

Введение

Каскадные таблицы стилей, или Cascading Style Sheets (CSS), обеспечивают творческую свободу в разметке и дизайне веб-страниц. Пользуясь CSS, вы сможете украсить текст страниц привлекательными заголовками, буквицами, рамками, как в красочных глянцевых журналах. Можно точно разместить и позиционировать изображения, создать столбцы и баннеры, выделить ссылки динамическими эффектами. Кроме того, можно добиться постепенного появления и исчезновения элементов, перемещения объектов по странице или медленного изменения цвета кнопки при прохождении над ней указателя мыши.

Вы думаете, что все это довольно сложно? Напротив! Каскадные таблицы стилей как раз и предназначены для упрощения процесса оформления веб-страниц. Следующие несколько страниц будут посвящены основам CSS, а при изучении гл. 1 вы перейдете к непосредственной работе по созданию веб-страниц, улучшенных с помощью CSS.

Как работает CSS

CSS работает с HTML, но не имеет к HTML никакого отношения. Это совершенно другой язык. HTML структурирует документ, упорядочивая информацию в заголовки, абзацы, маркированные списки и т. д., в то время как CSS тесно взаимодействует с браузером, чтобы оформление HTML-документа имело совершенный вид.

Например, вы могли бы использовать HTML, чтобы превратить фразу в заголовок, отделяя его от содержания страницы, но лучше применять CSS для форматирования заголовка, скажем, большим полужирным красным шрифтом с позиционированием на 50 пикселей от левого края окна. В CSS это форматирование текста включает в себя стиль — правило, описывающее внешний вид конкретной части веб-страницы. А таблица стилей (style sheet) является набором таких стилей.

Можно также создавать стили специально для работы с изображениями. Например, с помощью стилей можно выровнять изображение по правому краю веб-страницы, поместить его в цветную рамку, отделить от окружающего текста на 50 пикселей.

Создав стиль один раз, можно применять его к текстовым фрагментам, изображениям, заголовкам и любым другим элементам страницы сколько угодно. Например, вы можете выбрать абзац текста и применить к нему стиль, тут же изменяющий размер, цвет и шрифт текста. Можно также сделать стили для определенных HTML-тегов так, чтобы, например, все заголовки первого уровня (теги `<h1>`) на вашем сайте были отображены в одинаковом стиле, независимо от того, где они размещены.

Преимущества CSS

До появления CSS дизайнеры веб-страниц были ограничены возможностями разметки и оформления языка HTML. И если вы занимались серфингом в Интернете в 1995 году, то уясните разницу в возможностях CSS и HTML. Читая дальше эту книгу, вы поймете, что HTML все еще является основой создания страниц в Сети, но это отнюдь не средство формирования их дизайна. Безусловно, HTML обеспечивает простейшее форматирование текста, изображений, таблиц и других элементов страниц, и оформитель может придать им прекрасный внешний вид. Как правило, в результате веб-страницы получаются громоздкими и медленно загружаются.

Таблицы стилей CSS, напротив, имеют следующие преимущества.

- Больше возможностей форматирования, нежели в HTML. В CSS вы можете форматировать абзацы по мере их появления в тексте (например, с абзацным отступом и с произвольным интервалом между абзацами) и изменять межстрочный интервал (расстояние между двумя соседними строками текста в абзаце).
- При использовании CSS вы решаете, каким образом добавить фоновое изображение на страницу. Оно может отображаться в виде неперекрывающейся мозаики, повторяться и т. д.
- Еще более значимо то, что стили CSS занимают намного меньше места, чем формирующие команды HTML. Например, тег ``. Применяя CSS, вы можете уменьшить размер веб-страниц. В результате они будут стильно выглядеть и быстрее загружаться.
- Стилиевые таблицы также облегчают обновление сайта. Можно собрать все стили в единственный внешний файл и связать его со всеми страницами сайта. Когда вы редактируете стиль, изменения моментально затрагивают все элементы страниц сайта, где есть ссылка на описанный в таблице стиль. Вы можете полностью изменить внешний вид путем редактирования единственного файла таблицы стилей.

