

Эрик Фримен Элизабет Фримен

2-е
издание

Изучаем HTML XHTML и CSS

Начните
свою веб-карьеру
с первых строк
книги



Руководство
по созданию
сайтов
на основе
веб-стандартов



Узнайте
все хитрости работы
с HTML и CSS



Узнайте,
каким на самом деле
должен быть веб-дизайн

Решите около
100 упражнений
и головоломок



Научитесь
избегать ошибок
при проверке кода



Head First HTML and CSS

Wouldn't it be dreamy if there were an HTML book that didn't assume you knew what elements, attributes, validation, selectors, and pseudo-classes were, all by page three? It's probably just a fantasy...



Elisabeth Robson
Eric Freeman

O'REILLY®

Beijing • Cambridge • Farnham • Köln • Sebastopol • Tokyo

Изучаем HTML, XHTML и CSS

Как было бы замечательно
иметь понятную для всех книгу
по HTML, в которой допускается,
что вы не знаете о существовании
атрибутов, селекторов, валидации
и псевдоклассов. Наверное,
это так и останется
мечтой...



Элизабет Робсон
Эрик Фримен

2-е издание

 **ПИТЕР®**

Москва • Санкт-Петербург • Нижний Новгород • Воронеж
Ростов-на-Дону • Екатеринбург • Самара • Новосибирск
Киев • Харьков • Минск
2014

Робсон Э., Фримен Э.
Изучаем HTML, XHTML и CSS
2-е издание

Серия «Head First O'Reilly»

Перевел с английского *В. Черник*

Заведующий редакцией	<i>Д. Виницкий</i>
Научный редактор	<i>В. Черник</i>
Художник	<i>Л. Адуевская</i>
Корректор	<i>Н. Викторова</i>
Верстка	<i>Е. Лея</i>

ББК 32.988.02-018
УДК 004.438.5

Робсон Э., Фримен Э.

Ф88 Изучаем HTML, XHTML и CSS. 2-е изд. — СПб.: Питер, 2014. — 720 с.: ил. — (Серия «Head First O'Reilly»).
ISBN 978-5-496-00653-8

Устали от чтения книг по HTML, которые понятны только специалистам в этой области? Тогда самое время взять в руки второе издание «Изучаем HTML, XHTML и CSS». Хотите изучить HTML, чтобы уметь создавать веб-страницы, о которых вы всегда мечтали? Так, чтобы более эффективно общаться с друзьями, семьей и привередливыми клиентами? Тогда эта книга для вас. Прочитав ее, вы изучите все секреты создания веб-страниц. Вы узнаете, как работают профессионалы, чтобы получить визуально привлекательный дизайн, и как максимально эффективно использовать HTML, CSS и XHTML, чтобы создавать такие веб-страницы, мимо которых не пройдет ни один пользователь. Используя новейший стандарт HTML5, вы сможете поддерживать и совершенствовать свои веб-страницы в соответствии с современными требованиями, тем самым обеспечивая их работу во всех браузерах и мобильных устройствах.

12+ (В соответствии с Федеральным законом от 29 декабря 2010 г. № 436-ФЗ.)

ISBN 978-0596159900 англ.

Copyright © 2012 Elisabeth Robson and Eric Freeman. All rights reserved.
Authorized Russian translation of the English edition of Head First HTML and CSS,
2nd Edition (ISBN 9780596159900). This translation is published and sold by
permission of O'Reilly Media, Inc., which owns or controls all rights to publish and
sell the same.

ISBN 978-5-496-00653-8

© Перевод на русский язык ООО Издательство «Питер», 2014
© Издание на русском языке, оформление ООО Издательство «Питер», 2014

Права на издание получены по соглашению с O'Reilly. Все права защищены. Никакая часть данной книги не может быть воспроизведена в какой бы то ни было форме без письменного разрешения владельцев авторских прав.

Информация, содержащаяся в данной книге, получена из источников, рассматриваемых издательством как надежные. Тем не менее, имея в виду возможные человеческие или технические ошибки, издательство не может гарантировать абсолютную точность и полноту приводимых сведений и не несет ответственности за возможные ошибки, связанные с использованием книги.

ООО «Питер Пресс», 192102, Санкт-Петербург, ул. Андреевская (д. Волкова), 3, литер А, пом. 7Н.

Налоговая льгота — общероссийский классификатор продукции ОК 005-93, том 2; 95 3005 — литература учебная.

Подписано в печать 19.07.13. Формат 84×108/16. Усл. п. л. 75,600. Тираж 2500. Заказ 0000.

Отпечатано в полном соответствии с качеством предоставленных издательством материалов
в ГППО «Псковская областная типография». 180004, Псков, ул. Ротная, 34.

Содержание (сводка)

	Введение	23
1	Язык Сети. Знакомство с HTML	35
2	Знакомство с гипертекстом. <i>Идем дальше – используем гипертекст</i>	75
3	Конструирование веб-страниц. <i>Строительные блоки</i>	107
4	Путешествие в Webville. <i>Соединение</i>	149
5	Знакомство с медиа. <i>Добавление изображений на страницы</i>	187
6	Серьезный HTML. <i>Стандарты</i>	241
7	Начнем работать над дизайном. <i>Приступаем к работе с CSS</i>	275
8	Увеличиваем словарный запас. <i>Меняем шрифты и цвета</i>	329
9	Познакомимся с элементами поближе. <i>Блочная модель</i>	377
10	Современная веб-конструкция. <i>Элементы div и span</i>	427
11	Расставим элементы по местам. <i>Разметка и позиционирование</i>	485
12	Современный HTML. <i>HTML5-разметка</i>	557
13	Представление в табличной форме. <i>Таблицы и большие списки</i>	611
14	Переход на интерактивный режим. <i>HTML-формы</i>	653
	Приложение. Топ-10 тем, которые не были освещены в этой книге	705

Содержание (настоящее)

Введение

Поведение вашего мозга при изучении HTML и CSS. Когда вы пытаетесь что-либо выучить, ваш мозг неустанно следит за тем, чтобы процесс изучения не остановился. Он думает: «Лучше оставить место для более важной информации, чтобы, например, знать, встречи с какими дикими животными следует избегать. Или знать, что катание на сноуборде без специального снаряжения — не самая удачная идея». Как же убедить свой мозг в том, что для вас так же важно знать HTML и CSS?

Для кого написана эта книга?	24
Метапознание: учимся учиться	27
Вот что МЫ делали	28
А вот что можете сделать ВЫ, чтобы заставить свой мозг работать	29
Примите к сведению	30
Технические рецензенты (первое издание)	32
Благодарности (первое издание)	33
Технические рецензенты (второе издание)	34
Благодарности (второе издание)	34

1

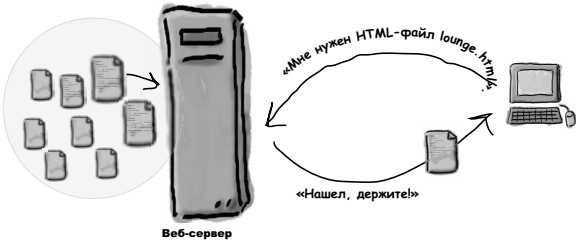
знакомство с HTML

Язык Сети

Единственное, что необходимо для того, чтобы успешно работать в Сети, — научиться говорить на ее специфическом языке: HyperText Markup Language (язык гипертекстовой разметки), или сокращенно HTML. Итак, приготовьтесь к нескольким урокам языка. После этой главы вы не только узнаете некоторые базовые понятия HTML, но и сможете разговаривать на этом языке, используя определенный **стиль**. Черт возьми, к концу этой книги вы сможете говорить на языке HTML так, будто выросли в Сети!



Видео убило радиозвезду	36
Что делает веб-сервер	37
Что делает браузер	37
Что пишете вы (HTML-код)	38
Что создает браузер	39
Большая перемена в кафе Starbuzz	43
Создание веб-страницы для Starbuzz	45
Создание HTML-файла (Mac)	46
Создание HTML-файла (Windows)	48
Между тем вернемся к кафе Starbuzz	51
Сохранение работы	52
Открытие веб-страницы в браузере	53
Тестирование страницы	54
Еще один тест	58
Разделение тегов	59
Познакомьтесь с элементом <style>	63
Придание определенного стиля странице Starbuzz	64



идем дальше — используем Гипертекст

2 Знакомство с гипертекстом

Кто-то сказал «гипертекст»? Что это? О, только чистая основа Сети. В главе 1 мы привели основные сведения о языке HTML, и, надеемся, вы пришли к выводу, что это хороший язык для разметки текста, используемый для описания структуры веб-страниц. Сейчас наша задача — разобраться с гипертекстом, который позволит освободиться от одиночных страниц и ссылаться на другие страницы. В процессе этого мы познакомимся с новым элементом `<a>` и поймем, какая превосходная штука — взаимосвязь страниц. Итак, пристегните ремни безопасности, вы вот-вот начнете изучение гипертекста.



Новая и усовершенствованная гостевая	76
Создание новой гостевой	78
Что делали мы	80
Что делает браузер	81
Что такое атрибуты	83
Технические трудности	90
Планирование путей	92
Восстановление «отсутствующих изображений»	98

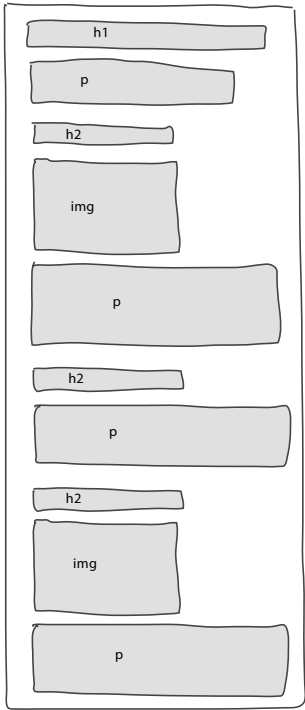


3

строительные блоки

Конструирование веб-страниц

Мы говорили вам, что в этой книге вы действительно будете создавать веб-страницы. Конечно, уже многое выучено: теги, элементы, ссылки, пути, однако все это бесполезно, если, используя полученные знания, не попробовать создать парочку потрясающих веб-страниц. В этой главе мы будем расширять строительство веб-страниц: перейдем от их общего осмысления к проектированию, зальем фундамент, построим их и даже выполним кое-какую отделку. Все, что вам нужно, — это усердие, трудолюбие и пояс для рабочих инструментов, так как мы будем добавлять некоторые новые инструменты и давать вам информацию, как пользоваться ими.



От дневника к сайту на скорости 12 миль в час	109
Черновик	110
От черновика к плану	111
От плана к веб-странице	112
Тестирование страницы Тони	114
Добавление новых элементов	115
Знакомство с элементом <q>	116
...и его тестирование	116
Дли-и-и-инные цитаты	120
Добавление элемента <blockquote>	121
Полное разоблачение тайны <q> и <blockquote>	124
Тем временем вернемся к сайту Тони...	130
Разработка HTML-списков в два этапа	132
Тестирование списков на примере перечня городов	134
Используйте вложенность, чтобы убедиться в соответствии тегов	139

4

соединение

Путешествие в Webville

Веб-страницы предназначены для того, чтобы располагаться и обслуживаться в Интернете. До сих пор вы создавали веб-страницы, которые «жили» только в вашем собственном компьютере. Вы также создавали ссылки только на те страницы, которые хранятся на вашем компьютере. Мы вот-вот изменим это навсегда. В этой главе мы научим вас размещать веб-страницы в Интернете, где все ваши родные, друзья и покупатели действительно смогут их увидеть. Мы также раскроем тайну создания ссылок на другие страницы, взломав код `h, t, t, p, :, /, /, w, w, w`. Итак, собирайте свои вещи, следующая остановка — Webville.

