

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ**

**КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ**

**Лабораторна робота № 4**

**Основи технології CSS**

З навчальної дисципліни

WEB-дизайн

Спеціальність – 121 Інженерія програмного забезпечення

Галузь знань – 12 Інформаційні технології

Освітньо–кваліфікаційний рівень – бакалавр

Київ–2017

**Мета роботи:** освоїти основи технології CSS: додавати каскадних таблиць стилів в веб-документ, навчитися працювати з CSS-селекторами, вміти налаштовувати кольорове оформлення документа, управляти текстом, знати режими відображення HTML-елементів.

В этом обучающем уроке мы исследуем элементы блочной модели CSS, по тренируемся в настройке параметров интервалов, промежутков объектов вебстраниц, применим к элементам красочные границы-рамки, поуправляем размерами и обтеканием элементов веб-страниц.

## Управление полями страницы

Начнем с простого HTML-файла, содержащего внутреннюю таблицу стилей со сбросом стандартных настроек стилей CSS. Пока на странице нет ничего интересного (рис. 4.1).

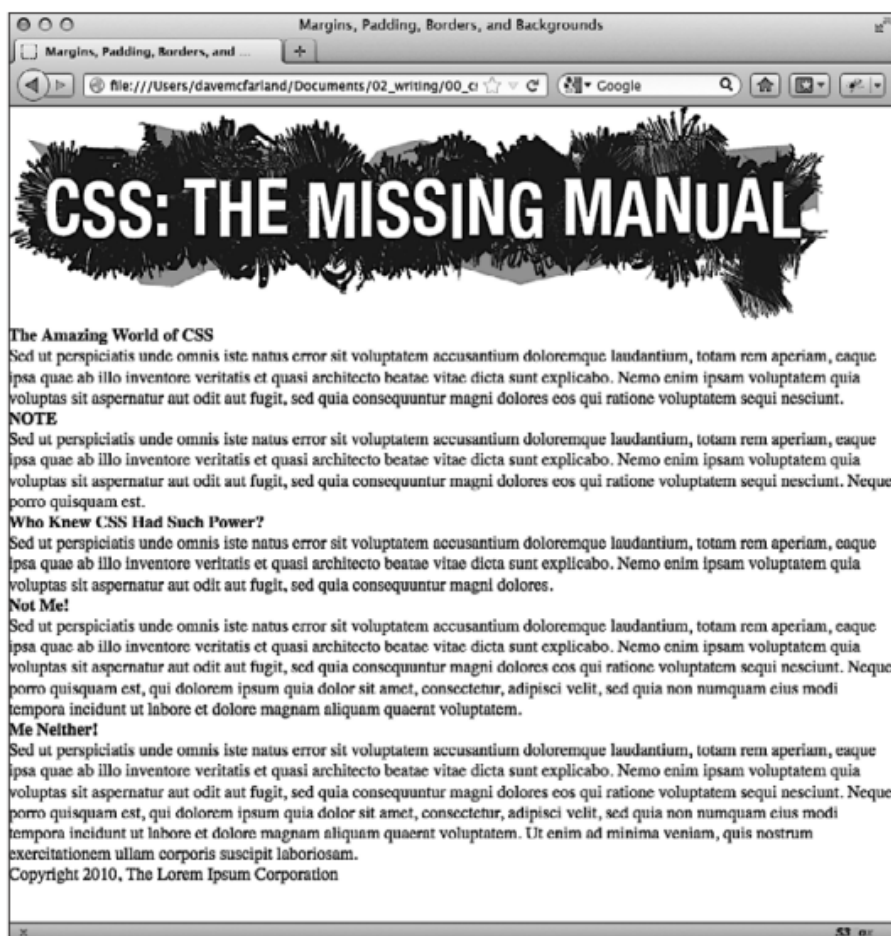


Рис. 4.1. Эта страница представляет собой основу HTML и содержит единственный стиль, устраняющий множество встроенных настроек форматирования браузера. Она будет выглядеть намного лучше после изменения с использованием блочной модели.

