### Специальные селекторы в CSS. Соседние, дочерние, контекстные селекторы и селекторы по атрибуту тега

**Комбинаторы в CSS (Соседний, дочерний и контекстный селекторы)**

Комбинаторы — это разновидность *css* селекторов, привязывающие правило стиля к элементу web-страницы на основании его местоположения относительно других элементов.

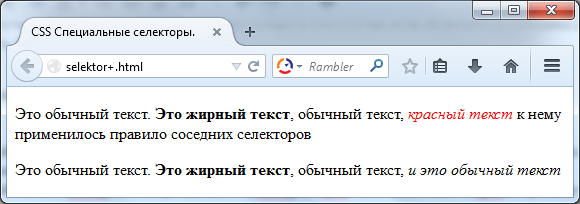
1. Первый комбинатор символ **плюс** (+) или **соседний селектор**. Плюс устанавливается между двумя селекторами:

<селектор 1> + <селектор 2> { <стиль> }

Стиль в этом случае применяется к ***селектору 2*,** но только в том случае если он является соседним для **селектора 1** и идет **сразу после него**. Например:

**strong + i** {  
color: red; /\* Красный цвет текста \*/  
}  
...  
<p>Это обычный текст. <strong>Это жирный текст</strong>, обычный текст, <i>красный текст</i> к нему применилось правило соседних селекторов.</p>  
<p>Это обычный текст. <strong>Это жирный текст</strong>, <span>обычный текст</span>, <i>и это обычный текст</i>.</p>

Результат:



Стиль, описанный в примере будет применен только к первому тексту, заключенному в тег <i>, т.к. он следует сразу же после тега <strong>.

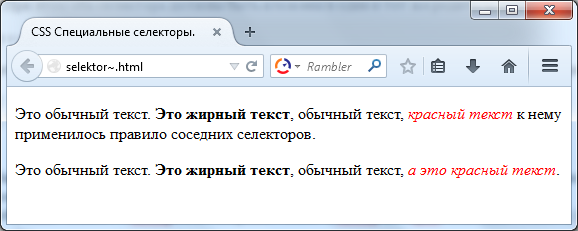
2. Комбинатор **тильда** (~) также применяется к соседним селекторам, но в этот раз *между ними могут быть другие элементы*. При этом оба селектора должны быть вложены в один и тот же родительский тег:

<селектор 1> ~ <селектор 2> { <стиль> }

Стиль будет применен к селектору 2, который должен следовать за селектором 1. Например:

strong ~ i {  
color: red; /\* Красный цвет текста \*/  
}  
...  
<p>Это обычный текст. <strong>Это жирный текст</strong>, обычный текст, <i>красный текст</i> к нему применилось правило соседних селекторов.</p>  
<p>Это обычный текст. <strong>Это жирный текст</strong>, <span>обычный текст</span>, <i>а это красный текст</i>.</p>

Результат:



Как видно, в этот раз правило стиля сработало для обоих текстов заключенных в тег <i>, несмотря на то, что во втором случае между тегом <strong> и <i> стоит тег <span>.

3. Комбинатор **>** относится к **дочерним селекторам**. Позволяет привязать стиль CSS к элементу веб-страницы, непосредственно вложенному в другой элемент:

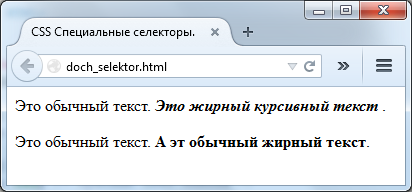
<селектор 1> > <селектор 2> { <стиль> }

Стиль будет привязан к селектору 2, который непосредственно вложен в селектор 1.

Например:

div > strong {  
font-style: italic /\* Курсив \*/  
}  
...  
<div>Это обычный текст. <strong>Это жирный курсивный текст </strong>.  
<p>Это обычный текст. <strong>А это обычный жирный текст</strong>.</p>  
</div>

И результат:



Как видно на рисунке, правило стиля подействовало только на первый тег <strong>, который непосредственно вложен в тег <div>. А непосредственным родителем второго тега <strong> является тег <p>, поэтому правило на него не действует.

4. Следующий **контекстный селектор - <пробел>**. Он позволяет привязать CSS стиль к элементу, вложенному в другой элемент, причем может быть любой уровень вложенности:

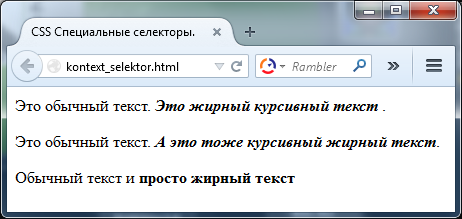
<селектор 1> <селектор 2> { <стиль> }

Стиль будет применен к селектору 2, если он так или иначе вложен в селектор 1.

