

## Создание внешней таблицы стилей

Поскольку во внутренних таблицах все стили сгруппированы в начале веб-страницы, создавать и редактировать их намного проще и удобнее, чем встроенные стили, с которыми вы имели дело до этого. Кроме того, внутренние таблицы стилей позволяют форматировать любое количество экземпляров элементов веб-страницы одновременно (как в примере с элементом `h1`), создав один простой стиль (правило). Внешние таблицы стилей не только наследуют преимущества внутренних таблиц, но и имеют дополнительные плюсы: в них можно хранить все стили для всех страниц сайта. Редактирование одного стиля во внешней таблице обновляет стиль целого сайта. В этом разделе вы возьмете стили, созданные в предыдущем уроке, и поместите их во внешнюю таблицу стилей.

1. В редакторе HTML-кода создайте новый файл и сохраните его под именем `styles.css` в той же самой папке, где находится веб-страница, над которой вы сейчас работаете.

Файлы внешних таблиц стилей должны заканчиваться расширением `.css`. Имя файла `styles.css` указывает на то, что стили, содержащиеся в файле, используются глобально для всего сайта (вы, конечно, можете использовать любое понравившееся имя).

Приступим к добавлению нового стиля к таблице стилей.

2. Наберите в файле `styles.css` следующее правило:

```
html {  
    padding-top: 25px;  
    background-image: url(images/bg_page.png);  
}
```

Это правило касается `html` — элемента, окружающего все остальные HTML-элементы, имеющиеся на странице. Свойство `padding-top` добавляет пространство между верхней частью элемента и помещаемым внутри него содержимым. Иными словами, набранный код приведет к добавлению 25 пикселей пустого пространства между верхней частью окна браузера и содержимым страницы. Свойство `background-image` добавляет к странице фоновое изображение. Свойство `background-image` может отобразить графику множеством различных способов, но в данном случае изображение будет выведено в виде неперекрывающейся мозаики слева направо и сверху вниз. Дополнительные сведения о свойствах фоновых изображений будут даны в главе 8.

3. Добавьте второе правило сразу же после только что введенного в файле `styles.css`:

```
body {  
    width: 80%;  
    padding: 20px;  
    margin: 0 auto;  
    border-radius: 10px;  
    box-shadow: 10px 10px 10px rgba(0,0,0,.5);  
    background-color: #E1EDEB;  
}
```

Это правило относится к телу веб-страницы (элементу `body`), которое включает в себя всю информацию, видимую в окне браузера. В этом определении стиля задается множество различных свойств, каждое из которых еще будет рассмотрено более подробно в книге. Но если дать стилю краткое описание, он создает прямоугольную область (блок) для содержимого страницы, составляющую 80 % от ширины окна браузера, имеющую внутри себя небольшое пустое пространство, сдвигающее текст от краев блока (свойство `padding`), и производит центрирование блока на странице (свойство `margin`), в частности конкретный прием центровки содержимого страницы, рассматриваемый в практикуме главы 3. И наконец, блок получает светло-синий цвет фона и прозрачную темную тень.

Вместо повторного набора стилей, созданных в предыдущем уроке, просто скопируем стили, созданные ранее, и вставим их в эту внешнюю таблицу стилей.

4. Откройте страницу `index.html`, над которой работали, и скопируйте весь текст, содержащийся внутри тегов элемента `style` (не копируйте сами теги элемента `style`).

Скопируйте код стилей тем же самым способом, которым скопировали бы любой текст. Например, с помощью меню Правка ► Копировать (`Edit ► Copy`) или нажатием сочетания клавиш `Ctrl+C`.

5. В файл `styles.css` вставьте этот код стилей либо посредством меню Правка ► Вставить (`Edit ► Paste`), либо нажатием сочетания клавиш `Ctrl+V`.

Внешняя таблица стилей никогда не должна содержать HTML-код, именно поэтому вы и не копировали теги элемента `style`.

6. Сохраните файл `styles.css`.

Теперь нужно из HTML-файла удалить CSS-код и связать новую таблицу стилей с этим файлом.

7. Вернитесь к файлу `index.html` в своем текстовом редакторе и удалите теги элемента `style` и все CSS-стили, определенные в нем ранее.

Вам больше не нужны эти стили внутренней таблицы, поскольку они перенесены во внешнюю таблицу стилей, которую сейчас нужно присоединить к HTML-файлу. В этом уроке вы окунетесь в захватывающий мир веб-шрифтов. Все, что касается веб-шрифтов, будет рассмотрено в главе 6, но основная идея заключается в том, что вы можете использовать на веб-страницах практически любые шрифты, даже те, которые посетители вашего сайта не установили на своих компьютерах, предоставив просто ссылку на файл с этим шрифтом. Существует множество различных способов применения веб-шрифтов, но в данном примере будет использована служба веб-шрифтов Google.