1. Откройте в HTML-редакторе страницу sidebar.html из папки 07.

Здесь уже есть внутренняя таблица стилей, содержащая набор стилей, который рассматривался в разделе «Начинаем с чистого листа» гл. 5. Этот набор стилей в основном удаляет все поля, отступы, установленные значения для размера шрифта, начертания шрифта из наиболее значимых элементов уровня блока и устраняет множество кросс-браузерных проблем с отображением элементов, с которыми вы могли столкнуться из-за этих свойств.

Как минимум вы должны всегда включать этот набор стилей во все создаваемые таблицы стилей. Наверное, наиболее важными свойствами в нем являются margin и padding, установленные в первом стиле. Существует достаточно кросс-браузерных странностей, связанных с этими двумя свойствами, поэтому вы должны всегда обнулять их и начинать с чистого листа. Сначала мы определим что-нибудь простое, например цвет фона.

2. Во внутренней таблице стилей щелкните кнопкой мыши сразу за комментарием CSS /\* end reset styles \*/ и добавьте стиль селектора тега:

```
html {  
    background-color: rgb(253,248,171);  
}
```

Этот стиль устанавливает светло-желтый фон для страницы. Если вы хотите задать цвет для фона веб-страницы, можно добавить свойство background-color либо к тегу <html>, либо к тегу <body>. Далее мы добавим поля, границы и другие свойства тега <body>.

## ПРИМЕЧАНИЕ

*Возможно, вы привыкли вместо RGB-модели задания цвета использовать шестнадцатеричную модель (в которой цвет представляется, например, так: #FDF8AB). Чтобы осуществлять преобразования между этими двумя моделями, можно воспользоваться онлайн-конвертером [www.colorhexa.com/](http://www.colorhexa.com/). Но использование модели RGB предпочтительнее, поскольку RGBA-формат этой модели, имеющий дополнительный параметр прозрачности, настолько полезен, что лучше остановиться на одной цветовой модели (RGB), а не смешивать две (RGB и шестнадцатеричную).*

3. Добавьте другой стиль во внутреннюю таблицу стилей.

```
body {  
    background-color: rgb(255,255,255);  
    border: 3px solid rgb(75,75,75);  
}
```

Этот стиль добавляет белый цвет фона для тега `<body>` и темно-серую трехпиксельную границу. Поскольку тег `<body>` расположен внутри тега `<html>`, браузер считает, что он находится «поверх» тега `<html>` и белый фон покроет фон желтого цвета, установленный в предыдущем шаге. Далее мы зададим ширину для тега тела и настроим его отступы и поля.

## СОВЕТ

*Обычно, если вы добавляете свойство `background-color` к тегу `<body>`, цвет заполняет все окно браузера. Однако если вы также добавите цвет фона для тега `<html>`, фон `<body>` будет заполнять только область с содержимым. Чтобы увидеть это в действии, просмотрите веб-страницу после шага 2, затем удалите стиль для тега `<html>` и просмотрите страницу снова. Странная, но полезная особенность CSS.*

4. Измените стиль тела (`<body>`), который вы только что создали, добавив пять новых свойств (изменения выделены полужирным шрифтом):

```
body {  
    background-color: rgb(255,255,255);  
    border: 3px solid rgb(75,75,75);  
    width: 760px;  
    margin-top: 20px;  
    margin-left: auto;  
    margin-right: auto;  
    padding: 15px;  
}
```

Свойство `width` ограничивает тело страницы 760 пикселями по ширине: если окно браузера посетителя окажется шире чем 760 пикселей, он увидит фоновый цвет стиля `<html>` и блок шириной 760 пикселей с белым фоном тега `<body>`.

Свойство `margin-top` добавляет 20 пикселей пространства от верхнего края окна браузера, «отталкивая» содержимое тега `<body>` чуть-чуть вниз, а левое и правое поля центрируют его, размещая посередине окна браузера. Значение `auto` —

это просто еще один способ сказать браузеру: «Разбирайся в этом сам», и поскольку данное значение применяется как к левому, так и к правому полю, браузер создает одинаковые промежутки слева и справа.

