

Псевдоклассы и Псевдоэлементы

Обычно стили CSS применяются к тем элементам веб-страницы, которые видны в ее исходном коде. Но существуют случаи, когда необходимо создать стиль для определенного состояния элемента (например, внешний вид посещенной ссылки либо вид ссылки при наведенном на нее курсоре), а также для элемента, который четко не обозначен в структуре страницы. Примером такого элемента может быть первый символ в абзаце или первая строка.

С помощью псевдоклассов в CSS можно устанавливать стиль для уже существующих элементов веб-страницы, который будет применяться в случае каких-то действий пользователя. Псевдоэлементы же отличаются тем, что могут определять стиль несуществующего содержимого, а также четко не обозначенных элементов.

Если псевдоклассы записываются с одним двоеточием, то псевдоэлементы – с двумя. Это было внедрено в CSS3 для того, чтобы различать псевдоклассы и псевдоэлементы между собой. Однако раньше этой разницы не существовало и с псевдоэлементами использовалось одно двоеточие. Сейчас браузеры поддерживают оба варианта написания (но не для всех случаев).

Псевдо-класс **:first-child**

Псевдо-класс **:first-child** позволяет выбрать элемент, который является первым потомком в его родительском элементе.

```
<html>
<head>
<style type='text/css'>
p:first-child
{
color:red;
}
</style>
</head>
<body>
<p>Это первый абзац, поэтому он будет оформлен.</p>
<p>Это второй абзац он не будет оформлен.</p>
<p>Это третий абзац он тоже не будет оформлен.</p>
</body>
</html>
```

Псевдокласс **:nth-child**

используется для добавления стиля к элементам на основе нумерации в дереве элементов.

Значения

odd - Все нечетные номера элементов.

even - Все четные номера элементов.

число - Порядковый номер дочернего элемента относительно своего родителя. Нумерация начинается с 1, это будет первый элемент в списке.

```
<html>
<head>
  <meta charset="utf-8">
  <title>nth-child</title>
  <style>
    table {
      width: 100%;
      border-spacing: 0;
    }
    tr:nth-child(even) {
      background: #f0f0f0;
    }
    tr:nth-child(1) {
      background: #666;
      color: #fff;
    }
  </style>
</head>
<body>
  <table border="1">
    <tr>
      <td>&nbsp;</td><td>2134</td><td>2135</td>
      <td>2136</td><td>2137</td><td>2138</td>
    </tr>
    <tr>
      <td>1</td><td>16</td><td>34</td>
      <td>62</td><td>74</td><td>57</td>
    </tr>
    <tr>
      <td>2</td><td>4</td><td>69</td>
      <td>72</td><td>56</td><td>47</td>
    </tr>
  </table>
</body>
</html>
```

```

</tr>
<tr>
  <td>3</td><td>7</td><td>73</td>
  <td>79</td><td>34</td><td>86</td>
</tr>
<tr>
  <td>4</td><td>23</td><td>34</td>
  <td>88</td><td>53</td><td>103</td>
</tr>
</table>
</body>
</html>

```

Псевдокласс **:nth-last-child**

используется для добавления стиля к элементам на основе нумерации в дереве элементов. В отличие от псевдокласса `:nth-child` отсчет ведется не от первого элемента, а от последнего.

Псевдокласс **:last-child**

задает стилевое оформление последнего элемента своего родителя

```

<html>
<head>
  <meta charset="utf-8">
  <title>last-child</title>
  <style>
    .block {
      background: #7da7d9;
      color: #fff;
    }
    .block :first-child {
      padding: 10px;
    }
    .block :last-child {
      background: #dda458 url(image.png) repeat-x;
      height: 5px;
    }
  </style>
</head>
<body>
  <div class="block">
    <div>

```

При истечении возможности понимания вышеизложенной информации вы имеете объективную возможность подать официальный запрос

```
</div>
<div></div>
</div>
</body>
</html>
```

