

8 Добавление графики на веб-страницы

Как бы вы ни украшали текст веб-страниц с помощью границ и полей, ничто так не влияет на внешний вид сайта, как изображения. Каскадные таблицы стилей предлагают исчерпывающий набор средств управления ими. В веб-дизайне можно работать с графическими элементами двумя способами: посредством элемента `img` и свойства `background-image` (которое позволяет применить изображение к фону любого тега на странице).

В этой главе углубленно рассмотрены некоторые оригинальные методы применения изображений средствами языка CSS. На самом деле лучший способ узнать о возможностях графики и научиться ее использовать — увидеть результаты практических примеров, поэтому в конце главы в практикуме приведены три урока. В них мы создадим веб-страницу фотогалереи и попрактикуемся в быстрой и профессиональной разработке дизайна с применением изображений.

Каскадные таблицы стилей и элемент `img`

Элемент `img` представляет собой средство добавления изображений в веб-страницы фотогалерей еще со времен появления Всемирной паутины. Даже сайты без фотографий используют его для добавления логотипов, кнопок навигации и иллюстраций. В CSS нет свойств, специально предназначенных для форматирования изображений. Однако вы можете использовать для этого общие свойства, с которыми познакомились в предыдущих главах книги. Например, свойство `border` обеспечивает простой и быстрый способ заключить изображение в рамку или унифицировать вид галереи. Ниже представлен список свойств, наиболее часто используемых в отношении изображений.

- `border` — можно указать любое из множества свойств границ (см. раздел «Добавление границ» главы 7) для обрамления изображений. Пример приведен в практикуме этой главы. Поскольку каждая сторона изображения может иметь свои параметры границ: цвет, стиль, толщину линии — ваши творческие возможности расширяются.
- `padding` — добавляет пустой промежуток между границей и изображением. Отделение рамки от фотографии небольшим промежутком имитирует подложку, которая традиционно используется для обрамления рисунков с целью отделе-

ния их друг от друга. Устанавливая цвет фона, вы даже можете изменить цвет самой подложки.

- `float` — выравнивание изображения с помощью свойства `float` перемещает его к левому или правому краю веб-страницы. Или, если изображение расположено внутри другого блочного элемента, например боковой панели, — к левой или правой стороне этого элемента. Таким образом, текст и другие элементы веб-страницы обтекают изображение с другой свободной стороны и отображаются под ним. Вы можете создать многократное обтекание рисунков для отображения фотогалереи в несколько строк и столбцов.
- `margin` — чтобы добавить пустой промежуток между изображением и остальным содержимым веб-страницы, используйте свойство `margin`. Когда вы устанавливаете для изображения обтекание, текст обычно располагается слишком близко. Добавление левого поля (с выравниванием по правому краю) или правого поля (с выравниванием по левому краю) создает пустой отделяющий промежуток между текстом и графическим элементом.
- `border-radius` — свойство `border-radius` позволяет делать углы элементов, включая изображения, закругленными. С его помощью можно «закруглить» углы изначально прямоугольной фотографии с прямыми углами.

В большинстве случаев вам не потребуется создавать стиль для самого элемента `img`. Его форматирование затрагивает слишком большой диапазон элементов веб-страницы и изменит *все* изображения, включая элементы, обеспечивающие совершенно другие функции. Так, наряду с изображениями одинаковое форматирование будут иметь логотип, навигационные кнопки, фотографии и даже рекламные баннеры. Наверное, вы не хотели бы видеть одну и ту же черную рамку, обрамляющую все изображения. Вместо этого следует использовать классы, например `.galleryImage` или `.logo`, для применения стилей к нужным элементам выборочно.

Другой подход заключается в использовании селекторов потомков для целевого форматирования изображений, сгруппированных вместе в одном фрагменте веб-страницы. Если у вас есть галерея фотографий, можно заключить весь контент внутрь элемента `div` с классом `.gallery` и затем создать стиль только для изображений, расположенных внутри этого блока: `.gallery img`.

СОВЕТ

Присвойте значение 50% свойству `border-radius`, чтобы сделать из прямоугольного изображения круглое. Например, вы можете создать класс `round` и применить его к рисунку, который должен отображаться в виде круга:

```
.round { border-radius: 50%; }
```

Добавление фоновых изображений

Свойство `background-image` — ключ к созданию визуально потрясающих сайтов. Стоит только научиться применять его и родственные свойства, и вы сможете заставить сайт выделяться среди прочих. Чтобы убедиться в эффективности фоновых изображений, перейдите по адресу csszengarden.com. HTML-код обеих веб-

страниц, показанных на рис. 8.1, совершенно одинаков; наибольшие визуальные различия достигаются благодаря использованию фоновых изображений (на самом деле, если вы взглянете на HTML-код этих веб-страниц, вы увидите, что в них нет ни единого элемента `img`).

Если вы уже создавали сайты раньше, то, наверное, использовали изображения в качестве фоновых рисунков веб-страниц. Скорее всего, они были небольших размеров, повторяющиеся на заднем плане окна браузера с едва различимым узором. Этот проверенный временем метод языка HTML использовал атрибут `background` элемента `body`. Каскадные таблицы стилей сделают все то же самое, но гораздо лучше.

ПРИМЕЧАНИЕ

Далее в книге вы найдете описание трех свойств фоновых изображений и изучите CSS-код каждого из них. Позже в этой главе вы познакомитесь с сокращенным методом набора — вариантами свойств, которые сэкономят немало времени.

Свойство `background-image` добавляет графический файл в качестве фона элемента. Чтобы поместить его на заднем плане всей веб-страницы, можно создать стиль для элемента `body`:

```
body {  
  background-image: url(images/bg.png);  
}
```

Свойство принимает единственное значение: ключевое слово `url`, за которым следует путь к графическому файлу, заключенный в круглые скобки. Вы можете использовать абсолютный URL-адрес, например, так: `url:(http://www.cosmofarmer.com/image/bg.gif)` — или относительный путь от документа или корневого каталога сайта:

```
url(../images/bg.gif) /* относительный путь от документа */  
url(/images/bg.gif) /* относительный путь от корневого каталога */
```

Как описывается в следующей врезке «В курс дела!», относительный путь указывает адрес файла таблицы стилей от документа CSS, а *не* формируемой HTML-страницы. Конечно, пути будут совпадать при использовании внутренней таблицы, но вы должны помнить об этом поведении, применяя *внешнюю* таблицу стилей. Предположим, у вас есть папка **styles** (содержащая таблицы стилей сайта) и папка **images** (с изображениями сайта). Обе расположены в главном (корневом) каталоге вместе с начальной (домашней) страницей сайта (рис. 8.2). Когда посетитель просматривает ее, загружается также внешняя таблица стилей (шаг 1 на рис. 8.2). Теперь допустим, что таблица включает стиль `body` с атрибутом фоновой изображения, в котором в качестве рисунка выбран файл `bg.png` из папки **images**. Относительный путь по отношению к документу приведет от таблицы стилей к рисунку (шаг 2 на рис. 8.2). Это будет выглядеть следующим образом:

```
body {  
  background-image: url(../images/bg.png);  
}
```

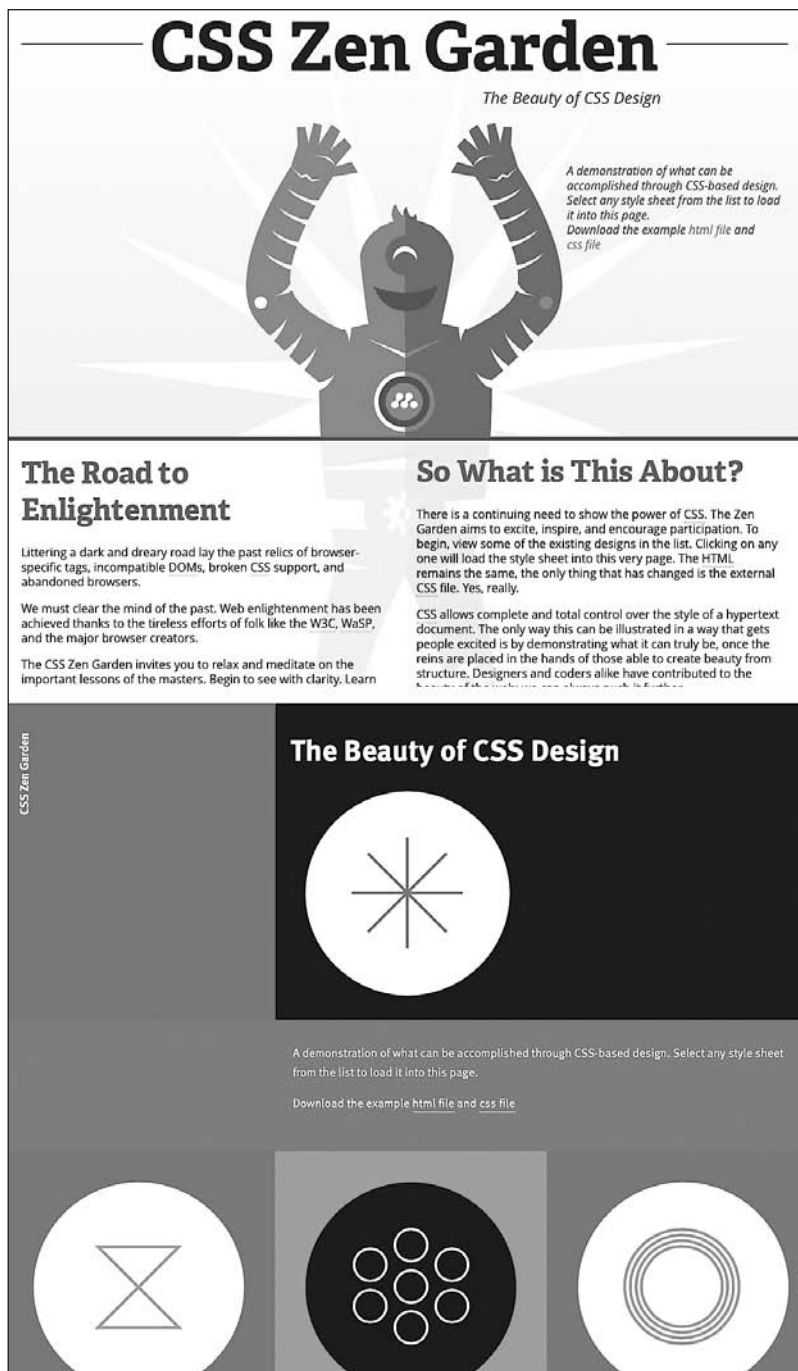


Рис. 8.1. Сайт csszengarden.com демонстрирует мощность каскадных таблиц стилей, показывая, как с помощью языка CSS вы можете превратить один и тот же HTML-файл в две совершенно разные веб-страницы

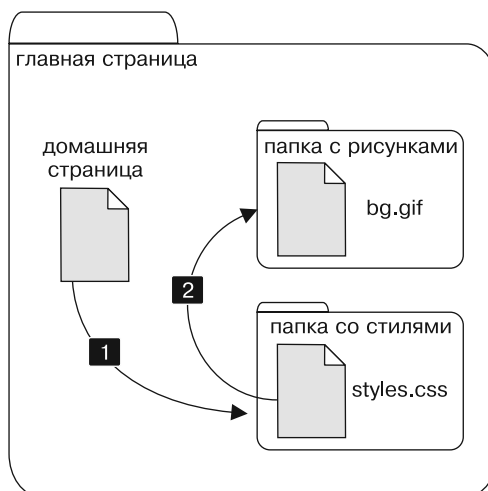


Рис. 8.2. Относительный путь от документа вычисляется применительно к файлу таблицы стилей, но не к формируемой веб-странице

Этот путь интерпретируется так: символы `../` означают «подняться вверх на один уровень», то есть к корневому каталогу, к папке, содержащей папку `styles`; `images/` означает «перейти к папке `images`», а `bg.png` определяет имя файла изображения.

В КУРС ДЕЛА!

GIF-, JPEG- и PNG-файлы: веб-графика

Компьютерная графика представлена сотнями различных форматов файлов с такими замысловатыми аббревиатурами, как JPEG, GIF, TIFF, PICT, BMP, EPS и т. д.

Веб-графика немного проще. Современные браузеры работают только с четырьмя графическими форматами: GIF, JPEG, PNG и SVG. Каждый из них обеспечивает хорошее *сжатие*. Благодаря возможностям компьютера сжатие уменьшает размер графического файла и тот может «путешествовать» через Интернет быстрее. Какой из этих форматов лучше выбрать, зависит от изображения, которое вы хотите добавить на свою страницу.

- **GIF** (Graphics Interchange Format). Такие файлы обеспечивают хорошее сжатие для изображений, в которых есть области со сплошным цветом: логотипы, текст, простые баннеры и т. д. GIF-файлы также предоставляют возможность установить один из цветов прозрачным, а это означает, что

сквозь такую область изображения будет виден фон веб-страницы. Кроме того, GIF-изображения могут включать в себя простую анимацию.

Изображения GIF содержат не более 256 оттенков, вследствие чего фотографии обычно выглядят *постеризованными* (пятнистыми и нереалистичного цвета, как плакат). Другими словами, фото заката, которое вы сделали цифровой камерой, в формате GIF будет выглядеть не очень хорошо. Если вам не нужно анимировать изображение, то формат PNG8, который мы обсудим ниже, станет лучшим выбором по сравнению с GIF.

- **JPEG** (Joint Photographic Experts Group). Такая графика имеет сильные стороны там, где у GIF наблюдаются недостатки. Изображения JPEG могут содержать миллионы различных цветов, что делает их идеальными для фотографий. Однако файлы JPEG имеют преимущества не только в плане

В КУРС ДЕЛА!

фотографий. Они способны сжимать изображения с множеством различных цветов гораздо лучше, чем GIF, потому что алгоритм сжатия JPEG предусматривает то, как человеческий глаз воспринимает смежные значения цветов. Когда графическое приложение сохраняет файл JPEG, происходит комплексный анализ цвета с целью снижения объема данных, необходимых для точного представления образа. С другой стороны, сжатие JPEG приводит к тому, что текст и большие области со сплошным цветом покрываются пятнами (*артефактами*).

- **PNG** (Portable Network Graphics) включает в себя лучшие черты GIF-и JPEG-форматов, но вы должны узнать, какую именно версию PNG использовать в конкретной ситуации. PNG8 в основном заменяет GIF. Как и GIF, он предлагает 256 цветов и базовую возможность сделать один цвет прозрачным. Тем не менее PNG8 обычно *эффективнее* сжимает изображения и делает их размер несколько меньше, чем GIF. По этой причине изображения PNG8 загружаются чуть быстрее, чем такие же изображения, сохраненные в формате GIF. То есть лучше использовать PNG8, а не GIF.

Форматы PNG24 и PNG32 (также известный как PNG24 с альфа-прозрачностью) предлагают расширенную цветовую палитру JPEG-изображений без потери качества. Это означает, что фотографии сохраняются в форматах PNG24 или PNG32 и, как правило, имеют более высокое качество, чем JPEG. Но прежде, чем перейти к формату PNG, следует помнить, что JPEG-изображения предла-

гают очень хорошее качество и меньший размер файла, чем PNG24 либо PNG32. В общем, JPEG станет гораздо лучшим выбором для фотографий и других изображений, которые состоят из множества цветов.

Однако PNG32 имеет одну особенность, которой нет у других форматов: это 256 уровней прозрачности (также называемой *альфа-прозрачностью*). Она позволяет оформить фон веб-страницы как тень или задать для графики прозрачность 50 %, чтобы можно было видеть нижележащий элемент сквозь нее, создавая эффект полупрозрачности.

- **SVG** — еще один графический формат, отличающийся от PNG, GIF и JPEG тем, что не является растровым. Вместо того чтобы размещать маленькие пиксели рядом друг с другом для формирования изображения, формат SVG (или Scalable Vector Graphics — масштабируемая векторная графика) определяет набор математических инструкций, которые описывают изображение на экране. Формат SVG, как и другие векторные форматы, не предназначен для фотографий или изображений. Вместо этого он используется для векторной графики, например логотипов и значков. Чтобы создать файл формата SVG, можно использовать программу Adobe Illustrator или аналогичный ей векторный редактор. Преимущество таких файлов заключается в том, что, как правило, при небольшом размере изображения можно масштабировать, увеличивая его размер без потери резкости и четкости. На сайте tinyurl.com/p7w6zkd приведен длинный список ресурсов, посвященных формату SVG.