## ПРИМЕЧАНИЕ

*Вы можете также использовать сокращенное свойство `margin`, чтобы сжать эти три строки, устанавливающие настройки для полей, в одну:*  
`margin: 20px auto 0 auto;`

Наконец, чтобы предотвратить прикосновение содержимого тега `<body>` к линии и границы, отступ шириной 15 пикселей добавлен к внутренней части содержимого с помощью свойства `padding`. Другими словами, для изображения и текста был задан отступ 15 пикселей от всех четырех краев. Затем мы добавим свечение вокруг блока, воспользовавшись свойством `box-shadow`.

5.

Отредактируйте только что созданный стиль элемента `body`, добавив к нему еще одно свойство после `border`, но перед `width` (изменения выделены полужирным шрифтом):

```
body {  
  background-color: rgb(255,255,255);  
  border: 3px solid rgb(75,75,75);  
  box-shadow: 0 0 15px 5px rgba(44,82,100,.75);  
  width: 760px;  
  margin-top: 20px;  
  margin-left: auto;  
  margin-right: auto;  
  padding: 15px;  
}
```

Этот стиль добавляет блоку свечение путем создания 15-пиксельной тени помещенной непосредственно за блоком (то, что параметры тени начинаются с 0 0, означает, что она не имеет смещения вправо-влево или вверх-вниз, и представляет собой просто фон). Значение 5px определяет расширение тени, выводя ее на 5 пикселей дальше всех четырех углов. И наконец, значение `rgba` устанавливает темно-синий цвет с показателем непрозрачности 75 % (то есть через него можно видеть желтый фон).

В вашей таблице стилей уже много всего, поэтому пришло время проверить, как выглядит страница.

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

Вы должны увидеть белый блок с изображением, кусок текста и серый контур с синеватым свечением, плавающие на желтом фоне (рис. 4.2). Теперь следует уделить внимание тексту, чем мы и займемся далее.

## Настройка интервалов между тегами

Поскольку стили, сбрасывающие стандартные настройки CSS, «оголили» текст на странице, вам необходимо создать новые стили, которые преобразили бы заголовки и абзацы. Начнем с тега `<h1>` вверху страницы.

1. Вернитесь к файлу `sidebar.html` в HTML-

редакторе. Установите курсор после закрывающей скобки селектора `body`, нажмите Enter и добавьте следующий стиль:

```
h1 {  
  font-size: 2.75em;  
  font-family: Georgia, "Times New Roman", Times, serif;  
  font-weight: normal;
```

