

Лабораторная работа 1. Каскадные таблицы стилей.

Цель работы: изучить основы технологии CSS, познакомиться со способами применения CSS к HTML-документу.

Каскадные таблицы стилей или CSS (Cascading Style Sheets) предоставляют возможность перейти на следующий уровень представления информации. Таблицы стилей позволяют разделить оформление и смысловое содержимое страницы.

Внутренние таблицы стилей. Таблица стилей, заданная внутри HTML-элементов при помощи атрибута STYLE, называется внутренней:

```
<h1 style="color:green"> CSS (Cascading Style Sheets)</h1>
```

Недостаток внутренних таблиц стилей заключается в их неуниверсальности. В результате использования атрибута STYLE только один заголовок в документе станет зеленого цвета, для остальных придется точно так же задавать атрибут STYLE.

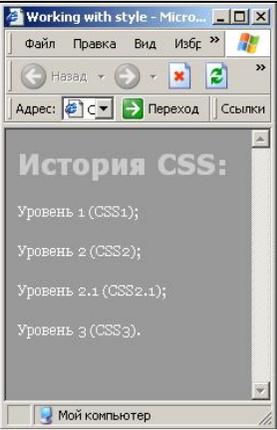
Встроенные таблицы стилей. Встроенная таблица стилей добавляется в HTML-документ между тегами STYLE в секции HEAD. Действия встроенной таблицы стилей распространяются на всю страницу. Например, чтобы сделать все заголовки первого уровня зелеными, можно написать следующую встроенную таблицу:

```
<head><title>Working with style</title>
<style type="text/css">
h1 {color: green;}
</style>
</head><body><h1> CSS (Cascading Style Sheets)</h1>
</body>
```

В результате для всех элементов h1, которые встретятся в данном HTML-документе, будет применен зеленый цвет.

Внешние таблицы стилей. Чтобы не дублировать таблицу стилей между тегами <STYLE> и </STYLE> на каждой странице, ее можно разместить в отдельный файл с расширением .CSS и при необходимости подключать к HTML-документу. Такие таблицы стилей

называются внешними и представляют собой отдельный файл с расширением .CSS, в котором размещаются все CSS-правила, но уже без HTML-тегов.

Внешняя таблица стилей	HTML-страница для таблицы стилей	Результат применения связанной таблицы стилей
<pre>body { background-color: #999; color: black; } font-family: Verdana; font-size: 24px; color: #ccc; } p { font-family: Georgia; font-size: 12px; color: white; }</pre>	<pre><html> <head> <link rel="stylesheet" type="text/css" href="css/mystyle.css" title="MyStyleSheet"> </head> <body> <h1>История CSS: </h1> <p>Уровень 1 (CSS1);</p> <p>Уровень 2 (CSS2); </p> <p>Уровень 2.1 (CSS2.1); </p> <p>Уровень 3 (CSS3). </p> </body> </html></pre>	

Преимущества таблиц стилей. Таблицы стилей концентрируют команды, реализующие визуальные эффекты в одной доступной области. При этом открывается возможность легко применять набор стилей ко всем однотипным элементам, что позволяет группировать в ассоциативные стили или отдельные элементы и существенно сократить объем работы разработчика веб-страниц.

В качестве примера предположим, что в документе все заголовки h2 должны быть красными. Работая с HTML, можно поместить тег font в каждый заголовок:

```
<h2><font color="red">Этот заголовок – красный </font></h2>
```

Эти действия должны быть проделаны для каждого заголовка второго уровня. Если в документе 100 заголовков, то необходимо вставить 100 элементов font, по одному для каждого заголовка. Вместо этого намного лучше было бы иметь одно правило:

```
h2 {color: red;}
```

Для того чтобы изменить цвет текста всех элементов h2, например, на серебристый, необходимо просто изменить правило:

```
h2 {color: silver;}
```

Данное правило не только намного быстрее модифицировать, т.е. для перехода от красного к зеленому цвету достаточно изменить единственное правило, а не заменять каждый из элементов font. Таким образом можно не только централизовать всю информацию о стилях страницы, но и создавать таблицу стилей, которая может применяться ко многим страницам.

