



Лекция 7

Каскадные таблицы стилей (CSS). Селекторы. Псевдоклассы. Псевдоэлементы.

Introduction to Web Development

План лекции

- Виды селекторов
- Селекторы атрибутов
- Псевдоклассы
- Псевдоэлементы
- Некоторые CSS свойства

Виды селекторов

В качестве селектора может выступать **любой тег HTML** для которого **определяются правила форматирования**, такие как: **цвет, фон, размер и т.д.**

Тег { свойство1: значение; свойство2: значение; ... }

Контекстные селекторы (селекторы потомков)

Для того, чтобы одновременно установить стиль для отдельного тега, а также для тега, который находится внутри другого.

Контекстный селектор состоит из **простых селекторов** разделенных пробелом:

Тег1 Тег2 { ... }

Стиль будет применяться к **Тегу2** когда он размещается внутри **Тега1**.

Виды селекторов

Часто приходится **вкладывать одни теги внутрь других**. Чтобы стили для этих тегов использовались корректно, **помогут селекторы, которые работают только в определенном контексте**.

Например, задать стиль для тега `` только когда он располагается внутри контейнера `<p>`.

Таким образом можно одновременно установить стиль для отдельного тега, а также для тега, который находится внутри другого.

Виды селекторов

Соседние селекторы

Для управления стилем **соседних элементов** используется символ плюса (+), который устанавливается между двумя селекторами:

Селектор 1 + Селектор 2 { Описание правил стиля }

Стиль при такой записи применяется к Селектору 2, но только в том случае, если он является **соседним** для Селектора 1 и **следует сразу после него**.

Дочерние селекторы

Дочерним селектором считается такой, который в дереве элементов находится **прямо внутри родительского элемента**.

Селектор 1 > Селектор 2 { Описание правил стиля }

Стиль применяется к Селектору 2, но только в том случае, если он является **дочерним** для Селектора 1.

Виды селекторов

По своей логике дочерние селекторы похожи на селекторы контекстные.
Разница между ними следующая:

- Стиль к дочернему селектору применяется только в том случае, когда он является **прямым потомком**, иными словами, **непосредственно располагается внутри родительского элемента**.
- Для контекстного селектора допустим **любой уровень вложенности**.

Виды селекторов

Универсальный селектор

Применяется, чтобы установить **одновременно один стиль для всех элементов веб-страницы**, например, задать шрифт или начертание текста.

Для обозначения универсального селектора применяется символ звездочки:

*** { Описание правил стиля }**

В некоторых случаях указывать универсальный селектор не обязательно. Так, например, записи *.class и .class являются идентичными по своему результату.



Селекторы атрибутов

Селекторы атрибутов позволяют установить стиль по присутствию определенного атрибута тега или его значения.

Простой селектор атрибута

Устанавливает стиль для элемента, если задан специфичный атрибут тега. Его значение в данном случае не важно.

[атрибут] { Описание правил стиля }
Селектор[атрибут] { Описание правил стиля }

Стиль применяется к тем тегам, внутри которых добавлен указанный атрибут.

```
<style type="text/css">
  Q {
    font-style: italic; /* Курсивное начертание */
    quotes: "\00AB" "\00BB"; /* Меняем вид кавычек в цитате */
  }
  Q[title] {
    color: maroon; /* Цвет текста */
  }
</style>
```


Селекторы атрибутов

Атрибут со значением

Устанавливает стиль для элемента в том случае, **если задано определенное значение специфичного атрибута.**

[атрибут="значение"] { Описание правил стиля }
Селектор[атрибут="значение"] { Описание правил стиля }

В первом случае стиль применяется ко всем тегам, которые содержат указанное значение. А во втором — только к определенным селекторам.

```
<title>Селекторы атрибутов</title>
<style type="text/css">
  A[target="_blank"] {
    background: url(blank.gif) no-repeat 0 2px; /* Параметры фонового рисунка */
    padding-left: 15px; /* Смещаем текст вправо */
  }
</style>
</head>
<body>
  <p><a href="link1.html">Обычная ссылка</a> |
    <a href="link2" target="_blank">Ссылка в новом окне</a></p>
</body>
```

Селекторы атрибутов

Значение атрибута начинается с определенного текста

Устанавливает стиль для элемента в том случае, если значение атрибута тега начинается с указанного текста.

