

Содержание

Об авторе	13
Введение	15
О чем эта книга	15
Как пользоваться книгой	15
Предположения автора	15
Условные обозначения	16
Структура этой книги	16
Часть I. Создание первого веб-сайта	16
Часть II. Создание веб-страниц	16
Часть III. Сделайте свой сайт особенным	17
Часть IV. Попробуем разбогатеть	17
Часть V. Великолепные десятки	17
Приложение	17
Пиктограммы, используемые в книге	17
Что дальше	18
От издательства “Диалектика”	18
Часть I. Создание первого веб-сайта	19
Глава 1. Дизайн хорошего сайта: вначале планирование	21
Все начинается с плана	21
Чего вы хотите добиться	21
Какова ваша цель	22
Обновляйте свой сайт	23
Содержимое, создаваемое посетителями	24
Внешний вид вашего сайта	25
Ориентация на пользователей	25
Избегайте суеты	26
Удобство использования сайта	27
Главное, о чем следует помнить при планировании сайта	28
Интернет-ресурсы по веб-дизайну	29
Глава 2. Создание веб-страницы с помощью базовых HTML-элементов	31
Основные концепции HTML	31
Работа с текстом	34
Абзацы	34
Заголовки	35
Шрифты	36
Линии	37
Работа с цветом	38
Создание ссылок	39
Работа с изображениями	39
Изображения	41
Фоновые рисунки	42

А теперь собираем страницу	42
Интернет-ресурсы по разработке веб-страниц	43
Глава 3. Форматирование текста с помощью таблиц и других средств	45
Создание таблиц	46
Определение строк и столбцов	46
Задание размеров таблицы	47
Определение размеров ячейки	47
Наполнение и разделение ячеек	49
Связывание строк и столбцов	52
Выравнивание таблиц и их элементов	53
Фреймовые структуры	54
Построение веб-страниц на основе фреймов	55
Добавление границ и полей к фреймам	56
Автоматическая загрузка фреймовой структуры средствами JavaScript	57
Фреймы, таблицы и новая спецификация HTML 5	58
Интернет-ресурсы по таблицам и фреймам	58
Глава 4. Работа с WYSIWYG-средствами	59
Выбор WYSIWYG-редактора	59
Установка редактора KompoZer	63
Окно редактора KompoZer	64
Создание веб-страницы	66
Свойства страницы	66
Сохранение страницы	68
Ввод и форматирование текста	68
Вставка элементов	70
Публикация сайта	76
Глава 5. Выбор хостинга и доменного имени	79
Выбор хостинга	80
Интернет-провайдеры	80
Виртуальные серверы	80
Выделенные серверы	81
Поиск подходящего хостинг-провайдера	82
Бесплатный хостинг	83
Технические характеристики хостинга	84
Получение собственного доменного имени	85
Выбор доменного имени	87
Поиск регистратора	88
Интернет-ресурсы по веб-хостингу и регистрации доменных имен	89
Часть II. Создание веб-страниц	91
Глава 6. Каскадные таблицы стилей	93
Объединение CSS и HTML	93
Селекторы, классы и идентификаторы	96
Классы	97

Идентификаторы	98
Переопределение элементов	98
Контекстные селекторы	100
Другие возможности CSS	100
CSS и цвета	101
Границы	101
CSS и WYSIWYG	103
Слои	105
Абсолютное позиционирование	106
Относительное позиционирование	107
Интернет-ресурсы по CSS	109
Глава 7. Добавление изображений	111
Где найти бесплатные изображения	111
Обращайте внимание на авторское право	111
Создание собственных изображений	112
Форматы графических изображений	113
Поиск изображений	114
Высококачественные изображения за небольшую плату	115
Создание логотипов на сайте CoolText.com	115
Редактирование изображений	119
Изменение размера изображения	119
Обрезка	121
Поворот и отражение	122
Настройка цветов	123
Работа с фильтрами специальных эффектов	125
Программа FastStone ImageViewer: пакетное изменение фотографий	126
Как обработать сотни фотографий за считанные минуты	126
Немного практики	127
Использование изображений в дизайне сайта	130
Размещение изображений для повышения эффективности сайта	131
Изменение размера изображения в HTML	133
Интернет-ресурсы с высококачественными графическими изображениями	134
Глава 8. Добавление форм	135
Элемент INPUT	136
Ввод данных в текстовые поля	138
Текстовые области	140
Переключатели и флажки	141
Списки: элементы SELECT и OPTION	142
Отправка запроса	144
Обработка формы	145
Почему провайдер не хочет помочь	145
Поиск CGI-провайдера	146
CGI-сценарий, хранящиеся на удаленном сервере	147

