/puppy.mp4" type='video/
        mp4;codecs="avc1.42E01E, mp4a.40.2"' />
      <source src="video/puppy.webm" type='video/
        webm;codecs="vp8, vorbis"' />
      <p>Щенок, играющий в снегу</p>
    </video>
  </body>
</html>
```

РЕЗУЛЬТАТ

src

Данный атрибут служит для указания пути к видеофайлу.

type

Используйте этот атрибут для указания формата видеофайла. В противном случае браузер загрузит часть видеофайла, чтобы проверить возможность воспроизведения, что займет некоторое время и повлечет лишний расход трафика.

codecs

В атрибуте `type` необходимо указать кодек для конвертирования видео. Обратите внимание на использование двойных и одинарных кавычек при указании атрибута `type`.

Если браузер не поддерживает элемент `<video>` или формат используемого видеофайла, то на экран будет выведена информация, помещенная между открывающим `<video>` и закрывающим `</video>` тегами.

ПРИМЕЧАНИЕ

На странице **www.htmlandcss-book.com/extras/encoding-videos-for-the-web/** размещено несколько ссылок на инструменты, позволяющие конвертировать видео- и аудиофайлы в требуемый формат.

HTML5: СОВМЕСТНОЕ ИСПОЛЬЗОВАНИЕ ТЕХНОЛОГИЙ FLASH И HTML5 ДЛЯ РАЗМЕЩЕНИЯ ВИДЕОКОНТЕНТА

Предоставляя видеоконтент в форматах HTML5 и Flash одновременно, вы будете уверены в том, что большинство посетителей вашего сайта его увидит.

Вы можете выбрать HTML5 в качестве первичного способа отображения видеоконтента, а Flash как вариант для тех посетителей, чьи браузеры не поддерживают элементы языка HTML5. Вы также можете поступить иначе.

Так как некоторые видеопроигрыватели обладают встроенной поддержкой Flash-видеороликов в формате H.264, то при их использовании вам потребуется лишь предоставить видеофайл форматах H.264 и WebM. (Копия в формате FLV вам не понадобится.) Вы увидите эту технологию в действии в примере в конце главы.

Если вы изучите возможности HTML5, предназначенные для размещения/воспроизведения видеоконтента, то сумеете:

- создавать собственные элементы управления воспроизведением;
- предоставлять разные версии видеороликов для браузеров устройств с различным разрешением экрана (таким образом вы сможете предоставлять видеоконтент с меньшим разрешением для воспроизведения на портативных устройствах);
- изменять контент отдельных частей страниц при достижении определенной позиции воспроизведения.

ДОБАВЛЕНИЕ АУДИОКОНТЕНТА НА СТРАНИЦУ

Наиболее популярный формат аудиофайлов, добавляемых на веб-страницы, — это MP3. Как и в случае с видеоконтентом, у вас есть три пути на выбор.

1

ИСПОЛЬЗОВАНИЕ САЙТА АУДИОХОСТИНГА.

Во Всемирной паутине доступно несколько сайтов, таких как **www.soundcloud.com** или **www.myspace.com**, позволяющих загружать аудиофайлы, а также предоставляющих аудиопроигрыватель, который вы можете встроить на свою страницу.

Может возникнуть вопрос, как достичь непрерывного воспроизведения аудиофайла при перемещении посетителя по страницам сайта.

2

ИСПОЛЬЗОВАНИЕ ТЕХНОЛОГИИ FLASH

Существует несколько Flash-проигрывателей, позволяющих воспроизводить аудиофайлы в формате MP3: от простых, на один аудиофайл, до сложных с поддержкой списков и сценариев воспроизведения.

На самом деле добиться подобного довольно сложно, поскольку это потребует использования таких технологий, как AJAX, позволяющих загружать контент страницы, или же создания Flash-сайта.

3

ИСПОЛЬЗОВАНИЕ ЯЗЫКА HTML5

В языке HTML5 доступен новый элемент `<audio>`. Браузеры, поддерживающие его, предоставляют для него собственные элементы управления, так же, как и при размещении видеоконтента.

Именно поэтому на многих сайтах аудиопроигрыватель открывается в отдельном окне, и, таким образом, при перемещении между страницами сайта воспроизведение аудиофайла не прерывается.

ДОБАВЛЕНИЕ FLASH-АУДИОПРОИГРЫВАТЕЛЯ

HTML **глава-09/adding-a-flash-mp3-player.html**

```
<!DOCTYPE html>
<html>
  <head>
    <title>Добавление
      Flash-аудиопроигрывателя</title>
    <script type="text/javascript"
      src="http://ajax.googleapis.com/ajax/libs/
      swfobject/2.2/swfobject.js"> </script>
    <script type="text/javascript">
      var flashvars = {};
      var params = {mp3: "audio/test-audio.mp3"};
      swfobject.embedSWF(
        "flash/player_mp3_1.0.0.swf",
        "music-player", "200", "20", "8.0.0",
        flashvars, params);</script>
  </head>
  <body>
    <div id="music-player">
      <p> Вы не можете прослушать данную
          аудиозапись, так как ваш браузер не
          поддерживает Flash-аудиопроигрыватель.</p>
    </div>
  </body>
</html>
```

РЕЗУЛЬТАТ

Было создано уже много проигрывателей с использованием технологии Flash для воспроизведения MP3-файлов, например:

`www.flash-mp3-player.net`;
`www.musicplayer.sourceforge.net`;
`www.wimpyplayer.com`.

Все они имеют разные возможности, поэтому всегда проверяйте функции проигрывателя перед добавлением его на свою страницу. В этом примере используется бесплатный проигрыватель с сайта **www.flash-mp3-player.net**, добавляемый на страницу с помощью сценария SWFObject. Параметр `mp3` сообщает проигрывателю путь к файлу формата MP3. После второго элемента `<script>` вы можете видеть объявление двух переменных JavaScript: `flashvars` и `params`. Несмотря на то что мы фактически не используем переменную `flashwars`, обработчик кода сценария SWFObject ожидает ее присутствия в этой строке кода перед переменной `params`, потому необходимо указать ее.

```
var flashvars = {};
var params = {
mp3: "music/noise.mp3"};
```

Затем эти переменные добавляются к концу строки, помещающей проигрыватель на страницу (как раз перед закрывающим тегом `</script>`).

HTML5: ДОБАВЛЕНИЕ АУДИОКОНТЕНТА НА СТРАНИЦУ

<audio>

В языке HTML5 используется элемент <audio>, позволяющий добавлять аудиозаписи на веб-страницу. Точно так же, как и в случае с добавлением видео при помощи HTML5, разные браузеры поддерживают различные форматы аудиофайлов. Для элемента <audio> был создан ряд атрибутов, позволяющих управлять воспроизведением аудиозаписи.

src

Данный атрибут определяет путь к нужному аудиофайлу.

controls

Данный атрибут указывает, должен ли проигрыватель отображать элементы управления воспроизведением. По умолчанию при его отсутствии элементы управления отображены не будут. С помощью языка JavaScript вы также можете настроить отображение ваших собственных элементов управления.

autoplay

Наличие этого атрибута указывает, что проигрывание аудиозаписи должно начинаться автоматически. (Однако рекомендуется позволить посетителям самостоятельно включать воспроизведение.)

глава-09/adding-html5-audio.html HTML

```
<!DOCTYPE html>
<html>
  <head>
    <title>HTML5: добавление аудиоконтента
      на страницу</title>
  </head>
  <body>
    <audio src="audio/test-audio.ogg" controls
      autoplay>
      <p>Ваш браузер не поддерживает формат данного
        аудиофайла.</p>
    </audio>
  </body>
</html>
```

РЕЗУЛЬТАТ

preload

Данный атрибут указывает, какие действия должен выполнять браузер, если для проигрывателя не указан атрибут autoplay. Данный атрибут может иметь те же самые значения, что и его аналог в элементе <video>.

loop

Данный атрибут указывает, что по завершении воспроизведения аудиофайла оно должно автоматически начаться вновь.

Приведенный пример работает только в браузерах, поддерживающих аудиоформат Ogg Vorbis (Firefox, Chrome и Opera). Чтобы данный пример работал также и в браузерах Safari и Internet Explorer, аудиофайл должен быть сохранен в формате MP3. Вы также можете воспользоваться элементом <source>, речь о котором пойдет на следующей странице, чтобы предоставить браузеру на выбор файлы нескольких форматов.

HTML5: НЕСКОЛЬКО ИСТОЧНИКОВ АУДИОКОНТЕНТА

HTML **глава-09/multiple-audio-sources.html**

```
<!DOCTYPE html>
<html>
  <head>
    <title> HTML5: несколько источников
            аудиоконтента</title>
  </head>
  <body>
    <audio controls autoplay>
      <source src="audio/test-audio.ogg" />
      <source src="audio/test-audio.mp3" />
      <p> Ваш браузер не поддерживает формат данного
          аудиофайла.</p>
    </audio>
  </body>
</html>
```

РЕЗУЛЬТАТ

src

Атрибут `src` используется с элементом `<source>` для указания местоположения нужного файла.

type

На момент написания книги атрибут `type` не так широко применялся с элементом `<source>`, в отличие от элемента `<video>`.

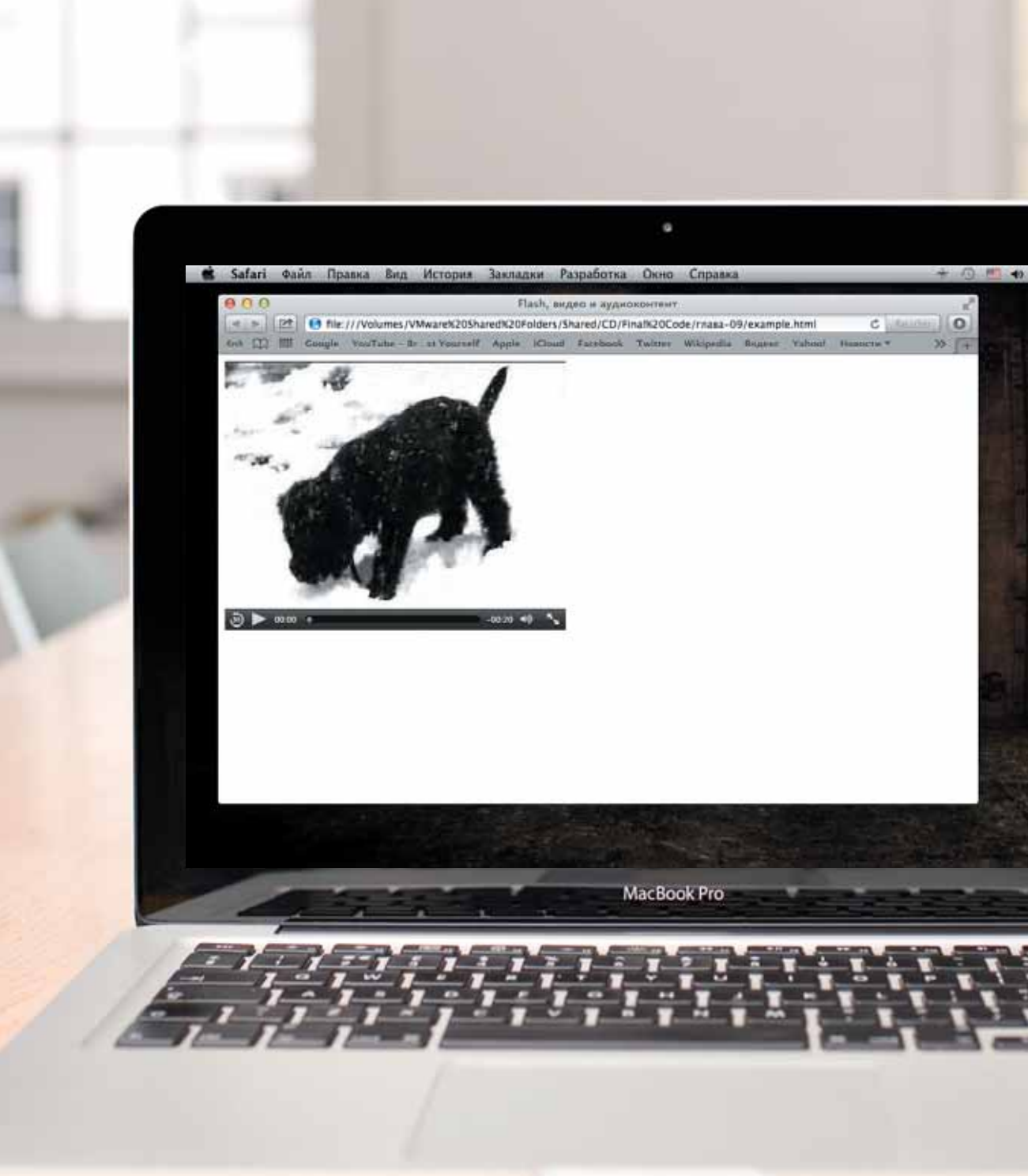
Элемент `<audio>` языка HTML5 не получил столь широкого распространения, как элемент `<video>`, и, кроме того, в первых версиях браузеров, поддерживавших этот элемент, возникали проблемы с качеством воспроизведения аудиозаписей.

<source>

С помощью элемента `<source>`, помещаемого между открывающим `<audio>` и закрывающим `</audio>` тегами, можно указать несколько аудиофайлов, используемых проигрывателем (вместо использования атрибута `src` элемента `<audio>`).
Это очень важно, так как разные браузеры поддерживают различные форматы файлов.

MP3: Safari 5, Chrome 6, Internet Explorer 9 и более поздние версии этих браузеров.
Ogg Vorbis: Firefox 3.6 Chrome 6, Opera 1.5, Internet Explorer 9 и более поздние версии этих браузеров.

Таким образом, чтобы все современные браузеры, поддерживающие элемент `<audio>`, без проблем воспроизводили ваш аудиоконтент, вам необходимо предоставить его в двух форматах. Кроме того, вы также можете воспользоваться технологией Flash в качестве альтернативы для старых версий браузеров, не поддерживающих элемент `<audio>`.


Safari Файл Правка Вид История Закладки Разработка Окно Справка
Flash, видео и аудиоконтент
file:///Volumes/VMware%20Shared%20Folders/Shared/CD/Final%20Code/глава-09/example.html
MacBook Pro

ПРИМЕР
FLASH, ВИДЕО- И АУДИОКОНТЕНТ

В данном примере для воспроизведения видеоролика используются возможности языка HTML5.

Данный видеофайл был конвертирован в форматы H.264 и WebM для воспроизведения в большинстве браузеров. Для браузеров, не поддерживающих элементы языка HTML5, был добавлен плагин Flash Player, встраиваемый с помощью сценария SWFObject. Если браузер посетителя не поддерживает ни HTML5, ни Flash, то на экране появится соответствующее текстовое сообщение.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Flash, видео и аудиоконтент</title>
    <script type="text/javascript"
      src="http://ajax.googleapis.com/ajax/libs/
      swfobject/2.2/swfobject.js"></script>
    <script type="text/javascript">
      var flashvars = {};
      var params = {movie: "../video/puppy.flv"};
      swfobject.embedSWF("flash/osplayer.swf", "snow",
      "400", "320", "8.0.0", flashvars, params);</script>
  </head>
  <body>
    <video poster="images/puppy.jpg" width="400"
      height="320" controls="controls">
      <source src="video/puppy.mp4" type='video/mp4;
        codecs="avc1.42E01E, mp4a.40.2"' />
      <source src="video/puppy.webm" type='video/webm;
        codecs="vp8, vorbis"' />
      <div id="snow">
        <p> Ваш браузер не поддерживает формат данного
            видеофайла.</p>
      </div>
    </video>
  </body>
</html>
```

ОБЗОР
FLASH, ВИДЕО- И АУДИОКОНТЕНТ

- Технология Flash позволяет добавлять на веб-страницы анимацию, видео- и аудиоконтент.
- Технология Flash не поддерживается на устройствах iPhone и iPad.
- В HTML5 появились новые элементы <video> и <audio>, позволяющие добавлять на веб-страницы видео- и аудиофайлы, но они поддерживаются только последними версиями браузеров.
- Не все браузеры, поддерживающие HTML5, могут воспроизводить аудио- и видеофайлы одинаковых форматов, поэтому вам необходимо предоставлять файлы разных форматов, чтобы все посетители вашего сайта могли просмотреть или прослушать их.

10

ВВЕДЕНИЕ В CSS

- Назначение каскадных таблиц стилей
- Принцип работы каскадных таблиц стилей
- Правила, свойства и значения

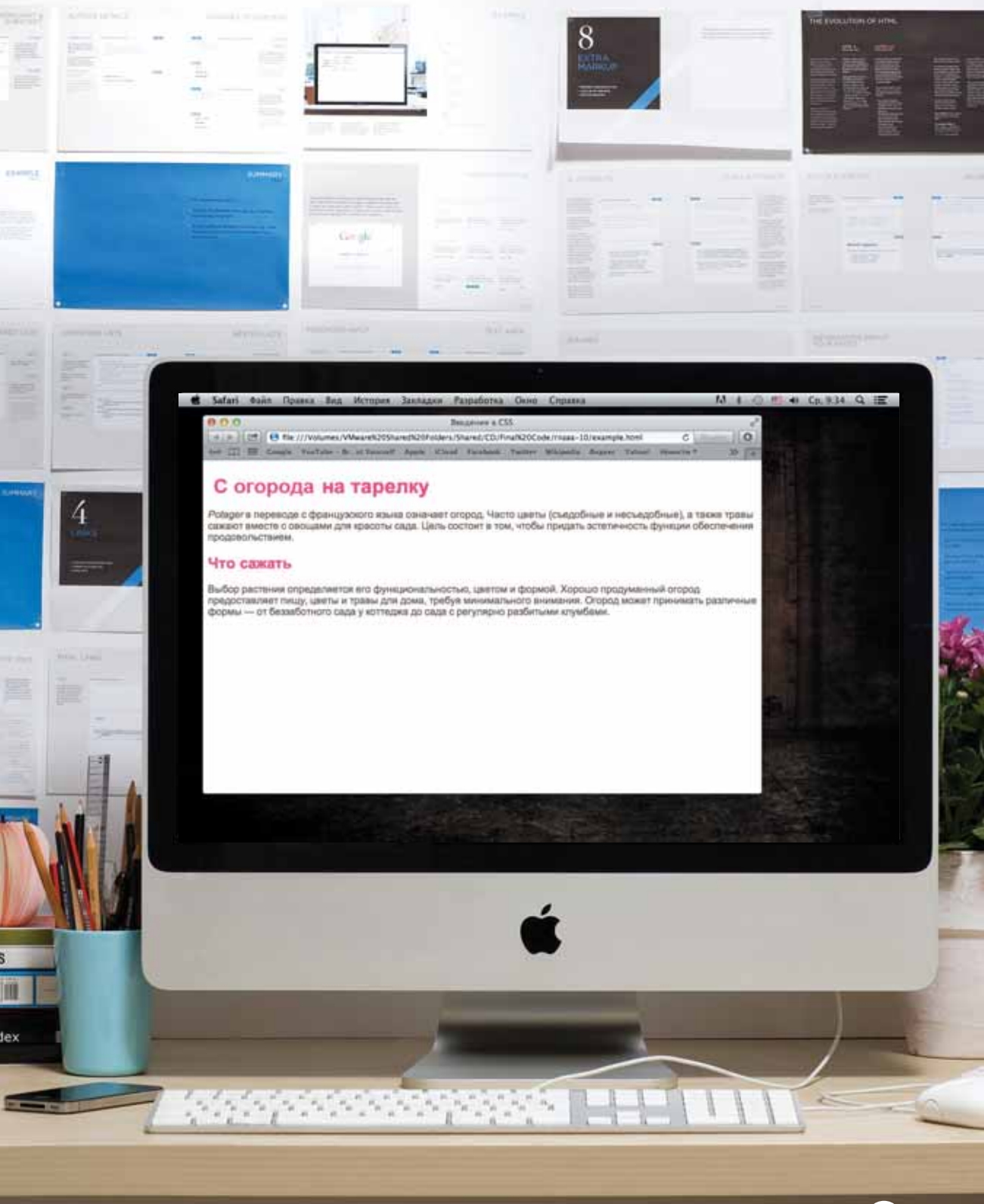
В этом разделе вы узнаете, как сделать веб-страницы более привлекательными, управляя их дизайном с помощью CSS.

Каскадные таблицы стилей позволяют создавать правила, которые определяют внешний вид контента. Например вы можете сделать так, чтобы фон страницы стал кремового оттенка, а текст абзацев был отформатирован серым цветом и гарнитурой Arial, или чтобы все заголовки первого уровня отображались гарнитурой Times, синим цветом и курсивом.

Когда вы узнаете, как писать правила CSS, освоение данной темы будет сводиться к изучению различных свойств, которые можно использовать.
Таким образом, в этой главе вы:

- узнаете, как работает CSS;
- научитесь писать правила CSS;
- увидите, как правила CSS применяются к HTML-страницам.

Остальные главы этого раздела будут посвящены различным свойствам CSS, которые вы можете использовать.


Safari Файл Правка Вид История Закладки Разработка Окно Справка
Ср, 9:34
Введение в CSS
file:///Volumes/VMware%20Shared%20Folders/Shared/CD/Final%20Code/ch10-10/example.html
С огорода на тарелку
Potager в переводе с французского языка означает огород. Часто цветы (съедобные и несъедобные), а также травы сажают вместе с овощами для красоты сада. Цель состоит в том, чтобы придать эстетичность функции обеспечения продовольствием.
Что сажать
Выбор растения определяется его функциональностью, цветом и формой. Хорошо продуманный огород предоставляет пищу, цветы и травы для дома, требуя минимального внимания. Огород может принимать различные формы — от безаботного сада у коттеджа до сада с регулярно разбитыми клумбами.

ПОНИМАНИЕ CSS: ИЗНУТРИ БЛОКА

Чтобы понять, как работает CSS, представьте себе, что каждый HTML-элемент заключен в невидимый блок.

На этом рисунке представлен простой HTML-документ.

На следующей странице показан тот же HTML-документ, все элементы которого выделены рамками, чтобы вы могли видеть, как CSS будет обрабатывать их.

БЛОЧНЫЕ И ВСТРОЕННЫЕ ЭЛЕМЕНТЫ

Ранее в книге говорилось о разнице между блочными и встроенными элементами и о том, как браузеры отображают их.

Блочные элементы начинаются с новой строки. Примерами могут быть элементы `<h1>` — `<h6>`, `<p>` и `<div>`.

Встроенные элементы располагаются в тексте и не начинаются с новой строки. Примеры таких элементов: `<b>`, `<i>`, `<img>`, `<em>` и `<span>`.

Язык CSS позволяет создавать правила, определяющие представление каждого отдельного блока (и его содержимого).

В данном примере блочные элементы выделены красной рамкой, а встроенные элементы — голубой. Элемент `<body>` формирует первый блок, затем внутри него элементы `<h1>`, `<h2>`, `<p>`, `<i>` и `<a>` создают собственные блоки. Используя CSS, вы можете создать рамку вокруг любого из блоков, указать ее ширину, высоту и цвет фона. Вы также можете форматировать текст внутри блока — например изменить цвет, размер и гарнитуру шрифта.

ПРИМЕРЫ СТИЛЕЙ

БЛОКИ

Ширина и высота;
границы (цвет, толщина и стиль);
цвет фона и фоновые изображения;
расположение в окне браузера.

ТЕКСТ

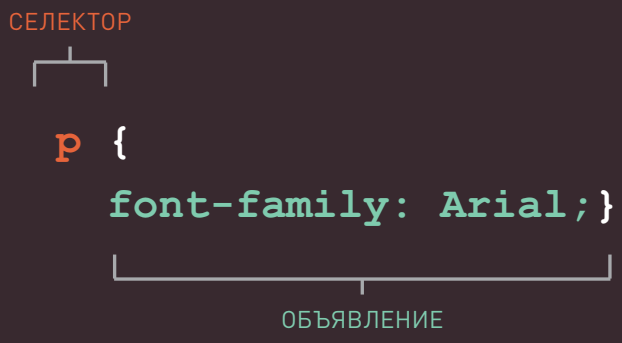
Гарнитура шрифта;
размер;
цвет;
курсив, полужирное начертание, прописные буквы, строчные буквы, капитель.

ОСОБЫЙ

Также существует возможность особым образом настроить стиль определенных элементов, например списков, таблиц и форм.

ТАБЛИЦЫ CSS СВЯЗЫВАЮТ ПРАВИЛА СТИЛЕЙ С HTML-ЭЛЕМЕНТАМИ

Каскадные таблицы стилей связывают правила с HTML-элементами. Эти правила определяют, как должны отображаться указанные элементы. Правило CSS состоит из двух частей: селектора и объявления.



Данное правило означает, что все элементы <p> должны быть отформатированы гарнитурой Arial.

Селекторы указывают, к какому элементу применяется правило. Одно и то же правило может применяться более чем к одному элементу, если вы отделите имена элементов запятыми.

Объявления определяют, как должны отображаться элементы, указанные в селекторе. Объявления состоят из двух частей (свойство и значение), разделенных двоеточием.

СВОЙСТВА CSS ОПРЕДЕЛЯЮТ ВНЕШНИЙ ВИД ЭЛЕМЕНТОВ

Объявления CSS заключаются в фигурные скобки и состоят из двух частей: свойства и значения, разделенных двоеточием. В одном объявлении вы можете указать несколько свойств, отделив их друг от друга точкой с запятой.

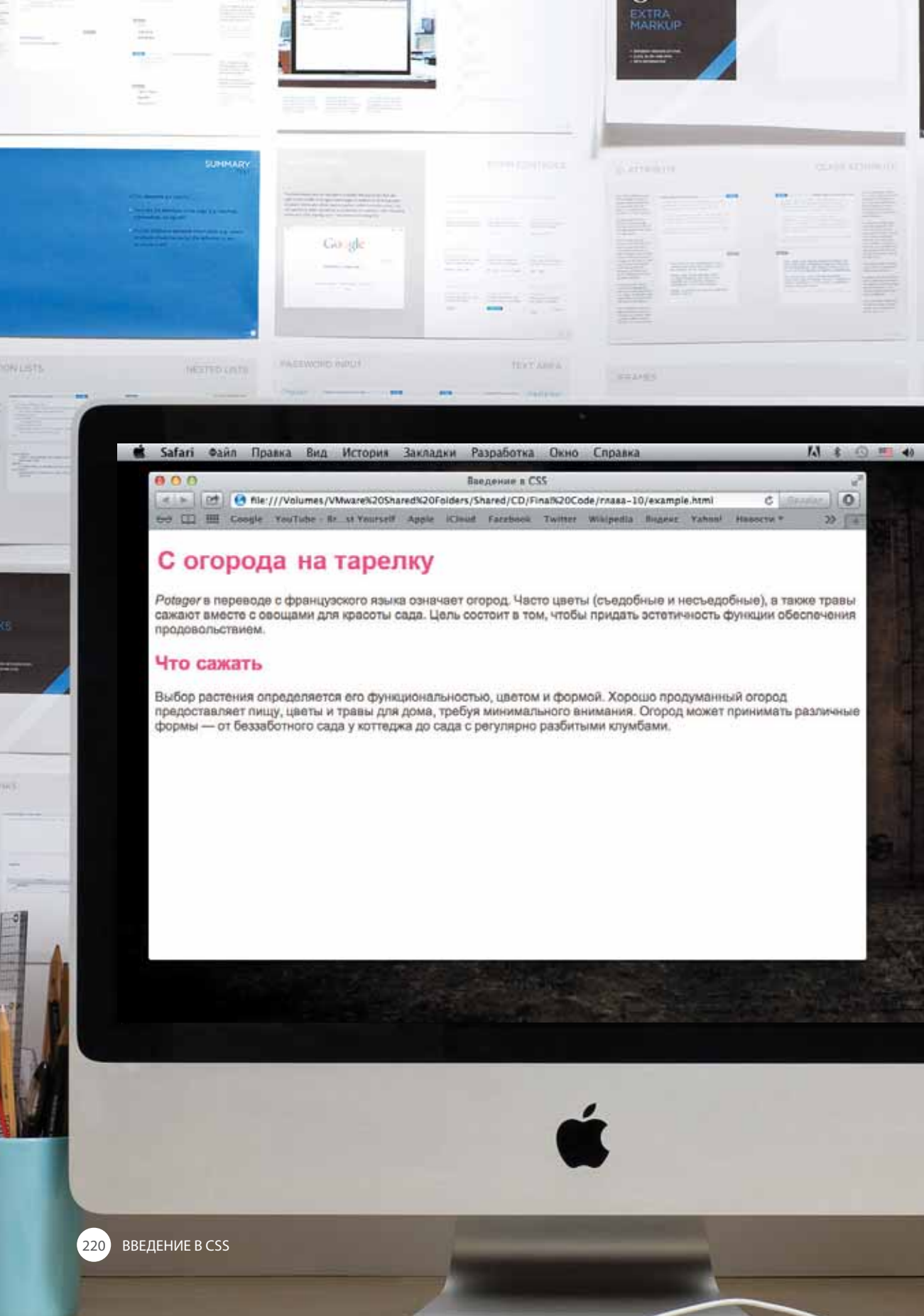
```
h1, h2, h3 {
        font-family: Arial;
        color: yellow;}
```

СВОЙСТВО ЗНАЧЕНИЕ

Данное правило означает, что все элементы <h1>, <h2> и <h3> должны быть отформатированы гарнитурой Arial и желтым цветом.

Свойства определяют аспекты элемента, которые вы хотите изменить. Например цвет, шрифт, ширину, высоту и границу.

Значения — это параметры, которые вы хотите использовать для выбранных свойств. Например если вам необходимо указать свойство цвета, то значением будет цвет, которым вы хотите выделить текст в указанных элементах.


Safari Файл Правка Вид История Закладки Разработка Окно Справка
Введение в CSS
file:///Volumes/VMware%20Shared%20Folders/Shared/CD/Final%20Code/глава-10/example.html
Google YouTube Apple iCloud Facebook Twitter Wikipedia Яндекс Yahoo! Новости
С огорода на тарелку
Potager в переводе с французского языка означает огород. Часто цветы (съедобные и несъедобные), а также травы сажают вместе с овощами для красоты сада. Цель состоит в том, чтобы придать эстетичность функции обеспечения продовольствием.
Что сажать
Выбор растения определяется его функциональностью, цветом и формой. Хорошо продуманный огород предоставляет пищу, цветы и травы для дома, требуя минимального внимания. Огород может принимать различные формы — от беззаботного сада у коттеджа до сада с регулярно разбитыми клумбами.

ПРИМЕР
ВВЕДЕНИЕ В CSS

Здесь вы можете увидеть простую веб-страницу, отформатированную с помощью CSS.

В данном примере используются два документа: файл HTML (*example.html*) и отдельный файл CSS (*example.css*). В пятой строке HTML-кода используется элемент `<link>`, указывающий, где находится файл CSS.
Далее вы узнаете, как правила CSS могут включаться в HTML-страницу, а также о ситуациях, в которых это бывает нужным.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Введение в CSS</title>
    <link href="css/example.css" type="text/css"
      rel="stylesheet" />
  </head>
  <body>
    <h1>С огорода на тарелку</h1>
    <p><i>  Potager</i> в переводе с французского языка
            означает огород...</p>
    <h2>    Что сажать</h2>
    <p>     Выбор растения определяется его
            функциональностью, цветом и формой...</p>
  </body>
</html>
```

```
body {
  font-family: Arial, Verdana, sans-serif;}
h1, h2 {
  color: #ee3e80;}
p {
  color: #665544;}
```

ВНЕШНИЕ КАСКАДНЫЕ ТАБЛИЦЫ СТИЛЕЙ

<link>

Элемент <link> в HTML-документе сообщает браузеру, где находится файл CSS, используемый для форматирования страницы. Это пустой элемент (он не нуждается в наличии закрывающего тега), который располагается внутри элемента <head>. Элемент <link> должен использовать три следующих атрибута.

href

Указывает путь к файлу CSS, который часто расположен в папке с именем *css* или *styles*.

type

Данный атрибут определяет тип документа, на который указывает ссылка. Должно быть указано значение text/css.

rel

Определяет отношение между HTML-страницей и связанным файлом. При создании ссылки на файл CSS должно быть указано значение stylesheet.

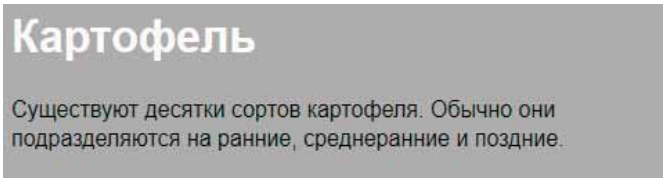
глава-10/using-external-css.html HTML

```
<!DOCTYPE html>
<html>
  <head>
    <title>Использование внешней каскадной таблицы
      стилей</title>
    <link href="css/styles.css" type="text/css"
      rel="stylesheet" />
  </head>
  <body>
    <h1>Картофель</h1>
    <p> Существуют десятки сортов картофеля.
        Обычно они подразделяются на ранние,
        среднеранние и поздние.</p>
  </body>
</html>
```

глава-10/styles.css CSS

```
body {
    font-family: arial;
    background-color: rgb(185,179,175);}
h1 {
    color: rgb(255,255,255);}
```

РЕЗУЛЬТАТ



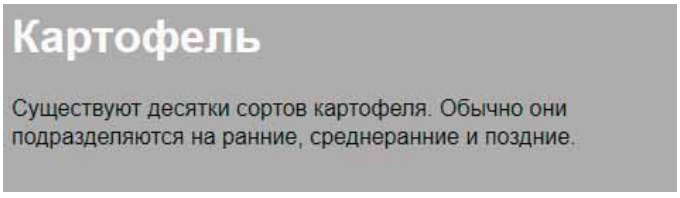
В коде HTML-страницы можно использовать более чем одну таблицу стилей. В этом случае для каждого файла CSS необходимо указать отдельный элемент <link>. Например некоторые дизайнеры используют один файл CSS для форматирования (например, шрифтами и цветами), а второй — для управления макетом.

ВНУТРЕННИЕ КАСКАДНЫЕ ТАБЛИЦЫ СТИЛЕЙ

HTML + CSS **глава-10/using-internal-css.html**

```
<!DOCTYPE html>
<html>
  <head>
    <title>Использование внутренней каскадной
      таблицы стилей</title>
    <style type="text/css">
       body {
          font-family: arial;
          background-color: rgb(185,179,175);}
       h1 {
          color: rgb(255,255,255);}
    </style>
  </head>
  <body>
    <h1>Картофель</h1>
    <p> Существуют десятки сортов картофеля.
        Обычно они  подразделяются на ранние,
        среднеранние и поздние.</p>
  </body>
</html>
```

РЕЗУЛЬТАТ



<style>

Вы можете включить правила CSS в HTML-страницу, поместив их внутри элемента `<style>`, который обычно находится в элементе `<head>`.

Элемент `<style>` должен использовать атрибут `type` для указания, что стили описаны в CSS. Нужно указать значение `text/css`.

При создании сайта, состоящего более чем из одной страницы, вам следует использовать внешнюю таблицу стилей, поскольку это:

- позволяет применять одни и те же правила ко всем страницам, вместо того чтобы повторять их на каждой;
- отделяет контент от представления;
- означает, что вы можете поменять используемые на всех страницах стили, отредактировав только один файл (а не каждую страницу).

В языке HTML 4 и переходной версии XHTML вы также можете применить атрибут `style` к большинству элементов, расположенных в теле страницы. Правила CSS, указанные в значении этого атрибута, будут применяться только к данному элементу. Вам следует избегать использования этого атрибута при создании нового сайта, мы упоминаем о нем здесь только потому, что он может встретиться вам на старых веб-страницах. Ниже приведен пример кода, который изменяет цвет текста в отдельном абзаце на красный:

```
<p style="color:red;">
```

СЕЛЕКТОРЫ CSS

Существует множество различных типов селекторов CSS, позволяющих применять правила к конкретным элементам HTML-документа.
В следующей таблице перечислены наиболее часто используемые селекторы.
На этой странице представлен HTML-файл, показывающий, к каким элементам будут применяться селекторы.

Селекторы CSS чувствительны к регистру, поэтому они должны точно соответствовать именам элементов и значениям атрибутов.
Существуют более продвинутые селекторы, позволяющие выбирать элементы на основании атрибутов и их значений. Об них вы узнаете далее в книге.
Последние два селектора в таблице (родственные селекторы) Internet Explorer поддерживает, начиная с версии 7, поэтому они применяются реже, чем другие перечисленные селекторы.

глава-10/css-selectors.html HTML

```
<!DOCTYPE html>
<html>
  <head>
    <title>Селекторы CSS</title>
  </head>
  <body>
    <h1 id="top">Посевной календарь</h1>
    <p id="introduction">Здесь вы можете прочитать
      нашу полезную инструкцию о том, что и когда
      следует делать.</p>
    <h2>Весна</h2>
    <ul>
      <li><a href="mulch.html"> Удобрение овощных
        грядок</a></li>
      <li><a href="potato.html"> Посадка раннего
        картофеля</a></li>
      <li><a href="tomato.html"> Посев семян
        томатов</a></li>
      <li><a href="beet.html"> Посев семян
        свеклы</a></li>
      <li><a href="zucchini.html"> Посев семян
        кабачка</a></li>
      <li><a href="rhubarb.html"> Срезание увядших
        листьев ревеня</a></li>
    </ul>
    <p class="note">Эта страница написана
      <a href="mailto:ivy@example.org">
        ivy@example.org</a> для
        <a href="http://www.example.org">примера</a>.
    </p>
    <p>
      <a href="#top">К началу страницы</a>
    </p>
  </body>
</html>
```

SELECTOR	MEANING	EXAMPLE
УНИВЕРСАЛЬНЫЙ СЕЛЕКТОР	Применяется ко всем элементам в документе	`* {}` Все элементы страницы
СЕЛЕКТОР ТИПА	Соответствует именам элементов	`h1, h2, h3 {}` Элементы <h1>, <h2> и <h3>

СЕЛЕКТОР КЛАССА	Соответствует элементу, чей атрибут `class` имеет значение, совпадающее с указанным после точки	`.note {}`
		Элемент, чей атрибут `class` имеет значение `note`
		`p.note {}`
		Только элементы <p>, чьи атрибуты `class` имеют значение `note`

СЕЛЕКТОР ИДЕНТИФИКАТОРА	Соответствует элементу, чей атрибут `id` имеет значение, совпадающее с указанным после символа #	`#introduction {}` Элемент, чей атрибут `id` имеет значение `introduction`
ДОЧЕРНИЙ СЕЛЕКТОР	Соответствует элементу, являющемуся прямым потомком другого элемента	`li>a {}` Элементы <a>, являющиеся прямыми потомками элемента , но не остальные элементы <a> на странице
СЕЛЕКТОР ПОТОМКА	Соответствует элементу, являющемуся потомком другого элемента (не обязательно прямым)	`p a {}` Любые элементы <a>, располагающиеся внутри <p>, даже в том случае, когда между ними находятся другие элементы
СЕЛЕКТОР СЕСТРИНСКОГО ЭЛЕМЕНТА	Соответствует элементу, являющемуся сестринским по отношению к другому элементу	`h1+p {}` Первый элемент <p>, располагающийся после любого элемента <h1>, но не любые другие элементы <p>
СЕЛЕКТОР ОБОБЩЕННЫХ РОДСТВЕННЫХ ЭЛЕМЕНТОВ	Соответствует элементу, являющемуся сестринским по отношению к другому элементу, хотя не обязательно расположенному перед ним	`h1~p {}` Если бы у вас было два элемента <p>, являющихся сестринскими по отношению к элементу <h1>, то это правило бы применялось к обоим

ПРИНЦИПЫ КАСКАДИРОВАНИЯ ПРАВИЛ CSS

Если к одному и тому же элементу применяется два или более правил, важно понимать, которое из них получит приоритет.

ПОСЛЕДНЕЕ ПРАВИЛО

При наличии двух одинаковых селекторов приоритет получает последний. В примере вы можете видеть, как второй селектор `i` получает приоритет над первым.

СПЕЦИФИЧНОСТЬ

Если один селектор является более специфичным, чем остальные, то он получает приоритет. В данном примере:
`h1` является более специфичным, чем `*`;
`p b` является более специфичным, чем `p`;
`p#intro` является более специфичным, чем `p`.

ВАЖНОСТЬ

Вы можете добавить текст `!important` после значения любого свойства, чтобы указать, что его следует считать самым важным из правил, относящихся к одному и тому же элементу.

Понимание принципов каскадирования правил CSS позволит вам писать более простые таблицы стилей, поскольку вы сможете создавать общие правила, которые относятся к большинству элементов, а затем менять свойства отдельных элементов, которые должны выглядеть по-другому.

глава-10/cascade.html HTML

```
<h1>Картофель</h1>
<p id="intro">Существуют <i>десятки</i> сортов
  <b>картофеля</b>.</p>
<p>Обычно они подразделяются на ранние, среднеранние
  и поздние.</p>
```

CSS

```
* {
  font-family: Arial, Verdana, sans-serif;}
h1 {
  font-family: "Courier New", monospace;}
i {
  color: green;}
i {
  color: red;}
b {
  color: pink;}
p b {
  color: blue !important;}
p b {
  color: violet;}
p#intro {
  font-size: 100%;}
p {
  font-size: 75%;}
```

РЕЗУЛЬТАТ

Картофель

Существуют *десятки* сортов **картофеля**.

Обычно они подразделяются на ранние, среднеранние и поздние.

НАСЛЕДОВАНИЕ

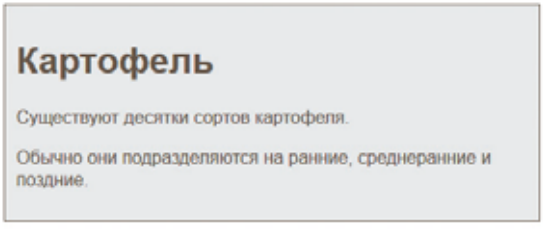
HTML **глава-10/inheritance.html**

```
<div class="page">
  <h1>Картофель</h1>
  <p> Существуют десятки сортов картофеля.</p>
  <p> Обычно они подразделяются на ранние,
      среднеранние и поздние.</p>
</div>
```

CSS

```
body {
  font-family: Arial, Verdana, sans-serif;
  color: #665544;
  padding: 10px;}
.page {
  border: 1px solid #665544;
  background-color: #efefef;
  padding: inherit;}
```

РЕЗУЛЬТАТ



Если вы укажете свойства `font-family` или `color` для элемента `<body>`, то они будут применяться к большинству дочерних элементов, поскольку значение свойства `font-family` **наследуется** дочерними элементами. Это избавляет вас от необходимости применять свойства к каждому элементу по отдельности, а также упрощает таблицу стилей.

Для сравнения взгляните на свойство `background-color` или `border` — они **не наследуются** дочерними элементами. Если бы они наследовались, то страница бы выглядела не очень красиво.

Чтобы свойства наследовали значения родительских элементов, вы можете использовать `inherit` в качестве их значения. В данном примере элемент `<div>` с классом `page` наследует размер отступа от правила CSS, которое относится к элементу `<body>`.

КОГДА СЛЕДУЕТ ИСПОЛЬЗОВАТЬ ВНЕШНИЕ ТАБЛИЦЫ СТИЛЕЙ

При создании сайта размещение правил CSS в отдельной таблице стилей дает ряд преимуществ.

Все ваши веб-страницы могут использовать одну и ту же таблицу стилей. Такое достигается путем использования на каждой из них элемента `<link>`, который связывает страницу с одним и тем же документом CSS. Это означает, что вам не нужно повторять один и тот же код на каждой странице, что приводит к уменьшению размера HTML-документов.

Таким образом, как только пользователь загрузит таблицу стилей CSS, остальная часть сайта будет загружаться быстрее. Если вы захотите изменить внешний вид сайта, вам придется отредактировать только один файл CSS, и все ваши страницы обновятся. Например вы можете изменить стиль каждого элемента `<h1>`, внеся правки только в таблицу стилей CSS, вместо того чтобы менять правила CSS на каждой странице. HTML-код станет более внятным и поддающимся редактированию, потому что он не будет содержать изрядного количества правил CSS. Хорошей практикой считается отделять содержимое сайта от правил, которые определяют его внешний вид.

Иногда возникает необходимость поместить правила CSS на страницу с HTML-кодом.

Если вы создаете единственную страницу, то можете предпочесть поместить правила в тот же файл, чтобы все было в одном месте. (Тем не менее многие дизайнеры считают, что лучше помещать правила CSS в отдельный файл.)

Если у вас есть страница, которой требуется несколько дополнительных правил, не используемых на остальном сайте, вы можете поместить их на ту же страницу. (Опять же, многие дизайнеры предпочитают помещать правила CSS в отдельный файл.)

В большинстве примеров в книге правила CSS расположены в разделе `<head>` документа (с использованием элемента `<style>`), а не в отдельном документе. Это сделано, чтобы вам не приходилось открывать два файла.

ПРОБЛЕМЫ СОВМЕСТИМОСТИ ПРАВИЛ CSS С БРАУЗЕРАМИ

Спецификация CSS1 была реализована в 1996 году, а CSS2 — двумя годами позже. Разработка версии CSS3 все еще продолжается, однако основные браузеры уже поддерживают ее.

Так же как и в случае с HTML, существуют различные версии CSS.

Браузеры не сразу стали поддерживать все функции CSS, поэтому некоторые старые версии не распознают все свойства.

В последнем случае вы увидите, что свойства CSS ведут себя не так, как ожидается.

Любой опытный пользователь CSS скажет вам, что иногда тот или иной браузер отображает некоторые свойства CSS не совсем так, как ожидается (или даже совсем не так). Поиск и устранение этих ошибок упрощаются, когда вы знаете, что нужно делать.

Перед запуском нового сайта важно протестировать его в нескольких браузерах, поскольку они могут по-разному отображать страницы.

Вам не обязательно иметь несколько компьютеров для проверки сайта. Во Всемирной паутине существуют инструменты, позволяющие оценить, как страница выглядит в нескольких браузерах:
BrowserCam.com;
BrowserLab.Adobe.com;
BrowserShots.org;
CrossBrowserTesting.com.

С помощью этих инструментов следует протестировать сайт в различных операционных системах (Windows, OS X и Linux), а также в различных версиях основных браузеров.

При просмотре сайта в разных браузерах вы можете заметить, что некоторые элементы на вашей странице выглядят не так, как вы ожидали.

Когда свойство CSS ведет себя неожиданным образом, это называется **проблемой совместимости с браузером** или **ошибкой CSS**.

Некоторые распространенные ошибки браузеров обсуждаются в этой книге, однако существует множество несущественных ошибок, возникающих в редких случаях или только в старых, малоиспользуемых браузерах.

Если вы столкнетесь с ошибкой CSS, попытайтесь найти решение с помощью поисковой системы.

ОБЗОР
ВВЕДЕНИЕ В CSS

- CSS обращается с каждым элементом HTML так, будто он заключен в блок, и использует правила, чтобы указать, как этот элемент должен выглядеть.
- Правила состоят из селекторов, определяющих элементы, к которым применяется правило, и объявлений, указывающих, как эти элементы должны выглядеть.
- Различные типы селекторов позволяют применять правила к разным элементам.
- Объявления состоят из двух частей: свойств элемента, которые вы хотите изменить, и значений этих свойств. Например свойство font-family определяет шрифт, а значение arial указывает в качестве предпочтительной гарнитуры Arial.
- Правила CSS обычно помещаются в отдельный документ, хотя и могут быть указаны в коде HTML-страницы.

11

ЦВЕТ

- Как указывать цвет
- Терминология цвета и контраст
- Цвет фона

Цвет может по-настоящему оживить ваши страницы.

В этой главе вы узнаете:

- три распространенных способа, которые вы можете использовать для указания цвета, а также дополнительные способы, доступные в CSS3;
- терминологию цвета, весьма полезную, когда дело доходит до его выбора;
- о контрастности и обеспечении читабельности текста;
- о фоновых цветах, которые вы можете назначить всей странице или только ее части.

То, что вы узнаете о цветах в этой главе, будет использоваться далее в книге, когда речь пойдет о цветах текста и рамок в CSS.

AVEC
METAPOL
Pâte ou liquide
FER·FONTE·ACIER
retrouvent
le Poli du neuf
Letratone
LT1000
HIGH TEA
Шкала pH

ОСНОВНОЙ ЦВЕТ
color

Свойство `color` позволяет указать цвет текста внутри элемента. В CSS вы можете задать любой цвет одним из трех способов.

ЗНАЧЕНИЯ RGB

Эти значения выражают цвет в пропорциях красного, зеленого и синего по модели RGB*, например `rgb(100,100,90)`.

ШЕСТНАДЦАТЕРИЧНЫЕ КОДЫ

Это шестизначные коды, определяющие количество красного, зеленого и синего цвета, в записи им предшествует символ решетки (#), например `#ee3e80`.

ИМЕНА ЦВЕТОВ

Существует 147 предопределенных имен цветов, которые распознаются браузерами, например `DarkCyan`.

глава-11/foreground-color.html CSS

```
/* имя цвета */
h1 {
  color: DarkCyan;}
/* шестнадцатеричный код */
h2 {
  color: #ee3e80;}
/* значение rgb */
p {
  color: rgb(100,100,90);}
```

РЕЗУЛЬТАТ

Далее в главе мы рассмотрим все три способа более подробно.
В CSS3 появился еще один способ задания цвета — HSLA, вы узнаете о нем в конце главы.

Над каждым правилом CSS в приведенном примере приведены комментарии. Все, что находится между символами `/*` и `*/`, не будет обработано браузером. Комментарии выделены голубым цветом. Использование комментариев помогает понять файл CSS, а также организовать его путем разделения длинного документа на разделы. Здесь мы поставили комментарии, чтобы показать, какой метод используется для задания каждого из типов цветов.

* RGB (red, green, blue) (*англ.*) — красный, зеленый, синий. — *Примеч. ред.*

ЦВЕТ ФОНА
background-color

CSS **глава-11/background-color.html**

```
body {
  background-color: rgb(200,200,200);}
h1 {
  background-color: DarkCyan;}
h2 {
  background-color: #ee3e80;}
p {
  background-color: white;}
```

РЕЗУЛЬТАТ

Каскадная таблица стилей обращается с каждым HTML-элементом так, будто он заключен в блок, а свойство `background-color` задает цвет фона этого блока.

Вы можете указать цвет фона теми же тремя способами, что и для задания основного цвета, используя значения RGB, шестнадцатеричные коды и имена цветов.

Если вы не укажете цвет фона, то он будет прозрачным.

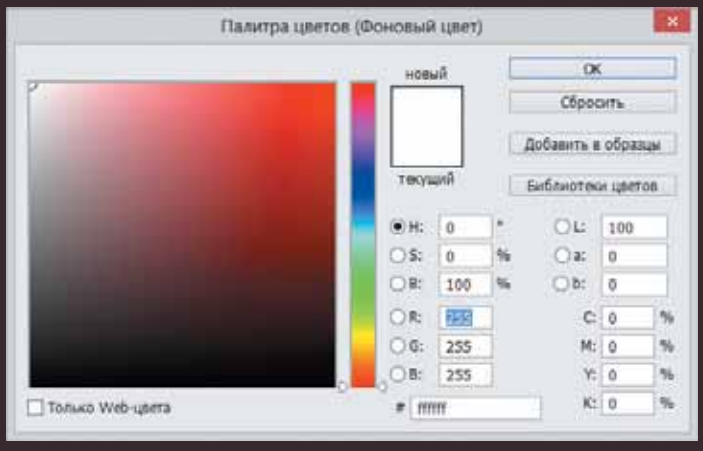
По умолчанию окна большинства браузеров имеют белый фон, однако пользователи могут самостоятельно задавать цвет фона в настройках браузера, поэтому, если вы хотите знать наверняка, что он будет нужного вам цвета, примените свойство `background-color` к элементу `<body>`.

Мы также использовали свойство `padding`, чтобы отделить текст от краев блоков и улучшить его восприятие. Вы узнаете больше об этом свойстве в главе 13.

ПОНИМАНИЕ ЦВЕТА

Каждый оттенок на экране компьютера создается путем смешивания в разных пропорциях красного, зеленого и синего цветов. Чтобы найти нужный вам цвет, вы можете использовать цветовую палитру.

Мониторы компьютеров состоят из тысяч крошечных квадратов, называемых пикселами (если вы очень внимательно посмотрите на свой монитор, то сможете увидеть их). Выключенный экран черный, поскольку он не излучает свет. Когда он включен, каждый пиксел принимает свой цвет, что создает изображение. Цвет каждого пиксела на экране компьютера выражается через сочетание красного, зеленого и синего — так же, как и на экране телевизора.



Инструменты для выбора цвета доступны в графических редакторах, например в Photoshop и GIMP. Значения RGB указаны рядом с переключателями **R**, **G** и **B**. Шестнадцатеричные коды указаны рядом с символом решетки (#). Удобный инструмент для выбора цвета можно найти по адресу **colorschemedesigner.com**.

ЗНАЧЕНИЯ RGB

Количество красного, зеленого и синего цветов выражается числами от 0 до 255.

`rgb(102,205,170)`

Данный цвет состоит из следующих значений:
102 — красный;
205 — зеленый;
170 — синий

ШЕСТНАДЦАТЕРИЧНЫЕ КОДЫ

Значения красного, зеленого и синего цветов выражаются в виде шестнадцатеричного кода.

`#66cdaa`

Значение красного цвета, 102, в шестнадцатеричном коде соответствует значению 66. Значение зеленого, 205, — cd, а значение синего, 170, — aa.

ИМЕНА ЦВЕТОВ

Цвета выражаются предопределенными именами, количество которых ограничено.

`MediumAquaMarine`

Существует 147 предопределенных имен цветов, которые распознаются браузерами (в данном примере используется цвет с именем MediumAquaMarine). Многие считают эту палитру ограниченной, кроме того, довольно сложно запомнить имя каждого цвета на английском языке, поэтому такой способ задания цвета используется не часто.

ЦВЕТОВОЙ ТОН

Цветовой тон близок к понятию цвета. Однако технически кроме тона цвет может также иметь значения насыщенности и светлоты.

НАСЫЩЕННОСТЬ

Насыщенность определяется количеством серого цвета. При максимальной насыщенности в цвете отсутствует серая составляющая. При малой насыщенности серый будет преобладать.

ЯРКОСТЬ

Яркость определяется количеством черного цвета. При максимальной яркости в цвете отсутствует черный. При минимальной яркости цвет будет очень темным.

КОНТРАСТ

При выборе основного и фонового цветов важно обеспечить достаточный контраст, чтобы текст был читаемым.

НИЗКИЙ КОНТРАСТ

При низком контрасте между основным и фоновым цветами текст трудно читать.
Недостаток контраста представляет особую сложность для людей, имеющих проблемы со зрением.
Также это мешает при работе с плохими мониторами или при попадании на экран солнечного света, что случается довольно часто при использовании портативных устройств на улице.

ВЫСОКИЙ КОНТРАСТ

Текст легко читать, когда между основным и фоновым цветами существует высокий контраст.
Однако если на вашей странице много текста, то слишком высокий контраст также может усложнить чтение.

Для упрощения восприятия светлого текста на темном фоне можно увеличить междустрочный интервал или вес шрифта.

СРЕДНИЙ КОНТРАСТ

При большом объеме текста некоторое уменьшение контрастности улучшает его удобочитаемость.
Вы можете уменьшить контраст, используя темно-серый цвет текста на белом фоне или желтоватый цвет текста на темном фоне.

Вы можете найти удобный инструмент для проверки контрастности по адресу **www.snook.ca/technical/colour_contrast/colour.html**

CSS3: НЕПРОЗРАЧНОСТЬ

opacity, rgba

CSS — глава-11/opacity.html

