

Каскадные таблицы стилей

CSS (англ. Cascading Style Sheets — каскадные таблицы стилей) — технология описания внешнего вида документа, оформленного языком разметки. Преимущественно используется как средство оформления веб-страниц в формате HTML и XHTML, но может применяться с любыми видами документов в формате XML, включая SVG и XUL. Каскадные таблицы стилей используются создателями веб-страниц для задания цветов, шрифтов, расположения и других аспектов представления веб-документа.

Основной целью разработки CSS являлось разделение содержимого (написанного на HTML или другом языке разметки) и оформления документа (написанного на CSS). Это разделение может увеличить доступность документа, предоставить большую гибкость и возможность управления его представлением, а также уменьшить сложность и повторяемость в структурном содержимом.

Кроме того, CSS позволяет представлять один и тот же документ в различных стилях или методах вывода, таких как экранное представление, печать, чтение голосом (специальным голосовым браузером или программой чтения с экрана), или при выводе устройствами, использующими шрифт Брайля.

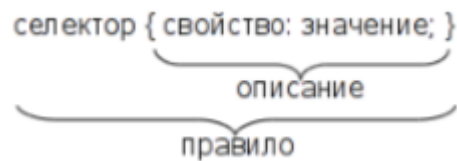
Что такое CSS?

Каскадные таблицы стилей (Cascading Style Sheets, CSS) — это стандарт, определяющий представление данных в браузере. Если HTML предоставляет информацию о структуре документа, то таблицы стилей сообщают как он должен выглядеть. Стил — это совокупность правил, применяемых к элементу гипертекста и определяющих способ его отображения. Стил включает все типы элементов дизайна: шрифт, фон, текст, цвета ссылок, поля и расположение объектов на странице. Таблица стилей — это совокупность стилей, применимых к гипертекстовому документу. Каскадирование — это порядок применения различных стилей к веб-странице.

Браузер, поддерживающий таблицы стилей, будет последовательно применять их в соответствии с приоритетом: сначала связанные, затем внедренные и, наконец, встроенные стили. Другой аспект каскадирования — наследование (inheritance), — означает, что если не указано иное, то конкретный стил будет применен ко всем дочерним элементами

гипертекстового документа. Например, если вы примените определенный цвет текста в теге <div>, то все теги внутри этого блока будут отображаться этим же цветом. Использование каскадных таблиц дает возможность разделить содержимое и его представление и гибко управлять отображением гипертекстовых документов путем изменения стилей.

Таблицы стилей строятся в соответствии с определенным порядком (синтаксисом), в противном случае они не могут нормально работать. Таблицы стилей состояются из определенных частей (рис. 1):



Селектор — это элемент, к которому будут применяться назначаемые стили. Это может быть тег, класс или идентификатор объекта гипертекстового документа.

Свойство (Property). Свойство определяет одну или несколько характеристик селектора. Свойства задают формат отображения селектора: отступы, шрифты, выравнивание, размеры и т.д. **Значение (Value).**

Значения — это фактические числовые или строковые константы, определяющие свойство селектора.

Описание (Declaration). Совокупность свойств и их значений.

Правило (Rule). Полное описание стиля (селектор + описание). Таким образом, таблица стилей — это набор правил, задающих значения свойств селекторов, перечисленных в этой таблице. Общий синтаксис описания правила выглядит так:

Итак, каскадная таблица стилей — это набор правил форматирования тегов HTML. Приведем несколько примеров написания таких правил: Основной текст с выравниванием по ширине, абзацный отступ 30px, гарнитура (шрифт) — Serif, кегль (размер шрифта) — 14px:

Это правило будет применено ко всем тегам <p>

```
p {  
    text-align: justify;  
    text-indent: 30px;
```

```
font-family: Serif;

font-size: 14px;

}
```

Синий цвет для заголовков с первого по третий уровень:

```
h1, h2, h3 {

    color: blue; /* тоже самое, что и #0000FF */

}
```

Таблицы и изображения выводить без обрамления:

```
table, img {

    border: none;

}
```

Ссылки в элементах списков показывать без подчеркивания:

```
li a {

    text-decoration: none;

}
```

Внутренние отступы слева и справа для блоков (<div>), заголовков таблиц и ячеек таблиц установить в 10px и залить фон желтым цветом:

```
div, th, td {

    padding-left: 10px; padding-right: 10px; background-
color: yellow;

}
```

Все ссылки в документе отображать черным цветом и полужирным шрифтом, а в основном тексте и списках — обычным, а также выделять их зеленым цветом и подчеркивать только при наведении курсора (в описании правил использован псевдоэлемент `a:hover`).

```
a {
    color: black;
    font-weight: bold;
}
p a, li a {
    font-weight: normal;
    text-decoration: none;
}
p a:hover, li a:hover {
    color: #00FF00;
    text-decoration: underline;
}
```

Классы

Стандарт CSS представляет возможности создания именованных стилей — стилевых классов. Это позволяет ответить на такой, например, вопрос: Как применить разные стили к одному и тому же селектору? Предположим, что в документе вам нужны два различных вида основного текста — один без отступа, второй — с левым отступом и шрифтом красного цвета. Для этого нужно создать правила для каждого из них, например так:

```
p {
    margin-left: 0;
}
p.warn {
```

```
margin-left: 40px;

color: #FF00;

}
```

Для применения созданного класса его имя нужно указать в атрибуте class для выбранных абзацев:

```
<p class="warn">Красный текст с отступом слева</p>
```