В приведенных примерах путь не заключен в кавычки, как требует HTML; они необязательны, хотя и могут указываться. В CSS все три следующие строки кода функционируют идентично:

```
background-image: url(images/bg.png);
background-image: url("images/bg.png");
background-image: url('images/bg.png');
```

ПРИМЕЧАНИЕ

Файлы формата SVG можно использовать в качестве фоновых изображений во всех браузерах, за исключением Internet Explorer 8. Сохраните файл формата SVG на своем сайте и ссылайтесь на него, как на любой другой графический файл:

```
background-image: url(images/icon.svg);
```

Управление повтором фоновых изображений

Если свойство `background-image` используется без указания дополнительных атрибутов, то фоновый рисунок многократно повторяется в виде мозаики, заполняя всю веб-страницу. На этот случай можно воспользоваться свойством `background-repeat`, чтобы определить, каким образом будет повторяться фоновый рисунок:

`background-repeat: no-repeat;`

Свойство может принимать четыре значения. Рассмотрим их по порядку.

- `repeat` — параметр по умолчанию, обеспечивает повторное отображение фонового рисунка слева направо и сверху вниз до полного заполнения всего пространства веб-страницы (рис. 8.3).

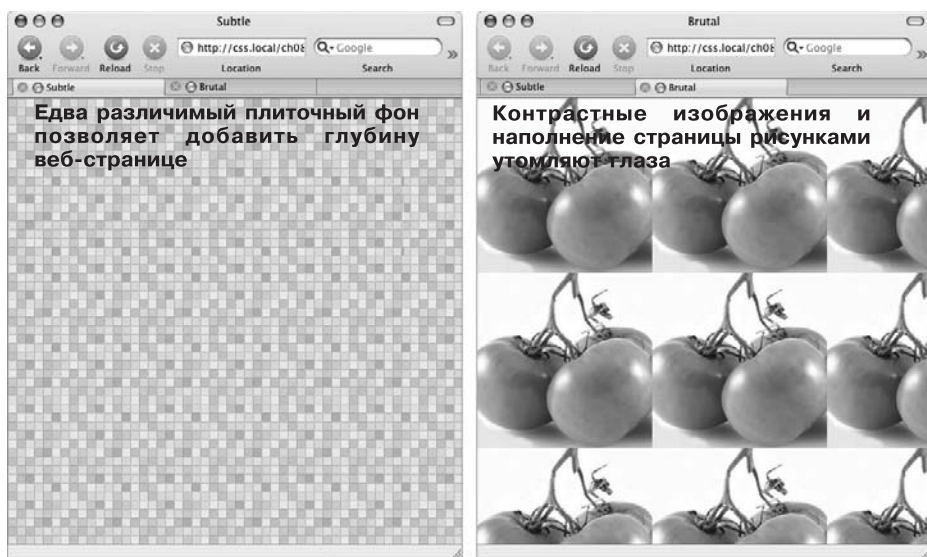


Рис. 8.3. Применяйте мозаичное отображение фоновых рисунков с осторожностью: выбирайте не слишком контрастные, удачно стыкующиеся изображения (*слева*); четкие, яркие, высоконтрастные изображения (*справа*) делают основной текст веб-страниц нечитаемым

- `no-repeat` — отображает фоновый рисунок один раз, без повторения и мозаики. Этот параметр используется на практике довольно часто, и мы также будем применять его для установки фоновых изображений элементов, отличных от `body`. Вы можете пользоваться им, чтобы поместить графический логотип в верхней части веб-страницы или создать собственные маркеры в списках (пример такого маркера приведен в практикуме этой главы).
- `repeat-x` — вызывает повторение фонового изображения горизонтально вдоль оси *X* (по всей ширине веб-страницы). Это замечательное средство для добавления графического баннера (заголовка) в верхней части веб-страницы (*рис. 8.4, слева*), декоративной границы сверху или снизу заголовка.

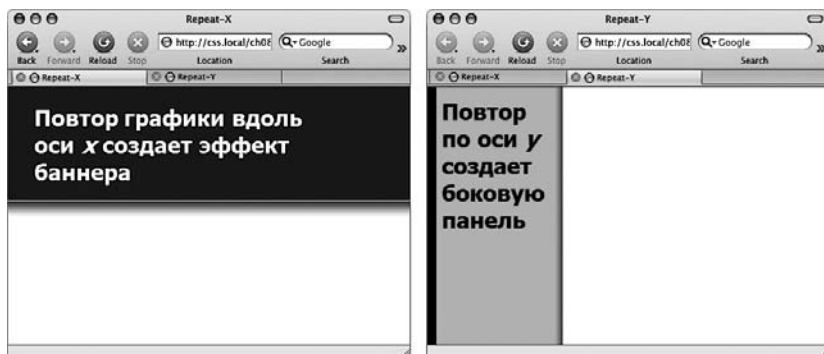


Рис. 8.4. Управляя повтором фоновых изображений, можно создать графический фон для заголовков (слева) и боковых панелей (справа) веб-страницы

- `repeat-y` — повторяет фоновое изображение вертикально вдоль оси Y (по всей высоте веб-страницы). Вы можете использовать этот параметр, чтобы добавить слева или справа веб-страницы графическую боковую панель (см. рис. 8.4, *справа*) или создать эффект тени с любой стороны элемента веб-страницы (возможно, той же боковой панели).
- `round` — мозаично повторяет фоновое изображение, как и предыдущие параметры, но никогда не обрезает его. Изображения масштабируются, чтобы заполнить область по вертикали и горизонтали без отсечения (рис. 8.5, *слева*). Для достижения указанного эффекта браузеры искажают копии изображений, нарушая исходные пропорции изображения.
- `space` — мозаично повторяет фоновое изображение, как и предыдущие параметры, но предотвращает искажение и обрезание изображений сверху или внизу. Другими словами, параметр `space` всегда отображает изображение без искажений. Соответственно, браузеры добавляют пространство между копиями изображения (см. рис. 8.5, *справа*).

Позиционирование фоновых изображений

Размещение фоновых изображений и управление их повтором — это только половина дела. Свойство позиционирования фонового изображения `background-position` позволяет несколькими способами управлять точным расположением графики. Вы можете определить начальную позицию фонового изображения по горизонтальной и вертикальной координате посредством трех ключевых слов, точных абсолютных и процентных значений.

Ключевые слова

Вы можете работать с двумя наборами ключевых слов: один из них имеет три параметра управления горизонтальным позиционированием: `left`, `center`, `right`, а другой — три параметра вертикального: `top`, `center`, `bottom` (рис. 8.6). Предположим, вы

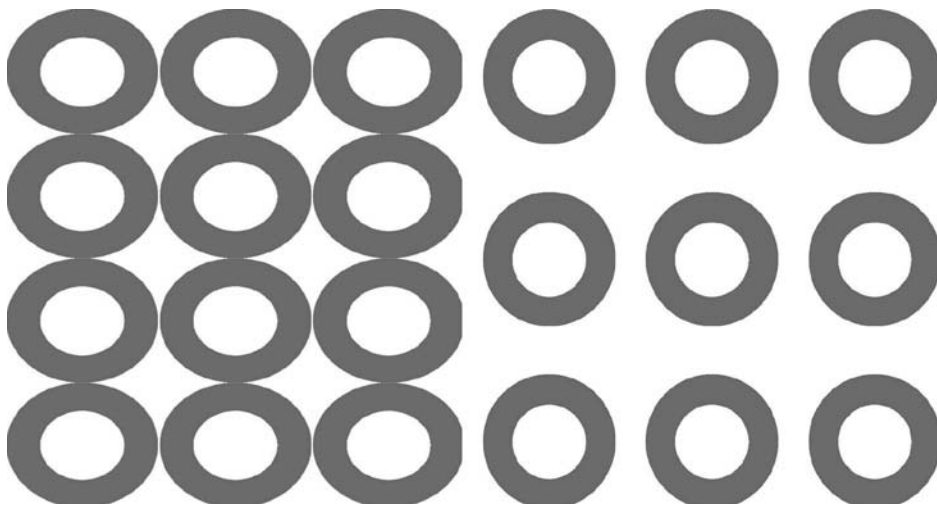


Рис. 8.5. Параметры *round* (слева) и *space* (справа) для узора *background-repeat* гарантируют, что ни одна из копий изображения не будет обрезана. Эта комбинация параметров заставляет браузер искажать пропорции изображения (слева) или добавлять дополнительное пространство между его копиями (справа)

хотите поместить рисунок прямо в центре веб-страницы. Для этого можно создать следующий стиль:

```
body {  
  background-image: url(bg_page.jpg);  
  background-repeat: no-repeat;  
  background-position: center center;  
}
```

В КУРС ДЕЛА!

Типы URL-адресов

В языке CSS при добавлении фонового изображения вы должны определить URL-адрес (*Uniform Resource Locator* — унифицированный указатель ресурса) — путь к файлу, расположенному в Интернете. Существует три типа путей: *абсолютный*, *корневой относительный* и *относительный от документа*. Все три варианта указывают, где браузер может найти определенный файл (например, другую веб-страницу, рисунок или внешнюю таблицу стилей).

Абсолютный путь похож на почтовый адрес — он содержит всю информацию, необходимую браузеру для нахождения файла. Абсолютный путь включает `http://`, имя компьютера (хоста) в Сети, папку и имя самого фай-

ла. Например: `http://www.cosmofarmer.com/images/bluegrass.jpg`.

Корневой относительный путь (относительный путь от корневого каталога) указывает, где находится файл самого верхнего уровня — корневой папки сайта. Он не содержит ни `http://`, ни доменного имени. Начинается с символа `/`, указывающего на корневой каталог сайта (папка, в которой располагается главная страница). Например, `/images/bluegrass.jpg` обозначает, что файл `bluegrass.jpg` находится в папке `images`, которая расположена в корневом каталоге. Самый простой способ определить корневой относительный путь — взять абсолютный и отбросить `http://` и имя хоста.

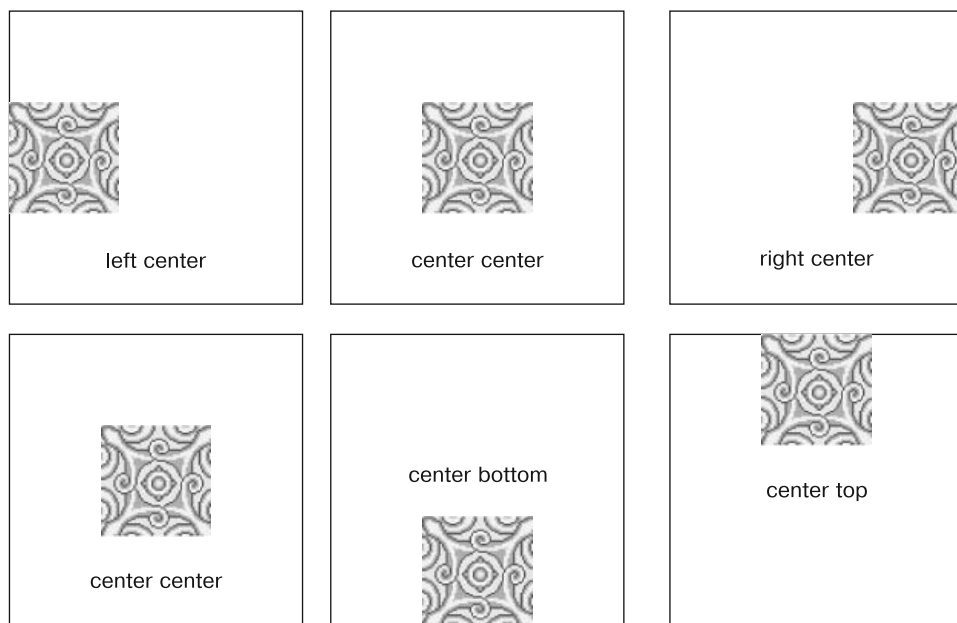


Рис. 8.6. Вы можете использовать для позиционирования фоновых изображений ключевые слова, последовательность написания которых не имеет значения. Записи `top center` или `center top` идентичны

В КУРС ДЕЛА!

Относительный от документа определяет путь к файлу от текущего документа. Когда он указан в таблице стилей, он определяет путь *до указанного файла от таблицы стилей*, а не от текущей веб-страницы.

Ниже приведено несколько подсказок относительно того, какой тип URL-адреса нужно использовать в каждом конкретном случае.

- Если вы обращаетесь к файлу, который находится на сервере, отличном от того, где размещена ваша таблица стилей, то должны использовать абсолютный путь. Это единственный тип URL-адреса, который может указывать на другой сайт.
- Корневой относительный путь хорош для доступа к изображениям, хранящимся на вашем собственном сайте. Поскольку отсчет всегда начинается с корневого каталога, вы можете перемещать таблицу стилей в любое другое место, не затрагивая при этом пути от корневого каталога до изображений сайта. Однако этим способом затруднительно

пользоваться, если вы только учитесь веб-дизайну: вы не сможете предварительно просмотреть страницу в браузере, пока не будете просматривать страницы с сервера в Интернете или установленного на компьютере для тестирования и отладки. Другими словами, если вы попытаетесь открыть веб-страницу на своем компьютере с помощью меню **Файл** ► **Открыть** (**File** ► **Open**), то вообще не увидите никаких изображений, которые помещены на веб-страницу с применением относительного пути от корневого каталога.

- Использование относительного пути от документа — лучший способ разработки веб-дизайна на собственном компьютере, без привлечения веб-сервера. Вы можете создать CSS-файлы и затем просмотреть их в браузере, открывая страницу, сохраненную на жестком диске компьютера. Они будут работать и тогда, когда вы загрузите их на рабочий сайт, но вам придется переписать URL к изображениям, если вы захотите переместить таблицу стилей в другое расположение на сервере.

Чтобы переместить изображение в верхний правый угол веб-страницы, измените параметр позиции фонового рисунка:

```
background-position: right top;
```

ПРИМЕЧАНИЕ

Если вы решили применить повторяющееся фоновое изображение (установив для свойства `background-repeat` одно из значений, описанных ранее), то начальную точку или координату первого отображаемого фонового рисунка определит свойство `background-position`. Например, если вы примените параметр `repeat`, то весь фон веб-страницы будет заполнен фоновым изображением. Но именно `background-position` укажет, с какой позиции нужно начать повторное отображение рисунка.

Ключевые слова полезны, когда необходимо создать вертикальные или горизонтальные баннеры-заголовки. Если вы хотите, чтобы фоновый рисунок был горизонтально центрирован и повторялся от верхнего до нижнего края веб-страницы, создавая фон для всего содержимого (рис. 8.7, *слева*), то можно создать следующий стиль:

```
body {  
    background-image: url(background.jpg);  
    background-repeat: repeat-y;  
    background-position: center top;  
}
```

На рис. 8.7 на левом скриншоте фоновое изображение представляет собой широкий белый блок с тенью с левой и правой сторон. Цвет фона веб-страницы — серый, и текст выглядит так, как будто он написан на белом листе бумаги, помещенном на экран компьютера.

Аналогично, применяя ключевые слова `bottom`, `center` и `top`, вы можете установить горизонтальное повторение фонового изображения в определенном месте веб-страницы (или внутри форматируемого элемента) при использовании параметра `repeat-x` (см. рис. 8.4, *слева*).

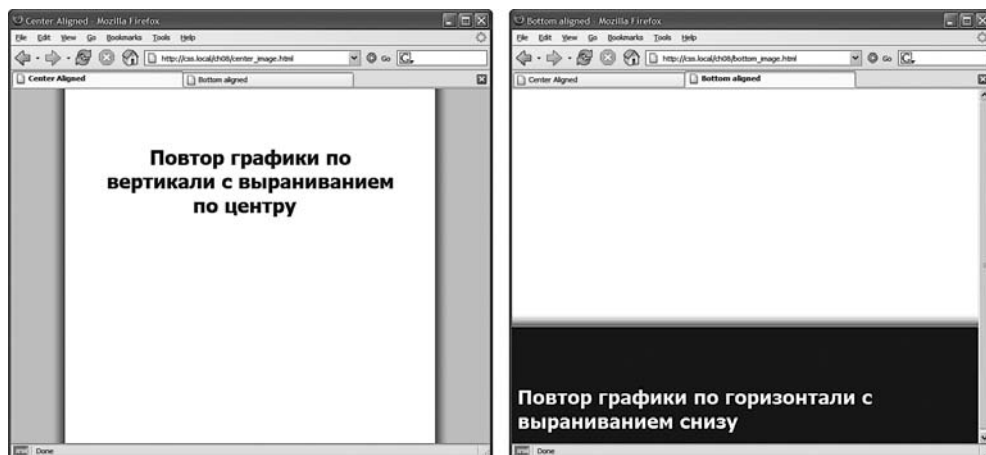


Рис. 8.7. При повторении фонового изображения вертикально (*слева*) или горизонтально (*справа*) пользуйтесь свойством `background-position`

СОВЕТ

Вы можете добавить фоновый рисунок к обоим элементам: и к `html`, и к `body`. Если вы повторяете оба изображения по горизонтали и располагаете рисунок элемента `body` вверх, а изображение элемента `html` внизу, то можете достичь эффекта двух полос, идущих через верхнюю и нижнюю части страницы, вне зависимости от высоты страницы.