Псевдо-класс **:first-letter**

Псевдо-класс **:first-letter** позволяет оформить первую букву указанного элемента.

```
<html>
<head>
<style type='text/css'>
p:first-letter
{
font-size:1.8em;
color:green;
font-family:'Comic Sans MS';
border:1px solid black;
padding:2px;
}
p
{
font-family:'Comic Sans MS';
}
</style>
</head>
<body>
<p>Я первый параграф.</p>
<p>Я второй параграф.</p>
</body>
</html>
```

Псевдо-класс **:focus**

Псевдо-класс **:focus** позволяет выбрать активные input элементы формы.

```
<html>
```

```

<head>
<style type='text/css'>
:focus
{
background-color:yellow;
}
</style>
</head>
<body>
<form>
Введите Ваше имя: <input type='text' />
</form>
<p><b>Обратите внимание:</b> щелкните в текстовое поле, чтобы
сделать его активным.</p>
</body>
</html>

```

Псевдокласс **:required**

Применяет стилевые правила к тегу `<input>`, у которого установлен атрибут `required`. Он позволяет выделять поля обязательные к заполнению перед отправкой формы.

```

<html>
<head>
<meta charset="utf-8">
<title>Комментарии</title>
<style>
#comment p {
margin: 0 0 5px;
}
#comment label {
width: 110px;
float: left;
text-align: right;
padding-right: 10px;
}
#comment input[type="text"], #comment textarea {
width: 320px;
}
input:required {
border: 1px solid red; /* Красная рамка для обязательных полей*/
}
</style>

```

```

</head>
<body>
  <form id="comment">
    <p><label for="user">Имя</label>
      <input id="user" type="text" required></p>
    <p><label for="email">E-mail</label><input id="email"
type="text"></p>
    <p><label for="text">Комментарий</label>
      <textarea cols="40" rows="10" id="text"></textarea></p>
    <p><label>&nbsp;</label><input type="submit" id="send"
value="Отправить"></p>
  </form>
</body>
</html>

```

Псевдо-класс **:first-line**

Псевдо-класс **:first-line** позволяет оформить первую строчку указанного элемента.

```

<html>
<head>
<style type='text/css'>
p:first-line
{
font-size:1.2em;
font-weight:bold;
color:green;
}
</style>
</head>
<body>
<p>Мост — искусственное сооружение, возведенное через реку,
озеро, пролив или любое другое водное препятствие. Мост,
возведенный через дорогу, называют путепроводом, мост через
овраг или ущелье — виадуком.</p>
<p>Мост является одним из древнейших инженерных изобретений
человечества.</p>
</body>
</html>

```

Псевдо-классы **:before** и **:after**

С помощью псевдо-классов **:before** и **:after** Вы можете вставлять произвольное содержимое до и после указанных элементов.

```
<html>
<head>
<style type='text/css'>
p:before
{
content:'CSS - ';
font-weight:bold;
}
p:after
{
content:' HTML документы.';
}
</style>
</head>
<body>
<p>позволяет оформлять</p>
</body>
</html>
```

Псевдо-класс **:link**

a:link Выберет все не посещенные ссылки на странице.

a:visited Выберет все посещенные ссылки на странице.

a:active Выберет все активные в данный момент ссылки (на которые щелкнули мышкой).

Псевдокласс **:not(селектор)**

задаёт правила стилей для элементов, которые не содержат указанный селектор.

```
<html>
<head>
<meta charset="utf-8">
<title>not</title>
<style>
input:not([type="submit"]) {
border: 1px solid #ccc;
padding: 3px;
}
</style>
</head>
```

```

<body>
  <form>
    <p>Ваше имя: <input name="user"></p>
    <p>Прилагаемый файл: <input type="file" name="file"></p>
    <p><input type="submit" value="Отправить"></p>
  </form>
</body>
</html>

```

Псевдокласс **:target**

Псевдокласс `target` задает то, как будет выглядеть целевой элемент. Целевой элемент - это тот, на `id` которого мы ссылаемся, указывая в конце урла его `id` после решетки `#` (пример: `/css/target.html#my-id`).

```

<html>
  <head>
    <meta charset="utf-8">
    <title>target</title>
    <style>
      h2:target {
        background: #fc0;
        padding: 3px;
      }
    </style>
  </head>
  <body>
    <ul>
      <li><a href="#h1">История 1</a></li>
      <li><a href="#h2">История 2</a></li>
    </ul>
    <h2 id="h1">История 1</h2>
    <p>История о том</p>
    <h2 id="h2">История 2</h2>
    <p>История о том</p>
  </body>
</html>