8. В позиции HTML-файла, где находилась встроенная таблица стилей (между закрывающими тегами `</title>` и `<head>`), введите следующий код:

```
<link href='http://fonts.googleapis.com/css?family=Kurale' rel='stylesheet'>
```

Как и прежде, вам пока не стоит вдаваться в подробности. На этом этапе нужно лишь знать, что, встретившись с данной ссылкой, браузер загружает с сервера Google шрифт, который называется `Kurale` и который могут свободно использовать ваши CSS-стили.

Затем вы укажете ссылку на созданную ранее внешнюю таблицу стилей.

9. После только что добавленного элемента `link` наберите следующий код:

```
<link href="styles.css" rel="stylesheet">
```

Элемент `link` определяет местонахождение внешней таблицы стилей. Атрибут `rel` оповещает браузер о том, что ссылка ведет на таблицу стилей.

---

#### ПРИМЕЧАНИЕ

В данном примере файл внешней таблицы стилей расположен в той же самой папке, что и веб-страница, так что использование имени файла в качестве значения `href` предполагает простой путь относительно документа. А если бы он находился в любой другой папке, путь был бы иным. В таком случае для указания местонахождения файла нужно задавать путь либо относительно самого документа, то есть веб-страницы, либо относительно корневого каталога сайта. Применяется такая же методика, как и при указании гиперссылки на другую веб-страницу в HTML-теге `<img>` (информацию по этой теме смотрите по адресу [tinyurl.com/oe6j3sr](http://tinyurl.com/oe6j3sr)).

---

10. Сохраните файл и просмотрите его в браузере.

Вы увидите контент с теми же стилями для элементов `h1` и `p`, которые были созданы ранее во внутренней таблице стилей. Кроме того, теперь есть пятнистый желто-коричневый фон (фоновое изображение, примененное в элементе `html`), а также светлый сине-зеленый прямоугольник. Он является элементом `body`, и его ширина составляет 80 % от ширины окна браузера. Попробуйте изменить размер окна браузера и обратите внимание, что размер прямоугольника также изменяется. Прямоугольник также отбрасывает тень, но сквозь нее можно увидеть пятнистый фон. Все это получается благодаря цветовой модели `rgba`, включающей установки прозрачности (которая будет рассмотрена далее).

---

#### ПРИМЕЧАНИЕ

Если на получившейся веб-странице отсутствует форматирование (к примеру, заголовок отображается шрифтом малого размера и никак не выделен), то, вероятно, в шаге 6 вы набрали код с ошибкой или сохранили файл `styles.css` в папке, отличной от той, в которой находится файл `index.html`. В этом случае просто переместите файл `styles.css` в ту же самую папку.

---

Теперь вы увидите веб-шрифт, ссылка на который была добавлена в шаге 8.

11. Запустите текстовый редактор и вернитесь к файлу `styles.css`. Добавьте к стилю селектора `h1` следующие две строки:

```
font-family: 'Kurale', 'Arial Black', serif;
font-weight: normal;
```

В конечном результате стиль должен приобрести следующий вид (добавления выделены полужирным шрифтом):

```
h1 {
  font-family: 'Kurale', 'Arial Black', serif;
  font-weight: normal;
  color: #6A94CC;
  font-size: 3em;
  margin: 0;
}
```

Теперь при просмотре страницы в браузере вы увидите заголовок, отформатированный шрифтом Kurale.

#### ПРИМЕЧАНИЕ

Если вы не видите новый шрифт с тонкими линиями (засечками) на концах букв, показанный на рис. 2.4, значит, какое-то из двух обстоятельств не сложилось. Может отсутствовать подключение к Интернету, что мешает загрузить шрифт с сервера Google, а кроме того, вы можете допустить опечатку либо в коде тега `<link>` (при выполнении шага 8), либо в объявлении `font-family` (во второй строке показанного выше кода).

Чтобы продемонстрировать пользу от хранения ваших стилей в их собственных отдельных файлах, вы можете прикрепить таблицу стилей к другой веб-странице.

12. Откройте файл `02\another_page.html`.

Эта веб-страница содержит те же самые HTML-элементы: `h1`, `h2`, `p` и др., с которыми вы уже работали.

13. Установите курсор после закрывающего тега `</title>` и нажмите клавишу `Enter`.