Общий синтаксис описания класса: селектор.имя_класса {описание}

При создании класса селектор можно не указывать, тогда это правило можно применять к любому селектору, поддерживающему тот же набор свойств. Вот несколько примеров:

Правило:

```
.solid_blue {

    color: blue;

}
```

Использование:

```
<p class="solid_blue">Синий текст абзаца</p>      <li
class="solid_blue">Синий текст элемента списка</li>
```

Правило:

```
h1.bigsans {

    font-family: Sans; font-size: 1.5em;

}

h1.smallserif {

    font-family: Serif; font-size: .84em;

}
```

Использование:

```
<h1 class="bigsans">Большой, но рубленый</h1> <h1  
class="smallserif">Маленький, но с засечками</h1>
```

Идентификаторы

В качестве селектора может выступать идентификатор элемента гипертекста, указанный в атрибуте `id`. Для назначения стилей таким элементам используется синтаксис, аналогичный описанию классов, но вместо точки ставится знак `#` (“решетка”). Например:

```
div#content {  
    position: absolute;  
    top: 10px;  
    left: 10%;  
    right: 10%;  
    border: solid 1px silver;  
}
```

```
<div id="content">Текст</div>
```

Следует помнить, что идентификаторы элементов должны быть уникальны в пределах документа.

Группировка свойств

Группировка (grouping) состоит в объединении значений родственных свойств. При этом таблица стилей становится более компактной, но предъявляются более жесткие требования к описанию правил. Ниже приведен пример обычного стиля, задающего отступы:

```
div {  
    margin-left: 10px;  
    margin-top: 5px;  
    margin-right: 40px;
```

```
margin-bottom: 15px;
}
```

Это же правило можно переписать с группировкой в следующем виде:

```
div {
    margin: 5px 40px 15px 10px;
} /*порядок: top right bottom left*/
```

Оба стиля будут отображаться одинаково. Группировка может применяться для таких свойств, как padding, font, border, background и еще некоторых (см. документацию CSS).

Использование в веб-страницах

Существует три способа применения таблицы стилей к документу HTML:

Встраивание (Inline). Этот метод позволяет применить стиль к заданному тегу HTML. Встраивание стилей предоставляет максимальный контроль над всеми элементами веб-страницы. Встроенный стиль применяется к любому тегу HTML с помощью атрибута style следующим образом:

```
<p style="font: 12pt Courier">
```

Это текст с кеглем 12 точек и гарнитурой Courier

```
</p>
```

Пример:

```
<div style="font-family: Garamond; font-size: 18 pt;>"
```

Весь текст в этом разделе имеет размер 18 точек и шрифт Garamond.

```
<span style="color:#ff3300;">
```

А этот фрагмент еще и выделен красным цветом.

```
</span>
```

```
</div>
```

Встроенные стили полезны, когда необходима тонкая настройка отображения некоторого элемента страницы или небольшой веб-страницы.

Внедренные стили используют тег `<style>`, который обычно размещают в заголовке HTML-документа (`<head>...</head>`):

```
<html> <head>
... <style>
правила CSS
</style>
...
</head> <body>
```

Связанные (linked), или внешние (external) таблицы стилей — наиболее удобное решение, когда речь идет об оформлении целого сайта. Описание правил помещается в отдельный файл (обычно, но не обязательно, с расширением `.css`). С помощью тега `<link>` выполняется связывание этой таблицы стилей с каждой страницей, где ее необходимо применить, например так:

```
<link rel=stylesheet href="sample.css" type="text/css">
```

Любая страница, содержащая такую связь, будет оформлена в соответствии со стилями, указанными в файле `sample.css`. Следует отметить, что файл со стилями физически может находиться на другом веб-сервере, тогда в `href` нужно указать абсолютный путь к нему.

Проблемы с браузерами

Обязательно просматривайте страницы с таблицами стилей в различных браузерах. Это связано с тем, что разные браузеры могут по-разному интерпретировать одно и то же правило, а некоторые свойства и/или значения и вовсе не поддерживать. Следует также тестировать страницы с отключенными стилями (например, в текстовых браузерах), чтобы убедиться, что страница читабельна.

Если вам нужна сотня-другая-третья страниц HTML — используйте внешнюю, глобальную, таблицу стилей. Если некоторые из этих страниц требуют корректировки общего оформления — используйте внедренный стиль. А если на странице нужно явно изменить оформление одного-двух элементов, то применяйте встроенные стили. Именно в таком порядке происходит перекрытие стилей при каскадировании, схематично это можно представить так: связанные стили -> внедренные стили -> встроенные стили

Практическое задание

1. Создайте стилевой файл. Используя связывание подключите его в документ.
2. Используя таблицы стилей, измените оформление текста в нескольких элементах.
3. Используя таблицы стилей, измените цвет и фон в нескольких элементах.
4. Используя таблицы стилей, измените свойства шрифта (написание, размер, оформление) в нескольких элементах.
5. Оформите нумерованные и ненумерованные списки с помощью стилей.
6. Продемонстрируйте селекторы с выборкой по элементу, классу, идентификатору, комбинированные.
7. Продемонстрируйте различные способы применения стилей к документу (внедрение, связывание, встраивание).
8. Создайте стилевой файл с альтернативным оформлением (сохраните его таким образом, чтобы можно было применить либо первый, либо второй стилевой файлы).