Например:

div strong {  
font-style: italic /\* Курсив \*/  
}  
...  
<div>Это обычный текст. <strong>Это жирный курсивный текст </strong>.  
<p>Это обычный текст. <strong>А это тоже курсивный жирный текст</strong>.</p>  
</div>  
<p>Обычный текст и <strong>просто жирный текст</strong></p>

Результат:



Как видно, на этот раз правило подействовало на оба тега <strong>, даже на тот, который вложен и в контейнер <div> и в абзац <p>. На тег <strong>, который просто вложен в абзац <p> правило css никак не действует.

**Селекторы по атрибутам тега**

Селекторы атрибутов — это специальные селекторы, которые привязывают стиль к элементу на основании, присутствует ли в нем определенный атрибут или имеет ли он определенное значение. Существует несколько вариантов применения таких селекторов.

**1. Простой селектор атрибута**

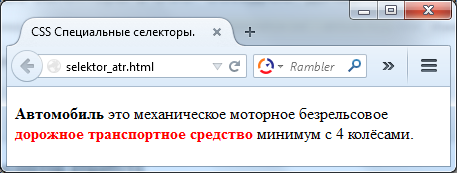
Имеет вид:

<селектор>[<имя атрибута тега>] { <стиль> }

И привязывает стиль к тем элементам, внутри которых добавлен указанный атрибут. Например:

strong[title]{  
color:red;  
}  
...  
<p><strong>Автомобиль</strong> это механическое моторное безрельсовое <strong title="транспортное средство">дорожное транспортное средство</strong> минимум с 4 колёсами.</p>

Результат:



На рисунке можно заметить, что правило css (красный цвет текста) применяется к элементу strong, к которому добавлен атрибут title.

**2. Селектор атрибута со значением**

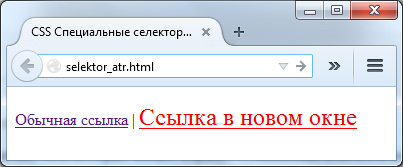
Синтаксис этого селектора следующий:

<селектор>[<имя атрибута тега>=<значение>] { <стиль> }

Привязывает стиль к элементам, теги которых имеют атрибут с указанным именем и значением. Пример:

a[target="\_blank"]{  
color:red;  
font-size:150%;  
}  
...  
<p><a href="http://webcodius.ru" target="\_self">Обычная ссылка</a> | <a href="http://webcodius.ru" target="\_blank">Ссылка в новом окне</a></p>

Результат:



Как видно, оба элемента типа гиперссылка имеют атрибут target, но правило css, которое увеличивает текст ссылки в полтора раза и изменяет его цвет в красный, применяется к тегу <a> у которого атрибут target имеет значение «\_blank».

**3. Одно из нескольких значений атрибута**

Некоторые значения атрибутов могут перечисляться через пробел, например имена классов. Чтобы задать правило стиля при наличии в списке значений атрибутов требуемого значения применяется следующий селектор:

[<имя атрибута тега>~=<значение>] { <стиль> }  
<основной селектор>[<имя атрибута тега>~=<значение>] { <стиль> }

Стиль применяется в том случае, если у атрибута имеется требуемое значение или оно входит в список значений, разделенных пробелами. Например:

[class~="tel"]{  
color:red;  
font-size:150%;  
}  
...  
<p>Наш телефон: <span class="tel block">777-77-77</span></p>  
<p>Наш адрес: <span class="adress block">Кишинев ул. Дечебал 5</span></p>

Правило применяется к элементу, у которого среди значений атрибута class имеется значение tel.

**4. Дефис в значении атрибута**

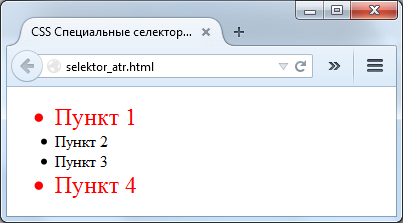
В значениях идентификаторов и классов допускается использовать дефис. Для привязки стиля к элементам, значения атрибутов которых могут содержать дефис можно воспользоваться следующей конструкцией:

[атрибут|="значение"] { стиль }  
Селектор[атрибут|="значение"] { стиль }

Стиль применяется к тем элементам, у которых значение атрибута начинается с указанного значения после, которого идет дефис. Например:

[class|="menu"]{  
color:red;  
font-size:150%;  
}  
...  
<ul>  
<li class="menu-item">Пункт 1</li>  
<li class="item-menu">Пункт 2</li>  
<li class="menuitem">Пункт 3</li>  
<li class="menu-item">Пункт 4</li>  
</ul>

Результат:



В примере правило стиля применяется только к тем элементам списка, у которых имя класса начинается со значения menu.