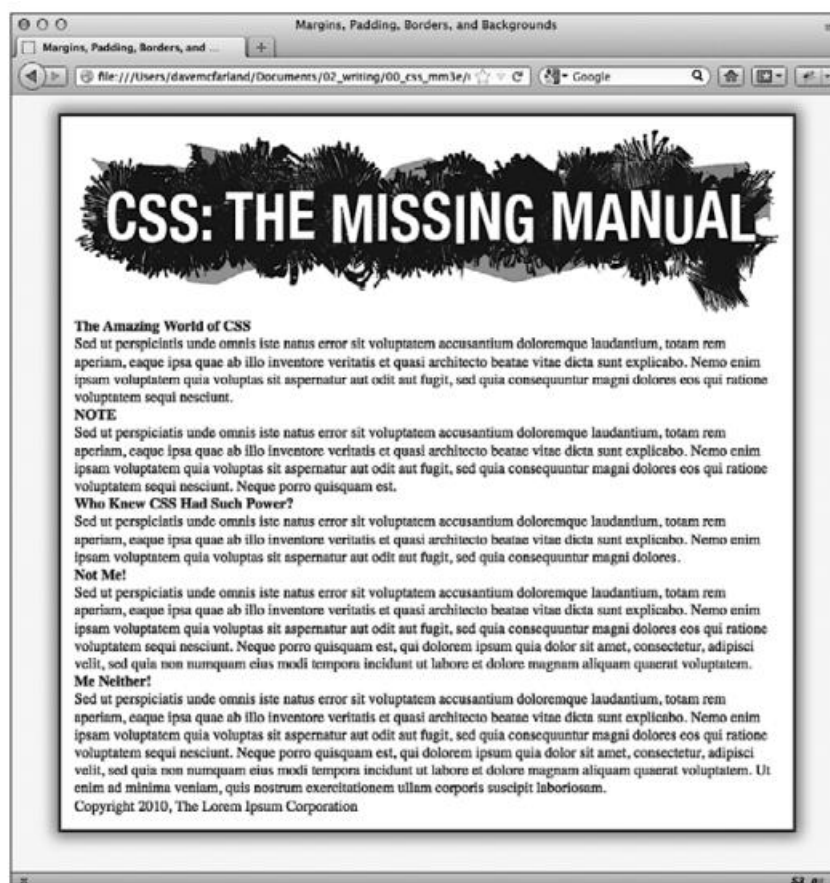


Рис. 4.2. Устанавливая значение auto для левого и правого полей любого элемента, имеющего заданную ширину, вы центрируете этот элемент

```
text-align: center;
letter-spacing: 1px;
color: rgb(133,161,16);
text-transform: uppercase;
}
```

Этот стиль использует множество свойств для форматирования текста, которые мы обсуждали в предыдущей главе. Верхний заголовок имеет высоту 2.75 em (44 пиксела в большинстве браузеров) и набран прописными буквами. К нему применяется шрифт Georgia зеленого цвета, есть небольшой промежуток между буквами. Свойство text-align обеспечивает центровку текста в середине блока. Очень интересным получается добавление фонового цвета для выделения заголовка.

## СОВЕТ

*Сохраняйте страницу и просматривайте ее в браузере после каждого шага из этого примера. Таким образом вы получите хорошее понимание того, как приведенные здесь свойства CSS воздействуют на элементы, которые они формируют.*

2.

Добавьте одно новое свойство к стилевому тегу `h1`, чтобы он выглядел следующим образом (изменения выделены полужирным шрифтом):

```
h1 {  
    font-size: 2.75em;  
    font-family: Georgia, "Times New Roman", Times, serif;  
    font-weight: normal;  
    text-align: center;  
    letter-spacing: 1px;  
    color: rgb(133,161,16);  
    text-transform: uppercase;  
    background-color: rgb(226,235,180);  
}
```

Если вы просмотрите страницу сейчас, то увидите, что заголовок имеет светло-зеленый фон. Когда фон применяется к блочным элементам, таким как заголовок, он заполняет все доступное горизонтальное пространство (другими словами, цвет не только появляется за текстом *The Amazing World of CSS*, но и простирается вплоть до правого края окна).

Текст заголовка немного стесненный —

буква *T*, с которой он начинается, касается края фона. С помощью небольших отступов вы можете исправить это.

3.

Добавьте другое свойство к стилю тега `h1`, чтобы он выглядел следующим образом (изменения выделены полужирным шрифтом):

```
h1 {  
    font-size: 2.75em;  
    font-family: Georgia, "Times New Roman", Times, serif;  
    font-weight: normal;  
    text-align: center;  
    letter-spacing: 1px;  
    color: rgb(133,161,16);  
    text-transform: uppercase;
```

```
background-color: rgb(226,235,180);  
padding: 5px 15px 2px 15px;  
}
```

Сокращенное свойство `padding` представляет собой быстрый способ добавить отступы вокруг всех четырех сторон содержимого — в данном случае отступ шириной 5 пикселей добавляется над текстом, 15 пикселей — справа, 2 пикселя — внизу и 15 пикселей — слева.