ОШИБКИ БРАУЗЕРОВ

**Проблема с отображением фоновых рисунков
в нижней части окон браузеров**

Отображая рисунок в качестве фона всей веб-страницы, большинство дизайнеров часто неправильно устанавливают вертикальное позиционирование изображения. Например, если вы используете вертикальную позицию `bottom`, изображение не всегда появляется в нижней части окна браузера. Это случается, когда основное содержимое веб-страницы меньше по высоте.

Если веб-страница имеет всего несколько абзацев текста и отображается на большом мониторе, то многие браузеры принимают за нижний край окна браузера нижний край последнего абзаца текста веб-страницы. Если вы столкнетесь с такой ситуацией, добавьте следующий стиль:

```
html { height: 100%; }
```

Точные значения

Позиционировать фоновые изображения можно, используя точные значения в пикселах или единицах `em`. При этом нужно указать два значения: одно из них определяет расстояние между левым краем изображения и левым краем элемента-контейнера, а другое — расстояние между верхним краем изображения и верхним краем элемента-контейнера (другими словами, первое значение указывает горизонтальную координату, а второе — вертикальную).

Допустим, вы хотите установить собственные маркеры для списка. При добавлении в элемент `li` фонового рисунка маркеры не всегда будут правильно выровнены (рис. 8.8, *вверху*). Придется выравнивать их с помощью свойства `background-position` (см. рис. 8.8, *внизу*). В нашем случае элементы списка будут выглядеть гораздо симпатичнее, если расположить маркеры на 5 пикселей правее и на 8 пикселей ниже. Для этого добавим в стиль фонового изображения следующее свойство:

```
background-position: 5px 8px;
```

Вы не сможете указать расстояние относительно нижнего (`bottom`) или правого (`right`) края веб-страницы или формируемого элемента в пикселах или `em`, поэтому, если хотите быть уверенными в том, что изображение помещено точно в правый нижний угол, используйте в качестве единиц измерения либо ключевые слова (`bottom right`), либо процентные значения, как описывается далее. Однако вы можете использовать отрицательные значения, чтобы сместить изображение относительно левого края или поднять относительно верхнего, скрывая часть. Возможно, вы захотите применить значения для обрезки части изображения. Или, если у фонового изображения имеется большое пустое поле с левой или верхней стороны, можно использовать отрицательные значения для его устранения.

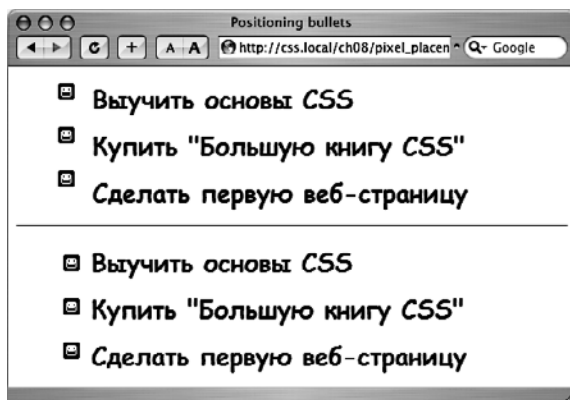


Рис. 8.8. Использование собственных изображений маркеров иногда требует их точного позиционирования так, чтобы они имели соответствующее центрирование и правильно отображались относительно текста элементов списка

Процентные значения

Для позиционирования фоновых изображений вы также можете применять процентные значения и тогда добьетесь интересного эффекта. В данном случае можно позиционировать элемент на расстояние, пропорциональное его ширине. Например, вам может понадобиться переместить рисунок на расстояние в три четверти размера элемента от заголовка, но вы не знаете ширины.

Как и в случае с пикселями или значениями в единицах `em`, вы должны указать два процентных значения: одно представляет собой горизонтальную координату, а второе — вертикальную. Относительно чего рассчитывается значение, указанное в процентах? Если объяснить в двух словах, то оно приравняется к процентному значению форматируемого элемента.

Лучший способ понять — рассмотреть несколько практических примеров. Чтобы позиционировать изображение в центре веб-страницы (аналогично изображению в центре на рис. 8.8), необходимо использовать следующий код:

```
background-position:50% 50%;
```

Свойство устанавливает координату x таким образом, что она указывает на точку изображения в 50 % от его левой стороны, которая находится в 50 % от левого края веб-страницы (или любого элемента, к которому применяется фоновое изображение). Координата y устанавливается так, что она указывает на точку изображения в 50 % от верхней стороны, которая находится в 50 % от соответствующего края веб-страницы или форматируемого элемента. Другими словами, центр изображения совпадает с центром элемента, для которого оно определено в качестве фонового. Это означает, что при использовании процентных значений точные координаты позиционирования динамически изменяются и похожи на «движущуюся мишень» (поэтому координаты позиционирования форматируемого элемента в процентах могут изменяться, если посетители веб-страницы меняют размеры окна браузера).

При использовании процентных значений в первую очередь необходимо определить *точку привязки* (координату) — позицию изображения, которую вы хотите закрепить. В этом примере отметки «50% 50%» на центральном изображении отображают его точку привязки. Далее определите точку 50/50 на самой странице (это должен быть центр страницы). Это место, в которое помещается точка привязки изображения. Три остальных изображения позиционируются аналогичным образом.

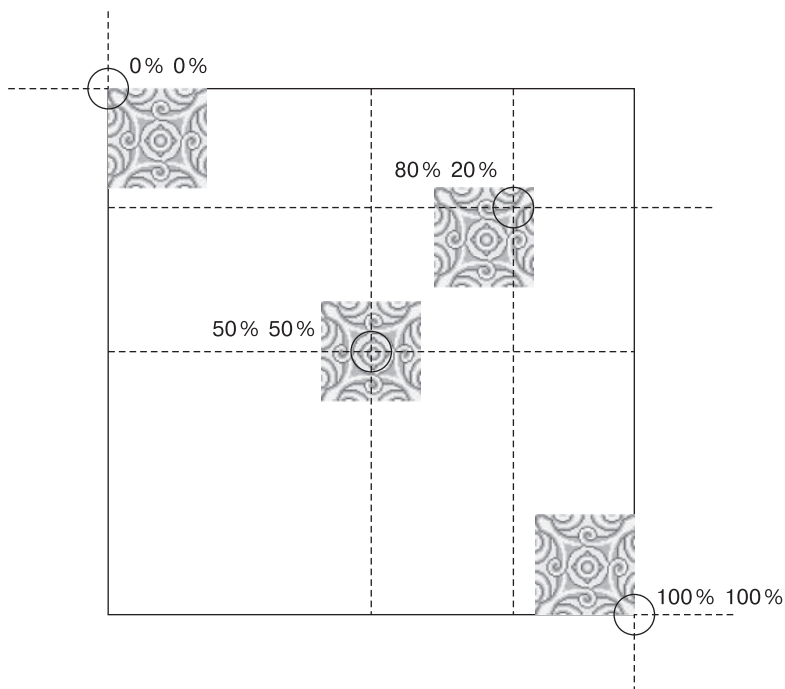


Рис. 8.9. Каждое из этих четырех изображений было позиционировано на странице (представленной большой черной рамкой) с помощью процентных значений

ПРИМЕЧАНИЕ

Позиционирование изображения по вертикали в фоне страницы с помощью процентов необязательно поместит его в правильное место, если содержимое страницы не заполняет всю высоту окна браузера.

Как и в случае с пикселями и `em`, можно назначать отрицательные процентные значения, но результаты трудно предугадать. Вы можете также комбинировать и подбирать значения в пикселях, `em` и процентах одновременно. Например, чтобы поместить изображение на расстоянии 5 пикселей от левого края элемента, но в центре элемента по высоте, применяют свойство со следующими параметрами:

```
background-position: 5px 50%;
```

ПРИМЕЧАНИЕ

Безусловно, фоновые изображения улучшают визуальное восприятие веб-страниц, но они, как правило, не отображаются при распечатке. В большинстве браузеров фоновые изображения могут распечатываться, но по умолчанию это требует дополнительной настройки параметров вывода на печать со стороны пользователя. Если вы хотите предоставить возможность посетителям сайта печатать веб-страницы с рисунками в таком виде, как они отображаются в браузере, то вместо фоновых изображений используйте элемент `img` для вставки таких важных фрагментов, как логотип сайта или схема проезда к интернет-магазину.

Фиксация позиции изображения

Когда посетитель прокручивает содержимое веб-страницы, чтобы увидеть остальную ее часть, фоновое изображение прокручивается вместе с содержимым. В результате рисунок на заднем плане перемещается вместе с текстом. Кроме того, если он не повторяется, то в процессе прокрутки исчезнет из виду. Когда вы устанавливаете в качестве фонового изображения веб-страницы логотип сайта или водяной знак, вы, вероятно, хотите, чтобы он всегда *оставался* в пределах видимости во время просмотра.

В CSS такая проблема решается с помощью свойства `background-attachment`, которое может принимать два значения: `scroll` и `fixed`. Значение по умолчанию, `scroll`, определяет такое поведение браузера, при котором фоновое изображение прокручивается вместе с текстом и другим контентом. Значение `fixed` предотвращает перемещение, жестко фиксируя его на заднем плане (рис. 8.10). Так, если вы хотите поместить логотип компании в левом верхнем углу веб-страницы и зафиксировать его там даже при прокрутке посетителями контента, можно создать следующий стиль:

```
body {  
    background-image: url(images/logo.gif);  
    background-repeat: no-repeat;  
    background-attachment: fixed;  
}
```

Определение начальной позиции фонового изображения и настройка отсечения

В CSS есть возможность сообщить браузеру, где фоновое изображение должно начаться по отношению к границам, отступам и содержимому элемента. Например, когда изображение используется для повторяющегося заполнения, оно появляется в верхнем левом углу отступа элемента (верхнее среднее изображение на рис. 8.11). Но начальную позицию изображения можно изменить, воспользовавшись свойством `background-origin`. Этому свойству присваивается одно из трех значений.

- `border-box` — изображение помещается в верхний левый угол области границы (см. рис. 8.11, *слева вверху*).
- `padding-box` — изображение помещается в верхний левый угол области отступа (см. рис. 8.11, *посередине вверху*). Это обычное место, куда браузер помещает фоновое изображение.

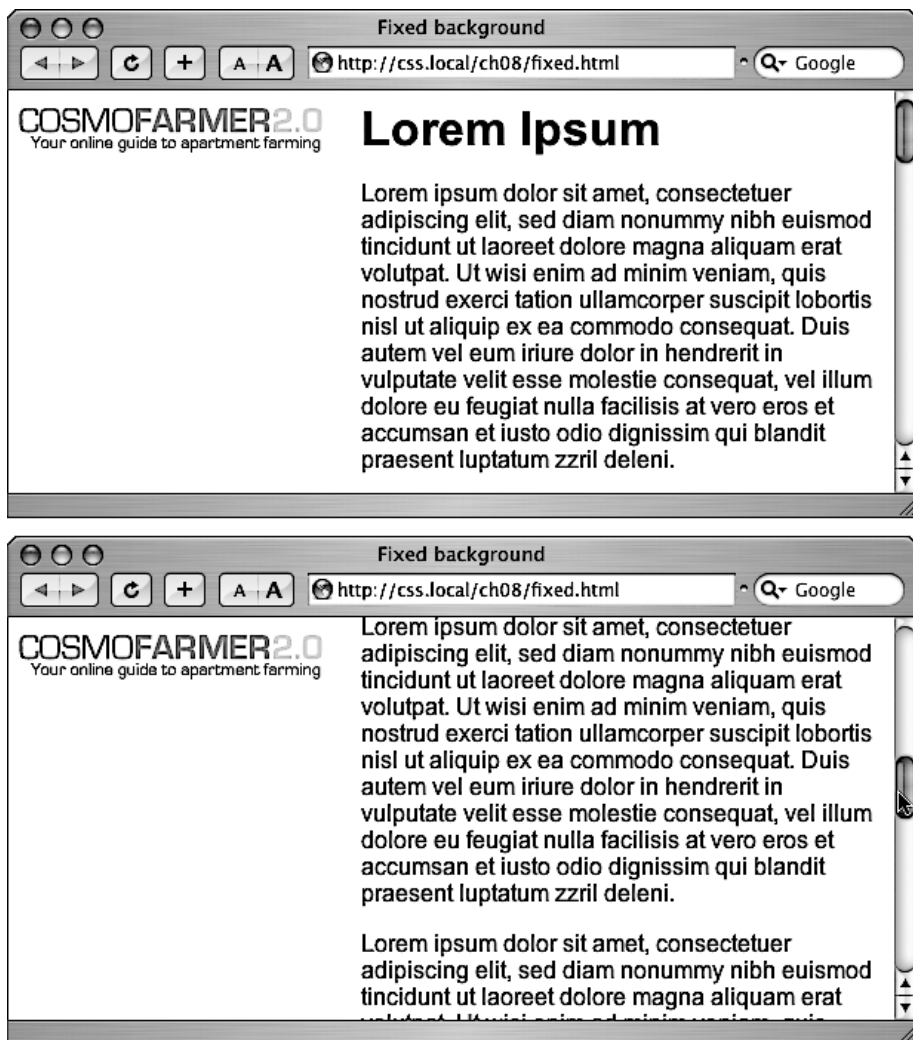


Рис. 8.10. Ищите способ закрепить логотип сайта (в примере изображение CosmoFarmer 2.0), чтобы он при прокрутке страницы остался на месте? Этого можно добиться с помощью свойства `background-attachment` и значения `fixed`

- `content-box` — изображение помещается в левый верхний угол области содержимого (см. рис. 8.11, *справа сверху*).

Разумеется, эти настройки не имеют смысла, если вокруг элемента нет ни отступа, ни границы. Кроме того, эффект может быть практически незаметен, особенно если у заполняющего повторяющегося изображения не видно швов.

ПРИМЕЧАНИЕ

Свойства `background-origin` и `background-clip` в Internet Explorer 8 и его более ранних версиях не работают.

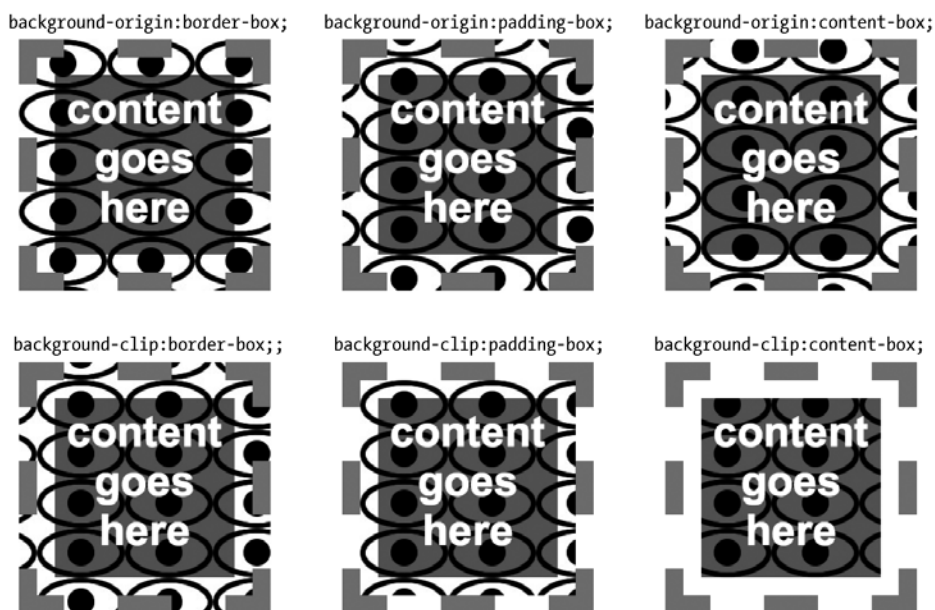


Рис. 8.11. Свойство `background-clip` можно также применить к фоновому цвету. Если добавить прерывистую границу (точечную, пунктирную или двойную), фоновый цвет обычно появляется позади линий границы в пустых областях. Но чтобы ограничить отображение цвета, не позволяя ему пересекать пределы границ, нужно присвоить свойству `background-clip` значение `padding-box`

Но эффект может быть весьма заметен, если изображение не повторяется. Например, если вы выберете настройку `no-repeat` и фоновое изображение появится только один раз, его можно поместить в верхний левый угол области (внутри любых отступов):

```
background-image: url(logo.png);
background-repeat: no-repeat;
background-origin: content-box;
```

Кроме того, свойство `background-origin` может эффективно использоваться с другим свойством — `background-clip`, которое ограничивает область появления фонового изображения. Обычно фоновые изображения заполняют *всю* область элемента, включая пространство позади границ и отступов (см. рис. 8.11, *слева внизу*). Но область отображения рисунка можно ограничить с помощью трех значений.