Рассмотрим сайт, в котором все заголовки выделены серым шрифтом на белом фоне. Этот цвет они получают из таблицы стилей:

```
h1, h2, h3, h4, h5, h6 {color: gray ; background: white;}
```

Предположим, что сайт состоит из 700 страниц, каждая из которых используют таблицу стилей. Если в определенный момент времени разработчиком будет принято решение, что заголовки должны быть белыми на сером фоне, он редактирует таблицу стилей следующим образом:

```
h1, h2, h3, h4, h5, h6 {color: white ; background: gray;}
```

Каскадное включение. CSS также поддерживает средства разрешения конфликтов правил, называемые каскадным включением (*cascade*). В дополнение к уже существующей таблице стилей можно создать еще одну таблицу, импортируемую в HTML-страницы, или разместить специальные стили в странице. Благодаря каскадному включению новое импортируемое правило переопределит правила существующей таблицы стилей. Применяя правила каскадирования, можно создавать сложные, изменяемые и объединяемые таблицы стилей для обеспечения профессионального представления страниц.

Так же CSS способствует уменьшению размеров документов, сокращая таким образом время загрузки. Группируя информацию о визуальных стилях в одном месте и представляя эти правила посредством компактного синтаксиса, можно удалить все элементы font и другие фрагменты тегов.

Структура правила. Каждое правило имеет две основные части:

Селектор (*selector*) и блок объявлений (*declaration block*). Блок объявлений состоит из одного или более объявлений, а каждое объявление представляет собой сочетание свойства (*property*) и значения (*value*). Части правила показаны на рис. 1.

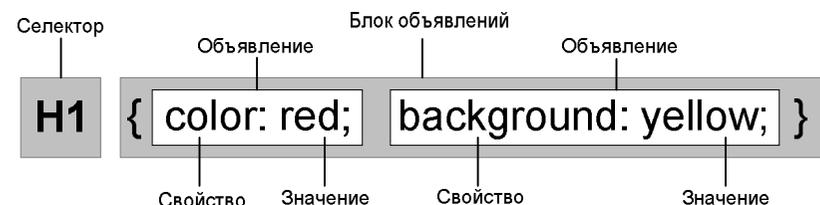


Рис. 1. Структура правила CSS

Селектор, расположенный в левой части правила, определяет на какие части документа распространяется правило.

В правой части находится блок объявлений, образованный одним или несколькими объявлениями. Каждое объявление представляет собой сочетание свойства CSS и значение этого свойства. На представленном выше рисунке, в качестве селектора выбраны элементы h1, а блок объявлений содержит два объявления. Первое объявление определяет, что согласно правилу цвет (color) указанных элементов документа будет красным (red), а второе – что фон (background) этих элементов будет желтым (yellow). Таким образом, все элементы h1 этого документа будут выводиться красным цветом на желтом фоне.

Группировка селекторов. Предположим, необходимо чтобы текст элементов h2 и абзацев был серого цвета. Для этого в левой части правила, располагаются селекторы (h2 и p), которые разделены запятой, а в правой части определяется объявление, согласно которому определяется соответствующий стиль:

```
h2, p {color:gray;}
```

При этом нет никаких ограничений на количество объединяемых в группу селекторов:

```
body, table, th, td, h1, h2, h3, h4, p {color:gray;}
```

Группировка объявлений. Селекторы возможно группировать в одно правило, а следовательно, также можно группировать и объявления. Предположим, необходимо, чтобы текст всех элементов h1 был представлен шрифтом Helvetica фиолетового цвета высотой в 18 пикселей на фоне цвета морской волны. Данные стили возможно написать следующим образом:

```
h1 {font: 18px Helvetica;}
h1 {color: purple;}
h1 {background: aqua;}
```

Представленный выше способ не совсем эффективен, если подобный список создается для элемента, у которого будет 10 или 15 стилей. В таком случае объявления можно сгруппировать следующим образом:

```
h1 {
font: 18px Helvetica;
color: purple;
background: aqua;
}
```