Добавление CGI-сценариев	148
Основные действия	148
Возможные проблемы	149
Генераторы форм и системы обработки данных	150
Сайт Response-O-Matic	151
Сайт FormSite.com	157
Интернет-ресурсы с информацией о CGI и формах	161
Часть III. Сделайте свой сайт особенным	163
Глава 9. JavaScript и DHTML	165
Принципы программирования	165
Константы и переменные	166
Немного математики	168
Ветвления	169
Циклы	170
Использование возможностей JavaScript	170
Реализация возможностей JavaScript	171
Основные методы	172
Проверка наличия ошибок в сценариях	173
Объектная модель документа (модель DOM)	174
Выбор событий в JavaScript	175
Наведение указателя мыши и щелчки	176
Ввод с клавиатуры	177
Добавление интерактивных меню	177
Создание раскрывающихся списков	177
Создание меню с помощью библиотеки jQuery	178
Интернет-ресурсы по JavaScript и DHTML	182
Глава 10. Добавление красивых кнопок и изображений-карт	183
Создание кнопок	184
Создание изображений-карт	187
Интернет-ресурсы по кнопкам и изображениям-картам	192
Глава 11. Медиакомпоненты для сайта	195
Поиск аудио- и видеокomпонентов	196
Поиск музыкальных коллекций	196
Поиск свободно распространяемых звукозаписей	197
Выбор формата файла	198
Сжатие файлов	201
Добавление аудио- и видеофайлов на сайт	203
Добавление на сайт Flash-плеера	204
Воспроизведение видео и музыки средствами HTML 5:	
теги <AUDIO> и <VIDEO>	205
Импорт видео с YouTube	208
Программы для работы с аудио и видео	209

Глава 12. Добавление блога	211
Создание блога при помощи службы Blogger	211
Блог на основе WordPress	216
Добавление записи в WordPress	218
Изменение общих параметров настройки	220
Интернет-ресурсы по блогам	221
Часть IV. Попробуем разбогатеть	223
Глава 13. Продвижение сайта	225
Добавление ключевых слов на веб-страницы сайта	227
Добавление HTML-тегов <META>	227
Добавление ключевых слов в основной текст веб-страницы	229
Не пытайтесь обмануть поискового робота	229
Анализ ключевых слов, используемых другими сайтами	231
Контекстная реклама	234
Регистрация сайта в поисковых системах	235
Самостоятельная регистрация	235
Продвижение сайта средствами интернет-маркетинга	237
Как скрыть веб-страницу от поискового робота	237
Позиция сайта в поисковых системах	239
Обмен ссылками	240
Поиск партнеров по ссылкам	240
Веб-кольца	241
Обмен баннерами	241
Интернет-ресурсы по продвижению сайтов	242
Глава 14. В погоне за прибылью: электронная коммерция	243
Секреты интернет-маркетинга	244
Правильное отношение к делу	244
Выбор сегмента рынка	245
Деловые контакты предпринимателя	247
Дизайн интернет-магазина	249
Добавление функции поиска	250
Система пользовательского поиска Google	250
Поисковая служба FreeFind	254
Поиск с помощью CGI-сценариев на языке Perl	260
Добавление страницы FAQ	263
Интернет-ресурсы по электронной коммерции	263
Глава 15. Оплата товаров и услуг через Интернет	265
Краткий обзор способов оплаты товара через Интернет	265
Платежная система PayPal	266
Альтернативные варианты регистрации заказов и их оплаты	266
Кредитные карты и банковские услуги	267
Выбор эквайера	268
Выбор пластиковых карт	268
Открытие торгового счета	269

Банковские сборы за предоставление услуг эквайринга	270
Мошенничество с пластиковыми картами	271
Конвертация валют	272
Программные модули для электронных магазинов	275
Часть V. Великолепные десятки	277
Глава 16. Десять полезных ресурсов	279
Информационные ресурсы и форумы	279
K-eCommerce	279
Oborot.ru	279
eComspace	281
PromoForum	281
HTML-форум	282
SITEM.RU	283
Правовые вопросы и интеллектуальная собственность	284
Юридические консультации	284
Интеллектуальная собственность	286
Глава 17. Десять замечательных инструментов для электронной коммерции	289
ECOMMERCELAW.RU	289
OSP.RU	290
LESSONS-TVA.INFO	290
ArwShop	292
INSALES.RU	293
PROM.UA	294
Art map	295
Электронные карты от издательства “Карта”	296
SSL-сертификаты	297
WebMoney	298
Глава 18. Десять других интересных веб-сайтов	299
Google Analytics	299
Сервис WhoIs на nic.ru	300
NameBoy	300
PromoSite.ru: защита от роботов с помощью нечетких изображений	301
XML-Sitemaps	303
MailList	304
Часы для сайта или блога	305
Информер “Часовые пояса”	306
CALEND.RU	306
Фотоархив на сайте autotravel.ru	306
Приложение. Словарь терминов	309
Предметный указатель	323

Создание веб-страницы с помощью базовых HTML-элементов

В этой главе...

- Основные концепции HTML
- Работа с текстом
- Работа с цветом
- Создание ссылок
- Работа с изображениями
- А теперь собираем страницу
- Интернет-ресурсы по разработке веб-страниц

Эта глава призвана освежить в памяти читателей основные моменты, относящиеся к разработке веб-страниц. Это просто необходимо сделать прежде, чем погрузиться в изучение дополнительных возможностей, модулей и программ, о которых рассказывается в остальной части книги. Здесь мы коснемся вопросов создания веб-страниц на языке HTML, добавления на эти страницы текста и графики. Также будет рассказано о том, как форматировать текст, задавать цвет веб-страницы, создавать текстовые и графические ссылки для перехода между страницами.