**5. Значение атрибута начинается с определенного текста**

Данный селектор устанавливает стиль для элемента, если значение атрибута тега начинается с определенного значения. Может быть два варианта:

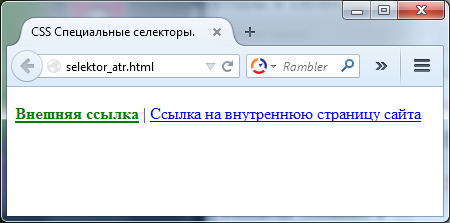
[<имя атрибута тега>^=<подстрока>] { <стиль> }  
<основной селектор>[<имя атрибута тега>^=<подстрока>] { <стиль> }

В первом случае стиль применяется ко всем элементам, теги которых имеют атрибут с указанным именем и значением, начинающимся с указанной подстроки. Во втором случае тоже самое, только к определенным элементам, указанным в основном селекторе.

Пример:

a[href^="http://"]{  
color:green;  
font-weight:bold;  
}  
...  
<p><a href=“http://webcodius.ru»>Внешняя ссылка</a> | <a href="/spravochnik-css/chto-takoe-css-i-kak-podklyuchit-kaskadnye-tablicy-stilej-k-html-dokumentu.html" target="\_blank">Ссылка на внутреннюю страницу сайта</a></p>

Результат:



В примере показано, как по разному отобразить внешние ссылки и внутренние ссылки. Внешние ссылки всегда начинаются со строки «http://». Поэтому в селекторе указываем, что стиль применять только к ссылкам, у которых атрибут href начинается со значения http://.

**6. Значение атрибута заканчивается определенным текстом**

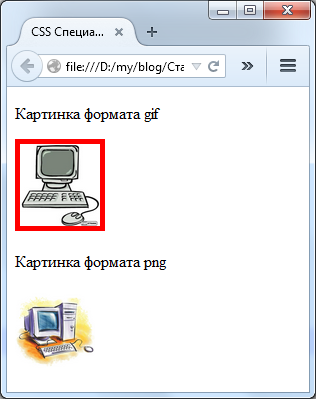
Привязывает стиль к элементам, у которых значение атрибута заканчивается указанным текстом. Имеет следующий синтаксис:

[<имя атрибута тега>$=<подстрока>] { <стиль> }  
<основной селектор>[<имя атрибута тега>$=<подстрока>] { <стиль> }

В первом случае стиль применяется ко всем элементам, у которых имеется атрибут с указанным именем и имеет значение оканчивающееся указанной подстрокой. Во втором случае тоже самое, только к указанным селекторам. Таким способом, например, можно по-разному отображать различные форматы графических изображений. Например:

IMG[SRC$=gif] {  
border: 5px solid red;  
}  
...  
<p>Картинка формата gif</p>  
<img src="1.gif">  
<p>Картинка формата png</p>  
<img src="2.jpg">

Результат:



В примере все картинки с расширением *gif* будут отображаться с красной рамкой толщиной пять пикселей.

**7. Значение атрибута содержит указанную строку**

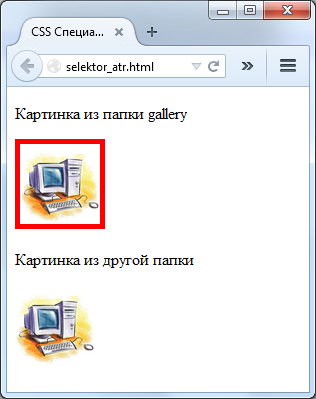
Этот селектор привязывает стиль к тегам, значение атрибута которых содержат определенный текст. Синтаксис:

[<имя атрибута тега>\*=<подстрока>] { <стиль> }  
<основной селектор>[<имя атрибута тега>\*=<подстрока>] { <стиль> }

Стиль привязывается к элементам, у которых имеется атрибут с указанным именем и его значение содержит указанную подстроку. Например:

IMG[SRC\*=gallery/] {  
border: 5px solid red;  
}  
...  
<p>Картинка из папки gallery</p>  
<img src="gallery/1.jpg">  
<p>Картинка из другой папки</p>  
<img src="2.jpg">

Результат:



В примере стиль применяется к картинкам, которые загружены из папки «gallery».