Селекторы классов. Самый распространенный способ применения стилей без учета элементов состоит в том, чтобы обратиться к селекторам классов (*class selectors*), позволяющих назначать стили независимо от элементов документа. Однако сначала придется изменить разметку документа так, чтобы обеспечить возможность работы селекторов класса. Для этого необходимо ввести атрибут class. Чтобы связать стили селектора класса с элементом, необходимо присвоить соответствующее значение атрибуту class данного элемента. В представленном ниже примере, значение warning было присвоено двум элементам: первому абзацу и элементу span во втором абзаце.

```
<p class="warning"> CSS (Cascading Style Sheets)</p>
```

```
<p>Технология описания внешнего вида документа, написанного языком разметки. Преимущественно используется как средство оформления веб-страниц в форматах: <span class="warning">HTML, XHTML, XML.</span>CSS используется создателями веб-страниц для задания цветов, шрифтов, расположения и других аспектов представления документа... </p>
```

В сочетании с показанным примером разметки, применение данного правила приводит к результату, показанному на рис. 2. Стил `font-weight` будет применен ко всем элементам, атрибут класс которых имеет значение `warning`.

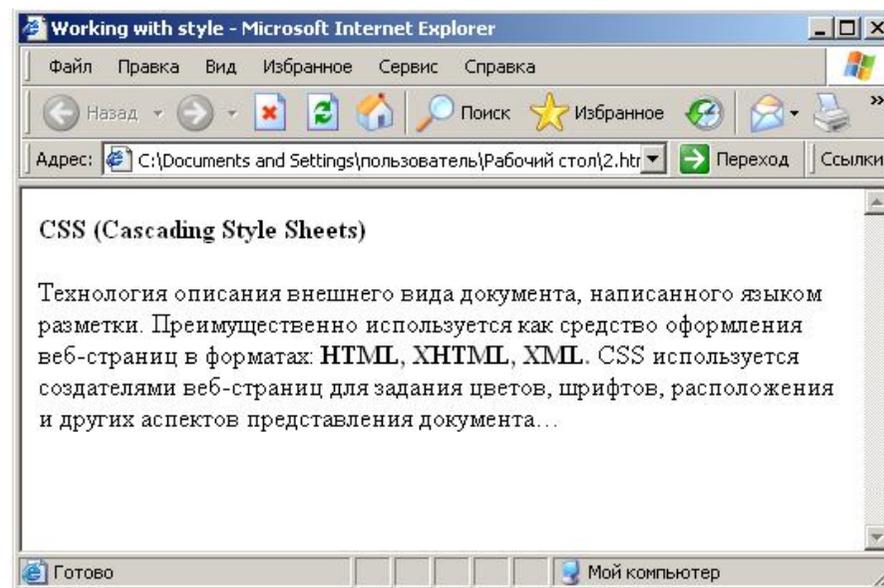


Рис. 2. Пример применения селекторов класса

Допустим необходимо выделить полужирным шрифтом текст только в том случае, если предупреждением является весь абзац:

```
p.warning {font-weight: bold;}
```

Теперь селектор выбирает только элементы p, у которых значение атрибута class равно warning, и игнорирует все другие элементы независимо от их класса.

Универсальные селекторы. Универсальные селекторы выбирают каждый элемент на странице для применения к ним стилей оформления. В HTML-документе возможна очень краткая запись, в которой имени класса предшествует точка (.), и оно может быть объединено с простым селектором:

```
*.warning {font-weight: bold;}
```

Например, следующее правило определяет, что для каждого элемента на странице должна быть добавлена сплошная черная граница толщиной 1 пиксель:

```
* {  
  border: 1px solid #000000;  
}
```

Комментарии CSS. Комментарии помещаются между символами `/*` и `*/`. Комментарии можно добавлять либо между правилами, либо внутри блока свойств — например, в следующем коде CSS второе и третье свойства помещены между ограничителями комментария, так что они будут игнорироваться браузером. Комментарии могут охватывать несколько строк, и браузер будет игнорировать эти строки:

```
/* Это комментарий CSS */  
/* Это комментарий CSS,  
и он может занимать несколько строк */
```

Объединение CSS и HTML. Тег link. Рассмотрим применение тега `link`. На рис. 3 показана общая структура ссылки на внешнюю таблицу стилей.