Тем читателям, которые хорошо владеют этим материалом, будет полезно бегло просмотреть главу и обратить внимание на веб-сайты, адреса которых приведены в таблицах.

Основные концепции HTML

В основном веб-страницы создаются с помощью языка гипертекстовой разметки (HyperText Markup Language — HTML). HTML — простой язык разметки текстовых страниц, основная задача которого заключается в форматировании страницы при отображении в веб-браузере, например в Firefox или Microsoft Internet Explorer. Создаваемые вами HTML-файлы будут содержать простой (неформатированный) текст, поэтому при работе над ними нет необходимости применять что-либо более сложное, чем стандартное Windows-приложение Блокнот или другой простой текстовый редактор.



Редактор этой книги рекомендует использовать программу Notepad2, поскольку она поддерживает подсветку синтаксиса языка HTML (и других языков разметки или программирования) и позволяет работать с файлами большого размера. Впрочем, подробный разговор о выборе программ еще предстоит в главе 4.

В данной главе представлен минимум информации, который совершенно необходимо знать для создания веб-страниц и их объединения в работающий сайт. Если у вас есть желание изучить язык HTML более досконально, прочитайте книгу *HTML 5 и CSS 3 для чайников*.



Спецификацию по языку HTML 5 можно найти по таким адресам: <http://htmlbook.ru/html5> и <http://www.w3.org/TR/html5/>. Если окажется так, что вам будет доступна книга по HTML 4, то такой вариант тоже вполне подходит. Прочтите ее, а затем просто ознакомьтесь с новшествами в версии HTML 5 по приведенным здесь адресам. На самом деле различия между HTML 4 и HTML 5 для начинающих пользователей совсем невелики, поэтому вы можете смело обратиться к любой книге по HTML 4, а после ее прочтения просто ознакомиться с тем, что есть нового в версии HTML 5.

Документ на языке HTML состоит из *элементов*. Например, абзац текста или изображение — это элементы. Элементы, в свою очередь, состоят из дескрипторов (тегов), их атрибутов и содержимого (последнее не всегда обязательно). Ниже приведена более подробная информация по каждому упомянутому выше виду элементов.

- ✓ **Теги.** Это простые текстовые дескрипторы (определители), которые сообщают веб-браузеру информацию относительно типа и расположения каждого элемента на странице, а также требуемого их форматирования.
 - *Открывающий тег.* Начало каждого элемента в HTML обозначается открывающим тегом, при этом тип данного элемента указывается текстом в угловых скобках. Например, открывающий тег абзаца текста — это `<P>`, а открывающий тег изображения — `<IMG>`.
 - *Закрывающий тег.* Конец элемента в языке HTML обозначается закрывающим тегом. Он точно такой же, как открывающий, только перед именем элемента ставится обратная косая черта. Следовательно, `</P>` — закрывающий тег абзаца. У некоторых элементов, например, у `IMG`, закрывающих тегов нет.
- ✓ **Атрибуты.** С помощью атрибутов определяют параметры элементов. Например, добавляя соответствующие атрибуты к тегу `IMG`, задают высоту и ширину определяемого им изображения, как в следующем фрагменте программного кода:
`<IMG width="100" height="30">`
- ✓ **Содержимое.** Само содержимое тега (если оно имеется) всегда располагается между открывающим и закрывающим тегами элемента:
`<P>Здесь расположено содержимое абзаца.</P>`

Далее рассказывается о наиболее часто используемых тегах и атрибутах. Изложенные выше основные принципы языка HTML помогут вам разобраться в вариантах выбора, которые предлагаются различными программными средствами разработки веб-страниц.

Базовая структура типовой веб-страницы содержит три элемента, которые определяются тегами HTML, HEAD и BODY. При этом элементы HEAD и BODY входят в состав элемента-контейнера HTML.

```
<HTML>

<HEAD>
</HEAD>

<BODY>
</BODY>

</HTML>
```



На самом деле технически можно реализовать веб-страницу и без использования тегов HTML, HEAD и BODY (стандарт HTML позволяет это сделать), однако удачным решением это не назовешь. Исключив эти теги из документа, разработчик ровным счетом ничего не добьется, а их использование позволяет правильно упорядочить остальные элементы страницы.

Все видимые на веб-странице элементы расположены внутри элемента BODY, представляющего ее основное содержимое (или *тело*). Элемент HEAD содержит некоторую общую заглавную информацию, в том числе и название страницы, которое записывается между тегами <TITLE> и </TITLE>.

```
<HEAD>
<TITLE>Название страницы</TITLE>
</HEAD>
```



Учтите, что название веб-страницы будет отображаться не на ней самой, а в заголовке окна браузера пользователя.