```
p.one {
  background-color: rgb(0,0,0);
  opacity: 0.5;}
p.two {
  background-color: rgb(0,0,0);
  background-color: rgba(0,0,0,0.5);}
```

РЕЗУЛЬТАТ

РЕЗУЛЬТАТ В СТАРОМ БРАУЗЕРЕ

В спецификации CSS3 появилось свойство `opacity`, которое позволяет указать степень непрозрачности элемента и любого из его дочерних элементов. Значением является число в диапазоне от 0,0 до 1,0, поэтому значение 0,5 соответствует 50%-ной непрозрачности, а 0.15 — 15%-ной.

Свойство CSS3 `rgba` позволяет указать цвет так же, как и `rgb`, однако добавляет четвертое значение для указания степени непрозрачности. Данный параметр, известный как альфа-канал, принимает значения в диапазоне от 0,0 до 1,0 (так, значение 0,5 соответствует 50%-ной непрозрачности, а 0,15 — 15%-ной). Значение `rgba` влияет только на тот элемент, к которому оно применяется (дочерние элементы не затрагиваются).

Поскольку некоторые браузеры не поддерживают цвета RGBA, вы можете предусмотреть для таких случаев отображение сплошного цвета. Если к одному и тому же элементу применяются два правила, то приоритет получает последнее них. Чтобы предусмотреть запасной вариант, вы можете указать цвет с помощью шестнадцатеричного кода, имени цвета или значения RGB, а далее поместить правило, указывающее значение RGBA. Если браузер поддерживает цвета RGBA, то он будет использовать это правило, а если нет — обратится к значениию RGB.

CSS3: ЦВЕТОВАЯ МОДЕЛЬ HSL

В CSS3 появился полностью новый и интуитивно понятный способ определения цвета с помощью значений цветового тона, насыщенности и светлоты.

ЦВЕТОВОЙ ТОН

Понятие цветового тона близко к понятию цвета. В модели HSL тон определяется либо с помощью цветового круга, где угол представляет цвет, либо в виде ползункового регулятора со значениями в диапазоне от 0 до 360.

НАСЫЩЕННОСТЬ

Насыщенность указывает на количество серого цвета. Степень насыщенности выражается в процентах. Значение 100% соответствует полной насыщенности, а 0% — оттенку серого.

СВЕТЛОТА

Светлота — это количество белого (яркий) или черного (темный) в цвете. Светлота выражается в процентах. 0% соответствует черному цвету, 100% — белому, 50% — это норма. Иногда светлоту называют *светимостью*..

Обратите внимание, что светлота* отличается от яркости**. В графических редакторах, таких как Photoshop и GIMP, цветовые палитры используют значения оттенка, насыщенности и яркости, однако яркость влияет только на количество черного цвета, в то время как параметр светлоты позволяет регулировать количество и черного, и белого.

* Lightness (*англ.*). — *Примеч. ред.*
** Brightness (*англ.*). — *Примеч. ред.*

CSS3: HSL И HSLA

hsl, hsla

CSS **глава-11/hsla.html**

```
body {
  background-color: #C8C8C8;
  background-color: hsl(0,0%,78%);}
p {
  background-color: #ffffff;
  background-color: hsla(0,100%,100%,0.5);}
```

РЕЗУЛЬТАТ

Поскольку старые версии браузеров не поддерживают значения HSL и HSLA, следует добавить дополнительное правило, определяющее цвет с помощью шестнадцатеричного кода, значения RGB или имени. Данное правило должно находиться *перед* правилом, использующим значение HSL или HSLA.

Это обеспечивает обходной путь, поскольку в CSS при наличии двух правил, применяемых к одному и тому же элементу, приоритет всегда получает последнее из них. Это означает, что если браузер поддерживает цвета HSL и HSLA, то он будет использовать второе правило, а если не поддерживает — первое.

Свойство `hsl` было добавлено в спецификации CSS3 в качестве альтернативного способа указания цвета. Значение данного свойства начинается с букв `hsl`, за которыми в скобках следуют значения.

ОТТЕНОК

Выражается как значение угла в диапазоне от 0 до 360 градусов.

НАСЫЩЕННОСТЬ

Выражается в процентах.

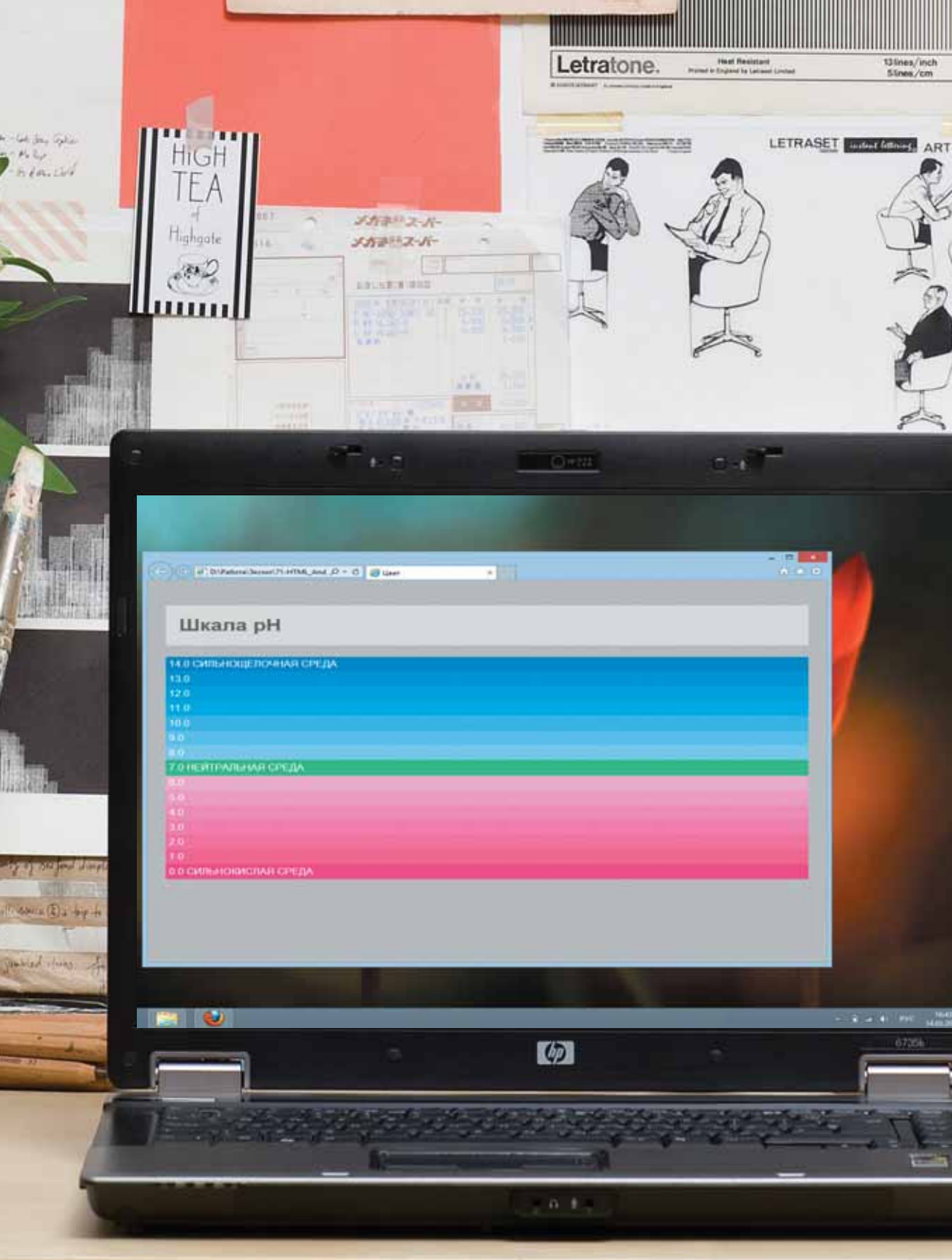
СВЕТЛОТА

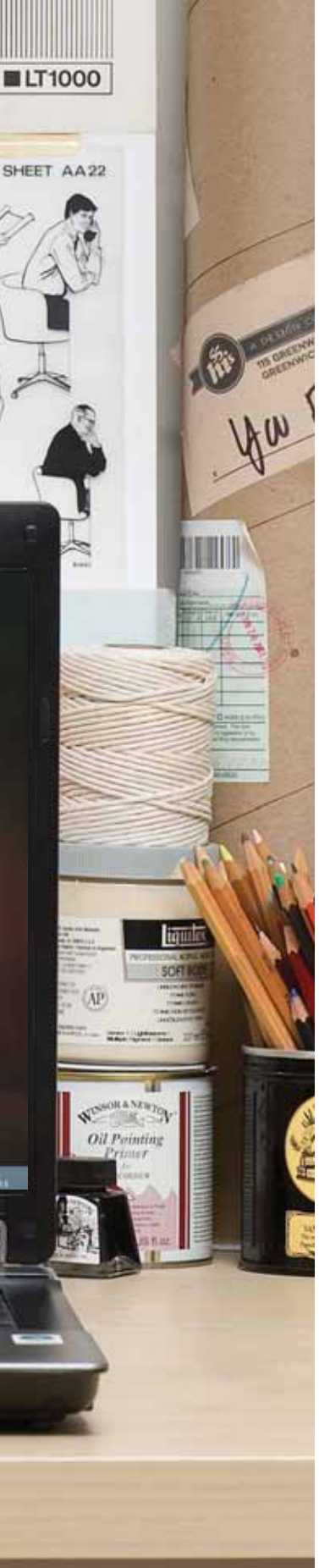
Принимает процентные значения: 0% — белый, 50% — нормальный цвет, 100% — черный.

Свойство `hsla` также позволяет выразить цвет через значения оттенка, насыщенности и светлоты, однако добавляет значение прозрачности (как свойство `rgba`). Буква `a` означает альфа-канал.

АЛЬФА

Принимает значения в диапазоне от 0 до 1,0. Например, 0,5 соответствует 50%-ной прозрачности, а 0,75 — 75%-ной.


Letratone.
Heat Resistant
13lines/inch
5lines/cm
LETRASET
HIGH TEA
of
Highgate
Шкала pH
14.0 СИЛЬНОЩЕЛОЧНАЯ СРЕДА
13.0
12.0
11.0
10.0
9.0
8.0
7.0 НЕЙТРАЛЬНАЯ СРЕДА
6.0
5.0
4.0
3.0
2.0
1.0
0.0 СИЛЬНОКИСЛАЯ СРЕДА



ПРИМЕР
ЦВЕТ

В этом примере шкала pH используется, чтобы продемонстрировать различные способы задания цвета с помощью CSS (с использованием имен цветов, шестнадцатеричных кодов, цветовых моделей RGB и HSL).

Правило для элемента `<body>` задает по умолчанию цвет для всего текста, а также фоновый. В данном правиле используются имена цветов.

Правило для элемента `<h1>` с помощью шестнадцатеричного кода задает цвет заголовка. Для свойства `background-color` элемента `<h1>` заданы два значения. Первое с помощью шестнадцатеричного кода предоставляет резервный вариант, а второе — значение HSLA для браузеров, которые поддерживают этот метод.

Каждый абзац выделен разными цветами, которые представляют различные уровни кислотности или щелочности, и эти цвета задаются с помощью значений RGB.

В данном примере также используется свойство `margin`, чтобы уменьшить интервал между блоками абзацев, и свойство `padding` для создания промежутка между краями блоков и текстом в них (подробнее об этих свойствах речь пойдет в главе 13).

ПРИМЕР
ЦВЕТ

```
<!DOCTYPE html>
<html>
  <head>
    <title>Цвет</title>
    <style type="text/css">
      body {
        background-color: silver;
        color: white;
        padding: 20px;
        font-family: Arial, Verdana, sans-serif;}
      h1 {
        background-color: #ffffff;
        background-color: hsla(0,100%,100%,0.5);
        color: #64645A;
        padding: inherit;}
      p {
        padding: 5px;
        margin: 0px;}
      p.zero {
        background-color: rgb(238,62,128);}
      p.one {
        background-color: rgb(244,90,139);}
      p.two {
        background-color: rgb(243,106,152);}
      p.three {
        background-color: rgb(244,123,166);}
      p.four {
        background-color: rgb(245,140,178);}
      p.five {
        background-color: rgb(246,159,192);}
      p.six {
        background-color: rgb(245,176,204);}
      p.seven {
        background-color: rgb(0,187,136);}
      p.eight {
        background-color: rgb(140,202,242);}
      p.nine {
        background-color: rgb(114,193,240);}
      p.ten {
```

ПРИМЕР
ЦВЕТ

```
        background-color: rgb(84,182,237);}
      p.eleven {
        background-color: rgb(48,170,233);}
      p.twelve {
        background-color: rgb(0,160,230);}
      p.thirteen {
        background-color: rgb(0,149,226);}
      p.fourteen {
        background-color: rgb(0,136,221);}
    </style>
  </head>
  <body>
    <h1>Шкала pH</h1>
    <p class="fourteen">14.0 СИЛЬНОЩЕЛОЧНАЯ СРЕДА</p>
    <p class="thirteen">13.0</p>
    <p class="twelve">12.0</p>
    <p class="eleven">11.0</p>
    <p class="ten">10.0</p>
    <p class="nine">9.0</p>
    <p class="eight">8.0</p>
    <p class="seven">7.0 НЕЙТРАЛЬНАЯ СРЕДА</p>
    <p class="six">6.0</p>
    <p class="five">5.0</p>
    <p class="four">4.0</p>
    <p class="three">3.0</p>
    <p class="two">2.0</p>
    <p class="one">1.0</p>
    <p class="zero">0.0 СИЛЬНОКИСЛАЯ СРЕДА</p>
  </body>
</html>
```

ОБЗОР
ЦВЕТ

- Цвет не просто оживляет сайт, но и позволяет создать настроение и вызвать определенную реакцию.
- Спецификация CSS предусматривает три способа задания цвета: через значения RGB, шестнадцатеричные коды и имена.
- Чтобы найти нужный цвет, вы можете воспользоваться цветовой палитрой.
- Важно обеспечить контраст между цветом текста и цветом фона, в противном случае пользователям будет сложно воспринимать ваш контент.
- В спецификации CSS3 для цветов RGB предусмотрено дополнительное значение, указывающее степень непрозрачности. Оно известно как RGBA.
- Спецификация CSS3 также позволяет указывать цвета с помощью значений HSL с дополнительным значением степени непрозрачности. Данный способ известен как HSLA.

12

ТЕКСТ

- Размер и гарнитура шрифта
- Полужирный, курсив, капитель, подчеркнутый
- Интервалы между строками, словами и буквами

Свойства, которые позволяют управлять внешним видом текста, можно разделить на две группы:

- те, которые непосредственно влияют на оформление, в том числе на гарнитуру, начертание и размер;
- те, которые одинаково влияют на весь текст, независимо от того, какой шрифт вы используете — сюда относятся цвет текста и расстояние между словами и буквами.

Форматирование текста может оказать существенное влияние на удобочитаемость страниц вашего сайта. По мере ознакомления с этими свойствами вам будут предоставлены некоторые советы по поводу представления текста.

Safari Файл Правка Вид История Закладки Разработка Окно Справка
Ср, 17:09
Текст
Бриары
СЕРДЦЕ, ОБЛАЧЕННОЕ В МЕХ
Бриар, или berger de brie, — это порода собак, традиционно использовавшихся в качестве пастухов и охранников для стада овец.
История породы
Порода Бриар, возникшая предположительно во Франции, выводилась веками для выпаса и охраны овец. Во французской армии собаки данной породы использовались в качестве часовых и посыльных. Они также помогали в поисках раненых солдат благодаря отличному слуху. Во время Первой мировой войны бриары оказались на грани вымирания. В настоящее время популяция бриаров понемногу восстанавливается. Владельцами бриаров были: Карл Великий, Наполеон, Томас Джефферсон и Лафайет.
Автор: Адам Даннтт

ТЕРМИНОЛОГИЯ ТИПОГРАФИКИ

ШРИФТ С ЗАСЕЧКАМИ*

Шрифты с засечками имеют дополнительные элементы по краям основных штрихов.

Шрифты с засечками традиционно используются для оформления больших объемов текста, поскольку считается, что они облегчают чтение.

* Serif (англ.). — *Примеч. ред.*

РУБЛЕНЫЙ ШРИФТ**

Рубленые шрифты (также называемые **гротесками**) характеризуются прямыми краями букв и, следовательно, имеют гораздо более «чистый» вид.

Разрешение экранов ниже, чем разрешение печати, поэтому при небольшом размере шрифта текста рубленые шрифты воспринимаются легче.

** Sans-Serif (англ.). — *Примеч. ред.*

МОНОШИРИННЫЙ ШРИФТ***

Все буквы моноширинного шрифта имеют одинаковую ширину (у букв в других шрифтах она разная).

Моноширинные шрифты, как правило, используются при написании кода, поскольку они легко выравниваются, что делает текст более удобным для восприятия.

*** Monospace (англ.). — *Примеч. ред.*

ВЕС

Тонкий;
средний;
полужирный;
жирный.

Вес шрифта не только добавляет акцент, но и влияет на пустое пространство и контраст на странице.

СТИЛЬ

Обычный;
курсивный;
наклонный.

Курсивный шрифт — это специально спроектированный шрифтовой набор, а наклонный — результат деформации обычного шрифта.

ПЛОТНОСТЬ

Сжатый;
обычный;
растянутый.

В сжатой версии шрифта буквы тоньше и расположены ближе друг к другу. В растянутой версии буквы толще и расположены дальше друг от друга.

ВЫБОР ГАРНИТУРЫ ШРИФТА ДЛЯ САЙТА

При выборе гарнитуры шрифта важно понимать то, что браузер отобразит его только в том случае, если данный шрифт установлен на компьютере пользователя.

ШРИФТ С ЗАСЕЧКАМИ

Шрифты с засечками имеют дополнительные элементы по краям основных штрихов.

ПРИМЕРЫ:

Georgia

Times

Times New Roman

РУБЛЕНЫЙ ШРИФТ

Рубленые шрифты (без засечек) характеризуются прямыми краями букв и, следовательно, имеют гораздо более чистый вид.

ПРИМЕРЫ:

Arial

Verdana

Helvetica

На сайтах часто используется небольшой набор гарнитур шрифтов, которые установлены на большинстве компьютеров (они перечислены выше). Существуют способы обойти это ограничение (речь о них пойдет далее в главе).

Можно указать несколько гарнитур шрифтов и перечислить их в порядке предпочтения (на случай, если у пользователя не установлена гарнитура, указанная первой). Этот прием иногда называют *последовательностью шрифтов*.

МОНОШИРИННЫЙ ШРИФТ

Все буквы моноширинного шрифта имеют одинаковую ширину (у буквы в других шрифтах она разная).

ПРИМЕРЫ:

Courier

Courier New

КУРСИВНЫЙ ШРИФТ

Курсивные шрифты имеют либо соединительные штрихи, либо другие характеристики, присущие, например, рукописным буквам.

ПРИМЕРЫ:

Comic Sans MS

Monotype Corsiva

ДЕКОРАТИВНЫЙ ШРИФТ

К декоративным обычно относятся шрифты, часто используемые для оформления заголовков. Они не предназначены для больших объемов текста.

ПРИМЕРЫ:

Impact

Haettenschweiler

Браузеры должны поддерживать по крайней мере по одной гарнитуре из перечисленных выше групп. По этой причине после указания выбранных гарнитур шрифтов следует добавлять общее название типа гарнитуры.

Например если вы выбрали шрифт с засечками, то можете написать так:

```
font-family: Georgia, Times, serif;
```

СПОСОБЫ ИСПОЛЬЗОВАНИЯ ДОПОЛНИТЕЛЬНЫХ ГАРНИТУР ШРИФТОВ

Существуют способы использования гарнитур шрифтов, отличных от перечисленных ранее. Тем не менее шрифты являются объектом авторского права, поэтому ваш выбор ограничивается соответствующими лицензиями.

FONT-FAMILY

Гарнитура шрифта должна быть установлена на компьютере пользователя. Таблица CSS используется для указания гарнитуры.

ОБСУЖДАЕТСЯ

См. разделы «Указание гарнитуры шрифта» и «Размер шрифта» далее в главе.

НЕДОСТАТКИ

У большинства пользователей установлен ограниченный набор гарнитур шрифтов.

ЛИЦЕНЗИРОВАНИЕ

Вы не занимаетесь распространением гарнитуры шрифта, так что вопроса о лицензировании не возникает.

ВЫБОР ГАРНИТУР

Выбор гарнитур ограничен, поскольку шрифт должен быть установлен на компьютерах пользователей.

FONT-FACE

В коде CSS указано, откуда можно загрузить шрифт в случае, если он не установлен на компьютере.

ОБСУЖДАЕТСЯ

См. разделы «Дополнительные шрифты» и «Форматы шрифтов» далее в главе.

НЕДОСТАТКИ

Пользователю необходимо загрузить файл шрифта, что может замедлить процесс отображения веб-страницы.

ЛИЦЕНЗИРОВАНИЕ

Лицензия на использование шрифта должна разрешать его распространение с помощью свойства `font-face`.

ВЫБОР ГАРНИТУР

Выбор ограничен, поскольку лишь некоторые шрифты могут распространяться таким способом.

FONT-FACE ПО ПОДПИСКЕ

Коммерческие сервисы предоставляют пользователям доступ к широкому спектру шрифтов с использованием свойства `font-face`.

ОБСУЖДАЕТСЯ

См. разделы «Дополнительные шрифты» и «Форматы шрифтов» далее в главе.

НЕДОСТАТКИ

Данный способ предусматривает регулярную плату за лицензии, которая направляется компании, разработавшей шрифт.

ЛИЦЕНЗИРОВАНИЕ

Сервис самостоятельно решает вопросы лицензирования с людьми, создавшими конкретный шрифт.

ВЫБОР ГАРНИТУР

Каждый сервис предлагает различные наборы шрифтов в зависимости от договоренностей с компаниями-разработчиками.

ДАННЫЕ СПОСОБЫ ПОДХОДЯТ ДЛЯ ТЕКСТА ЛЮБОГО ОБЪЕМА.

Если вы работаете на компьютере Mac, важно проверить, как выбранная гарнитура шрифта выглядят в операционной системе Windows, поскольку шрифты в ней отображаются не так четко. Однако если вы работаете компьютере под управлением Windows, то мы вас уверяем — в OS X текст будет смотреться прекрасно.

ИЗОБРАЖЕНИЯ

Вы можете создать графическое изображение, которое будет содержать текст, отформатированный выбранной гарнитурой шрифта.

ОБСУЖДАЕТСЯ

См. главу 5.

НЕДОСТАТКИ

Людям, использующим программы экранного доступа, потребуется замещающий текст, чтобы понять, о чем идет речь.

ЛИЦЕНЗИРОВАНИЕ

Вы можете использовать любую гарнитуру, поскольку в данном случае вы ее не распространяете.

ВЫБОР ГАРНИТУР

Очень широкий выбор, поскольку вы можете использовать любую гарнитуру из числа тех, на которые вы имеете лицензии.

ТЕХНОЛОГИЯ SIFR

Шрифт внедряется во Flash-файл, а код JavaScript замещает указанный HTML-текст Flash-версией.

ОБСУЖДАЕТСЯ

Для получения более подробной информации см. страницу **www.htmlandcssbook.com/extras/sifr-and-cufon/** (на английском языке).

НЕДОСТАТКИ

Данный метод работает только в том случае, если в устройстве пользователя активирована технология Flash и JavaScript.

ЛИЦЕНЗИРОВАНИЕ

Многие коммерческие разработчики шрифтов разрешают использование данного метода, хотя вам может потребоваться заплатить за дополнительную лицензию на использование шрифта во Всемирной паутине.

ВЫБОР ГАРНИТУР

Данный метод предоставляет большой выбор гарнитур, поскольку основные фирмы-разработчики шрифтов предусматривают такое использование.

ТЕХНОЛОГИЯ CUFON

Технология Cufon предоставляет функции, похожие на sIFR. Она использует JavaScript для создания SVG- или VML-версии текста.

ОБСУЖДАЕТСЯ

Для получения более подробной информации см. страницу **www.htmlandcssbook.com/extras/sifr-and-cufon/** (на английском языке).

НЕДОСТАТКИ

Требуется установка JavaScript. Пользователь не сможет выделить текст, и текст не будет меняться при наведении на него указателя мыши.

ЛИЦЕНЗИРОВАНИЕ

Также как и в случае с технологией sIFR, некоторые разработчики позволяют использовать свои шрифты с применением технологии Cufon, однако вам следует свериться с лицензионным соглашением.

ВЫБОР ГАРНИТУР

Данный метод предусматривает несколько меньший выбор гарнитур шрифтов по сравнению с технологией sIFR, поскольку некоторые фирмы-разработчики не очень жалуют эту технологию.

ДАННЫЕ СПОСОБЫ НЕ ПОДХОДЯТ ДЛЯ БОЛЬШИХ ОБЪЕМОВ ТЕКСТА.

УКАЗАНИЕ ГАРНИТУРЫ ШРИФТА

font-family

Свойство `font-family` позволяет указать гарнитуру шрифта для текста внутри элементов, к которым применяется правило CSS.

Значением этого свойства является имя гарнитуры, которую вы хотите использовать.

Чтобы ваш сайт корректно отображался, на компьютерах посетителей должна быть установлена указанная гарнитура шрифта.

Вы можете указать список шрифтов, разделенных запятыми, чтобы в случае отсутствия первой гарнитуры браузер использовал бы альтернативный шрифт из списка.

Кроме того, частой практикой является указание в конце списка общего названия типа шрифтов (с этим методом вы познакомились ранее в главе).

Если имя шрифта состоит более чем из одного слова, то его нужно указывать в двойных кавычках.

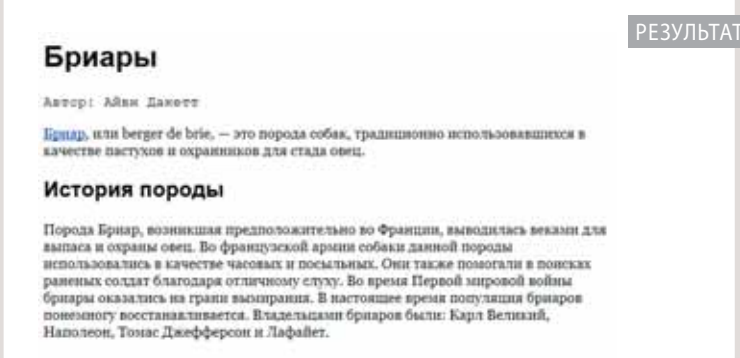
Дизайнеры считают, что страницы обычно выглядят лучше, если на них используется не более трех шрифтов.

Для всех примеров в этой главе мы будем использовать расширенный документ HTML, показанный далее.

глава-12/font-family.html HTML + CSS

```
<!DOCTYPE html>
<html>
  <head>
    <title>Семейство шрифтов</title>
    <style type="text/css">
      body {
        font-family: Georgia, Times, serif;}
      h1, h2 {
        font-family: Arial, Verdana, sans-serif;}
      .credits {
        font-family: "Courier New", Courier,
          monospace;}
    </style>
  </head>
  <body>
    <h1>Бриары</h1>
    <p class="credits">Автор: Айви Дакетт</p>
    <p class="intro"><a class="breed" href="http://
      ru.wikipedia.org/wiki/Бриар_(порода_собак)">
      Бриар</a>, или berger de brie, — это порода
      собак, традиционно использовавшихся в качестве
      пастухов и охранников для стада овец...</p>
  </body>
</html>
```

РЕЗУЛЬТАТ

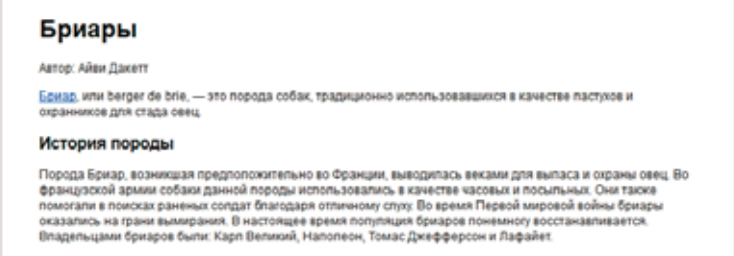


РАЗМЕР ШРИФТА
font-size

CSS **глава-12/font-size.html**

```
body {
  font-family: Arial, Verdana, sans-serif;
  font-size: 12px;}
h1 {
  font-size: 200%;}
h2 {
   font-size: 1.3em;}
```

РЕЗУЛЬТАТ



Бриары

Автор: Айви Дакетт

Бриар, или berger de brie, — это порода собак, традиционно использовавшихся в качестве пастухов и охранников для стада овец.

История породы

Порода Бриар, возникшая предположительно во Франции, выводилась веками для выпаса и охраны овец. Во французской армии собаки данной породы использовались в качестве часовых и посыльных. Они также помогали в поисках раненых солдат благодаря отличному слуху. Во время Первой мировой войны бриары оказались на грани вымирания. В настоящее время популяция бриаров понемногу восстанавливается. Владельцами бриаров были: Карл Великий, Наполеон, Томас Джефферсон и Лафайет.

Свойство `font-size` позволяет указать размер шрифта. Существует несколько способов сделать это. Наиболее распространенные следующие.

ПИКСЕЛЫ

Значения в пикселах используются часто, поскольку такой подход предоставляет веб-дизайнерам очень точный контроль над тем, сколько места будет занимать их текст. Числовое значение сопровождается буквами `px`.

ПРОЦЕНТЫ

По умолчанию размер текста в браузерах равен 16px. Таким образом, значение 75% соответствует 12px, а 200% — 32px.
Если вы создадите правило, задающее всему тексту внутри элемента `<body>` значение 75% от размера по умолчанию (что соответствует 12px), а затем создадите еще одно правило, которое задает значение 75% для содержимого элемента, вложенного в `<body>`, то размер шрифта в этом элементе будет равен 9px (75% от 12px).

ЕДИНИЦЫ EM

Em — это единица измерения, эквивалентная ширине буквы m.

Далее мы подробно рассмотрим каждый из перечисленных способов.

ТИПОГРАФСКАЯ ШКАЛА

Вы, вероятно, заметили, что такие программы, как Word, Photoshop и InDesign, предусматривают одни и те же размеры шрифтов.

Дело в том, что они созданы в соответствии со шкалой размеров шрифта, которая была разработана европейскими типографами в XVI веке.

Считается, что данная шкала размеров оптимальна для восприятия, и поэтому она практически не изменилась за последние 400 лет.

Вот почему использование размеров шрифта из этой шкалы при верстке сайта сделает его более привлекательным.

На следующей странице вы увидите, как данные размеры шрифта выражаются в пикселах, процентах и единицах em.

Дизайнеры, работающие в области полиграфии, часто выражают размер текста в пунктах, а не в пикселах (отсюда использование единиц измерения pt в шкале справа). Пиксел примерно равен одному пункту, поскольку пункт соответствует 1/72 дюйма, а большинство компьютерных дисплеев имеют разрешение 72 точки на дюйм.

По умолчанию размер текста в браузере равен 16 пикселам. Поэтому если вы используете значение в процентах или единицах em, то вы рассчитываете нужный размер текста на основе размера, который используется в браузерах по умолчанию. Например, вы можете уменьшить основной текст до 12 пикселов и увеличить заголовки до 24 пикселов.

В последнее время некоторые веб-дизайнеры начали оставлять размер по умолчанию (16 пикселов) для основного текста и настраивать остальные шрифты, используя относительные пропорции данной шкалы.

Когда вы впервые видите основной текст размером в 16 пикселов, он может показаться довольно крупным. Однако по прошествии некоторого времени, когда вы привыкнете, то, скорее всего, найдете такой текст более легким для восприятия. После этого при возвращении на страницу, где размер основного текста соответствует 12 пикселам, шрифт на ней покажется вам слишком маленьким.

8pt
9pt
10pt
11pt
12pt
14pt
18pt
24pt
36pt
48pt
60pt
72pt

ЕДИНИЦЫ ИЗМЕРЕНИЯ РАЗМЕРА ШРИФТА

ПИКСЕЛЫ

12-ПИКСЕЛЬНАЯ ШКАЛА

```
h1      24px
h2      18px
h3      14px
body    12px
```

=

16-ПИКСЕЛЬНАЯ ШКАЛА

```
h1      32px
h2      24px
h3      18px
body    16px
```

Указание размера шрифта в пикселах — это наилучший способ гарантировать, что шрифт будет отображаться в нужном размере, поскольку значения в процентах и единицах em могут варьировать, если пользователь изменит используемый в браузере размер текста по умолчанию. Пикселы соотносятся с разрешением экрана, поэтому один и тот же шрифт будет казаться крупнее при разрешении экрана 800 × 600, чем при разрешении 1280 × 800.
Вы также можете указать размер не в пикселах (`px`), а в пунктах (`pt`), однако это следует делать только при создании таблицы стилей для печатной версии страницы.

ПРОЦЕНТЫ

12-ПИКСЕЛЬНАЯ ШКАЛА

```
h1      200%
h2      150%
h3      117%
body     75%
```

=

16-ПИКСЕЛЬНАЯ ШКАЛА

```
h1      200%
h2      150%
h3      133%
body    100%
```

В веб-браузере размер текста по умолчанию равен 16 пикселам. Используя процентные величины, вы можете создать шкалу с размером текста по умолчанию в 12 пикселов и размерами заголовков, масштабирующимися относительно данного значения. Пользователи могут изменить размер текста, используемый по умолчанию, в своем веб-браузере. При этом шрифты будут отображаться в масштабе, задуманном дизайнером, но их размер окажется иным.

ЕДИНИЦЫ EM

12-ПИКСЕЛЬНАЯ ШКАЛА

```
h1      1.5em
h2      1.3em
h3     1.17em
body    100%
p      0.75em
```

=

16-ПИКСЕЛЬНАЯ ШКАЛА

```
h1        2em
h2      1.5em
h3    1.125em
body    100%
p         1em
```

Единицы em позволяют изменить размер текста относительно размера текста в родительском элементе. Поскольку по умолчанию размер текста в веб-браузерах составляет 16 пикселов, вы можете использовать правила, аналогичные тем, которые используются для значений в процентах. Поскольку пользователи способны изменять размер текста по умолчанию в своем браузере, шрифты могут казаться больше (или меньше). Дополнительное правило `p`, указанное выше, позволяет браузеру Internet Explorer версии 6 и 7 отображать шрифты в нужном размере. Без него программы Internet Explorer 6 и Internet Explorer 7 будут чрезмерно увеличивать относительные размеры остального текста.

ДОПОЛНИТЕЛЬНЫЕ ШРИФТЫ

@font-face

Свойство `@font-face` позволяет использовать шрифт, даже если он не установлен на компьютере пользователя, предоставляя путь к файлу шрифта, который в этом случае будет загружен.
Поскольку данный метод позволяет загружать файл шрифта на компьютер пользователя, очень важно, чтобы соответствующая лицензия предусматривала подобное использование.
Шрифт добавляется в таблицу стилей с помощью правила `@font-face`, как показано в следующем примере.

глава-12/font-face.html CSS

```
@font-face {
  font-family: 'ChunkFiveRegular';
  src: url('fonts/chunkfive.eot');}
h1, h2 {
  font-family: ChunkFiveRegular, Georgia, serif;}
```

РЕЗУЛЬТАТ

font-family

Данное свойство позволяет указать имя шрифта. Затем оно быть использовано в качестве значения свойства `font-family` в остальной части таблицы стилей (как показано в правиле для элементов `<h1>` и `<h2>`).

src

Данный атрибут определяет путь к файлу шрифта. Чтобы прием работал во всех браузерах, вам, вероятно, необходимо будет указать пути к нескольким различным версиям шрифта, как показано в следующем примере.

format

Определяет формат, в котором поставляется шрифт (далее обсуждается подробнее).

Многие разработчики не позволяют использовать свои шрифты таким способом, однако существуют шрифты, которые допускается использовать свободно.
Вы можете найти их на следующих сайтах:

www.fontsquirrel.com;
www.fontex.org;
www.openfontlibrary.org.

При использовании шрифтов с этих сайтов также важно проверить лицензионные соглашения, поскольку некоторые шрифты бесплатны только для личного пользования (то есть они не предусматривают использования на коммерческих сайтах).

Существуют сайты, которые предоставляют возможность использования коммерческих шрифтов, поскольку они заключили договор, позволяющий их посетителям использовать эти шрифты на платной основе:

www.typekit.com;
www.kernest.com;
www.fontspring.com.

Сервис Google также предоставляет бесплатные шрифты. Вместо добавления правила `@font-face` в таблицу стилей вам нужно будет вставить ссылку на файл CSS и на файлы шрифтов на сервере сервиса **www.google.com/webfonts**.

ФОРМАТЫ ШРИФТОВ

CSS **глава-12/understanding-font-formats.html**

```
@font-face {
  font-family: 'ChunkFiveRegular';
  src: url('fonts/chunkfive.eot');
  src: url('fonts/chunkfive.eot?#iefix')
    format('embedded-opentype'),
    url('fonts/chunkfive.woff') format('woff'),
    url('fonts/chunkfive.ttf')
      format('truetype'),
    url('fonts/chunkfive.svg#ChunkFiveRegular')
    format('svg');}
```

БРАУЗЕР	ФОРМАТ			
	eot	woff	ttf / otf	svg
Chrome (все версии)				●
Chrome 6 и выше		●	●	●
Firefox 3.5			●	
Firefox 3.6 и выше		●	●	
Internet Explorer 5–8	●			
Internet Explorer 9 и выше	●	●	◖	
Opera 10 и выше			●	●
Safari 3.1 и выше			●	●
iOS ранее 4.2				●
iOS 4.2 и выше			●	●

Поскольку для отображения шрифта браузер должен загрузить соответствующий файл, пользователи могут сперва увидеть неотформатированный текст. Есть два способа, которые позволяют свести к минимуму подобное. Вам нужно удалить все ненужные глифы из шрифта и/или разместить файл шрифта, используя технологию CDN*, которая обеспечивает более быструю передачу файлов.

* Content delivery network (*англ.*) — Сеть доставки (и дистрибуции) контента. — *Примеч. ред.*

Разные браузеры поддерживают различные форматы шрифтов (так же, как они поддерживают и различные аудио- и видеоформаты), поэтому вам может потребоваться предоставить шрифт в нескольких форматах, чтобы обеспечить его работу во всех браузерах.

Если у вас нет всех нужных форматов для вашего шрифта, вы можете загрузить его на сайт FontSquirrel, который поможет вам его конвертировать: **www.fontsquirrel.com/fontface/generator**. Сервис Font Squirrel также предоставляет CSS-код для правила `@font-face`. Это очень удобно, поскольку при использовании нескольких форматов шрифта свойства `src` и `format` правила `@font-face` могут сильно усложниться. Слева показан пример более сложного правила `@font-face`.

Форматы шрифтов должны быть перечислены в коде в следующем порядке:

1: `eot`

2: `woff`

3: `ttf/otf`

4: `svg`

ПОЛУЖИРНОЕ НАЧЕРТАНИЕ
font-weight

Свойство `font-weight` позволяет форматировать шрифт текста полужирным начертанием. Обычно данное свойство имеет два значения.

normal

Использование данного значения приводит к тому, что текст имеет обычный вес.

bold

Это значение соответствует полужирному начертанию.
В данном примере показано, что элемент, чей атрибут `class` имеет значение `credits`, выделен шрифтом с полужирным начертанием.

Вам, вероятно, интересно, зачем предусматривается нормальный вес шрифта. Если, например, вы создали правило для элемента `<body>`, указывающее, что весь текст в теле документа должен быть выделен полужирным, вам может потребоваться выделить текст в некоторых областях обычным шрифтом. Данный метод в основном используется в качестве «переключателя».

глава-12/font-weight.html CSS

```
credits {
font-weight: bold;}
```

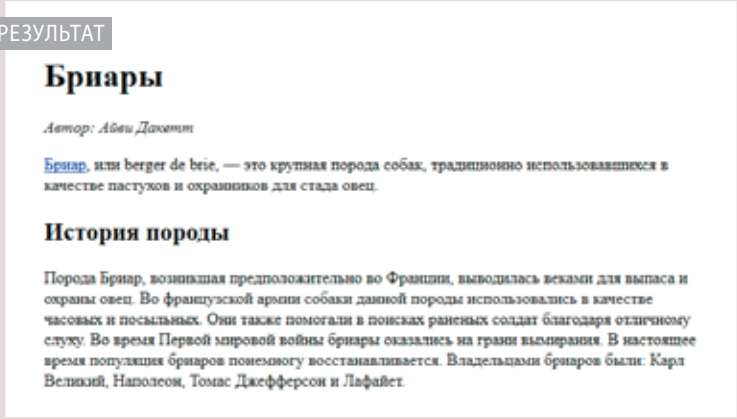
РЕЗУЛЬТАТ

КУРСИВ
font-style

CSS **глава-12/font-style.html**

```
credits {
  font-style: italic;}
```

РЕЗУЛЬТАТ



Бриары

Автор: Абби Дакетт

Бриар, или berger de brie, — это крупная порода собак, традиционно использовавшихся в качестве пастухов и охранников для стада овец.

История породы

Порода Бриар, возникшая предположительно во Франции, выводилась веками для выпаса и охраны овец. Во французской армии собаки данной породы использовались в качестве часовых и посыльных. Они также помогали в поисках раненых солдат благодаря отличному слуху. Во время Первой мировой войны бриары оказались на грани вымирания. В настоящее время популяция бриаров понемногу восстанавливается. Владельцами бриаров были: Карл Великий, Наполеон, Томас Джефферсон и Лафайет.

Если вы хотите выделить текст курсивом, то можете использовать свойство `font-style`. Данное свойство способно принимать три значения.

normal

Использование данного значения приводит к тому, что текст имеет обычное начертание, то есть не курсивное и не наклонное.

italic

При использовании данного значения текст выделяется курсивом.

oblique

Это значение соответствует наклонному начертанию. В примере вы можете видеть, что имя автора выделено курсивом.

Курсивные шрифты традиционно были стилизованными версиями шрифта, основанными на каллиграфии, в то время как наклонное начертание создавалось простым искажением обычного шрифта.

Браузеру часто не удается найти курсивную версию шрифта, и зачастую программа использует алгоритм наклона его нормальной версии. Это означает, что большая часть курсивного текста во Всемирной паутине на самом деле выделена наклонным начертанием.

ПРОПИСНЫЕ И СТРОЧНЫЕ СИМВОЛЫ

text-transform

Свойство `text-transform` используется для изменения регистра текста и принимает одно из следующих значений.

uppercase

При данном значении для всех символов в тексте используется верхний регистр.

lowercase

При данном значении для всех символов в тексте используется нижний регистр.

capitalize

При данном значении каждое слово начинается с прописной буквы.

В примере для символов элемента `<h1>` используется верхний регистр, для символов элемента `<h2>` — нижний регистр, а имя автора начинается с прописных букв. Если вы используете верхний регистр, то стоит рассмотреть возможность применения свойства `letter-spacing` для увеличения интервала между буквами, как показано в разделе «Межбуквенный интервал и интервал между словами» далее в главе. Это поможет повысить удобочитаемость текста.

глава-12/text-transform.html CSS

```
h1 {
  text-transform: uppercase;}
h2 {
  text-transform: lowercase;}
.credits {
  text-transform: capitalize;}
```

РЕЗУЛЬТАТ

БРИАРЫ

Автор: Айви Даветт

Бриар, или berger de brie, — это крупная порода собак, традиционно использовавшихся в качестве пастухов и охранников для стада овец.

история породы

Порода Бриар, возникшая предположительно во Франции, выводилась вначале для выпаса и охраны овец. Во французской армии собаки данной породы использовались в качестве часовых и посыльных. Они также помогали в поисках раненых солдат благодаря отличному слуху. Во время Первой мировой войны бриары оказались на грани вымирания. В настоящее время популяция бриаров понемногу восстанавливается. Владельцами бриаров были: Карл Великий, Наполеон, Томас Джефферсон и Лафайет.

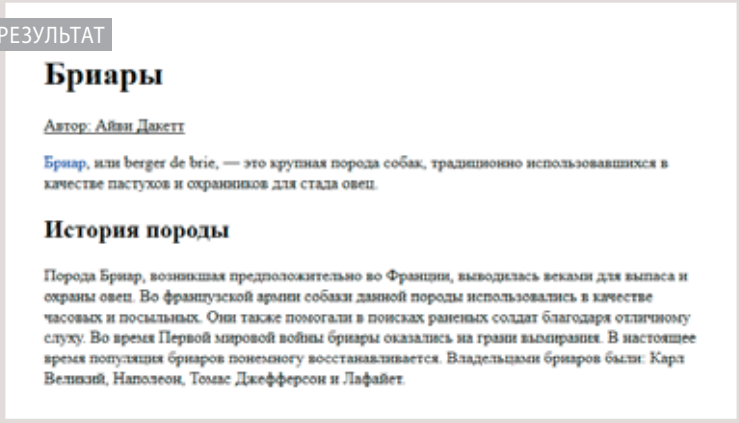
ПОДЧЕРКИВАНИЕ, ПЕРЕЧЕРКИВАНИЕ И ДРУГОЕ ФОРМАТИРОВАНИЕ

text-decoration

CSS **глава-12/text-decoration.html**

```
.credits {
  text-decoration: underline;}
a {
  text-decoration: none;}
```

РЕЗУЛЬТАТ



Бриары

Автор: Айви Дакетт

Бриар, или berger de brie, — это крупная порода собак, традиционно использовавшихся в качестве пастухов и охранников для стада овец.

История породы

Порода Бриар, возникшая предположительно во Франции, выводилась веками для выпаса и охраны овец. Во французской армии собаки данной породы использовались в качестве часовых и посыльных. Они также помогали в поисках раненых солдат благодаря отличному слуху. Во время Первой мировой войны бриары оказались на грани вымирания. В настоящее время популяция бриаров понемногу восстанавливается. Владельцами бриаров были: Карл Великий, Наполеон, Томас Джефферсон и Лафайет.

Свойство `text-decoration` принимает следующие значения.

none

Данное значение сбрасывает любое примененное к тексту оформление.

underline

Данное значение добавляет линию под текстом.

overline

Данное значение добавляет линию над текстом.

line-through

Данное значение добавляет линию, проходящую через слова.

blink

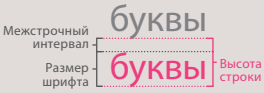
Данное значение заставляет текст мигать. Обычно эта практика не одобряется, поскольку такое поведение текста считается весьма раздражающим.

В примере имя автора подчеркнуто, а название породы — нет, хотя по умолчанию оно было бы подчеркнуто, поскольку является ссылкой.

Данное свойство обычно используется дизайнерами для удаления подчеркивания, которым браузеры выделяют ссылки. В разделах «Применение стилей к ссылкам» и «Реагирование на действия пользователя» далее в главе будет показано, как добавлять или удалять подчеркивание при наведении указателя мыши на ссылку.

МЕЖСТРОЧНЫЙ ИНТЕРВАЛ
line-height

Межстрочный интервал — это типографский термин, обозначающий расстояние между строками текста. Часть символа, которая располагается ниже базовой линии, называется **нижним выносным элементом**, в то время как самая высокая точка символа называется **верхним выносным элементом**. Межстрочный интервал измеряется от нижней части нижнего выносного элемента на одной строке до верхней части верхнего выносного элемента на следующей строке.



В CSS свойство `line-height` устанавливает высоту всей строки текста, поэтому разница между размером шрифта (`font-size`) и высотой строки (`line-height`) соответствует межстрочному интервалу (как показано на схеме выше). Увеличение значения свойства `line-height` приводит к увеличению интервала между строками текста.

глава-12/line-height.html CSS

```
p {
  line-height: 1.4em;}
```

РЕЗУЛЬТАТ

Бриары

Автор: Айви Дакетт

Бриар, или berger de brie, — это крупная порода собак, традиционно использовавшихся в качестве пастухов и охранников для стада овец.

История породы

Порода Бриар, возникшая предположительно во Франции, выводилась веками для выпаса и охраны овец. Во французской армии собаки данной породы использовались в качестве часовых и посыльных. Они также помогали в поисках раненых солдат благодаря отличному слуху. Во время Первой мировой войны бриары оказались на грани вымирания. В настоящее время популяция бриаров понемногу восстанавливается. Владельцами бриаров были: Карл Великий, Наполеон, Томас Джефферсон и Лафайет.

РЕЗУЛЬТАТ БЕЗ CSS

Бриары

Автор: Айви Дакетт

Бриар, или berger de brie, — это крупная порода собак, традиционно использовавшихся в качестве пастухов и охранников для стада овец.

История породы

Порода Бриар, возникшая предположительно во Франции, выводилась веками для выпаса и охраны овец. Во французской армии собаки данной породы использовались в качестве часовых и посыльных. Они также помогали в поисках раненых солдат благодаря отличному слуху. Во время Первой мировой войны бриары оказались на грани вымирания. В настоящее время популяция бриаров понемногу восстанавливается. Владельцами бриаров были: Карл Великий, Наполеон, Томас Джефферсон и Лафайет.

Увеличение межстрочного интервала позволяет повысить удобочитаемость текста. Расстояние между строками должно быть больше, чем расстояние между словами, поскольку это помогает глазу следовать слева направо, а не сверху вниз. Хорошим начальным значением является 1.4–1.5 em. Поскольку пользователи могут изменить размер шрифта по умолчанию в своем браузере, значение свойства `line-height` лучше указывать в единицах em, а не в пикселах, чтобы межстрочный интервал соотносился с размером текста, выбранным пользователем.

МЕЖБУКВЕННЫЙ ИНТЕРВАЛ И ИНТЕРВАЛ МЕЖДУ СЛОВАМИ

letter-spacing, word-spacing

CSS **глава-12/letter-and-word-spacing.html**

```
h1, h2 {
  text-transform: uppercase;
  letter-spacing: 0.2em;}
.credits {
  font-weight: bold;
  word-spacing: 1em;}
```

РЕЗУЛЬТАТ

БРИАРЫ

Автор: Айви Дакетт

Бриар, или berger de brie, — это крупная порода собак, традиционно использовавшихся в качестве пастухов и охранников для стада овец.

ИСТОРИЯ ПОРОДЫ

Порода Бриар, возникшая предположительно во Франции, выводилась веками для выпаса и охраны овец. Во французской армии собаки данной породы использовались в качестве часовых и посыльных. Они также помогали в поисках раненых солдат благодаря отличному слуху. Во время Первой мировой войны бриары оказались на грани вымирания. В настоящее время популяция бриаров понемногу восстанавливается. Владельцами бриаров были: Карл Великий, Наполеон, Томас Джефферсон и Лафайет.

РЕЗУЛЬТАТ БЕЗ CSS

Бриары

Автор: Айви Дакетт

Бриар, или berger de brie, — это крупная порода собак, традиционно использовавшихся в качестве пастухов и охранников для стада овец.

История породы

Порода Бриар, возникшая предположительно во Франции, выводилась веками для выпаса и охраны овец. Во французской армии собаки данной породы использовались в качестве часовых и посыльных. Они также помогали в поисках раненых солдат благодаря отличному слуху. Во время Первой мировой войны бриары оказались на грани вымирания. В настоящее время популяция бриаров понемногу восстанавливается. Владельцами бриаров были: Карл Великий, Наполеон, Томас Джефферсон и Лафайет.

Кернинг — это термин, используемый в типографии для обозначения интервала между буквами. Вы можете управлять им с помощью свойства `letter-spacing`.

Особенно полезно увеличивать кернинг, когда в заголовке или в предложении используются только прописные символы. Если ваш текст отформатирован обычным способом, то увеличение или уменьшение кернинга может усложнить восприятие.

Контролировать интервал между словами также позволяет свойство `word-spacing`.

При указании значений для этих свойств используйте единицы em, и тогда они будут прибавляться к значению по умолчанию для указанного шрифта.

По умолчанию интервал между словами устанавливается самим шрифтом (и соответствует примерно 0.25 em), и вам вряд ли потребуется часто менять значение этого свойства. Если шрифт выделен полужирным начертанием или вы увеличили межбуквенный интервал, то увеличение интервала между словами может повысить удобочитаемость текста.

ВЫРАВНИВАНИЕ
text-align

Свойство `text-align` позволяет контролировать выравнивание текста. Оно может принимать одно из четырех следующих значений.

left
Определяет, что текст должен быть выровнен по левому краю.

right
Определяет, что текст должен быть выровнен по правому краю.

center
Данное значение позволяет выровнять текст по центру.

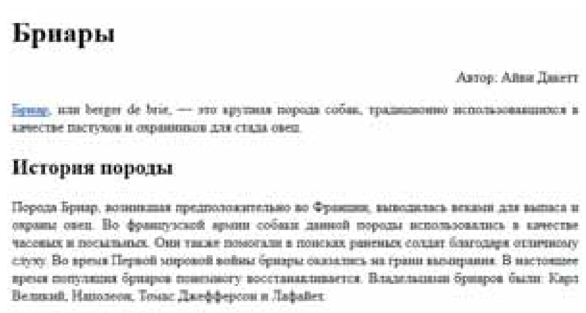
justify
Определяет, что каждая строка абзаца, за исключением последней, должна занимать всю ширину текстового поля.

При наличии нескольких абзацев текста его следует выравнивать по левому краю для упрощения восприятия. При выравнивании текста по ширине между словами в строке вставляются равные интервалы. Широкие пробелы между словами на одной строке и узкие на другой могут выглядеть странно. Так происходит, когда строки недостаточно широки или когда в тексте содержатся длинные слова.

глава-12/text-align.html CSS

```css
h1 {
  text-align: left;}
p {
  text-align: justify;}
.credits {
  text-align: right;}
```

РЕЗУЛЬТАТ


Бриары

Автор: Айви Дакетт

Бриар, или berger de brie, — это крупная порода собак, традиционно использовавшихся в качестве пастухов и охранников для стада овец.

История породы

Порода Бриар, возникшая предположительно во Франции, выводилась веками для выпаса и охраны овец. Во французской армии собаки данной породы использовались в качестве часовых и посыльных. Они также помогали в поисках раненых солдат благодаря отличному слуху. Во время Первой мировой войны бриары оказались на грани вымирания. В настоящее время популяция бриаров понемногу восстанавливается. Владельцами бриаров были: Карл Великий, Наполеон, Томас Джефферсон и Лафайет.

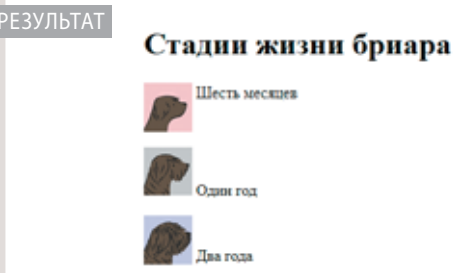
ВЕРТИКАЛЬНОЕ ВЫРАВНИВАНИЕ

vertical-align

CSS **глава-12/vertical-align.html**

```
#six-months {
  vertical-align: text-top;}
#one-year {
  vertical-align: baseline;}
#two-years {
  vertical-align: text-bottom;}
```

РЕЗУЛЬТАТ



Свойство `vertical-align` часто вводит в заблуждение. Оно *не предназначено* для вертикального выравнивания текста в блочных элементах, таких как `<p>` и `<div>`, хотя и производит этот эффект при использовании с ячейками таблицы (элементами `<td>` и `<th>`). Чаще данное свойство используется со встроенными элементами, такими как `<img>`, `<em>` или `<strong>` — в этом случае оно выполняет задачи, очень похожие на задачи HTML-атрибута `align`, применяемого к элементу `<img>`, о котором вы узнали в главе 5. Данное свойство может принимать следующие значения.

`baseline`;
`sub`;
`super`;
`top`;
`text-top`;
`middle`;
`bottom`;
`text-bottom`.

Это свойство также может принимать **значение длины** (обычно указываемой в пикселах или единицах em) или **процентное значение** высоты строки.

ТЕКСТ С ОТСТУПОМ
text-indent

Свойство `text-indent` позволяет создать отступ для первой строки текста внутри элемента. Значение отступа можно выразить несколькими способами, но, как правило, оно указывается в пикселах или единицах em. Данное свойство способно принимать и отрицательные значения, что означает возможность размещения текста за пределами окна браузера. Эта техника используется в приведенном далее примере, где в качестве заголовка в `<h1>` используется фоновое изображение. Текст сдвинут влево за пределы экрана (о фоновых изображениях речь пойдет в главе 16). Нам нужно, чтобы на странице присутствовал текст заголовка (для поисковых систем и пользователей, которые не способны увидеть изображение), но мы не можем отобразить его поверх логотипа, поскольку в этом случае он окажется нечитаемым. Если сдвинуть заголовок на 9999 пикселов влево, то он исчезнет из виду, но по-прежнему будет присутствовать в HTML-коде. Второе правило в этом примере создает отступ для строки с именем автора, сдвигая ее на 20 пикселов вправо.

глава-12/text-indent.html CSS

```
h1 {
  background-image: url("images/logo.gif");
  background-repeat: no-repeat;
  text-indent: -9999px;}
.credits {
  text-indent: 20px;}
```

РЕЗУЛЬТАТ

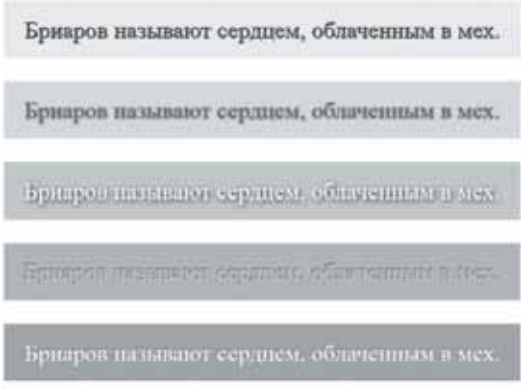
CSS3: ТЕКСТ С ТЕНЬЮ

text-shadow

CSS **глава-12/text-shadow.html**