[атрибут^="значение"] { Описание правил стиля }
Селектор[атрибут^="значение"] { Описание правил стиля }

В первом случае стиль применяется ко всем элементам, у которых значение атрибута начинаются с указанного текста. А во втором — только к определенным селекторам.

```
<style type="text/css">
  A[href^="http://"] {
    font-weight: bold; /* Жирное начертание */
  }
</style>
</head>
<body>
  <p><a href="link1.html">Обычная ссылка</a> |
  <a href="http://ya.ru" target="_blank">Внешняя ссылка на сайт ya.ru</a></p>
</body>
</html>
```

Селекторы атрибутов

Значение атрибута оканчивается определенным текстом

Устанавливает стиль для элемента в том случае, если значение атрибута оканчивается указанным текстом.

[атрибут\$="значение"] { Описание правил стиля }
Селектор[атрибут\$="значение"] { Описание правил стиля }

В первом случае стиль применяется ко всем элементам у которых значение атрибута завершается заданным текстом. А во втором — только к определенным селекторам.

```
<style type="text/css">
  A[href$=".ru"] { /* Если ссылка заканчивается на .ru */
    background: url(ru.gif) no-repeat; /* Параметры фонового рисунка */
    padding-left: 10px; /* Смещаем текст вправо */
  }
  A[href$=".com"] { /* Если ссылка заканчивается на .com */
    background: url(com.gif) no-repeat;
    padding-left: 10px;
  }
</style>
</head>
<body>
  <p><a href="http://www.yandex.com">Yandex.Com</a> |
  <a href="http://www.yandex.ru">Yandex.Ru</a></p>
```

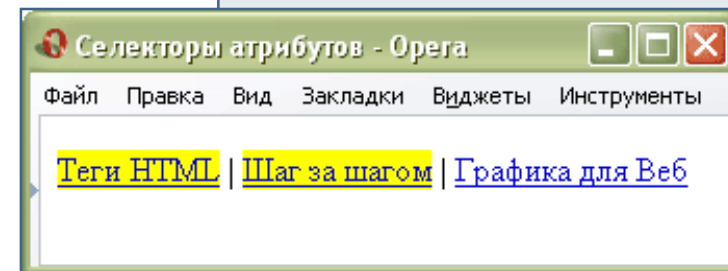
Селекторы атрибутов

Значение атрибута содержит указанный текст

Возможны варианты, когда стиль следует применить к тегу с определенным атрибутом, при этом частью его значения является некоторый текст. При этом точно не известно, в каком месте значения включен данный текст — в начале, середине или конце.

[атрибут*="значение"] { Описание правил стиля }
Селектор[атрибут*="значение"] { Описание правил стиля }

```
<style type="text/css">
  [href*="htmlbook"] {
    background: yellow; /* Желтый цвет фона */
  }
</style>
</head>
<body>
  <p><a href="http://www.htmlbook.ru/html/">Теги HTML</a> |
  <a href="http://stepbystep.htmlbook.ru">Шаг за шагом</a> |
  <a href="http://webimg.ru">Графика для Веб</a></p>
</body>
</html>
```



Селекторы атрибутов

Одно из нескольких значений атрибута

Некоторые значения атрибутов могут перечисляться через пробел, например имена классов. Чтобы **задать стиль при наличии в списке требуемого значения** применяется следующий синтаксис:

```
[атрибут~="значение"] { Описание правил стиля }  
Селектор[атрибут~="значение"] { Описание правил стиля }
```

Стиль применяется в том случае, если у атрибута имеется указанное значение или оно входит в список значений, разделяемых пробелом.

```
<style type="text/css">  
  [class~="block"] h3 { color: green; }  
</style>  
</head>  
<body>  
  <div class="block tag">  
    <h3>Заголовок</h3>  
  </div>
```

Селекторы атрибутов

Дефис в значении атрибута

Для изменения стиля элементов, в значении которых применяется дефис, следует воспользоваться следующим синтаксисом:

[атрибут]="значение" { Описание правил стиля }
Селектор[атрибут]="значение" { Описание правил стиля }

Стиль применяется к элементам, у которых атрибут начинается с указанного значения или с фрагмента значения, после которого идет дефис.

```
<style type="text/css">
  DIV[class="block"] {
    background: #306589; /* Цвет фона */
    color: #acdb4c; /* Цвет текста */
    padding: 5px; /* Поля */
  }
  DIV[class="block"] A {
    color: #fff; /* Цвет ссылок */
  }
</style>
</head>
<body>
  <div class="block-menu-therm">
    <h2>Термины</h2>
    <div class="content">
      <ul class="menu">
        <li><a href="t1.html">Буквица</a></li>
```

Псевдоклассы

определяют динамическое состояние элементов, которое изменяется с помощью действий пользователя, а также положение в дереве документа.

Селектор:Псевдокласс { Описание правил стиля }

Пример: текстовая ссылка, которая меняет свой цвет при наведении на неё курсора мыши.

При использовании **псевдоклассов** браузер **не перегружает** текущий документ, поэтому с помощью псевдоклассов можно получить разные **динамические эффекты** на странице.

Условно все псевдоклассы делятся на три группы:

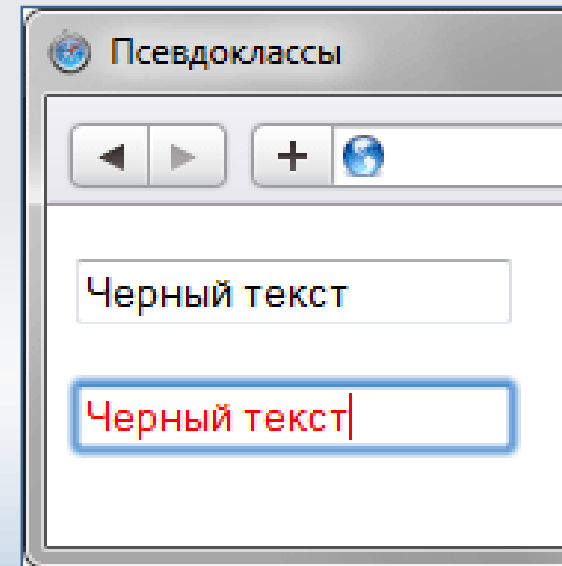
- **определяющие состояние элементов;**
- **имеющие отношение к дереву элементов;**
- **указывающие язык текста.**

Псевдоклассы

:focus

Применяется к элементу **при получении** им **фокуса**. Например, для текстового поля формы получение фокуса означает, что курсор установлен в поле, и с помощью клавиатуры можно вводить в него текст.

```
<style>
  INPUT:focus {
    color: red; /* Красный цвет текста */
  }
</style>
</head>
<body>
  <form action="">
    <p><input type="text" value="Черный текст"></p>
    <p><input type="text" value="Черный текст"></p>
  </form>
```



Псевдоклассы

:active

Происходит **при активации пользователем элемента**. Например, ссылка становится активной, если навести на неё курсор и щёлкнуть мышкой. Несмотря на то, что активным может стать практически любой элемент веб-страницы, псевдокласс :active **используется преимущественно для ссылок**.

:link

Применяется к **непосещенным ссылкам**, т. е. таким ссылкам, на которые пользователь ещё не нажимал. Браузер некоторое время сохраняет историю посещений, поэтому ссылка может быть помечена как посещенная хотя бы потому, что по ней был зафиксирован переход ранее.

Псевдоклассы

:visited

Данный псевдокласс применяется к **посещённым ссылкам**. Обычно такая ссылка меняет свой цвет по умолчанию на фиолетовый, но с помощью стилей цвет и другие параметры можно задать самостоятельно.

:hover

Псевдокласс :hover активизируется, когда **курсор мыши находится в пределах элемента, но щелчка по нему не происходит**.

a:hover должен быть расположен после **a:link** и **a:visited**!
a:active должен быть расположен после **a:hover**!
Псевдоклассы не чувствительны к регистру.