Ой, что здесь написано?

В окне браузера вместо текста вы видите какие-то каракули? Это означает, что браузеру не удалось правильно определить кодировку текста отображаемой им веб-страницы. Это промах разработчика, поскольку с помощью специальных

тегов всегда можно точно указать браузеру, какая именно таблица кодировки использовалась для представления текста на данной веб-странице. Далее в этой книге будет рассказано, как это следует делать.

Работа с текстом

Текст — основной способ передачи информации в Интернете. Очень редко встречаются веб-страницы без текста.

Приведенные в следующих разделах примеры программного кода показывают, как создавать различные элементы (содержащие слова), о которых пойдет речь.

Абзацы

Абзацы — это элементы, используемые на веб-странице для размещения текста. Текст располагают между тегами `<P>` и `</P>`.

```
<P>Здесь расположено текстовое содержимое.</P>
```

Наличие закрывающего тега `</P>` в программном коде совершенно не обязательно. Просто большинство программ для создания веб-страниц добавляют его в код автоматически.

Интервал между абзацами создается веб-браузером тоже автоматически. Если это вас почему-либо не устраивает, для увеличения интервала можно воспользоваться элементом `BR`.

```
<P>Это первый абзац.</P>
```

```
<BR>
```

```
<P>Обратите внимание на интервал между первым и вторым абзацами.</P>
```

На рис. 2.1 показано, что получается в результате помещения приведенного выше фрагмента кода на веб-странице.

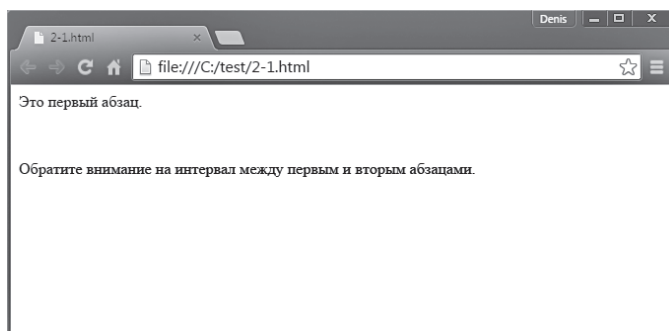


Рис. 2.1. На веб-странице увеличен интервал между абзацами



Новичку покажется очевидным, что для получения увеличенного интервала между двумя абзацами вместо элемента `BR` можно вставить “пустой” элемент `P` (без какого-либо содержимого между его открывающим и закрывающим тегами). К сожалению, такой прием работает не для всех веб-браузеров. Стандарт языка HTML не регламентирует применение пустых значений элемента `P`. Поэтому те браузеры, которые строго придерживаются стандарта, просто игнорируют такие элементы, и в результате пустая строка на экран не выводится.

Для стандартного решения этой проблемы в “пустой” элемент `P` следует поместить невидимое содержимое. Так как обычный пробел браузеры, скорее всего, также проигнорируют, между открывающим и закрывающим тегами `<P>` “пустого” элемента следует поместить специальный код ` `; — так называемый “неразрывный пробел”.

```
<P>Содержимое первого абзаца.</P>
```

```
<P>&nbsp;</P>
```

```
<P>Перед этим абзацем будет добавлена пустая строка.</P>
```



При использовании неразрывного пробела не забудьте поставить в конце его кода точку с запятой (;). Если этого не сделать, вместо пустой строки на экране будет просто отображаться текст ` `.

Заголовки

Заголовки также являются текстовыми элементами. Заголовки разных уровней отличаются размером шрифта (обычно полужирного). В языке допускается использовать полдюжины различных вариантов заголовков, начиная от огромного заголовка типа `H1` и заканчивая крошечным заголовком типа `H6`. Все остальные значения — `H2`, `H3`, `H4` и `H5` — промежуточные. Размер шрифта уменьшается по мере возрастания номера заголовка.

На веб-страницах заголовки применяются для визуального отделения одного раздела от другого. Заголовки меньшего уровня соответствуют подразделам более крупных разделов документа. Допустим, разрабатывается веб-сайт с новостями. Тогда в качестве основного заголовка для каждой статьи можно использовать элемент `H1`, за которым будет следовать текст сообщения, помещенный в элементы `P`. Для форматирования подзаголовков в пределах статьи перед каждым разделом будет вставлен элемент `H2`, а для подзаголовков следующего уровня внутри разделов — элемент `H3` и т.д. Пример HTML-кода с подобной структурой приведен ниже, а как этот код будет выглядеть в окне браузера, представлено на рис. 2.2.

```
<H1>Это заголовок первого уровня</H1>
```

```
<P>Далее можно разместить несколько абзацев текста.
```

```
В данном случае это один абзац.</P>
```

```
<H2>В этом примере для форматирования подзаголовка используется  
элемент H2</H2>
```

```
<P>А здесь можно разместить текст подраздела.
```

```
Сделайте несколько абзацев с помощью элемента P.</P>
```

