Селекторы

Урок 10

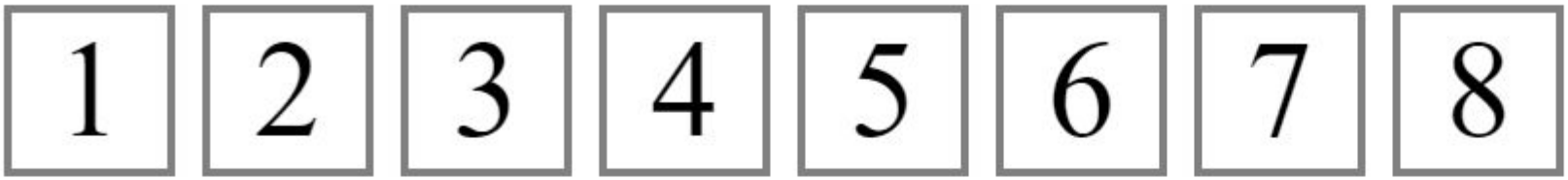
Псевдоклассы дочерних элементов

```
<ul>
<li>1</li>
<li>2</li>
<li>3</li>
<li>4</li>
<li>5</li>
<li>6</li>
<li>7</li>
<li>8</li>
</ul>
```

Псевдоклассы дочерних элементов

```
li {
    display: inline-block;
    width: 80px;
    height: 80px;
    border: 4px solid gray;
    margin-left: 10px;
    font-size: 70px;
    text-align: center;
}
```

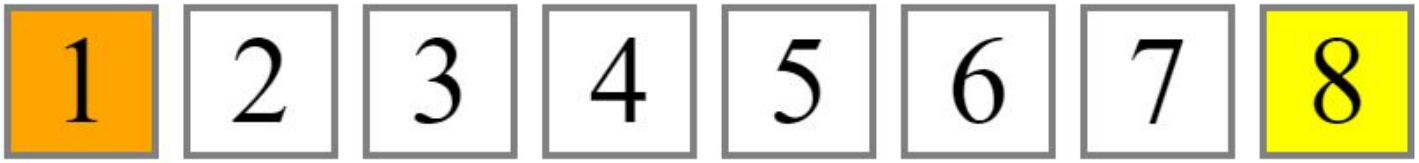
```
<ul>
<li>1</li>
<li>2</li>
<li>3</li>
<li>4</li>
<li>5</li>
<li>6</li>
<li>7</li>
<li>8</li>
</ul>
```



Псевдоклассы дочерних элементов

- **:first-child**: представляет элемент, который является первым дочерним элементом
- **:last-child**: представляет элемент, который является последним дочерним элементом

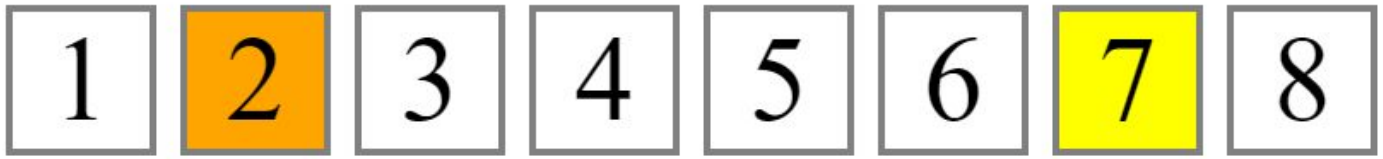
```
li:first-child {
        background-color:  orange;
    }
li:last-child {
        background-color:  yellow;
    }
```



Псевдоклассы дочерних элементов

- **:nth-child(n)**: представляет дочерний элемент, который имеет определенный номер n, например, второй дочерний элемент
- **:nth-last-child(n)**: представляет дочерний элемент, который имеет определенный номер n, начиная с конца

```
li:nth-child(2) {
        background-color: orange;
    }
li:nth-last-child(2) {
        background-color: yellow;
    }
```



Псевдокласс :nth-child

Псевдокласс :nth-child() позволяет выбрать дочерние элементы внутри родительского элемента в зависимости от их порядкового номера.