- `border-box` — позволяет рисунку отображаться позади содержимого и любых границ. Заметить это можно только при использовании линий границ с промежутками, например пунктирных (см. рис. 8.11, *слева внизу*). Это обычное поведение браузеров, поэтому использовать эту настройку нет смысла.
- `padding-box` — дает возможность ограничить любое фоновое изображение областью отступов и содержимого элемента (см. рис. 8.11, *посередине внизу*). Эта

настройка пригодится при использовании линий границ с промежутками, например пунктирных, и нежелании видеть рисунок в разрывах границы.

- `content-box` — позволяет ограничить фоновое изображение областью содержимого элемента (см. рис. 8.11, *справа внизу*).

Сочетая свойства `background-origin` и `background-clip`, можно отобразить рисунок в верхнем левом углу области содержимого и позади только содержимого:

```
background-origin: content-box;  
background-clip: content-box;
```

Масштабирование фоновых изображений

Обычно изображение, помещенное в качестве фона элемента страницы, принимает размер, который был задан при его создании. Однако свойство `background-size` позволяет управлять размером фонового рисунка. Чтобы установить размер, можно использовать как точные значения, так и ключевые слова.

- Для того чтобы установить размер изображения, нужно задать его высоту и ширину. Для этого можно воспользоваться абсолютными значениями, выраженными в пикселах:

```
background-size: 100px 200px;
```

Этот код устанавливает для фонового изображения ширину 100 пикселей и высоту 200 пикселей. Можно также установить только значения ширины или высоты, задав для оставшегося значения ключевое слово `auto`:

```
background-size: 100px auto;
```

В таком случае фоновое изображение будет иметь ширину 100 пикселей, а браузер автоматически установит его высоту, сохраняя пропорции (чтобы не возникло искажений).

Можно также использовать процентные значения. Если нужно масштабировать рисунок для полноценного заполнения фона, можно для обеих настроек использовать значения 100% (рис. 8.12, *слева*):

```
background-size: 100% 100%;
```

- Ключевое слово `contain` приводит к изменению размеров изображения с целью заполнения фона элемента с сохранением пропорций (см. рис. 8.12, *посередине*). В зависимости от формы изображения и элемента рисунок увеличивается, чтобы поместиться либо по ширине, либо по высоте элемента.

```
background-size: contain;
```

- Ключевое слово `cover` заполняет изображением фон элемента, соотнося высоту/ширину элемента и изображения, не искажая пропорции (см. рис. 8.12, *справа*).

```
background-size: cover;
```

Использование свойства `background-size` практически всегда приводит к изменению размеров исходного изображения: если изображение меньше заполняемого

элемента, браузер увеличивает его масштаб, что зачастую выражается заметной пикселизацией и ухудшением качества изображения (что видно по фоновым изображениям на рис. 8.12).



Рис. 8.12. Только свойство `background-size` позволяет менять размеры фонового изображения. Его поддерживает большинство браузеров, за исключением Internet Explorer 8, поэтому используйте свойство с оглядкой

СОВЕТ

Свойство `background-size` может оказаться особенно удобным при работе с элементами, размер которых определяется с использованием процентных значений, например при верстке адаптивных дизайнов, рассматриваемых в главе 14. Допустим, если поместить изображение в фон баннера, занимающего 960 пикселей при просмотре на мониторе настольного компьютера, но сжать его до 480 пикселей при просмотре на смартфоне, можно поместить в баннер большое изображение и воспользоваться следующей настройкой:

```
background-size: 100% auto;
```

Эта настройка инструктирует браузер изменить размер изображения таким образом, чтобы оно поместилось в сжатый баннер.

ИНФОРМАЦИЯ ДЛЯ ОПЫТНЫХ ПОЛЬЗОВАТЕЛЕЙ

Добавление специальных эффектов к фоновым изображениям

Если у вас есть опыт работы в таких графических редакторах, как программа Photoshop, вы, вероятно, знакомы с режимами наложения, которые позволяют преобразовать два изображения в одно, заставив их взаимодействовать визуально. Рассмотрим пример, как это работает. Представьте, что вы поместили в фоновый элемент фотографию, а также задали для нее фоновый цвет. Как правило, любое изображение, не имеющее прозрачных областей, перекроет фоновый цвет, расположенный позади него.

Но как быть, если вы хотите смешать изображение и фоновый цвет? Например, вы хотели бы, чтобы фоновый цвет был виден *только* сквозь белые области фотогра-

фии. Относительно новое свойство `background-blend-mode` предоставляет такую возможность. Необходимо применить это свойство к элементу с фоновым цветом и изображением. Оно использует один из 16 режимов наложения на выбор, каждый из которых позволяет добиться различных результатов.

Еще один пример. Предположим, вы поместили фотографию помидора в качестве фонового объекта элемента `div` и хотите добавить оранжевый цвет, чтобы смешать его с фотографией. В данном случае необходимо установить оранжевый цвет в качестве фонового, а затем задать режим наложения следующим образом:

ИНФОРМАЦИЯ ДЛЯ ОПЫТНЫХ ПОЛЬЗОВАТЕЛЕЙ

```
background-image: url(tomato.png);  
background-color: orange;  
background-blend-mode: screen;
```

Режим наложения можно применять к нескольким изображениям (см. раздел «Использование множественных фоновых изображений» главы 8). Это позволит создать очень интересный визуальный эффект.

По адресу tinyurl.com/nllpow7 опубликован видеорок использования режима наложения в Photoshop. Свойство `background-blend-mode` в CSS работает так же.

По адресу tinyurl.com/mnm6kzn есть пример использования режима наложения в CSS (сейчас ни одна версия Internet Explorer (а также Edge) не поддерживает эту функцию).

Сокращенная запись свойства `background`

Как вы видите из примеров предыдущих разделов, чтобы воспользоваться всеми преимуществами фоновых изображений, необходимо применять свойства, управляющие параметрами фоновых изображений. Но многократный повторный набор таких длинных свойств, как `background-image`, `background-attachment` и т. д., отнимает много времени. Существует более простой метод — применение сокращенного свойства `background`.

Фактически вы можете объединить и перечислить все свойства фона (включая `background-color`) в единственной строке CSS-кода. Используйте свойство `background`, за которым должны следовать значения свойств `background-image`, `background-position`, `background-size`, `background-repeat`, `background-origin`, `background-clip`, `background-attachment` и `background-color`.

ПРИМЕЧАНИЕ

Браузер Internet Explorer 8 не поддерживает свойства `background-size`, `background-origin` и `background-clip`. Если вы используете их в сокращенном свойстве `background`, IE 8 не будет применять их к элементу, проигнорировав.

Следующий стиль добавляет только одно изображение (`no-repeat`) по центру веб-страницы, уменьшает его размер до 50 %, фиксирует изображение (чтобы оно не прокручивалось вместе со страницей) и устанавливает фон белого цвета:

```
body {  
    background: url(bullseye.gif) center center / 50% no-repeat fixed #FFF;  
}
```

Если вы указываете положение (например, в предыдущем примере это `center center`), свойство `background-size` (значение которого в примере равно 50%), используйте между ними разделитель `/`.

Вовсе не обязательно указывать абсолютно все параметры свойства. Вы можете использовать один из них или любое сочетание. Например, `background: yellow` равнозначно `background-color: yellow`. Все те параметры свойства, которые вы не определите сами, будут иметь стандартные значения по умолчанию. Допустим, вы определили только само фоновое изображение:

```
background: url(image/bullseye.gif);
```

Этот код эквивалентен следующему стилю:

```
background: url(image/bullseye.gif) left top / 100% repeat scroll border-box border-box transparent;
```

Поведение, когда при отсутствии определения значения происходит возвращение к значениям по умолчанию, может привести к весьма неожиданным результатам. Предположим, что к стилю добавляются два объявления:

```
background-color: yellow;  
background: url(image/bullseye.gif) no-repeat;
```

Возможно, вы ожидали увидеть GIF-изображение на желтом фоне. Но вы его не увидите, потому что при обнаружении свойства `background` без указания цвета браузер присваивает свойству `background-color` значение `transparent` (невидимый). Чтобы выйти из этой сложной ситуации, нужно указать свойство `background-color` вторым:

```
background: url(image/bullseye.gif) no-repeat;  
background-color: yellow;
```

Кроме того, когда к одному и тому же элементу применяется сразу несколько свойств, можно в конечном итоге случайно стереть фоновые изображения. Предположим, необходимо добавить фоновое изображение к каждому абзацу страницы, для чего создается следующий стиль:

```
p {  
  background: url(icon.png) left top no-repeat rgb(0.30,0);  
}
```

Затем принимается решение, что после заголовка второго уровня каждый первый абзац должен иметь синий фон, и создается следующий стиль:

```
h2 + p {  
  background: blue;  
}
```

В этом втором стиле используется сокращенная запись, сбрасывающая все остальные свойства фона к их значениям по умолчанию. Что касается изображения, то по умолчанию оно вообще не используется, поэтому вместо простого добавления к абзацу синего фона с оставлением изображения без изменений этот стиль вообще его удаляет!

Иначе говоря, сокращенная запись свойства `background` может сэкономить время на написании кода, но может и принести проблемы, о чем не нужно забывать.

ЧАВО

Поиск бесплатных коллекций изображений

Я не художник, не умею рисовать, и у меня нет даже цифровой камеры. Где же можно найти графический материал для сайта?

Нужно отдать должное Всемирной паутине. Она представляет собой универсальное средство поиска, благодаря которому можно выйти из любой ситуации.

ЧАВО

Существует множество платных сайтов, содержащих коллекции готовых фотографий и иллюстраций, но наряду с ними также имеется довольно много бесплатного материала.

Ресурс foter.com содержит свыше 228 миллионов бесплатных фотографий. Он собирает лицензионные фотографии фонда Creative Commons из всех уголков Всемирной паутины. Ресурс также содержит бесплатный WordPress-плагин, с помощью которого можно легко добавлять стоковые изображения для вашего блога. На сервисе unsplash.com вы найдете потрясающие бесплатные фотографии природы, архитектуры и животных.

Образцы фотографий можно найти на сайте morguefile.com, где есть множество замечательных снимков, сделанных людьми, любящими природу. Сайт freeimages.com — еще один отличный фоторесурс. Сайт openphoto.net предлагает коллекции изображений, предоставляя возможность их применения с некоторыми ограничениями. Вы также можете воспользоваться поиском иллюстраций с лицензией Creative Commons и найти изображения (а также видео и музы-

ку), которые могут быть задействованы в личных или коммерческих проектах: search.creativecommons.org. Кроме того, вы можете искать иллюстрации с лицензией Creative Commons на сайте flickr.com/creativecommons (надо отметить, что за фотографии на бесплатных сайтах не нужно платить, но не все они могут использоваться в коммерческих проектах; прежде всего читайте комментарии мелким шрифтом ко всем снимкам, с которыми собираетесь работать).

Если вы ищете маркеры для списков, значки для панели навигации, файлы фоновых узоров для заполнения страницы, то вам доступно несколько сайтов. Ресурс flaticon.com предоставляет доступ к огромной базе бесплатных векторных знаков, и вы можете использовать их и в виде PNG- и SVG-файлов на своем сайте. Сайт tinyurl.com/k9zfhfz предлагает бесплатный набор из 121 значка. Если вас интересуют мозаичные фоновые узоры, посетите один из следующих сайтов: tinyurl.com/5kwqqd, tinyurl.com/6dva2f или tinyurl.com/6dva2f. Можете также создать собственный мозаичный фон с помощью следующих инструментов: bgpatterns.com, stripegenerator.com или patterncooler.com.

Использование множественных фоновых изображений

Хотя обычно достаточно и одного фоновое изображение, доступна возможность размещения сразу нескольких изображений в виде слоев. Предположим, что нужно добавить фоновое изображение к боковой панели, чтобы придать ей видимость свитка (рис. 8.13). Если поместить в качестве фона одно изображение, то сначала оно может работать (см. рис. 8.13, *слева вверху*), но если добавить к боковой панели слишком много текста, внешний вид ухудшится (см. рис. 8.13, *справа вверху*). Причина в том, что у изображения один размер и оно не будет становиться больше или меньше, чтобы подстроиться под размер боковой панели.

На этот случай вы можете добавить к фону элемента несколько изображений. В случае со свитком можно использовать три фоновых изображения: одно для верхней части свитка, одно для нижней и одно для его текстовой области. Последнее изображение представляет собой бесшовную плитку, поэтому при увеличении высоты боковой панели изображение заполняет новое пустое пространство плитками.

Разумеется, несколько изображений можно использовать и для решения менее сложных задач. Предположим, что к фону элемента нужно добавить красочное текстурированное изображение, а также двухцветный логотип. Как сказано во врез-

ке из раздела «Добавление фоновых изображений», для многоцветного изображения больше подходит формат JPEG, а для областей, содержащих сплошные цвета, к которым можно отнести логотип, больше подходит формат PNG8. В данном случае можно объединить преимущества, используя JPEG для многоцветного фона и PNG8 для логотипа.



Рис. 8.13. Учтите, что такой способ создания нескольких фоновых изображений не работает в Internet Explorer 8

Для создания нескольких фоновых изображений нужно воспользоваться свойством `background-image` со списком изображений с запятой в качестве разделителя:

```
background-image: url(scrollTop.jpg),  
                  url(scrollBottom.jpg),  
                  url(scrollMiddle.jpg);
```

ПРИМЕЧАНИЕ

В данном примере кода каждый URL-адрес находится на отдельной строке, но это соблюдать не обязательно. Все это можно набрать и одной (но очень длинной) строкой. Однако многие разработчики считают, что код будет легче читаться, если ссылку на каждое изображение поместить в отдельной строке и воспользоваться пробелами (или табуляцией) для отступа строк, располагая ссылки друг под другом. Нужно лишь помнить о необходимости указать запятую после каждой ссылки, за исключением последней, для которой обычно требуется точка с запятой, завершающая объявление.

Поскольку фоновые изображения обычно выстраиваются в плитку, зачастую необходимо также включать и объявление свойства `background-repeat` (если этого не сделать, изображения начнут выстраиваться в плитку одно над другим, перекрывая собой все нижние изображения). Можно добавить и другие свойства фона, состоящие из простого набора значений, для которого в качестве разделителя используется запятая:

```
background-repeat: no-repeat,  
                  no-repeat,  
                  repeat-y;
```

При использовании нескольких значений, подобных этим, первое значение (в примере — `no-repeat`) применяется для первого изображения, указанного в свойстве `background-image` (`scrollTop.png`); второе применяется со вторым изображением и т. д. Поскольку во всем этом можно быстро запутаться, многие веб-разработчики используют для указания нескольких изображений сокращенный метод записи:

```
background: url(scrollTop.jpg) center top no-repeat,  
            url(scrollBottom.jpg) center bottom no-repeat,  
            url(scrollMiddle.jpg) center top repeat-y;
```

ПРИМЕЧАНИЕ

Несколько фоновых изображений кладутся стопкой друг на друга, как слои в программе редактирования изображений. Какое из них появится в верхнем слое, определяется порядком перечисления фоновых изображений. Изображение, указанное первым, появляется в верхнем слое элемента, второе — во втором слое и последнее появляется в нижнем слое. В предыдущем примере кода верхняя часть свитка (`scrollTop.jpg`) находится выше его нижней части (`scrollBottom.jpg`), которая, в свою очередь, находится выше текстовой области свитка (`scrollMiddle.jpg`).

Использование градиентных фонов

Градиент — это плавный переход цвета, к примеру, от синего к красному или от черного к белому. Градиенты входят в состав основных элементов любой программы редактирования изображений. Создание тонких переходов от одного схожего цвета к другому придает изображению своеобразную расплывчатость;

Apple использует градиенты в изображениях кнопок и других элементов пользовательского интерфейса операционных систем OS X и iOS. Раньше приходилось обращаться к программе Photoshop и создавать огромные файлы с градиентом для последующего использования. Теперь задачу создания градиента можно возложить на браузер.

В CSS поддерживаются фоновые градиенты, представляющие собой, по сути, фоновые изображения, создаваемые браузером на лету. Фактически для создания градиента используется обычное свойство `background-image`. Существует несколько типов градиентов.