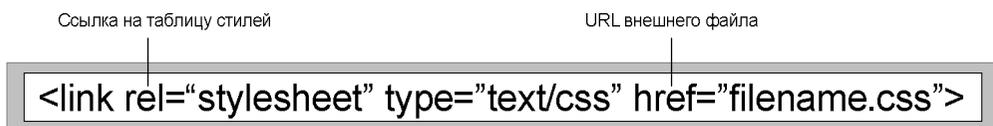


Рис. 3. Общая структура ссылки на внешнюю таблицу стилей

Чтобы внешняя таблица стилей была успешно загружена, тег `link` должен быть размещен внутри элемента `HEAD`. Атрибут `rel` отвечает за установку взаимосвязи и имеет значение `stylesheet`. Атрибут `type` описывает тип данных, которые будут загружаться с помощью тега `link`. Далее расположен атрибут `href`, значением этого атрибута является URL (Uniform Resource Locator) таблицы стилей. URL может быть абсолютным (`http://www.stylesheet.com/filename.css`) или относительным (`filename.css`).

С документом может быть ассоциирована не одна связанная таблица стилей:

```
<link rel = "stylesheet" type="text/css" href="sheet1.css" />  
<link rel = "stylesheet" type="text/css" href="sheet2.css" />
```

Альтернативная таблица стилей. Одной из наиболее весомых причин в пользу использования связанных через `<link>` таблиц стилей является возможность использования альтернативных стилей для клиентов. Такие браузеры, как Firefox и Opera поддерживают атрибут `rel="alternate stylesheet"` и, когда есть возможность, позволяют пользователям переключаться между таблицами стилей.

Элемент style. Элемент `style` — это единственный способ включение таблиц стилей, которые располагаются в самом документе:

```
<style type="text/css">
```

Далее следуют один или несколько стилей, и все завершается закрывающим тегом `</style>`.

Директива @import. Данная директива находится внутри тега `style`. На рис.4 показана общая структура команды `@import`.



Рис. 4. Общий синтаксис импортирования внешней таблицы стилей

Единственное основное отличие заключается в синтаксисе и размещении команды. Директива `@import` должна располагаться в контейнере `style` и пред всеми остальными правилами CSS:

```
<style type="text/css">  
@import url(sheet2.css);  
@import url(http://www.stylesheet.com/sheet.css);  
h1 {color: green;}  
</style>
```

С точки зрения стандартов нет разницы между ссылкой на внешние таблицы стилей или их импортом.

Задание 1. Приведите альтернативную запись, но в более компактной форме, напишите HTML-файл (example_1.html), к которому применяется данная запись таблицы стилей, сформируете аналогичный HTML-результат.

```
h1 {color:gray;}
h2 {color:gray;}
h3 {color:gray;}
h4 {color:gray;}
h5 {color:gray;}
h6 {color:gray;}
h7 {color:gray;}
```

Задание 2. Сочетая в одном правиле оба типа группировки (селекторов и объявлений) определите стиль документа посредством одного выражения, напишите HTML-файл (example_2.html), к которому применяется данная запись таблицы стилей, сформируете аналогичный HTML-результат.

```
h1 {color: gray;}
h2 {color: gray;}
h3 {color: gray;}
h4 {color: gray;}
h5 {color: gray;}
h6 {color: gray;}
h1 {background: white;}
h2 {background: white;}
h3 {background: white;}
h4 {background: white;}
h5 {background: white;}
h6 {background: white;}
```

Задание 3. Выполните две свои альтернативные группировки селекторов, чтобы все группы правил оставались эквивалентными, напишите HTML-файл (example_3.html), к которому применяется данная запись таблицы стилей, сформируете аналогичный HTML-результат.

```
/* группа 1 */
h1 {color: silver; background: white;}
h2 {color: silver; background: gray;}
h3 {color: white; background: gray;}
h4 {color: silver; background: white;}
b {color: gray; background: white;}
```

```
/* группа 2 */
/* группа 3 */
```

Задание 4. Определите селектор собственного класса. Для этого введите имя класса с предваряющей его точкой (.). Затем откройте определение с помощью фигурной скобки ({}). Имя класса может быть любым, но оно должно содержать только буквы и цифры. Напишите шесть любых определений данного класса (в файле example_4.html) и убедитесь, что они разделены точкой с запятой. Завершите правило закрывающейся фигурной скобкой.