```
p.one {
  background-color: #eeeeee;
  color: #666666;
  text-shadow: 1px 1px 0px #000000;}
p.two {
  background-color: #dddddd;
  color: #666666;
  text-shadow: 1px 1px 3px #666666;}
p.three {
  background-color: #cccccc;
  color: #ffffff;
  text-shadow: 2px 2px 7px #111111;}
p.four {
  background-color: #bbbbbb;
  color: #cccccc;
  text-shadow: -1px -2px #666666;}
p.five {
  background-color: #aaaaaa;
  color: #ffffff;
  text-shadow: -1px -1px #666666;}
```

РЕЗУЛЬТАТ



Свойство `text-shadow` широко используется, несмотря на то, что поддерживается не всеми браузерами.

Оно используется для создания тени, представляющей собой более темную и слегка смещенную версию слова, расположенную позади него. С помощью этого свойства также можно создать эффект тиснения, добавив тень, которая несколько светлее самого текста.

Для создания тени свойству нужно сообщить довольно сложное значение, состоящее из трех длин и цвета тени.

Первое значение длины указывает, насколько далеко влево или вправо должна падать тень.

Второе значение указывает, на какое расстояние вверх или вниз должна смещаться тень.

Третье значение необязательно и определяет степень размытия, применяемого к тени.

Четвертое — это цвет тени.

Свойство `text-shadow` стало очень популярным, однако не работает в браузере Internet Explorer до версии 9 (его поддержка осуществлена только в версии 10). Другие производители браузеров обеспечили поддержку этого свойства в довольно ранних версиях: Firefox 3.1, Safari 3, Chrome 2 и Opera 9.5.

ПЕРВАЯ БУКВА ИЛИ СТРОКА

:first-letter, :first-line

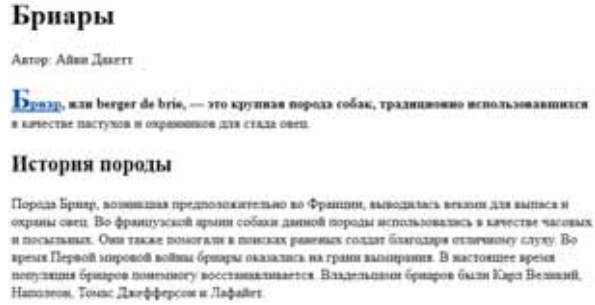
Вы можете задать различные значения первой букве или первой строке текста внутри элемента с помощью `:first-letter` и `:first-line`.

Технически это не свойства, а **псевдоэлементы**. Псевдоэлемент указывается в конце селектора, а за ним следуют объявления, как в любом другом элементе. Следующий пример стоит проверить в браузере, чтобы увидеть, как псевдоэлемент `first-line` будет влиять только на первую строку текста даже при изменении размера окна браузера и появлении на строке различного количества слов.

глава-12/first-letter-and-line.html CSS

```
p.intro:first-letter {
  font-size: 200%;}
p.intro:first-line {
  font-weight: bold;}
```

РЕЗУЛЬТАТ



Бриары

Автор: Айви Дакетт

Бриар, или berger de brie, — это крупная порода собак, традиционно использовавшихся в качестве пастухов и охранников для стада овец.

История породы

Порода Бриар, возникшая предположительно во Франции, выводилась вначале для выпаса и охраны овец. Во французской армии собаки данной породы использовались в качестве часовых и посыльных. Они также помогали в поисках раненых солдат благодаря отличному слуху. Во время Первой мировой войны бриары оказались на грани вымирания. В настоящее время популяция бриаров понемногу восстанавливается. Владельцами бриаров были Карл Великий, Наполеон, Томас Джефферсон и Лафайет.

В CSS предусматриваются как псевдоэлементы, так и псевдоклассы. Псевдоэлемент действует как дополнительный элемент в коде. При использовании псевдоэлементов `:first-letter` и `:first-line` вокруг первой буквы или первой строки текста как будто создается дополнительный элемент, к которому можно применить собственные стили.

Псевдокласс ведет себя как дополнительное значение атрибута `class`. Псевдокласс `:visited`, о котором речь пойдет далее, позволяет применять различные стили к посещенным ссылкам. Подобным же образом псевдокласс `:hover` позволяет менять стиль элементов при наведении на них указателя мыши.

ПРИМЕНЕНИЕ СТИЛЕЙ К ССЫЛКАМ

:link, :visited

CSS глава-12/link-visited.html

```
a:link {
  color: deeppink;
  text-decoration: none;}
a:visited {
  color: black;}
a:hover {
  color: deeppink;
  text-decoration: underline;}
a:active {
  color: darkcyan;}
```

РЕЗУЛЬТАТ

Породы собак: Б

- Бассет-хаунд
- Бедлингтон-терьер
- Бельгийская овчарка
- Бергамская овчарка
- Бигль
- Бишон фризе
- Бладхаунд
- Болоньез
- Бордер-колли
- Бордер-терьер
- Борзая русская псовая
- Бородатый колли
- Босерон
- Бриар
- Бувье фландрский
- Бульдог
- Бультерьер

По умолчанию браузеры, как правило, выделяют ссылки синим цветом и подчеркиванием, а также меняют цвет посещенной ссылки, чтобы помочь пользователю отследить страницы, на которых он уже побывал.
В CSS существуют два псевдокласса, позволяющих применять различные стили к посещенным и не посещенным ссылкам.

:link

Этот псевдокласс позволяет применить стиль к еще не посещенной ссылке.

:visited

Данный псевдокласс позволяет применить стиль к ссылке, по которой пользователь уже щелкнул.

Эти псевдоклассы обычно используются, чтобы управлять цветом и подчеркиванием ссылок.
Слева показано, что посещенные ссылки выделены другим цветом, чтобы помочь посетителю отследить страницы, которые он уже видел.
Часто для изменения внешнего вида ссылки при щелчке или наведении на нее указателя мыши используются псевдоклассы `:hover` и `:active`, речь о которых пойдет далее.

РЕАГИРОВАНИЕ НА ДЕЙСТВИЯ ПОЛЬЗОВАТЕЛЯ

:hover, :active, :focus

Существует три псевдокласса, позволяющих изменять внешний вид элементов в ответ на действия пользователя.

:hover

Применяется, когда пользователь наводит на элемент указатель мыши. Данный псевдокласс обычно используется для изменения внешнего вида ссылок и кнопок. Стоит отметить, что подобные события не работают на устройствах с сенсорными экранами (например, iPad), поскольку экран не может понять, когда пользователь наводит на элемент палец.

:active

Применяется, когда элемент активируется пользователем, например путем нажатия кнопки или щелчка по ссылке. Иногда этот псевдокласс используется, чтобы создать иллюзию нажатия, слегка изменяя стиль или положение элемента.

:focus

Применяется, когда элемент имеет фокус. Любой элемент, с которым вы можете взаимодействовать, например ссылка или элемент формы, способен иметь фокус.

глава-12/hover-active-focus.html CSS

```
input {
  padding: 6px 12px 6px 12px;
  border: 1px solid #665544;
  color: #ffffff;}
input.submit:hover {
  background-color: #665544;}
input.submit:active {
  background-color: chocolate;}
input.text {
  color: #cccccc;}
input.text:focus {
  color: #665544;}
```

РЕЗУЛЬТАТ

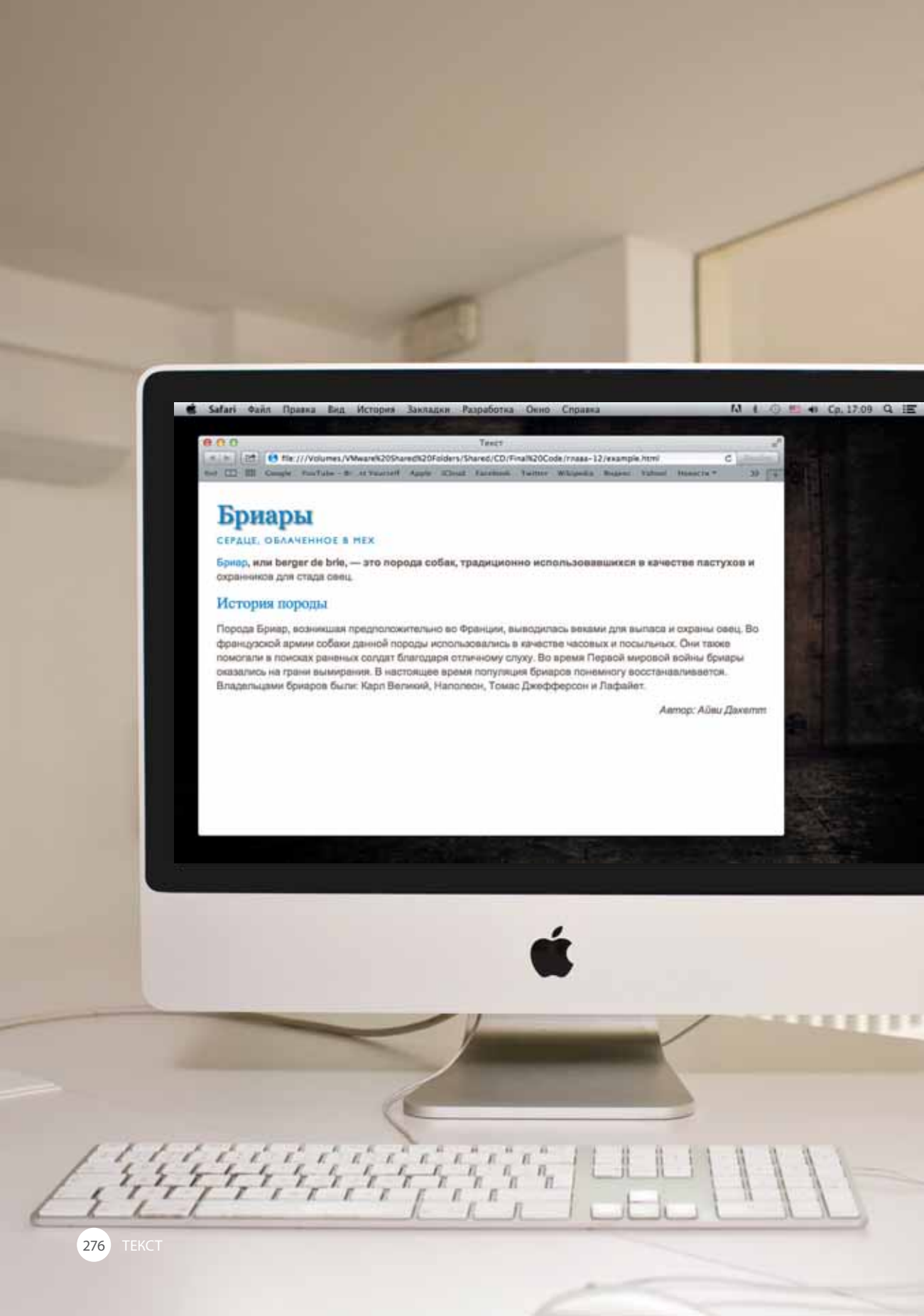
Фокус возникает, когда браузер понимает, что вы готовы взаимодействовать с элементом на странице. Например когда указатель мыши находится в поле ввода формы, готовой к заполнению, считается, что данный элемент обладает фокусом. Для последовательного перемещения по интерактивным элементам страницы также можно использовать клавишу табуляции. При использовании псевдоклассы должны перечисляться в следующем порядке:

:link;
:visited;
:hover;
:focus;
:active.

СЕЛЕКТОРЫ АТРИБУТОВ

С самыми популярными селекторами CSS вы познакомились в главе 10. Существует также набор селекторов атрибутов, позволяющих создавать правила, которые применяются к элементам, имеющим атрибут с конкретным значением.

СЕЛЕКТОР	ОПРЕДЕЛЕНИЕ	ПРИМЕР
СУЩЕСТВОВАНИЯ АТРИБУТА	**[]** Соответствует конкретному атрибуту независимо от его значения	`p[class]` Любой элемент `<p>` с атрибутом `class`
РАВЕНСТВА АТРИБУТА	**[=]** Соответствует конкретному атрибуту с конкретным значением	`p[class="dog"]` Любой элемент `<p>` с атрибутом `class`, значение которого — `dog`
АТРИБУТА С ПРОБЕЛАМИ	**[~=]** Соответствует конкретному атрибуту, чье значение присутствует в списке слов, разделенных пробелами	`p[class~="dog"]` Любой элемент `<p>` с атрибутом `class`, значением которого является список разделенных пробелами слов, среди которых присутствует `dog`
ПРЕФИКСА АТРИБУТА	**[^=]** Соответствует конкретному атрибуту, чье значение начинается с конкретной строки	`p[attr^"d"]` Любой элемент `<p>` с атрибутом, значение которого начинается с буквы `d`
ПОДСТРОКИ АТРИБУТА	**[*=]** Соответствует конкретному атрибуту, чье значение содержит конкретную подстроку	`p[attr*"do"]` Любой элемент `<p>` с атрибутом, значение которого содержит буквы `do`
СУФФИКСА АТРИБУТА	**[$=]** Соответствует конкретному атрибуту, чье значение заканчивается конкретной строкой	`p[attr$"g"]` Любой элемент `<p>` с атрибутом, значение которого заканчивается буквой `g`


Safari Файл Правка Вид История Закладки Разработка Окно Справка
Ср. 17:09
Текст
file:///Volumes/VMware%20Shared%20Folders/Shared/CD/Final%20Code/глава-12/example.html
Бриары
СЕРДЦЕ, ОБЛАЧЕННОЕ В МЕХ
Бриар, или berger de brie, — это порода собак, традиционно использовавшихся в качестве пастухов и охранников для стада овец.
История породы
Порода Бриар, возникшая предположительно во Франции, выводилась веками для выпаса и охраны овец. Во французской армии собаки данной породы использовались в качестве часовых и посыльных. Они также помогали в поисках раненых солдат благодаря отличному слуху. Во время Первой мировой войны бриары оказались на грани вымирания. В настоящее время популяция бриаров понемногу восстанавливается. Владельцами бриаров были: Карл Великий, Наполеон, Томас Джефферсон и Лафайет.
Автор: Айви Дакетт

ПРИМЕР
ТЕКСТ

Этот пример сочетает в себе многие из раскрытых в настоящей главе приемов.

Размеры шрифтов настраиваются с помощью свойства `font-size`. Начертание текста можно поменять с полужирного на нормальное, задав свойство `font-weight`. Мы также указали различные варианты шрифтов, используя свойство `font-family`.

В элементе `<h1>` используется свойство `text-shadow` спецификации CSS3 для создания тени позади заголовка. Регистр текста в элементе `<h2>` меняется на верхний с помощью свойства `text-transform`, а для повышения удобочитаемости текста, состоящего из заглавных букв, мы увеличили межбуквенный интервал, используя свойство `letter-spacing`.

С целью повышения удобочитаемости основного текста мы увеличили междустрочный интервал, воспользовавшись свойством `line-height`. Псевдоэлемент `:first-line` позволил нам применить полужирное начертание к первой строке первого абзаца. Наконец, имя автора выделено курсивом и выровнено по правой стороне страницы.

ПРИМЕР
ТЕКСТ

```
<!DOCTYPE html>
<html>
  <head>
    <title>Текст</title>
    <style type="text/css">
      body {
        padding: 20px;}
      h1, h2, h3, a {
        font-weight: normal;
        color: #0088dd;
        margin: 0px;}
      h1 {
        font-family: Georgia, Times, serif;
        font-size: 250%;
        text-shadow: 2px 2px 3px #666666;
        padding-bottom: 10px;}
      h2 {
        font-family: "Gill Sans", Arial, sans-serif;
        font-size: 90%;
        text-transform: uppercase;
        letter-spacing: 0.2em;}
      h3 {
        font-size: 150%;}
      p {
        font-family: Arial, Verdana, sans-serif;
        line-height: 1.4em;
        color: #665544;}
      p.intro:first-line {
        font-weight: bold;}
      .credits {
        font-style: italic;
        text-align: right;}
      a {
        text-decoration: none;}
      a:hover {
        text-decoration: underline;}
    </style>
  </head>
  <body>
```

ПРИМЕР
ТЕКСТ

```
    <h1>Бриары</h1>
    <h2>Сердце, облаченное в мех</h2>
    <p class="intro"><a class="breed" href=" http://ru.wikipedia.org/wiki/Бриар_
      (порода_собак)">Бриар</a>, или berger de brie, - это порода собак,
      традиционно использовавшихся в качестве пастухов и охранников для стада
      овец.</p>
    <h3>История породы</h3>
    <p>Порода бриар, возникшая предположительно во Франции, выводилась
      веками для выпаса и охраны овец. Во французской армии собаки данной породы
      использовались в качестве часовых и посыльных. Они также помогали
      в поисках раненых солдат благодаря отличному слуху. Во время Первой
      мировой войны бриары оказались на грани вымирания. В настоящее время
      популяция бриаров понемногу восстанавливается. Владельцами бриаров были
      Карл Великий, Наполеон, Томас Джефферсон и Лафайет.</p>
    <p class="credits">Автор: Айви Дакетт</p>
  </body>
</html>
```

ОБЗОР
ТЕКСТ

- Существуют свойства, позволяющие выбирать шрифт, управлять его размером, весом, стилизовать и устанавливать различные интервалы.
- Существует небольшое количество шрифтов, в наличии которых на компьютере пользователя вы можете быть уверены.
- Есть несколько методов, позволяющих использовать более широкий спектр гарнитур шрифтов, однако они требуют наличия определенных лицензий.
- Вы можете контролировать расстояние между строками текста, отдельными буквами и словами. Текст выравнивается по левому краю, по правому, по центру или по ширине строки. Также текст может иметь отступы.
- Псевдоклассы используются, чтобы стиль элемента менялся при наведении на него указателя мыши, при щелчке, а также для выделения посещенной ссылки.

13

БЛОКИ

- Управление размером блоков
- Блочная модель: границы, поля и отступы
- Отображение и скрытие блоков

Ранее вы узнали, что CSS обращается с каждым HTML-элементом так, как будто он заключен в блок.

Вы можете использовать несколько свойств для управления внешним видом этих блоков. В настоящей главе вы научитесь:

- контролировать размеры блоков;
- создавать границы вокруг блоков;
- устанавливать поля и отступы для блоков;
- отображать и скрывать блоки.

После того как вы научитесь управлять внешним видом каждого блока, мы перейдем к позиционированию этих блоков, когда будем разбираться с макетом страницы в главе 15.

PICCADILLY
EXPRESS
TRAINS
EALING COMMON
PARK ROYAL
SUDBURY TOWN
SOUTH HARROW
FROM 4 JULY
DUNLOP
FLICKR
ASP.NET 2.0
XML
2nd Edition
Web Programming with HTML, XHTML, and CSS
AMBROSE / HARRIS
HTML5

РАЗМЕРЫ БЛОКА

width, height

По умолчанию размер блока задается так, чтобы он был способен вместить контент. Чтобы установить собственные размеры блока, вы можете использовать свойства `height` и `width`. Чаще всего размер блока указывается в пикселах, процентах или единицах em. Пикселы наиболее популярны, поскольку позволяют точно контролировать размер.

При использовании процентов размер блока задается относительно размера окна браузера, а если блок заключен в другой блок, то относительно его размера. При использовании единиц em размер блока зависит от размера содержащегося в нем текста. В последнее время разработчики начали чаще использовать проценты и единицы em при создании гибких дизайнов, предусматривающих доступ с устройств с разными размерами экранов.

В следующем примере вы видите элемент `<div>`, ширина и высота которого равна 300 пикселам. Внутри него находится абзац текста, занимающий 75% ширины и высоты элемента `<div>`. Это означает, что размер абзаца составляет 225 пикселов в ширину и 225 пикселов в высоту.

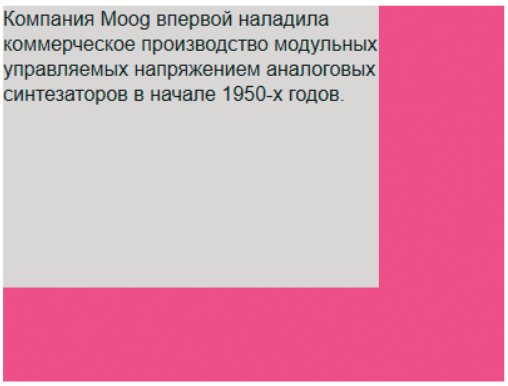
глава-13/width-height.html HTML

```
<div>
  <p> Компания Moog первой наладила коммерческое
      производство модульных управляемых напряжением
      аналоговых синтезаторов в начале
      1950-х годов.</p>
</div>
```

CSS

```
div.box {
  height: 300px;
  width: 300px;
  background-color: #bbbbaa;}
p {
  height: 75%;
  width: 75%;
  background-color: #0088dd;}
```

РЕЗУЛЬТАТ



ОГРАНИЧЕНИЕ ШИРИНЫ

min-width, max-width

HTML **глава-13/min-width-max-width.html**

```
<tr>
  <td><img src="images/rhodes.jpg" width="200"
      height="100" alt="Фендер-Родес" /></td>
  <td class="description">Родес-пиано — это
    разновидность электромеханического пианино. Оно
    было изобретено Гарольдом Родесом в пятидесятые
    годы и позднее выпускалось в ряде вариаций...</td>
  <td>240000 рублей</td>
</tr>
```

CSS

```
td.description {
  min-width: 450px;
  max-width: 650px;
  text-align: left;
  padding: 5px;
  margin: 0px;}
```

РЕЗУЛЬТАТ

Некоторые страницы создаются таким образом, чтобы контент менял свой размер в соответствии с размером экрана. В этом случае свойство `min-width` определяет наименьшую ширину блока при уменьшении окна браузера, а свойство `max-width` — наибольшую ширину блока при увеличении. Это очень полезные свойства, позволяющие гарантировать удобочитаемость страниц, особенно на маленьких экранах портативных устройств. Например, вы можете использовать свойство `max-width`, чтобы быть уверенным в том, что строки текста не окажутся слишком широкими в большом окне браузера. Свойство `min-width` позволит гарантировать, что строки текста не окажутся чересчур узкими.

Проверьте данный пример в своем браузере, чтобы понять, что происходит, когда вы увеличиваете или уменьшаете размер окна программы.

ОГРАНИЧЕНИЕ ВЫСОТЫ

min-height, max-height

Так же как и в случае с шириной блока, вы можете ограничить его высоту. Это достигается с помощью свойств `min-height` и `max-height`.

Следующий пример демонстрирует эти свойства в действии. Кроме того, он показывает, что происходит, когда контент занимает больше места, чем позволяет размер блока.

Если блок недостаточно велик, чтобы вместить контент, тот выходит за его пределы, что выглядит очень некрасиво. Чтобы решить проблему, возникающую при нехватке места в блоке, вы можете использовать свойство `overflow`, которое обсуждается далее.

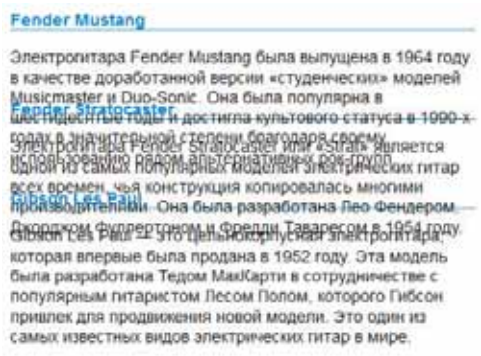
глава-13/min-height-max-height.html HTML

```
<h2>Fender Mustang</h2>
<p> Электрогитара Fender Mustang была выпущена
    в 1964 году в качестве доработанной версии...</p>
<h2>Fender Stratocaster</h2>
<p> Электрогитара Fender Stratocaster или "Strat"
    является одной из самых популярных моделей
    электрических гитар всех времен...</p>
<h2>Gibson Les Paul</h2>
<p> Gibson Les Paul — это цельнокорпусная
    электрогитара, которая впервые была продана
    в 1952 году...</p>
```

CSS

```
h2, p {
  width: 400px;
  font-size: 90%;
  line-height: 1.2em;}
h2 {
  color: #0088dd;
  border-bottom: 1px solid #0088dd;}
p {
  min-height: 10px;
  max-height: 30px;}
```

РЕЗУЛЬТАТ



ПЕРЕТЕКАНИЕ КОНТЕНТА

overflow

HTML **глава-13/overflow.html**

```
<h2>Fender Stratocaster</h2>
<p class="one">Электрогитара Fender Stratocaster или
    "Strat" является одной из самых популярных
    моделей электрических гитар всех времен, чья
    конструкция копировалась многими
    производителями. Она была разработана Лео...</p>
<h2>Gibson Les Paul</h2>
<p class="two">Gibson Les Paul — это цельнокорпусная
    электрогитара, которая впервые была продана
    в 1952 году. Эту модель разработал Тед
    Маккарти...</p>
```

CSS

```
p.one {
  overflow: hidden;}
p.two {
  overflow: scroll;}
```

РЕЗУЛЬТАТ

Свойство `overflow` сообщает браузеру, что делать, если объем контента блока превышает его размер. Это свойство может принимать одно из двух значений.

hidden

Данное значение просто скрывает любой дополнительный контент, который не вписывается в блок.

scroll

Данное значение добавляет к блоку полосу прокрутки, чтобы пользователи могли увидеть дополнительный контент.

В приведенном примере вы можете видеть два блока, содержимое которых выходит за их пределы. В первой части примера свойству `overflow` присвоено значение `hidden`, во второй части — `scroll`.

Свойство `overflow` особенно удобно, поскольку некоторые браузеры позволяют пользователям настраивать размер шрифта текста. При слишком большом размере шрифта страница становится нечитаемой. Сокрытие не помещающегося в блок контента помогает предотвратить наползание элементов на странице друг на друга.

ГРАНИЦЫ, ПОЛЯ И ОТСТУПЫ

Управлять внешним видом блока можно с помощью трех следующих характеристик.

1

ГРАНИЦА

Каждый блок имеет границы, даже если они не видны или их толщина равна 0 пикселам. Границы отделяют края одного блока от краев других.

2

ПОЛЯ

Поля находятся за пределами границ. Вы можете настроить ширину полей для создания интервала между границами двух соседних блоков.

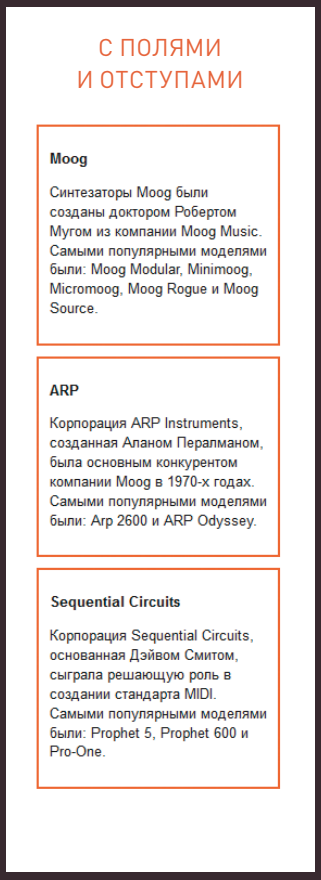
3

ОТСТУПЫ

Отступ — это интервал между границами блока и его содержимым. Добавление отступов может улучшить удобочитаемость контента.

Если вы задали ширину блока, то границы, поля и отступы будут добавлены к его ширине и высоте.

ВОЗДУХ И ВЕРТИКАЛЬНЫЕ ПОЛЯ



Свойства `padding` и `margin` позволяют добавлять пустое пространство между различными элементами на странице.

Интервалы между элементами на странице дизайнеры называют воздухом. Представьте, что вокруг блока есть границы. Текст не должен касаться этих границ, иначе его трудно будет воспринимать.

Или представьте два блока, расположенных рядом (оба с черными границами): если бы они соприкасались, то линия между ними оказалась бы вдвое толще.

Если нижнее поле одного блока соприкасается с верхним полем другого, то браузер будет отображать их не так, как вы ожидаете. Программа покажет только большее из двух полей. Если оба поля имеют одинаковый размер, то браузер отобразит только одно из них.

ТОЛЩИНА ГРАНИЦ
border-width

Свойство `border-width` используется для управления толщиной границ. Данное свойство может принимать значение в пикселах или одно из следующих значений:

`thin`;
`medium`;
`thick`.

Для этого свойства нельзя задавать значение в процентах.
Вы можете устанавливать размер отдельных границ, используя следующие четыре свойства:

`border-top-width`;
`border-right-width`;
`border-bottom-width`;
`border-left-width`.

Вы также можете указать разную толщину для четырех границ в одном свойстве.

```
border-width: 2px 1px
1px 2px;
```

В данном случае значения перечислены по часовой стрелке: верхняя, правая, нижняя и левая границы.

глава-13/border-width.html HTML

```
<p class="one">Клавинет Хонера — это
    "электрифицированная" версия клавикорда.</p>
<p class="two">Клавинет Хонера — это
    "электрифицированная" версия клавикорда.</p>
<p class="three">Клавинет Хонера — это
    "электрифицированная" версия клавикорда.</p>
```

CSS

```
p.one {
  border-width: 2px;}
p.two {
  border-width: thick;}
p.three {
  border-width: 1px 4px 12px 4px;}
```

РЕЗУЛЬТАТ

Клавинет Хонера — это «электрифицированная» версия клавикорда.

Клавинет Хонера — это «электрифицированная» версия клавикорда.

Клавинет Хонера — это «электрифицированная» версия клавикорда.

СТИЛЬ ГРАНИЦЫ
border-style

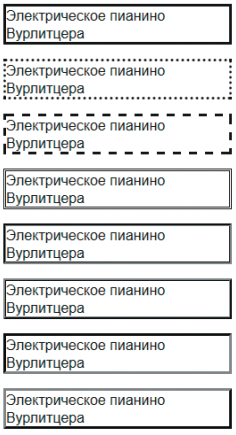
HTML **глава-13/border-style.html**

```
<p class="one">Электрическое пианино Вурлитцера</p>
<p class="two">Электрическое пианино Вурлитцера</p>
<p class="three">Электрическое пианино Вурлитцера</p>
<p class="four">Электрическое пианино Вурлитцера</p>
<p class="five">Электрическое пианино Вурлитцера</p>
<p class="six">Электрическое пианино Вурлитцера</p>
<p class="seven">Электрическое пианино Вурлитцера</p>
<p class="eight">Электрическое пианино Вурлитцера</p>
```

CSS

```
p.one {border-style: solid;}
p.two {border-style: dotted;}
p.three {border-style: dashed;}
p.four {border-style: double;}
p.five {border-style: groove;}
p.six {border-style: ridge;}
p.seven {border-style: inset;}
p.eight {border-style: outset;}
```

РЕЗУЛЬТАТ



Вы можете настраивать стиль границ, используя свойство `border-style`. Оно способно принимать следующие значения:

`solid` — непрерывная линия;

`dotted` — пунктирная в виде точек (если толщина границы равна 2 пикселам, то размер точки составляет 2 × 2 пиксела, а интервал между точками — 2 пиксела);

`dashed` — пунктирная;

`double` — две сплошные линии (значение свойства `border-width` создает сочетание двух линий);

`groove` — врезанная линия;

`ridge` — выпуклая линия;

`inset` — вдавленная линия;

`outset` — выдавленная линия;

`hidden`/**`none`** — граница не видна.

Вы можете изменить стиль отдельных границ, используя следующие свойства:

`border-top-style`;
`border-left-style`;
`border-right-style`;
`border-bottom-style`.

ЦВЕТ ГРАНИЦЫ
border-color

Вы можете указать цвет границ, используя значения RGB, шестнадцатеричные коды или имена цветов (как описано в главе 11). Цветами отдельных границ блока можно управлять с помощью следующих свойств:

`border-top-color`;
`border-right-color`;
`border-bottom-color`;
`border-left-color`.

Кроме того, можно указать разные цвета для всех четырех границ в одном свойстве.

```
border-color:
darkcyan
deeppink darkcyan
deeppink;
```

Здесь значения перечислены по часовой стрелке: верхняя, правая, нижняя и левая границы.
Вы также можете использовать значения HSL, чтобы указать цвет, как показано в главе 11. Однако этот метод был введен только в CSS3 и не сработает в старых версиях браузеров.

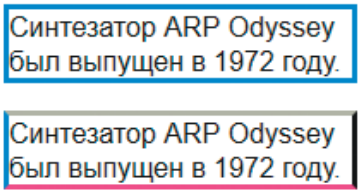
глава-13/border-color.html HTML

```
<p class="one">Синтезатор ARP Odyssey был выпущен
    в 1972 году.</p>
<p class="two">Синтезатор ARP Odyssey был выпущен
    в 1972 году.</p>
```

CSS

```
p.one {
  border-color: #0088dd;}
p.two {
  border-color: #bbbbaa #111111 #ee3e80 #0088dd;}
```

РЕЗУЛЬТАТ



СТЕНОГРАФИЧЕСКИЕ СВОЙСТВА

border

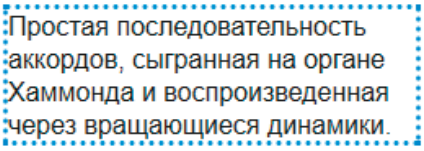
HTML **глава-13/border-shorthand.html**

```
<p> Простая последовательность аккордов, сыгранная
    на органе Хаммонда и воспроизведенная через
    вращающиеся динамики.</p>
```

CSS

```
p {
  width: 250px;
  border: 3px dotted #0088dd;}
```

РЕЗУЛЬТАТ



Свойство `border` позволяет одновременно указать толщину, стиль и цвет границы. При этом значения должны указываться именно в таком порядке.

ОТСТУПЫ
padding

Свойство `padding` позволяет указать, сколько места должно быть между содержимым элемента и его границами. Значение этого свойства чаще всего указывается в пикселах (хотя также можно использовать проценты или единицы em). При использовании процентных значений размер отступа рассчитывается в процентном отношении к размеру окна браузера (или окружающей блока, если он есть). Обратите внимание: если для блока указана ширина, то значение отступа прибавляется к значению ширины.

В данном примере второй абзац воспринимается легче благодаря интервалу между текстом и границами блока. Из-за отступа ширина блока стала больше.

Вы можете задать различные значения для каждой стороны блока, используя свойства:

`padding-top`;
`padding-right`;
`padding-bottom`;
`padding-left`.

Вы также можете использовать стенографическое свойство, где значения перечисляются по часовой стрелке: верх, правая сторона, низ, левая сторона: `padding: 10px 5px 3px 1px;`

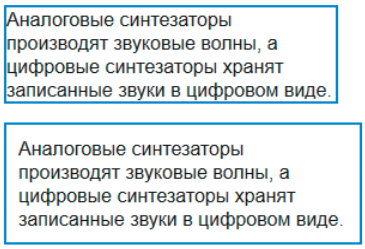
глава-13/padding.html HTML

```
<p> Аналоговые синтезаторы производят звуковые
    волны, а цифровые синтезаторы хранят записанные
    звуки в цифровом виде.</p>
<p class="example">Аналоговые синтезаторы производят
    звуковые волны, а цифровые синтезаторы хранят
    записанные звуки в цифровом виде.</p>
```

CSS

```
p {
  width: 275px;
  border: 2px solid #0088dd;}
p.example {
  padding: 10px;}
```

РЕЗУЛЬТАТ



Значение свойства `padding` не наследуется дочерними элементами таким же образом, как значение `color` свойства `font-family`, поэтому вам нужно указать величину отступов для каждого элемента, которому они необходимы.

До появления браузера Internet Explorer 6 ширина блока включала отступы и поля. Далее в главе вы узнаете об этом подробнее.

ПОЛЯ
margin

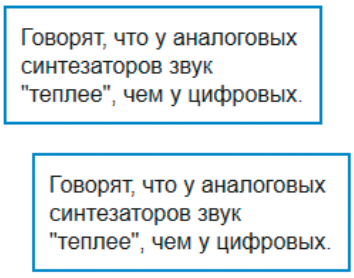
HTML **глава-13/margin.html**

```
<p> Говорят, что у аналоговых синтезаторов звук
    "теплее", чем у цифровых.</p>
<p class="example">Говорят, что у аналоговых
    синтезаторов звук "теплее", чем у цифровых.</p>
```

CSS

```
p {
  width: 200px;
  border: 2px solid #0088dd;
  padding: 10px;}
p.example {
  margin: 20px;}
```

РЕЗУЛЬТАТ



Значение свойства `margin` не наследуется дочерними элементами таким же образом, как значение `color` свойства `font-family`, поэтому вам нужно указать величину полей для каждого элемента, которому они необходимы.

До появления браузера Internet Explorer 6 ширина блока включала отступы и поля. Далее в главе вы узнаете об этом подробнее.

Свойство `margin` определяет интервал между блоками. Его значение обычно задается в пикселах, однако вы можете также использовать проценты или единицы em. Если один блок располагается поверх другого, то будут использоваться поля, имеющие наибольшее значение. Обратите внимание: если для блока указана ширина, то величина полей прибавляется к ее значению.
Вы можете указать значения для каждой стороны блока, используя свойства:

`margin-top`;
`margin-right`;
`margin-bottom`;
`margin-left`.

Вы также можете использовать стенографическое свойство, где значения перечисляются по часовой стрелке: верх, право, низ, лево:

```
margin: 1px 2px 3px
4px;
```

Вам также может встретиться следующее выражение:

```
margin: 10px 20px;
```

Оно означает, что значения левого и правого полей должны быть равны 10 пикселам, а значения верхнего и нижнего — 20 пикселам. Подобное краткое выражение может быть применено к значениям отступов.

ЦЕНТРИРОВАНИЕ КОНТЕНТА

Если вы хотите поместить блок в центре страницы (или в центре элемента, в котором он находится), вы можете присвоить значение `auto` свойствам `margin-left` и `margin-right`.

Чтобы центрировать блок на странице, вам нужно указать его ширину (`width`), в противном случае он займет всю ширину страницы.

После указания ширины блока, задание значения `auto` для правого и левого полей дает браузеру команду создать одинаковый интервал с каждого края блока. Это приведет к центрированию блока на странице (или в содержащем его элементе).

Чтобы это работало в старых браузерах (в частности, Internet Explorer 6), элемент, в котором находится блок, должен иметь свойство `text-align` со значением `center`.

Свойство `text-align` наследуется дочерними элементами, поэтому вам придется настроить его для блока, если вы не хотите, чтобы содержащийся в нем текст был отцентрирован.

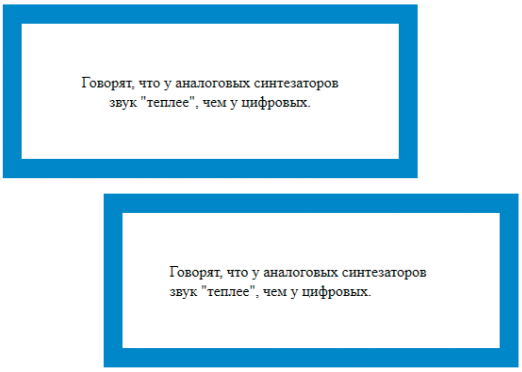
глава-13/centering-content.html HTML

```
<body>
  <p> Говорят, что у аналоговых синтезаторов звук
      "теплее", чем у цифровых.</p>
  <p class="example">Говорят, что у аналоговых
      синтезаторов звук "теплее", чем у цифровых.</p>
</body>
```

CSS

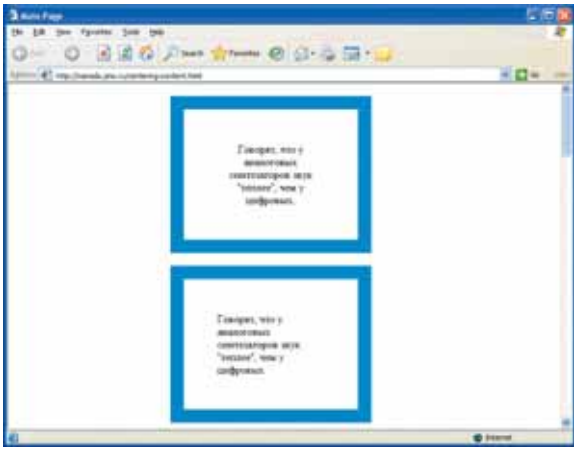
```
body {
  text-align: center;}
p {
  width: 300px;
  padding: 50px;
  border: 20px solid #0088dd;}
p.example {
  margin: 10px auto 10px auto;
  text-align: left;}
```

РЕЗУЛЬТАТ

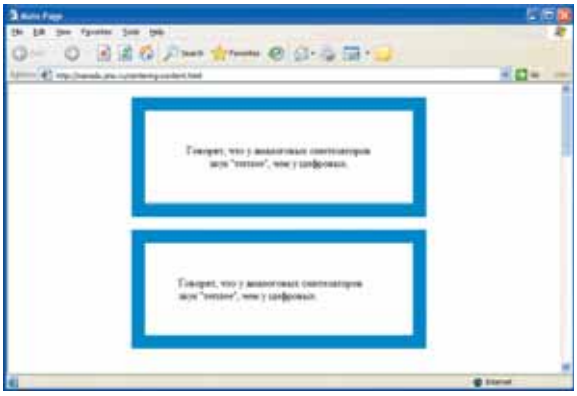


БЛОЧНАЯ МОДЕЛЬ INTERNET EXPLORER 6

РЕЗУЛЬТАТ В INTERNET EXPLORER 6 БЕЗ ТЕГА DOCTYPE



РЕЗУЛЬТАТ В INTERNET EXPLORER 6 С ТЕГОМ DOCTYPE



Когда вы задаете ширину блока, отступы и поля должны добавляться к значению ширины. Однако браузер Internet Explorer 6 включает отступы и поля в ширину блока.

Во избежание этой ошибки следует включить в HTML-код тег `DOCTYPE`, о котором говорилось в главе 8. Чтобы программа Internet Explorer 6 использовала корректную блочную модель, вы можете использовать тег `DOCTYPE` в HTML5, строгой и переходной версиях HTML 4. Рассмотрите два приведенных примера — без тега `DOCTYPE` (первый) и с тегом `DOCTYPE` языка HTML5 (второй).

ПРЕВРАЩЕНИЕ ВСТРОЕННЫХ ЭЛЕМЕНТОВ В БЛОЧНЫЕ И НАОБОРОТ

display

Свойство `display` позволяет превратить встроенный элемент в блочный и наоборот, а также может использоваться для скрытия элемента на странице. Оно способно принимать следующие значения.

inline

Это значение заставляет блочный элемент вести себя как встроенный.

block

Это значение заставляет встроенный элемент вести себя как блочный.

inline-block

Это значение заставляет блочный элемент располагаться как встроенный, но при этом сохранять остальные особенности блочного.

none

Это значение скрывает элемент на странице. В этом случае элемент действует так, будто вообще отсутствует (хотя посетитель может увидеть содержимое блока при просмотре исходного кода).

При использовании этого свойства важно знать, что встроенные блоки *не предназначены* для создания блочных элементов.

глава-13/display.html HTML

```
<ul>
  <li>Домой</li>
  <li>Товары</li>
  <li class="coming-soon">Услуги</li>
  <li>О нас</li>
  <li>Контакты</li>
</ul>
```

CSS

```
li {
  display: inline;
  margin-right: 10px;}
li.coming-soon {
  display: none;}
```

РЕЗУЛЬТАТ

Домой Товары О нас Контакты

В этом примере представлен список. Каждый элемент списка обычно рассматривается как блочный, однако правило для элементов `<li>` указывает на то, что они должны рассматриваться как встроенные. Это означает, что они будут располагаться рядом друг с другом, а не на новых строках. Данная техника часто используется для создания меню навигации. В примере справа от каждого элемента было добавлено поле, чтобы отделить их друг от друга. Правило, которое применяется к элементу `<li>` с классом `coming-soon`, скрыто, будто оно отсутствует на странице.

СОКРЫТИЕ БЛОКОВ
visibility

HTML **глава-13/visibility.html**

```
<ul>
  <li>Домой</li>
  <li>Товары</li>
  <li class="coming-soon">Услуги</li>
  <li>О нас</li>
  <li>Контакты</li>
</ul>
```

CSS

```
li {
  display: inline;
  margin-right: 10px;}
li.coming-soon {
  visibility: hidden;}
```

РЕЗУЛЬТАТ

Домой Товары О нас Контакты

Свойство `visibility` позволяет скрывать блоки, но при этом оставлять пустое пространство на месте элемента. Данное свойство может принимать два значения.

hidden

Данное значение скрывает элемент.

visible

Данное значение отображает элемент.

Если для свойства `visibility` задано значение `hidden`, то на месте элемента появится пустое пространство.
Если вы хотите избежать появления пустого пространства, следует использовать свойство `display` со значением `none` (как говорилось в предыдущем разделе).
Обратите внимание, что посетитель может просмотреть содержимое любого скрытого элемента в исходном коде страницы.

CSS3: ДЕКОРАТИВНЫЕ ГРАНИЦЫ border-image

Свойство `border-image` применяет к границам блока изображение. Оно разделяет фоновое изображение на девять частей.

Ниже приведен пример. Добавлены отметки разделения. В каждом углу расположен круг на расстоянии 18 пикселов от краев. Угловые фрагменты всегда располагаются в этих позициях, однако вы можете выбрать, будут боковые фрагменты растягиваться или повторяться.

Этому свойству требуется следующая информация:
URL-адрес изображения;
расположение мест разделения изображения;
что делать с боковыми фрагментами, при этом допустимые значения могут быть следующими:
`stretch` — растягивает изображение;
`repeat` — повторяет изображение;
`round` — повторяет изображение, но при несовпадении фрагментов масштабирует их.

Чтобы сделать изображение видимым, для блока нужно задать толщину границы.

Свойства `-moz-border-image` и `-webkit-border-image` отсутствуют в CSS, но позволяют более ранним версиям браузеров Chrome, Firefox и Safari отображать данный эффект.

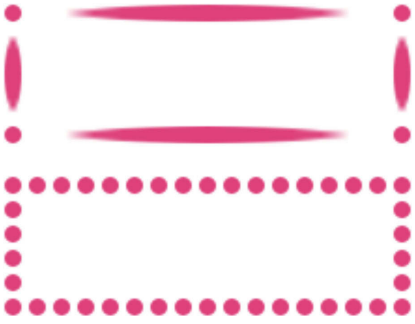
глава-13/border-image.html HTML

```
<p class="one"></p>
<p class="two"></p>
<p class="three"></p>
```

CSS

```
p.one {
  -moz-border-image: url("images/dots.gif")
    11 11 11 11 stretch;
  -webkit-border-image: url("images/dots.gif")
    11 11 11 11 stretch;
  border-image: url("images/dots.gif")
    11 11 11 11 stretch;}
p.two {
  -moz-border-image: url("images/dots.gif")
    11 11 11 11 round;
  -webkit-border-image: url("images/dots.gif")
    11 11 11 11 round;
  border-image: url("images/dots.gif")
    11 11 11 11 round;}
```

РЕЗУЛЬТАТ



CSS3: ТЕНИ БЛОКОВ

box-shadow

CSS **глава-13/box-shadow.html**

```
p.one {
  -moz-box-shadow: -5px -5px #777777;
  -webkit-box-shadow: -5px -5px #777777;
  box-shadow: -5px -5px #777777;}
p.two {
  -moz-box-shadow: 5px 5px 5px #777777;
  -webkit-box-shadow: 5px 5px 5px #777777;
  box-shadow: 5px 5px 5px #777777;}
p.three {
  -moz-box-shadow: 5px 5px 5px 5px #777777;
  -webkit-box-shadow: 5px 5px 5px 5px #777777;
  box-shadow: 5px 5px 5px 5px #777777;}
p.four {
  -moz-box-shadow: 0 0 10px #777777;
  -webkit-box-shadow: 0 0 10px #777777;
  box-shadow: 0 0 10px #777777;}
p.five {
  -moz-box-shadow: inset 0 0 10px #777777;
  -webkit-box-shadow: inset 0 0 10px #777777;
  box-shadow: inset 0 0 10px #777777;}
```

РЕЗУЛЬТАТ

Свойство `box-shadow` позволяет добавить тень вокруг блока. Оно работает так же, как свойство `text-shadow`, с которым вы познакомились в главе 12. Данное свойство должно использовать, по крайней мере, первое из следующих двух значений, а также цвет.

ГОРИЗОНТАЛЬНОЕ СМЕЩЕНИЕ

Отрицательные значения позиционируют тень слева от блока.

ВЕРТИКАЛЬНОЕ СМЕЩЕНИЕ

Отрицательные значения позиционируют тень над блоком.

РАЗМЫТИЕ

Если данное значение отсутствует, то тень изображается в виде сплошной линии, как граница.

РАСПРОСТРАНЕНИЕ ТЕНИ

Если данный параметр используется, то положительные значения заставят тень выйти за пределы блока во всех направлениях, а отрицательные значения заставят тень сжиматься внутри блока.

Для создания внутренней тени также можно использовать ключевое слово `inset`, добавленное перед перечисленными значениями.

Браузеры Chrome, Firefox и Safari поддерживают данный эффект с помощью свойств `-moz-box-shadow` и `-webkit-box-shadow`. Они не входят в спецификацию CSS, но их использование обеспечивает работоспособность эффекта в перечисленных браузерах.

CSS3: ЗАКРУГЛЕННЫЕ УГЛЫ

border-radius

В спецификации CSS3 появилась возможность создания закругленных углов блока с помощью свойства `border-radius`. Значение указывает радиус в пикселах.

Старые версии браузеров, которые не поддерживают это свойство, будут отображать блок с прямыми углами.

Свойства `-moz-border-radius` и `-webkit-border-radius` не входят в спецификацию CSS, однако их использование в некоторых версиях браузеров Chrome, Firefox и Safari обеспечивает работоспособность данного стиля.

Вы можете указать различные значения для каждого угла блока следующим образом:

`border-top-right-radius`;
`border-bottom-right-radius`;
`border-bottom-left-radius`;
`border-top-left-radius`.

Вы также можете использовать стенографические свойства (по часовой стрелке: верх, правый угол, низ, левый угол), например `border-radius: 5px, 10px, 5px, 10px;`

глава-13/border-radius.html HTML

```
<p> При записи альбома Pet Sounds был использован
    целый ряд нетрадиционных инструментов, таких как
    звонок велосипеда, орган, клавесин, флейты,
    свисток для собак, поезд, струнные инструменты
    с характерным гавайским звучанием, банки из-под
    кока-колы и лающие собаки.</p>
```

CSS

```
p {
  border: 5px solid #cccccc;
  padding: 20px;
  width: 275px;
  text-align: center;
  border-radius: 10px;
  -moz-border-radius: 10px;
  -webkit-border-radius: 10px;}
```

РЕЗУЛЬТАТ

При записи альбома Pet Sounds был использован целый ряд нетрадиционных инструментов, таких как звонок велосипеда, орган, клавесин, флейты, свисток для собак, поезд, струнные инструменты с характерным гавайским звучанием, банки из-под Кока-Колы и лающие собаки.

CSS3: ЭЛЛИПТИЧЕСКИЕ ФОРМЫ

border-radius

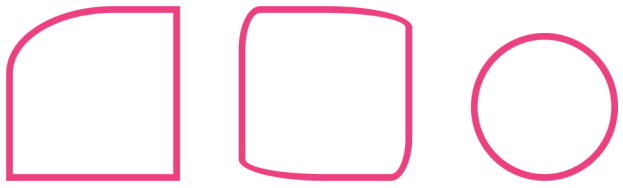
HTML глава-13/elliptical-shapes.html

```
<p class="one"></p>
<p class="two"></p>
<p class="three"></p>
```

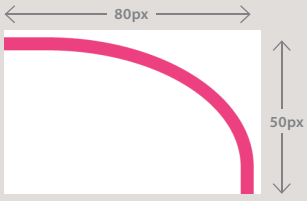
CSS

```
p.one {
  border-top-left-radius: 80px 50px;
  -moz-border-radius-top-left: 80px 50px;
  -webkit-border-radius-top-left: 80px 50px;}
p.two {
  border-radius: 1em 4em 1em 4em / 2em 1em 2em 1em;
  -moz-border-radius: 1em 4em 1em 4em
    / 2em 1em 2em 1em;
  -webkit-border-radius:  1em 4em 1em 4em
    / 2em 1em 2em 1em;}
p.three {
  padding: 0px;
  border-radius: 100px;
  -moz-border-radius: 100px;
  -webkit-border-radius: 100px;}
```

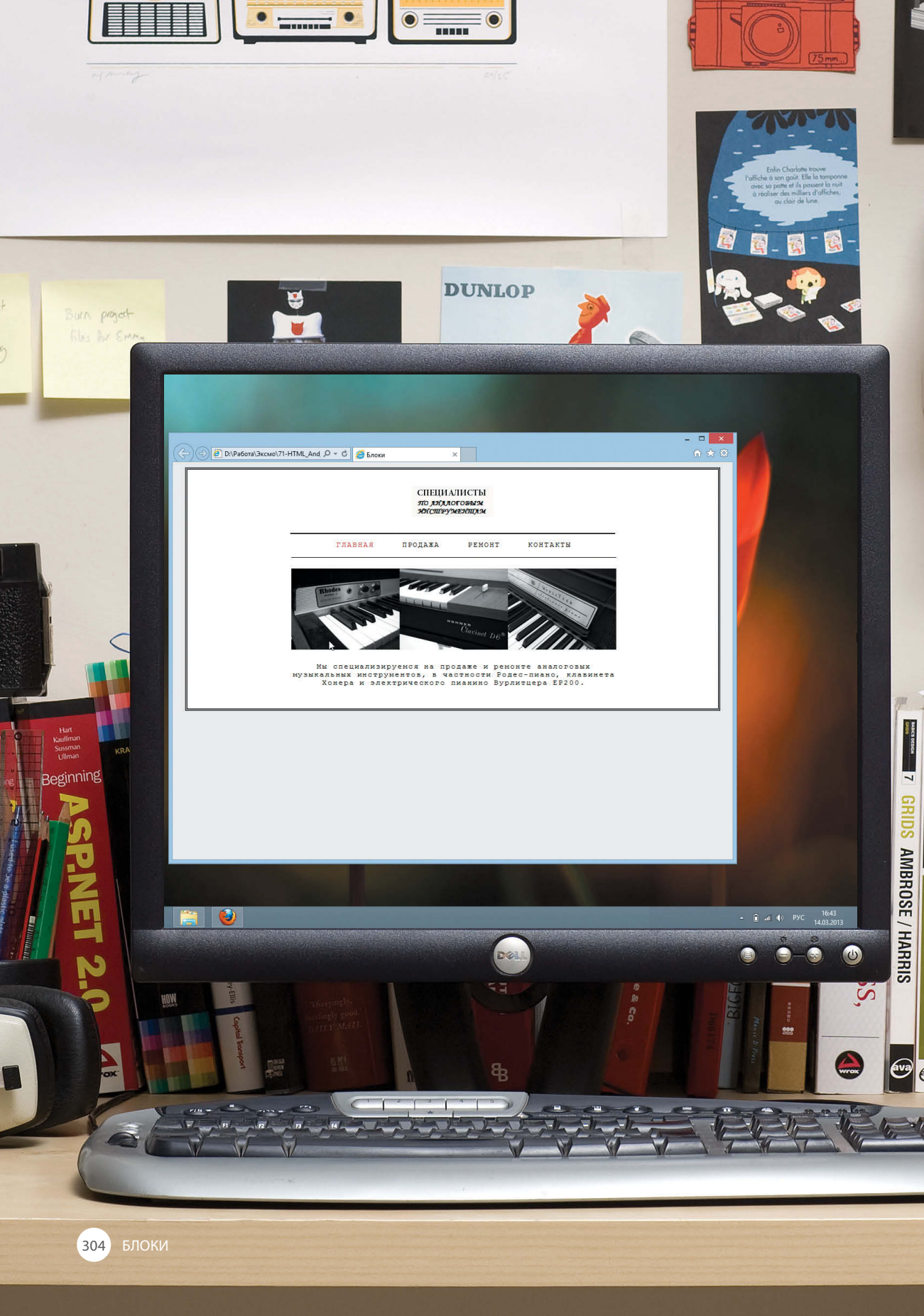
РЕЗУЛЬТАТ



Для создания более сложных форм вы можете задать различные расстояния для горизонтальных и вертикальных частей округлых углов.
Например данное выражение создаст закругление, ширина которого будет больше высоты:
`border-radius: 80px 50px;`



Вы можете настроить каждый угол в отдельности, используя разные свойства:
`border-top-left-radius: 80px 50px;`
Для одновременной настройки всех углов также существует стенографическое свойство: сначала вы указываете четыре горизонтальных значения, а затем четыре вертикальных (см. вторую форму на рисунке слева).
Вы также можете создать круг, взяв за основу квадрат и задав его высоту в качестве значения свойства `border-radius` (см. третью форму на рисунке слева).


D:\Работа\Эксмо\71-HTML_And
Блоки
СПЕЦИАЛИСТЫ
ПО АНАЛОГОВЫМ
МУЗЫКАЛЬНЫМ ИНСТРУМЕНТАМ
ГЛАВНАЯ ПРОДАЖА РЕМОНТ КОНТАКТЫ
Мы специализируемся на продаже и ремонте аналоговых музыкальных инструментов, в частности Родес-пиано, клавинета Хонера и электрического пианино Вурлитцера EP200.
DUNLOP
Beginning
ASP.NET 2.0
GRIDS AMBROSE / HARRIS
РУС
16:43
14.03.2013

ПРИМЕР
БЛОКИ

В данном примере представлена простая главная страница музыкального магазина.

Вся страница находится внутри элемента `<div>` с идентификатором `page`. Этот элемент отцентрирован с помощью свойства `margin` со значениями `auto` для правого и левого полей. Логотип и другой контент отцентрированы тем же способом.

Основной элемент `<div>` имеет границы в виде двойной линии, а его размер будет меняться вместе с размером окна браузера. Чтобы не позволить странице стать слишком узкой или широкой, используются свойства `min-width` и `max-width`.

Меню навигации создано с помощью неупорядоченного (маркированного) списка. Сверху и снизу от меню добавлены границы, чтобы выделить его. К каждому элементу списка применено свойство `display`, потому они ведут себя как встроенные (а не как блочные). Это приводит к тому, что навигационные ссылки располагаются друг рядом с другом. Для создания интервала между ссылками использовано свойство `padding`.

Значение свойства `width` для элемента `<ul>` равно 570 пикселам, а для элементов `<p>` — 600 пикселам.

В конечном итоге они имеют одинаковую ширину, поскольку элемент `<ul>` также предусматривает отступ для создания интервала между границами блока и ссылками внутри него, а любые отступы, границы и поля добавляются к ширине и высоте блока.

Если бы мы не указали тег `DOCTYPE` в начале этой страницы, то размеры блока в браузере Internet Explorer 6 были бы другими, поскольку он не корректно реализует блочную модель.

ПРИМЕР
БЛОКИ