Сейчас нужно добавить к этой веб-странице ссылку на веб-шрифт и уже созданную внешнюю таблицу стилей.

14. Наберите код элемента `link`, который применялся в шагах 8 и 9.

Код веб-страницы должен выглядеть следующим образом (код, который вы только что набрали, отмечен полужирным шрифтом):

```
<title>Другая страница</title>
  <link href='http://fonts.googleapis.com/css?family=Kurale' rel='stylesheet'>
<link href="styles.css" rel="stylesheet">
</head>
```

15. Сохраните страницу и просмотрите ее в браузере.

Достаточно всего двух строк кода, чтобы мгновенно преобразить внешний вид веб-страницы. Чтобы показать, насколько просто обновить стиль, описанный во внешней таблице стилей, попробуйте отредактировать один из стилей или добавить другие.

16. Откройте файл `styles.css` и добавьте объявление свойства `font-family`: `"Palatino Linotype", Baskerville, serif`; — в начале стиля элемента `p`.

Код должен выглядеть следующим образом (добавленный код выделен полужирным шрифтом):

```
p {
  font-family: "Palatino Linotype", Baskerville, serif;
  font-size: 1.25em;
  color: #616161;
  line-height: 150%;
  margin-top: 20px;
```

```
margin-left: 60px;
}
```

В данном случае веб-шрифт не используется и все зависит от наличия перечисленных шрифтов на машине посетителя сайта (все вопросы об использовании шрифтов будут рассмотрены позже). Затем создайте новый стиль для элемента h2.

17. Установите курсор после заключительной фигурной скобки } стиля элемента p, нажмите клавишу **Enter** и добавьте такое правило:

```
h2 {
  color: #B1967C;
  font-family: 'Kurale', 'Arial Black', serif;
  font-weight: normal;
  font-size: 2.2em;
  border-bottom: 2px white solid;
  background: url(images/head-icon.png) no-repeat 10px 10px;
  padding: 0 0 2px 60px;
  margin: 0;
}
```

С большинством этих CSS-свойств вы уже знакомы, однако некоторые из них новые, например border-bottom, используемое для добавления линии под заголовком. Свойство background вообще предоставляет возможность комбинировать различные свойства — в данном случае это background-image и background-repeat — в одно. Не беспокойтесь о назначении этих свойств, вы подробно изучите их в последующих главах. О свойствах шрифта читайте в главе 6, о свойствах, устанавливающих отступы и границы, — в главе 7, а о свойствах, устанавливающих параметры фона, — в главе 8.

Стили, которые вы создавали до сих пор, работают с основными элементами h1, h2 и p, изменяя оформление каждого их экземпляра. Другими словами, стиль p, который вы создали, форматирует каждый абзац на странице. Но если вы хотите воздействовать на конкретный абзац, нужно использовать другой вид стиля.

18. Перейдите к концу стиля h2, нажмите клавишу **Enter** после закрывающей скобки } и наберите следующий код:

```
.intro {
  color: #666666;
  font-family: 'Kurale', Helvetica, sans-serif;
  font-size: 1.2em;
  margin-left: 0;
  margin-bottom: 25px;
}
```

Если вы просмотрите файл index.html в браузере, то пока не заметите никаких изменений. Этот тип стиля, называемый *селектором класса*, форматирует только отдельные элементы, к которым вы примените класс. Для того чтобы этот новый стиль работал, придется немного отредактировать HTML-код.

19. Сохраните файл `styles.css` и перейдите к файлу `index.html` в своем текстовом редакторе. Найдите открывающий тег `<p>`, следующий за элементом `h1`, и добавьте код `class="intro"`. Открывающий тег должен выглядеть так:

```
<p class="intro">
```

Вам не нужно добавлять точку перед словом `intro`, как вы это делали, создавая стиль в шаге 18 (почему так, вы узнаете в следующей главе). Дополнительный HTML-код применяет стиль к первому абзацу (и только к первому).

Повторите этот шаг для файла `another_page.html` — добавьте `class="intro"` к первому тегу `<p>` на этой странице.

20. Сохраните все файлы и просмотрите страницы `index.html` и `another_page.html` в браузере.

Обратите внимание, что вследствие простого редактирования CSS-файла внешний вид обеих веб-страниц изменился. Закройте глаза и представьте, что ваш сайт содержит тысячи страниц. Мощные средства изменения дизайна, не правда ли?

Вам осталось выполнить еще одно небольшое изменение. Если вы посмотрите в нижнюю часть страницы в браузере, то увидите информацию об авторских правах. Форматирование этой строки отличается от стиля абзацев, расположенных над ней. Страница смотрелась бы гораздо лучше, если бы их стили совпадали.