Есть еще одна проблема с заголовком: из-за отступов, которые добавлены к тегу `<body>` (см. шаг 4 предыдущего задания), заголовок отодвинут на 15 пикселей от

левого и правого краев зеленого контура, окружающего содержимое. Заголовок станет выглядеть лучше, если будет касаться этого контура. Это не проблема: на помощь приходят отрицательные поля.

4. Добавьте одно последнее свойство к стилю тега `h1`, чтобы он выглядел следующим образом (изменения выделены полужирным шрифтом):

```
h1 {  
    font-size: 2.75em;  
    font-family: Georgia, "Times New Roman", Times, serif;  
    font-weight: normal;  
    text-align: center;  
    letter-spacing: 1px;  
    color: rgb(133,161,16);  
    text-transform: uppercase;  
    background-color: rgb(226,235,180);  
    padding: 5px 15px 2px 15px;  
    margin: 0 -15px 20px -15px;  
}
```

Здесь сокращенное свойство `margin` устанавливает 0 пикселей для верхнего поля, минус 15 пикселей для правого, 20 пикселей для нижнего и минус 15 пикселей для левого. Нижнее поле просто добавляет немного пространства между заголовком и абзацем, который идет за ним. Следующий прием заключается в использовании отрицательных значений для левого и правого полей. Как отмечалось ранее, у нас есть возможность назначить отрицательные поля для какого-либо элемента. Это свойство «тянет» элемент по направлению поля — в данном случае заголовок продлевается на 15 пикселей влево и 15 пикселей вправо, расширяясь и вытягиваясь над отступами тега `<body>`.



5.

Теперь немного отформатируем тег `<h2>`. Добавьте следующий стиль после стиля для `h1`:

```
h2 {  
  
    font-size: 1.5em;  
  
    font-family: "Arial Narrow", Arial, Helvetica, sans-serif;  
  
    color: rgb(249,107,24);  
  
    border-top: 2px dotted rgb(141,165,22);  
  
    border-bottom: 2px dotted rgb(141,165,22);  
  
    padding-top: 5px;  
  
    padding-bottom: 5px;  
  
    margin: 15px 0 5px 0;  
  
}
```

Этот стиль добавляет базовое форматирование текста и пунктирные границы сверху и снизу заголовка. Чтобы добавить немного пространства между текстом заголовка и строками, используются небольшие отступы сверху и снизу. Наконец, свойство `margin` добавляет поля шириной 15 пикселей над заголовком и 5 пикселей под ним.

6. Сохраните файл и просмотрите страницу в браузере.

Заголовки выглядят хорошо (рис. 4.3). Далее создадим боковую панель на правой стороне страницы.