Применение данного псевдокласса широко распространено, он позволяет чередовать стили строк в таблицах, списках, придать стиль сочетанию дочерних элементов и так далее.

В этот селектор мы можем передать номер стилизуемого элемента:

```
tr:nth-child(3) { background-color: #bbb; }
```

Еще одну возможность представляет использование заменителя для номера, который выражается буквой n:

```
tr:nth-child(2n+1) { background-color: #bbb; }
```

Таким образом, в данном случае выделение начинается с 1-го элемента, а следующим выделяется 2 * 1 + 1 = 3-й элемент, далее 2 * 2 + 1 = 5-й элемент и так далее.

Псевдокласс :nth-child

```
li:nth-child(3n+1) {
    background-color: orange;
}
```


```
li:nth-last-child(3n+1) {
    background-color: orange;
}
```

(even) — ключевое слово, выбирает четные элементы списка, как если бы мы задали (2n).

(odd) — ключевое слово, выбирает нечетные элементы, как если бы мы задали (2n+1).

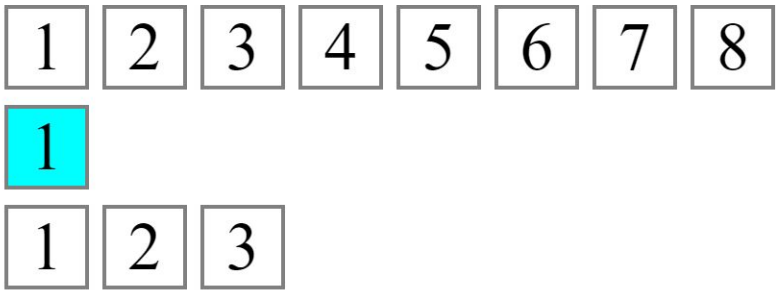
```
li:nth-last-child(even) {
    background-color: orange;
}
li:nth-last-child(odd) {
    background-color: yellow;
}
```

Псевдоклассы дочерних элементов

:only-child: представляет элемент, который является единственным дочерним элементом в каком-нибудь контейнере

```
<ul>
        <li>1</li>
        <li>2</li>
        <li>3</li>
        <li>4</li>
        <li>5</li>
        <li>6</li>
        <li>7</li>
        <li>8</li>
</ul>
<ul>
        <li>1</li>
</ul>
<ul>
        <li>1</li>
        <li>2</li>
        <li>3</li>
</ul>
```

```
li:only-child {
        background-color: cyan;
}
```

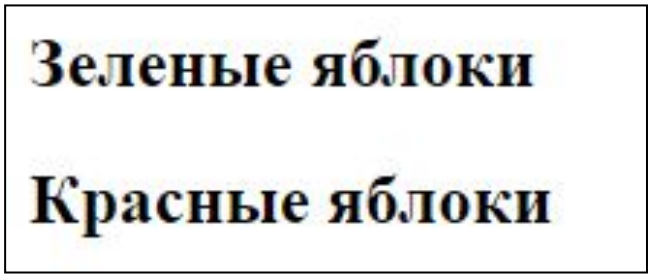


Псевдоклассы дочерних элементов

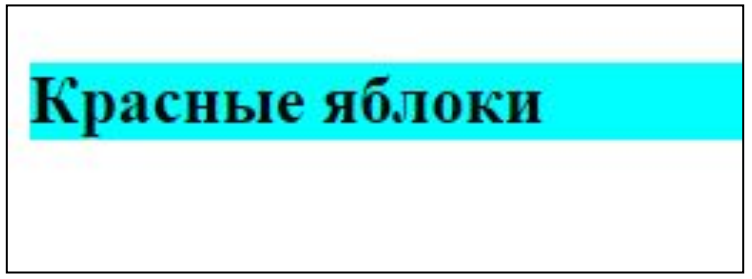
:only-of-type: выбирает элемент, который является единственным элементом определенного типа (тега) в каком-нибудь контейнере

```
h2:only-of-type {
    background-color: cyan;
}
```

```
<div>
<h2>Зеленые яблоки</h2>
<h2>Красные яблоки</h2>
</div>
```




```
<div>
<!--<h2>Зеленые яблоки</h2>-->
<h2>Красные яблоки</h2>
</div>
```