Размещение сайта Starbuzz (или вашего собственного сайта) в Сети	150
Поиск хостинговой компании	151
Привет, мое имя...	152
Как можно получить доменное имя	152
Заселение	154
Перемещение файлов в корневую папку	155
Столько информации об FTP, сколько может поместиться на две страницы	156
Вернемся к делу...	159
Что такое HTTP-протокол	161
Что такое абсолютный путь	162
Как работают страницы, выдаваемые по умолчанию	165
Как мы создаем ссылки на другие сайты	168
Создание ссылки на страницу о кофеине	169
А теперь протестируем...	170
Наивысший уровень качества веб-страниц	173
Тестирование атрибута title	174
Создание ссылки внутрь страницы	175
Использование атрибута id для указания пункта назначения для элемента <code><a></code>	176
Как сослаться на элемент с использованием идентификатора	177
Переход по ссылке в новое окно	181
Открытие нового окна с использованием атрибута target	182



5

Добавление изображений на страницы

Знакомство с медиа

Улыбнитесь и скажите «сыр». Теперь улыбнитесь и скажите «gif», «jpg» или «png» — это те форматы файлов, которые вы выберете, создавая рисунки для Сети. В этой главе вы узнаете все о том, как добавить на веб-страницу свой первый медиафайл — изображение. У вас есть парочка цифровых фотографий, которые вы хотите поместить в Сеть? Никаких проблем. У вас есть логотип, который нужен на веб-странице? И это легко. Или, может быть, сначала вы хотите более близко познакомиться с элементом ``? К концу этой главы вы будете знать все мельчайшие подробности того, как использовать этот элемент и его атрибуты. Вы также узнаете, как этот небольшой элемент побуждает браузер делать такую серьезную работу по поиску и отображению ваших картинок.

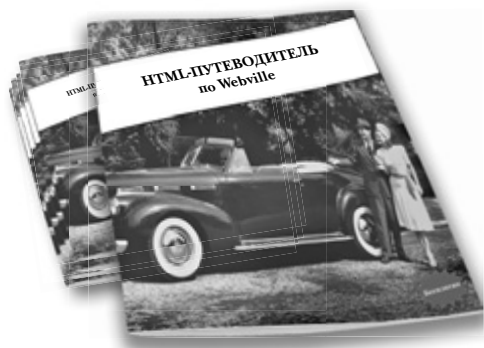


Как браузер работает с изображениями	188
Как работают рисунки	191
<code></code> : теперь не только относительные ссылки	195
Всегда имейте запасной вариант	197
Определение размеров изображений	198
Создание сайта для самых больших фанатов: myPod	199
Доработка файла index.html для сайта myPod	200
Ого! Рисунок слишком большой	202
Изменение размера изображения	204
Открытие изображения	206
Размеры изменены, теперь сохраняем	210
Исправление HTML-кода для myPod	212
А сейчас протестируем...	212
Еще больше фотографий для myPod	214
Еще один тест для myPod	215
Доработка сайта таким образом, чтобы использовались эскизы	216
Создание эскизов	217
И снова тест для myPod	219
Превращение эскизов в ссылки	220
Создание индивидуальных страниц для фотографий	221
Итак, как же создать изображения-ссылки?	222
Открытие логотипа myPod	226
Какой формат использовать?	227
Использовать прозрачность или нет?	228
Сохранение прозрачного PNG-изображения	229
Минуточку, а как узнать цвет фона веб-страницы?	230
Установка цвета подложки	230
Рассмотрим логотип с подложкой	231
Сохранение логотипа	232
Добавление логотипа на веб-страницу myPod	232

6 стандарты

Серьезный HTML

Что же еще нужно знать об HTML? Вы уже неплохо справляетесь с написанием HTML-страниц. Не настало ли время перейти к CSS и научиться придавать всей этой разметке еще и ошеломительный внешний вид? Перед тем как сделать это, мы должны убедиться, что ваши знания о HTML на должном уровне. Не поймите нас неправильно, вы и так создавали первоклассный HTML, но есть еще несколько вещей, которые вам нужно сделать, чтобы превратить его в «индустриально-стандартный» HTML. Пришло время, когда следует задуматься об обязательном использовании новейшего и самого лучшего HTML-стандарта, также известного как HTML5. Благодаря этому вы сможете гарантировать, что ваши страницы будут одинаково отображаться во всех браузерах (по крайней мере, в таких, которые для вас важны), не говоря уже о том, что они смогут нормально работать на устройствах от компании Apple, относящихся к самому последнему поколению (выберите свое любимое). Вы также сможете создавать страницы, которые быстрее загружаются, гарантированно хорошо взаимодействуют с CSS, идут в ногу с развитием стандартов. Приготовьтесь, в этой главе вы из любителя превратитесь в профессионала!



История развития HTML	244
Новое и усовершенствованное определение типа документа HTML5	249
HTML – новый «живой стандарт»	250
Добавление определения типа документа	251
Тест для DOCTYPE	252
Познакомьтесь с W3C-валидатором	255
Валидация гостевой Head First	256
Хьюстон, у нас проблема...	257
Исправление этой ошибки	258
Добавление тега <meta> для определения кодировки символов	261
Пообщавшись со всеми HTML-профессионалами, возьмите путеводитель	266

1 приступаем к работе с CSS

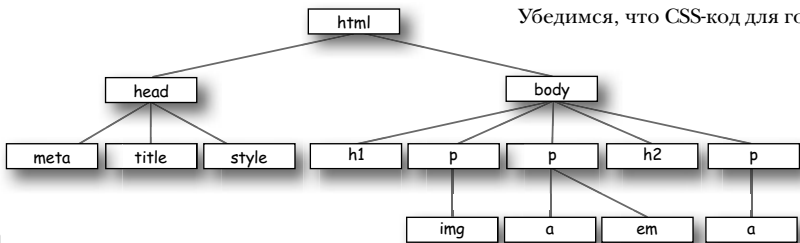
Начнем работать над дизайном

Раньше говорилось, что в книге будет материал про CSS. До сих пор мы изучали HTML, применяемый для создания структуры веб-страниц. Но, как видите, манера браузеров оформлять страницы оставляет желать лучшего. Конечно же, можно было позвать полицию моды, но нам это не нужно. Мы отлично справимся с дизайном страниц с помощью CSS, часто даже не меняя HTML-код. Действительно ли это так легко? Ну, придется выучить новый язык; в конце концов, Webville — это двуязычный город. После прочтения этой главы, посвященной CSS, вы будете в состоянии поддерживать разговор, находясь на любой из сторон Мейнстрит.

Пятиминутная Головоломка



Вы больше не в Канзасе	276
Послушаем, что происходит в реалити-шоу «Квартира соседа» в Webville	278
Использование CSS вместе с HTML	279
Добавление CSS в ваш HTML	281
Добавление стиля в гостевую	282
Тестирование стиля	283
Оформление заголовков	284
Подчеркнем заголовки с приветствием	285
Существует особая технология: указание второго правила только для <h1>	286
Как же на самом деле работают селекторы	287
Визуальное представление селекторов	290
Присвоение стиля основной страницы гостевой страницам с напитками и указателями	293
Создание файла lounge.css	294
Создание ссылки из lounge.html на внешний CSS-файл	295
Создание ссылок на внешние таблицы стилей из файлов elixir.html и directions.html	296
Тестирование всего сайта	297
Пришло время поговорить о наследовании	301
Что будет, если мы переместим font вверх по дереву?	302
Протестируем новый CSS-код	303
Переопределение наследуемых свойств	304
Добавление элемента в класс greentea	307
Создание селектора класса	308
Тестирование класса greentea	309
Поработаем с классами еще	310
Самое краткое в мире руководство по применению классов	312
Убедимся, что CSS-код для гостевой валидный	319

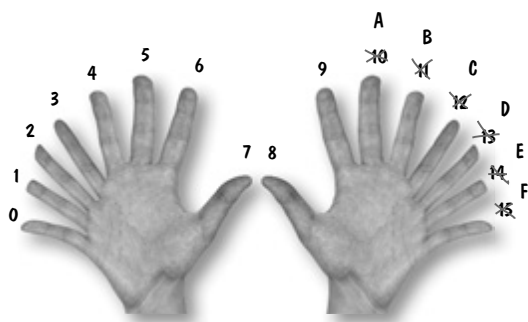


Оформление страницы с помощью шрифтов и цветов

Увеличиваем словарный запас

Ваше изучение языка CSS проходит успешно. Вы уже ознакомились с основами CSS и знаете, как создавать правила, выбирать элементы и определять для них стили. Теперь настало время увеличить ваш словарный запас, а это означает, что вам нужно познакомиться с некоторыми новыми свойствами и узнать, что они могут делать. В настоящей главе мы поработаем с несколькими наиболее используемыми свойствами, которые влияют на оформление текста. Для этого вам придется кое-что узнать о цветах и шрифтах. Вы поймете, что совершенно не обязательно устанавливать те шрифты, которые применяются повсеместно, или те размеры и стили, что по умолчанию используются браузерами для абзацев и заголовков. Вы также узнаете, что существует намного больше цветов, чем может различить ваш глаз.

Самое главное о тексте и шрифтах	330
Итак, что такое семейство шрифтов?	332
Определение семейств шрифтов в CSS	335
Как работает свойство font-family	335
Вновь поработаем с дневником Тони	336
Задаем новое свойство font-family	337
Тестирование новых шрифтов страницы Тони	338
Как быть, если у разных пользователей установлены различные шрифты?	339
Как работают веб-шрифты	341
Как добавить веб-шрифт в страницу...	343
Шаг шестой: загрузить страницу!	344
Тестирование шрифта дневника Тони	345
Размеры шрифта	346
Поменяем размеры шрифтов для веб-страницы Тони	350
Тестирование страницы с новыми размерами шрифтов	351
Настройка насыщенности шрифтов	353
Заголовки с плотностью normal. Тестирование страницы	354
Оформление шрифтов	355
Оформление цитаты курсивом на странице Тони	356
Как работают «безопасные» цвета?	358
Как задаются «безопасные» цвета? Рассмотрим разные способы...	361
Двухминутное руководство по использованию шестнадцатеричных кодов	364
Объединим все вместе	366
Где найти «безопасные» цвета	366
Вернемся к странице Тони... Сделаем заголовки оранжевыми и подчеркнем их	369
Тестирование оранжевых заголовков на странице Тони	370
Все, что вы хотели знать о декорировании текста	371
Удаление подчеркивания	372



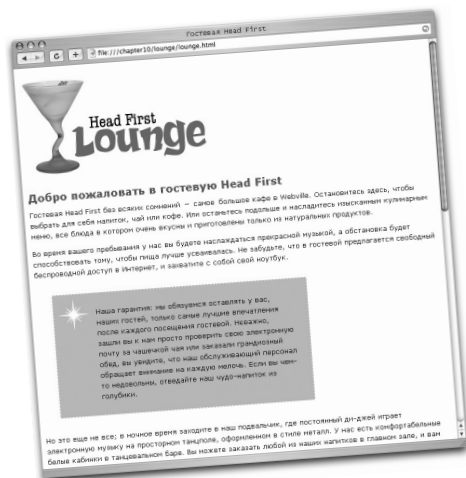
Блочная Модель

9

Познакомимся с элементами поближе

Чтобы создавать современные веб-сооружения, вам нужно на самом деле хорошо разбираться в строительных материалах. В этой главе мы подробно рассмотрим наши строительные материалы — элементы HTML. Мы буквально под микроскопом изучим, из чего сделаны все эти блочные и строчные элементы. Вы узнаете, как можно управлять практически всеми возможностями оформления элементов в CSS. Однако на этом мы не остановимся — вы также узнаете, как можно присваивать элементам уникальные идентификаторы. И, если этого будет недостаточно, вы выучите, в каком случае и как использовать несколько таблиц стилей. Итак, переворачивайте страницу и познакомьтесь с элементами поближе.