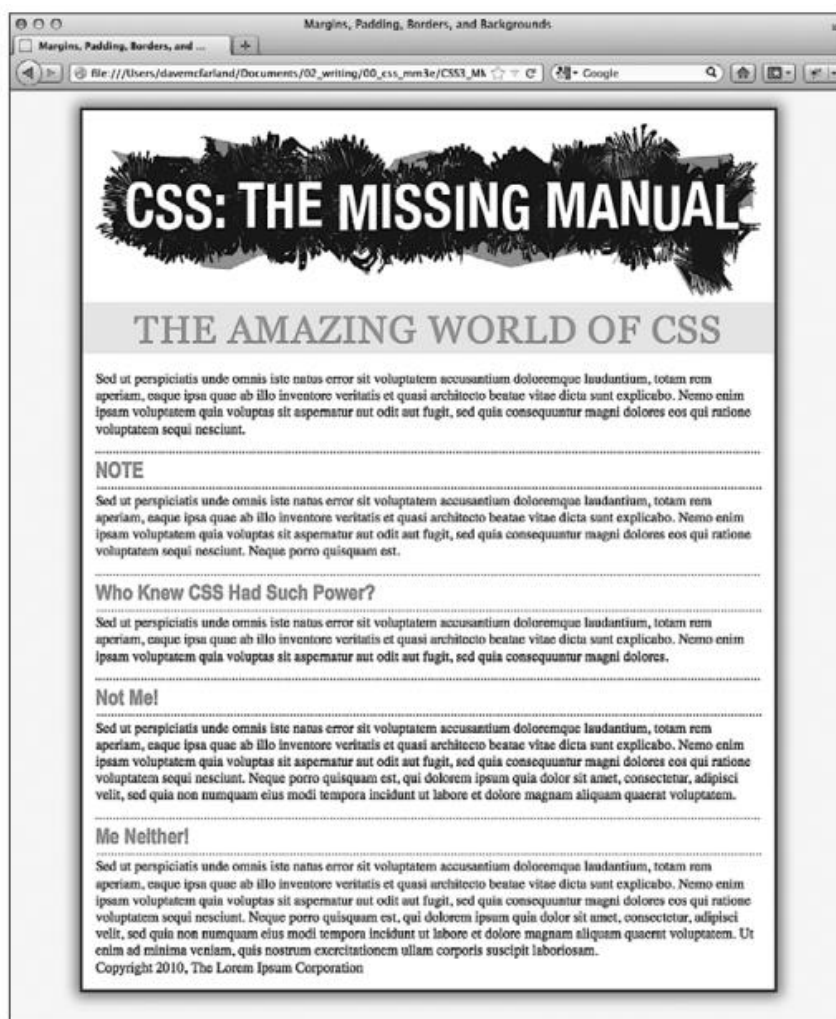


Рис. 4.3. С помощью нескольких стилей вы можете изменить фоновый цвет веб-страницы, добавить поля, регулировать интервалы между заголовками и абзацами

## Создание боковой панели

Боковые панели —

обычные элементы, применяемые в большинстве печатных изданий: в журналах, книгах, газетах. Они отделяют и подсвечивают небольшие блоки информации, например перечень ресурсов, тематичный анекдот. Но, чтобы бо-

ковые панели были достаточно эффективными и полезными, они не должны прерывать поток основного содержимого. Даже название «боковая панель» говорит

о том, что этот блок должен быть расположен обособленно и примыкать к краю веб-страницы, что можно легко сделать средствами CSS.

1. Вернитесь к файлу sidebar.html в HTML-редакторе.

Сначала нужно отделить область веб-страницы, составляющую боковую панель.

Для этого прекрасно подходит тег `<div>`. С его помощью можно заключить любой

объем HTML-кода в свой собственный отдельный блок.

2. Прокрутите страницу вниз к коду HTML и щелкните кнопкой мыши перед первым тегом `<h2>` (с заголовком NOTE). Наберите `<div class="sidebar">` и нажмите Enter.

Этот HTML-код отмечает начало блока боковой панели и применяет к нему стилевой класс. Мы создадим сам стилевой класс `.sidebar`, но сначала нужно определить завершение блока боковой панели, закрыв `<div>`.

3.

Перейдите к закрывающему тегу `</p>`, который следует сразу за тегом `<h2>` (это тот `</p>`, который появляется прямо перед `<h2>Who Knew CSS Had Such Power?</h2>`).

Нажмите после него Enter и введите `</div>`.

Мы только что заключили заголовок и маркированный список в тег `<div>`. Теперь создадим для него стиль.

4.

Прокрутите страницу вверх к таблице стилей и добавьте после созданного ранее стиля для `<h2>` следующий код:

```
.sidebar {  
    width: 30%;  
    float: right;  
    margin: 10px;  
}
```

Этот стиль устанавливает ширину области содержимого (в которой отображается текст) равной 30 %. Вам не нужно использовать абсолютное значение, например пиксели, для ширины. В этом случае ширина боковой панели составляет 30 % от ширины ее контейнера. Свойство `float` перемещает боковую панель в правую часть блока, а свойство `margin` добавляет 10 пикселей пространства вокруг панели.