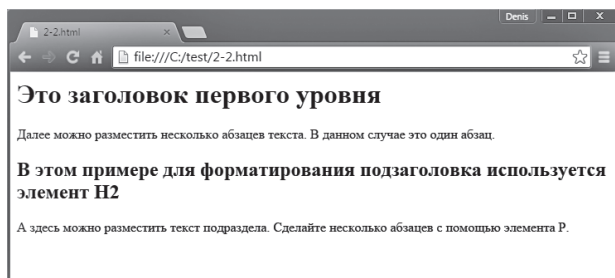


Рис. 2.2. Заголовки позволяют упорядочить текст на веб-странице

Шрифты

Шрифт определяет начертание символов. Шрифт **этого** слова отличается от **этого**. Можно взять шрифты, используемые по умолчанию в браузере, или задать какие-то свои. Для новичков вполне подойдут стандартные шрифты, однако по мере роста мастерства вполне естественно желание разработчика применять для оформления страниц специальные шрифты, которые будут полнее соответствовать его замыслу.

Два наиболее часто используемых элемента форматирования шрифтов — это **B** и **I**. Они позволяют задать **жирное** и *курсивное* начертание соответственно.

```
<P>Это обычное начертание.  
<B>Это -- пример жирного начертания.</B>  
<I>Пример курсивного начертания.</I></P>
```

Атрибут `face` тега `<FONT>` позволяет установить гарнитуру шрифта, например, Arial, Times New Roman или какую-либо другую. А с помощью атрибутов `size` и `color` можно задать соответственно размер и цвет шрифта. Более подробно о задании цвета шрифта рассказывается в разделе “Работа с цветом”. Следующий фрагмент кода показывает, как с помощью тега `<FONT>` установить гарнитуру, размер и цвет шрифта для отдельного абзаца.

```
<P><FONT face="Arial, Helvetica, sans-serif" size="5"  
  color="blue">Этот текст набран шрифтом Helvetica  
  синего цвета размером 5.</FONT>  
<FONT face="Times New Roman, Times, serif" size="3"  
  color="red"> Этот текст набран шрифтом Times New Roman  
  красного цвета размером 3.</FONT>  
<FONT face="Courier New, Courier, mono" size="7"  
  color="black"> Этот текст набран шрифтом Courier  
  черного цвета 7.</FONT></P>
```

На рис. 2.3 представлены результаты выполнения двух предыдущих фрагментов кода.

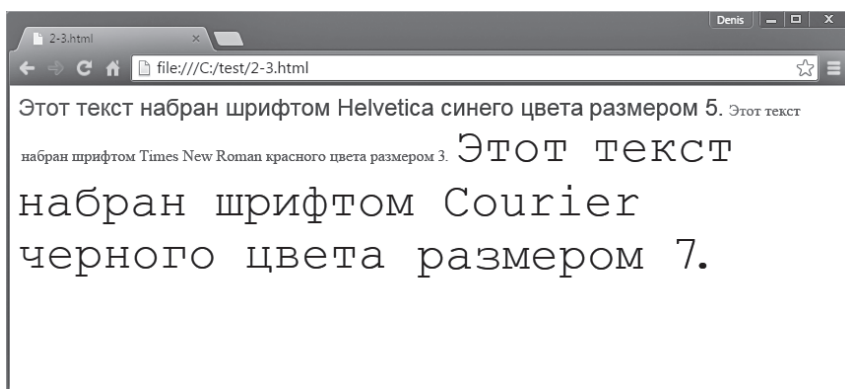


Рис. 2.3. Элемент `FONT` позволяет отформатировать текст разными шрифтами



В данном примере элемент `FONT` применялся к целому предложению, хотя его можно использовать и для более коротких структур — даже для одного символа.

Причина, по которой атрибуту `face` в приведенном выше примере каждый раз присвоено сразу по нескольку значений (через запятые), очень проста. Все дело в том, что пользователи в Интернете работают на компьютерах с разными операционными системами (Windows, Mac, Unix и других), поэтому далеко не каждый точно указанный разработчиком шрифт может быть найден браузером в той или иной системе. В результате браузер вынужден будет подбирать другой шрифт для замены. Предлагая набор допустимых вариантов, вы повышаете вероятность того, что в браузере все-таки окажется хотя бы какой-то подходящий случаю шрифт. Предыдущий фрагмент кода в трех предложениях включает указание на три разных типа гарнитур (шрифтов), наиболее широко используемых в Интернете.

Забегая наперед, хочу подсказать вам, что изменить шрифт можно двумя способами: как с помощью тега `<FONT>`, так и с помощью CSS-стилей. Последний стандарт языка HTML рекомендует в новых разработках не использовать тег `<FONT>`, а вместо него использовать CSS-стили. Пока вы новичок, смело используйте тег `<FONT>`, если это покажется вам проще, но позже, чтобы не ударить в грязь лицом перед другими веб-дизайнерами, настоятельно рекомендую использовать исключительно стили CSS.