Модернизация гостевой	378
Подготовка к работе с новой гостевой	380
Начнем с нескольких простых изменений	380
Поработаем с межстрочными интервалами	382
Подготовка к главной реконструкции	383
Рассмотрим блочную модель более подробно	384
Что можно делать с блоками	386
Вернемся к гостевой	389
Создание стиля оформления для «абзаца с гарантией»	391
Тест для границы абзаца	392
Отступ, граница и поля «абзаца с гарантией»	393
Добавление отступов	393
Тест для отступов	394
Теперь добавим поля	394
Тест для полей	395
Добавление фоновового рисунка	396
Тест для фоновового рисунка	398
Закрепление фоновового изображения	399
Еще один тест для фоновового изображения	400
Как увеличить отступ только с левой стороны?	400
Как увеличить размер поля только с правой стороны?	401
Двухминутное руководство по границам	402
Доведение границы до совершенства	405
Атрибут id	410
Как же селектор идентификатора применяется в CSS	411
Использование идентификатора для гостевой	412
Смешивание нескольких таблиц стилей	414
Использование нескольких таблиц стилей	415
Таблицы стилей — теперь не только для представления в окнах браузеров	416
Добавление медиазапросов прямо в CSS	417



10

элементы `div` и `span`

Современная веб-конструкция

Пришло время для подготовки массивной конструкции. В этой главе мы займемся такими HTML-элементами, как `<div>` и ``. Это вам уже не мелкие прутья, а большие стальные балки. С помощью `<div>` и `` вы построите серьезные опорные конструкции и, расставив их по местам, сможете стилизовать новыми, более действенными методами. Обратите внимание, что ваш пояс для CSS-инструментов практически заполнен, так что настало время показать вам несколько приемов для быстрого доступа к ним, что очень облегчит определение всех этих свойств. В эту главу мы также пригласили специальных гостей — псевдоклассы, с помощью которых вы сможете создавать очень интересные селекторы.



Рассмотрим HTML с описанием напитков	429
Вясним, как можно разбить страницу на логические разделы	431
Вернемся в гостевую	436
Тест для <code><div></code>	437
Добавление границы	438
Тест для границы	438
Что осталось добавить в стиль раздела с напитками	439
План действий	440
Поработаем над шириной раздела с напитками	440
Протестируем ширину раздела с напитками	441
Оформление раздела с напитками	445
Тест для новых стилей	446
Мы почти закончили...	449
Нам нужен способ выбрать потомков	451
Изменение цвета для заголовков раздела с напитками	453
Быстрый тест	453
Решение проблемы с межстрочными интервалами	454
Посмотрите, что получилось	455
Пришло время воспользоваться сокращениями	456
Добавление элементов <code></code> за три простых шага	462
Тестирование элементов <code></code>	463
Элемент <code><a></code> и его разносторонняя личность	466
Как можно по-разному оформить элементы с учетом их состояния?	467
Применение псевдоклассов на практике	469
Тест для ссылок	470
Не пора ли поговорить о каскадности?	471
Поиграем в игру «Каков мой приоритет?»	474

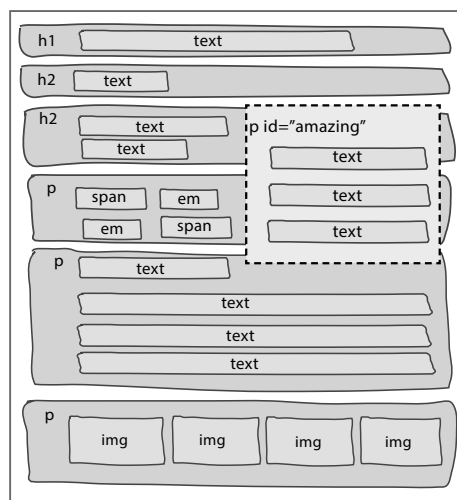
11

Разметка и позиционирование

Расставим элементы по местам



Пришло время обучить HTML-элементы новым трюкам. Пора разбудить их и заставить помочь нам создавать страницы с реальными схемами размещения элементов. Каким образом? Ну, вы хорошо разобрались со структурными элементами `<div>` и `` и теперь знаете, как работает блочная модель. Пришла пора применить эти знания и создать несколько дизайнов. Мы говорим не просто о цвете фона и шрифта, а о полностью профессиональном дизайне, в котором используется многоколоночная разметка.



Вы сделали упражнение «Мозговой штурм» повышенной сложности?	486
Используй поток, Люк	487
Как насчет строчных элементов?	489
Как создать плавающий элемент	493
Поэкспериментируем над плавающим элементом в гостевой	495
Новый сайт для Starbuzz	497
Посмотрим на разметку	498
А теперь посмотрим на стиль	500
Переведем Starbuzz на следующий уровень	501
Решение проблемы двух колонок	505
Задание поля для области с основным содержимым	506
Ох, у нас появилась еще одна проблема	507
Решение проблемы с наложением	509
Левее, выше, правее...	512
Дизайны с фиксированной и непостоянной шириной	515
Что находится между разметками с фиксированной и непостоянной шириной? Гибкая разметка, конечно же!	516
Тест для гибкой разметки	517
Как работает абсолютное позиционирование	518
Изменение CSS для Starbuzz-страницы	521
Теперь нужно подкорректировать <code><div></code> с идентификатором <code>main</code>	522
Протестируем страницу с абсолютным позиционированием	523
Как работает табличное представление CSS	525
Написание CSS и HTML для табличного представления	526
Добавление HTML-структуры для табличного представления	527
Как использовать CSS для создания табличных представлений	529
Вернемся к странице Starbuzz...	530
В чем проблема с промежутками?	531
Устранение проблемы с промежутками	532
Стратегии для вашего набора CSS-инструментов разметки	535
Проблемы с верхним колонтитулом	538
Позиционирование изображений для верхнего колонтитула с использованием <code>float</code>	539
Добавление награды	542
Как работает фиксированное позиционирование	545
Размещение купона на странице	546
Использование отрицательного значения свойства <code>left</code>	547

12 HTML5-разметка

Современный HTML

Мы уверены, что вам доводилось слышать о шумихе вокруг HTML5. И, принимая во внимание то, насколько далеко вы продвинулись в чтении данной книги, вы, вероятно, задаетесь вопросом о том, была ли ее покупка правильным поступком. Сейчас необходимо прояснить одну вещь: все, что вы изучили ранее по ходу книги, было HTML, точнее говоря, отвечало стандарту HTML5. Однако существует ряд новых аспектов HTML-разметки, появившихся с приходом стандарта HTML5, которые мы еще не рассматривали, чем как раз и займемся в этой главе. Большинство этих нововведений имеют эволюционный характер и, учитывая всю тяжелую работу, которую вы проделали по ходу изучения данной книги, вы без особого труда сможете в них разобраться. Среди них также имеются революционные элементы (например, `<video>`), о которых мы тоже поговорим в этой главе. Итак, давайте приступим и взглянем на эти нововведения!



Пересмотр HTML-структуры	558
Современное кафе Starbuzz	560
Обновление HTML-разметки Starbuzz	563
Прежде, чем вы продолжите...	565
Как обновить CSS, чтобы отразить новые элементы	566
Создание блог-страницы Starbuzz	574
Написание CSS для блог-страницы	575
Нам все еще нужно добавить дату в блог...	577
Добавление элемента <code><time></code> в файл <code>blog.html</code>	578
Как внедрить дополнительные элементы <code><header></code>	580
Что же случилось с верхними колонтитулами?	582
Обеспечение навигации	586
Добавление CSS для навигационных элементов	586
Кому нужна система GPS? Тестирование навигации	587
Добавление элемента <code><nav></code> ...	587
Делаем наш CSS более конкретным...	589
Оба-на! Посмотрите-ка на эту навигацию!	589
Создание нового блог-элемента	592
А теперь встречайте элемент <code><video></code>	592
Свет, камера, мотор...	593
Как работает элемент <code>video</code> ?	595
Пристальный взгляд на атрибуты элемента <code><video></code>	596
Что нужно знать о видеоформатах	598
Конкурирующие видеоформаты	599
Как «жонглировать» всеми этими форматами...	601
Дубль № 2: свет, камера, мотор...	602
Как обеспечить еще большую конкретизацию в случае с видеоформатами	602

таблицы и большие списки

13

Представление в табличной форме

Если данные лучше оформить в виде таблицы... Пришло время научиться работать с устрашающими табличными данными. Каждый раз, когда вам нужно создать страницу, на которой выводится список документов вашей компании за последний год или перечень вашей коллекции виниловых фигурок анимационных персонажей (не беспокойтесь, мы никому не расскажем об этом), вы знаете, что нужно использовать HTML. Но как? Мы готовы заключить с вами соглашение: выполняйте все наши инструкции — и за одну главу вы узнаете все секреты, позволяющие поместить данные прямо в HTML-таблицы. Но есть кое-что еще: с каждой инструкцией мы будем представлять вам информацию из нашего эксклюзивного руководства по стилизации HTML-таблиц. Если вы начнете прямо сейчас, то в качестве специального бонуса мы представим вам руководство по стилизации HTML-списков. Не сомневайтесь, соглашайтесь прямо сейчас!

Как мы создаем таблицы в HTML	613
Что создает браузер	615
Разделение таблицы	616
Добавление заголовка	619
Сделаем тест... и подумаем о стиле	620
Перед тем как перейти к приданию стиля, поместим таблицу на страницу Тони	621
Оформляем таблицу	622
Объединение границ	626
Как насчет цвета?	628
Мы говорили, что Тони сделал очень интересное открытие в Трут-ор-Консекуэнсес?	630
Посмотрим на таблицу Тони еще раз	631
Как сделать, чтобы ячейка охватила несколько строк	632
Проблема в раю?	635
Переопределение CSS для заголовков вложенной таблицы	639
Доведем сайт Тони до совершенства	640
Оформление списка	641
Если нужен маркер особой формы?	642



Город	Дата	Температура	Высота	Население	Рейтинг предпринимателя
Вала-Вала, штат Вашингтон	15 июля	75	1204 фута	29666	4/5
Маджик-Сити, штат Айдахо	25 июля	74	5312 футов	50	3/5
Базингфорд, штат Огайо	10 июля	95	4126 футов	41173	4/5
Дарт Чанд, штат Колорадо	23 июля	152	4780 футов	245	3/5
Трут-ор-Консекуэнсес, штат Нью-Мексико	9 августа	93	4242 фута	7289	5/5
	27 августа	96			Текст: 5/5 Тони: 4/5
Уай, штат Аризона	18 августа	104	960 футов	480	3/5

14 HTML-формы

Переход на интерактивный режим

До сих пор ваше веб-общение было односторонним: от веб-страницы к пользователям. Разве не было бы здорово, если бы пользователи смогли отвечать вам? Именно на этом этапе вступают в действие формы HTML. Как только вы снабдите свои страницы формами (прибегнув к определенной помощи веб-сервера), вы сможете собирать отзывы клиентов, принимать по Сети заказы, делать следующий ход в игре в режиме онлайн или проводить интернет-голосования. В этой главе вы познакомитесь с целой группой HTML-элементов, предназначенных для создания веб-форм. Вы также узнаете кое-что о том, что происходит на сервере для поддержки форм и как сделать сами формы стильными.