```
<!DOCTYPE html>
<html>
  <head>
    <title>Блоки</title>
    <style type="text/css">
      body {
        font-size: 80%;
        font-family: "Courier New", Courier, monospace;
        letter-spacing: 0.15em;
        background-color: #efefef;}
      #page {
        max-width: 940px;
        min-width: 720px;
        margin: 10px auto 10px auto;
        padding: 20px;
        border: 4px double #000;
        background-color: #ffffff;}
      #logo {
        width: 150px;
        margin: 10px auto 25px auto;}
      ul {
        width: 570px;
        padding: 15px;
        margin: 0px auto 0px auto;
        border-top: 2px solid #000;
        border-bottom: 1px solid #000;
        text-align: center;}
      li {
        display: inline;
        margin: 0px 3px;}
      p {
        text-align: center;
        width: 600px;
        margin: 20px auto 20px auto;
        font-weight: normal;}
```

ПРИМЕР
БЛОКИ

```
      a {
        color: #000000;
        text-transform: uppercase;
        text-decoration: none;
        padding: 6px 18px 5px 18px;}
      a:hover, a.on {
        color: #cc3333;
        background-color: #ffffff;}
    </style>
  </head>
  <body>
    <div id="page">
      <div id="logo">
        <img src="images/logo.gif" alt="Специалисты по аналоговым инструментам" />
      </div>
      <ul id="navigation">
        <li><a href="#" class="on">Главная</a></li>
        <li><a href="#">Продажа</a></li>
        <li><a href="#">Ремонт</a></li>
        <li><a href="#">Контакты</a></li>
      </ul>
      <p>
        <img src="images/keys.jpg" alt="Родес-пиано, клавинет Хонера
            и электрическое пианино Вурлитцера EP200" />
      </p>
      <p>
        Мы специализируемся на продаже и ремонте аналоговых музыкальных
        инструментов, в частности Родес-пиано, клавинета Хонера и электрического
        пианино Вурлитцера EP200.
      </p>
    </div>
  </body>
</html>
```

ОБЗОР
БЛОКИ

- Правила CSS обращаются с каждым HTML-элементом так, будто он заключен в блок.
- С помощью CSS вы можете управлять размерами блока.
- С помощью CSS вы также можете настраивать границы, поля и отступы каждого блока.
- Вы можете скрывать элементы, используя свойства display и visibility.
- Блочные элементы превращаются во встроенные, а встроенные — в блочные.
- Удобочитаемость текста можно повысить, настроив междустрочный интервал, а также ширину блока, в котором находится текст.
- В спецификации CSS3 появилась возможность создавать декоративные рамки и границы с закругленными углами.

14

СПИСКИ, ТАБЛИЦЫ И ФОРМЫ

- Создание маркеров
- Добавление границ и фона к таблицам
- Изменение внешнего вида элементов формы

В CSS есть несколько свойств, предназначенных для работы с такими HTML-элементами, как списки, таблицы и формы.

В этой главе вы узнаете, как:

- указывать тип маркера или нумерации в списках;
- назначать границы и фон ячейкам таблицы;
- настраивать внешний вид элементов формы.

Эти свойства обеспечивают более точный контроль над определенными частями страницы.

Finder Файл Правка Вид Переход Окно Справка
Чт. 0:15
Списки, таблицы и формы
file:///Volumes/VMware%20Shared%20Folders/Shared/CD/Final%20Code/глава-14/example.html
Поэтические чтения
Мы проведем ряд встреч и симпозиумов в течение года.
Вход на следующие мероприятия для участников свободный:
Поэтическая перспектива
Вечер поэзии в честь Д.Р. Пушкина А.С
Поэзия и поэты серебряного века
Москва
Санкт-Петербург
Екатеринбург
Поэтическая перспектива
СБ, 4 февраля 2012 11:00 - 14:00
СБ, 3 марта 2012 11:00 - 14:00
СБ, 17 марта 2012 11:00 - 14:00
Вечер поэзии в честь Д.Р. Пушкина А.С
СБ, 7 апреля 2012 11:00 - 13:00
СБ, 5 мая 2012 11:00 - 13:00
СБ, 19 мая 2012 11:00 - 13:00
Поэзия и поэты серебряного века
СБ, 9 июня 2012 11:00 - 14:00
СБ, 7 июля 2012 11:00 - 14:00
СБ, 21 июля 2012 11:00 - 14:00
Поэты 21 века
СБ, 4 августа 2012 11:00 - 16:00
СБ, 8 сентября 2012 11:00 - 16:00
СБ, 15 сентября 2012 11:00 - 16:00
ЗарегистрируйтесЬ
Ваше имя:
Ваш email:
Ближайший центр:
Москва
Являетесь участником? Да Нет

МАРКЕРЫ

list-style-type

Свойство `list-style-type` позволяет выбрать форму или стиль маркера. Данное свойство может использоваться в правилах, применяемых к элементам `<ol>`, `<ul>` и `<li>`.

НЕУПОРЯДОЧЕННЫЕ СПИСКИ

Для неупорядоченных (маркированных) списков вы можете использовать следующие значения:

- **none**
- **disc**
- **circle**
- **square**

УПОРЯДОЧЕННЫЕ СПИСКИ

Для упорядоченных (нумерованных) списков вы можете использовать следующие значения:

decimal
1 2 3

decimal-leading-zero
01 02 03

lower-alpha
a b c

upper-alpha
A B C

lower-roman
i. ii. iii.

upper-roman
I II III

глава-14/list-style-type.html HTML

```
<h1>Поэмы</h1>
<h2>Эмили Дикинсон</h2>
<ol>
  <li>Жизнь</li>
  <li>Природа</li>
  <li>Любовь</li>
  <li>Время и вечность</li>
  <li>Одинокий пес</li>
</ol>
```

CSS

```
ol {
  list-style-type: lower-roman;}
```

РЕЗУЛЬТАТ

Поэмы

Эмили Дикинсон

i. Жизнь
ii. Природа
iii. Любовь
iv. Время и вечность
v. Одинокий пес

МАРКЕРЫ-ИЗОБРАЖЕНИЯ

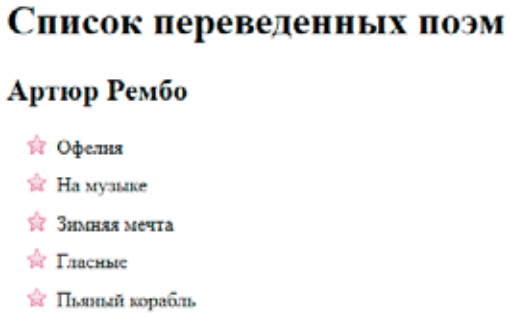
HTML **глава-14/list-style-image.html**

```
<h1>Список переведенных поэм</h1>
<h2>Артюр Рембо</h2>
<ul>
  <li>Офелия</li>
  <li>На музыке</li>
  <li>Зимняя мечта</li>
  <li>Гласные</li>
  <li>Пьяный корабль</li>
</ul>
```

CSS

```
ul {
  list-style-image: url("images/star.png");}
li {
  margin: 10px 0px 0px 0px;}
```

РЕЗУЛЬТАТ



С помощью свойства `list-style-image` вы можете указать в качестве маркера изображение.

Значение начинается с фрагмента `url`, за которым следует пара скобок. Внутри скобок в двойных кавычках указывается путь к изображению.

Данное свойство может использоваться в правилах, применяемых к элементам `<ul>` и `<li>`.

В приведенном примере также используется свойство `margin` для увеличения вертикального интервала между элементами списка.

ПОЗИЦИОНИРОВАНИЕ МАРКЕРА

list-style-position

Для списков по умолчанию создаются отступы, а свойство `list-style-position` указывает, должен ли маркер располагаться внутри или снаружи блока, содержащего элементы списка.
Данное свойство может принимать одно из двух значений.

outside

При этом маркер располагается слева от текстового блока (поведение по умолчанию, если данное свойство не используется).

inside

Маркер располагается внутри блока с текстом, для которого создан отступ.

В приведенном примере ширина списка была ограничена 150 пикселями. Это гарантирует перенос текста на новую строку, чтобы вы могли увидеть, как значение `inside` помещает маркер внутри первой строки текста.
Для создания интервала между элементами списка к каждому из них было добавлено поле.

глава-14/list-style-position.html HTML

```
<ul class="illuminations">
  <li>С желтою гривой и глазами черного цвета, без
    родных и двора, этот идол...</li>
  <li>Изящный сын Пана! Твоя голова, увенчанная
    цветами...</li>
  <li>Есть птица в лесу, чье пение вас
    останавливает...</li>
</ul>
<ul class="season">
  <li>О огромные улицы священной страны и террасы
    храма...</li>
  <li>Я - изобретатель, достойный совсем иной
    похвалы...</li>
  <li>Довольно того, что узрел. Виденья встречались
    повсюду...</li>
</ul>
```

CSS

```
ul {
  width: 150px;}
li {
  margin: 10px;}
ul.illuminations {
  list-style-position: outside;}
ul.season {
  list-style-position: inside;}
```

РЕЗУЛЬТАТ

Озарения

- С желтою гривой и глазами черного цвета, без родных и двора, этот идол...
- Изящный сын Пана! Твоя голова, увенченная цветами...
- Есть птица в лесу, чье пение вас останавливает...

РЕЗУЛЬТАТ

Одно лето в аду

- О огромные улицы священной страны и террасы храма...
- Я - изобретатель, достойный совсем иной похвалы...
- Довольно того, что узрел. Виденья встречались повсюду...

СПИСКИ. СТЕНОГРАФИЧЕСКОЕ СВОЙСТВО list-style

HTML **глава-14/list-style.html**

```
<h1>Цитаты Эдгара Аллана По</h1>
<ul>
  <li>Никакой транспорт не будет попутным, если не
    знаешь, куда идти.</li>
  <li>А ведь именно отклонение от простого
    и обычного освещает дорогу разуму в поисках
    истины.</li>
  <li>В глубокомыслии легко перемудрить. Истина не
    всегда обитает на дне колодца.</li>
</ul>
```

CSS

```
ul {
  list-style: inside circle;
  width: 300px;}
li {
  margin: 10px 0px 0px 0px;}
```

РЕЗУЛЬТАТ

Так же, как и в случае с некоторыми другими свойствами CSS, существует стенографическое свойство для оформления списков — `list-style`. Оно позволяет в любом порядке указать стиль маркеров, изображение и позиционирование.

СВОЙСТВА ТАБЛИЦЫ

Вы уже познакомились с некоторыми свойствами, которые часто используются с таблицами. Теперь мы соберем их все в одном примере:

`width` — устанавливает ширину таблицы;

`padding` — определяет расстояние между границей каждой ячейки и ее содержимым;

`text-transform` — используется для переключения регистра заголовков таблицы на верхний;

`letter-spacing, font-size` — добавляют дополнительные стили к заголовкам таблицы;

`border-top, border-bottom` — добавляют границы над и под заголовками таблицы;

`text-align` — выравнивает текст в ячейках по левому и по правому краю;

`background-color` — заменяет фоновый цвет чередующихся строк таблицы;

`:hover` — выделяет строку таблицы при наведении на нее указателя мыши;

глава-14/table-properties.html HTML

```
<h1>Книжный аукцион</h1>
<table>
  <tr>
    <th>Автор</th>
    <th>Название</th>
    <th class="money">Зарезервированная цена</th>
    <th class="money">Текущая ставка</th>
  </tr>
  <tr>
    <td>Эдвард Эстлин Каммингс</td>
    <td>Тюльпаны и дымоходы</td>
    <td class="money">60 000 рублей</td>
    <td class="money">78 500 рублей</td>
  </tr>
  <tr class="even">
    <td>Карл I Орлеанский</td>
    <td>Поэмы</td>
    <td class="money"></td>
    <td class="money">162 000 рублей</td>
  </tr>
  <tr>
    <td>Томас Стернз Элиот</td>
    <td>Поэмы 1909-1925</td>
    <td class="money">32 700 рублей</td>
    <td class="money">253 500 рублей</td>
  </tr>
  <tr class="even">
    <td>Сильвия-Плат</td>
    <td>Колосс</td>
    <td class="money"></td>
    <td class="money">30 300 рублей</td>
    </tr>
</table>
```

CSS

```
body {
  font-family: Arial, Verdana, sans-serif;
  color: #111111;}
table {
  width: 600px;}
th, td {
  padding: 7px 10px 10px 10px;}
th {
  text-transform: uppercase;
  letter-spacing: 0.1em;
  font-size: 90%;
  border-bottom: 2px solid #111111;
  border-top: 1px solid #999;
  text-align: left;}
tr.even {
  background-color: #efefef;}
tr:hover {
  background-color: #c3e6e5;}
.money {
  text-align: right;}
```

РЕЗУЛЬТАТ

Книжный аукцион

АВТОР	НАЗВАНИЕ	ЗАРЕЗЕРВИРОВАННАЯ ЦЕНА	ТЕКУЩАЯ СТАВКА
Эдвард Эстлин Каммингс	Тюльпаны и дымоходы	60 000 рублей	78 500 рублей
Карл I Орлеанский	Поэмы		162 000 рублей
Томас Стернз Элиот	Поэмы 1909-1925	32 700 рублей	253 500 рублей
Сильвия Плат	Колосс		30 300 рублей

Ниже приведены несколько советов, позволяющих гарантировать удобное восприятие таблиц.

СОЗДАВАЙТЕ ОТСТУПЫ В ЯЧЕЙКАХ

Если текст в ячейках таблицы касается границ (или другой ячейки), то таблицу гораздо труднее воспринимать. Добавление отступа улучшает удобочитаемость.

ВЫДЕЛЯЙТЕ ЗАГОЛОВКИ

Выделение заголовков полужирным начертанием шрифта (стиль по умолчанию для элементов `<th>`) улучшает читабельность. Вы также можете использовать верхний регистр для текста заголовков и добавить фоновый цвет или подчеркивание.

ОТТЕНЯЙТЕ ЧЕРЕДУЮЩИЕСЯ СТРОКИ

Назначение оттенка каждой второй строке способно помочь пользователям следовать по строкам. Используйте небольшое отличие от обычного цвета строк таблицы, чтобы она выглядела опрятно.

ВЫРАВНИВАЙТЕ НОМЕРА

Вы можете использовать свойство `text-align`, чтобы выровнять содержимое каждого столбца, содержащего номера, по правому краю так, чтобы крупные цифры отличались от более мелких.

ГРАНИЦЫ ПУСТЫХ ЯЧЕЕК
empty-cells

Если в вашей таблице есть пустые ячейки, то вы можете использовать свойство `empty-cells`, чтобы указать, следует ли отображать их границы.
Поскольку браузеры по-разному обращаются с пустыми ячейками, если вы хотите явно указать, что границы должны быть показаны или скрыты, то вам следует использовать данное свойство. Оно может принимать одно из трех значений.

show
При использовании этого значения границы пустых ячеек отображаются.

hide
При использовании этого значения границы пустых ячеек скрываются.

inherit
Если одна таблица вложена в другую, то данное значение приводит к тому, что к ячейкам вложенной таблицы применяются правила содержащей их таблицы.

В приведенном примере видно, что границы пустой ячейки в первой таблице отображаются, а во второй — нет.

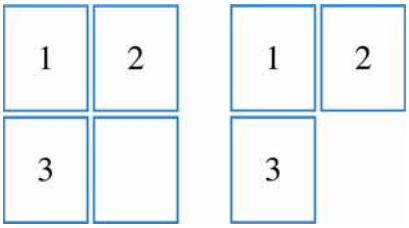
глава-14/empty-cells.html HTML

```
<table class="one">
  <tr>
    <td>1</td>
    <td>2</td>
  </tr>
  <tr>
    <td>3</td>
    <td></td>
  </tr>
</table>
```

CSS

```
td {
  border: 1px solid #0088dd;
  padding: 15px;}
table.one {
  empty-cells: show;}
table.two {
  empty-cells: hide;}
```

РЕЗУЛЬТАТ



ИНТЕРВАЛЫ МЕЖДУ ЯЧЕЙКАМИ

border-spacing, border-collapse

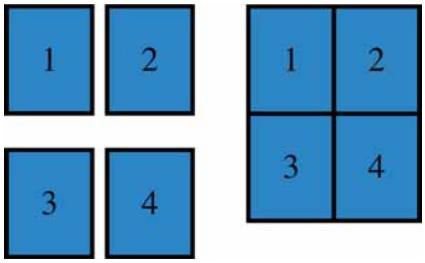
HTML **глава-14/gaps-between-cells.html**

```
<table class="one">
  <tr>
    <td>1</td>
    <td>2</td>
  </tr>
  <tr>
    <td>3</td>
    <td>4</td>
  </tr>
</table>
```

CSS

```
td {
  background-color: #0088dd;
  padding: 15px;
  border: 2px solid #000000;}
table.one {
  border-spacing: 5px 15px;}
table.two {
  border-collapse: collapse;}
```

РЕЗУЛЬТАТ



Свойство `border-spacing` позволяет настраивать расстояние между соседними ячейками. По умолчанию браузеры обычно оставляют небольшой интервал между ячейками таблицы, и, чтобы увеличить или уменьшить его, вы можете использовать свойство `border-spacing`. Значение этого свойства обычно задается в пикселах. Вы можете указать два значения, если хотите отдельно настроить горизонтальные и вертикальные интервалы.

Если для ячеек таблицы задана граница, то в местах соприкосновения ячеек линия будет в два раза толще, чем по краям таблицы. Чтобы предотвратить это, следует использовать свойство `border-collapse`. Возможные значения его перечислены ниже.

collapse

При использовании данного значения границы сливаются в одну (ячейки будут сдвинуты, а свойства `border-spacing` и `empty-cells` — проигнорированы).

separate

Границы отделяются друг от друга. (Свойства `border-spacing` и `empty-cells` выполняются.)

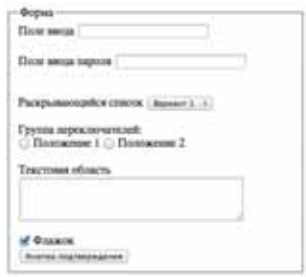
СТИЛИЗАЦИЯ ФОРМ

Нет никого, кому бы нравилось заполнять формы, поэтому если вы сделаете свои формы более удобными, то люди с большей вероятностью будут отправлять вам информацию. Имейте в виду, что браузеры по-разному отображают формы.
Язык CSS часто используется для управления внешним видом элементов формы, позволяя сделать их более привлекательными и поддерживая соответствие в различных браузерах.
Часто стили применяются:

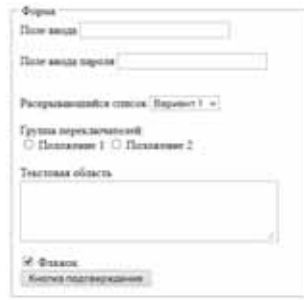
- к полям ввода и текстовым областям;
- к кнопкам подтверждения;
- к меткам элементов формы для выравнивания.

В данном разделе вы увидите, как все это можно осуществить с помощью CSS.

OS X: SAFARI

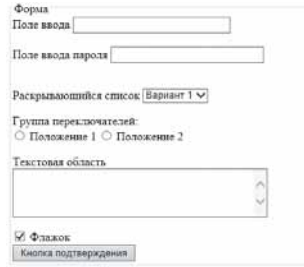


OS X: FIREFOX



WINDOWS: CHROME

WINDOWS: INTERNET EXPLORER



Применить стили к полям ввода текста и кнопкам подтверждения довольно просто. Гораздо сложнее заставить переключатели и флажки выглядеть одинаково во всех браузерах.

Чтобы этого достичь, вы можете воспользоваться CSS-файлами, доступными для загрузки на сайте **formalize.me**. Автор данного сайта проделал большую работу, чтобы заставить формы выглядеть одинаково в различных браузерах. Хотя в файлах и встречается язык JavaScript, чтобы их использовать, вам не потребуется знания этой технологии.

СТИЛИЗАЦИЯ ПОЛЕЙ ВВОДА

CSS **глава-14/styling-text-inputs.html**

```
input {
  font-size: 120%;
  color: #5a5854;
  background-color: #f2f2f2;
  border: 1px solid #bdbdbd;
  border-radius: 5px;
  padding: 5px 5px 5px 30px;
  background-repeat: no-repeat;
  background-position: 8px 9px;
  display: block;
  margin-bottom: 10px;}
input:focus {
  background-color: #ffffff;
  border: 1px solid #b1e1e4;}
input#email {
  background-image: url("images/email.png");}
input#twitter {
  background-image: url("images/twitter.png");}
input#web {
  background-image: url("images/web.png");}
```

РЕЗУЛЬТАТ

Этот пример демонстрирует свойства CSS, которые широко применяются по отношению к полям ввода текста. Большинство из них вам уже знакомо.

`font-size` — определяет размер текста, введенного пользователем;

`color` — задает цвет текста;

`background-color` — определяет цвет фона для введенного текста;

`border` — добавляет границы вокруг поля ввода;

`border-radius` — может использоваться для создания закругленных углов (в браузерах, которые поддерживают данное свойство);

`:focus` используется для изменения цвета фона введенного текста;

`:hover` применяет стили при наведении указателя мыши;

`background-image` — добавляет в поле фоновое изображение (поскольку в каждом поле ввода есть свое изображение, мы используем селектор атрибута в поисках значения атрибута `id` для каждого поля ввода).

Вы узнаете больше о фоновых изображениях и о том, как их располагать, в главе 16.

СТИЛИЗАЦИЯ КНОПОК ПОДТВЕРЖДЕНИЯ

Ниже приведены некоторые свойства, которые можно использовать для применения стилей к кнопкам подтверждения. Этот пример основан на предыдущем, а кнопка подтверждения наследует набор стилей элемента `<input>` с предыдущей страницы.

`color` — используется для изменения цвета текста на кнопке;
`text-shadow` — придает трехмерный вид тексту в браузерах, которые поддерживают данное свойство;
`border-bottom` — увеличивает толщину нижней границы кнопки, что создает трехмерный эффект;
`background-color` — позволяет выделить кнопку среди остальных элементов (создание единого стиля для всех кнопок помогает пользователям понять, как они должны взаимодействовать с сайтом);
для браузеров, которые поддерживают градиенты, добавлен градиентный фон (о градиентах речь пойдет в главе 16).
`:hover` использован для изменения внешнего вида кнопки при наведении на нее указателя мыши (в данном случае изменяется фон, цвет текста становится темнее, а верхняя граница кнопки — толще).

глава-14/styling-submit-buttons.html CSS

```
input#submit {
  color: #444444;
  text-shadow: 0px 1px 1px #ffffff;
  border-bottom: 2px solid #b2b2b2;
  background-color: #b9e4e3;
  background: -webkit-gradient(linear, left top,
    left bottom, from(#beeae9), to(#a8cfce));
  background:
    -moz-linear-gradient(top, #beeae9, #a8cfce);
  background:
    -o-linear-gradient(top, #beeae9, #a8cfce);
  background:
    -ms-linear-gradient(top, #beeae9, #a8cfce);}
input#submit:hover {
  color: #333333;
  border: 1px solid #a4a4a4;
  border-top: 2px solid #b2b2b2;
  background-color: #a0dbc4;
  background: -webkit-gradient(linear, left top,
    left bottom, from(#a8cfce), to(#beeae9));
  background:
    -moz-linear-gradient(top, #a8cfce, #beeae9);
  background:
    -o-linear-gradient(top, #a8cfce, #beeae9);
  background:
    -ms-linear-gradient(top, #a8cfce, #beeae9);}
```

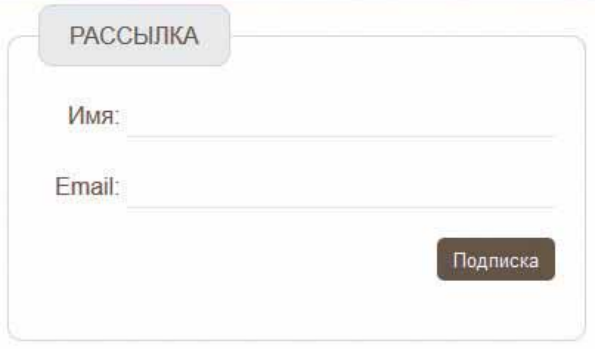
РЕЗУЛЬТАТ

СТИЛИЗАЦИЯ ЭЛЕМЕНТОВ FIELDSET И LEGEND

CSS **глава-14/styling-fieldsets-and-legends.html**

```
fieldset {
  width: 350px;
  border: 1px solid #dcdcdc;
  border-radius: 10px;
  padding: 20px;
  text-align: right;}
legend {
  background-color: #efefef;
  border: 1px solid #dcdcdc;
  border-radius: 10px;
  padding: 10px 20px;
  text-align: left;
  text-transform: uppercase;}
```

РЕЗУЛЬТАТ



Элемент `fieldset` особенно полезен при определении границ формы. В случае с длинной формой с помощью него вы можете группировать связанные данные.
Элемент `legend` используется для добавления к группе элементов информации. Обычно с этими двумя элементами используются следующие свойства:

`width` — используется для настройки ширины полей; в данном примере ширина группы элементов заставляет последние переходить на новую строку (если бы значение было больше, то эти элементы могли уместиться на одной строке);
`color` — используется для настройки цвета текста;
`background-color` — используется для изменения цвета позади элементов;
`border` — используется для настройки внешнего вида границы вокруг группы элементов и/или надписи;
`border-radius` — используется для создания закругленных углов в браузерах, которые поддерживает данное свойство;
`padding` — может использоваться для добавления пространства внутри элементов.

ВЫРАВНИВАНИЕ ЭЛЕМЕНТОВ ФОРМЫ: ПРОБЛЕМА

Метки элементов формы часто имеют различную длину, что препятствует выравниванию элементов по одной линии. Это демонстрируется в приведенном примере (без применения к элементам формы таблицы стилей). В данной форме каждый запрашиваемый у пользователя тип данных помещается внутри элемента `<div>`, чтобы вопросы отображались на отдельных строках. Пользователю проще заполнять форму, когда все ее элементы выровнены по одной вертикальной линии. CSS-код на следующей странице решает эту задачу.

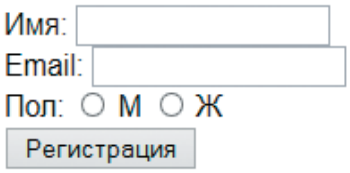
Если вы взглянете на фрагмент формы, где пользователь должен указать свой пол, то увидите два переключателя, каждый со своим элементом `<label>` (один для мужчин, другой для женщин). Элемент `<span>`, добавленный к названию, помогает выровнять эти элементы формы.

На предыдущей странице мы видели другой способ выравнивания элементов. Когда форма содержит только поля ввода, задание для них одинаковой ширины и выравнивание содержимого формы по правому краю приводит к выстраиванию полей в одну линию, а метки при этом располагаются в одних и тех же местах. Для более сложных форм необходимо решение, подобное приведенному ниже.

глава-14/aligning-form-controls-problem.html HTML

```
<form action="example.php" method="post">
  <div>
    <label for="name" class="title">Имя:</label>
    <input type="text" id="name" name="name" />
  </div>
  <div>
    <label for="email" class="title">Email:</label>
    <input type="email" id="email" name="email" />
  </div>
  <div>
    <span class="title">Пол:</span>
    <input type="radio" name="gender" id="male"
      value="M" />
    <label for="male">M</label>
    <input type="radio" name="gender" id="female"
      value="F" />
    <label for="female">Ж</label><br/>
  </div>
  <div>
    <input type="submit" value="Регистрация"
      id="submit" />
  </div>
</form>
```

РЕЗУЛЬТАТ БЕЗ CSS

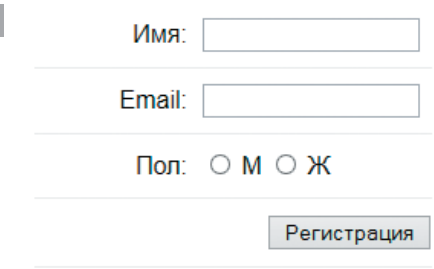


ВЫРАВНИВАНИЕ ЭЛЕМЕНТОВ ФОРМЫ: РЕШЕНИЕ

CSS **глава-14/aligning-form-controls-solution.html**

```
div {
  border-bottom: 1px solid #efefef;
  margin: 10px;
  padding-bottom: 10px;
  width: 260px;}
.title {
  float: left;
  width: 100px;
  text-align: right;
  padding-right: 10px;}
.radio-buttons label {
  float: none;}
.submit {
  text-align: right;}
```

РЕЗУЛЬТАТ



У каждой строки формы есть название, сообщающее пользователям, что именно им нужно ввести. Название для полей ввода располагается в элементе `<label>`. Для переключателей название указано в элементе `<span>`. Оба имеют атрибут `class` со значением `title`. Мы можем использовать свойство `float`, чтобы переместить названия в левую часть страницы (оно рассматривается подробнее в главе 15).

Применив к этим элементам свойство `width`, мы добьемся того, что названия будут иметь одинаковую ширину, благодаря чему элементы управления формы выровняются.

Свойство `text-align` используется для выравнивания названий по правому краю, а свойство `padding` гарантирует наличие интервала между текстом в полях и элементами формы.

Стили также применяются к элементам `<div>`, содержащим каждую строку формы (эти элементы фиксируют ширину строк и создают вертикальный интервал между ними). Кнопка подтверждения также выровнена по правому краю.

СТИЛИ УКАЗАТЕЛЕЙ МЫШИ

cursor

Свойство `cursor` позволяет выбрать вид указателя мыши, который должен быть отображен для пользователя.
Например вы можете сделать так, чтобы при наведении на какой-либо элемент указатель мыши принимал форму ладони.
Ниже приведены наиболее часто используемые значения для данного свойства:

`auto`;
`crosshair`;
`default`;
`pointer`;
`move`;
`text`;
`wait`;
`help`;
`url("cursor.gif")`.

Вы должны использовать эти значения, только чтобы добавлять полезную информацию для пользователей в тех местах, где они ожидают увидеть подобные указатели мыши. (Например появление указателя мыши в виде перекрестия при наведении на ссылку может запутать пользователей, поскольку они не привыкли видеть данный эффект в подобной ситуации.)

глава-14/cursor.html HTML

```
<a href="http://www.whitmanarchive.org">
  Александр Сергеевич Пушкин</a>
```

CSS

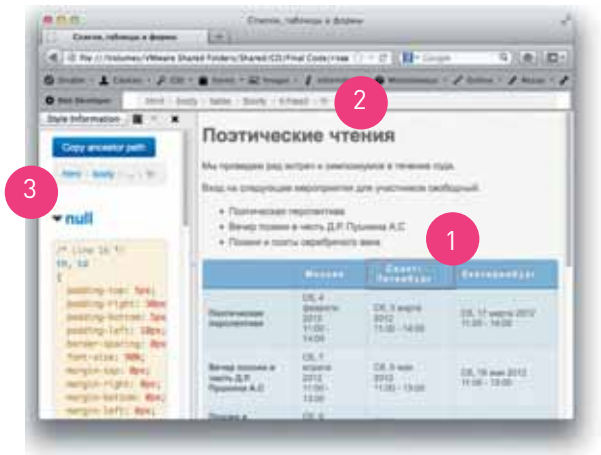
```
a {
  cursor: move;}
```

РЕЗУЛЬТАТ

ПАНЕЛЬ ИНСТРУМЕНТОВ ВЕБ-РАЗРАБОТЧИКА

Это полезное расширение для браузеров Firefox и Chrome позволяет увидеть стили CSS, примененные к элементу, при наведении на них указателя мыши, а также структуру HTML.

Загрузите этот инструмент с сайта **www.chrispederick.com/work/web-developer** Чтобы увидеть стили CSS и структуру HTML веб-страницы, выберите на появившийся панели команду меню **CSS** ⇒ Display Style Information



1: КОНТУРЫ

При наведении на элемент указателя мыши вокруг него появится красный контур, указывающий на область, которую занимает данный элемент.

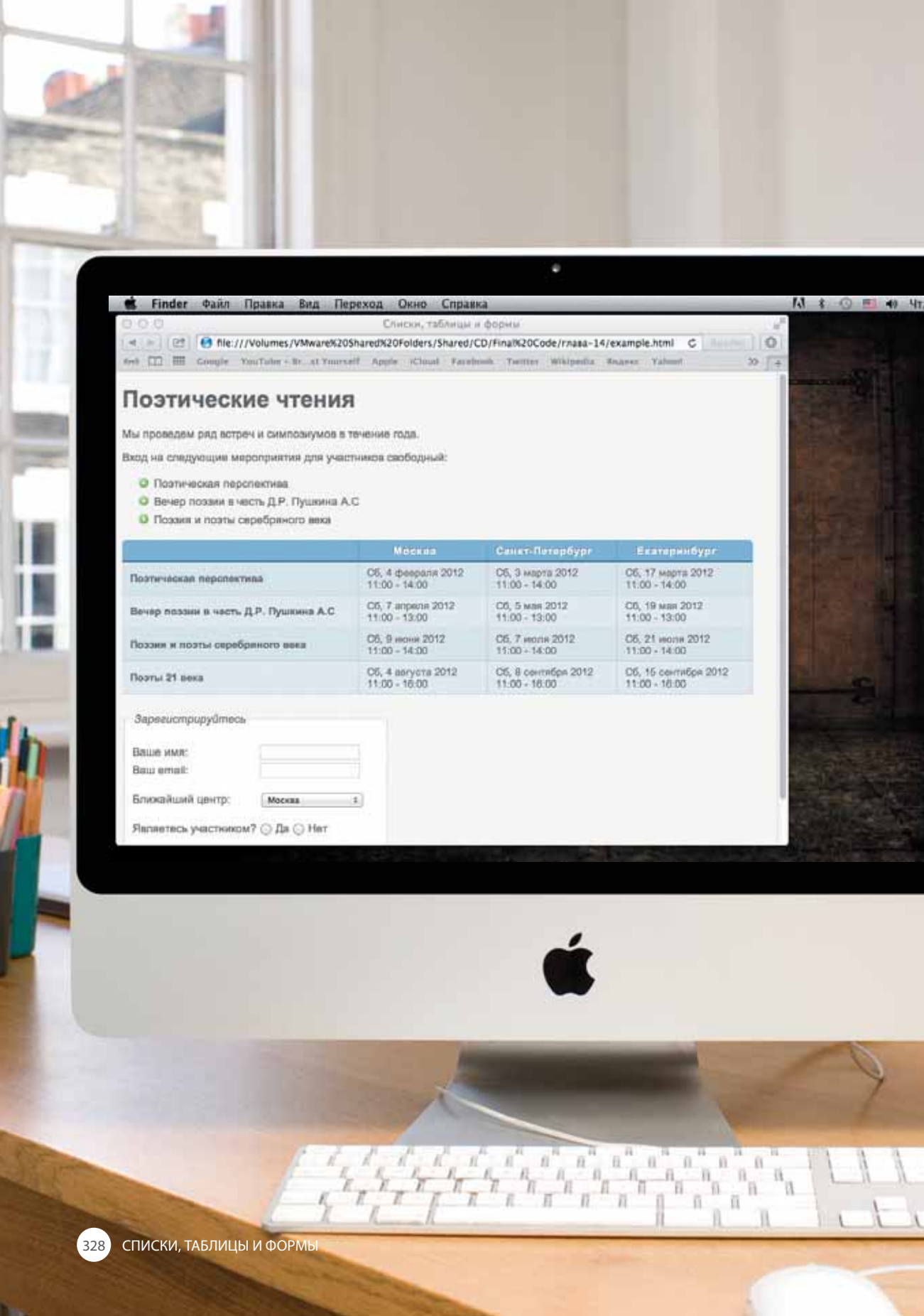
2: СТРУКТУРА

Пока указатель мыши наведен на элемент, в верхней части окна отображается структура. Здесь вы видите, что элемент `<th>` находится внутри элемента `<tr>` с классом `head`. Все это, в свою очередь, расположено внутри элемента `<tbody>` и внутри таблицы `<table>`, которая находится внутри элементов `<body>` и `<html>`.
Это может быть полезным при написании селекторов CSS, поскольку помогает указать целью правильный элемент.

3: СТИЛИ CSS

Щелкните мышью по элементу, чтобы отобразить CSS-код. Вы увидите правила, примененные к элементу, а также номера строк, в которых они указаны. Над правилами вы увидите имя файла таблицы стилей и путь к ней (если используется внешняя таблица CSS).
Это позволяет проверить, какие стили применены к элементу. Вы можете использовать данную функцию при анализе кода страниц собственного сайта или чтобы увидеть, какие стили используют другие разработчики.

Этот инструмент также позволяет изменить размер экрана, выполнить валидацию HTML- и CSS-кода, а также отключить отображение изображений.


Finder Файл Правка Вид Переход Окно Справка
Списки, таблицы и формы
file:///Volumes/VMware%20Shared%20Folders/Shared/CD/Final%20Code/глава-14/example.html
Поэтические чтения
Мы проведем ряд встреч и симпозиумов в течение года.
Вход на следующие мероприятия для участников свободный:
Поэтическая перспектива
Вечер поэзии в честь Д.Р. Пушкина А.С
Поэзия и поэты серебряного века
Москва
Санкт-Петербург
Екатеринбург
Поэтическая перспектива
Сб, 4 февраля 2012 11:00 - 14:00
Сб, 3 марта 2012 11:00 - 14:00
Сб, 17 марта 2012 11:00 - 14:00
Вечер поэзии в честь Д.Р. Пушкина А.С
Сб, 7 апреля 2012 11:00 - 13:00
Сб, 5 мая 2012 11:00 - 13:00
Сб, 19 мая 2012 11:00 - 13:00
Поэзия и поэты серебряного века
Сб, 9 июня 2012 11:00 - 14:00
Сб, 7 июля 2012 11:00 - 14:00
Сб, 21 июля 2012 11:00 - 14:00
Поэты 21 века
Сб, 4 августа 2012 11:00 - 16:00
Сб, 8 сентября 2012 11:00 - 16:00
Сб, 15 сентября 2012 11:00 - 16:00
Зарегистрируйтесь
Ваше имя:
Ваш email:
Ближайший центр:
Москва
Являетесь участником? Да Нет

ПРИМЕР
СПИСКИ, ТАБЛИЦЫ И ФОРМЫ

Этот пример демонстрирует несколько свойств CSS, с которыми вы познакомились в данной главе. Они позволяют управлять представлением списков, таблиц и форм.

Для списка мероприятий в начале страницы использовались маркеры-изображения. Интервал между элементами списка был увеличен с помощью свойства `line-height`. Интервалы между ячейками таблицы мы удалили с помощью свойства `border-spacing`. Для элементов `<td>` и `<th>` был установлен размер шрифта, поскольку они не наследуют размер от родительских элементов.
Заголовок таблицы имеет более темный фон, более светлый цвет текста и темную линию шириной в 2 пиксела, отделяющую его от содержимого таблицы. Закругленные углы заголовка таблицы были созданы с помощью псевдоклассов `:first-child` и `:last-child` (для браузеров, которые поддерживают закругленные углы).
Чередующиеся строки таблицы имеют разные оттенки и границы с разными текстурами с каждой стороны ячейки.
Схожие элементы формы находятся между тегами `<fieldset>` и `</fieldset>`. К меткам элементов формы также было применено свойство `float`, чтобы обеспечить вертикальное выравнивание.
Когда поля ввода формы получают фокус или когда пользователь наводит на них указатель мыши, фоновый цвет и цвет границы изменяются. К кнопке подтверждения мы также применили стили.

ПРИМЕР
СПИСКИ, ТАБЛИЦЫ И ФОРМЫ

```
<!DOCTYPE html>
<html>
  <head>
    <title>Списки, таблицы и формы</title>
    <style type="text/css">
      body {
        font-family: Arial, Verdana, sans-serif;
        font-size: 90%;
        color: #666;
        background-color: #f8f8f8;}
      li {
        list-style-image: url("images/icon-plus.png");
        line-height: 1.6em;}
      table {
        border-spacing: 0px;}
      th, td {
        padding: 5px 30px 5px 10px;
        border-spacing: 0px;
        font-size: 90%;
        margin: 0px;}
      th, td {
        text-align: left;
        background-color: #e0e9f0;
        border-top: 1px solid #f1f8fe;
        border-bottom: 1px solid #cbd2d8;
        border-right: 1px solid #cbd2d8;}
      tr.head th {
        color: #fff;
        background-color: #90b4d6;
        border-bottom: 2px solid #547ca0;
        border-right: 1px solid #749abe;
        border-top: 1px solid #90b4d6;
        text-align: center;
        text-shadow: -1px -1px 1px #666;
        letter-spacing: 0.15em;}
      td {
        text-shadow: 1px 1px 1px #fff;}
```

ПРИМЕР
СПИСКИ, ТАБЛИЦЫ И ФОРМЫ

```
tr.even td, tr.even th {
  background-color: #e8eff5;}
tr.head th:first-child {
  -webkit-border-top-left-radius: 5px;
  -moz-border-radius-topleft: 5px;
  border-top-left-radius: 5px;}
tr.head th:last-child {
  -webkit-border-top-right-radius: 5px;
  -moz-border-radius-topright: 5px;
  border-top-right-radius: 5px;}
fieldset {
  width: 310px;
  margin-top: 20px;
  border: 1px solid #d6d6d6;
  background-color: #ffffff;
  line-height: 1.6em;}
legend {
  font-style:italic;
  color:#666666;}
input[type="text"] {
  width: 120px;
  border: 1px solid #d6d6d6;
  padding: 2px;
  outline: none;}
input[type="text"]:focus,
input[type="text"]:hover {
  background-color: #d0e2f0;
  border: 1px solid #999;}
input[type="submit"] {
  border: 1px solid #006633;
  background-color: #009966;
  color: #FFFFFF;
  border-radius: 5px;
  padding: 5px;
  margin-top: 10px;}
input[type="submit"]:hover {
  border: 1px solid #006633;
```

ПРИМЕР
СПИСКИ, ТАБЛИЦЫ И ФОРМЫ

```
      background-color: #00CC33;
      color: #FFFFFF;
      cursor: pointer;}
    .title {
      float: left;
      width: 160px;
      clear: left;}
    .submit {
      width: 310px;
      text-align: right;}
  </style>
</head>
<body>
  <h1>Поэтические чтения</h1>
  <p>Мы проведем ряд встреч и симпозиумов
    в течение года.</p>
  <p>Вход на следующие мероприятия для участников
    свободный:</p>
  <ul>
    <li>Поэтическая перспектива</li>
    <li>Вечер поэзии в честь дня рождения А. С. Пушкина</li>
    <li>Поэзия и поэты Серебряного века</li>
  </ul>
  <table>
    <tr class="head">
      <th></th>
      <th>Москва</th>
      <th>Санкт-Петербург</th>
      <th>Екатеринбург</th>
    </tr>
    <tr>
      <th>Поэтическая перспектива</th>
      <td>Сб., 4 февраля 2012<br />11:00-14:00</td>
      <td>Сб., 3 марта 2012<br />11:00-14:00</td>
      <td>Сб., 17 марта 2012<br />11:00-14:00</td>
    </tr>
    <tr class="even">
      <th> Вечер поэзии в честь дня рождения
        А. С. Пушкина</th>
```

ПРИМЕР
СПИСКИ, ТАБЛИЦЫ И ФОРМЫ

```
        <td>Сб., 7 апреля 2012<br />11:00-13:00</td>
        <td>Сб., 5 мая 2012<br />11:00-13:00</td>
        <td>Сб., 19 мая 2012<br />11:00-13:00</td>
      </tr>
      <tr>
        <th>Поэзия и поэты Серебряного века</th>
        <td>Сб., 9 июня 2012<br />11:00-14:00</td>
        <td>Сб., 7 июля 2012<br />11:00-14:00</td>
        <td>Сб., 21 июля 2012<br />11:00-14:00</td>
      </tr>
      <tr class="even">
        <th>Поэты XXI века</th>
        <td>Сб., 4 августа 2012<br />11:00 - 16:00</td>
        <td>Сб., 8 сентября 2012<br />11:00 - 16:00</td>
        <td>Сб., 15 сентября 2012<br />11:00 - 16:00</td>
      </tr>
    </table>
    <form action="http://www.example.com/form.php" method="get">
      <fieldset>
        <legend>Зарегистрируйтесь</legend>
        <p><label class="title" for="name">Ваше имя:</label>
          <input type="text" name="name" id="name"><br />
          <label class="title" for="email">Ваш email:</label>
          <input type="text" name="email" id="email"></p>
        <p><label for="location" class="title">Ближайший центр:</label>
          <select name="location" id="location">
          <option value="ny">Москва</option>
          <option value="il">Санкт-Петербург</option>
          <option value="ca">Екатеринбург</option>
          </select></p>
        <span class="title">Являетесь участником?</span>
        <label><input type="radio" name="member" value="yes" /> Да</label>
        <label><input type="radio" name="member" value="no" /> Нет</label></p>
      </fieldset>
      <div class="submit"><input type="submit" value="Регистрация" /></div>
    </form>
  </body>
</html>
```

ОБЗОР
СПИСКИ, ТАБЛИЦЫ И ФОРМЫ

- В дополнение к свойствам CSS, рассмотренным в предыдущих главах, которые применяются к содержимому элементов, существует ряд других свойств, используемых для управления внешним видом списков, таблиц и форм.
- Вид маркеров можно настраивать с помощью свойств list-style-type и list-style-image.
- Ячейки таблицы могут иметь разные границы и интервалы в различных браузерах, однако есть свойства, которые помогают управлять ими и обеспечивать их соответствие.
- Формы легче использовать, если элементы выровнены по вертикали с помощью CSS.
- Формы воспринимаются лучше, если к ним применены стили, делающие их более интерактивными.

15

МАКЕТ

- Позиционирование элементов
- Создание макета сайта
- Разработка дизайна для экранов различных размеров

В этой главе вы узнаете, как располагать элементы на странице и как создавать привлекательные макеты.

Вы станете понимать, чем отличается дизайн для экрана от дизайна для других носителей (например, для печати). В этой главе вы:

- познакомитесь с разными способами позиционирования элементов, используя нормальный поток, относительное позиционирование, абсолютное позиционирование и плавающие объекты;
- узнаете о том, что разные устройства имеют различные размеры экрана и разрешение, и о том, как это влияет на процесс проектирования;
- узнаете разницу между макетами с фиксированной шириной и гибкими макетами, а также научитесь их создавать;
- узнаете, как дизайнеры используют сетки для придания дизайну страницы более профессионального вида.

КРУТИ ПЕДАЛИ
ЧТО ТАКОЕ VELOHELD?
MacBook Pro

ПОЗИЦИОНИРОВАНИЕ ЭЛЕМЕНТОВ. КЛЮЧЕВЫЕ КОНЦЕПЦИИ

СТРУКТУРНЫЕ ЭЛЕМЕНТЫ

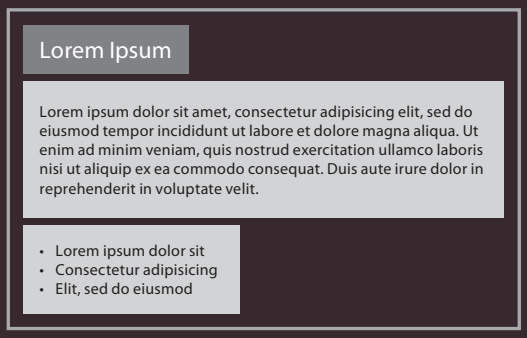
Каскадные таблицы стилей обращаются с каждым HTML-элементом так, будто он заключен в блок. При этом тот может быть как блочным, так и встроенным.

Блочные элементы располагаются на новой строке и ведут себя как основные структурные единицы макета, а встроенные элементы окружаются текстом. Вы можете указать, сколько пространства занимает каждый блок, установив его ширину (а иногда и высоту). Для отделения блоков друг от друга следует настроить границы, поля, отступы и фоновые цвета.

БЛОЧНЫЕ ЭЛЕМЕНТЫ РАСПОЛАГАЮТСЯ НА НОВОЙ СТРОКЕ

Примеры:

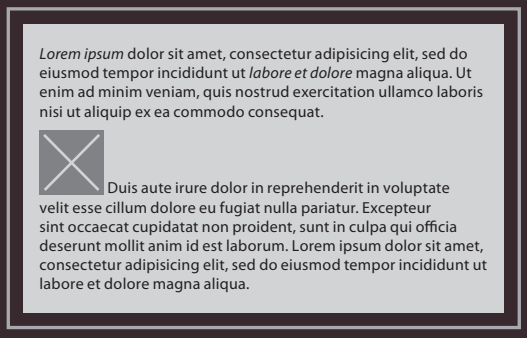
`<h1> <p> <ul> <li>`



ВСТРОЕННЫЕ ЭЛЕМЕНТЫ ОКРУЖЕНЫ ТЕКСТОМ

Примеры:

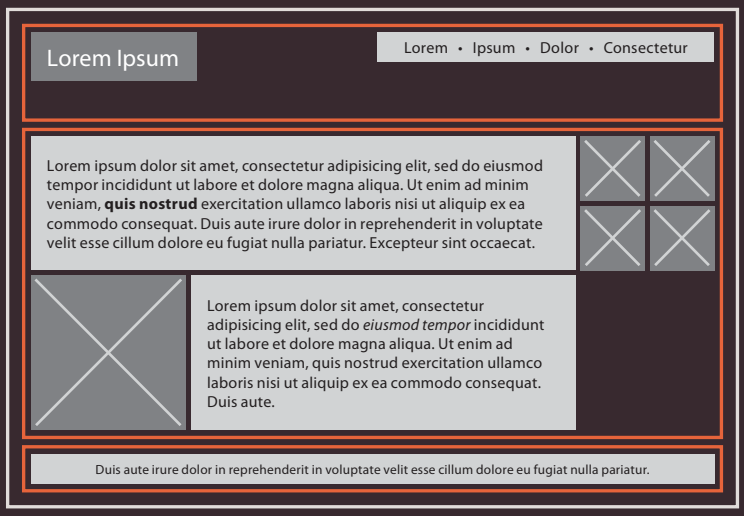
`<img> <b> <i>`



ЭЛЕМЕНТЫ-КОНТЕЙНЕРЫ

Если один блочный элемент расположен внутри другого блочного элемента, то внешний блок называется контейнером, или родительским элементом.

Часто осуществляется группировка нескольких элементов в элементе (или в другом блочном элементе). Например вы можете объединить все элементы, образующие заголовок сайта (логотип и главное меню навигации). Элемент , который содержит эту группу, будет элементом-контейнером.



Блок может быть вложен в несколько других блочных элементов. Элемент-контейнер всегда является прямым родителем данного элемента.

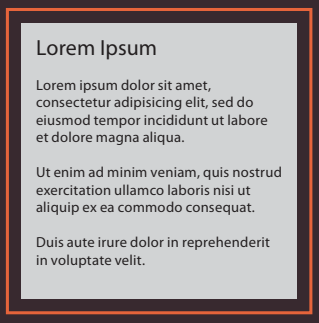
Оранжевые линии на приведенном рисунке обозначают элемент `<div>`. Заголовок, содержащий логотип и меню навигации, находится в одном элементе `<div>`, основной контент страницы — в другом, а нижний колонтитул — в третьем. `<body>` — элемент-контейнер для этих трех элементов `<div>`. Второй элемент `<div>` является элементом-контейнером для двух абзацев текста и изображений (представленных в виде перечеркнутых квадратов).

УПРАВЛЕНИЕ ПОЗИЦИОНИРОВАНИЕМ ЭЛЕМЕНТОВ

CSS предусматривает следующие схемы позиционирования, позволяющие контролировать макет страницы: нормальный поток, относительное позиционирование и абсолютное позиционирование. Вы можете указать схему с помощью свойства position. Вы также можете использовать плавающие элементы с помощью свойства float.

НОРМАЛЬНЫЙ ПОТОК

Каждый блочный элемент располагается на новой строке, в результате чего каждый последующий элемент на странице находится ниже, чем предыдущий. Даже если вы зададите ширину блоков, и на одной строке хватит места для двух элементов, они не будут располагаться рядом друг с другом. Это поведение используется по умолчанию, если вы не отдадите браузеру команду сделать что-то другое.

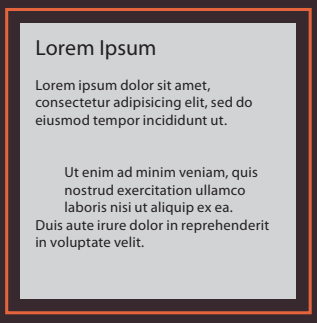


Абзацы на странице следуют один за другим сверху вниз.

См. раздел «Нормальный поток» далее в главе.

ОТНОСИТЕЛЬНОЕ ПОЗИЦИОНИРОВАНИЕ

При использовании данной схемы элемент перемещается вверх, вправо, вниз или влево от места, в котором он бы находился при использовании схемы нормального потока. Данная схема не влияет на положение окружающих элементов, они остаются там, где находились бы при использовании схемы нормального потока.

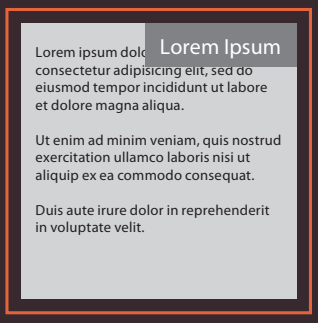


Второй абзац сдвинут вниз и вправо от того места, в котором он находился бы при использовании схемы нормального потока.

См. раздел «Относительное позиционирование» далее в главе.

АБСОЛЮТНОЕ ПОЗИЦИОНИРОВАНИЕ

При использовании данной схемы элемент позиционируется относительно элемента-контейнера. Эта схема не влияет на положение окружающих элементов. Абсолютно позиционированные элементы перемещаются при прокручивании страницы.



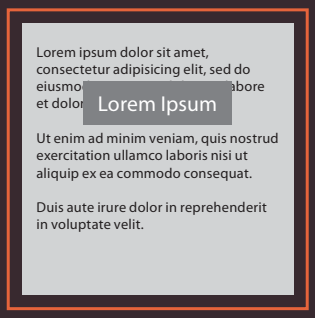
Заголовок позиционирован в правом верхнем углу, а абзацы начинаются вверху экрана так, будто заголовка там нет.

См. раздел «Абсолютное позиционирование» далее в главе.

Чтобы указать, где должен находиться блок, вы также можете использовать свойства смещения блока , чтобы сообщить браузеру, как далеко от верхнего, нижнего, левого или правого края должен быть помещен элемент. Далее в главе вы познакомитесь с этими свойствами подробнее.

ФИКСИРОВАННОЕ ПОЗИЦИОНИРОВАНИЕ

Это форма абсолютного позиционирования, которая располагает элемент относительно окна браузера, а не относительно элемента-контейнера. Элементы с фиксированным позиционированием не влияют на положение окружающих элементов и не двигаются при прокручивании страницы.

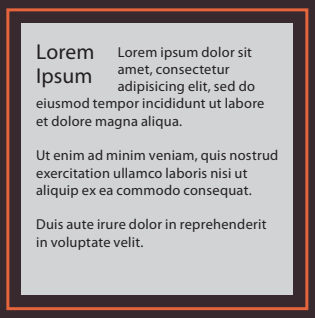


Заголовок помещен в центр страницы и на расстоянии 25% от верхнего края экрана. Остальные элементы позиционированы по схеме нормального потока.

См. раздел «Фиксированное позиционирование» далее в главе.

ПЛАВАЮЩИЕ ЭЛЕМЕНТЫ

Данная схема позволяет поместить элемент в правый или левый угол элемента-контейнера. Плавающий элемент становится блочным, вокруг которого может размещаться остальной контент.



Заголовок перемещен влево, а абзацы текста обтекают его.

См. раздел «Плавающие элементы» далее в главе.

При перемещении любого элемента, позиционированного с использованием схемы нормального потока, блоки могут пересекаться. Свойство `z-index` позволяет указать, какой из блоков должен быть помещен на переднем плане.

НОРМАЛЬНЫЙ ПОТОК
position:static

При использовании данной схемы каждый последующий блочный элемент на странице располагается ниже, чем предыдущий. Поскольку это поведение используется по умолчанию при обработке браузерами HTML-элементов, вам не нужно специальное свойство CSS, чтобы указать, что элементы должны позиционироваться в соответствии со схемой нормального потока, однако синтаксис в данном случае мог бы быть следующим:

```
position: static;
```

Свойство `width` для заголовка не указано, чтобы вы могли увидеть, как он по умолчанию растянется на всю ширину окна браузера. Ширина абзацев ограничена 450 пикселами. Это показывает, как при использовании схемы нормального потока элементы позиционируются на новой строке, даже если они не занимают всю ширину окна браузера. Все примеры, демонстрирующие схемы позиционирования, будут использовать аналогичную HTML-структуру.

глава-15/normal-flow.html HTML

```
<body>
  <h1>Эволюция велосипеда</h1>
  <p> В 1817 году барон фон Дрез изобрел машину,
      которая должна была помочь ему быстрее
      передвигаться по королевским садам...</p>
</body>
```

CSS

```
body {
  width: 750px;
  font-family: Arial, Verdana, sans-serif;
  color: #665544;}
h1 {
  background-color: #efefef;
  padding: 10px;}
p {
  width: 450px;}
```

РЕЗУЛЬТАТ

ОТНОСИТЕЛЬНОЕ ПОЗИЦИОНИРОВАНИЕ position:relative

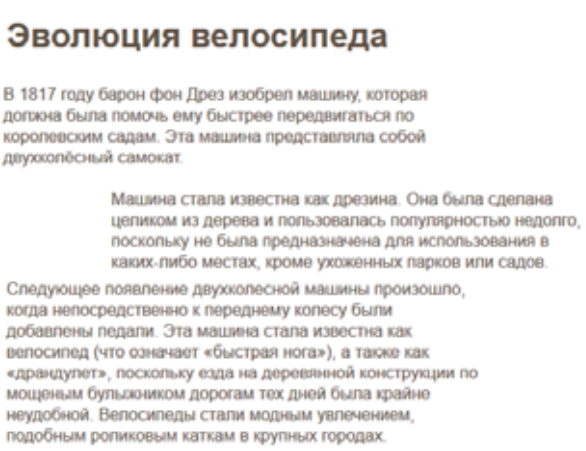
HTML **глава-15/position-relative.html**

```
<body>
  <h1>Эволюция велосипеда</h1>
  <p class="example"> В 1817 году барон фон Дрез
    изобрел машину, которая должна была помочь ему
    быстрее передвигаться по королевским садам...</p>
</body>
```

CSS

```
p.example {
  position: relative;
  top: 10px;
  left: 100px;}
```

РЕЗУЛЬТАТ



При использовании данной схемы элемент перемещается относительно места, в котором он бы находился при использовании схемы нормального потока. Например вы можете сместить элемент на 10 пикселов ниже или на 20% вправо.

Указать выбор данной схемы позиционирования позволяет свойство `position` со значением `relative`. Затем следует использовать свойства смещения (`top`, `bottom`, `left` или `right`), чтобы указать, на какое расстояние требуется переместить от того места, в котором он бы находился при использовании схемы нормального потока.

Чтобы переместить блок вверх или вниз, вы можете использовать свойство `top` или `bottom`.

Чтобы переместить блок по горизонтали, вы можете использовать свойство `left` или `right`.

Значения для свойств смещения блока обычно задаются в пикселах, процентах или единицах em.

АБСОЛЮТНОЕ ПОЗИЦИОНИРОВАНИЕ

position:absolute

Когда свойству `position` задано значение `absolute`, блок исключается из нормального потока и не влияет на положение других элементов страницы. (Для них этого блока словно не существует.)

Свойства смещения блока (`top`, `bottom`, `left` или `right`) указывают, где должен быть расположен элемент относительно контейнера.

В данном примере заголовок был позиционирован в верхней части страницы на расстоянии 500 пикселов от левого края. Ширина заголовка — 250 пикселов.

В примере свойство `width` также было применено к элементам `<p>` для сохранения удобочитаемости текста.

По умолчанию большинство браузеров добавляет поле к верхней части элемента `<h1>`. Именно потому между верхней частью окна браузера и блоком, содержащим элемент `<h1>`, существует промежуток. Если вам нужно удалить это поле, добавьте следующий фрагмент кода к правилам для элемента `<h1>`:

```
margin: 0px;
```

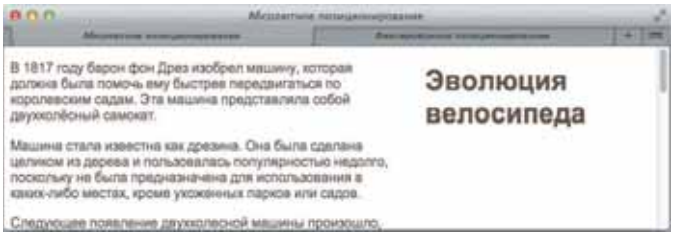
глава-15/position-absolute.html HTML

```
<body>
  <h1>Эволюция велосипеда</h1>
  <p> В 1817 году барон фон Дрез изобрел машину,
      которая должна была помочь ему быстрее
      передвигаться по королевским садам...</p>
</body>
```

CSS

```
h1 {
  position: absolute;
  top: 0px;
  left: 500px;
  width: 250px;}
p {
  width: 450px;}
```

РЕЗУЛЬТАТ



ФИКСИРОВАННОЕ ПОЗИЦИОНИРОВАНИЕ position:fixed

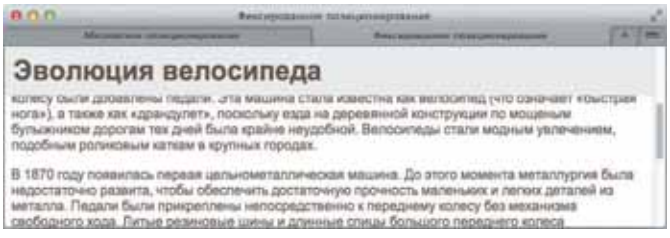
HTML **глава-15/position-fixed.html**

```
<body>
  <h1>Эволюция велосипеда</h1>
  <p class="example">В 1817 году барон фон Дрез
    изобрел машину, которая должна была помочь ему
    быстрее передвигаться по королевским садам...</p>
</body>
```

CSS

```
h1 {
  position: fixed;
  top: 0px;
  left: 50px;
  padding: 10px;
  margin: 0px;
  width: 100%;
  background-color: #efefef;}
p.example {
  margin-top: 100px;}
```

РЕЗУЛЬТАТ



Фиксированное позиционирование — это форма абсолютного позиционирования, которая требует, чтобы свойству `position` было присвоено значение `fixed`.

При использовании данной схемы элемент позиционируется относительно окна браузера. Таким образом, при прокрутке страницы элемент остается на том же месте. Посмотрите приведенный пример в браузере, чтобы увидеть данный эффект.

Для контроля над расположением элемента в окне браузера используются свойства смещения.

В данном примере заголовок был позиционирован в верхнем левом углу окна браузера. При прокрутке страницы абзацы исчезают за заголовком.

Элементы `<p>` позиционируются с помощью схемы нормального потока, поэтому игнорируют пространство, которое должно было быть занято элементом `<h1>`. Свойство `margin-top` используется для смещения первого элемента `<p>` вниз, где находится элемент `<h1>`.

ПЕРЕКРЫВАЮЩИЕСЯ ЭЛЕМЕНТЫ

z-index

При использовании относительного, фиксированного или абсолютного позиционирования блоки могут пересекаться. При этом элементы, которые указаны в HTML-коде позднее, располагаются поверх тех, которые указаны ранее. Если вы хотите задать, какой элемент будет находиться на переднем плане, вы можете использовать свойство `z-index`. Его значением является число, и чем выше это число, тем ближе данный элемент к переднему плану. Например элемент со значением свойства `z-index` 10 будет расположен поверх элемента со значением 5.

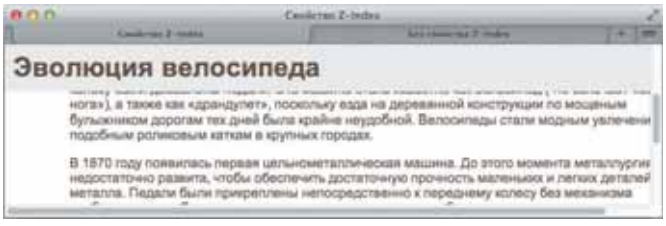
Следующий пример похож на предыдущий, однако здесь для элементов `<p>` используется схема относительного позиционирования, поэтому при прокрутке страницы они должны оказаться поверх заголовка. Чтобы элемент `<h1>` оставался на переднем плане, мы использовали свойство `z-index`.

Иногда свойство `z-index` называют **стековым контекстом** (как будто блоки складываются один на другой по оси Z). Если вы знакомы с программами предпечатной подготовки, то это эквивалентно использованию функций «на передний план» и «на задний план».

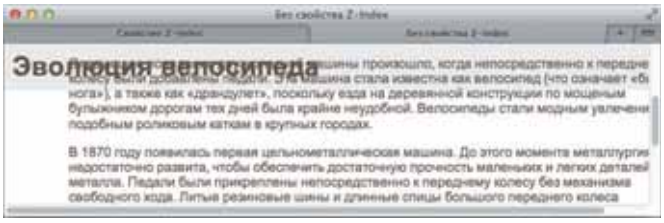
глава-15/z-index.html CSS

```
h1 {
  position: fixed;
  top: 0px;
  left: 0px;
  margin: 0px;
  padding: 10px;
  width: 100%;
  background-color: #efefef;
  z-index: 10;}
p {
  position: relative;
  top: 70px;
  left: 70px;}
```

РЕЗУЛЬТАТ БЕЗ СВОЙСТВА Z-INDEX



РЕЗУЛЬТАТ СО СВОЙСТВОМ Z-INDEX



ПЛАВАЮЩИЕ ЭЛЕМЕНТЫ
float

HTML **глава-15/float.html**

```
<h1>Эволюция велосипеда</h1>
<blockquote> "Жизнь подобна езде на велосипеде.
  Чтобы сохранять равновесие, нужно двигаться
  вперед". — Альберт Эйнштейн</blockquote>
<p> В 1817 году барон фон Дрез изобрел машину,
  которая должна была помочь ему быстрее
  передвигаться по королевским садам. Эта машина
  представляла собой двухколесный самокат...</p>
```

CSS

```
blockquote {
        float: right;
        width: 275px;
        font-size: 130%;
        font-style: italic;
        font-family: Georgia, Times, serif;
        margin: 0px 0px 10px 10px;
        padding: 10px;
        border-top: 1px solid #665544;
        border-bottom: 1px solid #665544;}
```

РЕЗУЛЬТАТ

Свойство `float` позволяет поместить элемент, позиционированный с помощью схемы нормального потока, с левого или с правого края элемента-контейнера.

Все, что находится внутри контейнера, будет обтекать плавающий элемент.

При использовании свойства `float` вы также должны использовать свойство `width`, чтобы указать ширину перемещаемого элемента. Если вы этого не сделаете, то результаты могут оказаться неожиданными, а блок, скорее всего, займет всю ширину контейнера (как происходит в случае использования схемы нормального потока).

В данном примере элемент `<blockquote>` содержит цитату и находится в элементе `<body>`.

Элемент `<blockquote>` расположен справа, а следующие за цитатой абзацы обтекают его.

ИСПОЛЬЗОВАНИЕ СВОЙСТВА FLOAT ДЛЯ РАЗМЕЩЕНИЯ ЭЛЕМЕНТОВ ДРУГ РЯДОМ С ДРУГОМ

Во многих макетах блоки помещаются друг рядом с другом. Для этого обычно используется свойство `float`.

Когда элементы являются плавающими, высота блока может влиять на расположение следующих элементов.

В этом примере вы видите шесть абзацев, каждый из которых имеет свойства `float` и `width`.

Вопреки ожиданиям, четвертый абзац не перемещается к левому краю страницы, а позиционируется прямо под третьим. Причина этого в том, что четвертый абзац начинается под третьим, однако не может сдвинуться влево, поскольку на его пути находится второй абзац.

Проблему можно решить, установив высоту абзацев равной высоте самого высокого из них, однако этот подход редко используется в связи с различным объемом текста в абзацах или колонках. Гораздо чаще используется свойство `clear` (оно обсуждается в следующем разделе).

глава-15/using-float.html HTML

```
<body>
  <h1>Эволюция велосипеда</h1>
  <p>В 1817 году барон фон Дрез изобрел машину,
      которая должна была помочь ему быстрее
      передвигаться по королевским садам...</p>
</body>
```

CSS

```
body {
  width: 750px;
  font-family: Arial, Verdana, sans-serif;
  color: #665544;}
p {
  width: 230px;
  float: left;
  margin: 5px;
  padding: 5px;
  background-color: #efefef;}
```

РЕЗУЛЬТАТ

ОЧИСТКА ПЛАВАЮЩИХ ЭЛЕМЕНТОВ

clear

HTML **глава-15/clear.html**

```
<p class="clear">Позднее непосредственно к переднему
    колесу добавили педали, однако езда
    на деревянной конструкции была крайне
    неудобной.</p>
```

CSS

```
body {
  width: 750px;
  font-family: Arial, Verdana, sans-serif;
  color: #665544;}
p {
  width: 230px;
  float: left;
  margin: 5px;
  padding: 5px;
  background-color: #efefef;}
.clear {
  clear: left;}
```

РЕЗУЛЬТАТ

Свойство `clear` позволяет указать, что ни один элемент, находящийся в том же элементе-контейнере, не должен касаться левой или правой стороны блока. Это свойство может принимать следующие значения.

left

Левая сторона блока не должна касаться любого другого элемента, находящегося в том же контейнере.

right

Правая сторона блока не должна касаться любого другого элемента, находящегося в том же контейнере.

both

Ни левая, ни правая сторона блока не должны касаться любого другого элемента, находящегося в том же контейнере.

none

Элементы могут касаться любой стороны блока.

В этом примере четвертый абзац имеет класс `clear`. Правило CSS для данного класса использует свойство `clear`, чтобы указать, что ничто не должно соприкасаться с левой стороной блока. Таким образом, четвертый абзац сдвигается вниз, чтобы никакой другой элемент не касался его левой стороны.

РОДИТЕЛЬСКИЕ ЭЛЕМЕНТЫ ПЛАВАЮЩИХ БЛОКОВ: ПРОБЛЕМА

Если в элементе-контейнере содержатся только плавающие элементы, то некоторые браузеры будут воспринимать их так, будто они имеют высоту в 0 пикселов. В данном примере границы в один пиксел, назначенные элементу-контейнеру, свернулись и поэтому выглядят как линия шириной в два пиксела.

глава-15/float-problem.html HTML

```
<body>
  <h1>Эволюция велосипеда</h1>
  <div>
    <p>В 1817 году барон фон Дрез изобрел машину,
      которая должна была помочь ему быстрее
      передвигаться по королевским садам...</p>
  </div>
</body>
```

CSS

```
div {
  border: 1px solid #665544;}
```

РЕЗУЛЬТАТ

РОДИТЕЛЬСКИЕ ЭЛЕМЕНТЫ ПЛАВАЮЩИХ БЛОКОВ: РЕШЕНИЕ

HTML **глава-15/float-solution.html**

```
<body>
  <h1>Эволюция велосипеда</h1>
  <div>
    <p> В 1817 году барон фон Дрез изобрел машину,
        которая должна была помочь ему быстрее
        передвигаться по королевским садам...</p>
  </div>
</body>
```

CSS

```
div {
  border: 1px solid #665544;
  overflow: auto;
  width: 100%;}
```

РЕЗУЛЬТАТ

Традиционно разработчики обходили данную проблему, добавляя дополнительный элемент после последнего плавающего блока (внутри элемента-контейнера). К такому дополнительному элементу применялось правило CSS, содержащее свойство `clear` со значением `both`. Однако это означало, что дополнительный элемент добавлялся в HTML-код просто для того, чтобы зафиксировать высоту контейнера.

Совсем недавно разработчики нашли решение, основанное исключительно на CSS, которое позволило обойтись без добавления дополнительных элементов в HTML-код. Данное решение применяет два правила CSS к элементу-контейнеру (в примере — к элементу `<div>`):

- свойству `overflow` присваивается значение `auto`;
- Свойству `width` присваивается значение `100%`.

СОЗДАНИЕ МНОГОКОЛОНОЧНЫХ МАКЕТОВ С ПОМОЩЬЮ ПЛАВАЮЩИХ ЭЛЕМЕНТОВ

В дизайне многих веб-страниц используются многоколоночные макеты. Это достигается путем представления каждой колонки с помощью элемента . Для позиционирования колонок друг рядом с другом используются следующие три свойства CSS.

width
Задает ширину колонок.

float
Позиционирует колонки друг рядом с другом.

margin
Создает интервал между колонками.

Двухколоночный макет, показанный на этой странице, требует наличия двух элементов : одного для основного контента, другого — для боковой панели.
Внутри каждого элемента могут находиться заголовки, абзацы, изображения и даже другие элементы .

глава-15/columns-two.html HTML

```
<h1>Эволюция велосипеда</h1>
<div class="column1of2">
  <h3>Первый велосипед</h3>
  <p> В 1817 году барон фон Дрез изобрел машину,
      которая должна была помочь ему быстрее
      передвигаться по королевским садам...</p>
</div>
<div class="column2of2">
  <h3>История велосипеда</h3> ...
</div>
```

CSS

```
.column1of2 {
  float: left;
  width: 620px;
  margin: 10px;}
.column2of2 {
  float: left;
  width: 300px;
  margin: 10px;}
```

РЕЗУЛЬТАТ

HTML **глава-15/columns-three.html**

```
<h1>Эволюция велосипеда</h1>
<div class="column1of3">
  <h3>Первый велосипед</h3> ...
</div>
<div class="column2of3">
  <h3>Дальнейшие инновации</h3> ...
</div>
<div class="column3of3">
  <h3>История велосипеда</h3> ...
</div>
```

CSS

```
.column1of3, .column2of3, .column3of3 {
  width: 300px;
  float: left;
  margin: 10px;}
```

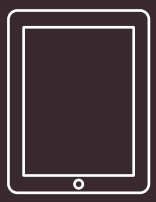
РЕЗУЛЬТАТ

Подобным образом можно создать трехколоночный макет, расположив друг рядом с другом три элемента , как показано на этой странице.

РАЗМЕРЫ ЭКРАНОВ УСТРОЙСТВ

У посетителей вашего сайта различные размеры экранов устройств, которые отображают разные объемы информации, потому дизайн ваших страниц должен подстраиваться под эту особенность.

iPhone 5
Размер: 4 дюйма
Разрешение: 1136 × 640 пикселов



iPad
Размер: 9,7 дюйма
Разрешение: 2048 × 1536 пикселов

В типографском дизайне вы всегда знаете размер листа бумаги, на котором будет производиться печать. Когда речь идет о дизайне для Всемирной паутины, вы сталкиваетесь с проблемой, заключающейся в том, что посетители для просмотра веб-контента пользуются устройствами с экранами различных размеров.

С момента начала продаж компьютеров населению размеры экранов мониторов постоянно увеличивались. Это означает, что одни посетители вашего сайта могут использовать 13-дюймовый монитор, а другие — 27-дюймовый.

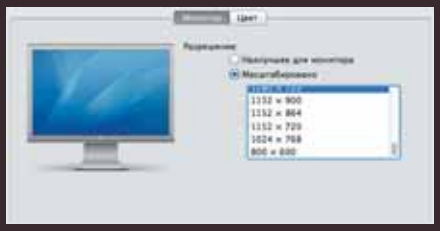
Размер экрана определяет, насколько большим может быть окно браузера и какую часть страницы увидят посетители. Помимо этого увеличивается число пользователей портативных устройств (смартфонов и планшетов) с экранами небольших размеров.

РАЗРЕШЕНИЕ ЭКРАНА

Разрешение определяется числом точек, отображаемых экраном на каждый дюйм. Некоторые мобильные устройства имеют более высокое разрешение, чем настольные компьютеры, и большинство операционных систем позволяют пользователям настраивать разрешение своих экранов.

13" MacBook Air
Размер: 13,3 дюйма
Разрешение: 1440 × 900 пикселов

27" iMac
Размер: 27 дюймов
Разрешение: 2560 × 1440 пикселов



Большинство компьютеров позволяет пользователям настраивать разрешение экрана, то есть указывать, сколько пикселов должно на нем отображаться. Например на рисунке слева показаны значения разрешения экрана в диапазоне от 800 × 600 пикселов до 1280 × 720 пикселов.

Важно отметить, что, чем выше разрешение, тем меньше размер текста. При этом разрешение экранов многих мобильных устройств превышает разрешение настольных компьютеров.

РАЗМЕР СТРАНИЦЫ

Поскольку размеры и разрешение экранов сильно разнятся, веб-дизайнеры обычно стараются создавать страницы шириной около 960–1000 пикселов (так как большинство пользователей сможет просматривать страницу такой ширины на экранах своих устройств).

Судить о высоте страницы, которую пользователи могут увидеть без необходимости прокрутки, гораздо сложнее. В течение нескольких лет дизайнеры полагали, что пользователи могут просматривать верхние 570–600 пикселов на странице без прокрутки, и некоторые пытались уместить ключевой контент в область такого размера, боясь, что люди не станут проматывать страницу.

С увеличением размера экрана и ростом популярности портативных устройств область просмотра стала менее предсказуемой.

Область страницы, которую пользователи могут увидеть без необходимости прокрутки, называется «первым экраном»*. Термин был изначально придуман журналистами для описания части первой страницы, которую можно увидеть, если газета сложена пополам.

В настоящее время признано, что если посетитель заинтересован контентом страницы, то он, скорее всего, прокрутит ее, чтобы увидеть больше. Исследования в области юзабилити показали, что посетители способны оценить страницу менее чем за секунду, поэтому важно показать им, что сайт соответствует их интересам.

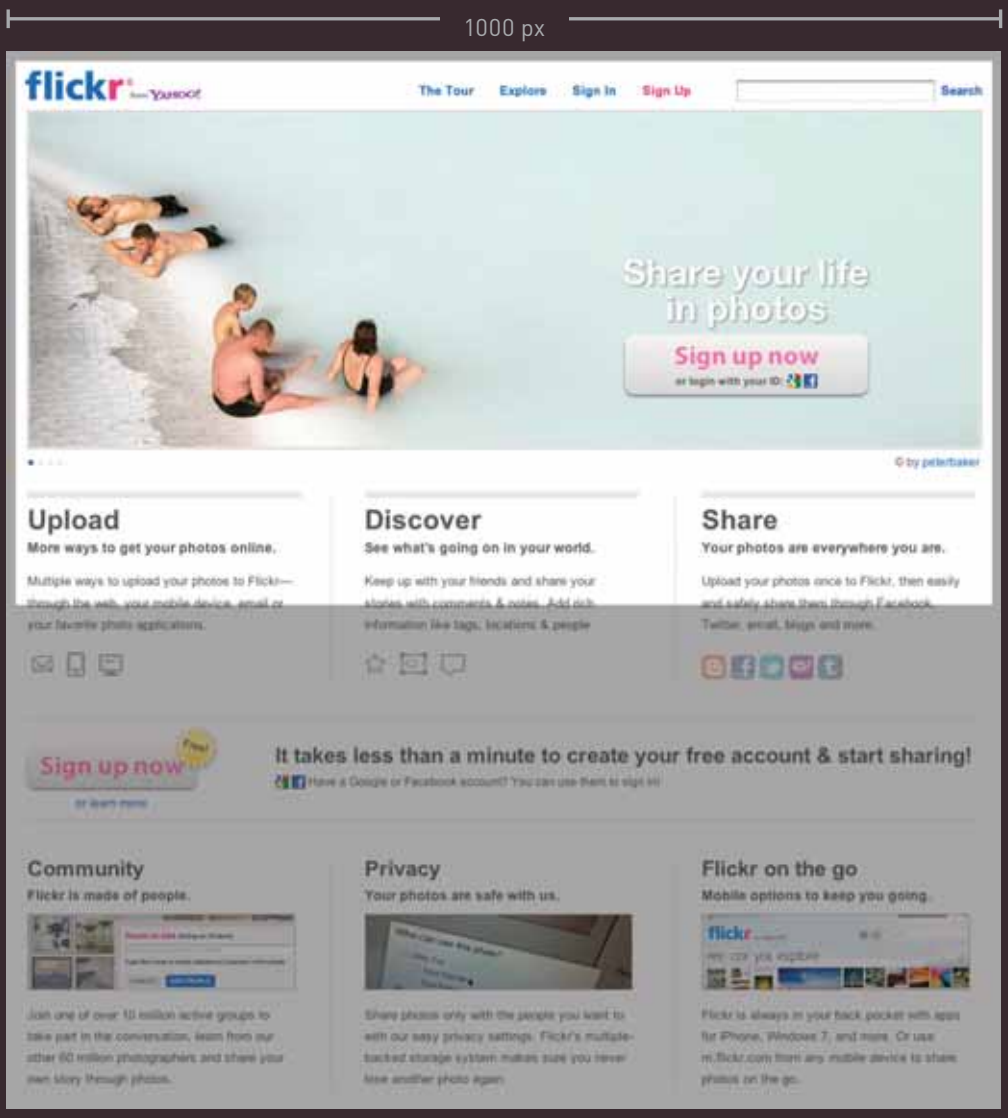
В результате многие дизайнеры до сих пор стараются показать людям, о чем их сайт, на «первом экране» в 570–600 пикселов, а также усердно намекают на более полный контент, расположенный ниже. Однако не стоит пытаться втиснуть слишком много информации в эту верхнюю область.

* Above the fold (*англ.*) — над сгибом. — *Примеч. ред.*

Тем не менее пользователь получает представление о том, что находится ниже на странице, и понимает, что увидит больше, если промотает вниз.

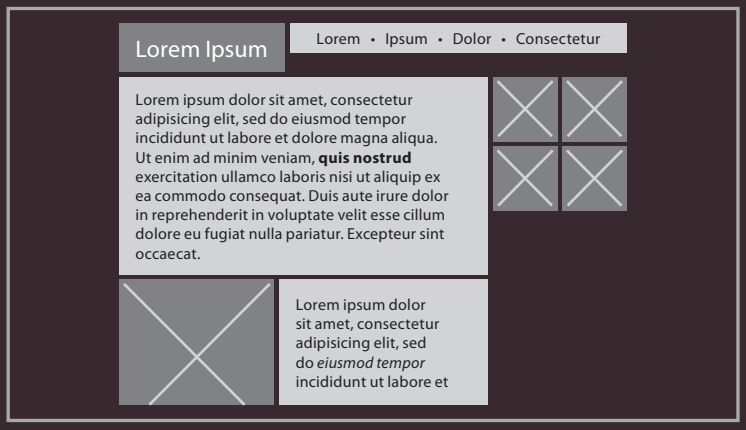
На момент написания данной книги был популярен адаптивный дизайн, при котором размеры страницы меняются в зависимости от размера экрана, а контент при этом перестраивается.

Затененная область скрыта границами окна браузера, так что пользователь должен прокрутить страницу для просмотра нижней части.



ФИКСИРОВАННЫЕ МАКЕТЫ

Страницы, созданные на основе фиксированного макета, не изменяют размер, когда пользователь увеличивает или уменьшает окно браузера. Измерения в данном случае, как правило, выражаются в пикселах.



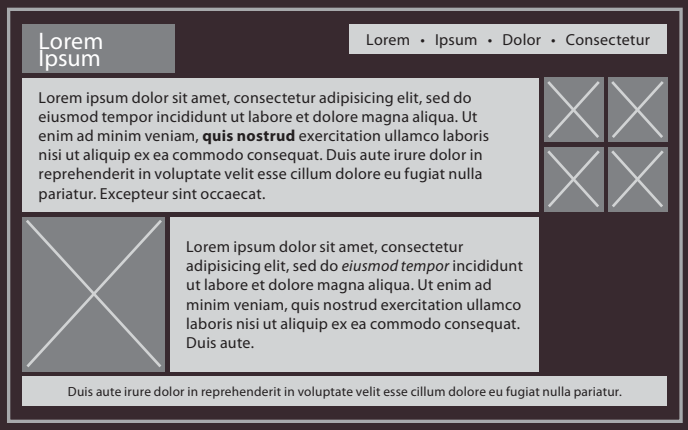
ПРЕИМУЩЕСТВА

- Значения в пикселах позволяют точно контролировать размер и позиционирование элементов.
- Дизайнер имеет большую степень контроля над внешним видом и расположением элементов на странице, чем при использовании «жидкого» макета (о «жидких» макетах см. в следующем разделе).
- Можно контролировать длину строк текста, она не зависит от размера окна пользователя.
- Размер изображения всегда останется неизменным по отношению к остальным элементам страницы.

НЕДОСТАТКИ

- По краям страницы могут образовываться большие промежутки.
- Если разрешение экрана пользователя гораздо выше, чем разрешение экрана дизайнера, то страница может оказаться уменьшенной, а текст будет труднее читать.
- Если пользователь увеличит размер шрифта, то текст может не поместиться в отведенное для него пространство.
- Такой дизайн лучше всего работает на устройствах с разрешением, близким к разрешению настольного компьютера или ноутбука.
- Страница часто будет занимать больше вертикального пространства, чем при использовании «жидкого» макета с тем же контентом.

«ЖИДКИЕ» МАКЕТЫ



Страницы, созданные на основе «жидких» макетов, растягиваются или сжимаются, когда пользователь увеличивает или уменьшает размер окна своего браузера. Измерения, как правило, выражаются в процентах.

ПРЕИМУЩЕСТВА

- Страницы растягиваются, чтобы заполнить все окно браузера, поэтому на большом экране по краям страницы не образуются пробелы.
- При небольшом окне браузера страница может сжаться до нужного размера, и пользователю не придется прокручивать ее в сторону.
- Когда пользователь увеличивает размер шрифта, проблем не возникает, поскольку страница способна растягиваться.

НЕДОСТАТКИ

- Если вы не настроили ширину разделов страницы, то дизайн может выглядеть иначе, чем вы ожидаете, поскольку около некоторых элементов с определенной вероятностью появятся промежутки.
- В очень широком окне браузера строки текста могут стать чрезмерно длинными, что затруднит восприятие.
- В очень узком окне браузера на каждой строке может оказаться лишь по несколько слов.
- Если элемент фиксированной ширины (например, изображение) находится в блоке, который его не вмещает (поскольку пользователь уменьшил размер окна), изображение может перекрыть текст.

Так как «жидкие» макеты растягиваются на всю ширину окна браузера, это часто приводит к увеличению длины строк текста и усложнению его восприятия, потому некоторые макеты такого типа позволяют расширяться и сжиматься только определенным частям страницы, при этом другие части имеют заданную минимальную и максимальную ширину.

СОЗДАНИЕ ФИКСИРОВАННОГО МАКЕТА

Для создания фиксированного макета ширина основных блоков на странице, как правило, указывается в пикселах (а иногда и высота тоже).

В приведенном примере вы можете видеть несколько элементов `<div>`, каждый из которых использует атрибут `id` или `class` для обозначения своей функции цели на странице.

В книге фиксированный и «жидкий» макеты выглядят одинаково. Чтобы понять, чем они отличаются, необходимо просмотреть их в браузере и увидеть, как они реагируют на изменение размера окна программы.

Фиксированный макет останется прежним, вне зависимости от изменения ширины окна браузера, в то время как жидкий макет будет растягиваться или сжиматься, чтобы заполнить экран.

HTML-код для обоих типов макетов — один и тот же.

глава-15/fixed-width-layout.html HTML

```
<body>
  <div id="header">
    <h1>Логотип</h1>
    <div id="nav">
      <ul>
        <li><a href="">Домой</a></li>
        <li><a href="">Товары</a></li>
        <li><a href="">Услуги</a></li>
        <li><a href="">О нас</a></li>
        <li><a href="">Контакты</a></li>
      </ul>
    </div>
  </div>
  <div id="content">
    <div id="feature">
      <p>Характеристика</p>
    </div>
    <div class="article column1">
      <p>Колонка 1</p>
    </div>
    <div class="article column2">
      <p>Колонка 2</p>
    </div>
    <div class="article column3">
      <p>Колонка 3</p>
    </div>
  </div>
  <div id="footer">
    <p>&copy; Собственность 2013</p>
  </div>
</body>
```

CSS

```
body {
  width: 960px;
  margin: 0 auto;}
#content {
  overflow: auto;
  height: 100%;}
#nav, #feature, #footer {
  background-color: #efefef;
  padding: 10px;
  margin: 10px;}
.column1, .column2, .column3 {
  background-color: #efefef;
  width: 300px;
  float: left;
  margin: 10px;}
li {
  display: inline;
  padding: 5px;}
```

РЕЗУЛЬТАТ

Правило для элемента <body> фиксирует ширину страницы на значении 960 пикселов, а сам элемент центрируется путем установления для свойства margin значения auto.

Для основных блоков на странице заданы поля в 10 пикселов, чтобы создать интервалы между ними.

Меню навигации, панель характеристик и нижний колонтитул растягиваются по ширине содержащего их элемента (в данном случае элемента <body>), поэтому не обязательно указывать для них ширину.

Для трех столбцов шириной в 300 пикселов используется свойство float, которое располагает их друг рядом с другом.

Иногда вместо фиксации ширины элемента <body> используются дополнительные HTML-элементы, в которых содержится страница. Это позволяет сделать так, чтобы фон окна браузера имел цвет, отличный от фона контента.

СОЗДАНИЕ «ЖИДКОГО» МАКЕТА

В случае с «жидким» макетом для определения ширины каждого блока используются значения в процентах, чтобы страница могла растягиваться в соответствии с изменением размера экрана.
При просмотре этого примера в браузере не забудьте несколько раз поменять размер окна.

глава-15/liquid-layout.html HTML

```
<body>
  <div id="header">
    <h1>Логотип</h1>
    <div id="nav">
      <ul>
        <li><a href="">Домой</a></li>
        <li><a href="">Товары</a></li>
        <li><a href="">Услуги</a></li>
        <li><a href="">О нас</a></li>
        <li><a href="">Контакты</a></li>
      </ul>
    </div>
  </div>
  <div id="content">
    <div id="feature">
      <p>Характеристика</p>
    </div>
    <div class="article column1">
      <p>Колонка 1</p>
    </div>
    <div class="article column2">
      <p>Колонка 2</p>
    </div>
    <div class="article column3">
      <p>Колонка 3</p>
    </div>
  </div>
  <div id="footer">
    <p>&copy; Собственность 2013</p>
  </div>
</body>
```

CSS

```
body {
  width: 90%;
  margin: 0 auto;}
#content {overflow: auto;}
#nav, #feature, #footer {
  margin: 1%;}
.column1, .column2, .column3 {
  width: 31.3%;
  float: left;
  margin: 1%;}
.column3 {margin-right: 0%;}
li {
  display: inline;
  padding: 0.5em;}
#nav, #footer {
  background-color: #efefef;
  padding: 0.5em 0;}
#feature, .article {
  height: 10em;
  margin-bottom: 1em;
  background-color: #efefef;}
```

РЕЗУЛЬТАТ

Правило для элемента `<body>` устанавливает для ширины страницы значение 90%, так что между левой и правой границами окна браузера и содержимым страницы образуются небольшие интервалы.

Поля для трех колонок равны 1%, а ширина — 31,3%. В сумме это равно 99,9% ширины элемента `<body>`, потому в некоторых браузерах правый край третьей колонки может не оказаться точно выровненным с другими элементами на странице.

Элементы `<div>`, которые содержат меню навигации, характеристики и нижний колонтитул, будут растягиваться, чтобы заполнить ширину содержащего их элемента `<body>`. Для них задано поле в 1%, чтобы выровнять их с колонками.

Если окно браузера очень широкое или очень узкое, то строки текста могут стать чрезвычайно длинными или короткими.

Свойства `min-width` и `max-width` позволяют создать границы, в пределах которых макет может растягиваться (Internet Explorer 7 стал первой версией этого браузера, поддерживающей данные свойства).

МАКЕТНЫЕ СЕТКИ

Композиция в искусстве (в дизайне, живописи или фотографии) — это организация визуальных элементов. Многие дизайнеры используют для расположения элементов сетки, так же делают и веб-дизайнеры.

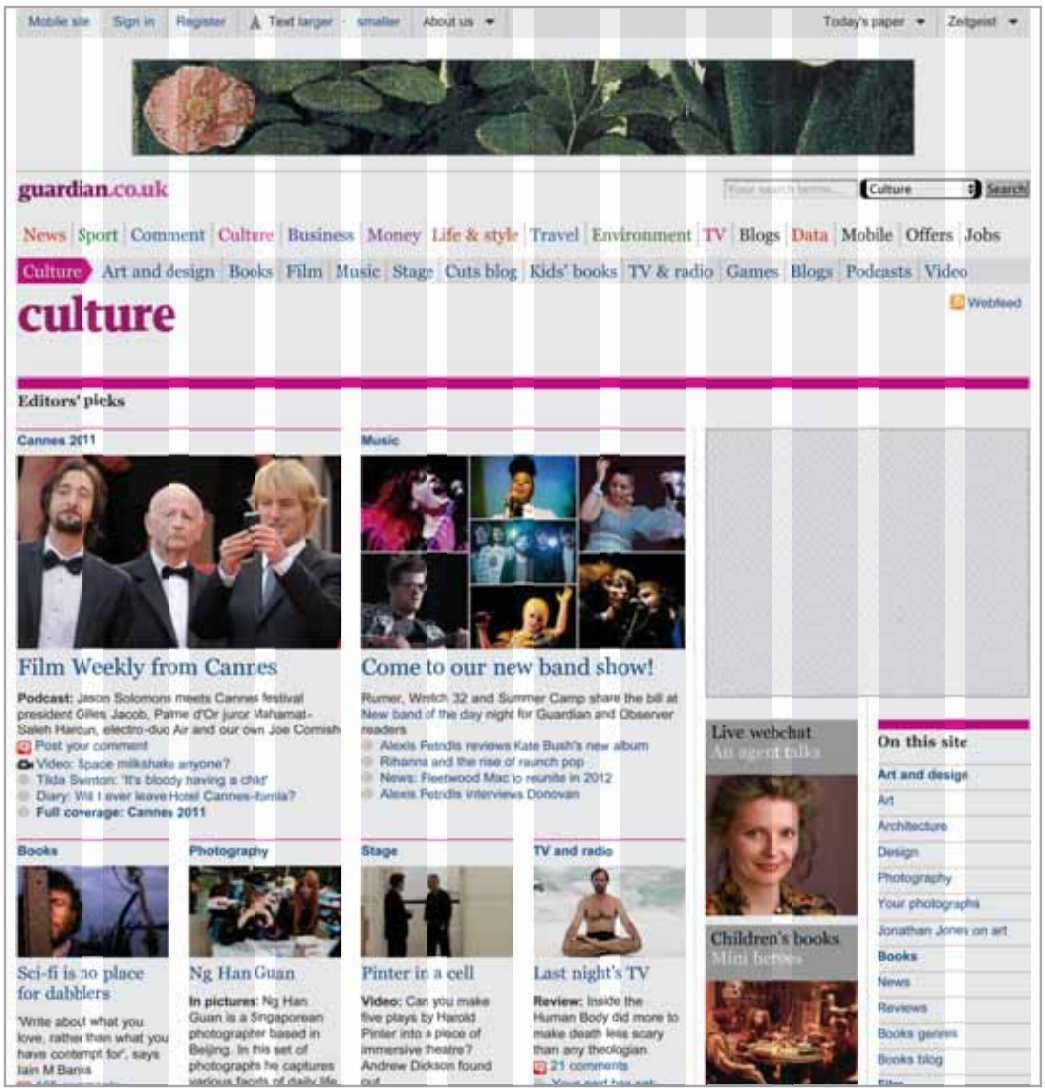
На следующем рисунке вы можете увидеть множество толстых вертикальных линий, наложенных поверх сайта и показывающих, что дизайн страницы разрабатывался с помощью сетки. Эта сетка относится к так называемой **системе сеток 960** и широко используется веб-дизайнерами.

Сетки позволяют задать одинаковые пропорции и интервалы между элементами, что помогает создать профессиональный дизайн. Если вы пролистаете страницы данной книги, то увидите, что она тоже построена на основе сетки, состоящей из трех колонок.
На следующих страницах вы увидите, что, используя универсальную сетку, можно создать много различных макетов.

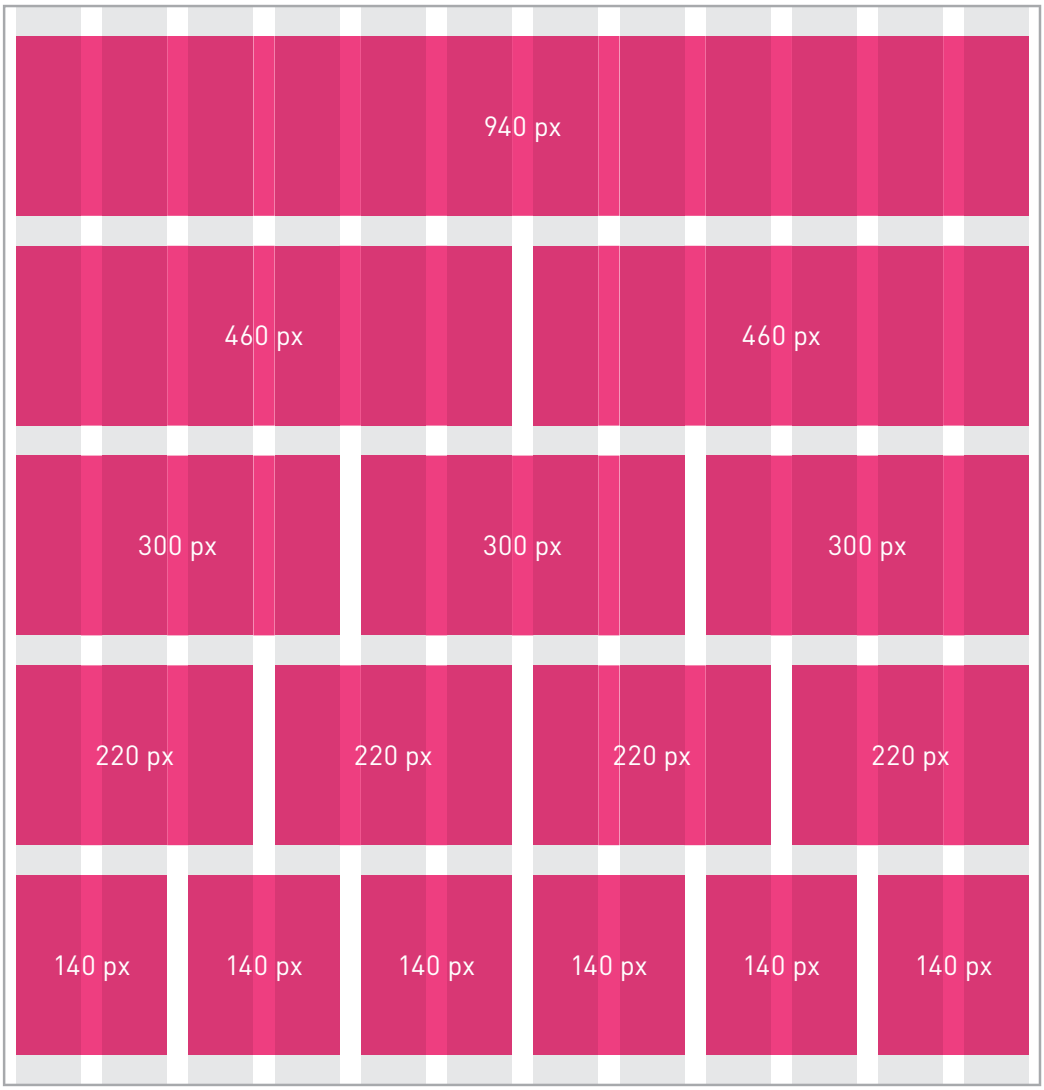
Сетка может показаться ограничением, однако на самом деле она:

- создает соответствие между страницами, которые могут использовать разный дизайн;
- помогает пользователям предугадать, где они найдут ту или иную информацию на разных страницах;
- облегчает процесс последовательного добавления нового контента на сайт;
- помогает нескольким людям работать над дизайном сайта.

ПРИМЕР СЕТКИ



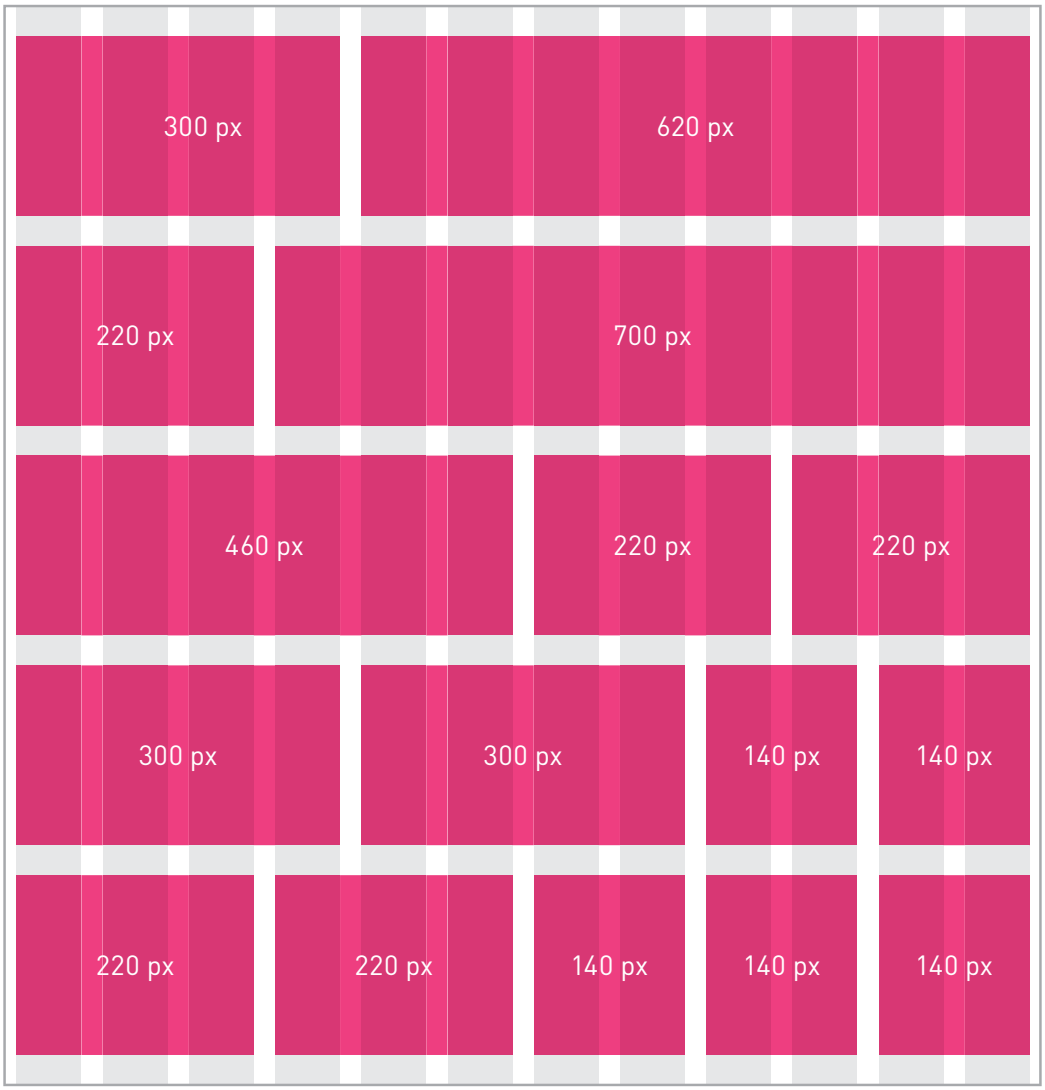
ВОЗМОЖНЫЕ МАКЕТЫ: СЕТКА ШИРИНОЙ В 960 ПИКСЕЛОВ, СОСТОЯЩАЯ ИЗ 12 КОЛОНОК



Эти две страницы демонстрируют сетку шириной в 960 пикселов, состоящую из 12 колонок. Примеры показывают, как, используя одну сетку, можно создать множество макетов.

Ширина страницы равна 960 пикселам, страница содержит 12 колонок (они выделены серым цветом) одинакового размера — в 60 пикселов.

Для каждой колонки установлено поле в 10 пикселов, которое создает интервал в 20 пикселов между колонками и интервал в 10 пикселов от правого и левого края страницы.



CSS-ФРЕЙМВОРКИ

CSS-фреймворк предоставляет код для решения таких общих задач, как создание макетных сеток, стилизация форм, создание версий страницы для печати и т.д. Вы можете использовать уже готовый CSS-код в своих проектах, а не писать его самостоятельно.

ПРЕИМУЩЕСТВА

- Вам не придется постоянно писать код для решения одних и тех же задач.
- Код был протестирован в разных версиях браузеров, что позволяет избежать появления ошибок.

НЕДОСТАТКИ

- Использование CSS-фреймворков часто требует использования в HTML-коде имен классов, которые управляют представлением страницы, а не описывают ее контент.
- Для удовлетворения разнообразных потребностей CSS-фреймворки часто содержат больше кода, чем вам необходимо для конкретной веб-страницы.

ОБЗОР CSS-ФРЕЙМВОРКА 960.GS

Чаще всего CSS-фреймворки используются для создания макетов страниц. Для этой цели существует несколько фреймворков, однако мы будем работать с системой сеток 960 (**www.960.gs**). Система 960.gs предоставляет таблицу стилей, которую вы можете включить в свою HTML-страницу. После присоединения страницы к этой таблице стилей вам нужно будет добавить соответствующие классы в HTML-код, и она создаст многоколоночный макет.

На сайте 960.gs также доступны для загрузки шаблоны для создания страниц с использованием сетки из 12 колонок. Кроме того, есть вариации сетки, состоящей из 16 колонок.

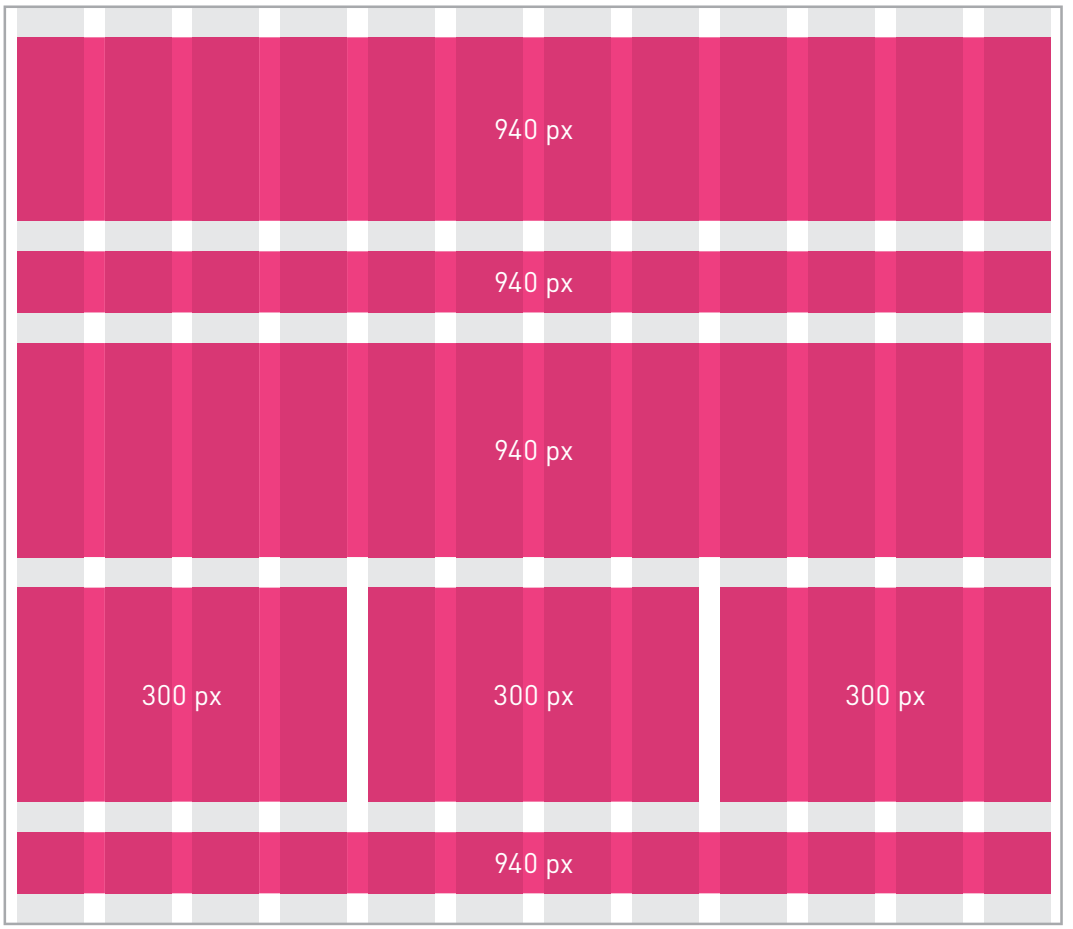
Для создания сетки из 12 колонок элементу, содержащему всю страницу, присваивается атрибут `class` со значением `container _ 12`. Это устанавливает для контента страницы ширину в 960 пикселов и указывает на использование сетки с 12 колонками.

Для блоков, занимающих колонки с 1-й по 12-ю, используются разные классы. Каждый блок использует такие имена классов, как `grid _ 3` (для блока, расположенного в трех колонках), `grid _ 4` (для блока, расположенного в четырех колонках) и т.д. вплоть до `grid _ 12` (для блока, занимающего всю ширину страницы). Эти плавающие колонки выравниваются по левой стороне и имеют поля в 10 пикселов слева и справа.

Во Всемирной паутине существуют и другие CSS-фреймворки, например **blueprintcss.org**; **lessframework.com**; **developer.yahoo.com/yui/grids/**.

ИСПОЛЬЗОВАНИЕ СЕТКИ 960.GS

На следующем рисунке вы видите образец макета страницы, подобный примеру фиксированного макета. В следующем примере мы воссоздадим его с помощью таблицы стилей 960.gs. Вместо того чтобы самостоятельно писать CSS-код для настройки макета, мы добавим в HTML-код классы, указав тем самым ширину каждого раздела.



МАКЕТ НА ОСНОВЕ СЕТКИ 960.GS

Давайте взглянем на HTML-страницу и на то, как она была размечена с помощью системы сеток 960.gs. Вы можете видеть, что мы включили CSS-код для сетки с помощью элемента `<link>` внутри элемента `<head>`.
Стили, написанные нами, показаны на следующей странице.
Таблица стилей *960_12_col.css* содержит все правила, необходимые для управления сеткой макета. В HTML-коде используются следующие имена классов:

`container _ 12` выступает в качестве контейнера для всей страницы и указывает на то, что мы используем сетку из 12 колонок;
`clearfix` сообщает браузерам высоту контейнера, поскольку он содержит только плавающие элементы (это решает проблему, о которой мы говорили в разделе «Использование свойства float для размещения элементов друг рядом с другом»);
`grid _ 12` создает блок, ширина которого соответствует ширине 12 колонок;
`grid _ 4` создает блок, ширина которого соответствует ширине 4 колонок.

глава-15/grid-layout.html HTML