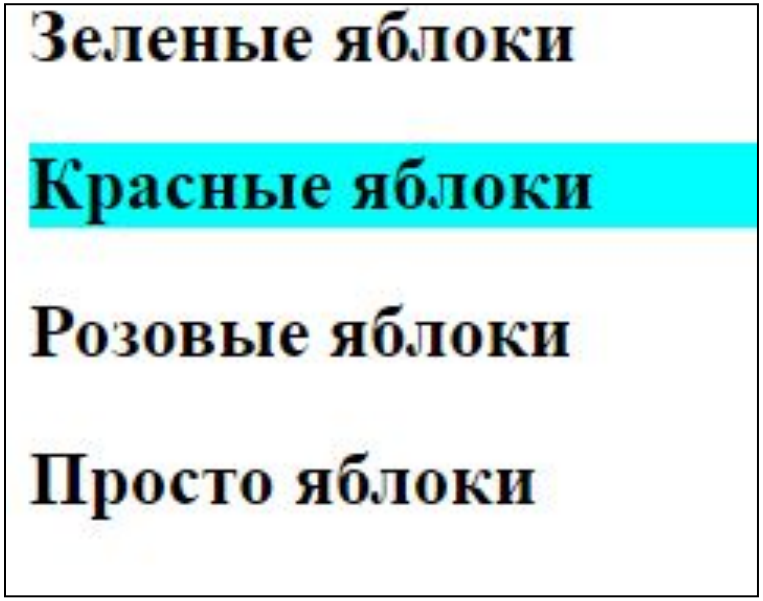


Псевдоклассы дочерних элементов

:nth-of-type(n): выбирает дочерний элемент определенного типа, который имеет определенный номер

```
h2:nth-of-type(2) {
background-color: cyan;
}

<div>
<h2>Зеленые яблоки</h2>
<h2>Красные яблоки</h2>
<h2>Розовые яблоки</h2>
<h2>Просто яблоки</h2>
</div>
```

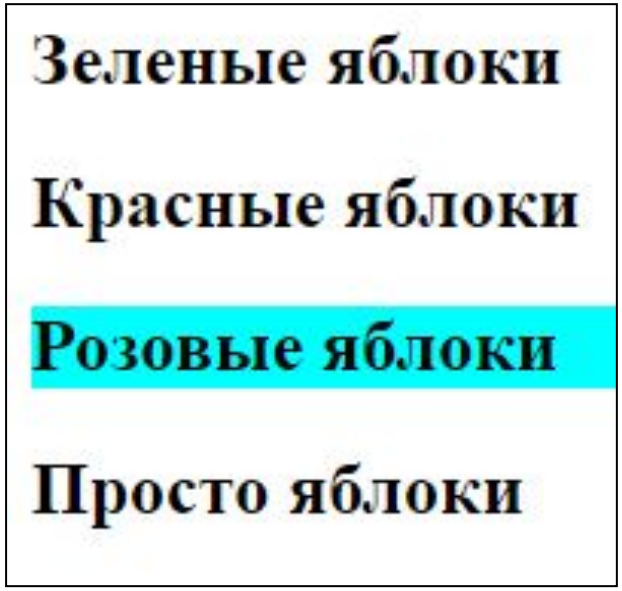


Псевдоклассы дочерних элементов

:nth-last-of-type(n): выбирает дочерний элемент определенного типа, который имеет определенный номер, начиная с конца

```
h2:nth-last-of-type(2) {
background-color: cyan;
}

<div>
<h2>Зеленые яблоки</h2>
<h2>Красные яблоки</h2>
<h2>Розовые яблоки</h2>
<h2>Просто яблоки</h2>
</div>
```