Как работают формы	654
Как формы работают в браузере	655
Что вы пишете в HTML	656
Что создает браузер	657
Как работает элемент <code><form></code>	658
Что может входить в форму	660
О, в форму может входить и еще кое-что!	664
Подготовка к созданию формы для Bean Machine	668
Из чего состоит элемент <code><form></code>	668
Добавление элемента <code><form></code>	669
Как работают атрибуты name элементов формы	670
Вернемся к размещению элементов <code><input></code> в HTML	672
Добавим в форму еще несколько элементов <code><input></code>	673
Добавление элемента <code><select></code>	674
Предоставьте клиенту выбор — молотый кофе или кофе в зернах	676
Оформление переключателей	677
Использование других типов <code><input></code>	678
Добавление таких типов <code><input></code> , как <code>number</code> и <code>date</code>	679
Дополнение формы	680
Добавление флажков и многострочного текстового поля	681
Проверим GET на практике	687
Размещение элементов формы в HTML-структуре для макета табличного представления	692
Оформление формы с помощью CSS	694
Несколько слов о доступности	696
Что еще может входить в форму?	697
Дополнительные вещи, которые могут входить в форму	698

Приложение

Топ-10 тем, которые не были освещены в этой книге

В книге мы рассмотрели очень много тем, и вы уже почти закончили свое обучение по ней. Однако мы не можем со спокойной душой отпустить вас, не рассказав еще кое-что напоследок. Как бы мы ни старались, но не сможем уместить в это маленькое приложение все то, что вам нужно знать, поэтому мы сначала включили в него все, что вам нужно знать об HTML и CSS (все то, что не было освещено в предыдущих главах), уменьшив размер шрифта на 0,00004. Все вместилось, но прочесть это было нереально. В итоге мы все же выбросили большую часть информации и поместили в приложение только десять самых важных тем.

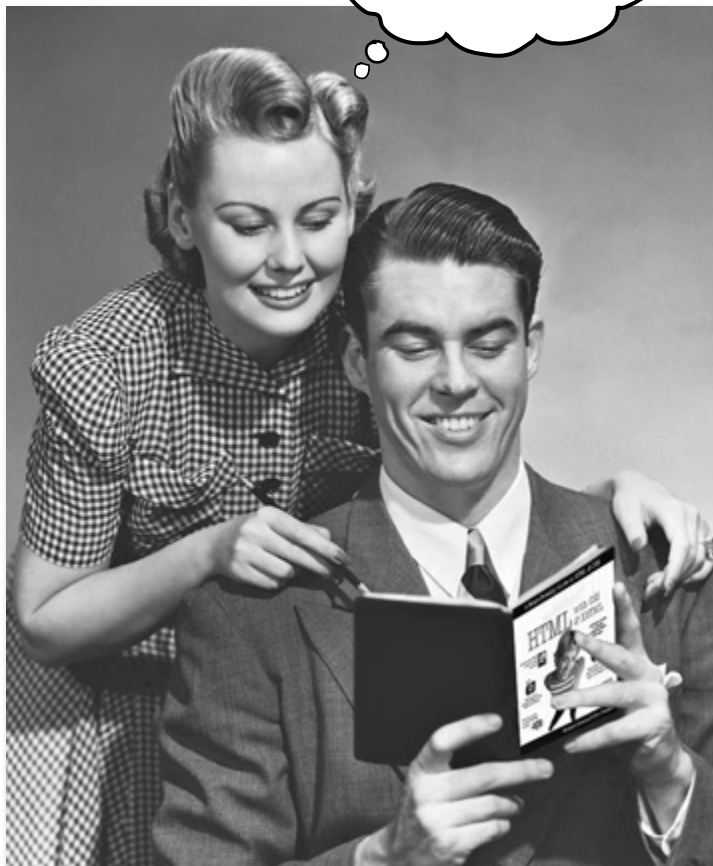


№ 1. Дополнительные типы CSS-селекторов	706
№ 2. Определяемые поставщиками CSS-свойства	708
№ 3. CSS-трансформации и переходы	709
№ 4. Интерактивность	711
№ 5. API-интерфейсы HTML5 и веб-приложения	712
№ 6. Подробнее о веб-шрифтах	714
№ 7. Сервисные программы для создания веб-страниц	715
№ 8. XHTML5	716
№ 9. Серверные сценарии	717
№ 10. Элемент <audio>	718

как работать с этой книгой

Введение

Не могу поверить,
что они пишут **такое**
в книге по HTML!



В этой части мы ответим на животрепещущий вопрос:
«Итак, почему они пишут такое в книге по HTML?»»

Для кого написана эта книга?

Если вы ответите «да» на все следующие вопросы...

- ① У вас есть доступ к компьютеру с **браузером и текстовым редактором**?
- ② Вы хотите **понять и запомнить**, как **создавать** веб-страницы, используя наилучшие методы и самые современные стандарты?
- ③ Вы предпочитаете **эмоциональные беседы** на званых обедах **сухим, скучным академическим лекциям**?

Если у вас есть доступ к любому компьютеру не более чем десятилетней давности выпуска, ответ — «да».

...значит, эта книга для вас.

Кому не стоит читать эту книгу?

Если вы ответите «да» хотя бы на один из следующих вопросов...

- ① Вы **абсолютно незнакомы с компьютерами**?
(Вам не обязательно быть продвинутым пользователем, но вы должны понимать, что такое файлы и папки, как работать в простых текстовых редакторах, и уметь пользоваться браузером.)
- ② Вы опытный веб-программист, которому нужен **справочник**?
- ③ Вы **боитесь попробовать что-нибудь новенькое**? Вы предпочитаете один скучный цвет рисунку в клетку? Вы не верите, что техническая книга может быть серьезной и что теги HTML наделены человеческими качествами?

...то эта книга не для вас.



[Замечание от отдела маркетинга:
вообще-то эта книга для любого, у кого
есть деньги.]

Мы знаем, что вы думаете

«Как эта книга может быть серьезной?»

«Зачем здесь столько картинок?»

«Неужели так можно чему-то научиться?»

Мы также знаем, о чем думает ваш мозг

Ваш мозг страстно желает нововведений. Он постоянно ищет и *ждет* чего-нибудь необычного. Он так устроен, и это помогает вам выжить.

Сегодня у вас меньше шансов стать закуской для тигра. Но ваш мозг все еще на чеку, а вы просто не замечаете этого.

Итак, что же ваш мозг делает с рутинными, обычными, нормальными ситуациями, с которыми вы сталкиваетесь? Все, что он *может*, — остановить их *вмешательство* в действительно *важную* для себя работу. Он не утруждает себя, чтобы сохранить что-то скучное. Все неинтересное и обычное просто проходит через фильтр «абсолютно неважных вещей».

Каким же образом ваш мозг узнает, что важно, а что нет? Предположим, вы выбрались на прогулку за город и перед вами высочил тигр. Что произойдет в вашей голове и вашем теле? Возбуждение нервных клеток. *Эмоциональный всплеск.*

И тогда ваш мозг понимает...

Это очень важно! Не забудь это!

Но представьте, что вы дома или в библиотеке. Здесь сухо и тепло и нет никаких тигров. Вы учитесь. Готовитесь к экзамену. Или пытаетесь разобраться в очень сложной технической теме, на что, как полагает ваш начальник, достаточно недели или в крайнем случае десяти дней.

Но вот одна проблема. Ваш мозг пытается оказать вам большую услугу. Он пытается *обеспечить сохранение* этой явно неважной информации таким образом, чтобы не загромождать и без того небольшие ресурсы для ее хранения. Ресурсы, которые лучше потратить для хранения чего-то действительно *важного*. Как тигры. Как опасность огня. Как то, что вы больше никогда не должны кататься на лыжах в шортах.

И нет простого способа сказать мозгу: «Эй, мозг, большое спасибо, но какой бы скучной тебе ни казалась эта книга и что бы ты ни думал сейчас, я действительно хочу, чтобы ты запомнил все это».

Ваш мозг думает:
ЭТО важно.



Великолепно.
Остается еще
695 скучных,
сухих страниц.

Ваш мозг думает,
что ЭТО запоминать не стоит.



Мы считаем, что читатель этой книги учится

Итак, что же нужно для того, чтобы *научиться* чему-либо? Сначала вы должны это *узнать*, а затем убедиться, что не *забыли*. Дело вовсе не в зубрежке. Последние исследования в областях когнитологии, нейробиологии и педагогической психологии показали, что для обучения необходимо намного больше, чем просто текст на странице. Мы знаем, что заставляет ваш мозг заработать в полную силу.

Некоторые принципы этой книги

Добиваться визуализации. Изображения запоминаются намного лучше, чем слова, и делают процесс обучения более эффективным. Кроме того, визуализация делает предмет изучения более понятным.

Лучше поместить надписи внутри графиков, к которым они имеют отношение, или рядом с ними, чем внизу текущей страницы или на следующей странице. Тогда учащиеся будут в состоянии в два раза быстрее решить задачу по этому материалу.

Использовать разговорный стиль и обращение к читателю.

Новейшие исследования показали, что студенты выполняют тесты после изучения материала до 40 % лучше, если в книге идет обращение напрямую к читателю, от первого лица и с использованием разговорного стиля вместо официального. Рассказывай историю вместо чтения лекции. Используй шуточный язык. Не будь слишком серьезным. Кто сильнее привлечет внимание: интересный собеседник на званом обеде или лектор?

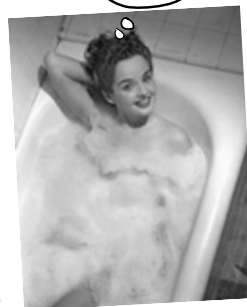
Действительно обидно забыть главное.



Главное — как распределена информация на странице.

Стимулировать ученика мыслить более глубоко. Иными словами, пока вы активно напрягаете свои извилины для обучения, в вашей голове больше ничего не происходит. Читатель должен иметь хорошую мотивацию учиться, быть заинтересованным, любознательным. Он должен с воодушевлением решать задачи, делать выводы и осваивать новые темы. Для этого ему нужны задачи, вопросы и упражнения, в которых есть над чем задуматься, и действия, для которых необходимо напрягать оба полушария головного мозга и даже использовать различные органы чувств.

Ведь нет смысла улучшать внешний вид ванной комнаты с помощью новых крючков для полотенец, лучше переделать всю комнату.



Захватывать и удерживать внимание читателя. Каждый из нас знаком с таким: «Я действительно хочу это выучить, но засыпаю после первой же страницы». Ваш мозг обращает внимание на все необыкновенное, интересное, странное, броское, неожиданное. Изучение новых, трудных для понимания технических вопросов не должно быть скучным. Ваш мозг намного быстрее освоит новую информацию, если она не **будет нудной**.