21. Вернитесь к файлу `styles.css` в текстовом редакторе. Найдите стиль с селектором `p`. Добавьте запятую, пробел и слово `address`.

Стиль должен выглядеть следующим образом:

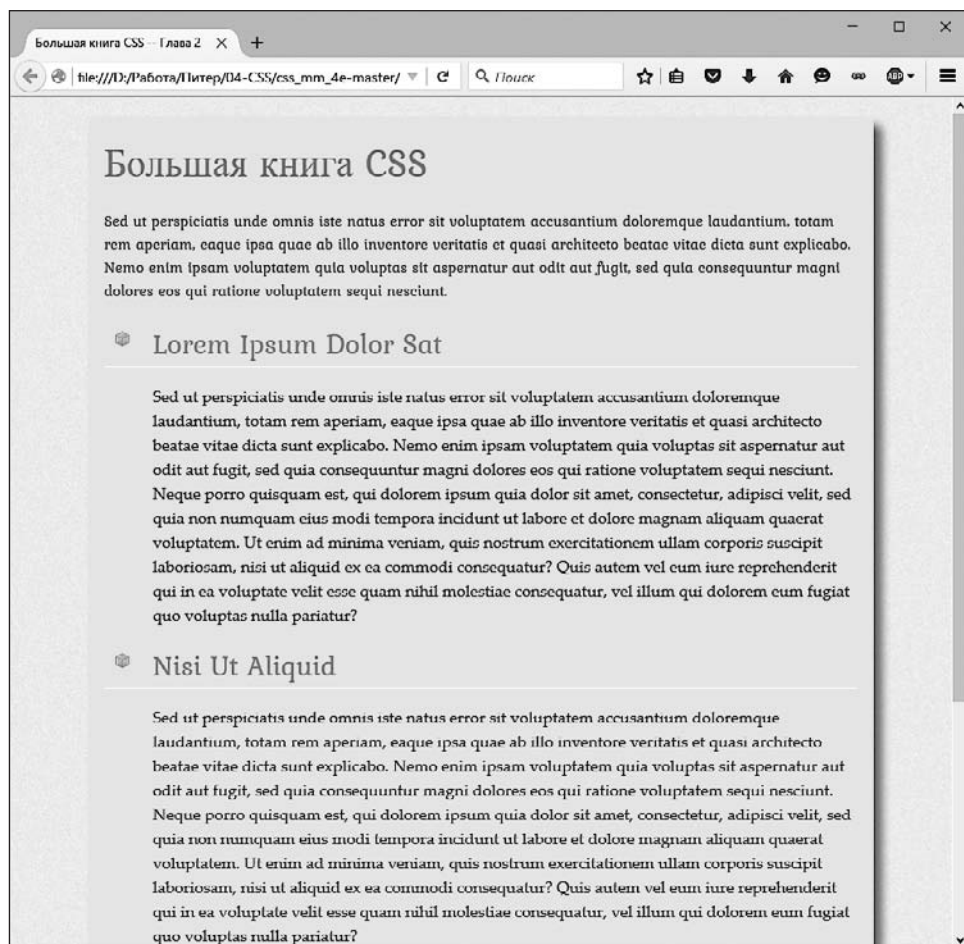
```
p, address {
  font-family: "Palatino Linotype", Baskerville, serif;
  font-size: 1.25em;
  color: #616161;
  line-height: 150%;
  margin-top: 20px;
  margin-left: 60px;
}
```

Вы изменили не свойства стиля, а его селектор. Фактически вы создали групповой селектор, который очень эффективен в случаях, когда необходимо применить один и тот же стиль ко множеству элементов страницы. Подробнее об этом способе вы узнаете позже. В данном случае стиль применился к двум элементам: `p` и `address`.

22. Сохраните все файлы и просмотрите страницу `index.html`.

На рис. 2.4 представлен окончательный вид страницы (вы можете найти ее в папке `02_finished`).

Для практики попробуйте внести в файл `styles.css` какие-либо изменения. Поэкспериментируйте с различными значениями свойств. Можно, например, изменить значение свойства `width` стиля `body` или изменить размер шрифта.



**Рис. 2.4.** Применение внешних таблиц стилей позволяет обновить дизайн всех страниц сайта за один прием посредством редактирования единственного CSS-файла. Полное исключение CSS-кода из HTML-документа и вынесение его в отдельный файл может значительно уменьшить размер самих веб-страниц, следовательно, они будут загружаться намного быстрее