

# Селекторы

# План

1. посмотрим виды селекторов
2. узнаем сферу применений
3. вес селектора
4. псевдоклассы

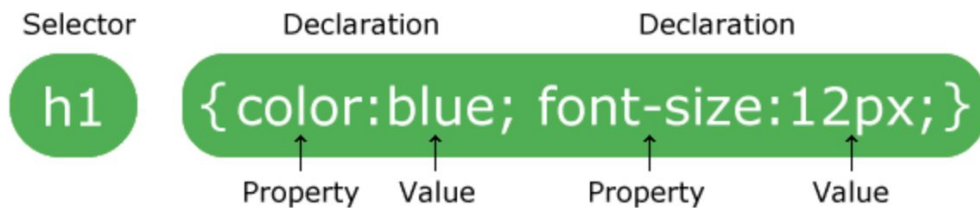
# Css

Формальный язык описания внешнего вида документа, написанного с использованием языка разметки. Для связи правил с html используется селектор

# Селектор - [ссылка](#)

Это выражения, которые говорят браузеру, к какому элементу HTML нужно применить те или иные свойства CSS, определённые внутри блока объявления стиля.

# Состав селектора



```
1 h1 {  
2   color: red;  
3   font-size: 40px;  
4 }
```

# Виды селекторов - тег - список

Используется в основном для задания глобальных стилей, например [браузерных](#), например [обнуления](#), так как элементов одинакового типа может быть много на странице (отчасти это решается в [module css](#))

- 1
- 2
- 3

```
ul { таблица стилей агента пользователя
  display: block;
  list-style-type: disc;
  margin-block-start: 1em;
  margin-block-end: 1em;
  margin-inline-start: 0px;
  margin-inline-end: 0px;
  padding-inline-start: 40px;
}
```

- 1
- 2
- 3

# Виды селекторов - #id - списки

В css такой вид не используется, так как id может быть только 1 на странице, что затрудняет переиспользование, id часто используют для обращения к элементам через js

```
1 <ul id="list-1">
2   <li>1</li>
3   <li>2</li>
4   <li>3</li>
5 </ul>
6
7 <ul id="list-2">
8   <li>1</li>
9   <li>2</li>
10  <li>3</li>
11 </ul>
```

```
1 #list-1 {
2   margin: 0;
3   padding: 0;
4   list-style: none;
5 }
6
7 #list-2 {
8   margin: 0;
9   padding: 0;
10  list-style: none;
11 }
```

# Виды селектор - .класс - списки

Самый популярный вид задания стилей, так как есть переиспользуемость и можно использовать с любым html элементом

```
1 <ul class="list">
2   <li>1</li>
3   <li>2</li>
4   <li>3</li>
5 </ul>
6
7 <ul class="list">
8   <li>1</li>
9   <li>2</li>
10  <li>3</li>
11 </ul>
12
13 <ul>
14   <li>1</li>
15   <li>2</li>
16   <li>3</li>
17 </ul>
```

1  
2  
3  
1  
2  
3

- 1
- 2
- 3



# Виды селектор - атрибуту - форма

Используется реже, но в ряде случаев удобен, например для задания форм или ссылок

A diagram of a web form with rounded corners. It contains four elements: a text input field with the placeholder text 'text', an email input field with the placeholder text 'email', a number input field with the value '-1', and a button labeled 'send'.

# Сложные селектор - пример

Через пробел мы обращаемся во вложенный элемент, а без пробела к одному элементу, такой прием используют редко, так как это повышает вес(об этом позже)

```
1 ▾ <div class="text-wrp">
2 ▾   <p>text</p>
3   </div>
4 ▾ <div>
5 ▾   <p class="text">text</p>
6   </div>
```

```
1 ▾ .text-wrp p{
2   color: red;
3 }
4
5 ▾ p.text {
6   color: blue;
7 }
```

# Сложные селектор (+) - [пример](#)

Выбирает следующий(1) селектор на одном уровне вложенности ([практический пример](#))

```
1 ▼ .wrp+p {  
2     color: red;  
3 }
```

text after .wrp

text after .box

**header after .wrp**

text in .wrp

text in div after wrp

# Сложные селектор (~) - пример

Выбирает следующие(все подходящие) селекторы на одном уровне вложенности

```
1 ▾ .wrp~p {  
2   color: red  
3 }
```

1 text after .wrp

2 text after .wrp

text after .box after box

**header after .wrp**

txt after .wrp after h1

text in .wrp

text in div after .wrp

# Сложные селектор (>) - [пример](#)

Выбирает элементов внутри другого, но только на первом уровне вложенности

```
1 ▾ .wrp > p {  
2   color: red;  
3 }  
4  
5 ▾ /* .wrp p {  
6   color: red;  
7 } */
```

text inside .wrp

text inside .wrp

text inside box inside wrp

# Вес селекторов - пример

```
1 <div>
2   <p class="text" id="id-text">text</p>
3 </div>
```

```
1 div {
2   color: blue;
3 }
4
5 p {
6   color: red;
7 }
8
9 .text {
10  color: yellow;
11 }
12
13 #id-text {
14   color: black;
15 }
```