Использовать эмоциональную составляющую. Вы знаете, что способность запоминать что-либо намного выше, если затрагиваются чувства? Вы запоминаете то, что вас волнует. Вы запоминаете свои ощущения. Нет, мы, конечно же, не говорим о душеспасительных историях про мальчика и его собаку. Мы говорим об удивлении, любознательности, забаве, о возникающем у вас вопросе «О, как же это возможно?...» и таком чувстве, как «Я гений!», которое приходит после того, как вы разрешили головоломку, выучили что-то, что остальным кажется очень сложным, или реализовали что-то, что инженер Боб, считающий себя намного более продвинутым в этой области, не смог сделать.



Браузеры делают запросы на HTML-страницы или на другие ресурсы, такие как изображения.



Метапознание: учимся учиться

Если вы действительно хотите чему-то научиться, и притом научиться быстрее и глубже, подумайте над тем, как вы думаете. Изучите то, как вы учите.

Большинство из нас в детстве не проходили курс по метапознанию, или теории обучения. *Предполагалось*, что мы будем обучаться, но вряд ли нас учили это делать.

Но мы надеемся, что если вы держите в руках эту книгу, то действительно хотите научиться создавать веб-страницы. И, вероятно, вы не хотите тратить много времени, а хотите *запомнить* то, что прочитаете, чтобы затем суметь применить это на практике. А для этого нужно *понять* прочитанное. Для того чтобы вынести как можно больше из этой (или любой другой) книги или иного источника знаний, возьмите на себя ответственность за работу своего мозга.

Хитрость заключается в том, чтобы внушить мозгу, что новый материал, который вы изучаете, действительно важен. Является ключевым для вашего благополучия. Так же важен, как тигр. В противном случае вы будете постоянно воевать со своим мозгом, прилагая все усилия для запоминания новой информации и ее сохранения.



Итак, как же внушить мозгу, что HTML и CSS так же важны, как и тигр?

Существует два способа: нудный и медленный и более быстрый и эффективный. Медленный способ использует постоянное повторение. Вы наверняка знаете, что способны выучить и запомнить даже самый скучный материал, если постоянно повторять одно и то же. При достаточном количестве повторений ваш мозг скажет: «Я не думаю, что это важно для него, но он постоянно смотрит на это. Поэтому я допускаю, что это может иметь для него значение».

Быстрый способ — сделать что-нибудь, что *увеличит активность работы мозга*, и особенно различные способы его работы. Описанное выше — важная составляющая решения проблемы, и уже доказано, что все это поможет вашему мозгу оказать вам услугу. Например, исследования показали, что если расположить слова *внутри* картинки, которую они описывают (вместо того чтобы помещать их в какой-то другой части страницы, например в заголовке или в основном тексте), то это мотивирует мозг попытаться понять, как же взаимосвязаны слова и картинка, что, в свою очередь, приводит к возбуждению большего количества нервных клеток. Чем больше нервных клеток возбудится, тем больше шансов, что мозг воспримет информацию как ту, на которую нужно обратить внимание и, возможно, сохранить.

Разговорный стиль помогает усвоить информацию, так как люди имеют тенденцию лучше концентрироваться, если осознают, что беседуют с кем-то, а не когда они изучают материал самостоятельно.

Удивительно, но вашему мозгу не важно, что беседа идет между вами и книгой! С другой стороны, если стиль изложения официальный и скучный, то мозг воспринимает информацию так, будто вы сидите на лекции в зале, полном пассивных слушателей. И нет необходимости бодрствовать.

Но картинки и разговорный стиль — это только начало.

Вот что Мы делали

Мы использовали *рисунки*, потому что ваш мозг больше настроен на восприятие изображений, а не текста. С точки зрения мозга рисунок действительно стоит 1024 слов. А еще лучше, если текст и изображение работают вместе. Мы добавляли текст внутрь картинки, потому что мозг работает более эффективно, если текст находится внутри того, что он описывает, а не, например, в названии или еще где-то.

Мы использовали *повторение*, говоря об одном и том же разными способами и применяя различные формы представления информации. Так повышается вероятность того, что информация будет закодирована в нескольких зонах вашего мозга.

Мы использовали *необычные* понятия и рисунки, так как ваш мозг хорошо усваивает все новое. Рисунки и идеи наделялись хотя бы некоторым *эмоциональным* содержанием, потому что ваш мозг тесно связан с биохимией эмоций. Если вы что-то почувствуете, вероятность того, что вы это запомните, — больше. Пусть даже это всего лишь *шутка*, *сюрприз* или *интересный факт*.

Мы использовали *разговорный стиль*, потому что мозг больше настроен на усвоение информации, если полагает, что вы с кем-то беседуете, а не пассивно слушаете. Это происходит даже тогда, когда вы читаете.

Мы включили в книгу более 100 *упражнений*, так как ваш мозг лучше учит и запоминает, когда вы что-то *делаете*. Мы сделали упражнения сложными, но выполнимыми, так как именно такие упражнения предпочитает большинство людей.

Мы использовали *различные стили обучения*, потому что одни предпочитают изучение шаг за шагом, в то время как другие хотят сначала получить общее представление о проблеме, а третьи просто хотят увидеть пример. Однако, несмотря на личные предпочтения, каждому будет полезно ознакомиться с одним и тем же материалом различными способами.

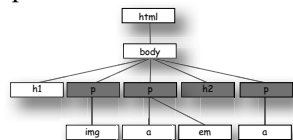
Мы включили в книгу информацию для *обоих полушарий мозга*, потому что чем большую часть мозга вы используете, тем более успешно вы учитесь и запоминаете и тем дольше можете концентрироваться на изучаемой теме. Поскольку при работе одного полушария головного мозга второе имеет возможность отдохнуть, вы можете эффективно учиться более продолжительное время.

Мы также включили в издание *истории*, которые представляют *несколько точек зрения*, так как мозг начинает лучше усваивать материал, если вынужден оценивать и анализировать.

Мы включили в книгу *сложные задачи*, упражнения и *вопросы*, на которые не всегда есть однозначный ответ, потому что ваш мозг лучше учится и запоминает, если ему есть над чем поработать. Согласитесь, вы не можете привести свое тело в спортивную форму, просто наблюдая за тем, как другие занимаются в спортзале. Мы сделали все возможное, чтобы обеспечить для вас *максимальный результат при условии вашей прилежной работы*. Иными словами, вам не придется тратить время на разбор сложных для понимания упражнений, трудного, нагруженного непонятными терминами или излишне сжатого текста.

Мы использовали *людей* в историях, примерах, рисунках и т. д., потому что вы тоже человек. И ваш мозг обращает больше внимания на людей, чем на предметы.

Мы использовали принцип *80/20*. Мы полагаем, что если вы собираетесь стать опытным веб-программистом, то эта книга не станет для вас единственным источником знаний. Поэтому мы не рассказываем обо всем. Речь пойдет лишь о том, что действительно важно.



СТАНЬ браузером



КЛЮЧЕВЫЕ
МОМЕНТЫ



ГОЛОВОЛОМКИ





А вот что можете сделать Вы, чтобы заставить свой мозг работать

Вырежьте листок с написанием и наклейте его на холодильник.

Итак, мы свою часть работы сделали. Остальное зависит от вас. Следующие подсказки — лишь отправная точка; прислушивайтесь к своему мозгу и выясняйте, что для вас работает, а что — нет. Пробуйте какие-то новые способы.

① Не торопитесь все прочитать. Чем больше вы поймете, тем меньше вам придется запоминать.

Не *читайте* просто так. Делайте паузы и вдумывайтесь. Если в книге вам задан вопрос, не пропускайте его. Представьте, что кто-то действительно обращается к вам с вопросом. Чем более глубоко вы заставляете свой мозг вникать в суть проблемы, тем выше вероятность того, что вы запомните и усвоите материал.

② Выполняйте упражнения. Делайте свои собственные заметки.

Мы включили в книгу упражнения, но выполнять их за вас не собираемся. Не *смотрите* на упражнения просто так. Пользуйтесь карандашом. Существует множество доказательств тому, что физическая активность *во время* обучения повышает его эффективность.

③ Читайте «Часто задаваемые вопросы»

Все вопросы важны. Они являются частью основного материала! Не пропускайте их.

④ Не читайте другие книги перед сном. Или, по крайней мере, другие интересные книги.

Часть обучения (особенно сохранение информации в долговременную память) происходит уже *после* того, как вы отложили книгу. Вашему мозгу нужно время для обработки и запоминания информации. Если вы в это время добавите какую-то новую информацию, то часть предыдущей будет потеряна.

⑤ Пейте воду. Много воды.

Мозг лучше работает в условиях «высокой влажности». Обезвоживание (которое может случиться еще до того, как вы почувствуете жажду) снижает функцию запоминания.

⑥ Проговаривайте все, что читаете. Говорите громко.

За речевую активность отвечает отдельная зона мозга. Если вы пытаетесь что-то понять или хотите лучше что-то запомнить, громко проговорите это. А лучше попытаться объяснить это вслух кому-то другому. В результате вы будете учиться быстрее и, возможно, обнаружите что-то новое, чего не замечали в процессе чтения.

⑦ Прислушивайтесь к своему мозгу.

Следите за тем, чтобы мозг не перегружался информацией. Если вы ловите себя на том, что читаете невдумчиво или забываете то, что сейчас прочитали, значит, настало время отдохнуть. Перечитывая одно и то же снова и снова, пытаясь впихнуть в себя больше информации, вы не выучите быстрее и даже, возможно, навредите всему процессу обучения.

⑧ Пытайтесь прочувствовать прочитанное!

Вашему мозгу необходимо осознавать, что информация *важна*. Пытайтесь увлечься сюжетом. Выдумывайте свои собственные комментарии к фотографиям. Лучше, например, поворчать, что шутка несмешная, чем остаться полностью равнодушным.

⑨ Создавайте что-нибудь!

Применяйте прочитанное для проектирования чего-либо нового или для переработки старых проектов. Просто делайте *что-нибудь* (кроме заданий и упражнений из книги), чтобы приобрести опыт. Все, что вам нужно, — это карандаш и задача, которую нужно решить. Задача, которая может быть полезна для изучения HTML и CSS.

Примите к сведению

Эта книга — учебник, а не просто справочник. Мы рассмотрели все ситуации, которые могут возникнуть в процессе обучения, и создали книгу с учетом этого. Читая ее первый раз, вы должны начать с самого начала, поскольку мы постоянно предполагаем, что вы уже знаете все вышеизложенное.

Мы начинаем с изучения исходных положений HTML, затем переходим к HTML5.

Чтобы писать на HTML, основанном на общепринятых стандартах, нужно понимать много технических деталей, которые бесполезны, пока вы еще только знакомитесь с основами языка. Наш подход заключается в том, чтобы сначала научить вас основным понятиям HTML (не останавливаясь на деталях), а затем, когда вы будете все хорошо понимать, научить вас писать на HTML (самой последней версией которого является HTML5), основанном на общепринятых стандартах. В этом есть дополнительная польза, так как технические детали имеют больше смысла после того, как изучены основы.

Важно также, что к моменту использования CSS вы уже будете писать на HTML, основанном на общепринятых стандартах.

Мы не описываем все подряд элементы и атрибуты HTML и свойства CSS, которые когда-либо были созданы.

Существует *множество* элементов и атрибутов HTML и свойств CSS. Бесспорно, все они интересны, но нашей целью было написание книги, которая весит меньше, чем ее читатель, поэтому мы не описываем их все. Мы обращаем внимание только на основные элементы HTML и свойства CSS, которые действительно *важны* для начинающих, и надеемся, что вы по-настоящему поймете, как и когда их использовать. В любом случае после прочтения этой книги вы сможете взять любой справочник и быстро ознакомиться со всеми элементами и свойствами, которые мы пропустили.