Линейные градиенты

Самым простым является *линейный градиент*. Он распространяется по прямой от одного конца элемента к другому, демонстрируя плавный переход от одного цвета к другому (рис. 8.14). Нужно только указать его направление в градусах или с помощью ключевых слов `top`, `bottom`, `right`, `left` либо их комбинаций, например `bottom left`.



Рис. 8.14. Линейные градиенты позволяют отказаться от устаревшего метода добавления градиентов к фону элемента: создания градиента в виде графического изображения в программе Photoshop или Fireworks с последующим использованием свойства `background-image` для помещения этого изображения в качестве фона элемента

Например, чтобы нарисовать градиент, изменяющий цвет от черного к белому от левого угла к правому, нужно воспользоваться следующим кодом:

```
background-image: linear-gradient(to right, black, white);
```

Чтобы нарисовать градиент, изменяющий цвет от черного к белому от верхней части элемента к нижней, нужно воспользоваться таким кодом:

```
background-image: linear-gradient(to bottom, black, white);
```

Вы также можете нарисовать градиент под углом, используя ключевые слова, такие как `bottom right`, для указания направления градиента. Чтобы нарисовать градиент, изменяющий цвет от оранжевого к красному от верхнего левого угла элемента к его правому нижнему углу, нужно воспользоваться следующим кодом:

```
background-image: linear-gradient(to bottom right, orange, red);
```

Градиенты можно создавать с помощью любых значений цвета, принятых в языке CSS (см. раздел «Форматирование текста цветом» главы 6), например ключевых слов, таких как `white` или `black`, шестнадцатеричного кода (`#000000`) или значений RGB, к примеру `rgb(0,0,0)`, и т. д.

Но вы не ограничены применением ключевых слов. Можно указать угол, определяющий направление градиента. Угол записывается в виде значения в диапазоне от 0 до 360, а за ним указывается ключевое слово `deg`: `0deg` (рис. 8.15). Например, значение `0deg` соответствует верхнему краю элемента, поэтому градиент начнется в нижней части элемента и закончится в верхней. Другими словами, код выглядит следующим образом:

```
background-image: linear-gradient(to top, black, white);
```

или

```
background-image: linear-gradient(0deg, black, white);
```

Значения в углах повторяют часовую стрелку на циферблате, поэтому значение `90deg` соответствует правому краю элемента (то же самое, что ключевое слово `right`), `180deg` — нижнему краю элемента (ключевое слово `bottom`), а `270deg` — левому краю элемента (`left`).

При использовании значений углов браузер проводит воображаемую линию через центр элемента. Указанный угол является углом наклона этой линии, а также определяет точку, в которой заканчивается градиент. Например, градиент, показанный на рис. 8.15, создан под углом 48° . Браузер проводит воображаемую линию через центр элемента, отмечая угол 48° . Это означает, что градиент начинается у левой нижней границы элемента и движется к правой верхней границе.

Например, для градиента, показанного на рис. 8.14, *слева внизу*, используется следующее объявление:

```
background-image: linear-gradient(135deg, rgb(0,0,0), rgb(204,204,204));
```

По адресу tinyurl.com/nu47ro6 вы найдете интерактивный пример, как различные значения углов влияют на градиент.

Узлы градиента

В рассматриваемых выше градиентах использовались только два цвета, но их можно добавить любое количество. Дополнительные цвета называются *цветовыми*

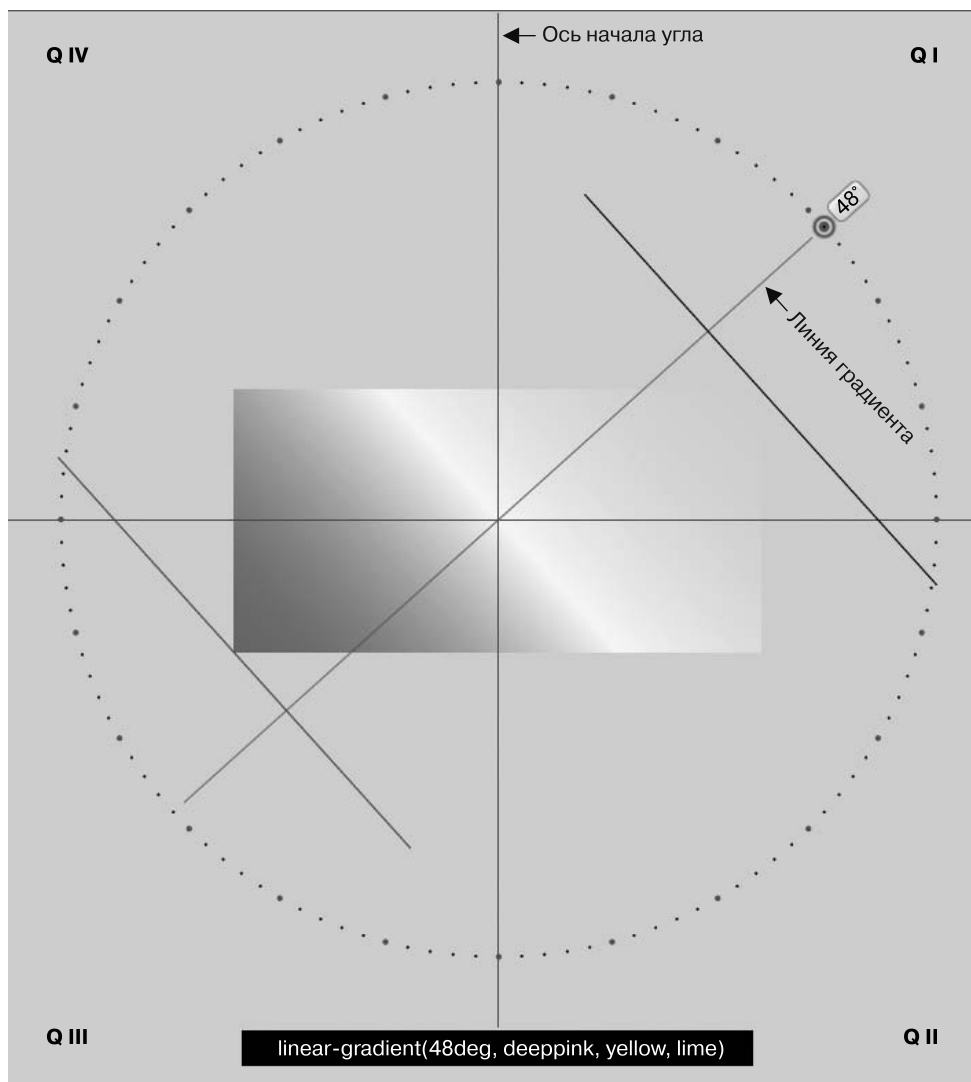


Рис. 8.15. Значение в градусах, указанное для линейного градиента, применяется для создания линии, проходящей через середину воображаемой точки центра элемента. Градиент рисуется вдоль этой линии. Значение определяет как точку старта, так и наклон градиента

узлами и позволяют фону осуществлять плавный переход от одного цвета ко второму, затем от второго к третьему и т. д., пока градиент не будет завершен последним цветом. Для добавления цветовых узлов нужно добавить к градиенту дополнительные значения цвета. Например, нижнее правое изображение на рис. 8.14 состоит из трех цветов и градиент переходит от белого к черному и от черного к белому цвету:

```
background-image: linear-gradient(to right, black, white, black);
```

Разумеется, можно применять любые значения цвета (имена, шестнадцатеричные или RGB-значения) и даже перемешивать их друг с другом. Следовательно, предыдущую строку кода можно записать следующим образом:

```
background-image: linear-gradient(to right, black, rgb(255,255,255), #FFFFFF, HSL(0,0%,0%));
```

Браузеры распределяют цвета равномерно, поэтому в этом примере в самой крайней левой точке фон черный, в центре фон белый, а в самой крайней правой точке он снова черный. Но можно разместить разные цветовые узлы и в более конкретно указанных местах фона, добавив после цвета второе значение.

СОВЕТ

Если добавить к элементу с градиентным фоном свойство `background-position`, можно получить весьма необычные эффекты. По сути, можно изменить то, что браузером рассматривается в качестве стартовой точки градиента.

Предположим, нужно получить в начале градиента темно-красный цвет, затем быстро перейти к ярко-оранжевому примерно на 10 % протяженности элемента, оставляя оранжевый цвет на большем протяжении элемента, а затем быстро вернуться к темно-красному цвету, как на верхнем изображении на рис. 8.16. Для этого нужно использовать четыре цветовых узла — красный, оранжевый, оранжевый и красный, и указать, где должны располагаться два оранжевых узла:

```
background-image: linear-gradient(to right, #900, #FC0 10%, #FC0 90%, #900);
```

Обратите внимание на то, что значение 10% применяется ко второму цвету (оранжевому): тем самым браузеру сообщается, что он должен достичь этого цвета к 10 % ширины элемента. Точно так же значение 90% показывает, что браузеру нужно оставить оранжевый цвет до достижения 90 % ширины элемента, а затем приступить к плавному переходу к темно-красному цвету на правом краю элемента.

ПРИМЕЧАНИЕ

Для указания позиций цветовых узлов не обязательно указывать проценты. Можно также использовать пиксели или единицы `em`. Но указание процентных значений является более гибким решением, приспособляющимся к изменениям ширины и высоты элемента. Одним из исключений является рассматриваемый далее повторяющийся линейный градиент. При создании этого типа плиточных градиентов очень хорошо подойдут пиксельные значения.

Как для первого, так и для последнего цвета указывать какие-либо значения положения не нужно, поскольку браузер предполагает, что первый цвет начинается с 0 %, а последний заканчивается на 100 %. Но если нужно сохранять начальный цвет неизменным на некоторой протяженности элемента, можно после первого цвета указать значение этой протяженности. Например, нижнее изображение на рис. 8.16 создано с помощью такого объявления:

```
background-image: linear-gradient(to right, #900 20%, #FC0 30%, #FC0 70%, #900 80%);
```

Обратите внимание, что у первого цвета — `#900` — также есть позиция 20%. Это означает, что первые 20 % протяженности элемента (слева направо) будут иметь фон из сплошного красного цвета. Затем, от 20%-ной до 30%-ной точки градиент плавно перейдет от красного к оранжевому цвету.

СОВЕТ

Ключевое слово `transparent` можно использовать для любого цвета в градиенте, чтобы видеть область, расположенную позади него, например фоновый цвет элемента или даже другой линейный градиент. Помните, что к элементу может быть применено несколько фоновых изображений, поэтому вы можете использовать несколько градиентов в одном элементе. Например, попробуйте добавить следующий код для достижения невероятного эффекта:

```
background-image:  
  linear-gradient(cyan, transparent),  
  linear-gradient(225deg, magenta, transparent),  
  linear-gradient(45deg, yellow, transparent);
```



Рис. 8.16. Область появления цветового узла на градиенте можно указать вполне конкретно. В верхнем примере к 10 % ширины элемента слева направо темно-красный цвет переходит в оранжевый

Поддержка браузера Internet Explorer

Градиенты не поддерживаются в программе Internet Explorer 9 и более ранних ее версиях. Если вы решили использовать градиенты, для IE 9 и более ранних версий нужно указывать резервный цвет. Подберите сплошной цвет, соответствующий общему тону вашего градиента, и объявите его первым, а за ним разместите все объявления градиентов:

```
background-color: #FC0;  
background-image: linear-gradient(to bottom, #900, #FC0, #900);
```

Браузер Internet Explorer 9 применит фоновый цвет и пропустит все остальные, неподдерживаемые объявления. Другие браузеры применяют фоновый цвет, но также создадут градиент, который перекроет фоновый цвет. Если применяются RGBA-цвета с некоторой степенью прозрачности, то вам не захочется, чтобы сквозь них просматривался фоновый цвет. В таком случае воспользуйтесь сокращенной формой записи свойства background, и свойство background-color будет замещено (благодаря особому поведению сокращенной формы записи свойства background). Тогда для объявления цвета в формате RGBA можно будет добавить следующий код:

```
background-color: #FC0;  
background: linear-gradient(to bottom, rgba(153,0,0,.5), #FC0,  
    rgba(153,0,0,.5));
```

Повторяющиеся линейные градиенты

Обычно линейный градиент заполняет весь элемент от первого цвета с одного края элемента до последнего цвета на его противоположном краю. Но при желании можно создать их повторяющийся вариант. По сути, при этом определяется градиент с указанными цветовыми узлами, браузер рисует градиент, а затем повторяет этот шаблон, распространяя его в виде плиток по фону элемента. Например, чтобы получить повторяющийся градиент, как на изображении на рис. 8.17, *слева*, следует написать такой код:

```
background-image: repeating-linear-gradient(45deg, #900 20px, #FC0 30px, #900 40px)
```

В подобных случаях для цветовых узлов лучше всего подойдут пиксельные значения. Получается, что браузер рисует градиент, начинающийся в нижнем левом углу с 20 пикселей темно-красного цвета, затем до 30 пикселей идет переход к оранжевому цвету, а затем до 40 пикселей происходит обратный переход к темно-красному цвету. После того как этот градиент будет нарисован, браузер разместит его в виде плитки в фоне элемента.

ПРИМЕЧАНИЕ

Для экспериментов с градиентами имеются широчайшие возможности. Чтобы посмотреть только на малую часть тех удивительных результатов, которых можно достичь в ходе таких экспериментов, посетите галереи шаблонов CSS по адресам tinyurl.com/85cb36t, tinyurl.com/3ashxxj. Кроме того, оцените удивительные возможности использования градиентов для создания флагов со всего мира по адресу tinyurl.com/qc4l6wk и некоторые невероятные градиенты, созданные с помощью режима наложения background-blend, описанного во врезке «Добавление специальных эффектов к фоновым изображениям» в разделе «Позиционирование фоновых элементов», tinyurl.com/nl9opcw.

Можно также использовать повторяющиеся градиенты для создания однотонных полос без каких-либо излишеств и переходов между цветами. Например, изображение, показанное на рис. 8.17, *справа*, создано с помощью такого объявления:

```
background-image: repeating-linear-gradient(45deg, #900 0, #900 10px, #FC0 10px, #FC0 20px);
```

Оно начинается с темно-красного цвета (#900) в точке 0 и снова переходит к красному цвету в точке 10px. Поскольку переход задается между одинаковыми цветами, браузер рисует его в виде сплошного цвета. Затем на 10 пикселах задается переход к оранжевому цвету (#FC0). Поскольку это та же самая точка, в которой заканчивается красный цвет, никакого плавного перехода не происходит, шаблон не содержит перехода от красного к оранжевому цвету. И наконец, переход к такому же оранжевому цвету осуществляется до 20 пикселей, создавая еще одну сплошную линию. Поскольку это повторяющийся линейный градиент, браузер заполняет шаблоном фон элемента.