Что необходимо знать

Эта книга предполагает, что вы уже знакомы с языком HTML (и, возможно, имеете небольшой опыт работы с CSS). Подразумевается, что вы создали пару сайтов (или по крайней мере несколько веб-страниц) и знакомы с основными тегами, такими как `<html>`, `<p>`, `<h1>`, `<table>` и т. д., составляющими основу языка гипертекстовой разметки документов. CSS бесполезен без HTML, поэтому, чтобы продолжить изучение CSS, вы должны знать, как создать простейшую веб-страницу с использованием основных HTML-тегов.

Если вы раньше создавали веб-страницы на HTML, но чувствуете, что знания требуется освежить, вам поможет следующий раздел книги.

ПРИМЕЧАНИЕ

Если вы только знакомитесь с HTML и возможностями применения его на практике, то посетите бесплатные обучающие сайты: HTML Dog (www.htmldog.com/guides/htmlbeginner) и W3Schools (www.w3schools.com/html).

HTML: структура языка

В языке гипертекстовой разметки HTML используются простые команды, именуемые *тегами*, для определения различных частей — фрагментов. Ниже приведен код HTML простой веб-страницы:

```
<!doctype html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Это заголовок веб-страницы</title>
  </head>
  <body>
    <p>А это абзац этой веб-страницы</p>
  </body>
</html>
```

Конечно, пример очень простой, но демонстрирует все основные элементы, необходимые обычной веб-странице. В нем вы заметите то, что называется *объявлением типа документа* — `doctype`, за ним следует тег `<html>` (со скобками), потом `head` (голова, заголовок), следом `body` (тело, тело документа), а в нем непосредственно содержимое веб-страницы. Все это завершается закрывающим тегом `</html>`.

Типы документов

Все веб-страницы начинаются с объявления `doctype` — строки кода, определяющей, какой разновидностью HTML вы пользовались при написании страницы. В течение многих лет использовались два типа документов — HTML 4.01 и XHTML 1.0, и каждый из них имеет два стиля: строгий и промежуточный. Например, промежуточный `doctype` HTML 4.01 имеет следующий вид (другие `doctype`-объявления для HTML 4.01 и XHTML 1.0 выглядят примерно так же):

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
```

ПРИМЕЧАНИЕ

Примеры всех разновидностей `doctype`-объявлений можно найти на веб-сайте www.webstandards.org/learn/reference/templates.

Если посмотреть на код примера HTML-страницы, показанный в этом разделе, то вы увидите, что в нем используется краткая форма объявления `doctype`:

```
<!doctype html>
```

Это новое упрощенное объявление `doctype`, появившееся в HTML5. По сравнению с предшественниками в HTML5 заложена простота и рациональность использования. В этой книге используется объявление `doctype` из HTML5, поддерживаемое любым популярным браузером (даже старым Internet Explorer 6). И, поскольку HTML5 является будущим Интернета, далее применять старые `doctype`-объявления не имеет никакого смысла.

ПРИМЕЧАНИЕ

То, что HTML `doctype`-объявление работает в старых браузерах, еще не значит, что эти браузеры понимают все теги или особенности HTML5. Например, Internet Explorer 8 или более ранние версии

не распознают новые теги HTML5. Чтобы внести в этих версиях в теги стиль с помощью CSS, нужно будет воспользоваться кодом JavaScript. Как вывести старые браузеры на современный уровень, будет показано далее.

Независимо от предпочитаемого типа документа, объявляемого с помощью doctype, важно, чтобы использовалось объявление хотя бы одного из них. Без этого ваши страницы будут выглядеть по-разному в зависимости от браузера, применяемого вашим посетителем, поскольку браузеры, не имеющие в качестве руководства объявления doctype, по-разному отображают информацию, которая стилизована с помощью CSS.

Каждое doctype-объявление требует от вас написания HTML-кода определенным образом. Например, тег для разбиения строк имеет в HTML 4.01 следующий вид:

```
<br>
```

Но в XHTML у этого тега следующий вид:

```
<br />
```

И здесь проявляется еще одно преимущество HTML5: он допускает применение любого из этих вариантов.

Как работают HTML-теги

В приведенном выше примере, как и в HTML-коде любой веб-страницы, большинство команд-тегов используются парами, начиная и завершая какой-то фрагмент — блок текста или другие команды. Будучи заключенными в скобки, эти теги представляют собой команды, которые говорят браузеру, каким образом отображать веб-страницу. Теги являются разметочной (markup) частью гипертекстового языка разметки — Hypertext Markup Language.

Открывающий тег каждой пары показывает браузеру, где команда начинается, а закрывающий — где заканчивается. Закрывающий тег всегда предваряется прямой слешем (/) после первого символа скобки (<).

На любой веб-странице обычно имеются как минимум следующие четыре элемента.

- Самая первая строка примера содержит объявление типа документа — doctype, рассмотренное в предыдущем разделе.
- Тег <html> требуется в начале веб-страницы и (с добавленным слешем) в конце: </html>. Этот тег говорит браузеру, что документ является программным кодом на языке HTML. Все содержимое страницы, включая остальные теги, находится между открывающим и закрывающим <html>.

Если представить веб-страницу в виде дерева, то <html> будет его стволом. Две основные части любой веб-страницы — заголовок и тело — представляют собой ветви.

- *Заголовок* (head) веб-страницы включает в себя ее название. Здесь также может содержаться другая информация, невидимая при просмотре веб-страницы, например описание страницы, которая предназначена для браузеров и поисковых машин. Раздел заголовка заключается в открывающий и закрывающий <head>-теги.

Кроме того, раздел заголовка может содержать информацию, которая используется браузером для оформления HTML, имеющегося на странице и для придания странице интерактивности. Вы увидите, что `<head>`-раздел может включать в себя код CSS (вроде того, который вы будете учиться создавать с помощью данной книги) или ссылку на другой файл, содержащий CSS-информацию.

- Тело (`body`) веб-страницы, следующее непосредственно за заголовком и заключенное в теги `<body>`, содержит все, что должно появиться в окне браузера: заголовки, текст, изображения и т. д.

Внутри тега `<body>`, как правило, можно найти следующие теги:

- `<p>` — открывающий тег начинает абзац, а закрывающий `</p>` — заканчивает;
- `` — выделяет текст полужирным шрифтом; например, фрагмент `Warning!` сообщит браузеру о том, что слово `Warning!` должно быть выделено;
- `<a>`, или тег привязки (якорный), — создает гиперссылку, при щелчке на которой можно переместиться в другое место веб-страницы или на другой сайт (нужно указать браузеру эту ссылку путем размещения ее внутри `<a>`, например можно набрать `Щелкните здесь!`).

Браузер знает, что при щелчке кнопкой мыши на ссылке со словами *Щелкните здесь!* посетитель вашей страницы должен перейти на сайт с адресом `http://www.missingmanuals.com`. Часть тега `a` — слово `href` — называют *атрибутом*, а URL (унифицированный указатель информационного ресурса, или URL-адрес) является его *значением*. В этом примере `http://www.missingmanuals.com` — значение атрибута `href`.

Немного слов о XHTML

Как и всякая другая технология, язык HTML со временем претерпевал изменения. Хотя стандартный HTML справлялся со своей задачей достаточно неплохо, он всегда допускал некоторую небрежность при написании программ. Кроме всего прочего он позволял использовать строчные или прописные буквы в тегах или вообще их смесь (например, теги `<body>`, `<BODY>`, `<BoDy>` абсолютно идентичны), опускать закрывающие теги (например, можно создать абзац, используя только открывающий тег `<p>`, а закрывающий `</p>` даже и не указывать). С одной стороны, подобная гибкость позволяет создавать сайты гораздо быстрее, но с другой — она доставляет массу проблем для различных браузеров, смартфонов и другой техники, взаимодействующей с данными Интернета.

Кроме того, HTML не работает с одним из самых мощных языков Интернета: XML, или с расширяемым языком разметки — Extensible Markup Language.

Чтобы идти в ногу со временем, еще в 2000 году была введена улучшенная версия HTML, которую назвали XHTML, и зачастую ее можно найти на многих сайтах (фактически XHTML — это просто подогнанная под XML версия HTML). XHTML рассматривался (в 2000 году) как будущее развитие языка, но был вытеснен HTML5. Хотя браузеры по-прежнему разбираются в языке XHTML (и, наверное, еще долго будут его понимать), в этой книге он не рассматривается.