В книге четко разделены понятия структуры и представления ваших страниц.

В наши дни при создании серьезных веб-страниц применяется HTML для структурирования их содержимого, а также CSS для стилизации и дизайна. В веб-страницах 1990-х годов часто использовалась другая модель, где HTML применялся и для структуризации, и для стилизации. Мы научим вас использовать HTML для структуризации, а CSS для стилизации; мы не видим смысла учить вас дурным устаревшим привычкам.

Мы поощряем использование нескольких браузеров.

Пока мы учим вас писать на HTML и CSS, основанных на стандартах, вы по-прежнему (и, скорее всего, и далее) будете сталкиваться с небольшими различиями в способах отображения страниц разными браузерами. Поэтому

мы советуем выбрать по крайней мере два современных браузера и тестировать в них веб-страницы. Это научит вас понимать различия между браузерами и создавать страницы, которые хорошо работают во множестве браузеров.

Мы часто используем название «элемент» вместо названия «тег».

Вместо того чтобы говорить «тег `<a>`», мы говорим «элемент `<a>`». Пусть формально это некорректно (так как `<a>` — это открывающий тег, а не целый элемент), но это улучшает читабельность текста, и мы обычно используем слово «элемент» во избежание путаницы.

Упражнения обязательны для выполнения.

Задания и упражнения — это не дополнения, а часть основного материала книги. Одни из них предназначены для облегчения запоминания, другие — для применения выученного материала. *Не пропускайте упражнения.*

Повторения используются специально, и это важно.

Одна отличительная особенность книги — то, что мы действительно хотим донести информацию до читателя. Мы также хотим, чтобы читатель ничего не забыл после прочтения книги. Большинство справочников не ставят своей целью запоминание и воспроизведение читателем содержащейся в них информации, но эта книга готова *обучать*, поэтому некоторые понятия появляются не один раз.

Примеры настолько компактны, насколько это возможно.

Наши читатели жаловались на то, что в некоторых примерах приходится просматривать 200 строк листинга, чтобы добраться до двух действительно важных для понимания. Большинство примеров в этой книге очень краткие, и их суть понятна сразу. Не ожидайте, что все примеры будут корректными, они пишутся специально для обучения и не всегда работают полностью правильно.

Все файлы с примерами мы выложили в Интернете, и вы можете скачать их на сайте <http://wickedlysmart.com/hfhtml/css>.

К упражнениям «Мозговой штурм» ответы не прилагаются.

У некоторых из них правильных ответов вовсе не существует, в других случаях используйте свой опыт, чтобы определить правильный ответ. В некоторых примерах повышенной сложности есть подсказки, которые приведут к правильному ответу.

1 знакомство с HTML

Язык Сети



Не так быстро... Сначала вы должны научиться говорить на **универсальном языке**, то есть на HTML и CSS.

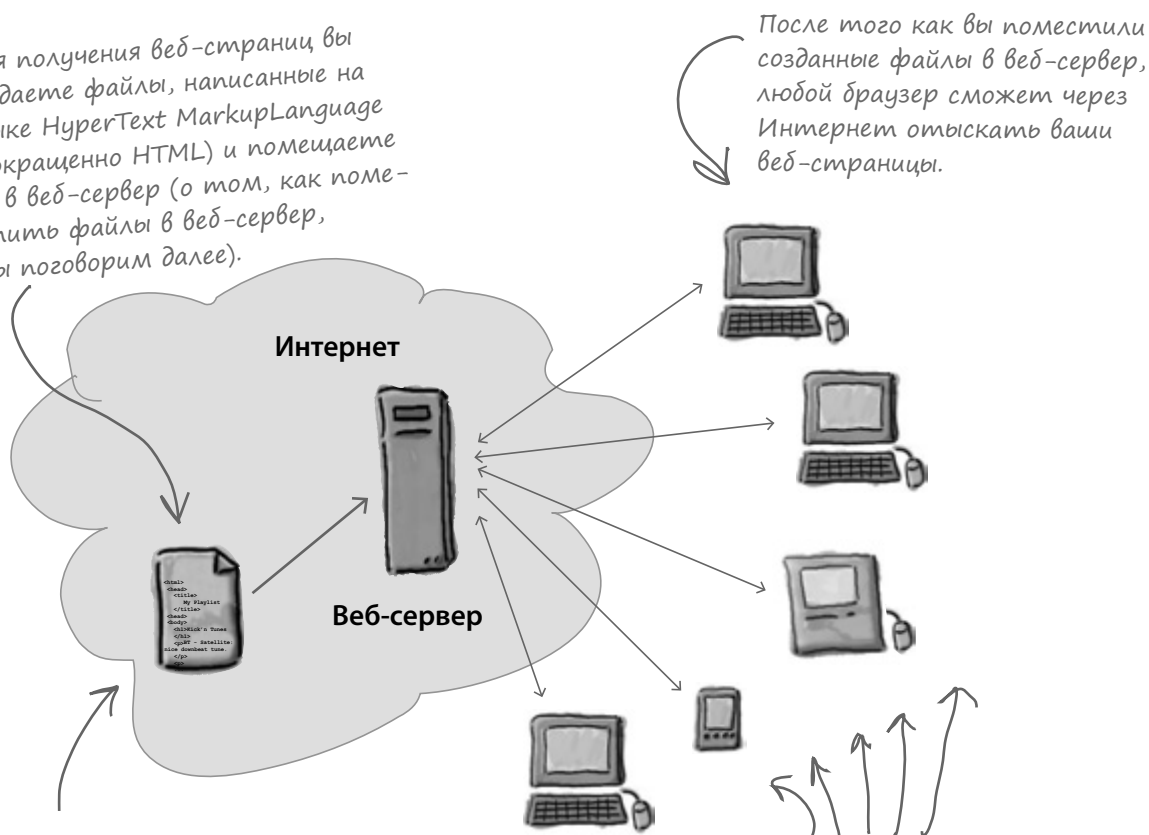
Единственное, что необходимо для того, чтобы успешно работать в Сети, — научиться говорить на ее специфическом языке: HyperText Markup Language (язык гипертекстовой разметки), или сокращенно HTML. Итак, приготовьтесь к нескольким урокам языка. После этой главы вы не только узнаете некоторые базовые понятия HTML, но и сможете разговаривать на этом языке, используя определенный стиль. Черт возьми, к концу этой книги вы сможете говорить на языке HTML так, будто выросли в Сети!

Сеть ~~Видео~~ убило радиозвезду

Хотите поделиться своей новой идеей? Продать что-то? Просто нужен творческий выход? Обратитесь к Сети — нет необходимости говорить вам, что она стала универсальной формой коммуникации. Кроме того, это форма коммуникации, в которой **ВЫ** можете участвовать.

Но если вы действительно хотите эффективно использовать возможности Сети, то вам необходимо узнать кое-какие вещи про **HTML**, не говоря уже о том, что нужно понимать, как работает сама Сеть. Начнем с самого главного.

Для получения веб-страниц вы создаете файлы, написанные на языке HyperText Markup Language (сокращенно HTML) и помещаете их в веб-сервер (о том, как поместить файлы в веб-сервер, мы поговорим далее).

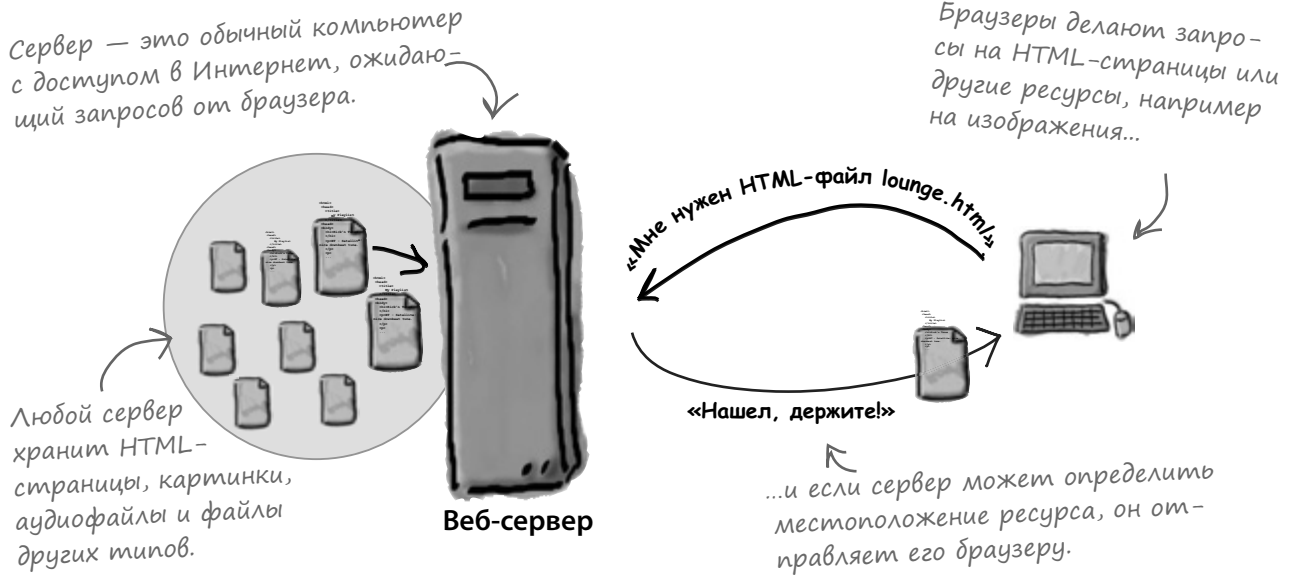


Язык HTML вашей веб-страницы говорит браузеру все, что ему необходимо знать для отображения этой страницы. А если вы выполнили свою работу хорошо, то ваши страницы будут так же корректно отображаться на экранах сотовых телефонов и прочих мобильных устройств и смогут работать с речевыми браузерами и экранными дикторами для людей с плохим зрением.

Существует множество компьютеров и устройств, подключенных к Интернету и отображающих веб-страницы. Очень важно, что ваши друзья, члены семьи, болельщики и потенциальные покупатели могут пользоваться всеми этими устройствами.

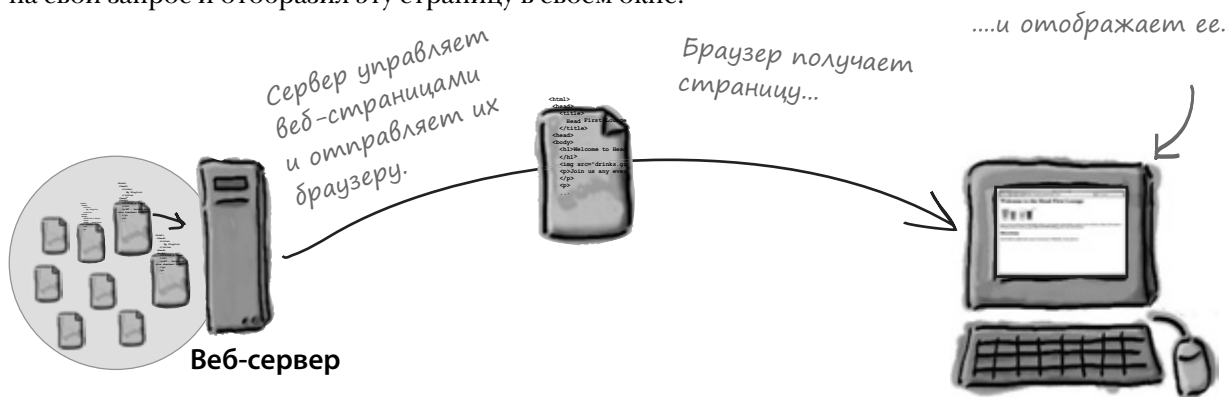
Что делает веб-сервер

Веб-серверы непрерывно работают в Интернете, неустанно ожидая запросов от браузеров. Каких именно запросов? Запросов на веб-страницы, изображения, аудиофайлы и, возможно, даже видеоролики. Когда сервер получает запрос на какой-нибудь из таких ресурсов, он находит этот ресурс и высылает его браузеру.