Если вы просмотрите веб-страницу в браузере, то увидите, что форма и положение блока боковой панели определены, но есть одна проблема: границы тегов `<h2>` отображаются под самим блоком. Несмотря на то что плавающая боковая панель смещает текст заголовков, границы остаются на том же месте. Они выступают в качестве фона плавающей боковой панели. Один из способов решить эту проблему — просто добавить фоновый цвет для боковой панели, чтобы не видеть границ `h2` (но есть и другой способ, который вы будете использовать в шаге 8).

5. Добавьте другое свойство к стилю `.sidebar`, чтобы он выглядел следующим образом (изменения выделены полужирным шрифтом):

```
.sidebar {  
    width: 30%;  
    float: right;
```

```
margin: 10px;
background-color: rgb(250,235,199);
padding: 10px 20px;
}
```

Это свойство добавляет светло-оранжевый цвет к боковой панели и оттесняет текст от границ боковой панели, чтобы он не касался границ, которые вы собираетесь добавить.

6. Добавьте еще два свойства к стилю `.sidebar`, чтобы он выглядел следующим образом (изменения выделены полужирным шрифтом):

```
.sidebar {
    width: 30%;
    float: right;
    margin: 10px;
    background-color: rgb(250,235,199);
    padding: 10px 20px;
    border: 1px dotted rgb(252,101,18);
    border-top: 20px solid rgb(252,101,18);
}
```

Это пример удобной методики, описанной ранее. Если вы хотите, чтобы большая часть границ вокруг элемента была одинаковой, можно сначала определить границу для всех четырех краев — в данном случае однопиксельную пунктирную оранжевую линию вокруг всей боковой панели. Затем можно применить новые свойства границы для отдельных краев, которые вы хотите изменить, — в данном примере верхняя граница будет сплошной и будет иметь высоту 20 пикселей. Этот способ позволяет использовать всего две строки кода CSS, а не четыре (`border-top`, `border-bottom`, `border-left` и `border-right`).

Затем мы добавим скругленные углы и тени, чтобы выделить эту боковую панель по-настоящему.

7. Добавьте еще два свойства к стилю `.sidebar`, придав ему следующий вид (изменения выделены полужирным шрифтом):

```
.sidebar {
    width: 30%;
    float: right;
    margin: 10px;
    background-color: rgb(250,235,199);
```

```
padding: 10px 20px;
border: 1px dotted rgb(252,101,18);
border-top: 20px solid rgb(252,101,18);
border-radius: 10px;
box-shadow: 5px 5px 10px rgba(0,0,0,.5);
}
```

Свойство `border-radius` позволяет создать скругленные углы. В данном случае 10-пиксельная установка предоставляет заметное скругление. Свойство `box-shadow` добавляет тень, отбрасываемую вниз и вправо от блока, придавая ему вид парящего над страницей. Теперь мы близки к завершению работы.

Заголовок внутри боковой панели выглядит не совсем так, как должен. К нему применяются те же свойства, что и к другим тегам `<h2>` (из-за стиля тега `h2`, который мы создали в шаге 4). Границы отвлекают внимание, а верхнее поле слишком отталкивает заголовок вниз от верхней части боковой панели. К счастью, вы можете использовать селектор потомка, чтобы переопределить эти свойства.

8.

После стиля `.sidebar` во внутренней таблице стилей добавьте селектор потомка:

```
.sidebar h2 {
  border: none;
  margin-top: 0;
  padding: 0;
}
```

Из-за `.sidebar` этот стиль является доминирующим, то есть имеет большую значимость по сравнению с обычным стилем `h2`. Он стирает границу, полученную от оригинального стиля тега `<h2>`, вместе с верхним полем и всеми отступами. Тем не менее, поскольку в этом стиле не определены размер, цвет и начертание шрифта, эти свойства по-прежнему передаются от стиля `h2` — каскадность в действии!

Страница стала хорошо выглядеть, но границы тегов `<h2>` все еще появляются под боковой панелью. На это не очень приятно смотреть, но все можно легко исправить.