Псевдоклассы

```
a {  
    color: red; /* Цвет непосещенных ссылок */  
}  
  
a:visited {  
    color: blue; /* Цвет посещенных ссылок */  
}  
  
a:hover {  
    color: grey; /* Цвет ссылок при наведении на них курсора мыши */  
}  
  
a:active {  
    color: black; /* Цвет активных ссылок */  
}
```

Псевдоклассы

:first-child

Применяется к **первому дочернему элементу селектора**, который расположен в дереве элементов документа.

```
p:first-child {  
  color: blue;  
}
```

```
<body>  
<div>  
  <p>Первый абзац</p>  
  <p>Второй абзац</p>  
  <p>Третий абзац</p>  
</div>  
  
<div>  
  <p>Еще один абзац, но в другом блоке</p>  
</div>  
</body>
```

Первый абзац

Второй абзац

Третий абзац

Еще один абзац, но в другом блоке

Псевдоклассы

:lang

Определяет язык, который используется в документе или его фрагменте. В коде HTML язык устанавливается через атрибут lang, он обычно добавляется к тегу <html>. С помощью псевдокласса :lang можно задавать определённые настройки, характерные для разных языков, например, вид кавычек в цитатах.

В качестве языка могут выступать следующие значения: **ru** — русский; **en** — английский ; **de** — немецкий ; **fr** — французский; **it** — итальянский.

Больше псевдо-классов:

http://www.w3schools.com/css/css_pseudo_classes.asp

Псевдоэлементы

Псевдоэлемент	Пример
<u>::after</u>	p::after
<u>::before</u>	p::before
<u>::first-letter</u>	p::first-letter
<u>::first-line</u>	p::first-line

Псевдоэлементы

Селектор::Псевдоэлемент { Описание правил стиля }

::after

Применяется для вставки назначенного контента после содержимого элемента. Этот псевдоэлемент работает совместно со стилевым свойством **content**, которое определяет содержимое для вставки.

```
<p>HyperText Markup Language</p>
<p>Cascading Style Sheets</p>
```

```
<p><b>Важно:</b> Чтобы этот псевдоэлемент работал в IE8, DOCTYPE должен быть указан, <br>
и нужно использовать старый синтаксис :after вместо ::after.</p>
```

```
<style>
p::after {
    content: " - Читать далее...";
    background-color: yellow;
    color: red;
    font-weight: bold;
}
</style>
```

Псевдоэлементы

HyperText Markup Language - Читать далее...

Cascading Style Sheets - Читать далее...

Важно: Чтобы этот псевдоэлемент работал в IE8, DOCTYPE должен быть указан, и нужно использовать старый синтаксис :after вместо ::after. - Читать далее...

::before

По своему действию **::before** аналогичен псевдоэлементу **::after**, но вставляет контент **до** содержимого элемента.

Псевдоэлементы

::first-letter

Определяет **стиль первого символа в тексте элемента**, к которому добавляется. Это позволяет создавать в тексте буквицу и выступающий инициал.

::first-line

Определяет **стиль первой строки блочного текста**. Длина этой строки зависит от многих факторов, таких как используемый шрифт, размер окна браузера, ширина блока, языка и т.д.

CSS СВОЙСТВА

color

```
p{  
  color: red; /*a valid color name (color names are case-insensitive: Red=RED=red)*/  
  color: rgb(255, 0, 0); /*an RGB(red, green, blue) value*/  
    /*Each parameter (red, green, blue) defines the intensity of the color  
    between 0 and 255.*/  
  color: #ff0000; /*a HEX value - #RRGGBB,  
    where RR (red), GG (green) and BB (blue) are hexadecimal values  
    between 00 and FF (same as decimal 0-255).  
    Case-insensitive*/  
}
```

Отдавайте предпочтение HEX значениям

CSS СВОЙСТВА

background

```
body{  
    background-color: lightblue;  
    background-image: url("micro.png");  
    background-repeat: repeat-x; /*Repeat Horizontally*/  
    background-position: right top; /*background position*/  
    background-attachment: fixed; /*will not scroll with the rest of the page*/  
}
```

Или так:

```
body {  
    background: #ffffff url("img_tree.png") no-repeat right top;  
}
```