# Вес селекторов - [статья](#)

0	0	0	0
style=""	#id	.class	<tag>

# Рассчитаем вес - пример

```
1 <div>
2   <p class="text" id="id-text">text</p>
3 </div>
```

```
div {
  color: blue;
}
/* 0 0 0 1 */

p {
  color: red;
}
/* 0 0 0 1 */

.text {
  color: yellow;
}
/* 0 0 1 0 */

#id-text {
  color: black;
}
/* 0 1 0 0 */
```



# Рассчитаем вес для сложных селекторов

0	0	0	0
style=""	#id	.class	<tag>

```
1 ▾ div p {  
2   color: blue;  
3 }  
4  
5 ▾ div .text {  
6   color: yellow;  
7 }  
8  
9 ▾ .wrp .text {  
10  color: green;  
11 }
```

# Псевдоклассы - состояния

У нас есть интерактивные элементы (кнопки, ссылки, поля ввода). Обычно при смене визуальных эффектов мы подсказываем пользователю, что можно повзаимодействовать. Наиболее используемые:

:hover - наведение

:focus - фокус

:active - нажатие

```
1 ▾ a:hover {  
2   color: red;  
3 }  
4  
5 ▾ a:focus {  
6   outline: 2px solid green;  
7   color: green;  
8 }  
9  
10 ▾ a:active {  
11   opacity: 0.5;  
12 }
```

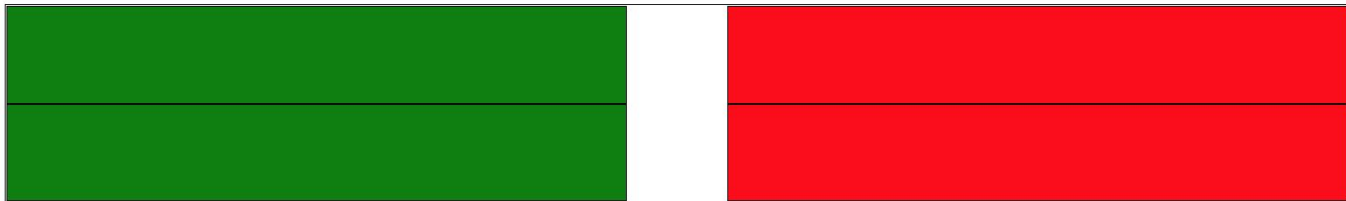
# Псевдоклассы - выбор элементов

Иногда у нас есть задача выбрать определенные элементы, например, каждый второй или начиная с 4-го, последний или первый, для этого может быть использован:

- nth-of-type

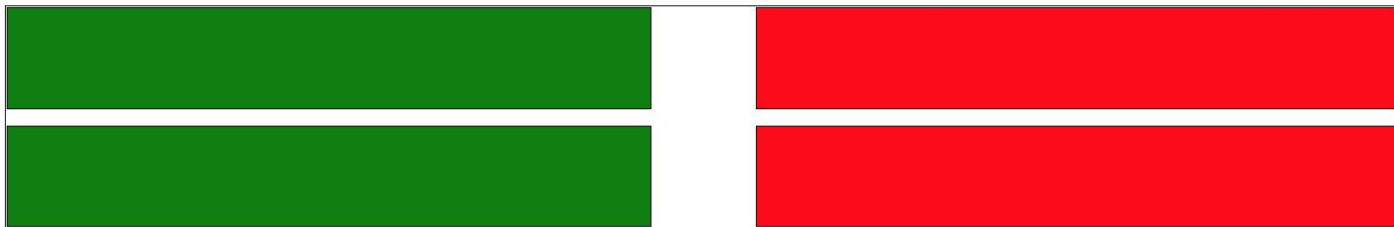
# Псевдоклассы - [nth-of-type](#)

Например у каждого второго элемента будет другой фон(это также можно сделать при помощи [классов](#))

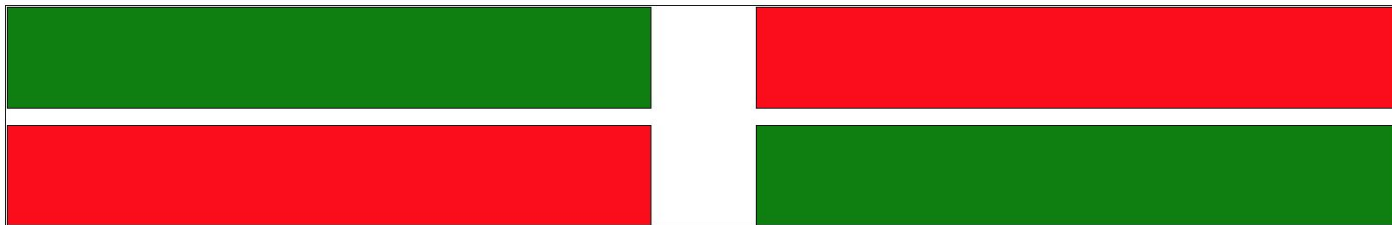


# Псевдоклассы - [nth-of-type](#)

Теперь сделаем у элемента начиная с третьего верхний отступ (также можно [классами](#))



# Псевдоклассы - доска



# Выводы

1. Селекторы бывают по тегу, классу, атрибуту
2. Бывают сложные селекторы (пробел также важен)
3. В зависимости от вида селектора у него может быть вес. Селекторы с большим весом главнее
4. Также есть псевдоклассы, которые помогают нам в специфических ситуациях

Вопросы?



Упехов!