```
<head>
  <title>Макет Grid Layout</title>
  <link rel="stylesheet" type="text/css"
      href="css/960_12_col.css" />
    <style>См. файл .css</style>
</head>
<body>
  <div class="container_12 clearfix">
    <div id="header" class="grid_12">
      <h1>Логотип</h1>
        <div id="nav">
          <ul>
            <li><a href="">Домой</a></li>
            <li><a href="">Товары</a></li>
            <li><a href="">Услуги</a></li>
            <li><a href="">О нас</a></li>
            <li><a href="">Контакты</a></li>
          </ul>
        </div>
    </div>
    <div id="feature" class="grid_12">
      <p>Характеристика</p>
    </div>
    <div class="article grid_4">
      <p>Колонка 1</p>
    </div>
    <div class="article grid_4">
      <p>Колонка 2</p>
    </div>
    <div class="article grid_4">
      <p>Колонка 3</p>
    </div>
    <div id="footer" class="grid_12">
      <p>&copy; Собственность 2013</p>
    </div>
  </div><!-- .container_12 -->
</body>
```

CSS

```
* {
  font-family: Arial, Verdana, sans-serif;
  color: #665544;
  text-align: center;}
#nav, #feature, .article, #footer {
  background-color: #efefef;
  margin-top: 20px;
  padding: 10px 0px 5px 0px;}
#feature, .article {
  height: 100px;}
li {
  display: inline;
  padding: 5px;}
```

РЕЗУЛЬТАТ

Таблица стилей 960.gs позаботилась о макете, создав колонки нужной ширины и установив пробелы между ними. Все необходимые правила показаны на этой странице. Данные правила:

- настраивают шрифт и положение текста в блоках;
- устанавливают цвет фона для блоков;
- устанавливают высоту блоков с характеристиками и статьями;
- добавляют поля к верхней и нижней части каждого блока.

НЕСКОЛЬКО ТАБЛИЦ СТИЛЕЙ

@import

Некоторые веб-дизайнеры размещают правила CSS в разных таблицах стилей. Например они могут использовать одну таблицу стилей для управления макетом, а другую — для настройки шрифтов, цвета и т.д.

Другие разработчики используют еще более сложный подход, создавая отдельные таблицы стилей для управления шрифтами, макетом, формами, таблицами и даже разными стилями для каждого подраздела сайта.

Существует два способа добавления нескольких таблиц стилей на страницу. HTML-страницу можно связать с одной таблицей CSS, содержащей правило `@import` для импортирования других таблиц стилей. Вы можете использовать в HTML-коде отдельный элемент `<link>` для каждой таблицы стилей.

Пример на этой странице использует один элемент `<link>` в HTML-коде для создания ссылки на таблицу стилей *styles.css*. Она затем использует правило `@import` для импортирования файлов *typography.css* и *tables.css*.

При использовании в таблице стилей правила `@import` его необходимо помещать перед всеми остальными правилами.

глава-15/multiple-style-sheets-import.html HTML

```
<!DOCTYPE html>
<html>
<head>
  <title>Несколько таблиц стилей - Импорт</title>
  <link rel="stylesheet" type="text/css" href="css/
    styles.css" />
  </head>
  <body>
    <!—Контент HTML-страницы-->
  </body>
</html>
```

глава-15/styles.css CSS

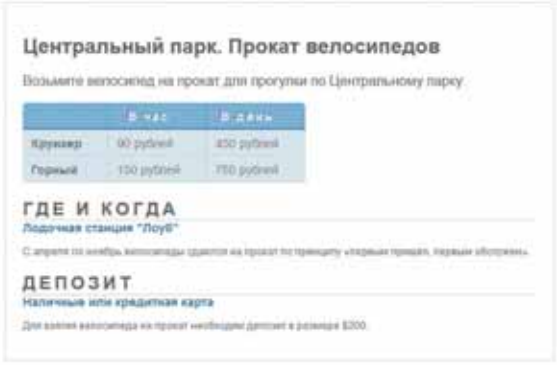
```
@import url("tables.css");
@import url("typography.css");
body {
  color: #666666;
  background-color: #f8f8f8;
  text-align: center;}
#page {
  width: 600px;
  text-align: left;
  margin-left: auto;
  margin-right: auto;
  border: 1px solid #d6d6d6;
  padding: 20px;}
h3 {
  color: #547ca0;}
```

НЕСКОЛЬКО ТАБЛИЦ СТИЛЕЙ
link

HTML **глава-15/multiple-style-sheets-link.html**

```
<!DOCTYPE html>
<html>
  <head>
    <title>Несколько таблиц стилей - Ссылка</title>
    <link rel="stylesheet" type="text/css"
      href="css/site.css" />
    <link rel="stylesheet" type="text/css"
      href="css/tables.css" />
    <link rel="stylesheet" type="text/css"
      href="css/typography.css" />
  </head>
  <body>
    <!-- Контент HTML-страницы -->
  </body>
</html>
```

РЕЗУЛЬТАТ



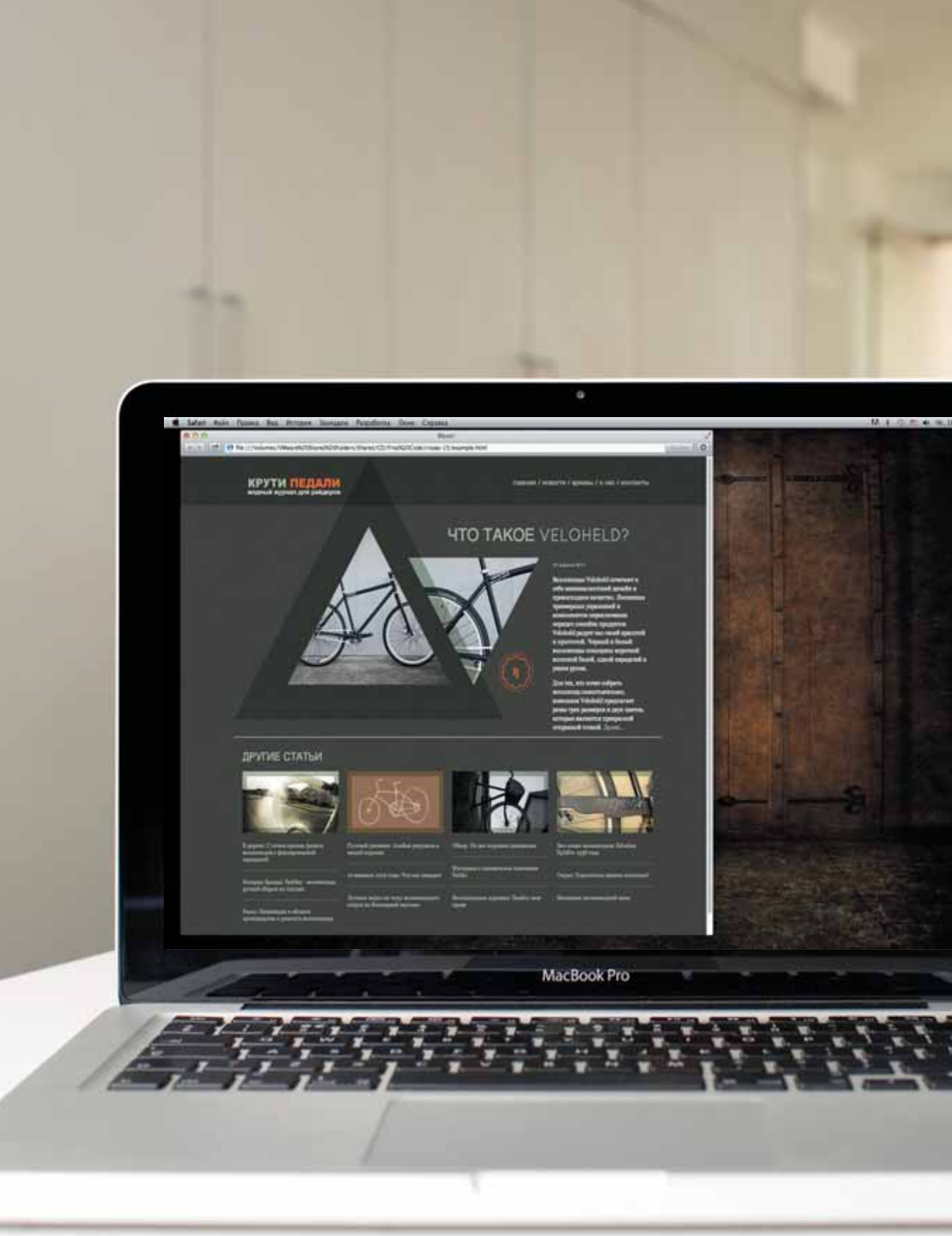
На этой странице представлен другой способ присоединения нескольких таблиц стилей. Внутри элемента `<head>` помещен отдельный элемент `<link>` для каждой таблицы стилей.

Содержимое файла *site.css* идентично содержимому файла *styles.css*, показанному на предыдущей странице, за исключением того, что код первого не содержит правила `@import`.

Как и в случае со всеми таблицами стилей, если два правила применяются к одному и тому же элементу, то указанные последними будут иметь приоритет по сравнению с предыдущими.

В примере на этой странице любые правила в файле *typography.css* будут иметь приоритет над правилами в файле *site.css* (поскольку таблица *typography.css* указана позже).

В примере на предыдущей странице правила в файле *styles.css* имели бы приоритет над правилами в файле *typography.css*, поскольку при использовании `@import` браузер считает, что правила находятся в одной таблице стилей.


КРУТИ ПЕДАЛИ
ЧТО ТАКОЕ VELOHELD?
ДРУГИЕ СТАТЬИ
MacBook Pro

ПРИМЕР
МАКЕТ

Пример демонстрирует макет, основанный на системе сеток 960.gs. Использование этой таблицы стилей избавляет нас от необходимости создавать весь CSS-код самостоятельно.

Несколько классов из таблицы стилей 960.gs были добавлены в код, чтобы указать, на ширину скольких колонок должен растянуться каждый элемент. Как вы видели в этой главе, таблица стилей 960.gs использует свойство `float` для позиционирования блоков друг рядом с другом.

Для заголовка в начале страницы используется фиксированное позиционирование (это означает, что он не двигается, когда пользователь прокручивает страницу). В заголовок добавлено свойство `z-index`, чтобы он оставался на переднем плане при промотке.

Заголовок и нижний колонтитул содержатся в элементе `<div>`, который растянут на ширину всей страницы. Внутри этих элементов-контейнеров находятся другие элементы, которые используют классы из таблицы стилей 960.gs, обеспечивающие их выравнивание с остальным контентом.

Описание характеристик занимает всю ширину страницы. Классы `push _ 7` и `push _ 9` являются частью таблицы стилей 960.gs и используются, чтобы переместить заголовок и содержимое описания вправо.

Под основной статьей находятся четыре блока, ширина каждого из которых соответствует 3 колонкам. Они содержат изображения и ссылки на другие статьи.

В этом примере также используются фоновые изображения для создания текстурированного фона для страницы и заголовка, а также для размещения изображения в статье о характеристиках. Вы узнаете об изображениях более подробно в следующей главе.

ПРИМЕР
МАКЕТ

```
<!DOCTYPE html
<html>
  <head>
    <title>Макет</title>
    <link rel="stylesheet" type="text/css" href="css/960_12_col.css" />
    <style type="text/css">
      @font-face {
        font-family: 'QuicksandBook';
        src: url('fonts/Quicksand_Book-webfont.eot');
        src: url('fonts/Quicksand_Book-webfont.eot?#iefix') format('embedded-opentype'),
          url('fonts/Quicksand_Book-webfont.woff') format('woff'),
          url('fonts/Quicksand_Book-webfont.ttf') format('truetype'),
          url('fonts/Quicksand_Book-webfont.svg#QuicksandBook') format('svg');
        font-weight: normal;
        font-style: normal;}
      body {
        color: #ffffff;
        background: #413f3b url("images/bg.jpg");
        font-family: Georgia, "Times New Roman", Times, serif;
        font-size: 90%;
        margin: 0px;
        text-align: center;}
      a {
        color: #b5c1ad;
        text-decoration: none;}
      a:hover {
        color: #ffffff;}
      .header {
        background-image: url("images/bg-header.jpg");
        padding: 0px 0px 0px 0px;
        height: 100px;
        position: fixed;
        top: 0px;
        width: 100%;
        z-index: 50;}
      .nav {
        float: right;
        font-family: QuicksandBook, Helvetica, Arial, sans-serif;
        padding: 45px 0px 0px 0px;
        text-align: right;}
```

```
        .wrapper {
          width: 960px;
          margin: 0px auto;
          background-image: url("images/bg-triangle.png");
          background-repeat: no-repeat;
          background-position: 0px 0px;
          text-align: left;}
        .logo {
          margin-bottom: 20px;}
        h1, h2 {
          font-family: QuicksandBook, Helvetica, Arial, sans-serif;
          font-weight: normal;
          text-transform: uppercase;}
        h1 {
          font-size: 240%;
          margin-top: 140px;}
        .date {
          font-family: Arial, Helvetica, sans-serif;
          font-size: 75%;
          color: #b5c1ad;}
        .intro {
          clear: left;
          font-size: 90%;
          line-height: 1.4em;}
        .main-story {
          background-image: url("images/triangles.png");
          background-repeat: no-repeat;
          background-position: 122px 142px;
          height: 570px;}
        .more-articles {
          border-top: 1px solid #ffffff;
          padding: 10px;}
        .more-articles p {
          border-bottom: 1px solid #807c72;
          padding: 5px 0px 15px 0px;
          font-size: 80%;}
.more-articles p:last-child {
border-bottom: none;}
.footer {
```

ПРИМЕР
МАКЕТ

```
clear: both;
background: rgba(0, 0, 0, 0.2);
padding: 5px 10px;}
.footer p {
font-family: Helvetica, Arial, sans-serif;
font-size: 75%;
text-align: right;}
.footer a {
color: #807c72;}
    </style>
  </head>
  <body>
    <div class="header">
      <div class="container_12">
        <div class="grid_5">
          <img src="images/logo.png" alt="Крути педали — Модный журнал для
            велосипедистов" width="216" height="37" class="logo" />
          <img src="images/header-triangle.png" alt="" width="116" height="100" />
        </div>
        <div class="nav grid_7">
          <a href="">главная</a> / <a href="">новости</a> / <a href="">архивы</a> /
            <a href="">о нас</a> / <a href="">контакты</a>
        </div>
      </div>
    </div>
    <div class="wrapper">
      <div class="main-story container_12">
        <div class="grid_6 push_6">
          <h1><a href="">Что такое Veloheld?</a></h1>
        </div>
        <div class="intro grid_3 push_9">
          <p class="date">12 апреля 2013</p>
          <p> Велосипеды Veloheld сочетают в себе минималистский дизайн и превосходное
              качество. Лишенная чрезмерных графических украшений и компонентов
              переключения передач, линейка продуктов Veloheld радует нас своей
              красотой и простотой... </p>
        </div>
      </div><!-- .main-story -->
      <div class="more-articles container_12">
```

ПРИМЕР
МАКЕТ

```
        <h2 class="grid_12"><a href="">Другие статьи</a></h2>
        <div class="grid_3">
          <img src="images/more1.jpg" alt="В дороге" width="220" height="125" />
          <p> <a href="">В дороге: С точки зрения фаната велосипедов с фиксированной
              передачей</a></p>
          <p> <a href="">История бренда: Pashley - велосипеды ручной сборки
              из Англии</a></p>
          <p> <a href="">Рамы: Инновации в области производства и ремонта
              велосипедов</a></p>
        </div>
        <div class="grid_3">
          <img src="images/more2.jpg" alt="Дневник" width="220" height="125" />
          <p><a href="">Путевой дневник: Альбом рисунков в вашей корзине</a></p>
          <p><a href="">10 новинок 2012 года: Что нас ожидает</a></p>
          <p> <a href="">Лучшие видео на тему велосипедного спорта во Всемирной
              паутине</a></p>
        </div>
        <div class="grid_3">
          <img src="images/more3.jpg" alt="Ремонт_логотип" width="220" height="125" />
          <p><a href="">Обзор: не все корзины одинаковы</a></p>
          <p><a href="">Интервью с основателем компании Public</a></p>
          <p><a href="">Велосипедная дорожка: знайте свои права</a></p>
        </div>
        <div class="grid_3">
          <img src="images/more4.jpg" alt="Schwinn Spitfire" width="220" height="125" />
          <p><a href="">Зал славы велосипедов: Schwinn Spitfire 1958 года</a></p>
          <p><a href="">Опрос: поделитесь своими мыслями!</a></p>
          <p><a href="">Эволюция велосипедной цепи</a></p>
        </div>
      </div><!-- .more-articles -->
    </div><!-- .wrapper -->
    <div class="footer clearfix">
      <div class="container_12">
        <p class="grid_12"><a href="">Юридическая информация</a> | <a href="">Положения
            о конфиденциальности</a> | <a href="">Собственность &copy; Крути педали
            2013</a></p>
      </div>
    </div>
  </body>
</html>
```

ОБЗОР
МАКЕТ

- Элементы часто используются в качестве контейнеров для группировки разделов страницы.
- Если вы не укажете схему относительного, абсолютного или фиксированного позиционирования, то браузеры будут отображать страницу в соответствии со схемой нормального потока.
- Свойство float перемещает контент в левую или в правую часть страницы и может быть использовано для создания макетов, состоящих из нескольких колонок. (плавающие элементы требуют указания ширины).
- Макеты бывают двух видов: фиксированной ширины и «жидкие», или «резиновые».
- Дизайнеры создают страницы шириной в 960–1000 пикселов и стараются в общих чертах показать контент сайта на «первом экране» (в области высотой в 600 пикселов, доступной для просмотра без прокрутки страницы).
- Сетки помогают создать профессиональный и гибкий дизайн.
- CSS-фреймворки предусматривают правила для решения общих задач.
- Вы можете включить в одну страницу несколько файлов CSS.

16

ИЗОБРАЖЕНИЯ

- Управление размером изображений с помощью CSS
- Выравнивание изображений с помощью CSS
- Добавление фоновых изображений

При управлении размером и выравниванием изображений с помощью CSS правила, влияющие на представление страницы, находятся в таблице стилей, а не в HTML-разметке.

Вы можете создать интересные эффекты, используя фоновые изображения. В этой главе вы научитесь:

- указывать размер и выравнивание изображения;
- добавлять фоновые изображения к блокам;
- создавать ролловеры с помощью CSS.

POUDRE D'IRIS
G. DUPONT
SAVON SURFIN
VIOLETTE
HELLO! ISN'T IT A LOVELY DAY?
AMER LÉON
GIRAUD-DUQUESNE
Galerie Botanique
MacBook Pro

УПРАВЛЕНИЕ РАЗМЕРОМ ИЗОБРАЖЕНИЙ С ПОМОЩЬЮ CSS

Вы можете управлять размером изображения, используя свойства `width` и `height` CSS так же, как вы использовали бы их по отношению к любому другому блоку. Указание размеров изображения оптимизирует процесс загрузки страницы, поскольку обычно сначала загружается HTML-код и CSS. Когда браузер знает, сколько места нужно оставить для изображения, он может отобразить остальную часть страницы, не ожидая загрузки графики.

Вы, вероятно, думаете, что на вашем сайте будут присутствовать изображения разных размеров, однако многие сайты используют изображения одинаковой величины.

Например на сайте интернет-магазина фотографии товаров часто имеют один и тот же размер. Если дизайн вашего сайта основан на сетке, то вы можете выбрать из нескольких размеров изображений, которые будут использоваться на всех страницах:

малая книжная: 220 × 360;
малая альбомная: 330 × 210;
фотография характеристик: 620 × 400.

Когда по всему сайту вы используете изображения фиксированной величины, вы можете применять CSS для управления размерностями, вместо того чтобы указывать их в HTML-коде.

глава-16/image-sizes.html HTML

```
<img src="images/magnolia-large.jpg"
     class="large" alt="Магнолия" />
<img src="images/magnolia-medium.jpg"
     class="medium" alt="Магнолия" />
<img src="images/magnolia-small.jpg"
     class="small" alt="Магнолия" />
```

CSS

```
img.large {
  width: 500px;
  height: 500px;}
img.medium {
  width: 250px;
  height: 250px;}
img.small {
  width: 100px;
  height: 100px;}
```

РЕЗУЛЬТАТ

Сначала вам нужно выбрать размеры изображений, которые будут использоваться на сайте, а затем назначить каждому размеру имя, например:
`small`;
`medium`;
`large`.
При использовании в HTML-коде элементов `<img>` вместо атрибутов `width` и `height`, вы можете использовать эти имена в качестве значений атрибута `class`.
В CSS вам нужно будет добавить селекторы для каждого имени класса, а затем использовать свойства `width` и `height` для управления размерами изображений.

ВЫРАВНИВАНИЕ ИЗОБРАЖЕНИЙ С ПОМОЩЬЮ CSS

В предыдущей главе вы видели, как можно использовать свойство `float` для перемещения элемента в левую или правую часть блока-контейнера, позволяя тексту обтекать его. Вместо атрибута `align` элемента `<img>` веб-дизайнеры все чаще используют свойство `float` для выравнивания изображений. Обычно это достигается следующими двумя способами.

Свойство `float` добавляется к классу, созданному для обозначения размера изображения (класс `small` в данном примере). Создаются новые классы с именами вроде `align-left` или `align-right` для выравнивания изображений по левому или правому краю страницы. Эти имена классов используются вместе с классами, определяющими размеры изображений.

В данном примере показано использование классов `align-left` и `align-right` для выравнивания изображений.

Также часто к изображению добавляются поля, чтобы текст не соприкасался с его краями.

глава-16/aligning-images.html HTML

```
<p><img src="images/magnolia-medium.jpg"
  alt="Магнолия" class="align-left medium" />
  <b> <i>Магнолия</i></b> — это род цветковых
      растений, содержащий более 200 видов...</p>
<p><img src="images/magnolia-medium.jpg"
  alt="Магнолия" class="align-right medium" />
  Некоторые виды магнолий, например <i>Магнолия
  звёздчатая</i> и <i>Магнолия Суланжа</i>, цветут
  ранней весной...</p>
```

CSS

```
img.align-left {
  float: left;
  margin-right: 10px;}
img.align-right {
  float: right;
  margin-left: 10px;}
img.medium {
  width: 250px;
  height: 250px;}
```

РЕЗУЛЬТАТ

ЦЕНТРИРОВАНИЕ ИЗОБРАЖЕНИЙ С ПОМОЩЬЮ CSS

HTML **глава-16/centering-images.html**

```
<p><img src="images/magnolia-medium.jpg"
    alt="Магнолия" class="align-center medium" />
  <b><i> Магнолия </i></b> — это род цветковых
    растений, содержащий более 200 видов.
    Род назван в честь французского ботаника
    Пьера Маньоля. Возникнув до появления пчел,
    цветы опылялись жуками.</p>
```

CSS

```
img.align-center {
  display: block;
  margin: 0px auto;}
img.medium {
  width: 250px;
  height: 250px;}
```

РЕЗУЛЬТАТ

Магнолия — это род цветковых растений, содержащий более 200 видов. Род назван в честь французского ботаника Пьера Маньоля. Возникнув до появления пчел, цветы опылялись жуками.

По умолчанию изображения являются встроенными элементами. Это означает, что они окружены текстом. Для центрирования изображения его нужно превратить в блочный элемент, используя свойство `display` со значением `block`.

После превращения изображения в блочный элемент вы можете центрировать его двумя способами:

применить к элементу-контейнеру свойство `text-align` со значением `center`;

применить к изображению свойство `margin` и установить для левого и правого полей значения `auto`.

Вы можете указать имена классов, которые позволят центрировать любой элемент, так же как и при указании размеров или выравнивании изображений. Методы определения размера и выравнивания изображений также могут использоваться с элементом HTML5 `<figure>`, с которым вы познакомились в главе 5.

ФОНОВЫЕ ИЗОБРАЖЕНИЯ

background-image

Свойство `background-image` позволяет поместить изображение позади любого HTML-элемента. Это может быть вся страница или ее часть. По умолчанию фоновое изображение повторяется, заполняя весь блок.
Путь к изображению следует после букв `url` и помещается в круглые скобки и кавычки.

Данное изображение используется в этом примере.

В первом примере изображение становится фоном всей страницы, поскольку селектор CSS применяется к элементу `<body>`. Во втором примере фоновое изображение применяется только к абзацу.
Во Всемирной паутине вы можете найти множество сайтов, предоставляющих фоновые текстуры.
Фоновые изображения обычно загружаются в последнюю очередь, что может создать впечатление медленной загрузки сайта. Как и в случае с любыми другими изображениями, которые вы используете во Всемирной паутине, чем больше размер файла, тем дольше он будет загружаться.

глава-16/background-image-body.html CSS

```
body {
  background-image: url("images/pattern.gif");}
```

РЕЗУЛЬТАТ

глава-16/background-image-element.html CSS

```
p {
  background-image: url("images/pattern.gif");}
```

РЕЗУЛЬТАТ

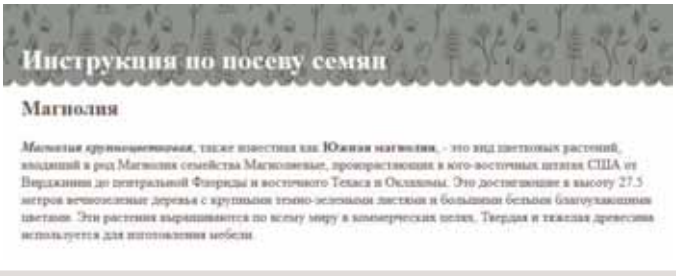
ПОВТОРЯЮЩИЕСЯ ИЗОБРАЖЕНИЯ

background-repeat
background-attachment

CSS **глава-16/background-repeat.html**

```
body {
  background-image: url("images/header.gif");
  background-repeat: repeat-x;}
```

РЕЗУЛЬТАТ



CSS **глава-16/background-attachment.html**

```
body {
  background-image: url("images/tulip.gif");
  background-repeat: no-repeat;
  background-attachment: fixed;}
```

РЕЗУЛЬТАТ

Свойство `background-repeat` может принимать четыре значения.

repeat
Фоновое изображение повторяется по горизонтали и вертикали (это поведение по умолчанию, если свойство `background-repeat` не используется).

repeat-x
Изображение повторяется только по горизонтали (как показано в первом примере).

repeat-y
Изображение повторяется только по вертикали.

no-repeat
Изображение появляется только один раз.
Свойство `background-attachment` определяет, должно ли фоновое изображение оставаться на одном месте или передвигаться при прокручивании страницы. Это свойство может принимать одно из двух значений.

fixed
Фоновое изображение остается в одном и том же положении на странице.

scroll
Фоновое изображение передвигается вверх и вниз при прокручивании страницы.

ПОЛОЖЕНИЕ ФОНОВОГО ИЗОБРАЖЕНИЯ

background-position

Если фоновое изображение не повторяется, то вы можете использовать свойство `background-position`, чтобы указать, в каком месте окна браузера оно должно быть помещено.
Это свойство обычно имеет пару значений. Первое представляет положение по горизонтали, второе — по вертикали.

- `left top`
- `left center`
- `left bottom`
- `center top`
- `center center`
- `center bottom`
- `right top`
- `right center`
- `right bottom`

Если указать только одно значение, то вторым значением по умолчанию будет `center`.
Вы также можете использовать пару значений в пикселах или процентах. Они определяют расстояние от верхнего левого угла окна браузера (или объекта-контейнера). Верхний левый угол соответствует значению 0% 0%. Приведенный пример со значением 50% 50% центрирует изображение по горизонтали и вертикали.

глава-16/background-position.html CSS

```
body {
  background-image: url("images/tulip.gif");
  background-repeat: no-repeat;
  background-position: center top;}
```

РЕЗУЛЬТАТ

глава-16/background-position-percentage.html CSS

```
body {
  background-image: url("images/tulip.gif");
  background-repeat: no-repeat;
  background-position: 50% 50%;}
```

РЕЗУЛЬТАТ

СТЕНОГРАФИЧЕСКОЕ СВОЙСТВО background

CSS **глава-16/background-shorthand.html**

```
body {
  background: #ffffff url("images/tulip.gif")
    no-repeat top right;}
```

РЕЗУЛЬТАТ

Инструкция по посеву семян

Магнолия

Magnolia grandiflora, также известная как **Южная магнолия**, - это вид цветковых растений, входящий в род Магнолия семейства Магнолиевые, произрастающих в юго-восточных штатах США от Вирджинии до центральной Флориды и восточного Техаса и Оклахомы. Это достигающий в высоту 27,5 метров вечнозеленые деревья с крупными темно-зелеными листьями и большими белыми благоухающими цветами. Эти растения выращиваются по всему миру в декоративных целях. Твердая и тяжелая древесина используется для изготовления мебели.

Лютик

Лютик азиатский является одним из видов лютиков, произрастающих в восточном Средиземноморье, в юго-западной Азии, юго-восточной Европе (Крит, Карпатос и Родос) и северо-восточной Африке. Это травянистое многолетнее растение с простыми или разветвленными стеблями достигает 45 см в высоту. Трехлопастные прикорневые листья сменяются рассеченными листьями. Как и стебли, листья покрыты пушком. Цветы достигают 3-5 см в диаметре. На одном стебле может расти от одного до нескольких цветов от красного до розового, желтого или белого цвета.

Тюльпан

Тюльпан Гесснера - это вид растений семейства Лилейные. Этот вид имеет неопределенное происхождение, вероятно, азиатское, однако прижился в юго-западной Европе. Многие из выращиваемых видов и подвидов тюльпана происходят от Тюльпана Гесснера. Цветы и луковицы могут вызвать дерматит из-за аллергена, тулипозида А, хотя луковицы можно употреблять в пищу. Душистые цветы появляются в апреле и мае. Луковицы чрезвычайно устойчивы к морозу и способны выдерживать очень низкие температуры. Период низкой температуры необходим для нормального роста и цветения. Луковицы можно сушить, измельчать и добавлять в хлеб или муку.

Свойство `background` используется в качестве стенографического свойства для всех остальных свойств фона, с которыми вы познакомились, а также для свойства `background-color`. Свойства должны указываться в следующем порядке, однако вы можете пропустить любое значение, если оно вам не нужно.

1: **`background-color`**
2: **`background-image`**
3: **`background-repeat`**
4: **`background-attachment`**
5: **`background-position`**

Язык CSS3 также допускает использование нескольких фоновых изображений путем повторения свойства `background`. Поскольку на момент написания этой книги данное свойство поддерживалось не всеми браузерами, оно использовалось не часто.

```
div {
  background:
    url(example-1.jpg)
    top left no-repeat,
    url(example-2.jpg)
    bottom left no-repeat,
    url(example-3.jpg)
    centre top repeat-x;}
```

Первое изображение находится вверху, а последнее — внизу.

ИЗОБРАЖЕНИЯ-РОЛЛОВЕРЫ И СПРАЙТЫ

Используя CSS, можно создать ссылку или кнопку, меняющую свой стиль при наведении на нее указателя мыши и при щелчке по ней — этот эффект называется **ролловером**.

Подобное достигается путем назначения для ссылки или кнопки фонового изображения, имеющего три различных стиля, при этом в каждый момент времени отображается только один из них. На следующей странице вы увидите используемое в данном примере изображение. На самом деле оно сочетает два изображения кнопки.

Когда пользователь наводит указатель мыши на элемент или щелкает по нему, фоновое изображение смещается, и отображается другое.

Когда одно изображение используется для нескольких различных частей интерфейса, оно называется **спрайтом**. Вы можете добавить логотип и другие элементы интерфейса, а также изображения кнопок.

Преимущество использования спрайтов состоит в том, что веб-браузеру необходимо запросить только одно изображение, что ускоряет процесс загрузки веб-страницы.

глава-16/image-rollovers-and-sprites.html HTML

```
<a class="button" id="add-to-basket">
   Add to basket</a>
<a class="button" id="framing-options">
   Framing options</a>
```

CSS

```
a.button {
  height: 36px;
  background-image: url("images/button-sprite.jpg");
  text-indent: -9999px;
  display: inline-block;}
a#add-to-basket {
  width: 174px;
  background-position: 0px 0px;}
a#framing-options {
  width: 210px;
  background-position: -175px 0px;}
a#add-to-basket:hover {
  background-position: 0px -40px;}
a#framing-options:hover {
  background-position: -175px -40px;}
a#add-to-basket:active {
  background-position: 0px -80px;}
a#framing-options:active {
  background-position: -175px -80px;}
```

СПРАЙТ ИЗОБРАЖЕНИЙ

РЕЗУЛЬТАТ

В данном примере вы видите две ссылки, похожие на кнопки. Каждая кнопка имеет три различных состояния. Все они представлены в одном изображении.

Поскольку элемент `<a>` является встроенным, мы назначили свойству `display` данных ссылок значение `inline-block`. Это позволяет задать ширину и высоту каждого элемента `<a>`, чтобы они совпадали с размером соответствующей кнопки.

Свойство `background-position` используется для перемещения изображения и показа нужного состояния кнопки. Когда пользователь наводит на ссылку указатель мыши, активируется псевдокласс `:hover`, содержащий правило, которое перемещает фоновое изображение, чтобы показать другое состояние кнопки.

Когда пользователь щелкает по ссылке, активируется псевдокласс `:active`, содержащий правило, показывающее третье состояние.

Устройства с сенсорным экраном меняют состояния ссылки только при ее активации.

CSS3: ГРАДИЕНТЫ

background-image

В CSS3 появилась возможность задавать для блока градиентный фон. Градиент создается с помощью свойства `background-image`. На момент написания данной книги для различных браузеров необходимо было использовать разный синтаксис.

Поскольку подобное поддерживается не всеми браузерами, можно сначала указать фоновое изображение для блока (которое будет представлять градиент), а затем предоставить альтернативный CSS-код для браузеров, которые поддерживают градиенты.

На этой странице мы сконцентрируемся на линейных градиентах. Вы видите, что для создания линейного градиента нужно указать два цвета, между которыми создается плавный переход. Некоторые браузеры позволяют указать угол градиента или даже разные типы градиентов (например, радиальный), однако линейный градиент поддерживается лучше остальных.

глава-16/gradient.html CSS

```
#gradient {
  /* fallback color */
  background-color: #66cccc;
  /* fallback image */
  background-image: url(images/fallback-image.png);
  /* Firefox 3.6+ */
  background-image: -moz-linear-gradient(#336666,
    #66cccc);
  /* Safari 4+, Chrome 1+ */
  background-image: -webkit-gradient(linear, 0% 0%,
    0% 100%, from(#66cccc), to(#336666));
  /* Safari 5.1+, Chrome 10+ */
  background-image: -webkit-linear-gradient(#336666,
    #66cccc);
  /* Opera 11.10+ */
  background-image: -o-linear-gradient(#336666,
    #66cccc);
  height: 150px;
  width: 300px;}
```

РЕЗУЛЬТАТ

КОНТРАСТНОСТЬ ФОНОВЫХ ИЗОБРАЖЕНИЙ

Если вы хотите наложить текст на фоновое изображение, то это изображение должно иметь низкую контрастность, чтобы текст был читаемым.

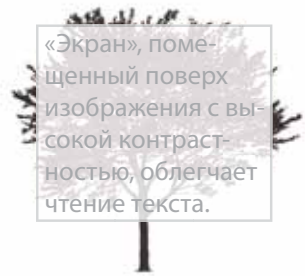
ВЫСОКАЯ КОНТРАСТНОСТЬ

Большинство фотографий имеют довольно высокую контрастность, что делает их непригодными для использования в качестве фонового изображения.

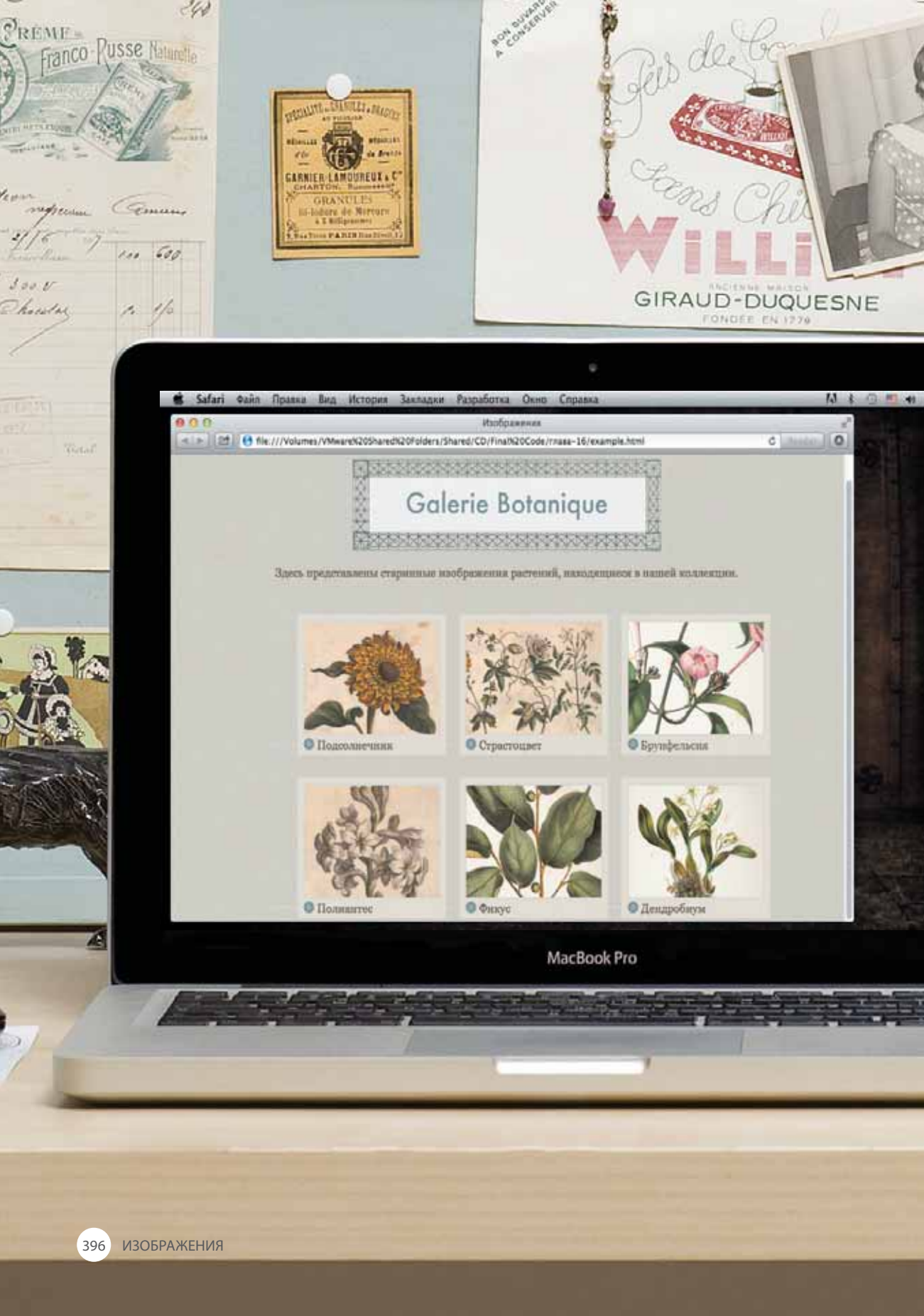
НИЗКАЯ КОНТРАСТНОСТЬ

Приложения для редактирования изображений, например, Photoshop и GIMP, предоставляют инструменты, которые позволяют вручную настроить контрастность изображения.

ЭКРАН



Чтобы наложить текст на изображение с высокой контрастностью, вы можете поместить полупрозрачный фон (или «экран») позади текста для улучшения его восприятия.


Safari Файл Правка Вид История Закладки Разработка Окно Справка
Изображения
file:///Volumes/VMware%20Shared%20Folders/Shared/CD/Final%20Code/глава-16/example.html
Galerie Botanique
Здесь представлены старинные изображения растений, находящиеся в нашей коллекции.
Подсолнечник
Страстоцвет
Брунфельсия
Полиантес
Фикус
Дендробиум
MacBook Pro
GIRAUD-DUQUESNE
FONDÉE EN 1770

ПРИМЕР
ИЗОБРАЖЕНИЯ

Этот пример демонстрирует использование CSS для создания простого макета галереи.

Фоновая текстура была применена ко всей странице путем повтора изображения с текстурой позади элемента `<body>`. Такое мозаичное фоновое изображение иногда называют **обоями**.

Содержимое страницы мы поместили в элемент `<div>` с классом `wrapper`, чтобы зафиксировать ширину страницы на значении 720 пикселов. Значение `auto` для левого и правого полей поместило контент в центр экрана.

Изображения были расположены внутри HTML5-элемента `<figure>`, а подписи к ним — внутри элемента `<figcaption>`. Для задания размера и цвета фона каждого элемента `<figure>` мы использовали CSS. Размеры изображений также были указаны с помощью CSS, и к ним мы добавили серые границы толщиной в один пиксел.

Для подписей было использовано фоновое изображение, расположенное слева от текста. Мы не планировали, чтобы изображение заполнило весь фон, поэтому указали, что оно не должно повторяться. Для текста был создан отступ, чтобы слова не заходили на фоновое изображение.

Каждый элемент `<figure>` был помещен в элементе `<div>`, мы сделали это, имея в виду две цели: во-первых — создание макета из трех колонок путем задания ширины и полей для элемента, а затем перемещения его в левую часть страницы; во-вторых, чтобы добавить едва заметные тени под каждым изображением — это создает трехмерный эффект. Чтобы гарантировать, что тени будут располагаться под изображением, мы использовали свойство `background-position`.

ПРИМЕР
ИЗОБРАЖЕНИЯ

```
<!DOCTYPE html>
<html>
  <head>
    <title>Изображения</title>
    <style type="text/css">
      body {
        color: #665544;
        background-color: #d4d0c6;
        background-image: url("images/backdrop.gif");
        font-family: Georgia, "Times New Roman", serif;
        text-align: center;}
      .wrapper {
        width: 720px;
        margin: 0px auto;}
      .header {
        margin: 40px 0px 20px 0px;}
      .entry {
        width: 220px;
        float: left;
        margin: 10px;
        height: 198px;
        background-image: url("images/shadow.png");
        background-repeat: no-repeat;
        background-position: bottom;}
      figure {
        display: block;
        width: 202px;
        height: 170px;
        background-color: #e7e3d8;
        padding: 9px;
        text-align: left;}
      figure img {
        width: 200px;
        height: 150px;
        border: 1px solid #d6d6d6;}
      figcaption {
        background-image: url("images/icon.png");
        padding-left: 20px;
        background-repeat: no-repeat;}
    </style>
```

ПРИМЕР
ИЗОБРАЖЕНИЯ

```
  </head>
  <body>
    <div class="wrapper">
      <div class="header">
        <img src="images/title.gif" alt="Galerie Botanique" width="456" height="122" />
        <p> Здесь представлены старинные изображения растений, находящиеся в нашей
            коллекции.</p>
      </div>
      <div class="entry">
        <figure><img src="images/print-01.jpg" alt="Helianthus" />
          <figcaption>Подсолнечник</figcaption>
        </figure>
      </div>
      <div class="entry">
        <figure><img src="images/print-02.jpg" alt="Passiflora" />
          <figcaption>Страстоцвет</figcaption>
        </figure>
      </div>
      <div class="entry">
        <figure><img src="images/print-03.jpg" alt="Nyctocalos" />
          <figcaption>Брунфельсия</figcaption>
        </figure>
      </div>
      <div class="entry">
        <figure><img src="images/print-04.jpg" alt="Polianthes" />
          <figcaption>Полиантес</figcaption>
        </figure>
      </div>
      <div class="entry">
        <figure><img src="images/print-05.jpg" alt="Ficus" />
          <figcaption>Фикус</figcaption>
        </figure>
      </div>
      <div class="entry">
        <figure><img src="images/print-06.jpg" alt="Dendrobium" />
          <figcaption>Дендробиум</figcaption>
        </figure>
      </div>
    </div>
  </body>
</html>
```

ОБЗОР
ИЗОБРАЖЕНИЯ

- Вы можете указать размеры изображения с помощью CSS. Это бывает очень полезно при использовании изображений одного и того же размера на нескольких страницах вашего сайта.
- С помощью CSS изображения можно выровнять по горизонтали и вертикали.
- Вы можете расположить фоновое изображение позади блока, созданного любым элементом на странице.
- Фоновые изображения могут использоваться только один раз или повторяться, заполняя весь блок.
- Вы можете создавать ролловеры, изменяя положение фонового изображения.
- Чтобы уменьшить количество изображений, которые должен загружать браузер, вы можете создать спрайты.

17

МАКЕТ В HTML5

- Элементы макета HTML5
- Как старые версии браузеров воспринимают новые элементы
- Применение стилей к элементам макета HTML5 с помощью CSS

В спецификации HTML5 появились новые элементы, которые помогают задать структуру страницы.

Они обсуждаются здесь (а не вместе с другими HTML-элементами, с которыми вы познакомились ранее в этой книге), поскольку вам легче будет понять, как они используются, теперь, когда вы знаете, как CSS управляет макетом страницы. Новые элементы будут играть важную роль в создании макетов. В этой главе вы узнаете:

- о новых элементах макета в HTML5 и о том, как они используются;
- о том, какую полезную альтернативу для элемента `<div>` они предоставляют;
- о том, как обеспечить поддержку новых элементов старыми версиями браузеров.

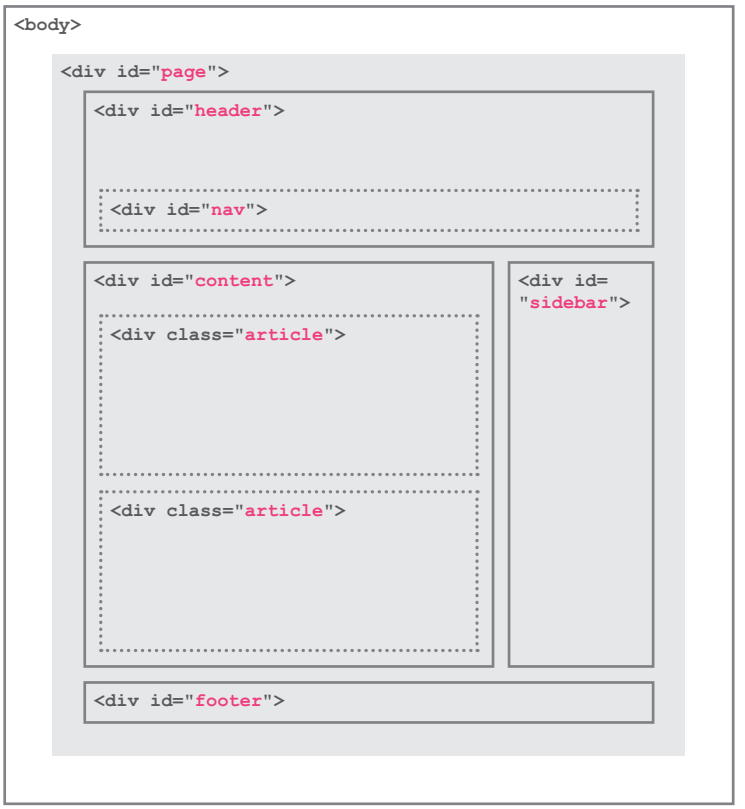
Как и в случае со всеми другими составными частями HTML5 и CSS3, в их применении возможны изменения, однако они уже широко используются веб-разработчиками и вполне вероятно, что вы тоже захотите их опробовать.

Кафе "У Такеши"
MacBook Pro

ТРАДИЦИОННЫЕ HTML-МАКЕТЫ

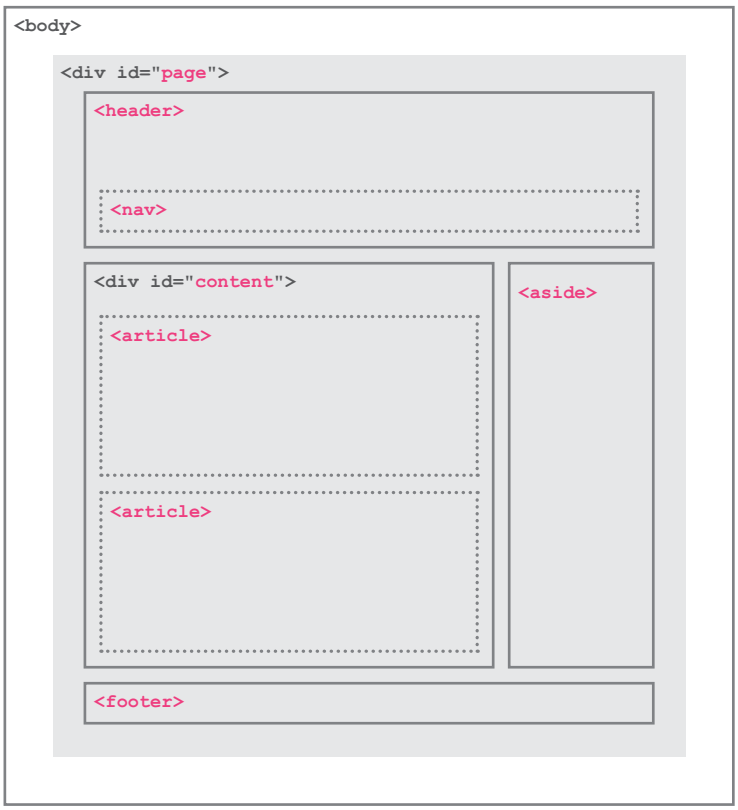
В течение долгого времени веб-дизайнеры использовали элементы для группировки элементов на странице (например, формирующих заголовок, статью, нижний колонтитул или боковую панель). Авторы использовали атрибуты class или id, чтобы указать роль элемента в структуре страницы.

На данной странице представлен довольно распространенный макет (нередко используемый для блогов).
В верхней части страницы находится заголовок, содержащий логотип и меню навигации.
Под заголовком располагаются одна или несколько статей или сообщений. Иногда они бывают обзорными и содержат ссылки на другие статьи.
Справа находится боковая панель (вероятно, предусматривающая функцию поиска, ссылки на статьи и разделы сайта или даже рекламные объявления).
При кодировании сайта таким образом разработчики, как правило, помещают эти основные разделы страницы внутри элементов `<div>` и используют атрибуты `class` или `id`, чтобы указать назначение конкретной части страницы.



НОВЫЕ ЭЛЕМЕНТЫ МАКЕТА В HTML5

В HTML5 появились новые элементы, которые позволяют разделить страницу на части. Имена элементов определяют тип контента, который будет в них располагаться. Имена еще могут измениться в финальной версии спецификации, однако это не мешает веб-дизайнерам уже использовать их.



Этот пример имеет такую же структуру, как и пример на предыдущей странице. Однако многие из элементов `<div>` в нем заменены новыми элементами HTML5. Например заголовок помещен внутри элемента `<header>`, меню навигации — в элементе `<nav>`, а статьи — в отдельных элементах `<article>`. Цель создания этих новых элементов заключается в том, чтобы помочь веб-дизайнерам описать структуру страницы. Например программы экранного доступа могут позволить пользователям проигнорировать заголовки и нижние колонтитулы и перейти прямо к содержимому. Кроме того, поисковые машины могут придать больший вес содержимому элементов `<article>`, чем содержимому `<header>` или `<footer>`. Скорее всего, вы согласитесь, что это также упрощает чтение кода.

ЗАГОЛОВКИ И НИЖНИЕ КОЛОНТИТУЛЫ

<header> <footer>

Элементы <header> и <footer> могут использоваться в качестве:

- основного заголовка или нижнего колонтитула, которые располагаются в верхней или нижней части каждой страницы сайта;
- заголовка или нижнего колонтитула отдельного элемента <article> или <section> на странице.

В данном примере элемент <header> содержит название сайта и основное меню навигации. Элемент <footer> содержит информацию об авторских правах, а также ссылки на положения политики конфиденциальности и правила использования сайта. Каждый элемент <article> и <section> также может иметь собственные элементы <header> и <footer> для размещения в них информации, касающейся конкретного раздела страницы.

Например на странице с несколькими сообщениями каждое из них можно рассматривать как отдельный раздел. В элемент <header> можно поместить название и дату сообщения, а в элемент <footer> — ссылки для размещения статьи в социальных сетях.

Обратите внимание, что весь код, приведенный в данной главе, находится в одном HTML-документе с именем *example.html*.

глава-17/example.html HTML

```
<header>
  <h1>Кафе "У Такеши"</h1>
  <nav>
    <ul>
      <li><a href="" class="current">Главная</a></li>
      <li><a href="">Кафе</a></li>
      <li><a href="">Меню</a></li>
      <li><a href="">О нас</a></li>
      <li><a href="">Контакты</a></li>
    </ul>
  </nav>
</header>
```

глава-17/example.html HTML

```
<footer>
  &copy; 2013 Кафе "У Такеши"
</footer>
```

МЕНЮ НАВИГАЦИИ

<nav>

HTML **глава-17/example.html**

```
<nav>
  <ul>
    <li><a href="" class="current">Главная</a></li>
    <li><a href="">Кафе</a></li>
    <li><a href="">Меню</a></li>
    <li><a href="">О нас</a></li>
    <li><a href="">Контакты</a></li>
  </ul>
</nav>
```

В элемент <nav> вставляется основное меню навигации сайта.

Возвращаясь к нашему примеру с блогом, если вам нужно завершить статью ссылками на соответствующие сообщения в блоге, то они не будут считаться основным меню навигации и поэтому их не следует помещать в элемент <nav>.

На момент написания данной книги некоторые разработчики, применяющие HTML5, использовали элемент <nav> для ссылок, расположенных в нижней части каждой страницы (ссылки на положения политики конфиденциальности и правила использования сайта). Станет ли это общепринятой практикой, пока не ясно.

СТАТЬИ
<article>

Элемент <article> используется в качестве контейнера для любого независимого раздела страницы.

Это может быть отдельная статья или запись в блоге, комментарий или сообщение на форуме, а также любая другая самостоятельная часть контента.

Если страница содержит несколько статей (или их краткие содержания), то каждая отдельная статья окажется заключена в собственный элемент <article>.

Элементы <article> бывают вложены друг в друга. Например сообщение в блоге может находиться в одном элементе <article>, а каждый комментарий к сообщению — в отдельном дочернем элементе <article>.

глава-17/example.html HTML

```
<article>
  <figure>
    <img src="images/bok-choi.jpg" alt="Bok Choi" />
    <figcaption>Китайская капуста</figcaption>
  </figure>
  <hgroup>
    <h2>Японская вегетарианская кухня</h2>
    <h3>Пятинедельные курсы в Лондоне</h3>
  </hgroup>
  <p> Знакомство с традиционными японскими
      вегетарианскими блюдами из риса и лапши.</p>
</article>
<article>
  <figure>
    <img src="images/teriyaki.jpg"
       alt="Соус Терияки" />
    <figcaption>Соус Терияки</figcaption>
  </figure>
  <hgroup>
    <h2>Мастер-класс по приготовлению соусов</h2>
    <h3>Однодневный курс</h3>
  </hgroup>
  <p> Однодневный интенсив поможет вам научиться
      готовить самые вкусные соусы для использования
      в приготовлении японских блюд.</p>
</article>
```

СТОРОННИЙ КОНТЕНТ

<aside>

HTML **глава-17/example.html**

```
<aside>
  <section class="popular-recipes">
    <h2>Популярные рецепты</h2>
    <a href="">Якитори (жареная курица)</a>
    <a href="">Цукунэ (пирожки с куриным фаршем)</a>
    <a href="">Окономияки (блины из различных
ингредиентов)</a>
    <a href="">Мизутаки (тушеная курица)</a>
  </section>
  <section class="contact-details">
    <h2>Адрес</h2>
    <p> Кафе "У Такеши"<br />
        Москва<br />
        Новый Арбат, 19</p>
  </section>
</aside>
```

Элемент <aside> имеет две цели, в зависимости от того, находится ли он внутри элемента <article> или нет.

Когда элемент <aside> расположен внутри элемента <article>, он должен содержать информацию, относящуюся к статье, но не обязательно к ее общему смыслу. Например врезка с цитатой или глоссарий может выступать в качестве стороннего содержимого для статьи, к которой они относятся.

Когда элемент <aside> расположен за пределами элемента <article>, он выступает в качестве контейнера для контента, относящегося ко всей странице. Например он может содержать ссылки на другие разделы сайта, список последних сообщений, строку поиска или последние твиты, добавленные автором.

РАЗДЕЛЫ
<section>

Элемент <section> группирует связанный контент, и обычно каждый раздел имеет собственный заголовок.

Например на главной странице может быть несколько элементов <section>, содержащих такие разделы страницы, как последние новости, самые популярные товары и подписка на рассылку.

Поскольку элемент <section> группирует связанный контент, он может содержать несколько различных элементов <article>, объединенных общей темой или целью.

Если страница содержит длинную статью, то элемент <section> может использоваться для разбивки ее на отдельные разделы.

Элемент <section> не следует использовать в качестве оболочки для всей страницы (если только страница не содержит один независимый фрагмент контента). Если вам нужен контейнер для всей страницы, то для решения этой задачи лучше всего подойдет элемент .

глава-17/example.html HTML