9. Найдите стиль `h2` и добавьте свойство `overflow`:

```
h2 {
  font-size: 1.5em;
  font-family: "Arial Narrow", Arial, Helvetica, sans-serif;
  color: rgb(249,107,24);
}
```

```
border-top: 2px dotted rgb(141,165,22);
border-bottom: 2px dotted rgb(141,165,22);
padding-top: 5px;
padding-bottom: 5px;
margin: 15px 0 5px 0;
overflow: hidden;
}
```

Установив для свойства `overflow` значение `hidden`, вы спрячете границы, которые проходят за пределами текста заголовка и под плавающим элементом.

10. Сохраните файл и просмотрите веб-страницу в браузере. Она должна иметь вид, представленный на рис. 4.4.

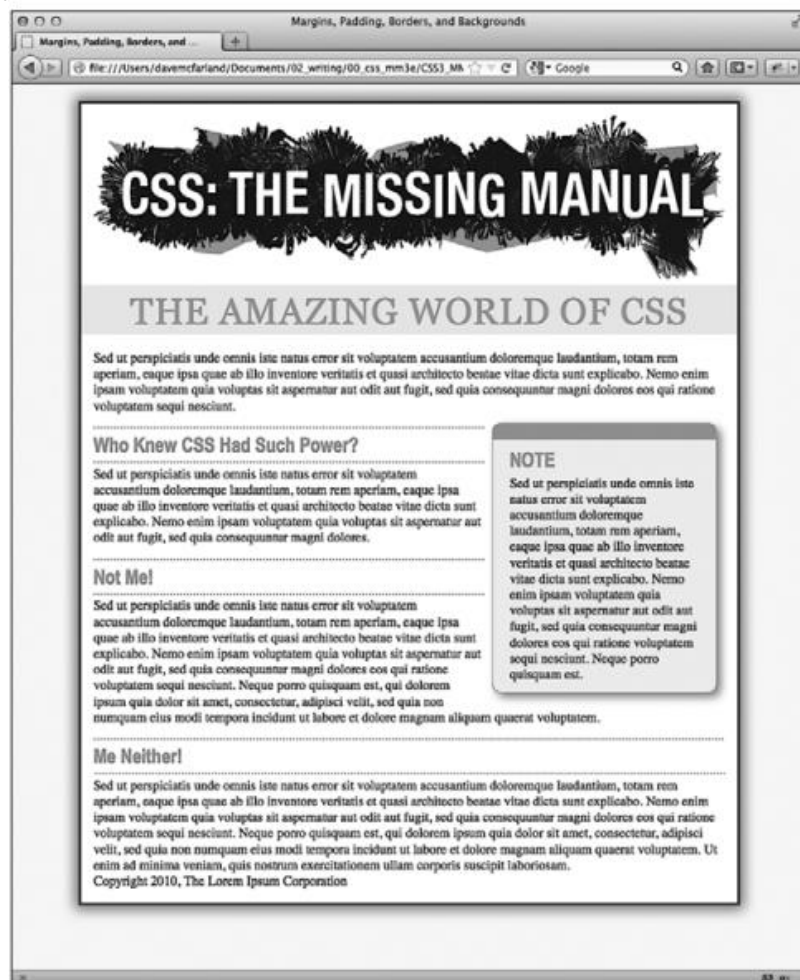


Рис. 4.4. Небольшой набор CSS-стилей придает непривлекательному HTML-коду элегантный вид

## Завдання для самостійної роботи

### Завдання:

1. Создайте для тега `<p>` стиль, который смог бы украсить абзац: попробуйте добавить поля, указать цвет шрифта и т. д. Затем создайте класс для стилизации примечания с информацией об авторском праве, которое должно отображаться в нижней части страницы `sidebar.html` (например, с именем `.copyright`). Добавьте в этот стиль верхнюю границу (над текстом примечания), измените цвет текста, уменьшите размер шрифта и измените регистр букв на прописные (используйте для этого свойство `text-transform`, описанное в разделе «Форматирование символов и слов» гл. 6). После создания стиля добавьте соответствующий атрибут класса к тегу `<p>` в HTML-коде.
2. Оформити звіт.