Повторяющиеся градиенты работают в большинстве браузеров, но только не в Internet Explorer 9 и более ранних версиях (работает и в Edge). Поэтому лучше будет добавить запасной вариант фоновой заливки:

```
background-color: #FC0;  
background: repeating-linear-gradient(to bottom left, #900 20px, #FC0 30px, #900 40px)
```

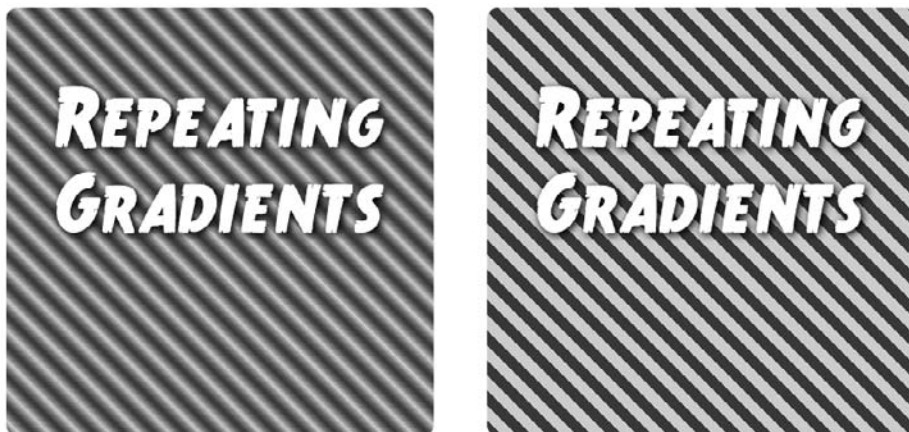


Рис. 8.17. Используя повторяющиеся линейные градиенты, можно создавать полосатые шаблоны. С помощью различных цветовых узлов можно создавать полосы из четких или расплывчатых по краям линий

Радиальные градиенты

В CSS также предоставляется способ создания *радиальных градиентов*, то есть градиентов, расходящихся наружу по круговой или эллиптической схеме (рис. 8.18). Синтаксис похож на тот, что применяется для линейных градиентов, нужно только задать начальный цвет (цвет в середине градиента) и конечный (цвет в конце

градиента). Например, верхнее левое изображение на рис. 8.18 создано с помощью следующего кода:

```
background-image: radial-gradient(red, blue);
```

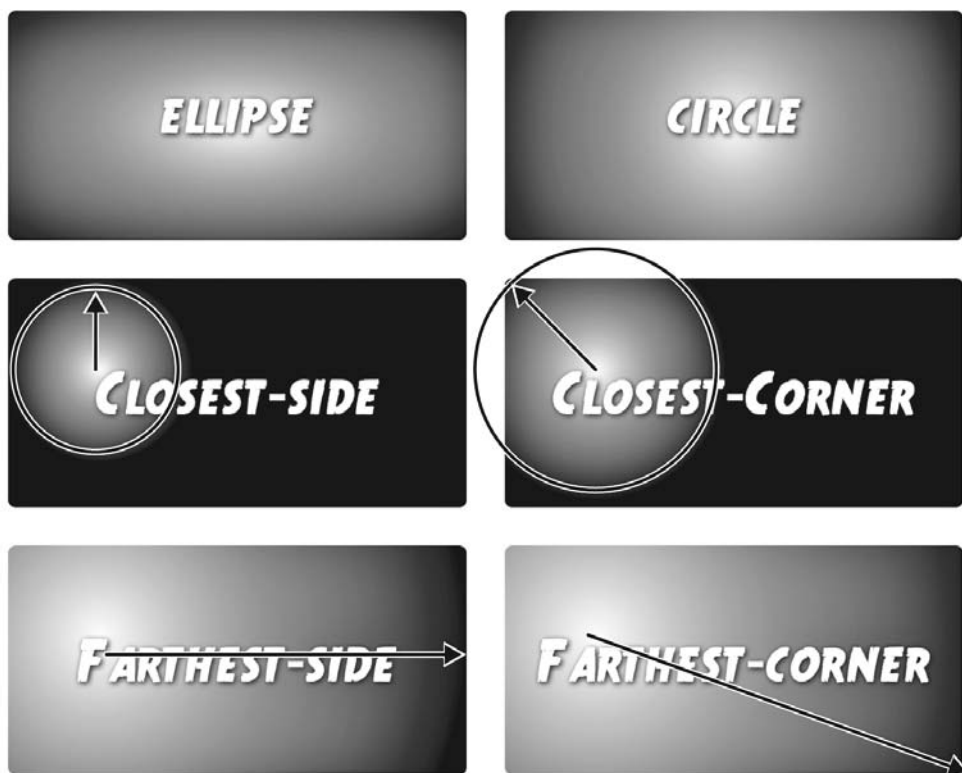


Рис. 8.18. Радиальные градиенты бывают эллиптические (слева сверху) или круговые (справа сверху). Позицией градиента в фоне элемента, а также его размерами можно управлять

Этот код создает эллиптическую форму, помещающуюся в высоту и ширину элемента с центром градиента (где начинается красный цвет) в центре самого элемента.

Можно также создать круговые градиенты, добавив перед указанием цветов ключевое слово `circle`. Например, верхнее левое изображение на рис. 8.19 создано с помощью следующего кода:

```
background-image: radial-gradient(circle, red, blue);
```

Обычно браузер рисует центр радиального градиента в центре элемента, но центр градиента можно позиционировать с помощью таких же ключевых слов позиционирования и значений, которые используются для свойства `background-position`. Например, среднее левое изображение на рис. 8.18 является радиальным градиентом, который начинается с отступом 20 % от левого края элемента и с отступом 40 % от его

верхнего края. Значения позиционирования указываются перед ключевыми словами, определяющими форму и цвета:

```
background-image: radial-gradient(circle at 20% 40%, red, blue);
```

Чтобы указать размер градиента, можно воспользоваться одним из четырех ключевых слов.

- `closest-side` — предписывает браузеру создать градиент, распространяющийся из центра только до ближайшей к центру стороне элемента. Например, в среднем левом изображении на рис. 8.18 ближайшей к центру градиента стороной является верхний край элемента, поэтому радиус окружности простирается от центра до этого края. Тем самым весь градиент остается внутри элемента. Код для создания этого градиента имеет следующий вид:

```
background-image: radial-gradient(closest-side circle at 20% 40%, red, blue);
```

Когда ключевое слово `closest-side` применяется к эллиптическому градиенту, для вычисления значений x и y радиусов эллипса используются обе ближайшие стороны (то есть верхняя или нижняя и левая или правая).

- `closest-corner` — приводит к вычислению ширины градиента из его центра до ближайшего угла элемента (см. рис. 8.18, *справа посередине*). Это может означать выход градиента за пределы элемента. Например, окружность, нарисованная в верхней части среднего правого изображения на рис. 8.18, показывает фактический размер градиента. Часть градиента выходит за границы элемента:

```
background-image: radial-gradient(closest-corner circle at 20% 40%, red, blue);
```

- `farthest-side` — приводит к вычислению радиуса окружности от ее середины до самой дальней стороны элемента. В случае эллиптического градиента это расстояние от центра до любой самой дальней верхней или нижней стороны и до любой самой дальней левой или правой стороны. Код, с помощью которого создано нижнее левое изображение на рис. 8.18, имеет следующий вид:

```
background-image: radial-gradient(farthest-side circle at 20% 40%, red, blue);
```

- `farthest-corner` — приводит к вычислению радиуса окружности от ее центра до самого дальнего угла элемента. Код, с помощью которого создано нижнее правое изображение на рис. 8.18, имеет такой вид:

```
background-image: radial-gradient(farthest-corner circle at 20% 40%, red, blue);
```

Как и в случае линейных градиентов, можно использовать несколько цветовых узлов и указывать для них конкретные позиции. Предположим, нужно создать круговой градиент, составленный из темно-красного, оранжевого и желтого цветов. Нужно, чтобы красный цвет сохранялся на некотором протяжении до его перехода в оранжевый, а затем оранжевый цвет сохранялся на некотором протяжении до перехода в желтый, появляющийся в конце градиента. Чтобы указать позиции появления того или иного цвета, можно воспользоваться процентными значениями:

```
background-image: radial-gradient(circle at 20% 40%, red 20%, orange 80%, yellow);
```

Разумеется, как и в случае линейного градиента, можно воспользоваться допустимыми в языке CSS значениями цвета и переписать этот код следующим образом:

```
background-image: radial-gradient(circle at 20% 40%, rgb(255,0,0) 20%, rgb(255,165,0) 80%, #FFFF00);
```

Как и в случае с линейными градиентами, Internet Explorer 9 и более ранние его версии не поддерживают радиальные градиенты и не будут понимать, о чем идет речь. Нужно добавить резервный фоновый цвет:

```
background-color: red;  
background: radial-gradient(circle at 20% 40%, red 20%, orange 80%, yellow);
```

Повторяющиеся радиальные градиенты

Как и в случае с линейными градиентами, радиальные градиенты можно создать повторяющимися, что подойдет для гипнотизирования посетителей вашего сайта. Чтобы браузер «знал» размер отдельно взятого радиального градиента и смог его повторить, нужно обязательно добавить к различным цветовым узлам процентные, пиксельные значения или значения `em`. Например:

```
background-image: repeating-radial-gradient(circle, red 20px, orange 30px, yellow 40px, red 50px);
```

Учтите, что для создания повторяющихся радиальных градиентов без резких переходов градиент нужно завершать тем же цветом, с которого он начинался (в примере это красный цвет). Тем самым будет обеспечено плавное возвращение цвета к начальному. Если этого не сделать (например, если последним цветом в показанном выше коде сделать желтый), то получится резкая граница там, где заканчивается последний цвет и в повторяющемся градиенте начинается первый цвет.

СОВЕТ

Поскольку браузеры считают линейные и радиальные градиенты фоновым изображением, для них можно использовать и другие свойства фоновых изображений, такие как `background-size`, `background-position` и т. д. Кроме того, в одном стиле можно использовать несколько градиентов, перечислив их через запятую, точно так же, как ранее это делалось с несколькими фоновыми изображениями. В одном элементе можно также смешивать изображения с градиентами.

Практикум: совершенствуем изображения

Фотогалерея — отличный пример привлекательной веб-страницы. Этот практикум воедино собирает и наглядно демонстрирует различные способы форматирования изображений. Мы попрактикуемся в добавлении рамок и подрисовочных подписей, создадим универсальную фотогалерею, легко адаптируемую для просмотра на устройствах с различными размерами экрана, применим рассмотренное ранее свойство `box-shadow` для создания профессиональных визуальных эффектов тени.

Чтобы начать обучение, вы должны иметь в распоряжении файлы с учебным материалом. Для этого нужно загрузить файлы для выполнения заданий практикума, расположенные по адресу github.com/mrightman/css_4e. Перейдите по ссылке

и загрузите ZIP-архив (нажав кнопку **Download ZIP** в правом нижнем углу страницы). Файлы текущего практикума находятся в папке 08.

Заключение изображения в рамку

Мы будем работать над веб-страницей вымышленного сайта CosmoFarmer.com (рис. 8.19). У нас уже есть присоединенная внешняя таблица стилей, применяющая форматирование к веб-странице и обеспечивающая ей простейший дизайн.

1. Откройте файл `image.html` из папки `08\01_image_ex` в браузере.

Изображение занимает на странице слишком много места (см. рис. 8.19, *вверху*). С помощью CSS-кода вы легко можете заключить рисунок в симпатичную рамку и визуальнo выделить его из основного текстового контента (см. рис. 8.19, *внизу*).

2. Откройте файл `styles.css` из папки `01_image_ex` в редакторе HTML-кода.

Он представляет собой внешнюю таблицу стилей, используемую файлом `image.html`. Вы начнете с добавления класса в эту таблицу, а затем примените класс к элементу `img` в HTML-файле.

3. Перейдите к концу файла и наберите следующее:

```
img.figure {  
  
}
```

Селектор `img.figure` воздействует на любой элемент `img`, к которому применен класс `figure`. Вы будете использовать его для выборочного форматирования некоторых изображений (вы могли бы назвать стиль `.figure`, но в таком случае он применялся бы к *любому* элементу с этим классом, а не только к изображениям).

4. Добавим в только что созданный стиль свойства `float` и `margin`:

```
float: right;  
margin: 10px;
```

Свойство `float` смещает изображение к правой стороне веб-страницы, приподнимая текст вверх и создавая обтекание фотографии с левой стороны. Поля обеспечивают небольшие свободные промежутки, отделяющие фото от текста. Теперь добавим границу и небольшие отступы, чтобы изображение больше походило на настоящий фотоснимок.

5. Добавим границу, цвет фона и отступы. Законченный вариант стиля должен иметь следующий вид:

```
img.figure {  
  float: right;  
  margin: 10px;  
  border: 1px solid #666;  
  background-color: #CCC;  
  padding: 10px;  
}
```

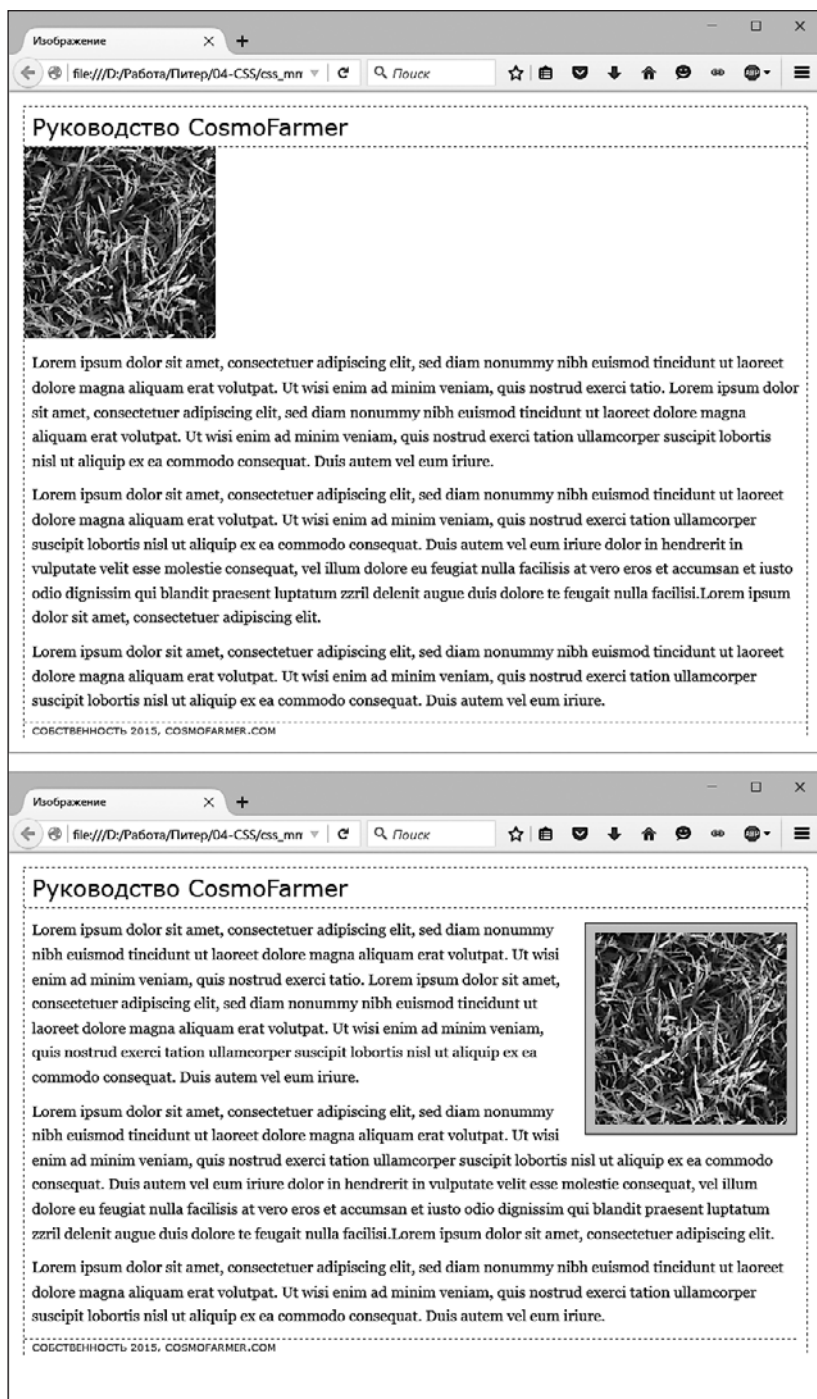



Рис. 8.19. Два варианта веб-страницы: до (вверху) и после (внизу) форматирования с помощью CSS

Если вы сохраните и просмотрите веб-страницу в браузере прямо сейчас, то не будет видно никаких изменений, поскольку класс не возымает эффекта, пока его не применить к элементу.

6. Сохраните и закройте файл `styles.css` и откройте файл `image.html`. Найдите элемент `img` и примените к нему класс `class="figure"` так, чтобы код имел следующий вид:

```

```

Теперь изображение приобретет все свойства форматирования, определенные для класса `.figure`.

7. Просмотрите веб-страницу в браузере. Она должна выглядеть так, как показано на рис. 8.19, *внизу*.

Финальную версию веб-страницы, получившейся в данном практикуме, вы сможете найти в папке `08_finished\01_image_ex`, файл `image.html`.

Изображение может заменить тысячу слов, но нередко требуется добавить к изображению текстовую подпись с подробной информацией об объекте, месте съемки и т. д. При этом вы наверняка захотите, чтобы комментарий был единым целым с изображением, чтобы соседний текст обтекал надпись так же, как и само изображение. Лучший способ — заключить изображение и текст в контейнер, для которого будет определено единое форматирование. В таком случае фотография и связанный с ней текст обрабатываются как цельный блочный элемент. Если вы позже решите изменить их размещение на веб-странице и, возможно, установить обтекание с выравниванием по левому краю страницы — никаких проблем: вам потребуется изменить форматирование для всего элемента-контейнера.

В языке HTML5 именно для решения нашей задачи включены два новых элемента: `figure` предназначен для заключения в него изображения, из которого хотят сделать иллюстрацию. В дополнение к нему может использоваться элемент `figcaption`, который позволяет добавить подрисуночную подпись под изображением. Сначала нужно немного подправить HTML-код.

8. Вернитесь в редактор HTML-кода, к файлу `image.html`. Найдите в коде элемент `img` и удалите значение `class="figure"`, которое было добавлено при выполнении шага 6. Добавьте перед элементом `img` тег `<figure>`.