```
<section class="popular-recipes">
  <h2>Популярные рецепты</h2>
  <a href="">Якитори (жареная курица)</a>
  <a href="">Цукунэ (пирожки с куриным фаршем)</a>
  <a href="">Окономияки (блины из различных
ингредиентов)</a>
  <a href="">Мизутаки (тушеная курица)</a>
</section>
<section class="contact-details">
  <h2>Адрес</h2>
  <p> Кафе "У Такеши"<br />
      Москва<br />
      Новый Арбат, 19</p>
</section>
```

ГРУППЫ ЗАГОЛОВКОВ

<hgroup>

HTML **глава-17/example.html**

```
<hgroup>
  <h2>Японская вегетарианская кухня</h2>
  <h3>Пятинедельные курсы в Лондоне</h3>
</hgroup>
```

Элемент <hgroup> предназначен для группировки элементов <h1> — <h6>, чтобы они рассматривались как один заголовок. Например элемент <hgroup> может содержать заголовок внутри элемента <h2> и подзаголовок внутри элемента <h3>.

Этот элемент получил неоднозначный прием. Когда он был впервые предложен людьми, создающими HTML5, поступил ряд жалоб, и этот элемент исключили из HTML5. Тем не менее некоторые разработчики изменили свое мнение, и его вернули обратно. Многие веб-дизайнеры не любят использовать элемент <hgroup>, предпочитая помещать подзаголовки в элемент <p> и указывать его назначение с помощью атрибута.

Кое-кто считает, что в использовании этого элемента мало смысла. Тем не менее он довольно популярен среди разработчиков, считающих целесообразным группировку основного заголовка и подзаголовка, поскольку оба могут быть неразрывны по смыслу.

РИСУНКИ

<figure> <figcaption>

Вы уже встречались с элементом <figure> в главе 5, когда мы говорили об изображениях. Он может содержать контент, относящийся к содержимому статьи (причем не только изображения).

Важно отметить, что статья должна по-прежнему иметь смысл в случае перемещения содержимого элемента <figure> в другую часть страницы или вообще на другую страницу.

По этой причине данный элемент стоит использовать только тогда, когда контент просто ссылается на элемент, а не тогда, когда он является неотъемлемой частью содержимого страницы.

Примеры использования включают:

изображения;
видео;
графики;
диаграммы;
образцы кода.
текст, который поддерживает основной смысл статьи;

Элемент <figure> также должен содержать элемент <figcaption>, обеспечивающий описание для его контента. В данном примере элемент <figure> был добавлен в элемент <article>.

глава-17/example.html HTML

```
<figure>
  <img src="images/bok-choi.jpg" alt="Bok Choi" />
  <figcaption>Китайская капуста</figcaption>
</figure>
```

ЭЛЕМЕНТЫ ДЕЛЕНИЯ

HTML **глава-17/example.html**

```
<div class="wrapper">
  <header>
    <h1>Кафе "У Такеши"</h1>
    <nav>
      <!-- nav контент -->
    </nav>
  </header>
  <section class="courses">
    <!-- section контент -->
  </section>
  <aside>
    <!-- aside контент -->
  </aside>
  <footer>
    <!-- footer контент -->
  </footer>
</div><!-- .wrapper -->
```

Это может показаться странным — после изучения новых элементов вернуться к , ведь те, новые, часто будут использоваться вместо него.

Тем не менее останется важным способом группировки связанного контента, поскольку новые элементы можно использовать только для тех целей, для которых они предназначены.

Когда для группировки контента нет подходящего элемента, вы будете использовать . В данном примере он выступает в качестве оболочки для всей страницы.

Некоторые спрашивали, почему нет элемента <content>, содержащего основную часть страницы. Причина в том, что контентом может считаться все, что находится за пределами элементов <header>, <footer> или <aside>.

СВЯЗЫВАНИЕ БЛОЧНЫХ ЭЛЕМЕНТОВ

HTML5 позволяет веб-дизайнерам разместить элемент <a> вокруг блочного элемента, содержащего дочерние элементы. Таким образом весь блок превращается в ссылку.
Это не является новинкой HTML5, однако в более ранних версиях языка подобное использование элемента <a> не считалось правильным.

глава-17/example.html HTML

```
<a href="introduction.html">
  <article>
    <figure>
      <img src="images/bok-choi.jpg"
          alt="Bok Choi" />
      <figcaption> Китайская капуста</figcaption>
    </figure>
    <hgroup>
      <h2>Японская вегетарианская кухня</h2>
      <h3>Пятинедельные курсы в Лондоне</h3>
    </hgroup>
    <p> Знакомство с традиционными японскими
        вегетарианскими блюдами из риса и лапши.</p>
  </article>
</a>
```

ОБЕСПЕЧЕНИЕ ПОДДЕРЖКИ HTML5 В СТАРЫХ ВЕРСИЯХ БРАУЗЕРОВ

CSS **глава-17/example.html**

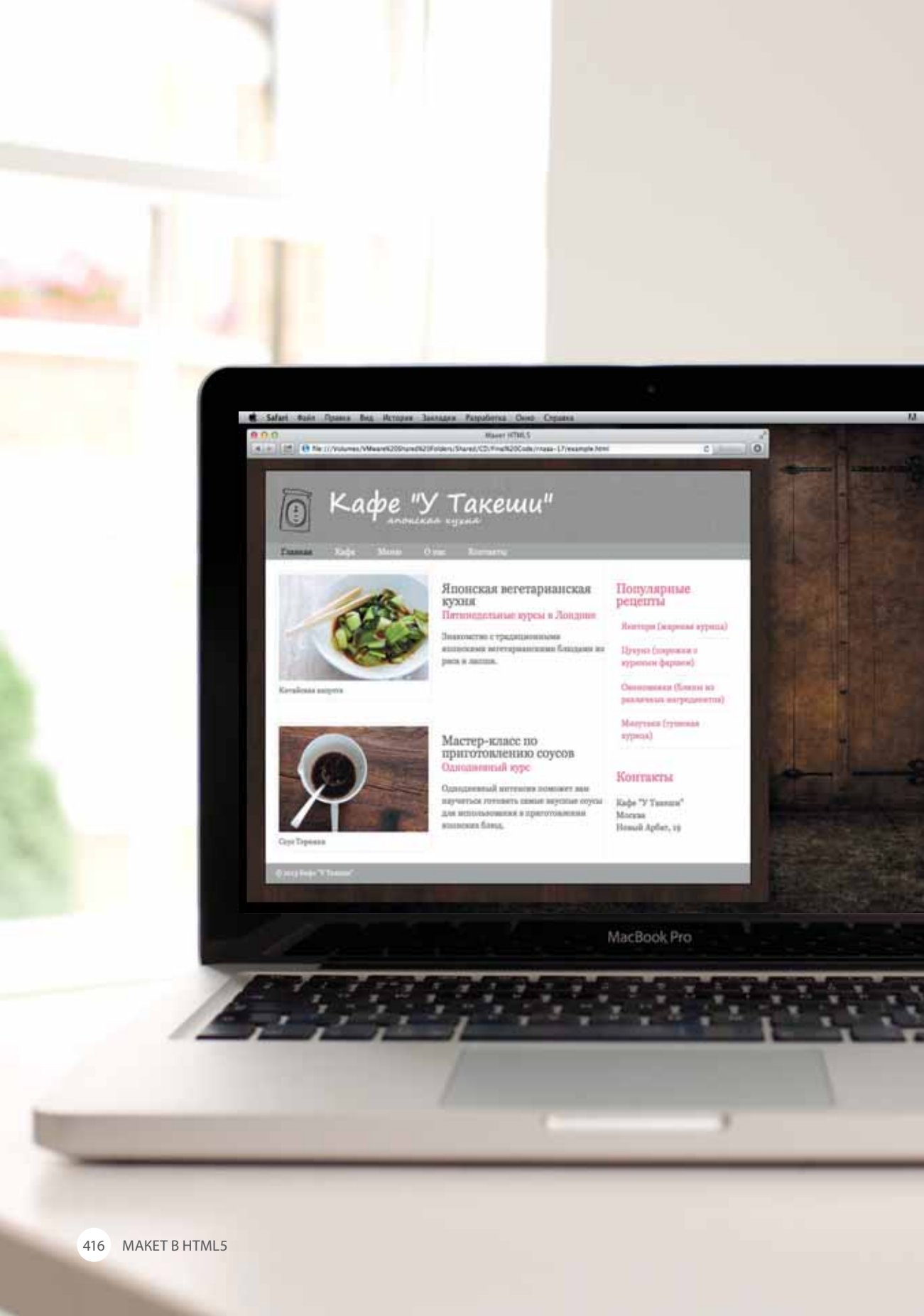
```
header, section, footer, aside, nav, article, figure
{
  display: block;}
```

HTML **глава-17/example.html**

```
<!--[if lt IE 9]>
  <script src="http://html5shiv.googlecode.com/svn/
    trunk/html5.js"></script>
<![endif]-->
```

Старые версии браузеров, не распознающие новые элементы HTML5, будут автоматически рассматривать их как встроенные. Чтобы обеспечить поддержку HTML5 в старых браузерах, вам следует добавить приведенную на данной странице строку CSS-кода, в котором говорится, какие из новых элементов следует рассматривать в качестве блочных. Кроме того, Internet Explorer 9 был первой версией данного браузера, позволяющей связывать правила CSS с этими новыми элементами макета HTML5. Чтобы применить к ним стили, используя более ранние версии IE, вам понадобится использовать простой код JavaScript, известный как **HTML5 shiv** или **HTML5 shim**. Вам не обязательно понимать язык JavaScript, чтобы использовать его. Вы можете просто добавить ссылку на файл с кодом, который хранится на серверах сервиса Google. Она должна находиться внутри **условного комментария**, который проверяет, является ли версия браузера более ранней, чем Internet Explorer 9.

К сожалению, этот метод требует, чтобы в браузере посетителя, использующего Internet Explorer 8 или более ранние версии, был активирован JavaScript. В противном случае он не увидит содержимого этих элементов HTML5.


Кафе "У Такеши"
Японская вегетарианская кухня
Популярные рецепты
Контакты
MacBook Pro

ПРИМЕР
МАКЕТ В HTML5

В этом примере демонстрируется сайт, созданный с использованием новых элементов HTML5 для описания структуры страницы вместо простой группировки контента с помощью элементов .

Заголовок и нижний колонтитул страницы мы поместили внутри элементов `<header>` и `<footer>`. Курсы были сгруппированы внутри элемента `<section>`, чей атрибут `class` имеет значение `courses` (для отличия его от других элементов `<section>` на странице). Боковая панель оказалась внутри элемента `<aside>`.
Каждый курс находится внутри элемента `<article>` и использует элементы `<figure>` и `<figcaption>` в качестве контейнеров для изображений. Названия курсов имеют подзаголовки, поэтому они группируются внутри элемента `<hgroup>`. Рецепты и контактные данные мы расположили в отдельных элементах `<section>` в боковой панели.
Стили были применены к странице с помощью CSS. Разница лишь в том, что наши селекторы используют новые элементы HTML5, позволяя создавать работающие с ними правила. Чтобы CSS-код работал в более ранних версиях программы Internet Explorer, внутри условного комментария мы поместили ссылку на код JavaScript HTML5 shiv, размещенный на серверах Google.

ПРИМЕР
МАКЕТ В HTML5

```
<!DOCTYPE html>
<html>
  <head>
    <title>Макет HTML5</title>
    <style type="text/css">
      header, section, footer, aside, nav, article, figure, figcaption {
        display: block;}
      body {
        color: #666666;
        background-color: #f9f8f6;
        background-image: url("images/dark-wood.jpg");
        background-position: center;
        font-family: Georgia, times, serif;
        line-height: 1.4em;
        margin: 0px;}
      .wrapper {
        width: 940px;
        margin: 20px auto 20px auto;
        border: 2px solid #000000;
        background-color: #ffffff;}
      header {
        height: 160px;
        background-image: url(images/header.jpg);}
      h1 {
        text-indent: -9999px;
        width: 940px;
        height: 130px;
        margin: 0px;}
      nav, footer {
        clear: both;
        color: #ffffff;
        background-color: #aeaca8;
        height: 30px;}
      nav ul {
        margin: 0px;
        padding: 5px 0px 5px 30px;}
      nav li {
        display: inline;
        margin-right: 40px;}
      nav li a {
```

```
  color: #ffffff;}
nav li a:hover, nav li a.current {
  color: #000000;}
section.courses {
  float: left;
  width: 659px;
  border-right: 1px solid #eeeeee;}
article {
  clear: both;
  overflow: auto;
  width: 100%;}
hgroup {
  margin-top:40px;}
figure {
  float: left;
  width: 290px;
  height: 220px;
  padding: 5px;
  margin: 20px;
  border: 1px solid #eeeeee;}
figcaption {
  font-size: 90%;
  text-align: left;}
aside {
  width: 230px;
  float: left;
  padding: 0px 0px 0px 20px;}
aside section a {
  display: block;
  padding: 10px;
  border-bottom: 1px solid #eeeeee;}
aside section a:hover {
  color: #985d6a;
  background-color: #efefef;}
a {
  color: #de6581;
  text-decoration: none;}
h1, h2, h3 {
  font-weight: normal;}
h2 {
```

ПРИМЕР
МАКЕТ В HTML5

```
      margin: 10px 0px 5px 0px;
      padding: 0px;}
    h3 {
      margin: 0px 0px 10px 0px;
      color: #de6581;}
    aside h2 {
      padding: 30px 0px 10px 0px;
      color: #de6581;}
    footer {
      font-size: 80%;
      padding: 7px 0px 0px 20px;}
  </style>
  <!--[if lt IE 9]>
  <script src="http://html5shiv.googlecode.com/svn/trunk/html5.js"></script>
  <![endif]-->
</head>
<body>
  <div class="wrapper">
    <header>
      <h1>Кафе "У Такеши"</h1>
      <nav>
        <ul>
          <li><a href="" class="current">Главная</a></li>
          <li><a href="">Кафе</a></li>
          <li><a href="">Меню</a></li>
          <li><a href="">О нас</a></li>
          <li><a href="">Контакты</a></li>
        </ul>
      </nav>
    </header>
    <section class="courses">
      <article>
        <figure>
          <img src="images/bok-choi.jpg" alt="Bok Choi" />
          <figcaption>Китайская капуста</figcaption>
        </figure>
        <hgroup>
          <h2>Японская вегетарианская кухня</h2>
          <h3>Пятинедельные курсы в Лондоне</h3>
        </hgroup>
```

ПРИМЕР
МАКЕТ В HTML5