```

Псевдокласс **:empty**

Псевдокласс `empty` задает то, как будет выглядеть пустой элемент

```

<ul>
  <li>пункт</li>
  <li>пункт</li>
  <li>пункт</li>

```



```
<li>пункт</li>
<li></li>
</ul>
li:empty {
  border: 1px solid blue;
  list-style-type: circle;
  color: blue;
}
```

Псевдоэлемент **::selection**

Псевдоэлемент selection позволяет задать стили выделенному тексту (только цвет и фон).

```
p::selection {
  color: red;
  background-color: black;
}
```

Псевдоэлемент **::placeholder**

Если цвет подсказки placeholder не задан через псевдоэлемент placeholder, то он будет такой же, какой задан для всего элемента через свойство color, но более бледный, так как по умолчанию браузер добавляет полупрозрачность подсказке placeholder. Отменить ее можно через свойство opacity для псевдоэлемента placeholder (не для элемента!).

```
<input type="text" placeholder="подсказка:">
```

```
input::placeholder {
  color: red;
  opacity: 1;
}
```

Псевдокласс **:checked**

Псевдокласс checked позволяет поймать отмеченный чекбокс или radio. К сожалению, стили для их украшения задать нельзя. К примеру нельзя задать фон или цвет границы (можно задавать, к примеру, ширину и высоту, но они окажут не такое влияние, как можно было бы ожидать).

Несмотря на это, можно задать стили для соседних элементов, например, так: input:checked+p - зададим стили для абзаца, который находится непосредственно под отмеченным инпутом.

```
<input type="checkbox" checked id="checkbox1"><label  
for="checkbox1">Первый чекбокс</label><br>  
<input type="checkbox" id="checkbox2"><label  
for="checkbox2">Второй чекбокс</label><br>  
<input type="radio" id="radio1" name="radio"><label  
for="radio1">Радио</label>  
<input type="radio" id="radio2" name="radio"><label  
for="radio2">Радио</label>
```

```
input:checked+label {  
    color: red;  
}
```

Псевдокласс **:disabled**

Псевдокласс disabled задает стили для неактивного инпута (которому задан атрибут disabled), а enabled - для активного (у которого нет атрибута disabled).

```
input:disabled {  
    border: 1px solid red;  
    box-shadow: 0px 0px 3px red;  
}  
input:enabled {  
    border: 1px solid green;  
    box-shadow: 0px 0px 3px green;  
}
```

Скроллбар CSS

-webkit-scrollbar состоит различных псевдо-элементов.

::-webkit-scrollbar — это фон самого скроллбара.

::-webkit-scrollbar-button — кнопки направления на полосе прокрутки.

::-webkit-scrollbar-track — пустое пространство под индикатором прокрутки.

::-webkit-scrollbar-thumb — индикатор прокрутки, перетаскиваемый элемент.

Реализуйте пример:

HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width,
initial-scale=1.0">
  <title>Document</title>
<style>
body {
  overflow-y: scroll;
}

.overflow {
  height: 4000px;
}

::-webkit-scrollbar-button {
background-image:url("");
background-repeat:no-repeat;
width:5px;
height:0px
}

::-webkit-scrollbar-track {
background-color:#ecedee
}

::-webkit-scrollbar-thumb {
-webkit-border-radius: 0px;
border-radius: 0px;
background-color:#6dc0c8;
}

::-webkit-scrollbar-thumb:hover{
background-color:#56999f;
}

::-webkit-resizer{
```

```
background-image:url("");
background-repeat:no-repeat;
width:4px;
height:0px
}
```

```
::-webkit-scrollbar{
width: 4px;
}
</style>
</head>
<body>
    <div class="overflow"></div>
</body>
</html>
```