Линии

Конечно, горизонтальные линии к тексту не имеют никакого отношения, но именно в этой части книги рассказ о них кажется мне наиболее уместным. В языке HTML такие линии называются *горизонтальными разделительными линиями* (horizontal rules), а элемент, который их представляет, — `HR`. Горизонтальные линии позволяют визуальным образом отделить один раздел на странице от другого, подчеркнуть изображение и т.д.

Длину горизонтальной линии можно задать двумя способами: процентным соотношением от ширины экрана либо конкретным значением в пикселях. По умолчанию длина линии равна 100% ширины экрана, поэтому, если не задать значение атрибута `width`, разделитель будет создан на всю ширину экрана монитора. В следующем фрагменте кода определена 50-процентная длина разделителя:

```
<HR width="50%">
```

В свою очередь, чтобы задать длину горизонтальной линии в 400 пикселей, нужно использовать следующий код:

```
<HR width="400">
```

Для установки *толщины* линии используется атрибут `size` (задаваемый в пикселях):

```
<HR size="6">.
```

По умолчанию линия не заполнена цветом. Линия без заливки называется *линией с тенью*, потому что раньше, когда веб-страницы были не такими красочными, как сейчас, текст черного цвета размещался на нейтральном сером фоне, а “пустые” горизонтальные линии создавали трехмерный эффект, или тень. Эффект тени, в отличие от цвета фона, виден не отчетливо. Чтобы сделать линию сплошной, в тег `<HR>` нужно дополнительно добавить атрибут `noshade`:

```
<HR noshade>
```

На рис. 2.4 представлена веб-страница, которая получается на основе следующего фрагмента кода:

```
<HR>
<HR width="100%">
<HR width="50%">
<HR width="200">
<HR width="400">
<HR size="10">
<HR size="6" noshade>
```

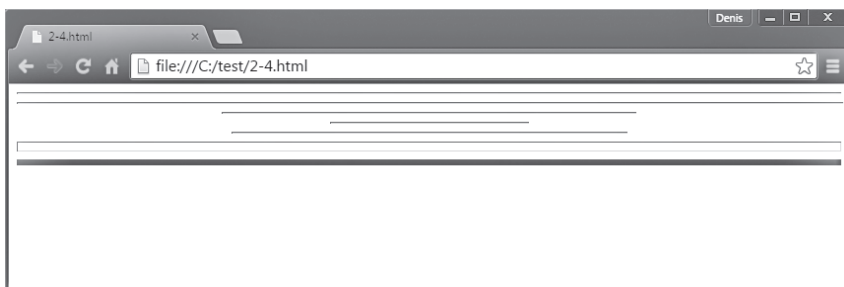


Рис. 2.4. Форматирование горизонтальных разделительных линий меняется в зависимости от значений их длины и толщины



Вместо рисования линий на странице с помощью HTML-тегов многие дизайнеры используют графику, например GIF-файлы. В качестве примеров можно посмотреть электронные ресурсы графических заготовок, адреса которых приведены в разделе “Работа с изображениями”.

Работа с цветом

Сегодня уже достаточно сложно найти черно-белые изображения, разве что старые телепередачи и фильмы или профессиональные фотографии, претендующие на художественную ценность. Мир состоит из ярких красок, и посетитель сайта будет разочарован, не увидев их.

Тема цветового оформления шрифтов затрагивалась вкратце в предыдущем разделе, однако различные оттенки цвета могут использоваться не только для шрифтов. Несомненно, со временем вы сможете научиться задавать цвет для любого HTML-элемента на веб-странице.



Каскадные таблицы стилей (CSS) предоставляют гораздо больше возможностей управления цветовым оформлением, чем теги языка HTML. Краткий обзор возможностей технологии CSS приведен в главе 6.

В разделе “Шрифты” рассказывалось о том, как определить цвет некоторого набора символов. Однако задать цвет можно и для всего текста, и для фона страницы, и для ссылок. При этом для ссылок устанавливают *три* разных цвета: для не просмотренной, для активной и для уже просмотренной.

Все эти цветовые настройки можно выполнить с помощью элемента BODY, задав в нем требуемые значения следующих атрибутов:

- ✓ text — цвет текста;
- ✓ bgcolor — цвет фона документа;
- ✓ link — цвет не просмотренной ссылки;
- ✓ alink — цвет активной ссылки (ссылки, на которой щелкнул пользователь в данный момент);
- ✓ vlink — цвет уже просмотренной ссылки.

Задание всех перечисленных атрибутов одновременно выглядит примерно так:

```
<BODY text="black" bgcolor="white" link="blue" vlink="red"
alink="purple">
```

Создание ссылок

В Интернете *ссылки* играют очень важную роль, так как именно они связывают между собой разные веб-страницы. Без них существование Интернета немислимо. В языке HTML ссылки создаются с помощью элемента A (от англ. “anchor” — привязка). Атрибут href (от англ. “hypertext reference” — гипертекстовая ссылка) данного элемента определяет веб-адрес файла, на который разработчик создает ссылку. Этот адрес называется “URL” (сокращение от англ. “uniform resource locator” — унифицированный указатель ресурса). Вот как выглядит ссылка в HTML-документе:

```
<A href="http://www.dialektika.com/">содержимое</A>
```

После щелчка мышью на ссылке (это может быть текст или графический объект — то, что в этом элементе представлено параметром *содержимое*) браузер загружает документ, адрес которого присвоен атрибуту href. Если ссылка текстовая, по умолчанию она отображается шрифтом синего цвета с подчеркиванием, а если графическая — вокруг изображения создается голубая рамка.