```
        <p> Знакомство с традиционными японскими вегетарианскими
            блюдами из риса и лапши.</p>
      </article>
      <article>
        <figure>
          <img src="images/teriyaki.jpg" alt="Соус Терияки" />
          <figcaption>Соус терияки</figcaption>
        </figure>
        <hgroup>
          <h2>Мастер-класс по приготовлению соусов</h2>
          <h3>Однодневный курс</h3>
        </hgroup>
        <p> Однодневный интенсив поможет вам научиться готовить самые
            вкусные соусы для использования в приготовлении японских
            блюд.</p>
      </article>
    </section>
    <aside>
      <section class="popular-recipes">
        <h2>Популярные рецепты</h2>
        <a href="">Якитори (жареная курица)</a>
        <a href="">Цукунэ (пирожки с куриным фаршем)</a>
        <a href="">Окономияки (блины из различных ингредиентов)</a>
        <a href="">Мизутаки (тушеная курица)</a>
      </section>
      <section class="contact-details">
        <h2>Адрес</h2>
        <p> Кафе "У Такеши"<br />
            Москва<br />
            Новый Арбат, 19</p>
        </section>
    </aside>
    <footer>
      &copy; 2013 Кафе "У Такеши"
    </footer>
  </div><!-- .wrapper -->
 </body>
</html>
```

ОБЗОР
МАКЕТ В HTML5

- Новые элементы HTML5 определяют функциональное назначение различных частей веб-страницы и помогают описать ее структуру.
- Новые элементы делают код понятнее по сравнению с использованием нескольких элементов .
- Старым версиям браузеров, которые не распознают код HTML5, необходимо сообщить, какие элементы следует считать блочными.
- Чтобы элементы HTML5 работали в программе Internet Explorer 8 (и в более ранних версиях), необходимо наличие дополнительного кода JavaScript, бесплатно предоставляемого сервисом Google.

18

ПРОЦЕСС РАЗРАБОТКИ

- Как подойти к разработке сайта
- Понимание аудитории и ее потребностей
- Как представить информацию, необходимую посетителям

В этом разделе рассматривается процесс создания нового сайта.

Речь пойдет о возможных посетителях вашего сайта и о том, как обеспечить их нужной информацией. Также будут рассмотрены некоторые ключевые аспекты теории создания сайтов, помогающие придать им профессиональный вид. В этой главе вы:

- научитесь понимать потенциальных посетителей вашего сайта и узнаете, какую информацию они ожидают на нем найти;
- узнаете, как организовать информацию, чтобы посетители смогли найти то, что они ищут;
- познакомитесь с теорией создания сайтов, позволяющей представить информацию таким образом, чтобы посетители сумели достичь своих целей;
- получите ряд советов, которые помогут вам создавать более привлекательные и профессиональные сайты.

Events
NEWS
STAR EVENTS ON...

ДЛЯ КОГО ПРЕДНАЗНАЧЕН САЙТ

Каждый сайт должен создаваться с учетом потребностей целевой аудитории, а не просто «чтоб был» или для владельца сайта. Таким образом, очень важно понимать своих посетителей.

Бывает полезно задать несколько вопросов о людях, которые, по вашему мнению, могут заинтересоваться вашим сайтом.
Если вы спросите клиента, для кого предназначен сайт, то часто можете услышать в ответ: «Для всех».
Если посмотреть на вещи более реалистично, то сайт вряд ли будет интересен каждому. Если на вашем сайте продаются электрические лампочки, то люди, скорее всего, не станут заказывать их из другой страны, хотя большинство посетителей, использующих компьютер, вероятно, используют и лампочки. Даже если сайт имеет широкую направленность, вам все же следует подумать о демографических характеристиках целевой аудитории.

ЦЕЛЕВАЯ АУДИТОРИЯ: ФИЗИЧЕСКИЕ ЛИЦА

- Каков возрастной диапазон вашей целевой аудитории?
- Рассчитан ли ваш сайт скорее на женщин или мужчин? Какова пропорция тех и других?
- В какой стране живут ваши посетители?
- Они живут в городской или в сельской местности?
- Каков средний доход посетителей?
- Какой у них уровень образования?
- Каково их семейное положение?
- Какова профессия ваших посетителей?
- Сколько часов в неделю они работают?
- Как часто они выходят в Интернет?
- Какие устройства они используют для доступа к Интернету?

ЦЕЛЕВАЯ АУДИТОРИЯ: ЮРИДИЧЕСКИЕ ЛИЦА

- Каков размер компании или соответствующего отдела?
- Каково положение посетителей вашего сайта внутри компании?
- Будут ли посетители использовать ваш сайт для себя или для кого-то другого?
- Каким бюджетом они располагают?

Придумайте несколько типичных представителей вашей целевой аудитории. Они станут вашими друзьями. Они повлияют на дизайн, начиная от цветов и заканчивая степенью детальности описаний.

ИМЯ	ГЕННАДИЙ	МАРИЯ	ИВАН	АНДРЕЙ	АННА
Пол	М	Ж	М	М	Ж
Возраст	28	47	19	32	35
Местоположение	Владивосток	Санкт-Петербург	Москва	Калининград	Екатеринбург
Профессия	Учитель	Адвокат	Студент	Продавец	Журналист
Доход (в год)	120 000 руб.	500 000 руб.	60 000 руб.	240 000 руб.	200 000 руб.
Использование Интернета	2–3 раза в неделю	Ежедневно	Ежедневно	4–5 раз в неделю	Ежедневно

Если у вас возникнет вопрос о том, как будет использоваться сайт или каковы его приоритеты, то вы сможете вернуться и спросить себя: «А что бы хотели получить Геннадий или Мария?»

ПОЧЕМУ ЛЮДИ ПОСЕЩАЮТ ВАШ САЙТ

Теперь, когда вы знаете своих посетителей, вам необходимо понять, *зачем* они туда приходят. Некоторые люди попадут на ваш сайт случайно, однако большинство посетителей придет, имея определенную причину.

Ваш контент и дизайн должны соответствовать целям аудитории.
Для определения потребностей посетителей вы можете задать вопросы, относящиеся к двум основным категориям:
основные **мотивы** посещения сайта;
конкретные **цели** посетителей — то, что заставляет их зайти на сайт *прямо сейчас*.

ОСНОВНЫЕ МОТИВЫ

- Хотят ли посетители просто развлечься или им необходимо достичь конкретной цели?
- Если конкретная цель есть, является ли она личной или профессиональной?
- Рассматривают ли они трату времени на это занятие необходимой или излишней?

КОНКРЕТНЫЕ ЦЕЛИ

- Ищут ли посетители общую информацию / проводят исследование (какой-либо темы/компании) / нуждаются в конкретных данных (например, в фактах или информации о товаре)?
- Знакомы ли они уже с вашими товарами и услугами или нуждаются в их представлении?
- Ищут ли они чувствительную ко времени информацию, например последние новости или обновления по определенной теме?
- Нужны ли им данные по конкретному товару или услуге для принятия решения о покупке?
- Необходимо ли им связаться с вами? Если да, то могут ли они посетить вас лично (в этом случае потребуется время работы и схема проезда) или им нужен номер телефона и адрес электронной почты?

ЧЕГО ХОТЯТ ДОБИТЬСЯ ВАШИ ПОСЕТИТЕЛИ

Маловероятно, что вы сможете перечислить все причины для посещения вашего сайта, однако вам нужно определить ключевые задачи и мотивы. Эта информация поможет вам в создании сайта.

Сначала вам следует перечислить причины, по которым люди будут посещать ваш сайт. Затем вы сможете назначить задачи для ваших вымышленных посетителей.

ГЕННАДИЙ купил теннисную ракетку несколько лет назад, а теперь хочет приобрести еще одну на вашем сайте для своей подруги.

МАРИЯ прочитала в прессе о вашей новой услуге по уходу за собаками и хочет узнать, подойдет ли та для нее.

ИВАН собирается изучать архитектуру и хочет узнать больше о предлагаемом курсе.

АНДРЕЙ получил негативный опыт пребывания в гостинице и хочет подать жалобу.

АННА работает редактором фотографий и хочет посмотреть сайт с работами фотографа, чтобы решить, стоит ли их покупать.

КАКАЯ ИНФОРМАЦИЯ ТРЕБУЕТСЯ ВАШИМ ПОСЕТИТЕЛЯМ

Итак, вы знаете, кто и почему приходит на сайт, и теперь вам нужно понять, в какой информации нуждаются ваши посетители для быстрого и эффективного достижения своих целей.

Вы способны предложить дополнительную информацию, которая, по вашему мнению, может оказаться полезной посетителям.

Рассмотрите каждую из причин, по которой люди будут посещать ваш сайт, и определите, что им потребуется для достижения их целей.

Вы можете расположить данные в порядке приоритетов — от ключевых моментов до несущественной или дополнительной информации.

Если вы предоставите посетителям нужную им информацию, то ваш сайт окажется для них более актуальным.

Таким образом, у вас будет больше возможностей сообщить им любую дополнительную информацию, по вашему мнению, полезную для них (или познакомить их с другими вашими товарами и услугами).

Если вы не предоставите информацию, отвечающую потребностям ваших посетителей, то они, вероятно, станут искать ее в другом месте.

Ниже приведено несколько вопросов, которые помогут вам решить, какой информацией необходимо обеспечить посетителей вашего сайта.

КЛЮЧЕВАЯ ИНФОРМАЦИЯ

- Будут ли посетители знать о вашей теме/бренде или вам придется их с ними познакомить?
- Будут ли они знакомы с товаром/услугой/информацией, которые вы предоставляете, или им потребуются дополнительные сведения?
- Каковы наиболее важные признаки того, что вы предлагаете?
- Что особенного о том, что вы предлагаете? Что отличает вас от других подобных сайтов?
- После того, как посетители добьются своей цели, останутся ли у них общие вопросы по данной теме?

КАК ЧАСТО ЛЮДИ БУДУТ ПОСЕЩАТЬ ВАШ САЙТ

Некоторые сайты нуждаются в более частом обновлении, чем другие. Часть информации (например, новости) может постоянно меняться, в то время как другой контент будет оставаться относительно неизменным.

Сайт о модных тенденциях придется обновлять гораздо чаще, чем ресурс, продвигающий услуги, которыми люди пользуются нерегулярно (например, ремонт водопровода или остекление лоджий).

Частое обновление сайта может потребовать значительного количества временны*х* и других ресурсов.

Поняв, как часто люди будут пересматривать ваш сайт, вы сможете решить, как часто вам следует его обновлять.

Весьма полезно установить график обновления сайта, вместо того чтобы делать это от случая к случаю.

Вы можете обнаружить, что одни части сайта выигрывают от обновления сильнее, чем другие.

Далее перечислены некоторые вопросы, которые помогут вам решить, как часто следует обновлять содержимое сайта.

ТОВАРЫ/УСЛУГИ

- Как часто одни и те же люди совершают у вас покупки?
- Как часто обновляются ваши запасы или меняются ваши услуги?

ИНФОРМАЦИЯ

- Как часто обновляется ваша тема?
- Сравните процент посетителей, регулярно возвращающихся за обновленной информацией, с процентом тех, кому необходимо получить информацию только единожды.

КАРТА САЙТА

Теперь, когда вы знаете, что должно содержаться на вашем сайте, вы можете приступить к организации информации по разделам или страницам.

Цель состоит в том, чтобы создать диаграмму страниц, которая станет использоваться для структурирования сайта. Эта **карта сайта** должна показывать, как могут быть сгруппированы страницы.

Чтобы решить, какая информация должна присутствовать на каждой из страниц, вы можете использовать технику под названием **карточная сортировка**.

Она предполагает размещение каждого отдельного фрагмента информации, которая может понадобиться посетителю, на отдельном листе бумаги. Затем нужно организовать эти фрагменты информации по группам.

Каждая группа соответствует странице, а в случае с большими сайтами страницы, в свою очередь, могут быть объединены в различные разделы.

Сгруппированная информация затем трансформируется в схему — карту сайта.

Иногда организовать информацию способен помочь опрос представителей вашей целевой аудитории.

Карта обычно начинается с главной страницы. Кроме того, если сайт большой и содержит множество разделов, то каждый может содержать собственную домашнюю страницу, включающую ссылки на подразделы внутри него.

Например, большинство интернет-магазинов содержит главные страницы разделов, посвященных определенному виду товаров, которые, в свою очередь, включают ссылки на страницы с описанием конкретного товара.

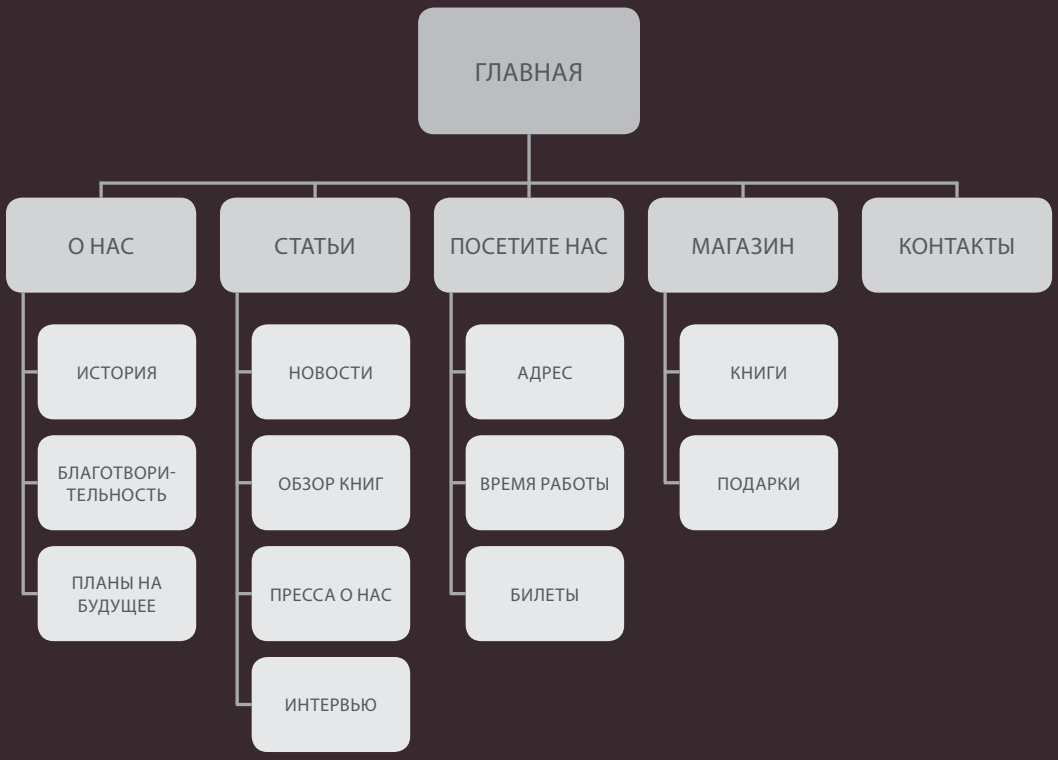
Возможно, вам придется дублировать некоторую информацию, если ее необходимо поместить более чем на одной странице.

Страницы (или группы страниц) определяют то, как пользователи будут перемещаться по сайту.

Постарайтесь сосредоточиться на целях, которых хотят достичь ваши посетители.

Стоит отметить, что владелец сайта может организовать информацию способом, отличным от того, который ожидают посетители. Очень важно учесть не только мнение владельца, но и мнение аудитории.

ПРИМЕР КАРТЫ САЙТА



СТРУКТУРНЫЕ СХЕМЫ

Структурная схема (или каркас) — это набросок, содержащий ключевые фрагменты информации, которая должна находиться на каждой из страниц сайта. Она отображает информационную иерархию и позволяет определить, сколько места потребуется для размещения того или иного фрагмента.

Многие дизайнеры начинают создание схемы с размещения элементов, которые будут находиться на каждой странице (например, логотип, главное меню навигации, заголовки и основные фрагменты текста, окно авторизации пользователей и т.д.).
Создание схемы гарантирует, что страницы будут содержать все необходимые элементы.

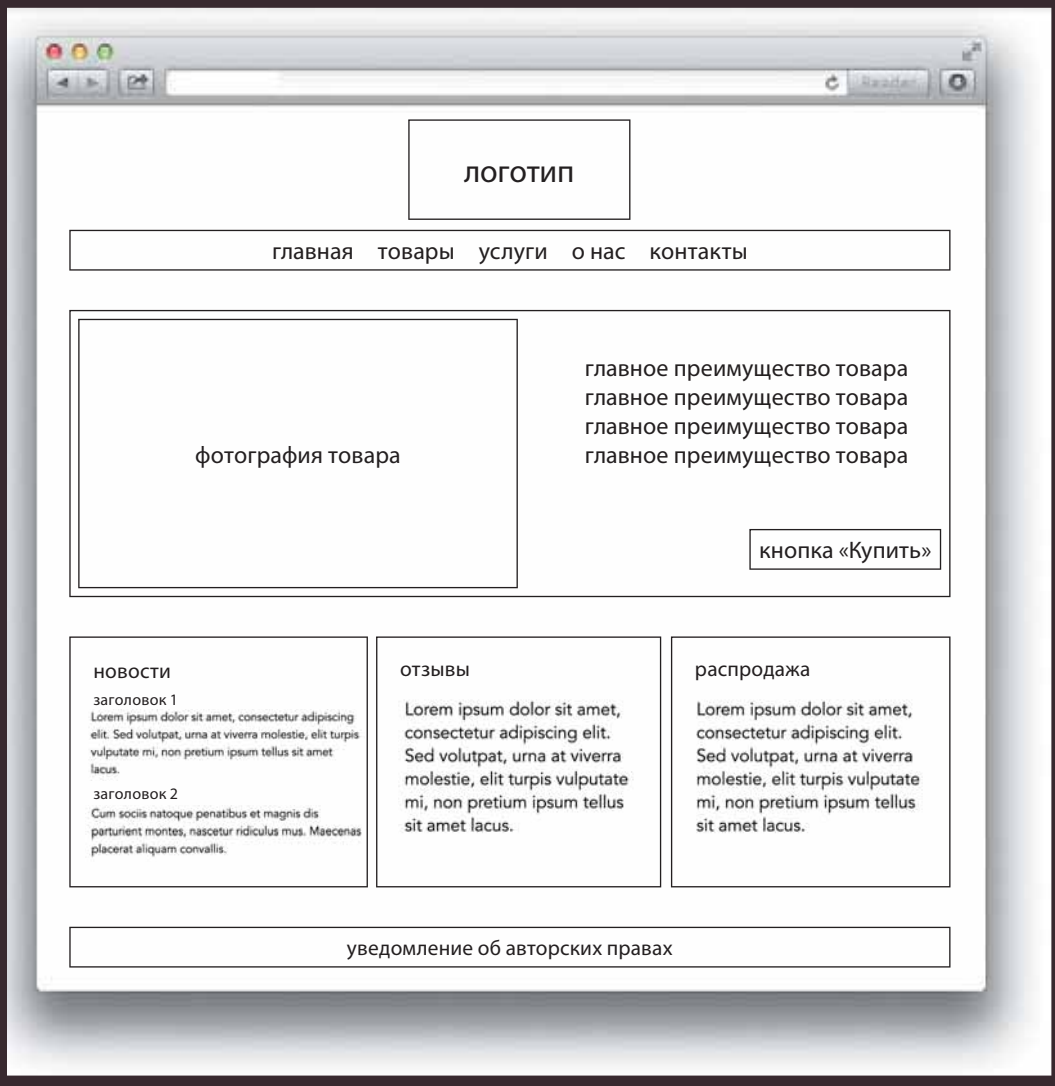
Схема не должна содержать информацию о цветовой гамме, шрифте, фоне или изображениях. Она предназначена для создания визуальной иерархии и определения самых важных частей каждой страницы.
Схемы облегчают процесс разработки сайта, поскольку позволяют увидеть, какая информация должна находиться на той или иной странице, до принятия решения, как эта страница должна выглядеть. Иногда полезно предоставить клиенту схему сайта, прежде чем показывать дизайн. Это позволяет клиенту убедиться, что сайт содержит всю необходимую информацию и функции.
Если вы просто представите клиенту дизайн сайта, то он, скорее всего, сосредоточится на его внешнем виде и не сумеет обнаружить возможные недостатки функциональности до того, как сайт будет готов.

Схема, показанная на следующем рисунке, была создана в программе Photoshop с использованием системы сетки 960.gs.

Вы можете создавать схемы на бумаге или использовать графический редактор (например, Illustrator или InDesign).

Существуют также онлайновые инструменты для создания схем: **gomockingbird.com**; **lovelycharts.com**.

ПРИМЕР СТРУКТУРНОЙ СХЕМЫ САЙТА



ПЕРЕДАЧА СООБЩЕНИЯ ПОСРЕДСТВОМ ДИЗАЙНА

Основная цель любого визуального дизайна — это передача сообщения. Организация информации и расставление приоритетов на странице помогают пользователям определить ее важность и понять, в каком порядке ее нужно читать.

КОНТЕНТ

Веб-страницы часто содержат очень много информации. Например, в главе интернет-газеты находятся элементы, отсутствующие на каждой из страниц ее печатного эквивалента:

- логотип;
- ссылки для навигации по сайту;
- ссылки на контент, имеющий отношение к конкретным материалам, и на другие популярные статьи;
- окно регистрации и авторизации пользователей;
- поле для ввода комментария;
- информация об авторских правах;
- ссылки на положения политики конфиденциальности, правила пользования сайтом, рекламную информацию, RSS-каналы и варианты подписки.

При таком объеме данных на странице дизайнер должен **организовать информацию** и **расставить приоритеты**, чтобы донести смысл сообщения и помочь пользователям найти то, что они ищут.

РАССТАНОВКА ПРИОРИТЕТОВ

Если вся информация на странице представлена в одном и том же стиле, то ее будет трудно воспринять, поскольку ключевое сообщение окажется слабовыделенным.

Делая части страницы **отличными** от окружающего контента, дизайнеры привлекают внимание к этим элементам (или наоборот). Дизайнеры создают так называемую **визуальную иерархию**, чтобы помочь пользователям сосредоточиться на привлекающих внимание ключевых моментах, а затем направить их к следующим сообщениям.

Мы поговорим о визуальной иерархии далее в этой главе.

ОРГАНИЗАЦИЯ

Группировка логически связанного контента в **блоки** упрощает восприятие страницы.

Пользователи должны быть в состоянии определить назначение любого блока без необходимости в обработке каждого отдельного элемента.

Когда вы представляете определенные типы информация в одном и том же стиле, например используете одно и то же оформление для всех кнопок или ссылок, пользователи начинают ассоциировать конкретный стиль с определенным типом контента.

О группировке и подобии мы поговорим далее в этой главе.

Давайте рассмотрим пример эффективного использования дизайна для передачи сообщения об услугах компании.

ВИЗУАЛЬНАЯ ИЕРАРХИЯ

Посетитель сразу обращает внимание на изображение, дающее представление об оказываемых компанией услугах, и на поясняющий заголовок. Размер и цветной фон указывают на то, что данное сообщение является главным на странице.
Если представленные услуги актуальны для посетителя, то ниже на странице он может получить более подробную информацию о том, сколько они стоят и кто ими пользуется.

ГРУППИРОВКА

Данная страница содержит несколько информационных блоков.
В верхней части находятся логотип и меню навигации. Под ними располагается описание услуг, предоставляемых компанией.
Еще ниже можно видеть три блока, содержащих более подробное описание услуг, цены и список некоторых пользователей.

ПОДОБИЕ

На этой странице присутствует несколько примеров подобия.
Четыре пункта (в левой нижней части рисунка) представлены похожим образом и содержат заголовки и значки одного стиля.
Все ссылки в тексте выделены синим цветом, чтобы было понятно, какой текст является интерактивным.

ВИЗУАЛЬНАЯ ИЕРАРХИЯ

Большинство веб-пользователей не читают страницы целиком. Вместо этого они быстро пробегают страницу глазами в поисках нужной им информации. Для создания визуальной иерархии вы можете использовать контрастность, которая позволит передать ваше ключевое сообщение и поможет пользователям найти то, что они ищут.

РАЗМЕР

Более крупные элементы будут привлекать внимание пользователей в первую очередь. По этой причине следует использовать заголовки и ключевые фразы относительно большого размера.

Lorem ipsum
dolor sit amet, consectetur adipiscing elit. Lorem ipsum dolor sit amet, consectetur adipscing elit.

ЦВЕТ

Основной и фоновый цвета могут акцентировать внимание на ключевых сообщениях. Более яркие разделы привлекают внимание пользователей в первую очередь.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Lorem ipsum dolor sit amet, consectetur adipscing elit.

СТИЛЬ

Элемент может быть такого же размера и цвета, как и окружающий контент, но при этом выделяться благодаря другому стилю.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Lorem ipsum dolor sit amet, consectetur adipscing elit.

Визуальная иерархия — это порядок, в котором ваши глаза воспринимают то, что они видят. Она создается путем добавления **визуального контраста** между отображаемыми элементами. Элементы, обладающие более высокой контрастностью, распознаются и обрабатываются в первую очередь.

ИЗОБРАЖЕНИЯ

Изображения создают высокую визуальную контрастность и воспринимаются в первую очередь. Они могут использоваться для привлечения внимание к конкретному сообщению на странице. В некоторых случаях правильно подобранное изображение способно передать больше смысла, чем целая страница текста.

Хорошо продуманная визуальная иерархия воздействует в основном на подсознание. Правильная иерархия требует соблюдения баланса. Отсутствие выделяющихся элементов может сделать сайт неинтересным, а наличие множества фрагментов, борющихся за внимание зрителя, затруднит нахождение ключевых сообщений. Приведенный пример обладает четкой иерархией, которая учитывает потребности посетителей сайта.

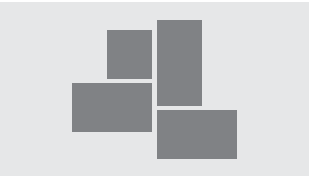
ГРУППИРОВКА И ПОДОБИЕ

При восприятии дизайна мы разделяем визуальные элементы на группы. Группировка логически связанных фрагментов помогает ориентироваться. Далее приведены некоторые способы достижения этой цели.

БЛИЗОСТЬ

Когда несколько элементов располагаются близко друг к другу, они воспринимаются как логически связанные, в отличие от элементов, находящихся далеко друг от друга. (Вы также можете вложить группу фрагментов информации в более крупную группу.)

ЗАВЕРШЕННОСТЬ

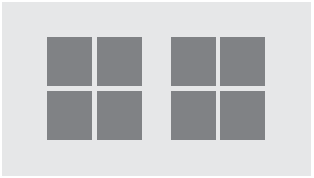


Когда нам попадаются сложно скомпонованные элементы, мы часто пытаемся найти единую или узнаваемую форму. На основании близости и выравнивания элементов их можно заключить в реальный или воображаемый блок.

ПОСЛЕДОВАТЕЛЬНОСТЬ

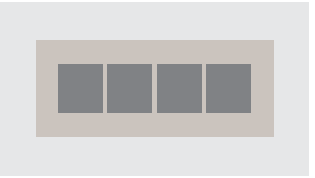
Когда элементы располагаются вдоль одной прямой или кривой линии, они воспринимаются как логически более связанные, чем те, которые не следуют в том же направлении. Эта особенность может использоваться, чтобы направлять внимание читателя от одной части страницы к другой.

ИНТЕРВАЛ



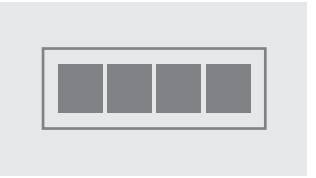
Размещение логически связанных элементов ближе друг к другу и создание большего интервала между несвязанными элементами.

ЦВЕТ



Связь между элементами можно подчеркнуть с помощью фонового цвета.

ГРАНИЦЫ



Вокруг группы элементов можно нарисовать границы, чтобы обозначить ее или отделить ее от соседних элементов.

Мы естественным образом замечаем подобие элементов дизайна, при этом похожие элементы воспринимаются как логически более связанные. Использование похожих цветов, размеров, ориентации, текстур, шрифтов или форм означает, что соответствующие элементы имеют близкое значение.

Фрагмент может иметь собственную иерархию, как показано в данном примере, где для каждой книги выделяются подразделы с названием, именем автора, текстом и ссылкой.

СОГЛАСОВАННОСТЬ

В этом примере обзоры книг оформлены в едином стиле, то есть название книги, имя автора и ссылки на сайт интернет-магазина выглядят одинаково. Прочитав один из блоков, можно сделать вывод о значении других элементов, оформленных в таком же стиле.

ЗАГОЛОВКИ

Заголовок позволяет пользователю определить, является ли содержимое фрагмента актуальным для него. Если нет, то он может проигнорировать все элементы в этом фрагменте. Заголовки также пригодятся пользователям программ экранного доступа.

РАЗРАБОТКА НАВИГАЦИИ

Меню для навигации по сайту не только помогает людям решить, куда они хотят пойти, но и позволяет им понять, чему посвящен ваш сайт и как он организован. Продуманная навигация характеризуется следующими чертами.

КРАТКОСТЬ

В идеале меню навигации должно быть кратким и легким для чтения. Следует ограничить количество вариантов меню до восьми. Они могут ссылаться на главные страницы разделов, содержащие, в свою очередь, ссылки на другие страницы.

ЯСНОСТЬ

Перед тем как щелкнуть по ссылке, пользователи должны иметь представление об информации, которую они найдут на следующей странице. По возможности используйте для каждой ссылки отдельные описательные слова, а не фразы.

ИЗБИРАТЕЛЬНОСТЬ

Главное меню навигации должно содержать только ссылки на основные разделы сайта. Окна поиска и авторизации, а также ссылки на юридическую информацию, вроде правил использования сайта, следует помещать отдельно от навигационного меню.

Главная Художники Выставки и события Галереи Книги и журналы
Об этом сайте Свяжитесь с нами Авторизация Регистрация
Правила использования сайта Политика конфиденциальности

Главная Художники Выставки Галереи Публикации О нас Контакты

Большие сайты могут иметь первичную, вторичную и даже третичную навигацию. Первичное навигационное меню часто располагается в верхней или левой части страницы. Вторичная навигация может находиться под основным навигационным меню или в боковой части страницы.

Третичная навигация часто располагается в нижнем колонтитуле страницы. Меню не должно быть единственным способом навигации по сайту. Пользователи могут также использовать ссылки на странице. Кроме того, некоторые сайты также предусматривают функцию поиска.

ПРИМЕЧАНИЕ

Посетите страницу **www.htmlandcssbook.com/extras/google-search/**, чтобы узнать о том, как использовать сервис Google Search для добавления на сайт поиска.

КОНТЕКСТ

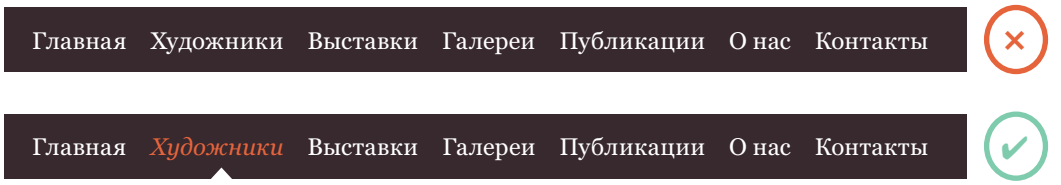
Продуманная навигация обеспечивает контекст. Она дает пользователю представление о том, в каком месте сайта он находится в данный момент. Это можно обеспечить путем использования цвета или другой визуальной метки для обозначения просматриваемой страницы.

ИНТЕРАКТИВНОСТЬ

Каждая ссылка должна иметь достаточные размеры, чтобы по ней можно было щелкнуть. Кроме того, внешний вид ссылки должен меняться при наведении на нее указателя мыши и при выполнении щелчка. Ссылка также должна зрительно отличаться от другого содержимого страниц.

ПОСЛЕДОВАТЕЛЬНОСТЬ

Чем больше страниц содержит сайт, тем больше элементов в навигационном меню. Хотя вторичная навигация будет изменяться от страницы к странице, первичную навигацию лучше всего оставить без изменений.



ОБЗОР
ПРОЦЕСС РАЗРАБОТКИ

- Очень важно понимать, кто является вашей целевой аудиторией, зачем они приходят на ваш сайт, какую информацию хотят найти и когда смогут вернуться.
- Карты сайта помогают разработать структуру сайта.
- Использование структурной схемы позволяет распределить информацию по страницам.
- Дизайн — это коммуникация. Визуальная иерархия помогает донести ваше сообщение до посетителей.
- Вы можете выделить фрагменты информации с помощью размера, цвета и стиля.
- Используйте группировку и подобие, чтобы упростить восприятие информация.

19

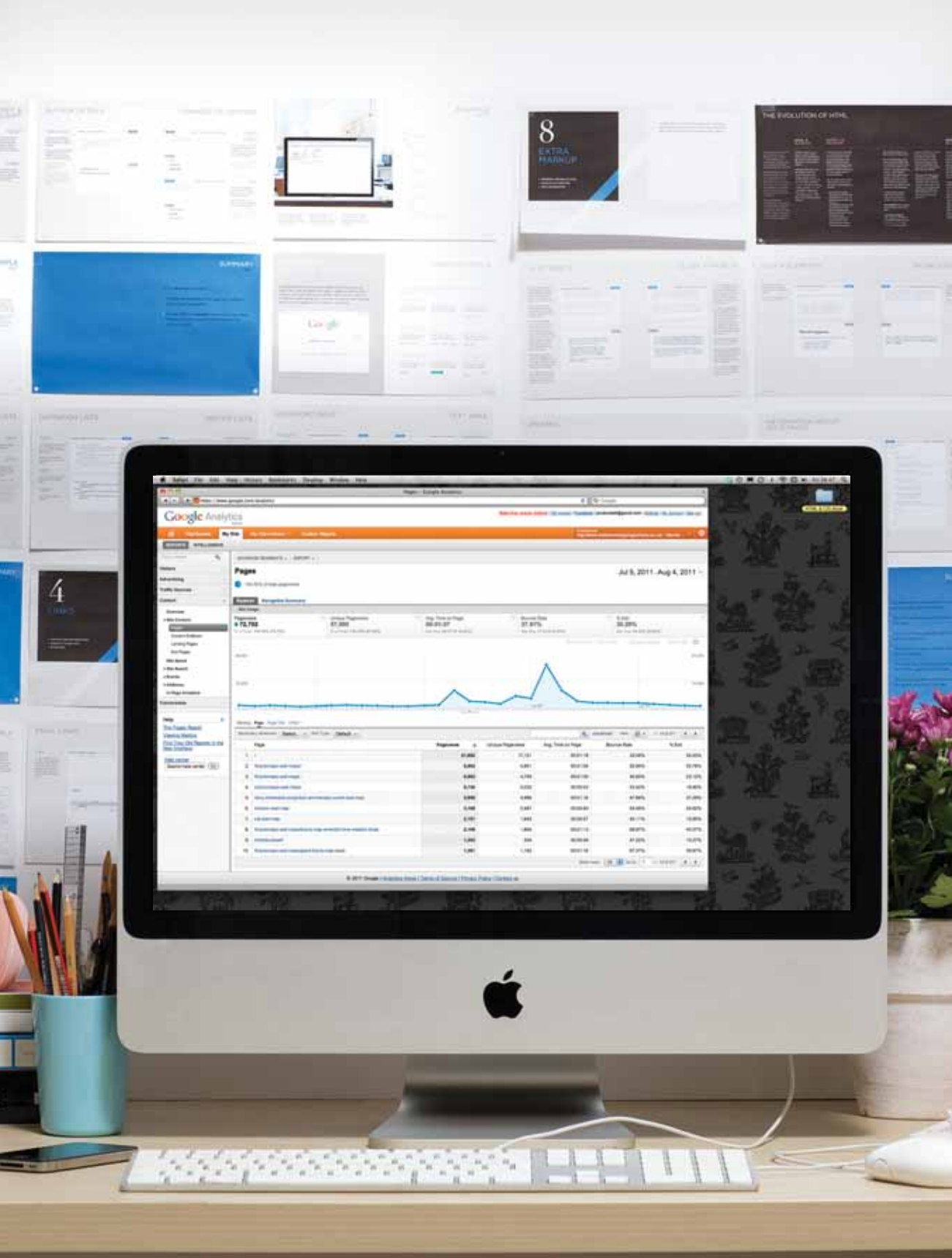
ПРАКТИЧЕСКАЯ ИНФОРМАЦИЯ

- Поисковая оптимизация
- Использование аналитических инструментов для изучения целевой аудитории
- Публикация сайта во Всемирной паутине

В заключение будет представлена некоторая практическая информация, которая поможет вам запустить успешный сайт.

Каждой теме данной главы посвящены целые книги, и, рассмотрев ключевые моменты этих тем, вы сформируете представление о том, с чем вам придется столкнуться. В данной главе вы:

- познакомитесь с основами поисковой оптимизации
- узнаете, как использовать инструмент Google Analytics, чтобы понять, что посетители делают на вашем сайте;
- изучите процесс публикации сайта во Всемирной паутине.


8
EXTRA
MARKUP
THE EVOLUTION OF HTML
4
Google Analytics
Pages
Jul 5, 2011 - Aug 4, 2011

ПОИСКОВАЯ ОПТИМИЗАЦИЯ (SEO)

Поисковая оптимизация — это очень объемная тема. На следующих страницах вы найдете информацию, которая поможет вам понять ключевые концепции данной темы и позволит улучшить позицию вашего сайта в поисковых системах.

ОСНОВЫ

Поисковая оптимизация (SEO) — это комплекс мер, направленных на повышение позиции сайта в поисковой выдаче, когда запрос соответствует его теме.

По сути, процесс SEO заключается в определении ключевых слов, часто используемых при поиске информации по теме, которой посвящен сайт, и в дальнейшем использовании этих слов в подходящих местах на сайте с целью увеличения вероятности того, что поисковые системы включат ссылку на него в свои выдачи.

При определении позиции в списке результатов поисковые системы оценивают не только содержимое сайта, но и количество ссылающихся на него других ресурсов (а также релевантность последних). По этой причине поисковая оптимизация часто предусматривает два направления: так называемые методы on-page и off-page.

МЕТОДЫ ON-PAGE

Методы on-page — это техники, используемые на странице для повышения ее рейтинга в поисковых системах.

В основе их лежит определение ключевых слов, используемых людьми при поиске информации по определенной теме, с дальнейшим включением этих слов в текст и HTML-код сайта, чтобы сообщить поисковым системам, что информация по данной теме содержится на вашем сайте.

Поисковые системы в основном полагаются на текст веб-страниц, поэтому очень важно включать в него те слова, которые с наибольшей вероятностью будут использоваться посетителями при поиске. Ключевые слова можно разместить в семи основных областях страницы.

Использование подходящего замещающего текста для графики также позволяет поисковым системам понять содержимое изображений.

МЕТОДЫ OFF-PAGE

Кроме использования методов on-page, важно сделать так, чтобы ссылки на ваш сайт присутствовали на сторонних ресурсах. При определении рейтинга поисковые системы также учитывают количество сайтов, ссылающихся на ваш. Наибольший вес имеют те из них, тема которых близка теме вашего сайта. Например, если вы продаете рыболовные снасти, то ссылка на ваш сайт, расположенная на веб-странице парикмахерской, вероятно, будет иметь меньший вес, чем ссылка на сайте сообщества рыболовов.

Поисковые системы также учитывают слова, расположенные между открывающим `<a>` и закрывающим `</a>` тегами ссылки. Если текст в ссылке содержит ключевые слова (а не просто фразу «нажмите здесь» или адрес вашего сайта), то она может рассматриваться как более релевантная.

Слова, указанные в ссылке на страницу вашего сайта, должны также присутствовать в тексте этой самой страницы.

ПОИСКОВАЯ ОПТИМИЗАЦИЯ ON-PAGE

Существует семь ключевых мест на веб-странице, в которых следует помещать ключевые слова для облегчения их обнаружения (ключевые слова — это слова, которые люди используют в поисковых запросах, разыскивая информацию по теме вашего сайта).

1: ЗАГОЛОВОК СТРАНИЦЫ

Заголовок страницы отображается в верхней части окна или на вкладке браузера и указывается в элементе `<title>`, который находится внутри элемента `<head>`.

2: URL-АДРЕС

Имя файла является частью URL-адреса. При возможности присваивайте осмысленные имена файлов.

3: ЗАГОЛОВКИ

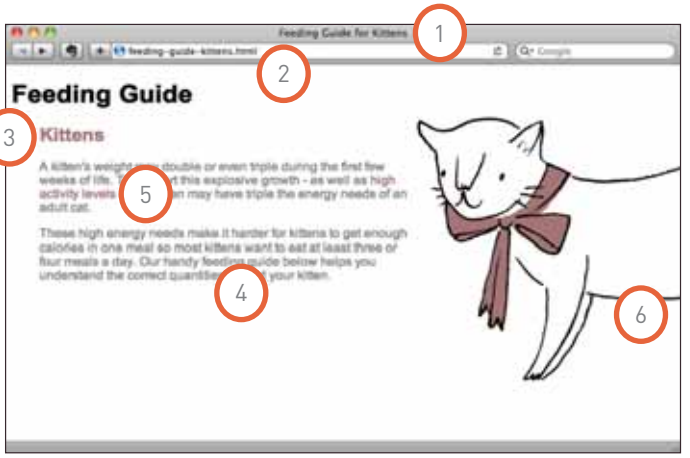
Если ключевые слова находятся в элементе `<h1>` — `<h6>`, то поисковая система придаст данной странице больший вес, поскольку посчитает, что она целиком посвящена конкретной теме.

4: ТЕКСТ

При возможности старайтесь использовать ключевые слова в основном тексте как минимум 2–3 раза. Однако не переусердствуйте, текст должен оставаться легким для восприятия.

5: ТЕКСТ ССЫЛОК

Используйте ключевые слова в тексте ссылок между



страницами, вместо того чтобы использовать такие общие выражения, как «нажмите здесь».

6: ЗАМЕЩАЮЩИЙ ТЕКСТ

Поисковые системы ожидают, что в качестве замещающего текста вы предоставите точное описание изображения. Благодаря этому ваши картинки будут отображаться в результатах поиска среди изображений.

7: ОПИСАНИЕ СТРАНИЦ

Описание также располагается внутри элемента `<head>` и указывается с помощью тега `<meta>`. Это должно быть предложение, которое коротко характеризует содержимое страницы. (Описание не отображается в окне браузера, но может присутствовать в результатах поиска.)

Никогда не пытайтесь обхитрить поисковые системы! За это бы будете наказаны. Например никогда не помещайте на странице текст, цвет которого совпадает с фоном страницы, поисковые системы способны это обнаружить.

КАК ВЫБРАТЬ КЛЮЧЕВЫЕ СЛОВА И ФРАЗЫ

Выбор ключевых слов для использования на сайте — одна из самых сложных задач поисковой оптимизации. Далее перечислены шесть этапов данного процесса.

1: МОЗГОВОЙ ШТУРМ

Перечислите слова, которые, по вашему мнению, станут использоваться при поиске информации по теме вашего сайта. Не забудьте указать различные темы, товары или услуги, представленные на вашем сайте. Спросите других людей, какие слова они использовали бы при поиске чего-нибудь по вашей теме. Менее знакомые с нею люди могут включить в поисковый запрос слова, отличные от выбранных вами. В частности, они вряд ли будут использовать специальные термины. Ваш список может содержать не просто отдельные слова, а целые ключевые фразы, если у вас есть темы, которые нельзя описать одним словом.

2: ОРГАНИЗАЦИЯ

Сгруппируйте ключевые слова в отдельные списки для разных разделов или категорий вашего сайта. Например, если ваш сайт представляет собой интернет-зоомагазин, то вы можете предусмотреть отдельную категорию для каждого животного (например, для собак, кошек и кроликов). В случае с большим сайтом вы можете разбить эти категории на подкатегории (например, на основе марок кормов для домашних животных).

3: ИССЛЕДОВАНИЕ

Существуют инструменты, которые предлагают варианты ключевых слов в дополнение к введенным вами, например **adwords.google.co.uk/select/KeywordToolExternal** — при использовании этого инструмента выбирайте параметр exact match («точное соответствие») вместо broad match («неточное соответствие»); **www.wordtracker.com**; **www.keyworddiscovery.com**. После того как с помощью этих инструментов вы найдете дополнительные ключевые слова, добавьте самые подходящие из них в свои списки. (Среди предложенных вариантов, скорее всего, будут слова, не имеющие отношения к вашей теме, поэтому смело исключайте те, которые вам не подходят.)

4: СРАВНЕНИЕ

Очень маловероятно, что ваш сайт будет отображаться в верхней части поисковой выдачи по каждому ключевому слову. Это особенно касается популярных тем. Чем больше сайтов оптимизировано под определенное ключевое слово, тем труднее вам будет повысить рейтинг вашего в поисковой выдаче по данному слову.
Некоторые сервисы для исследования ключевых слов могут сообщить вам, сколько людей запрашивали то или иное слово. Это даст вам представление об уровне конкуренции.
Вы также можете использовать расширенные функции сервиса Google для произведения поиска только по названию веб-страниц. Это позволит вам определить количество сайтов, содержащих данное ключевое слово в названии своих страниц. Чем больше страниц содержат в названии ключевое слово, тем выше уровень конкуренции.

5: УТОЧНЕНИЕ

Теперь вам нужно выбрать ключевые слова, на которых следует сосредоточиться. Это должны быть слова, имеющие наибольшее отношение к разделам вашего сайта.
Если вы нашли подходящую для своего сайта фразу, но уровень конкуренции по ней очень высок, вам все равно следует ее использовать. Вы также можете попробовать включить в эту фразу другие слова. Например, если информация или предлагаемая вами услуга зависит от конкретного местоположения, то указание местоположения в ключевой фразе приведет к тому, что вас найдут именно те люди, которым вы больше всего нужны.
Если ваш сайт продвигает услуги австралийской кровельной компании, то для вас будет гораздо лучшим вариантом, если удастся привлечь 100 клиентов из Австралии, чем 10 000 из США.

6: РАСПРЕДЕЛЕНИЕ

Теперь, когда у вас есть уточненный список ключевых слов, уровни их конкуренции и вы знаете, какие из них являются наиболее актуальными, пришло время выбрать ключевые слова для каждой из страниц.
Возьмите 3–5 слов или фраз для каждой страницы вашего сайта и используйте их в качестве ключевых.
Не следует повторять одни и те же ключевые слова на всех страницах. Вполне вероятно, что чем дальше вы уйдете от главной страницы сайта, тем более конкретными станут ключевые слова.

АНАЛИТИКА: ИЗУЧЕНИЕ АУДИТОРИИ САЙТА

Как только люди начнут приходить на ваш сайт, вы сможете начать анализировать, как они его нашли, что просматривали и с какой страницы покинули сайт. Одним из лучших инструментов для проведения такого рода анализа является бесплатный сервис Google Analytics.

РЕГИСТРАЦИЯ

Чтобы воспользоваться сервисом Google Analytics, необходимо зарегистрировать учетную запись на сайте **www.google.com/analytics**. Сервис предоставит вам код отслеживания, который нужно будет поместить на каждую страницу вашего сайта.

ПРИНЦИП РАБОТЫ

Каждый раз, когда кто-то загружает страницу вашего сайта, код отслеживания отправляет данные на серверы Google. Используя веб-интерфейс Google, вы можете увидеть, как посетители используют ваш сайт.

КОД ОТСЛЕЖИВАНИЯ

Код отслеживания предоставляется сервисом Google Analytics для каждого сайта, за которым вы наблюдаете. Данный код необходимо поместить непосредственно перед закрывающим тегом `</head>`. Код отслеживания не влияет на внешний вид веб-страниц.

СКОЛЬКО ЛЮДЕЙ ПОСЕЩАЕТ ВАШ САЙТ?

Обзорная страница содержит ключевую информацию, которую вы, скорее всего, захотите получить. В частности, на этой странице вы узнаете, сколько людей посещает ваш сайт.

КОЛИЧЕСТВО ПОСЕЩЕНИЙ

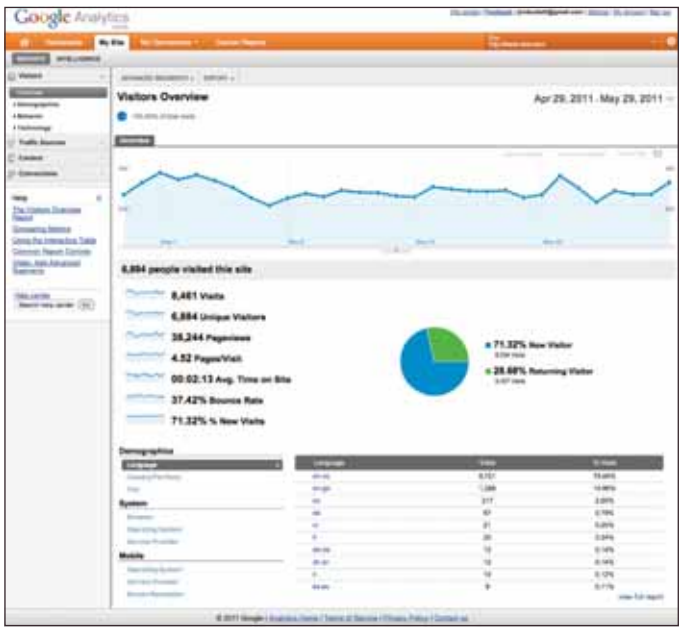
Указывает, сколько раз посетители заходили на ваш сайт. Если посетитель в течение 30 минут не проявляет активности, а затем переходит на другую страницу вашего сайта, то этот переход будет считаться новым посещением.

КОЛИЧЕСТВО УНИКАЛЬНЫХ ПОСЕТИТЕЛЕЙ

Это общее количество людей, зашедших на ваш сайт за определенный период. Число уникальных посетителей будет меньше числа посещений, если в течение указанного периода времени люди заходили на сайт более одного раза.

КОЛИЧЕСТВО ПРОСМОТРЕННЫХ СТРАНИЦ

Общее количество страниц вашего сайта, просмотренных всеми посетителями.



КОЛИЧЕСТВО СТРАНИЦ, ПРОСМАТРИВАЕМЫХ ЗА ОДНО ПОСЕЩЕНИЕ

Среднее количество страниц, просматриваемых каждым посетителем за одно посещение.

СРЕДНЕЕ ВРЕМЯ ПРЕБЫВАНИЯ НА САЙТЕ

Средняя продолжительность времени, проведенного каждым посетителем на сайте за одно посещение.

ВЫБОР ДАТЫ

Используя функцию выбора даты в верхнем правом углу страницы, вы можете изменить период отчета. Как правило, в качестве периода отчета устанавливается прошлый месяц, однако вы можете указать любой другой отрезок времени.

ЭКСПОРТ

Ссылка для экспорта, расположенная над заголовком visitors overview («обзор сведений о посетителях») позволяет сохранить статистику с этой страницы для работы в других приложениях, например в Microsoft Excel.

ЧТО ИМЕННО ПРОСМАТРИВАЮТ ПОСЕТИТЕЛИ ВАШЕГО САЙТА?

Вы можете узнать о том, что именно просматривают посетители, когда заходят на ваш сайт, щелкнув по соответствующей ссылке в сервисе Google Analytics.

СТРАНИЦЫ

Здесь указаны самые популярные страницы, а также страницы, на которых посетители проводят больше всего времени.

ЦЕЛЕВЫЕ СТРАНИЦЫ

Другими словами, страницы, на которые попадают посетители при переходе на ваш сайт. Это очень полезный показатель, поскольку люди не всегда начинают просматривать сайт с главной страницы.

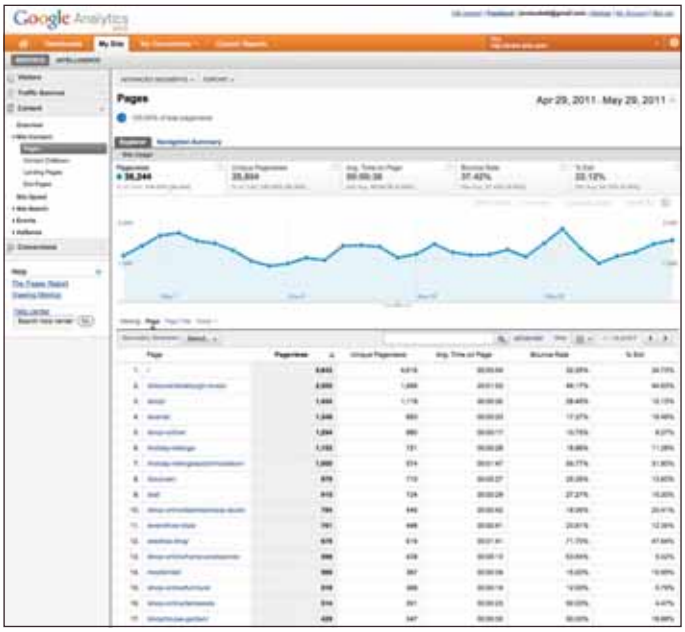
САМЫЕ ПОПУЛЯРНЫЕ СТРАНИЦЫ ВЫХОДА

Это страницы, с которых люди чаще всего покидают ваш сайт. Если вы замечаете, что посетители уходят с какой-то определенной страницы, то вы можете постараться изменить или улучшить ее.

ПОКАЗАТЕЛЬ ОТКАЗОВ

Отражает число людей, покинувших сайт со страницы, на которую они попали при переходе на ваш сайт. Высокое значение показателя отказов говорит о том, что содержимое сайта не соответствует тому, что искали посетители, или что целевая страница не побудила их к продолжению просмотра остальной части сайта. Отказом считается:

- щелчок по ссылке на другой сайт;
- щелчок по рекламному объявлению;
- ввод нового URL-адреса;
- использование кнопки «Назад»;
- закрытие браузера.



ОТКУДА ПРИХОДЯТ ВАШИ ПОСЕТИТЕЛИ?

Ссылка Traffic sources (Источники трафика) позволяет узнать, откуда приходят ваши посетители.

ПЕРЕХОД ПО РЕФЕРАЛЬНОЙ ССЫЛКЕ

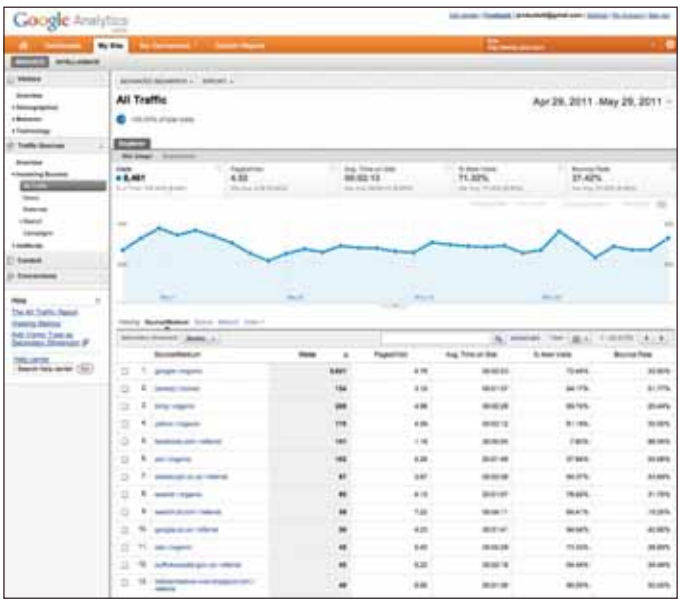
Здесь перечислены все сайты, содержащие ссылки на ваш, а также количество посетителей, перешедших по этим ссылкам. Если с какого-либо сайта к вам переходит большое количество людей, имеет смысл связаться с его владельцем и принять совместные меры для поддержания существующего трафика. Вы также можете поискать похожие сайты и попросить их владельцев поместить ссылку на вас.

ПРЯМОЙ ПЕРЕХОД

Здесь указаны страницы, на которые попадали пользователи в тех случаях, когда для перехода на ваш сайт они водили URL-адрес в адресную строку своего браузера, использовали закладку или переходили по ссылке в электронном письме, PDF-файле или документе Word.

ПОИСКОВЫЕ ЗАПРОСЫ

Здесь перечислены слова, используемые посетителями в поисковых системах для нахождения вашего сайта. Они позволяют узнать, как посетители описывают то, что они ищут (часто это отличается от того, каким образом владелец описывает собственный сайт). Информация в данном разделе позволит вам доработать контент и уточнить ключевые слова, используемые для поисковой оптимизации.



РАСШИРЕННЫЕ ФУНКЦИИ

Вы лишь поверхностно ознакомились в возможностями сервиса Google Analytics. Расширенные функции описаны в файлах справки. Если вы управляете интернет-магазином, следует изучить функцию, предназначенную для электронной коммерции, которая предоставляет информацию о реализованной продукции, средней стоимости произведенных покупок и т.д. Вы также можете указать пути, по которым должны следовать ваши посетители, а затем смотреть, как далеко они по ним продвинулись. Это бывает особенно полезно при сборе информации о пользователях.

ДОМЕННОЕ ИМЯ И ХОСТИНГ

Чтобы опубликовать свой сайт во Всемирной паутине, вам потребуется доменное имя и веб-хостинг.

ДОМЕННОЕ ИМЯ

Ваше доменное имя — это веб-адрес вашего сайта (например, google.ru или yandex.ru). Существует множество сайтов, позволяющих регистрировать домены. Обычно вам необходимо ежегодно совершать взнос, чтобы сохранить зарегистрированное доменное имя за собой.

Такие сайты обычно предусматривают форму, позволяющую проверить, не занято ли интересующее вас имя. Поскольку миллионы доменных имен уже зарегистрированы, поиск подходящего может занять некоторое время.

Многие сайты, позволяющие зарегистрировать домен, также предлагают услуги веб-хостинга.

ВЕБ-ХОСТИНГ

Чтобы другие люди смогли посетить ваш сайт, вам необходимо загрузить его на веб-сервер. Веб-серверы — это специальные компьютеры, постоянно подключенные к сети Интернет. Они специально настроены на обслуживание запрашиваемых веб-страниц.

За исключением некоторых очень крупных веб-ресурсов, большинство сайтов размещены на веб-серверах, управляемых хостинговой компанией.

Это, как правило, намного дешевле и надежнее, чем использовать собственный веб-сервер.

Набор услуг веб-хостинга зависит от конкретной компании. Далее мы рассмотрим некоторые ключевые моменты, на которые вам следует обратить внимание при выборе хостинг-провайдера.

ОБЪЕМ ДИСКОВОГО ПРОСТРАНСТВА

Это общий размер всех файлов, составляющих ваш сайт (все HTML- и CSS-файлы, изображения и скрипты).

ПРОПУСКНАЯ СПОСОБНОСТЬ

Это объем данных, посылаемых хостинг-провайдером посетителям вашего сайта. Если представить, что 10 человек одновременно просматривают все страницы вашего сайта, то это эквивалентно десятикратному объему дискового пространства, которое вы используете.

РЕЗЕРВНЫЕ КОПИИ

Узнайте, выполняет ли хостинговая компания резервное копирование сайта (и если да, то как часто). Некоторые провайдеры создают резервные копии только для использования при нарушениях в работе сервера. Другие предоставляют доступ к резервной копии, что может быть полезно, если вы сами случайно нарушите работу сайта при обновлении.

ПОЧТОВЫЕ ЯЩИКИ

Большинство хостинговых компаний включают в пакет услуг возможности использования почтовых серверов. Узнайте доступный размер и количество почтовых ящиков, которые вы можете использовать.

СЕРВЕРНЫЕ ЯЗЫКИ ПРОГРАММИРОВАНИЯ И БАЗЫ ДАННЫХ

Если вы применяете систему управления контентом, она, скорее всего, использует язык серверного программирования и базу данных (например, язык PHP с базой данных MySQL или язык ASP.Net с базой данных SQL Server). Удостоверьтесь, что ваш хостинг-провайдер поддерживает технологии, используемые этим программным обеспечением.

Вам также стоит ознакомиться с отзывами о работе хостинговой компании. К сожалению, зачастую оценить работу хостинг-провайдера можно только при возникновении проблемы, так что вполне вероятно, вы найдете негативные отзывы о работе любой компании.

ВЕБ-ПЛАТФОРМЫ

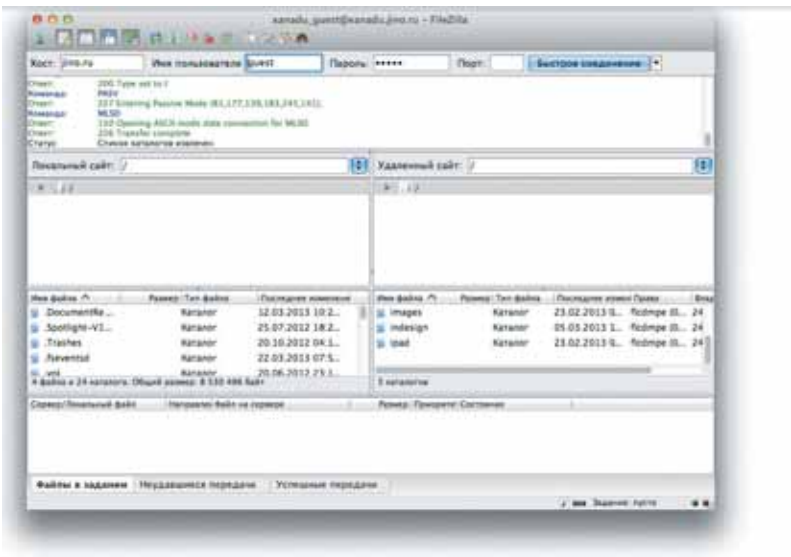
Существует целый ряд интернет-сервисов, позволяющих зарегистрировать ваше доменное имя на их сервере. Такие блог-платформы, как WordPress.com и Tumblr, а также платформы электронной коммерции, например, Big Cartel и Shopify, позволяют разместить ваш сайт на своих серверах. Использование подобных платформ устраняет необходимость в хостинге для сайта, однако вам, скорее всего, понадобится почтовый хостинг. В этом случае вы можете воспользоваться услугами отдельного провайдера, предоставляющего почтовый сервис.

FTP И ИНСТРУМЕНТЫ СТОРОННИХ РАЗРАБОТЧИКОВ

Для переноса кода и изображений с компьютера на сервер хостинговой компании используется протокол передачи файлов.

Как следует из названия, File Transfer Protocol* (FTP) позволяет передавать файлы через Интернет с вашего компьютера на веб-сервер хостинговой компании.

Существует множество **FTP-клиентов**, предоставляющих простой интерфейс, который отображает файлы на вашем компьютере и на веб-сервере. Эти программы позволяют копировать файлы с вашего компьютера на сервер или наоборот путем перетаскивания.



* File Transfer Protocol (*англ.*) — протокол передачи файлов. — *Примеч. ред.*

Существует множество сайтов, предоставляющих разработанные профессионалами инструменты, что устраняет необходимость создавать их самостоятельно.

Некоторые хостинг-компании предлагают инструменты для загрузки файлов на их серверы с помощью веб-браузера, однако чаще для этой цели используется FTP-программа, обеспечивающая более высокую скорость передачи.
При покупке услуг веб-хостинга вы получите данные FTP, которые вам нужно ввести в вашу FTP-программу для подключения к серверу. Чаще всего это будет адрес (например, ftp://mydomain.com), имя пользователя и пароль. Очень важно хранить такого рода информацию в тайне, чтобы не позволить посторонним людям получить доступ к вашему серверу.

Ниже перечислены несколько популярных FTP-программ.

FileZilla
filezilla-project.org
Windows, OS X, Linux

FireFTP
fireftp.mozdev.org
Windows, OS X, Linux

CuteFTP
cuteftp.com
Windows, OS X

SmartFTP
smartftp.com
Windows

Transmit
panic.com/transmit
OS X

А вот инструменты сторонних разработчиков.

БЛОГИ
wordpress.com;
tumblr.com;

ЭЛЕКТРОННАЯ КОММЕРЦИЯ
shopify.com;
bigcartel.com;
go.magento.com.

РАССЫЛКА ЭЛЕКТРОННОЙ ПОЧТЫ
campaignmonitor.com;
mailchimp.com.

СРЕДСТВА ДЛЯ ОРГАНИЗАЦИИ СОВМЕСТНОЙ РАБОТЫ
addthis.com;
addtoany.com.

ОБЗОР
ПРАКТИЧЕСКАЯ ИНФОРМАЦИЯ

- Поисковая оптимизация позволяет посетителям найти ваш сайт, используя поисковые системы.
- Аналитические инструменты, например Google Analytics, позволяют узнать, сколько людей посещают ваш сайт, как они его находят и что делают на нем.
- Чтобы опубликовать сайт во Всемирной паутине, вам необходимо доменное имя и веб-хостинг.
- FTP-программы позволяют передавать файлы с локального компьютера на веб-сервер.
- Многие компании предоставляют платформы для блогов, рассылки электронной почты, электронной коммерции и другие популярные инструменты, так что вам нет нужды разрабатывать их с нуля.

ПРЕДМЕТНЫЙ УКАЗАТЕЛЬ

- А–Я
- Решение часто возникающих проблем
- Элементы и атрибуты HTML, свойства CSS

<!-- --> 173
960.GS 368
!DOCTYPE 172

C

CDN (Content Delivery Network) 261
CSS, каскадные таблицы стилей 214
CSS-фреймворк 368
  960.GS 368

E

Em, единица измерения 257

F

FileTransfer Protocol 458
Flash
  аудиоплееры 207
  добавление на страницу 196
  определение 190
  подготовка видео 200
  проигрыватели FLV 200

G

GIF-анимация 114
Google Analytics, сервис 452

H

hCard 58
HSLA 234
HSL, цветовая модель 240
HTML
  История языка 170
  определение 33
HTML5
  группы заголовков 411
  заголовок 406
  меню навигации 407
  нижний колонтитул 406
  раздел 410
  рисунки 412
  статья 408
  стороннее содержимое 409
HTML5 shim 415, 477
HTML5 shiv 415, 477
HTML-атрибуты
  href 222
  rel 222
  type 222

S

src, атрибут 260
SWFObject 196, 197, 200, 201, 207, 211

T

TextWrangler
  текстовый редактор MacOS 36

А

Абсолютное позиционирование 340, 344
Абсолютный URL адрес 78
Альфа-канал 239
Атрибут
  определение 30

Б

Блочная модель IE6 297
Блочные элементы 216

В

Веб-сервер 456
Веб-хостинг 456
Вертикальное выравнивание 269
Вертикальные поля 289
Верхний выносной элемент 266
Визуальная иерархия 437, 438
  изображения 439
  размер 438
  стиль 438
  цвет 438
Визуальные редакторы 52
Встроенные элементы 216
Выравнивание
  вертикальное 269
Выравнивание изображений 386
Выравнивание текста 268
  по левому краю 268
  по правому краю 268
  по центру 268

Г

Гиперссылки 74
Гипертекст
  определение 33
Глобальные атрибуты
  определение 174
Горизонтальная линия 51
Градиенты 394
Граница
  стиль 291
  толщина 290
  цвет 292
Группировка 437

Д

Добавления и удаления 59
Доменное имя 456

З

Заголовки 46
Закругленные углы 302
Значения RGB 237

И

Изображение
  выравнивание по вертикали 102
  размеры 110
  формат JPEG 106
  форматы GIF/PNG 109
Изображения
  векторная графика 113
  выравнивание 386
  выравнивание по горизонтали 100
  высота и ширина 97
  градиенты 394
  добавление на страницу 96
  загрузка на компьютер 116
  контрастность фонового 395
  обрезка 111
  повторяющиеся 389
  подписи к изображениям (HTML5) 117
  положение фонового 390
  проверка размеров 116
  прозрачность 115
  размер 384
  разрешение (ppi) 112
  ролловеры 392
  спрайт 392
  фоновые 388
  формат SVG 113
  центрирование 387
Индекс
  верхний 49
  нижний 49
Информация о странице 182

К

Карта сайта 432
Карточная сортировка 432
Каскадирование правил CSS 226
Каскадные таблицы стилей (CSS) 214
  внешние 222, 228
  внутренние 223, 228
  правило 218
  совместимость с браузерами 229
Кернинг 267
Код отслеживания 452
Контрастность фоновых изображений 395
Корневой каталог
  определение 80

Л

Легенда 323

М

Макет
  блочные элементы 338
  встроенные элементы 338
  резиновый или жидкий 359, 362
  сетка 364
  фиксированной ширины 358, 360
Макетная сетка 364
Маркер-изображение 313
Маркеры 312
Метаинформация о странице 182
Многоколоночный макет 352

Н

Набор полей 323
Навигация, разработка 442
Названия цветов 237
Насыщенность 237, 240
Неупорядоченный список 312
Нижний выносной элемент 266
Нормальный поток 340, 342

О

Ограничение высоты 286
Ограничение ширины 285
Онлайн-тестирование сайта 229
Определения 57
Относительное позиционирование 340, 343
Относительный URL-адрес 79
Отступы 294
Отступы в тексте 270
Ошибка CSS 229

П

Перекрывающиеся элементы 346
Перетекание содержимого 287
Перечеркивание 59
Пиксел 236
Плавающие фреймы 180
Плавающие элементы 341, 347
Повторяющиеся изображения 389
Подчеркивание и перечеркивание текста 265
Позиционирование 340
Поисковая оптимизация (SEO) 448
  off-page методы 448
  on-page методы 448
Положение фонового изображения 390
Поля 295
Правила CSS
  блок объявлений 218
  наследование 227
  принципы каскадирования 226
  селектор 218

Применение стилей
к ссылкам 273
Принципы группировки
элементов
близость 440
границы 440
завершенность 440
интервал 440
последовательность 440
цвет 440
Пробелы 289
Проблемы совместимости
CSS с браузерами
229
Проверка введенных
данных 158
Прописные и строчные
символы 264
Просмотр исходного кода
40
Протокол передачи файлов,
FTP 458
Псевдокласс 272
Псевдоэлемент 272
Пункт 258
Пустые элементы
определение 51

P

Размер изображения 384
Размер рамки 284
единицы измерения 284
Размер страницы 356
Размер шрифта
в единицах em 257
в пикселах 257
в процентах 257
Разрешение экрана 355
Разрыв строки 51
Рамка
высота 284
граница 288
декоративная 300
закругленные углы 302
отступы 288
поля 288
размер 284
тень от 301
толщина границы 290
Реагирование на действия
пользователя 274
Ролловер 392

C

Сайт
карта 432
поисковая оптимизация
448
структурная схема 434
целевая аудитория 426
Светимость 240
Светлота 240
Свойства CSS
right 343
Свойства
border-width 290
Свойства CSS
background 391
background-attachment
389
background-image 388
background-image
(градиент) 394
background-position 390
background-repeat 389
border 293, 321
border-bottom-color 292
border-bottom-left-radius
302
border-bottom-right-radius
302
border-bottom-style 291
border-bottom-width 290
border-collapse 319
border-color 292
border-image 300
border-left-color 292
border-left-style 291
border-left-width 290
border-radius 302, 321
border-right-color 292
border-right-style 291
border-right-width 290
border-spacing 319
border-style 291
border-top-color 292
border-top-left-radius 302
border-top-right-radius 302
border-top-style 291
border-top-width 290
border-width 290
box-shadow 301
clear 349
color 234
cursor 326
display 298
empty-cells 318
float 347
@font-face 254
font-face 254
font-family 254
font-size 257
font-style 263
height 384
hsl 241
left 343
letter-spacing 264
line-height 266
list-style 315
list-style-image 313
list-style-position 314
list-style-type 312
margin 295
margin-bottom 295
margin-left 295
margin-right 295
margin-top 295
max-width 285
min-height 286
min-width 285
-moz-box-shadow 301
opacity 239
overflow 287
padding 294
padding-bottom 294
padding-left 294
padding-right 294
padding-top 294
padding (таблица) 316
position 343
rgba 239
text-align 268
text-decoration 265
text-indent 270
text-shadow 271
text-transform 264
top 343
vertical-align 269
visibility 299
-webkit-box-shadow 301

width (плавающие блоки) 348
width (таблица) 316
word-spacing 267
z-index 346
Свойства смещения 343
Сворачивание пробела 50
Селекторы CSS 224
Селекторы атрибутов 275
Семантическая разметка
определение 44
Сеть доставки (и дистрибуции) контента (CDN) 261
Система сеток 960 364
Скрытие рамок 299
Смещение, свойства 343
Содержимое сайта
визуальная иерархия 437, 438
группировка 437, 440
подобие 437, 440
Сокращения или аббревиатуры 56
Списки вложенные 69
Список
маркер 312
неупорядоченный 312
нумерация 312
упорядоченный 312
Список маркированный
определение 64
Список нумерованный
определение 64
Список определений
определение 64
Способы задания цвета в CSS 234
Способы использования шрифта
изображение 255
технология Cufon 255
технология sIFR 255
Спрайт 392
Ссылка
применение стилей к 273
Ссылки 74
Ссылки на электронную почту 84
Стековый контекст 346
Стенографические свойства 293
Стиль границы 291
Структура веб-страниц 25
Структурная разметка
определение 44
Структурная схема 434
Схема позиционирования
абсолютное позиционирование 340, 344
нормальный поток 340, 342
относительное позиционирование 340, 343
плавающие элементы 341
Схемы позиционирования 340

Т

Таблица
интервал между ячейками 319
пустые ячейки 318
свойства 316
Таблицы
длинные таблицы 130
объединение столбцов 128
объединение строк 129
создание 126
строки 126
ячейки 126
Теги
определение 25
структура 28
Текст
выравнивание 268
с отступом 270
с тенью 271
Тень текста 271
Типографическая шкала 258

У

Указатель мыши 326
Упорядоченный список 312
Условный комментарий 415

Ф

Фиксированное позиционирование 341, 345
Фоновые изображения 388
Форма
кнопка отправки данных 322
легенда 323
набор полей 323
поле для ввода текста 321
применение стилей 320
Форматы видео 198
H264 202
WebM 202
Форматы шрифтов 261
Формы
ввод URL/email-адреса (HTML5) 160
ввод даты (HTML5) 159
группировка элементов управления 157
загрузка файлов на сервер 152
кнопка подтверждения 153, 154
обзор элементов управления 141
переключатели 148
подпись элементов управления 156
поле ввода пароля 146
поле ввода поискового запроса (HTML5) 161
поле ввода текста 145
поле ввода текста (многострочное) 147
Проверка введенных данных (HTML5) 158
раскрывающийся список 150
скрытые элементы 155
структура 144
флажки 149

Ц

Цвет
  значения RGB 237
  название 237
  насыщенность 237, 240
  светимость 240
  светлота 240
  тон 237, 240
  шестнадцатеричные коды 237
  яркость 237
Цвет в CSS
  HSLA 234
  значения RGB 234
  названия цветов 234
  основной цвет 234
  способы задания цвета 234
  шестнадцатеричные коды 234
Цвет границы 292
Цветовой тон 237, 240
Целевая страница 454
Центрирование изображений 387
Центрирование содержимого 296
Цитаты 55

Ш

Шестнадцатеричные коды 237
Шрифт
  без засечек 250, 252
  верхний выносной элемент 266
  вес 251
  декоративный 253
  кернинг 267
  курсив 48
  курсивный 253
  межстрочный интервал 266
  моноширинный 250, 253
  нижний выносной элемент 266
  перечеркнутый 59
  подчеркивание и перечеркивание 265
  подчеркнутый 59
  полужирный 48
  последовательность шрифтов 252
  прописные и строчные символы 264
  пункт 258
  растяжение 251
  сжатие 251
  с засечками 250, 252
  стиль 251
  термины 250
  форматы 261
  шкала размеров 258
Шрифт, курсив 263

Э

Экран
  размеры 354
  разрешение 355
Элемент
  <a> 76, 78
  <abbr> 56
  <acronym> 56
  action 144
  <address> 58
  align 100
  alt 96
  auto 203
  autoplay 203, 208
  <b> 48
  <blockquote> 55
  <body> 27, 32

 51
  <button> 155
  <cite> 57
  codecs 204
  controls 203, 208
  <dd> 68
  <del> 59
  description 182
  <dfn> 57
  <dl> 68
  <dt> 68
   54
  expires 183
  <fieldset> 157
  <form> 144
  frameborder 181
  <h1> 46
  <h2> 46
  <head> 32
  height 97, 180, 203
  <hr /> 51
  <html> 27
  <i> 48
  id 144
  <iframe> 180
  <img> 96
  <ins> 59
  keywords 182, 183
  <label> 156
  <legend> 157
   66, 67
  loop 203, 208
  <meta> 182
  metadata 203
  method 144
  none 203
   66
  <p> 47
  poster 203
  pragma 183
  preload 208
  preloaded 203
  <q> 55
  <s> 59
  <script> 197
  scrolling 181
  seamless 181
   179
  src 96, 180, 203, 204, 208, 209
   54
  <sub> 49
  <sup> 49
  <table> 126
  <tbody> 130
  <td> 126
  <tfoot> 130
  <th> 127
  <thead> 130
  <title> 32
  title 96
  <tr> 126
  type 204, 209
  <u> 59
   67
  <video> 203
  width 97, 180, 203

блочный элемент 99
внутристрочный элемент 99
Элемент 352
элемент Html
определение 25
Элемент HTML5
<audio> 208
<figcaption> 117
<figure> 117
<source> 204, 209
<video> 202
Элемент <link> 222
атрибуты 222
Элемент-контейнер 339
Элемент разметки
определение 33
Элементы
блочные 216
блочные элементы 176
внутристрочные элементы 177
встроенные 216
группировка блочных элементов 178
Эллиптические формы 303

Я

Яркость 237

РЕШЕНИЕ ЧАСТО ВОЗНИКАЮЩИХ ПРОБЛЕМ

Далее приведено описание часто возникающих проблем, а также указаны разделы книги, с помощью которых их можно решить.

САМОЕ ОСНОВНОЕ

Браузер отображает разметку, а не готовую страницу.
Убедитесь, что файл имеет расширение *.html*, а не *.txt*.
См. раздел «Создание веб-страницы в операционной системе Windows» главы 1.

Форматирование (полужирный шрифт, курсив, заголовки или текст ссылок) применено к большему объему текста, чем ожидалось.
Убедитесь, что вы закрыли соответствующий тег, например, поставили `</a>`.
См. раздел «Язык HTML описывает структуру страниц с помощью элементов» главы 1.

Страница сайта, на которую указывает ссылка, не найдена.
Проверьте относительный URL-адрес.
См. раздел «Относительные URL-адреса» главы 4.

ИЗОБРАЖЕНИЯ

Изображения не видны.
Проверьте относительный URL-адрес.
См. раздел «Относительные URL-адреса» главы 4.

Изображения размыты.
Убедитесь, что изображения сохранены в нужном формате и размере.
См. раздел «Формат изображений: JPEG» главы 5.

Изображения для границ не видны.
Данная функция поддерживается только последними версиями браузеров. (Другой вариант: проверьте относительный URL-адрес).
См. раздел «CSS3: декоративные границы» главы 13.
См. раздел «Относительные URL-адреса» главы 4.

В GIF-файлах, которые были отмасштабированы, появилась зернистость или зазубренные края.
Проверьте цветовое пространство в графическом редакторе. Должно использоваться цветовое пространство RGB, а не индексированные цвета.

ТЕКСТ

Текст мерцает, прежде чем отобразить нужный шрифт.
Это так называемая «вспышка нестилизованного текста».
См. раздел «Форматы шрифтов» главы 12.

При копировании текста из текстового редактора в разметке появляется большое количество дополнительных тегов.
Скопируйте текст в редактор неформатированного текста (для удаления форматирования), а затем скопи-

руйте уже оттуда и вставьте в HTML-редактор.
См. раздел «Визуальные редакторы и режим просмотра кода» главы 2.

На некоторых экранах текст выглядит больше/меньше.
Обычно это связано с разницей в разрешениях экранов. Кроме того, проверьте, какие единицы измерения используются для задания размера шрифта в элементе `<body>`.
См. раздел «Размеры экранов устройств» главы 15.
См. раздел «Единицы измерения размера шрифта» главы 12.

При использовании единиц измерения em браузер Internet Explorer неправильно отображает размер шрифта.
См. раздел «Единицы измерения размера шрифта» главы 12.

Выбранный шрифт отображается не во всех браузерах.
Шрифт должен быть установлен на компьютере.
См. раздел «Единицы измерения размера шрифта» главы 12.
Если вы используете свойство `@font-face`, необходимо предоставить шрифт в нескольких форматах.
См. раздел «Форматы шрифтов» главы 12.

При отображении на ПК у шрифтов наблюдаются зазубренные края.
У разных шрифтов разная степень сглаживания. Попробуйте использовать другую гарнитуру шрифта или более жирное начертание.
См. раздел «Способы использования дополнительных гарнитур шрифтов» главы 12.

КАСКАДНЫЕ ТАБЛИЦЫ СТИЛЕЙ

Указанный стиль не отображается.
Помните, что селекторы CSS чувствительны к регистру.

Убедитесь в правильности написания селекторов.
См. раздел «Селекторы CSS» главы 10.

Найти нужный селектор можно с помощью панели инструментов веб-разработчика.
См. раздел «Панель инструментов веб-разработчика» главы 14.

Если селекторы указаны корректно, проверьте CSS-код на предмет наличия других селекторов, имеющих целью тот же элемент.
См. раздел «Принципы каскадирования правил CSS» главы 10.

В некоторых браузерах каскадные таблицы стилей могут работать иначе.
Существует ряд CSS-ошибок браузера, из-за которых страница отображается не так, как ожидалось. Попробуйте найти проблему и проверьте, не является ли она распространенной CSS-ошибкой браузера.
См. раздел «Проблемы совместимости CSS с браузерами» главы 10.

ЯЗЫК HTML5

Браузер не применяет стили к элементам HTML5.
Если речь идет о программе IE, то вам может понадобиться использовать код JavaScript, известный как HTML5 shiv или HTML5 shim.
См. раздел «Обеспечение поддержки HTML5 в старых версиях браузеров» главы 17.

Блочные элементы отображаются как встроенные.
Используйте фрагмент кода `display:block`, чтобы указать браузеру, какие элементы HTML5 являются блочными.
См. раздел «Обеспечение поддержки HTML5 в старых версиях браузеров» главы 17.

МАКЕТ

На некоторых экранах дизайн кажется больше/меньше.
На размер элементов страницы влияет разрешение монитора.
См. раздел «Размеры экранов устройств» главы 15.

Над и под блоком не отображаются поля.
Схлопывание вертикальных полей.
См. раздел «Воздух и вертикальные поля» главы 13.

Контент не помещается в элемент-контейнер / окно браузера.
Вы можете решить данную проблему, используя свойство перетекания контента.
См. раздел «Перетекание содержимого» главы 13.
См. раздел «Блочная модель Internet Explorer 6» главы 13.

В браузере Internet Explorer 6 отображаются блоки другого размера.
В некоторых версиях программы Internet Explorer используется иная блочная модель. Используйте тег `DOCTYPE`, чтобы ранние версии Internet Explorer работали так же, как и другие браузеры.
См. раздел «Блочная модель Internet Explorer 6» главы 13.

Блоки не выравниваются по центру при использовании значения `auto` для левого и правого полей.
Возможно, вам придется применить свойство `text-align` к элементу-контейнеру.
См. раздел «Центрирование контента» главы 13.

Элементы перекрывают друг друга.
При исключении элементов из нормального потока они иногда начинают перекрывать друг друга. Используя свойство `z-index`, вы можете указать, какой элемент должен находиться на переднем плане.
См. раздел «Перекрывающиеся элементы» главы 15.

Почему использование свойства `vertical-align` не приводит к вертикальному центрированию блочного элемента?
Данное свойство не предназначено для этой цели. Оно используется для центрирования встроенных элементов страницы. С помощью поисковой системы Google вы можете найти несколько способов вертикального центрирования блочного элемента.

Фоновое изображение блока не отображается.
Указана ли высота и ширина для блока, к которому вы применяете стиль?
См. раздел «Размеры блока» главы 13.
Указано ли для свойства `overflow` блока-контейнера значение `auto`?
См. раздел «Родительские элементы плавающих блоков» главы 15.

При печати страницы отсутствуют фоновые изображения.
Большинство браузеров по умолчанию не печатают фоновые изображения в целях экономии чернил. Вы можете изменить это в настройках печати.

Между окном браузера и контентом страницы отображается промежуток.
Возможно, вам придется применить свойства `margin` и `padding` к элементу `<body>`.
См. разделы «Отступы» и «Поля» главы 13.

ПЛАВАЮЩИЕ ЭЛЕМЕНТЫ МАКЕТА

Блок не располагается рядом с другим плавающим элементом.
Убедитесь в том, что элемент-контейнер предусматривает достаточно места, чтобы два плавающих блока могли разместиться друг рядом с другом.
См. раздел «Размеры блока» главы 13.

Поля и отступы добавляются к значению ширины блока (за исключением старых версий программы Internet Explorer, которая использует иную блочную модель).
В связи с этим ширина блока может быть больше, чем значение, указанное для соответствующего свойства.
См. раздел «Блочная модель Internet Explorer 6» главы 13.
Указали ли вы значение ширины плавающего элемента? (См. следующий пункт.)

Плавающий элемент занимает всю ширину окна

браузера (или элемента-контейнера).
Убедитесь, что вы указали значение ширины плавающего элемента.
См. раздел «Использование свойства float для размещения элементов друг рядом с другом» главы 15.

Высота элемента-контейнера для плавающих блоков равна одному пикселу.

Элементу-контейнеру не известна высота расположенных в нем плавающих блоков. Вы можете добавить элемент со свойством `clear` или применить свойство `overflow` со значением `auto`.
См. раздел «Родительские элементы плавающих блоков» главы 15

Браузер Internet Explorer добавляет дополнительные поля для плавающих элементов.
Задайте значение `inline` для свойства `display`.

ЭЛЕМЕНТЫ HTML

<a>, 76
<abbr>, 56
<acronym>, 56
<address>, 58
<article>, 405,408
<aside>, 409
<audio>, 208
<b>, 48
<blockquote>, 55
<body>, 27,32

, 51
<button>, 155
<cite>, 57
<dd>, 68
<del>, 59
<dfn>, 57
, 352
<dl>, 68
<dt>, 68
, 54
<embed>, 192
<fi eldset>, 157
<fi gcaption>, 117
<fi gure>, 117
<footer>, 406
<form>, 144
<h1>, 46
<h2>, 46
<h3>, 46
<h4>, 46
<h5>, 46
<h6>, 46
<head>, 32
<header>, 406
<hgroup>, 411
<hr />, 51
<html>, 27
<i>, 48
<iframe>, 180
<img>, 96
<ins>, 59
<input>, 145
<label>, 156
<legend>, 157
, 66
<link>, 222
<meta>, 182
<nav>, 407
<object>, 192
, 66
<option>, 150
<p>, 47
<q>, 55
<s>, 59
<script>, 197
<select>, 150
<section>, 410
<source>, 204,209
, 179
, 54
<style>, 223
<sub>, 49
<sup>, 49
<table>, 126
<tbody>, 130
<td>, 126
<textarea>, 147
<tfoot>, 130
<th>, 127
<thead>, 130
<title>, 96
<tr>, 126
<u>, 59
, 67
<video>, 202

АТРИБУТЫ HTML

action, 144
align, 100
alt, 96
auto, 203
autoplay, 203
bgcolor, 133
border, 133
checked, 148
cite, 55
class, 175
codecs, 204
cols, 147
colspan, 128
content, 182
controls, 203
description, 182
for, 156
frameborder, 181
height, 97
href, 76,78, 222
http-equiv, 183
id, 86,144,174
keywords, 182
loop, 203
maxlength, 145
metadata, 203
method, 144
multiple, 151
name, 145-182
none, 203
placeholder, 161
poster, 203
preload, 203
rel, 222
robots, 182
rows, 64
rowspan, 129
scrolling, 181
seamless, 181
selected, 66
size, 145
src, 96
style, 110
target, 85
title, 96
type, 222
type="date", 159
type="email", 160
type="file", 152
type="hidden", 155
type="image", 154
type="password", 146
type="radio", 148
type="search", 161
type="submit", 153
type="text", 145
type="url", 160
value, 148
width, 97
Глобальные атрибуты, 174
background, 391
background-attachment, 389
background-color, 388
background-image (градиенты), 388
background-image, 388
background-position, 389
background-repeat, 389
border, 293
border-bottom-color, 293
border-bottom-style, 292
border-collapse, 319
border-color, 292
border-image, 300
border-left-color, 293
border-left-style, 292
border-radius, 302
border-right-color, 292
border-right-style, 292
border-spacing, 319
border-style, 291
border-top-color, 292
border-top-style, 292
border-width, 290
bottom, 284
box-shadow, 301
clear, 349
color, 234
cursor, 326
display, 298
empty-cells, 318
fl oat, 325, 347
@font-face, 260
font-family, 256
font-size, 257
font-style, 263
height, 284
hsl, 240
hsla, 241
left, 268
letter-spacing, 264, 267
line-height, 266
list-style, 315
list-style-image, 313
list-style-position, 314
list-style-type, 314
margin, 289,295
margin-bottom, 295
margin-left, 295
margin-right, 295
margin-top, 295
opacity, 239
padding, 289, 293, 294
padding-bottom, 294
padding-left, 294
padding-right, 294
padding-top, 294
position, 340
rgba, 239
right, 268
text-align, 268
text-decoration, 265
text-indent, 270
text-shadow, 271
text-transform, 264
top, 284
vertical-align, 269
visibility, 299
width, 284
word-spacing, 267
z-index, 341, 346

СВОЙСТВА CSS

background, 215
background-attachment, 214
background-color, 117
background-image (градиенты), 217
background-image, 212
background-position, 214
background-repeat, 214
border, 153
border, 153, 172
border-bottom-color, 153
border-bottom-style, 152
border-collapse, 170
border-color, 152
border-image, 158
border-left-color, 153
border-left-style, 152
border-radius, 160
border-right-color, 152
border-right-style, 152
border-spacing, 170
border-style, 152
border-top-color, 152
border-top-style, 152
border-width, 151
bottom, 185, 189
box-shadow, 159
clear, 190
color, 116
cursor, 176
display, 156
empty-cells, 169
float, 175, 189, 211
@font-face, 127
font-face, 127
font-family, 127
font-size, 130
font-style, 136
height, 210
hsl, 121
hsla, 122
left, 185
letter-spacing, 137
line-height, 138
list-style, 167
list-style-image, 165
list-style-position, 166
list-style-type, 164
margin, 150, 154
margin-bottom, 155
margin-left, 155
margin-right, 155
margin-top, 154
opacity, 120
padding, 150, 153, 167
padding-bottom, 154
padding-left, 154
padding-right, 154
padding-top, 154
position, 185
rgba, 120
right, 185
text-align, 139
text-decoration, 137
text-indent, 140
text-shadow, 141
text-transform, 136
top, 185
vertical-align, 140
visibility, 157
width, 147, 167, 189, 210
word-spacing, 138
z-index, 184, 187

ПСЕВДОКЛАССЫ, ЭЛЕМЕНТЫ И ПРАВИЛА

:active, 274
:first-letter, 272
:first-line, 272
:focus, 274
@font-face, 260
:hover, 274
@import, 372
:link, 273
:visited, 273

Научно-популярное издание

МИРОВОЙ КОМПЬЮТЕРНЫЙ БЕСТСЕЛЛЕР

Джон Дакетт

HTML И CSS
РАЗРАБОТКА И ДИЗАЙН ВЕБ-САЙТОВ

Директор редакции *Е. Капьёв*
Ответственный редактор *В. Обручев*
Художественный редактор *Е. Мишина*

ООО «Издательство «Эксмо»
127299, Москва, ул. Клары Цеткин, д. 18/5. Тел. 411-68-86, 956-39-21.
Home page: **www.eksmo.ru** E-mail: **info@eksmo.ru**

Өндіруші: Издательство «ЭКСМО»ЖШҚ, 127299, Мәскеу, Ресей, Клара Цеткин көш., үй 18/5.
Тел. 8 (495) 411-68-86, 8 (495) 956-39-21
Home page: www.eksmo.ru E-mail: info@eksmo.ru.
Тауар белгісі: «Эксмо»
Қазақстан Республикасында дистрибьютор және өнім бойынша арыз-талаптарды
қабылдаушының
өкілі «РДЦ-Алматы» ЖШС, Алматы қ., Домбровский көш., 3«а», литер Б, офис 1.
Тел.: 8(727) 2 51 59 89,90,91,92, факс: 8 (727) 251 58 12 вн. 107; E-mail: RDC-Almaty@eksmo.kz
Өнімнің жарамдылық мерзімі шектелмеген.
Сертификация туралы ақпарат сайтта: www.eksmo.ru/certification

Өндірген мемлекет: Ресей
Сертификация қарастырылмаған

Сведения о подтверждении соответствия издания согласно законодательству РФ
о техническом регулировании можно получить по адресу: http://eksmo.ru/certification/

Подписано в печать 04.07.2013. Формат 70x100$^{1}/_{16}$.
Печать офсетная. Усл. печ. л. 38,89.
Тираж экз. Заказ

ISBN 978-5-699-64193-2

Оптовая торговля книгами «Эксмо»:
ООО «ТД «Эксмо». 142700, Московская обл., Ленинский р-н, г. Видное,
Белокаменное ш., д. 1, многоканальный тел. 411-50-74.
E-mail: **reception@eksmo-sale.ru**

По вопросам приобретения книг «Эксмо» зарубежными оптовыми покупателями *обращаться в отдел зарубежных продаж ТД «Эксмо»*
E-mail: **international@eksmo-sale.ru**

International Sales: International wholesale customers should contact Foreign Sales Department of Trading House «Eksmo» for their orders.
international@eksmo-sale.ru

По вопросам заказа книг корпоративным клиентам, в том числе в специальном оформлении, *обращаться по тел. +7 (495) 411-68-59, доб. 2261, 1257.*
E-mail: **vipzakaz@eksmo.ru**

Оптовая торговля бумажно-беловыми и канцелярскими товарами для школы и офиса «Канц-Эксмо»:
Компания «Канц-Эксмо»: 142702, Московская обл., Ленинский р-н, г. Видное-2,
Белокаменное ш., д. 1, а/я 5. Тел./факс +7 (495) 745-28-87 (многоканальный).
e-mail: **kanc@eksmo-sale.ru**, сайт: www.**kanc-eksmo.ru**

Полный ассортимент книг издательства «Эксмо» для оптовых покупателей:
В Санкт-Петербурге: ООО СЗКО, пр-т Обуховской Обороны, д. 84Е.
Тел. (812) 365-46-03/04.
В Нижнем Новгороде: ООО ТД «Эксмо НН», 603094, г. Нижний Новгород,
ул. Карпинского, д. 29, бизнес-парк «Грин Плаза». Тел. (831) 216-15-91 (92, 93, 94).
В Ростове-на-Дону: ООО «РДЦ-Ростов», пр. Стачки, 243А. Тел. (863) 220-19-34.
В Самаре: ООО «РДЦ-Самара», пр-т Кирова, д. 75/1, литера «Е». Тел. (846) 269-66-70.
В Екатеринбурге: ООО «РДЦ-Екатеринбург», ул. Прибалтийская, д. 24а.
Тел. +7 (343) 272-72-01/02/03/04/05/06/07/08.
В Новосибирске: ООО «РДЦ-Новосибирск», Комбинатский пер., д. 3.
Тел. +7 (383) 289-91-42. E-mail: **eksmo-nsk@yandex.ru**
В Киеве: ООО «РДЦ Эксмо-Украина», Московский пр-т, д. 9. Тел./факс: (044) 495-79-80/81.
В Донецке: ул. Артема, д. 160. Тел. +38 (032) 381-81-05.
В Харькове: ул. Гвардейцев Железнодорожников, д. 8. Тел. +38 (057) 724-11-56.
Во Львове: ТП ООО «Эксмо-Запад», ул. Бузкова, д. 2. Тел./факс (032) 245-00-19.
В Симферополе: ООО «Эксмо-Крым», ул. Киевская, д. 153.
Тел./факс (0652) 22-90-03, 54-32-99.
В Казахстане: ТОО «РДЦ-Алматы», ул. Домбровского, д. 3а.
Тел./факс (727) 251-59-90/91. **rdc-almaty@mail.ru**

Полный ассортимент продукции издательства «Эксмо»
можно приобрести в магазинах «Новый книжный» и «Читай-город».
Телефон единой справочной: 8 (800) 444-8-444. Звонок по России бесплатный.

Интернет-магазин ООО «Издательство «Эксмо»
www.fiction.eksmo.ru
Розничная продажа книг с доставкой по всему миру.
Тел.: +7 (495) 745-89-14. E-mail: **imarket@eksmo-sale.ru**

КОГДА ВЫ ДАРИТЕ КНИГУ, ВЫ ДАРИТЕ ЦЕЛЫЙ МИР

ХОТИТЕ ЗНАТЬ БОЛЬШЕ?

Заходите на сайт:
https://eksmo.ru/b2b/

Звоните по телефону:
+7 495 411-68-59, доб. 2261

ВАШ ЛОГОТИП НА ОБЛОЖКЕ

ВАШ ЛОГОТИП НА КОРЕШКЕ

ОБРАЩЕНИЕ К КЛИЕНТАМ НА ОБЛОЖКЕ

Эта книга предназначена для

веб-дизайнеров и программистов, контент-менеджеров и редакторов контента, маркетологов и менеджеров по электронным продажам, блогеров и любителей.

Вы научитесь

верстать страницы на языке HTML и CSS3, узнаете о структуре веб-страниц и сайтов, подготовке изображений, аудио и видеоконтента, управлению типографикой и дизайном макета.

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. Хотя код веб-страниц может немного вас пугать, загляните в книгу, и вы увидите, насколько это руководство отличается от большинства остальных книг по веб-дизайну.

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. Никакие специальные навыки для освоения книги не требуются.

Джон Дакетт проектирует и разрабатывает веб–сайты уже более десяти лет. Он работает как с небольшими стартапами, так и глобальными брендами. Под его авторством выпущено несколько книг, посвященных веб–дизайну, программированию, юзабилити и доступности веб–сайтов.

ISBN 978-5-699-64193-2

Это самая приятная книга по веб-дизайну из всех, которые я читал. Она логично организована, для объяснений в ней используются простые термины, примеры кода и великолепные иллюстрации. Книга сфокусирована на основах HTML и CSS, но в ней уделено внимание и процессу веб-дизайна, поисковой оптимизации и веб-аналитике.

Чад Уорнер, веб-дизайнер, http://optimwise.com