Задание 5. Сопоставьте множественному селектору класса: p.class1.class2 {background: red;} соответствующий HTML-код в файле example_5.html.

Задание 6. С помощью CSS определите стиль и вид текста, представленного на рис. 6, напишите HTML-файл (example_6.html), к которому применяется данная запись таблицы стилей, сформируете аналогичный HTML-результат.

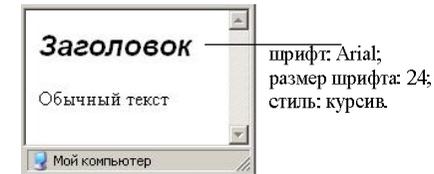


Рис.6. Результат ссылки на таблицу стилей

Задание 7. Необходимо чтобы все элементы, атрибут class которых имеет значение bold, были выделены полужирным шрифтом, элементы, атрибут class которых имеет значение Italic, были выделены курсивом, а элементы абзаца, имеющие значение silver, получили серебряный фон.

Основные параметры шрифта

Свойство	Значение	Описание	Пример
font-family	имя шрифта	задает список шрифтов	p {font-family: Arial, serif}
font-style	normal italic oblique	нормальный шрифт курсив Наклонный шрифт	p {font-style: italic}
font-variant	normal small-caps	капитель (особые прописные буквы)	P {font-variant: small-caps}
font-size	normal pt px %	нормальный размер пункты пиксели проценты	font-size: normal font-size: 12pt font-size: 12px font-size: 120%

Параметры для управления видом текста

Свойство	Значение	Описание	Пример
text-decoration	none underline overline line-through blink	убрать все оформление подчеркивание линия над текстом перечеркивание мигание текста	text-decoration: none
text-transform	none capitalize uppercase lowercase	убрать все эффекты начинать С Прописных ВСЕ ПРОПИСНЫЕ все строчные	text-transform: capitalize
text-align	left; right; center; justify	выравнивание текста	text-align: justify выравнивание по ширине
text-indent	точно %	отступ первой строки	text-indent: 15px; text-indent: 10%

Параметры абзаца

Свойство	Значение	Описание	Пример
text-indent:	number	отступ красной строки	text-indent: 5px;
padding-left	number	отступ от текста слева	padding-left: 10%
padding-right	number	отступ от текста справа	padding-right: 90px
padding-top	number	отступ от текста сверху	padding-top: 10%;
padding-bottom	number	отступ от текста снизу	padding-bottom: 10%
margin-left	number	отступ от границы слева	margin-left: 20%;
margin-right	number	отступ от границы справа	margin-right: 25%;
margin-top	number	отступ от границы сверху	margin-top: 20%;
margin-bottom	number	отступ от границы снизу	margin-bottom: -7px

Параметры цвета

Свойство	Значение	Описание	Пример
color	цвет элемента	определяет цвет элемента	I {color: green;} p {color: #ff0000}
background-color	Цвет фона элемента	устанавливает цвет фона для элемента	h4 {background-color: yellow;}
background	Цвет фона	свойство для установки параметров фона.	body{background:black;}

Параметры таблицы

Свойство	Значение	Описание	Пример
caption-side	top right left bottom	подпись сверху таблицы (по умолчанию) справа от таблицы слева от таблицы снизу таблицы.	table {caption-side : top}
table-layout	auto fixed	вид таблицы определяется фиксированный вид таблицы	table {table-layout : fixed}
border-spacing	number	интервалы между табличными границами	table {border-spacing: 4px}

Именованные цвета

aqua	fuchsia	lime	olive	red	white	blue	purple
black	gray	maroon	orange	silver	yellow	green	navy

Параметры списков

Свойство	Значение	Описание	Пример
list-style-image	ссылка	устанавливает адрес изображения	list-style-image: url('images/book.gif');
list-style-position	inside outside	маркер является частью текстового блока маркер находится вне пределах текстового блока	list-style-position: inside;
list-style-type	Disc, circle, square, decimal, lower-roman, upper-roman, lower-alpha, upper-alpha	<ul style="list-style-type: none"> • ○ ■ 1, 2, 3, ... i, ii, iii, ... I, II, III, ... a, b, c, ... A, B, C, ... 	list-style-type: square;