Работа с изображениями

Если на веб-странице будет только текст, большинство пользователей сочтут этот сайт скучным. Довольно редко встречаются сайты без изображений. При размещении

изображений в Интернете обычно используются три формата файлов: GIF, JPEG (также называемый JPG) и PNG. О форматах файлов более подробно рассказывается в главе 7.

Где брать изображения? Их можно создавать с “чистого листа”, а можно загрузить уже готовые файлы с одного из сайтов, которые перечислены в табл. 2.1. При использовании чужих изображений, а так делают большинство дизайнеров, необходимо сначала ознакомиться с особыми условиями договора (обычно их набирают мелким шрифтом), предложенного на сайте. Возможен и другой вариант, когда изображения защищены авторским правом. Тогда их нельзя использовать просто так, без разрешения. К счастью, большинство владельцев изображений в Интернете с удовольствием дадут такое разрешение, попросив взамен лишь создать ссылку на свой сайт. Такое соглашение обеспечивает рост популярности сайта, с которого разработчик заимствует изображение, а разработчику, в свою очередь, оно дает бесплатный высококачественный материал для создания интересного сайта с графикой. Все остаются довольны.

Таблица 2.1. Ресурсы по работе с изображениями

Название веб-сайта	Веб-адрес
Библиотека изображений	www.fotobank.com
Фотобанк изображений премиум-качества	http://ru.depositphotos.com/
Коллекция анимированных изображений в формате GIF	www.penguin00.chat.ru/gifs.htm
Сайт современного художника Дениса Мезенцева	http://www.surrealism.ru/index.html
Изображения по различным темам	www.wallpaper.onru.ru
Тематические изображения и анимация	www.oboi.kards.ru

Также для поиска изображений можно использовать сервис Картинки Google (images.google.com). Просто введите поисковый запрос, а Google отобразит картинки, которые соответствуют ему.



Бывает, что собственник графики, защищенной авторским правом, не соглашается предоставить ее разработчику просто в обмен на ссылку. Для использования такого изображения разработчик должен заплатить определенную сумму. Особые условия договора могут предусматривать бесплатное использование графики для некоммерческого сайта, тогда как для коммерческого такую услугу придется оплатить. Вот почему столь важно ознакомиться с этим документом. Ни в коем случае нельзя просто взять и присвоить изображение. Это может быть чревато нежелательными последствиями. Конечно, можно загрузить любое изображение, которое отображается в веб-браузере, однако кража графики может плохо обернуться для разработчика (даже уголовной ответственностью). Но в Интернете доступно много ресурсов бесплатной графики (см. табл. 2.1) и взяв изображение оттуда, можете смело его использовать.

Изображения

Наиболее распространенные элементы на веб-страницах, кроме текста, — изображения. Для их вставки на страницу предназначен тег `<IMG>`. Его обязательный атрибут — `src`. Он указывает имя графического файла, а при необходимости и путь к нему (если этот файл расположен не в той же папке, что и HTML-файл данной страницы). Таким образом, в простейшем случае для размещения изображения на веб-странице (рис. 2.5) достаточно будет следующего кода:

```
<IMG src="seated16.jpg">
```

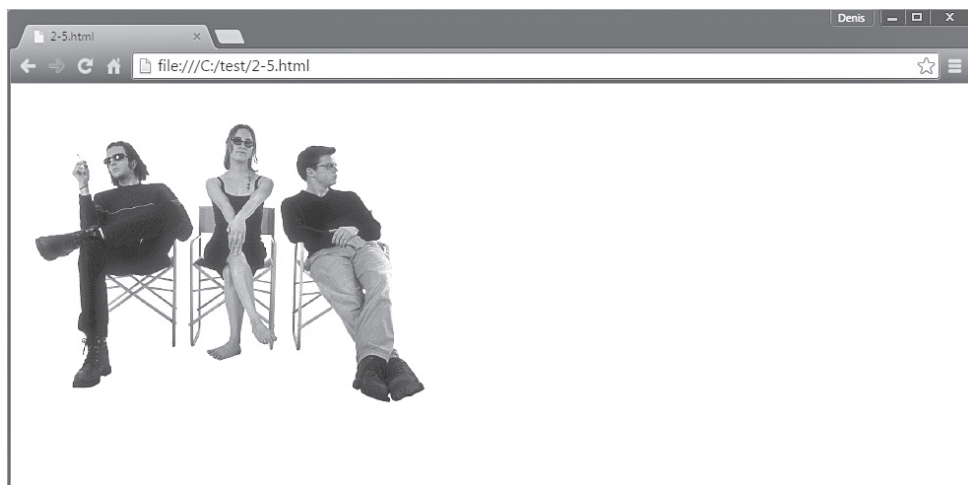


Рис. 2.5. Изображение скучающих людей в формате JPEG

Если графический файл расположен в другой папке, нужно указать, где именно, как в следующем фрагменте программного кода (здесь `pets` — имя другой папки):

```
<IMG src="pets/seated16.jpg">
```