Что делает браузер

Вы уже знаете, как работает браузер: вы бродите по различным сайтам в Сети, щелкая на ссылках для перехода на различные страницы. Такой щелчок служит поводом для того, чтобы ваш браузер сделал запрос на HTML-страницу веб-серверу, получил ответ на свой запрос и отобразил эту страницу в своем окне.



Но каким образом браузер узнает о том, как именно отображать страницу? Вот здесь начинает работать язык HTML. Он говорит браузеру всё о содержании и структуре страницы. Посмотрим, как это работает...

Что пишете вы (HTML-kog)

Итак, вы знаете, что HTML — это ключ, благодаря которому браузер отображает страницы, но как именно выглядит HTML и что он делает?

Давайте посмотрим на небольшой фрагмент HTML-кода. Представьте, что вы собираетесь создать веб-страницу — рекламную *гостевую страничку* с хорошими мелодиями, освежающими напитками и беспроводным доступом. Рассмотрим код, который вы напишете на HTML:

```
<html>
```

```
<head>
```

```
<title>Гостевая Head First</title> (A)
```

```
</head>
```

```
<body>
```

```
<h1>Добро пожаловать в гостевую Head First</h1>
```

```
 (B)
```

```
<p>
```

^(D) Заходите к нам каждый вечер, чтобы попробовать освежающие напитки, поболтать и, возможно,

станцевать разок-другой. ^(E)

Всегда обеспечен беспроводной доступ
(захватите с собой свой ноутбук).

```
</p>
```

```
<h2>Указатели</h2> (F)
```

```
<p>
```

^(G) Вы найдете нас в самом центре Webville.

Присоединяйтесь к нам!

```
</p>
```

```
</body>
```

```
</html>
```



РАССЛАБЬТЕСЬ

Мы не ожидаем, что вы уже знаете HTML.

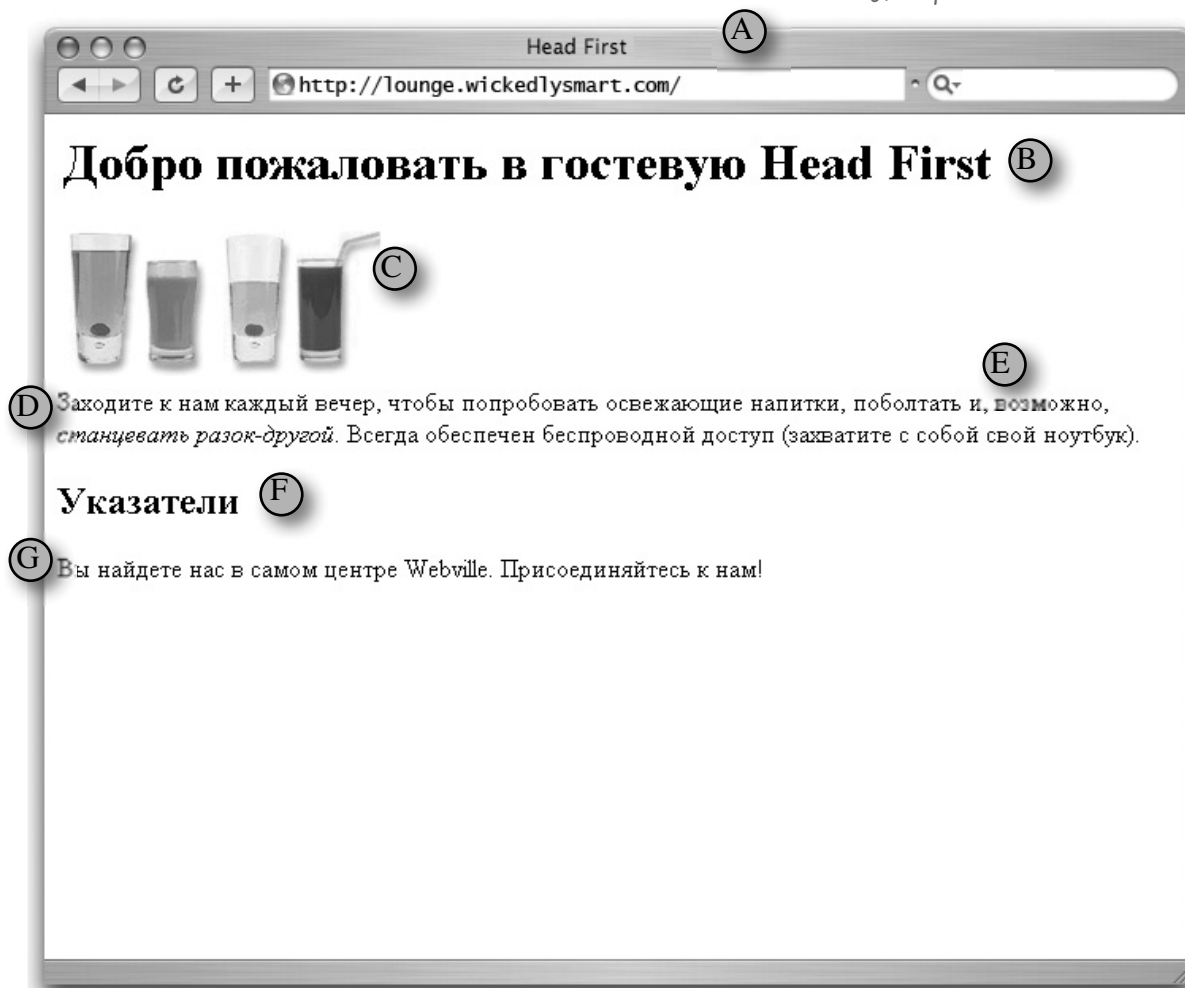
На данном этапе вы просто должны представлять, как выглядит HTML. Далее мы постепенно опишем всё более детально. А пока изучайте HTML-код и наблюдайте, как он отображается в окне браузера. Обязательно обращайтесь внимание на каждое замечание и ссылку, а также на то, как и где они отображаются в браузере.

Что создает браузер

Когда браузер читает вашу HTML-страницу, он интерпретирует все *теги*, которые окружают основной текст программы. Теги — это обычные слова или символы в угловых скобках, например `<head>`, `<p>`, `<h1>` и т. д. С помощью тегов браузер распознает *структуру и значение* вашего текста. Итак, вместо того, чтобы просто передавать браузеру кусок текста, благодаря HTML вы можете использовать теги, чтобы сказать браузеру, какая часть текста является названием, где начинается новый абзац, что нужно подчеркнуть и где расположить изображения.

Давайте проверим, как браузер интерпретирует теги в гостевой Head First.

Обратите внимание, как каждый тег в HTML соответствует тому, что отображается в браузере.



Часть Задаваемые Вопросы

В: Итак, HTML — это всего лишь набор тегов, которые я расставляю вокруг основного текста?

О: Для начинающих. Запомните, что HTML — это *HyperText Markup Language* (язык гипертекстовой разметки), то есть он дает возможность «пометить» ваш текст тегами, которые описывают браузеру его структуру. Помимо этого, у HTML есть *гипертекстовый* аспект, о котором мы поговорим далее в книге.

В: Как браузер определяет, каким образом отображать HTML?

О: HTML описывает браузеру структуру вашего документа: где заголовки, где начинаются новые абзацы, какой текст нужно подчеркнуть и т. д. Получив эту информацию, браузер использует встроенные в него по умолчанию правила о том, как отображать каждый из этих элементов.

Но вы можете добавить свой собственный стиль и правила редактирования с помощью языка CSS, который определяет шрифт, цвет, размер и многие другие характеристики страницы. Мы вернемся к CSS позже.

В: В HTML-коде для гостевой *Head First* используются отступы и соблюдаются интервалы, но все же я не вижу, где это отображается в браузере. Как так?

О: Браузеры игнорируют символы табуляции, абзацы и большинство пробелов в документе HTML. Вместо этого они используют заданную вами в коде разметку документа для определения мест, где появляется разрыв строки и конец абзаца. Для чего же мы форматируем текст, если браузер все равно проигнорирует это? Чтобы нам было легче читать HTML-документ, когда мы его редактируем. Поскольку HTML-документ со временем усложняется, то, обнаружив в нем

при редактировании пробелы, символы табуляции и абзацы, вы поймете, что они улучшают читабельность кода.

В: Существует два уровня заголовков: `<h1>` и `<h2>`?

О: На самом деле их шесть: от `<h1>` до `<h6>`. Браузер обычно отображает их, различая благодаря последовательному уменьшению размера шрифта. Заголовки дальше `<h3>` используются для сложных документов.

В: Почему необходим тег `<html>`? Разве и так не очевидно, что это HTML-документ?

О: Тег `<html>` говорит браузеру, что ваш документ является HTML-документом. Если одни браузеры не обратят внимания на то, что вы опустите этот тег, то другие вам этого не простят. Постараемся вам доказать, что использование этого тега важно.

В: Что делает файл HTML-файлом?

О: По своей сути HTML-файл — это обычный текстовый файл. В отличие от файлов специальных текстовых редакторов, он не имеет возможности форматирования. Обычно в конце имени файла добавляется `.html` или `.htm` (в системах, которые поддерживают только три буквы в расширении файла), чтобы система определила тип файла. Но вы же понимаете: имеет значение лишь содержимое файла.

В: Все говорят о HTML5. А используем ли мы его? Если да, то почему мы не говорим «HTML-ПЯТЬ» вместо «HTML»?

О: Вы изучаете HTML, а HTML5 — это просто самая последняя версия языка HTML. За последнее время HTML5 было уделено много внимания, благодаря чему он упрощает многие из способов написания

HTML-разметки и привносит новую функциональность, которую мы рассмотрим в данной книге. Он также обеспечивает ряд продвинутых опций посредством своих Javascript-интерфейсов прикладного программирования API (Application Programming Interface), которые рассматриваются в книге «*Head First HTML5 Programming*» («Изучаем программирование на HTML5»).

В: Мне кажется, что размечать документ — глупо. Программы, основанные на принципе «что видишь на экране, то и получишь при печати», используются с 1970-х годов. Почему же Сеть не основывается на формате, аналогичном Microsoft Word?

О: Сеть состоит из текстовых файлов без какого-то специального форматирования. Благодаря этому каждый браузер в любом уголке света может найти веб-страницу и «понять» ее содержимое. Программы, основанные на принципе «что видишь на экране, то и получишь при печати», например, Dreamweaver, отлично работают. В данной книге мы начнем с основы — с текста. Затем вы будете готовы понять, что приложение Dreamweaver делает «за кадром».

В: Есть ли какой-нибудь способ вставлять комментарии в HTML-код?