CSS СВОЙСТВА

Для форматирования текста

text-decoration

Значения: overline; line-through; underline; none;

```
h1 {  
    text-decoration: overline;  
}  
h2 {  
    text-decoration: line-through;  
}  
h3 {  
    text-decoration: underline;  
}
```

Пример 1

~~Пример 2~~

Пример 3

CSS СВОЙСТВА

text-transform

Для форматирования текста

Значения: uppercase; lowercase; capitalize; none;

```
p.uppercase {  
    text-transform: uppercase;  
}  
p.lowercase {  
    text-transform: lowercase;  
}  
p.capitalize {  
    text-transform: capitalize;  
}
```

ЭТО ВЕРХНИЙ РЕГИСТР

это нижний регистр

Все Слова Начинаются С Заглавной Буквы

CSS СВОЙСТВА

text-indent

```
p {  
  text-indent: 30px;  
}
```

letter-spacing

```
h1 {  
  letter-spacing: 3px;  
}  
h2 {  
  letter-spacing: -3px;  
}
```

Для форматирования текста

Отступ справа первой линии абзаца

Это абзац с отступом первой строки. Установлен с помощью text-indent. Это абзац с отступом первой строки. Установлен с помощью text-indent. Это абзац с отступом первой строки. Установлен с помощью text-indent. Это абзац с отступом первой строки. Установлен с помощью text-indent.

Уплотненность/разреженность текста

This is heading 1

This is heading 2

CSS СВОЙСТВА

line-height

Для форматирования текста

Межстрочный интервал

```
p.small {  
    line-height: 0.7;  
}  
p.big {  
    line-height: 1.8;  
}
```

This is a paragraph with a standard line-height.
The default line height in most browsers is about 110% to 120%.

This is a paragraph with a smaller line-height.
This is a paragraph with a smaller line-height.

This is a paragraph with a bigger line-height.
This is a paragraph with a bigger line-height.

CSS свойства

Для форматирования текста

word-wrap

указывает, переносить или нет длинные слова, которые не помещаются по ширине в заданную область.

```
word-wrap: normal | break-word | inherit
```

word-break

указывает, как делать перенос строк внутри слов, которые не помещаются по ширине в заданную область (перенос с любого символа)

```
word-break: normal | break-all | keep-all
```


CSS свойства

text-overflow

Для форматирования текста

Если текст целиком не помещается в заданную область:

- текст обрезаается;
- текст обрезаются и к концу строки добавляется многоточие.

text-overflow работает в том случае, если для блока значение свойства overflow установлено как auto, scroll или hidden.

```
text-overflow: clip | ellipsis
```

CSS СВОЙСТВА

Иконки

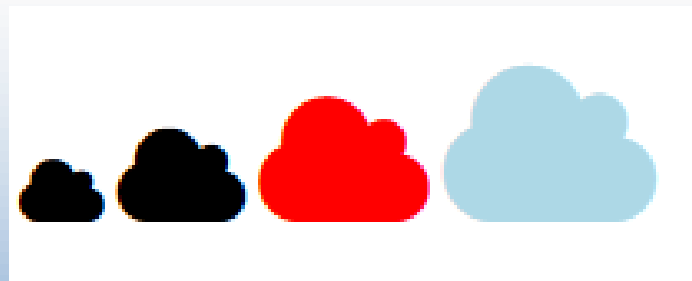
```
<link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">
```

Указываем нужный класс (ищем здесь: <http://fontawesome.io/icons/>)

```
<i class="fa fa-cloud"></i>
<i class="fa fa-heart"></i>
<i class="fa fa-car"></i>
<i class="fa fa-file"></i>
<i class="fa fa-bars"></i>
```



```
<i class="fa fa-cloud" style="font-size:24px;"></i>
<i class="fa fa-cloud" style="font-size:36px;"></i>
<i class="fa fa-cloud" style="font-size:48px;color:red;"></i>
<i class="fa fa-cloud" style="font-size:60px;color:lightblue;"></i>
```



Рекомендованная литература:

- <https://www.w3.org/>
- <https://www.w3schools.com/css/default.asp>
- <https://webref.ru/css>
- <https://www.codecademy.com>
- <https://htmlacademy.ru/>
- Новая большая книга CSS

и...

<http://www.google.com/>