Если графический файл размещен вообще на другом сервере, следует указать полный путь к нему (URL):

```
<IMG src="http://www.имя_сайта.com/seated16.jpg">
```

Выше в разделе “Создание ссылок” упоминалось о том, что графику можно использовать в качестве ссылок, точно так же, как и текст. Для этого достаточно поместить тег `<IMG>` между открывающим и закрывающим тегами `<A>` ссылки:

```
<A href="http://www.dummies.com/"><IMG src="seated16.jpg"></A>
```



У тега `<IMG>` нет закрывающего тега.

Фоновые рисунки

В отличие от обычной графики, размещаемой в различных местах веб-страницы с помощью тега `<IMG>`, *фоновые* рисунки создаются с помощью атрибута `background` тега `<BODY>`, при этом тег `<IMG>` не используется:

```
<BODY background="guitar2.jpg">
```

Обнаружив такое указание, браузер выполняет *мозаичное* заполнение фона страницы, при этом данное изображение повторяется на экране до тех пор, пока не будут достигнуты края экрана как в горизонтальном, так и в вертикальном направлении. Благодаря такому подходу большинство разработчиков предпочитают использовать небольшие фоновые изображения, как, например, на рис. 2.6, где для заполнения фона страницы используется миниатюрное фото гитары.

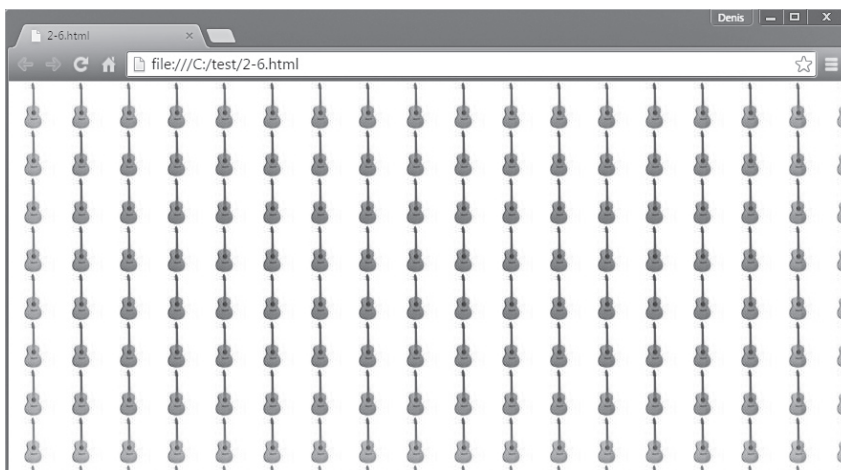


Рис. 2.6. Страница с мозаичным заполнением фона

Фоновые рисунки надо подбирать очень тщательно, чтобы потом не оказалось, что они плохо сочетаются с остальными элементами на веб-странице. Для таких целей предпочтительнее графика с приглушенными тонами и нечеткими линиями. К тому же совершенно необходимо всегда подбирать такую графику, которая подходит сайту по его тематике.

А теперь собираем страницу

Итак, мы уже знакомы с частью элементов, обычно присутствующих на веб-страницах. Пришло время собрать из них страницу. Конечно, вы, возможно, никогда не воспользуетесь одновременно каждой из возможностей, поскольку некоторые их комбинации приводят к конфликтам при проектировании (например, если будут одновременно заданы и цвет фона, и фоновое изображение, цвет фона будет перекрыт фоновым изображением), но данный пример показывает, как рассмотренные выше HTML-элементы следует собирать на веб-странице.

```

<HTML>
<HEAD>
<TITLE>Здесь указывается название страницы.</TITLE>
</HEAD>

<BODY text="black" bgcolor="white" link="blue" vlink="red"
      alink="purple"><H1>Заголовок первого уровня</H1>

<HR width="50%">
  <P>Первый абзац,<I>часть текста которого</I> выделена
    курсивом.</P>
<IMG src="TODe11.jpg">

<H2>Заголовок второго уровня</H2>
  <P>Второй абзац.</P>
<P>Щелкните <A href="http://www.dummies.com/">здесь </A>для
  получения дополнительной информации

</BODY>

</HTML>

```

Интернет-ресурсы по разработке веб-страниц

В табл. 2.2 приведен перечень интернет-ресурсов, на которых можно найти более подробную информацию по рассмотренным темам.

Таблица 2.2. Ресурсы по разработке веб-страниц

Название веб-сайта	Веб-адрес
Учебник по HTML 5	http://htmlbook.ru/html5
Современные технологии разработки сайтов с примерами программного кода	www.weblibrary.biz