Этим тегом будет помечено начало контейнера. Теперь будет добавлена подпись и элемент `figure` будет закрыт, чтобы обозначить конец контейнера.

9. После элемента `` добавьте строки, выделенные ниже полужирным шрифтом. HTML-код приобретет такой вид:

```
<figure>

<figcaption>Рисунок 1: Полевица побегообразующая лучше всего растет в открытом
грунте.</figcaption>
</figure>
```

10. Вернитесь к файлу `styles.css`. Прокрутите файл до стиля `img.figure`, созданного ранее, и удалите этот стиль.

Для элемента `figure` будет добавлен новый стиль.

11. Добавьте следующий стиль в файл `styles.css`:

```
figure {  
    float: right;  
    width: 222px;  
    margin: 15px 10px 5px 10px;  
}
```

В предыдущем уроке свойство `float: right` уже использовалось. Свойство `margin` добавит вокруг всех четырех сторон элемента `figure` небольшое пустое пространство. Возникает вопрос: а для чего нужно свойство `width`? Хотя у фотографии и есть установленная ширина (200 пикселей), ее нет у абзаца подписи. Если не указать ширину, абзац заставит элемент `figure` стать шире фотографии. В данном случае нужно, чтобы подпись была такой же по ширине, как фотография и ее рамка.

Значение 222 пикселя получилось в результате небольших математических вычислений общей области, занимаемой фотографией на странице: хотя сама фотография занимает в ширину 200 пикселей, у нее появится (вы добавите на следующем шаге) по 10 пикселей отступов слева и справа. Кроме того, у изображения имеется рамка, занимающая слева и справа по 1 пикселу, что в целом и составляет ширину фотографии от одной границы к другой, то есть 222 пикселя (см. подраздел «Вычисление фактических размеров блочных элементов» раздела «Изменение высоты и ширины» главы 7).

12. Добавьте в файл `styles.css` следующий стиль:

```
figure img {  
    border: 1px solid #666;  
    background-color: #CCC;  
    padding: 10px;  
}
```

Этот селектор потомков воздействует на любой элемент `img`, который находится внутри элемента `figure`. Поскольку здесь используется селектор потомков, класс к элементу `img` добавлять не нужно. Затем добавим стиль к подписи.

13. Добавьте в файл `styles.css` следующий стиль:

```
figcaption {  
    font: bold 1em/normal Verdana, Arial, Helvetica, sans-serif;  
    color: #333;  
    text-align: center;  
}
```

Этот стиль использует некоторые свойства, изученные вами в главе 6 и позволяющие создать подрисовочную подпись, отформатированную полужирным шрифтом Verdana серого цвета с выключкой по центру. Сокращенный способ записи свойства `font` позволяет записать четыре различных свойства в одном объявлении (на одной строке).

Чтобы подрисуночную подпись выделить еще больше, добавим цветной фон и границу.

14. Добавьте в стиль `figcaption` следующие три свойства:

```
figcaption {
  font: bold 1em/normal Verdana, Arial, Helvetica, sans-serif;
  color: #333;
  text-align: center;
  background-image: linear-gradient(to bottom, #e6f3ff, white);
  border: 1px dashed #666;
  padding: 5px;
}
```

Чтобы создать цветную рамку вокруг заголовка, значения свойств `background-image`, `border` и `padding` должны быть сброшены. Мы используем линейный градиент с помощью свойства `background-image`, чтобы создать позади подписи градиент с переходом из светло-голубого в белый цвет.

15. Сохраните файлы `image.html` и `styles.css`, а затем просмотрите файл `image.html` в браузере.

Теперь можно понять одну из причин, по которой разработку дизайна вести проще с использованием внешней таблицы стилей — приходится работать только с одним файлом, а не с двумя, и сохранять только его. Страница должна иметь вид, показанный на рис. 8.20. (Полную версию этой страницы можно найти по адресу `08_finished/01_image_ex`.)

Практикум: создание фотогалереи

Раньше веб-дизайнеры создавали фотогалереи на основе HTML-элемента `table`, помещая изображения в ячейки таблицы, образуемые строками и столбцами. Но сейчас вы можете достигнуть того же эффекта с помощью лаконичного CSS-кода в сочетании с применением гораздо менее объемного HTML-кода.

1. Откройте файл `gallery.html` из папки `08\02_gallery_ex`.

Во-первых, взгляните на HTML-код, который использован для создания фотогалереи. Веб-страница содержит шесть фотографий и подписей к ним. Каждая фотография и подпись к ней хранятся в контейнере `figure`. Сама фотография находится в элементе `img`, а подпись к ней — в элементе `figcaption`.

```
<figure>
  
  <figcaption>Рисунок 6: Фотопокрытия изображают природные склоны. </figcaption>
</figure>
```

Если вы просмотрите страницу сейчас, то увидите серию фотографий, размещенных друг над другом, что выглядит не очень привлекательно. Галерея будет выглядеть лучше, если фотографии будут находиться друг рядом с другом, объединяясь в ряды. Вы можете сделать это с помощью всего нескольких стилей.



Рис. 8.20. Используя в качестве контейнера элемент `figure`, а также применив обтекание слева и некоторое форматирование, вы с легкостью добавили подрисуночную подпись к снимку

2. Откройте файл `styles.css`.

Таблица стилей уже содержит код сброса CSS и несколько стилей для заголовка и остального текста. Для начала вы добавите несколько стилей для фотографий и их заголовков.

3. Добавьте два новых стиля в верхней части файла `styles.css`:

```
figure img {
    border: 1px solid #666;
    background-color: #FFF;
    padding: 4px;
}

figcaption {
    font: 1.1em Arial, Helvetica, sans-serif;
    text-align: center;
    margin: 10px 0 0 0;
}
```

Эти стили добавляют границы-рамки ко всем изображениям галереи и устанавливают шрифт, выравнивание и поля для подрисуночных подписей. В первом стиле для выбора только тех изображений, которые находятся внутри элементов `figure`, используется селектор потомков.

Теперь поместим фотографии друг рядом с другом.

4. Добавьте в таблицу стилей такой код:

```
figure {
    float: left;
    width: 210px;
    margin: 0 10px 10px 10px;
}
```

Он создает такое обтекание, при котором все пары «фотография/подпись» выравниваются по левому краю окна браузера. На самом же деле браузер размещает фотографии на одном уровне, рядом друг с другом, пока не закончится свободное место в строке. Затем браузер переносит следующие изображения на строку ниже, пока не отобразит все, и т. д. Общая ширина *складывается* из ширины самой фотографии плюс отступы и границы-рамки. В данном примере имеем: 200 пикселей на фотографию, 8 пикселей на левый и правый отступы и 2 пиксела на левую и правую границы.

5. Сохраните файл и просмотрите веб-страницу `gallery.html` в браузере. Она должна быть похожа на страницу, представленную на рис. 8.21, *слева*.

Измените ширину окна браузера так, чтобы оно стало уже или шире, и понаблюдайте, как перераспределяются изображения. Вы увидите, что не совсем все правильно. Во второй строке изображений имеется два пустых места, где должны располагаться фотографии. Эта проблема возникает по той причине, что подпись ко второму изображению на первой строке больше по высоте, чем другие на этой же строке. Изображения, которые переносятся на эту подпись, не могут разместиться рядом (об этой путанице, появляющейся при использовании свойства `float`, читайте в разделе «Решение проблем с обтекаемыми элементами» главы 13). Существует простое решение такой проблемы.

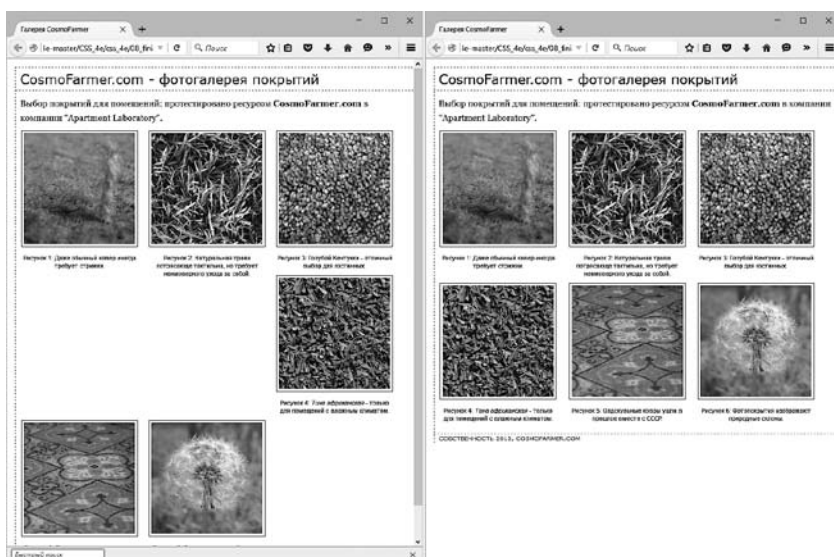


Рис. 8.21. Плавающие элементы, расположенные рядом друг с другом, — один из способов имитации столбцов и строк таблицы

- Вернитесь к файлу `styles.css` в редакторе HTML-кода. Найдите стиль `figure`. Удалите свойство `float:left` и добавьте свойства `display: inline-block;` и `vertical-align: top`. Код должен приобрести такой вид:

```
figure {  
    display: inline-block;  
    vertical-align: top;  
    width: 210px;  
    margin: 0 10px 10px 10px;  
}
```

Свойство `display: inline-block` (см. пункт «Использование свойств `display:inline` и `display:inline-block`» подраздела «Горизонтальные панели навигации» раздела «Создание панелей навигации» главы 9) рассматривает каждую пару «изображение/подпись» в качестве блока (у которого есть высота и ширина), но также и в качестве строчного элемента (то есть блоки могут выстраиваться в ряд). Кроме того, свойство `vertical-align` со значением `top` гарантирует выравнивание каждого элемента `figure` по верхнему краю других элементов `figure`, имеющих-ся в данном ряду.

- Сохраните файл и воспользуйтесь предварительным просмотром веб-страницы в браузере. Взгляните на правую часть рис. 8.21.

Если вы измените размеры окна браузера, то вид фотогалереи тоже изменится. В более широкое окно может вестись четыре или даже пять изображений, но, если размер уменьшить, вы увидите, что на одной строке отображается всего один или два рисунка.

Добавление теней

Наша фотогалерея выглядит хорошо, но ее можно сделать еще выразительнее. Добавление эффектов тени к каждой фотографии придаст веб-странице иллюзию глубины и обеспечит реалистичность трехмерного пространства. Однако прежде, чем запускать графический редактор типа Photoshop, стоит узнать, что добавить эффекты тени к любому изображению веб-страницы можно средствами CSS.

- Откройте в редакторе HTML-кода файл `styles.css`, над которым вы работали в предыдущей части практикума.

Изменение коснется созданного ранее стиля `img`.

- Добавьте в конец стиля `figure img` свойство `box-shadow: 2px 2px 4px rgba(0,0,0,.5);`, чтобы придать ему следующий вид (изменения выделены полужирным шрифтом):

```
figure img {  
    border: 1px solid #666;  
    background-color: #FFF;  
    padding: 4px;  
    box-shadow: 2px 2px 4px rgba(0,0,0,.5);  
}
```


Свойство `box-shadow` было рассмотрено в разделе «Добавление тени» главы 7. Здесь добавляется тень, распространяющаяся на 2 пиксела вправо от изображения и на 2 пиксела ниже его, которая расширяется наружу на 4 пиксела. Использование цветовой модели RGBA (см. подраздел «RGBA» раздела «Форматирование текста цветом» главы 6), можно установить для тени черный цвет с 50%-ной прозрачностью.

3. Сохраните файл и просмотрите веб-страницу в браузере. Она должна иметь такой же вид, как на рис. 8.22.

Каждый рисунок отображается со своей собственной тенью, и вам не пришлось даже запускать Photoshop.

Законченную версию веб-страницы вы сможете найти в папке `08_finished\02_gallery_ex` учебного материала.

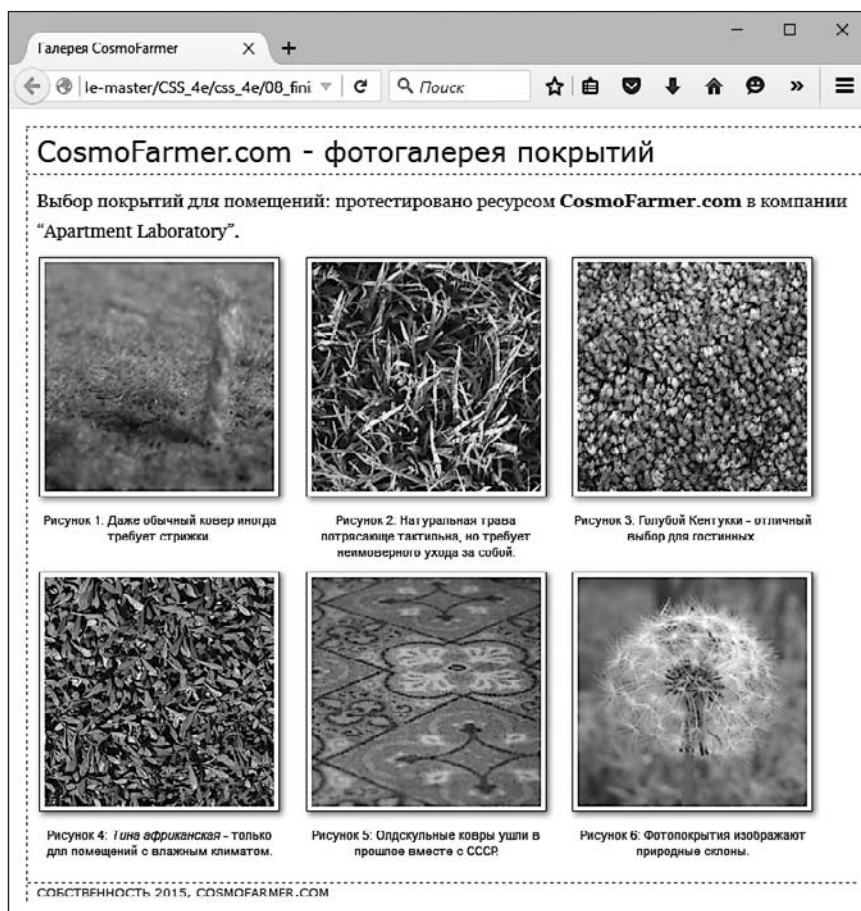


Рис. 8.22. Добавление эффекта тени к фотографиям придает веб-странице объем и улучшает визуальное восприятие любой фотогалереи. Используя таблицы каскадных стилей, можно добавить тень к любому изображению, не прибегая к помощи Photoshop

Практикум: использование фоновых изображений

Свойство `background-image` — секретное оружие современного веб-дизайна. Оно превращает скучный текст в великолепный графический образ (рис. 8.23). Вы можете использовать его для добавления фоновых изображений к любому HTML-элементу, и дизайн, который сумеете создать, ограничен только воображением. Пример эффекта тени, приведенный в предыдущем практикуме, — всего лишь один из способов креативного использования фоновых изображений. Основное их применение — в качестве общего фона веб-страницы, а также для создания собственных маркеров в списках. Эти наиболее распространенные способы применения свойства `background-image` мы и рассмотрим в текущем практикуме.

Добавление на веб-страницу фонового изображения

Что бы это ни было: сложный, замысловатый узор, логотип компании или полно-экранная фотография, — изображения очень часто используются в качестве фоновых рисунков веб-страниц. Фактически это и есть основное применение свойства `background-image`.

1. Откройте файл `bg_images.html` из папки `08\03_bg_ex` в редакторе HTML-кода.

Веб-страница имеет двухколоночный дизайн: она очень проста, содержит лишь немного отформатированного текста на белом фоне (см. рис. 8.23, *вверху*). Для начала добавим линейный градиент. К странице привязана внешняя таблица стилей с базовым форматированием. Вы используете ее, чтобы добавить новый стиль.

2. Откройте файл `styles.css`. Введите следующий код в нижней части страницы:

```
html {  
    background-image: linear-gradient(to bottom, rgb(176,194,213), white 700px);  
}
```

Этот код добавляет линейный градиент шириной 700 пикселей (см. раздел «Использование градиентных фонов» выше), начинающийся голубым цветом вверху и плавно переходящий в белый внизу.

3. Сохраните файл таблицы стилей и просмотрите страницу `bg_images.html` в браузере.

Произошло кое-что странное. Градиент переходит от голубого цвета к белому, а затем повторяет переход снова. Это неожиданно. Чтобы избежать такого эффекта, необходимо сообщить браузеру, что HTML-элемент должен заполнить окно браузера, присвоив параметру `height` значение 100%.