О: Да, если вы поместите свой комментарий между символами `<!--` и `-->`, то браузер их просто проигнорирует. Например, чтобы написать комментарий «Это начало нашей гостевой», нужно сделать так:

```
<!-- Это начало нашей гостевой -->
```

Обратите внимание, что комментарий может быть написан в несколько строк. Помните, что все написанное между `<!--` и `-->`, даже HTML-код, будет проигнорировано браузером.

Создание HTML-файла (Mac)

Все HTML-файлы — это текстовые файлы. Для создания текстового файла нужна программа, которая позволяет создавать обыкновенный текст без использования причудливого форматирования и специальных символов. Вам нужен простой, чистый текст.

В этой книге мы используем редактор TextEdit в Mac. Тем не менее если вы предпочитаете другой текстовый редактор, все тоже будет хорошо работать. А если вы работаете в Windows, то можете пропустить пару страниц и перейти сразу к инструкциям для Windows.

Шаг первый

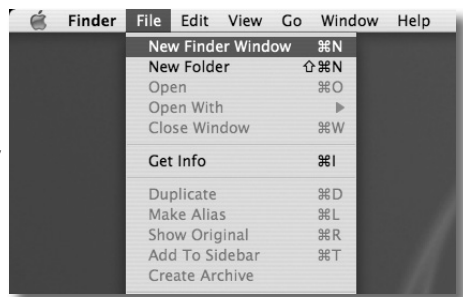
Направляйтесь в папку Applications (Приложения).

Ярлык программы TextEdit находится в папке Applications (Приложения). Самый простой способ туда попасть — выбрать пункт New Finder Window (Поиск в новом окне) в меню Finder's File (Файлы поиска) и затем искать папку Applications (Приложения) прямо в ярлыках. Когда найдете ее, щелкните на ней кнопкой мыши.

Шаг второй

Найдите и запустите TextEdit.

Вероятно, в вашей папке Applications (Приложения) будет много программ. Прокручивайте список вниз до тех пор, пока не увидите TextEdit. Для запуска программы дважды щелкните на ее значке.



Здесь TextEdit

Ваши ярлыки папок

Шаг третий (необязательный)

Держите TextEdit на панели запуска.

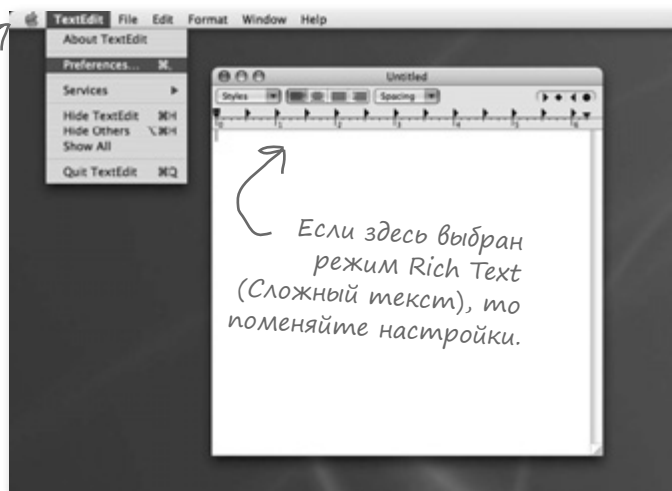
Щелкните (и держите кнопку мыши нажатой) на значке TextEdit на панели запуска (этот значок там появляется сразу после запуска программы). В появившемся меню выберите пункт Options (Опции), а затем — Keep in Dock (Установить на панель запуска). Таким образом, значок TextEdit всегда будет на панели запуска и вам не придется каждый раз, когда нужно открыть эту программу, искать ее в папке Applications (Приложения).



Шаг четвертый

Поменяйте настройки в TextEdit.

По умолчанию в TextEdit выбран режим RichText (Сложный текст), это означает, что в ваш файл при сохранении будут добавляться форматирование и специальные символы. Однако вам это не нужно. Поэтому необходимо поменять настройки в TextEdit, чтобы текст в вашем файле сохранялся в обычном виде. Для этого в меню TextEdit выберите подменю Preferences (Настройки).



Шаг пятый

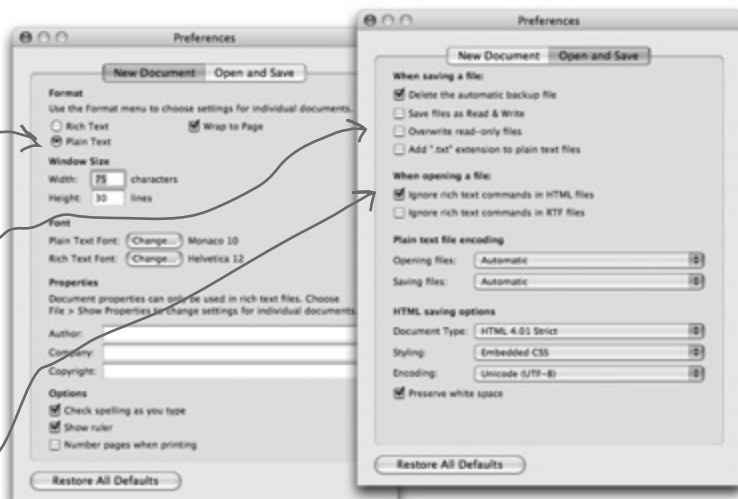
Установите режим Plain Text (Обычный текст).

В окне Preferences (Настройки) сделайте следующее.

Во-первых, выберите на вкладке New Document (Создание документа) в качестве используемого по умолчанию режим Plain Text (Обычный текст).

Во-вторых, убедитесь, что на вкладке Open and Save (Открытие и сохранение) установлен флажок Ignore rich text commands in HTML files (Игнорировать команды усложнения текста в HTML-файлах).

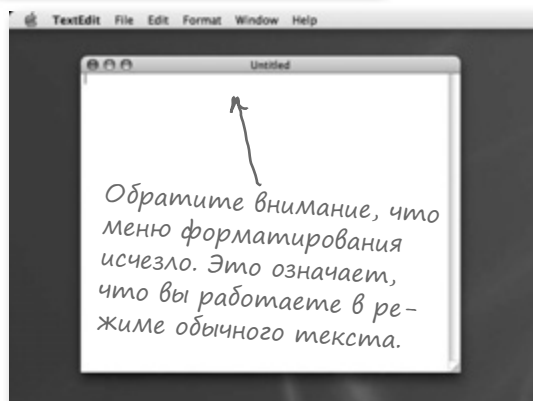
И последнее, убедитесь, что снят флажок Add .txt extension to plain text files (Добавлять расширение .txt к файлам с обычным текстом). Вот и все; щелкните на красной кнопке в левом верхнем углу, чтобы закрыть диалоговое окно Preferences (Настройки).



Шаг шестой

Закройте программу и откройте заново.

Сейчас, выбрав в TextEdit меню пункт Quit (Выход), выйдите из программы TextEdit и потом заново ее запустите. На этот раз сверху не будет причудливых меню форматирования. Можно приступить к созданию HTML-файлов.



Создание HTML-файла (Windows)

Если вы читаете эту страницу, то вы, вероятно, пользователь Windows 7. Если нет, то можете пропустить несколько следующих страниц. Но мы не против, если вы все равно ознакомитесь с предложенной информацией.

Для создания HTML-файлов в Windows 7 мы будем использовать программу Notepad (Блокнот). Она всегда есть в каждой новой версии Windows, и ею легко пользоваться. Если же у вас есть свой любимый редактор, который работает в Windows 7, то замечательно – можете использовать его. Только убедитесь, что в нем можно создавать файлы с расширением HTML.

Рассмотрим, как вы будете создавать свой первый HTML-файл при использовании программы Notepad (Блокнот).

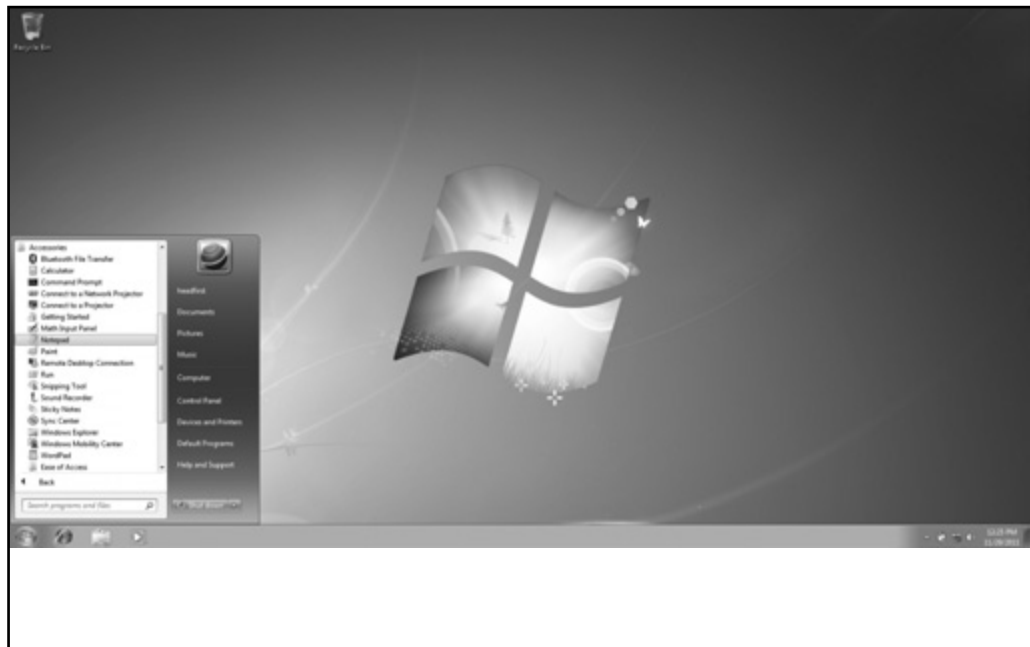
Или другой версии Windows.

Блокнот есть и в любой другой версии Windows.

Шаг первый

Откройте меню Start (Пуск) и найдите программу Notepad (Блокнот).

Программа Notepad (Блокнот) находится в стандартных программах. Самый простой способ туда попасть – открыть меню Start (Пуск), затем выбрать All Programs ► Accessories (Все программы ► Стандартные). Здесь в списке вы найдете программу Notepad (Блокнот).



Шаг второй

Откройте Notepad (Блокнот).

Выбрав в папке Accessories (Стандартные) программу Notepad (Блокнот), щелкните на ней кнопкой мыши. Откроется пустое окно, в котором вы можете начать набирать HTML-код.

Но рекомендуем.

Шаг третий (необязательный)

Не прячьте расширения файлов.

По умолчанию Explorer (Проводник) в Windows прячет зарегистрированные расширения файлов. Например, файл Irule.html будет показан как Irule без расширения .html.

Намного меньше путаницы будет, если вы укажете системе показывать расширения, поэтому мы расскажем вам, как это сделать.

Во-первых, откройте Folder Options (Свойства папки), щелкнув на кнопке Start (Пуск) → Control Panel (Панель управления) → Appearance and Personalization (Оформление и персонализация) → Folder Options (Свойства папки).

Затем на вкладке View (Вид) прокрутите список Advanced settings (Дополнительные параметры) вниз, пока не увидите флажок Hide extensions for known file types (Скрывать расширения для зарегистрированных типов файлов). Снимите его.

Все. Нажмите кнопку OK, чтобы применить настройки, и теперь в Explorer (Проводник) будут отображаться все типы файлов.