Селекторы атрибутов

Кроме селекторов элементов мы также можем использовать селекторы их атрибутов. Например, у нас есть на веб-странице несколько полей input, а нам надо окрасить в красный цвет только текстовые поля. В этом случае мы как раз можем проверять значение атрибута type: если оно имеет значение text, то это текстовое поле, и соответственно его надо окрасить в красный цвет.

Селекторы атрибутов

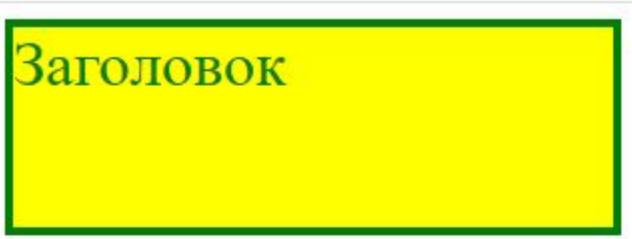
```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title>Селекторы в CSS3</title>
        <style>
            input[type="text"]{
                border: 2px solid red;
            }
        </style>
    </head>
    <body>
        <p><input type="text" id="login" /></p>
        <p><input type="password" id="password" /></p>
        <input type="submit" value="Send" />
    </body>
</html>
```

Селектор по id

Для идентификации уникальных на веб-станице элементов используются идентификаторы, которые определяются с помощью атрибутов id.

Определение стилей для идентификаторов аналогично определению классов, только вместо точки ставится символ решетки #.

```
#header{
        height: 100px;
        width: 300px;
        border: 4px solid green;
        font-size: 24pt;
        color: green;
        background-color: yellow;
    }
```




```
<div id="header">Заголовок</div>
```

Универсальный селектор

Кроме селекторов тегов, классов и идентификаторов в css также есть так называемый **универсальный селектор**, который представляет знак звездочки (*). Он применяет стили ко всем элементам на html-странице:

```
*{
        background-color: red;
   }
```

Наследование стилей

Для упрощения определения стилей в CSS применяется механизм наследования стилей. Этот механизм предполагает, что вложенные элементы могут наследовать стили своих элементов-контейнеров. Поскольку и элемент p, и элемент h2 находятся в элементе body, то они наследуют от этого контейнера - элемента body многие стили.

```
body {color: red;}
```

```
<body>
	<h2>Наследование стилей</h2>
	<p>Текст про наследование стилей в CSS 3</p>
</body>
```

Наследование стилей

- Однако не ко всем свойствам CSS применяется наследование стилей. Например, свойства, которые представляют отступы (margin, padding) и границы (border) элементов, не наследуются.
- Кроме того, браузеры по умолчанию также применяют ряд предустановленных стилей к элементам. Например, заголовки имеют определенную высоту и т.д.

Каскадность стилей

В CSS действует механизм каскадности, которую можно определить как набор правил, определяющих последовательность применения множества стилей к одному и тому же элементу.

Если же стили конфликтуют между собой, например, определяют разный цвет текста, то в этом случае применяется сложная система правил для вычисления значимости каждого стиля. Все эти правила описаны в спецификации по CSS: Calculating a selectors specificity.

Каскадность стилей

Для определения стиля к элементу могут применяться различные селекторы, и важность каждого селектора оценивается в баллах. Чем больше у селектора пунктов, тем он важнее, и тем больший приоритет его стили имеют над стилями других селекторов.

- Селекторы тегов имеют важность, оцениваемую в 1 балл.
- Селекторы классов, атрибутов и псевдоклассов оцениваются в 10 баллов.
- Селекторы идентификаторов оцениваются в 100 баллов.
- Встроенные inline-стили (задаваемые через атрибут style) оцениваются в 1000 баллов.