4. В файл `styles.css` добавьте еще одно свойство (выделено полужирным шрифтом):

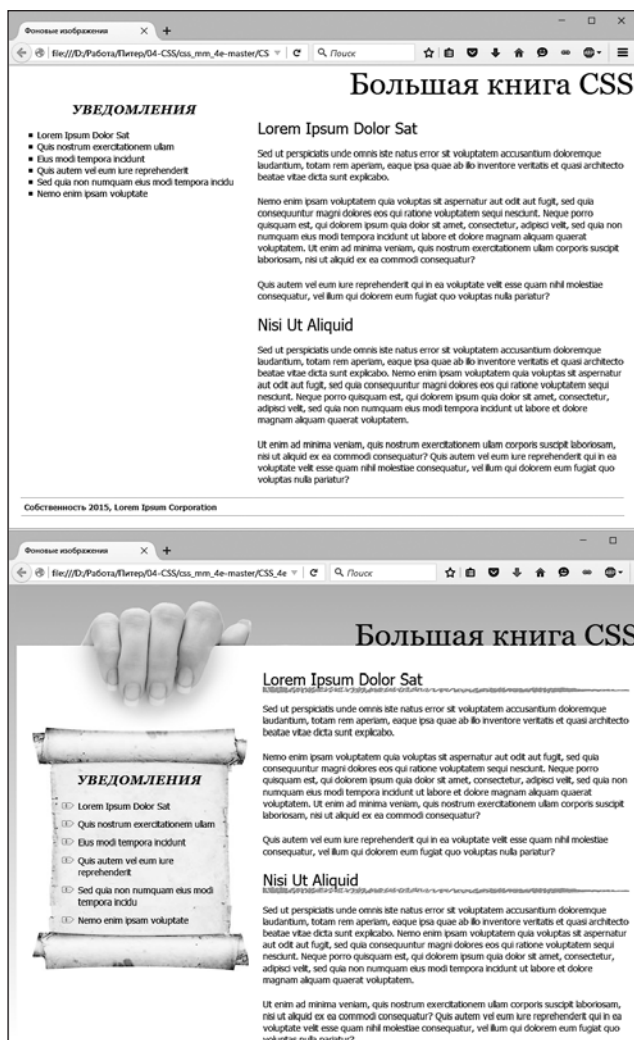


Рис. 8.23. С помощью фоновых изображений вы можете превратить упорядоченную веб-страницу (вверху) в еще более привлекательный сайт (внизу). Поскольку изображения могут добавляться в фон любого элемента на странице, возможности размещения графики на странице практически безграничны

```
html {
  height: 100%;
  background-image: linear-gradient(to bottom, rgb(176,194,213), white 700px);
}
```

Теперь, если вы сохраните файл таблицы каскадных стилей и просмотрите файл `bg_images.html`, градиент, расположенный в фоне, не повторится. Смотрится неплохо, но синий цвет есть и в фоне текста. Вы можете выделить текст, задав для него другой фоновый цвет.

5. Вернитесь к редактору HTML-кода и файлу `style.css`. Добавьте еще один стиль для элемента `div`, в котором находится контент страницы:

```
.wrapper {  
    background-color: #FFF;  
}
```

У этого элемента `div` есть фиксированная ширина, он центрирован на странице и содержит весь ее текст. Стиль задает тексту белый фоновый цвет, но с помощью изображения вы можете сделать его лучше.

6. Отредактируйте стиль, который вы создали в шаге 5, добавив фоновое изображение:

```
.wrapper {  
    background-color: #FFF;  
    background-image: url(images/bg_main.jpg);  
    background-position: left top;  
    background-repeat: no-repeat;  
}
```

Выделенные три строки кода добавляют фоновое изображение в левом верхнем углу контейнера `div`; значение `no-repeat` свойства `background-repeat` означает, что изображение появляется только один раз. Если вы сохраните файл и просмотрите его в браузере, то увидите изображение руки, которая словно держит эту страницу. Очень здорово. Но единственная проблема заключается в том, что текст находится слишком высоко сверху и прикрывает рисунок. Далее вы разместите на странице большой заголовок и левую боковую панель.

7. Добавьте еще два стиля следующим образом:

```
.banner {  
    margin-top: 48px;  
}  
.announcement {  
    margin-top: 115px;  
}
```

Первая строка добавляет небольшие отступы, смещающие вниз баннер, содержащий заголовок, чтобы он соприкасался с белой страницей, а второй стиль смещает вниз левую боковую панель, чтобы освободить достаточно места для изображения с рукой. Страница должна выглядеть так, как показано на рис. 8.24.

Замена границ изображениями

Свойство `border` (см. раздел «Добавление границ» главы 7) — полезный инструмент, но ограниченность предлагаемых языком CSS стилей границ быстро надоедает. Линия, нарисованная карандашом от руки, привлечет внимание посетителей сайта гораздо больше, нежели прямая черная. Можете смело проигнорировать свойство

border и добавить на свой вкус любую линию в виде фонового рисунка, — это очень просто. В этом практикуме мы заменим линии подчеркивания под элементами заголовков h2 собственным рисунком, похожим на рукописную линию.

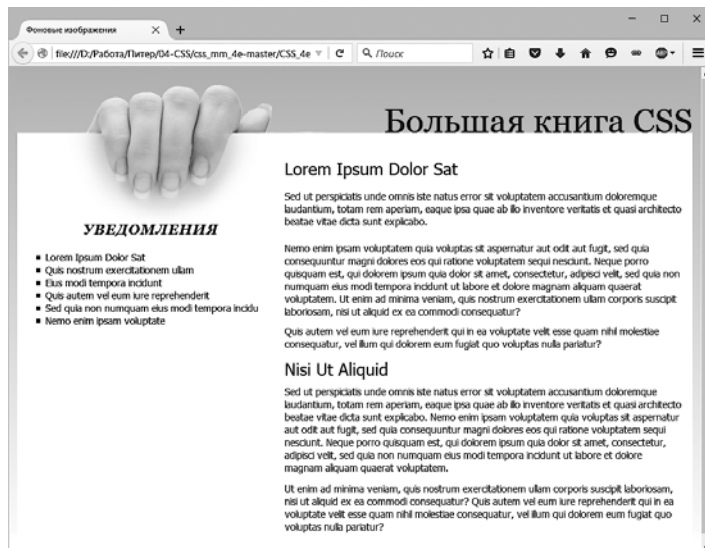


Рис. 8.24. Каскадные таблицы стилей позволяют комбинировать цвет фона и фоновое изображение. К области основного текста применен белый цвет, который выделяет текст на фоне градиента. Кроме того, изображение руки, держащей лист бумаги, вносит в дизайн страницы глубины

1. Вернитесь к файлу `styles.css` в редакторе HTML-кода. Добавьте стиль для элемента `h2` внутри контейнера `div` с классом `main`:

```
.main h2 {
  background-image: url(images/underline.png);
  background-repeat: no-repeat;
}
```

Свойство `background-image` назначает фоновое изображение заголовкам `h2`, находящимся внутри элемента с классом `main`; а значение `no-repeat` гарантирует, что изображение появится только один раз.

Если сейчас вы просмотрите файл в браузере, то увидите, что рисунок подчеркивания расположен не там, где ему положено быть. Он находится *над* заголовками!

2. Добавьте в стиль `.main h2` после свойства `background-repeat` строку со следующим кодом:

```
background-position: left bottom;
```

Мы изменили начальную позицию фонового изображения. Теперь оно выводится, начиная от левого нижнего угла элемента `h2`. Однако при просмотре

веб-страницы вы не заметите серьезных улучшений: подчеркивание сливается с текстом заголовка.

Есть простое решение. Поскольку нижняя координата фонового рисунка расположена у основания блока, образуемого элементом `h2`, необходимо всего лишь увеличить его общую высоту, чтобы сместить линию фонового рисунка немного вниз. Для этого воспользуемся небольшим отступом снизу.

3. Отредактируйте стиль `.main h2` последний раз, чтобы он выглядел таким образом:

```
.main h2 {  
  background-image: url(images/underline.png);  
  background-repeat: no-repeat;  
  background-position: left bottom;  
  padding-bottom: 7px;  
}
```

Как вы помните, отступ представляет собой промежуток между границей (или краем фона) и содержимым. Код также увеличивает суммарную высоту блока — в данном случае добавляется 7 пикселей нижнего отступа. Теперь граница изображения находится в нижней части блока `h2`, но в пустой области, созданной нижним отступом.

4. Сохраните файл и просмотрите страницу `bg_images.html` в браузере.

Теперь все элементы `h2` подчеркнуты рукописной линией. Займемся блоком боковой панели, сделаем его менее угловатым и плоским, а также улучшим вид маркированных списков.

Использование графики для маркированных списков

Стандартный маркер, используемый для списков, представляет собой черное пятнышко, что совсем не впечатляет. Но вы можете создать интересные маркеры с помощью свойства `background-image`, заменив однообразные и скучные значки собственным изображением. Первое, что необходимо сделать, — скрыть стандартные маркеры, которые предваряют элементы-пункты списка.

1. Вернитесь к файлу `styles.css` в редакторе HTML-кода. Добавьте стиль для форматирования списка пунктов левой боковой панели.

```
.announcement li {  
  list-style: none;  
}
```

Маркированный список находится внутри элемента `div` с классом `announcement`, так что этот селектор потомков воздействует только на пункты списка (элементы `li`) внутри этого контейнера `div`. Стиль удаляет маркеры. Теперь добавим наш рисунок.

ПРИМЕЧАНИЕ

Вы можете точно так же применить свойство `list-style: none`; к элементам `ul` или `ol`, чтобы удалить маркеры (или цифры) всех элементов-пунктов списка.

2. Добавьте в стиль `.announcement li` следующие два свойства:

```
background-image: url(images/bullet.png);  
background-repeat: no-repeat;
```

Мы встречались с ними раньше. Одно из них добавляет фоновое изображение, а другое отменяет его повтор, чтобы рисунок отображался однократно.

При просмотре веб-страницы вы увидите, что маркеры накладываются на текст элементов списка и пункты списка отображаются очень тесно друг к другу (рис. 8.25, *вверху*). Небольшие отступы и поля исправят эту проблему.

3. Добавьте в стиль `.announcement li` еще два свойства:

```
padding-left: 25px;  
margin-bottom: 10px;
```

Левый отступ добавляет пустой промежуток, смещая текст в сторону, чтобы отобразить новый значок маркера. Нижнее поле обеспечивает рассредоточение элементов списка, образуя небольшие интервалы (см. рис. 8.25, *посередине*).

Однако остался еще один недостаток. Изображение маркера чуть выступает над текстом строки, и значок выделяется над текстом пунктов списка. Это легко исправить с помощью свойства `background-position`.

4. Завершив этот стиль, добавив свойство `background-position: 0px 4px`. Законченный код стиля должен выглядеть следующим образом:

```
.announcement li {  
  list-style: none;  
  background-image: url(images/bullet.png);  
  background-repeat: no-repeat;  
  background-position: 0 4px;  
  padding-left: 25px;  
  margin-bottom: 10px;  
}
```

Последнее изменение стиля позиционирует значок маркера в крайнюю левую позицию (это 0), отстоящую на 4 пиксела (4px) от верхнего края элемента-пункта списка, что обеспечивает совсем незначительное смещение значка, тем не менее достаточное для того, чтобы маркер выглядел безупречно.

СОВЕТ

Я рекомендую использовать именно свойство `background` вместо `list-style-image` для добавления собственных графических маркеров списков. Оно обеспечивает возможность точного позиционирования значков маркеров.

5. Сохраните файл и просмотрите веб-страницу в браузере.

Теперь пункты списка имеют трехмерные маркеры вместо скучных черных кружков (см. рис. 8.25, *внизу*).

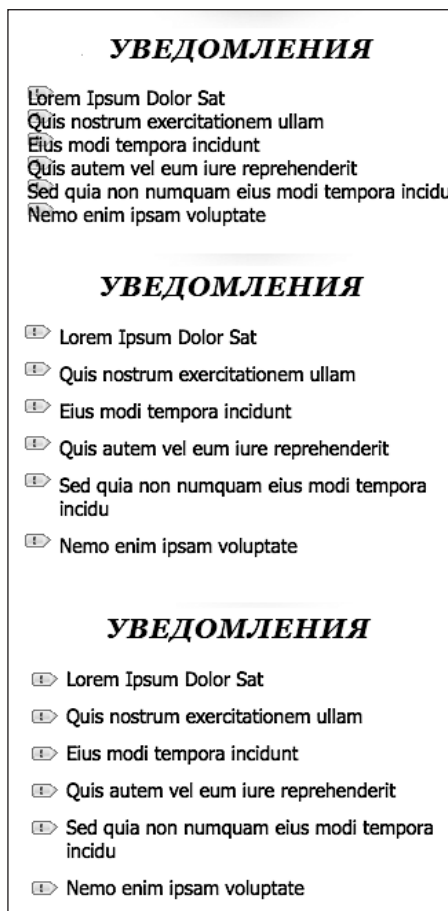


Рис. 8.25. Заменить стандартные маркеры списков своими собственными графическими значками чрезвычайно просто: всего несколько дополнительных шагов обеспечат точное и правильное размещение маркеров и текста пунктов списка

Персонализация боковой панели

На данный момент боковая панель выглядит неплохо. Текст приятно отформатирован, маркеры отлично смотрятся, но сама боковая панель теряется на белом фоне. Добавление фонового изображения позволит ее заметно выделить. Вы можете использовать отдельное изображение в виде свитка, показанное на рис. 8.23, в качестве фона элемента `div`. А чтобы убедиться, что весь текст попадает в изображение, вам следует ограничить количество содержимого, которое вы размещаете на боковой панели. Если будет слишком много текста, он не поместится в пределах изображения (верхнее левое изображение на рис. 8.12), и, наоборот, если будет слишком мало текста, у вас окажется много пустого пространства.

Более гибкий подход позволяет изображению увеличиваться по мере того, как на боковой панели увеличивается количество текста (см. рис. 8.13, *справа внизу*).

Хорошо, что эту маленькую хитрость не так сложно реализовать — вам понадобится три разных изображения и три разных стиля.

1. Вернитесь к редактору HTML-кода и файлу `styles.css`. Перейдите к стилю `.announcement` (который вы создали в шаге 6 первого раздела этого практикума) и добавьте одно свойство:

```
.announcement {  
    background: url(images/scroll_top.jpg) no-repeat center top,  
                url(images/scroll_bottom.jpg) no-repeat center bottom,  
                url(images/scroll_middle.jpg) repeat-y center top;  
    margin-top: 115px;  
}
```

Да, только одно свойство (`background`), но оно содержит три изображения. Очередность отображения картинок на странице зависит от того, как они перечислены в списке. В данном случае первое изображение выводится в верхней части свитка по центру и не повторяется (благодаря параметру `no-repeat`). Второе изображение находится в нижней части свитка. Оно также повторяется один раз, но в конце элемента `div`. Наконец, средняя часть свитка (изображение `scroll_middle.jpg`) появится под двумя другими изображениями (поскольку является последним в списке) и будет повторяться по оси *Y* (по вертикали), поэтому, если элемент `div` поднимется выше, третье изображение заполнит освободившуюся область.

Если вы просмотрите страницу, то заметите несколько проблем. Во-первых, текст отображается в верхней и нижней областях свитка, которые свернуты. Небольшой отступ исправит это.

2. Обновите стиль `.announcement`, добавив верхний и нижний отступы. Внесите изменение, выделенное полужирным шрифтом:

```
.announcement {  
    background: url(images/scroll_top.jpg) no-repeat center top,  
                url(images/scroll_bottom.jpg) no-repeat center bottom,  
                url(images/scroll_middle.jpg) repeat-y center top;  
    padding: 70px 0 60px 0;  
    margin-top: 115px;  
}
```

Еще одна проблема проявляется в том, что маркированный список выдается как с левой, так и с правой стороны изображения боковой панели. Сделаем так, чтобы он соответствовал этому изображению, добавив правое и левое поля.

3. Перейдите к стилю `.announcement li`, созданному ранее, и добавьте два свойства в конце:

```
.announcement li {  
    list-style: none;  
    background-image: url(images/bullet.png);  
    background-repeat: no-repeat;  
    background-position: 0 4px;  
    padding-left: 25px;
```

```
margin-bottom: 10px;  
margin-left: 30px;  
margin-right: 40px;  
}
```

Эти свойства перемещают левый и правый края каждого пункта маркированного списка на достаточное расстояние, чтобы очистить края фонового изображения.

4. Сохраните файл и просмотрите его в браузере.

Страница должна выглядеть как правое изображение на рис. 8.23. С помощью нескольких изображений и языка CSS очень легко придать жизнь скучным страницам.