

# CSS Селекторы

---

**WEB  
COURSE  
ORT DNIPRO**

---

**ORT**[DNIPRO.ORG/WEB](https://ORTDNIPRO.ORG/WEB)

# Селекторы в CSS

**CSS селектор**, говорит к каким тегам (элементам) будет применяться описываемый стиль (**css selector**).

```
h3 { color: red; margin: 16px auto; }
```

**CSS селектор**, в составе правила, говорит браузеру к каким тегам необходимо применить правило применить, т.е. задать условие, по которому браузер определит, подходит тег чтобы применить к нему правило или нет.

# 1. Простые селекторы

# Немного практики

1. DIV

2. DIV with class CAT;

3. DIV with id TIGER and class DOG;

4. DIV with class CAT, DOG, TIGER;

5. P with class DOG, BIRD;

6. P

7. P with class CAT, DOG;

8. P with class BIRD, HORSE;

9. ARTICLE with class HORSE;

10. P with class BIRD, DOG.

Воспользуйтесь разметкой:

<https://filebase.xyz/storage/web/selectors/v1/template.html>

# Селектор CSS по названию (типу) тега

```
6  <style>  
7  
8      p {  
9          background: yellow;  
10     }  
11  
12 </style>
```

Стиль применяется ко всем тегам указанного типа (имени).

# Селектор по имени класса

```
6  <style>
7
8      .bird {
9          background: yellow;
10     }
11
12     .cat {
13         text-decoration: line-through;
14     }
15
16 </style>
```

`<div class="bird"></div>`

**.имя-класса** – селектор, который позволяет выбрать теги у которых есть искомый класс. Если у тега несколько классов, то среди имеющихся должен быть искомый.

## Селектор по атрибуту id (знак '#')

```
6  <style>
7
8      #tiger {
9          background: yellow;
10     }
11
12 </style>
```

`<div id="tiger"></div>`

**#идентификатор** – css-селектор, который позволяет выбрать теги у которых есть атрибут **id** равный заданному

# Тег и множество правил

Селектор определяет правила по которым браузер определяет теги к которым будет применены стили. **Тег может подходить под селекторы нескольких правил одновременно.**



## 2. Сложные селекторы

# Сложный селектор

```
6  <style>
7
8      div.cat {
9          background: purple;
10     }
11
12     .dog.bird{
13         background: cyan;
14     }
15
16 </style>
```

Сложный селектор позволяет задать правило для тегов которые должны соответствовать нескольким простым селекторам одновременно, например: *иметь два определенных класса*, или *тег должен быть определенного типа и иметь определённый класс*.

## Псевдокласс :not()

```
6  <style>
7
8      .cat:not(.dog) {
9          background: cyan;
10     }
11
12 </style>
```

Селектор с отрицанием позволит выбрать все теги с классом **.cat** за исключением тех, которые еще имеют и класс **.dog**  
Селектор отрицания может использоваться и в более сложных выражениях. **:not()** принимает только простой селектор!!!

# Псевдоклассы

Если вы встречаете в CSS-селекторе конструкцию записанную через двоеточие, то такую конструкцию называют **псевдокласс**. Псевдоклассы используют для того, чтобы указать на тег основываясь на его позиции в документе или динамическое состояние или на основании других отличительных особенностей.

**Селектор: Псевдокласс { . . . }**

# 3. Комбинированные селекторы

## Селектор «соседей»

```
6  <style>
7
8      #tiger ~ .dog {
9          background: green;
10     }
11
12 </style>
```

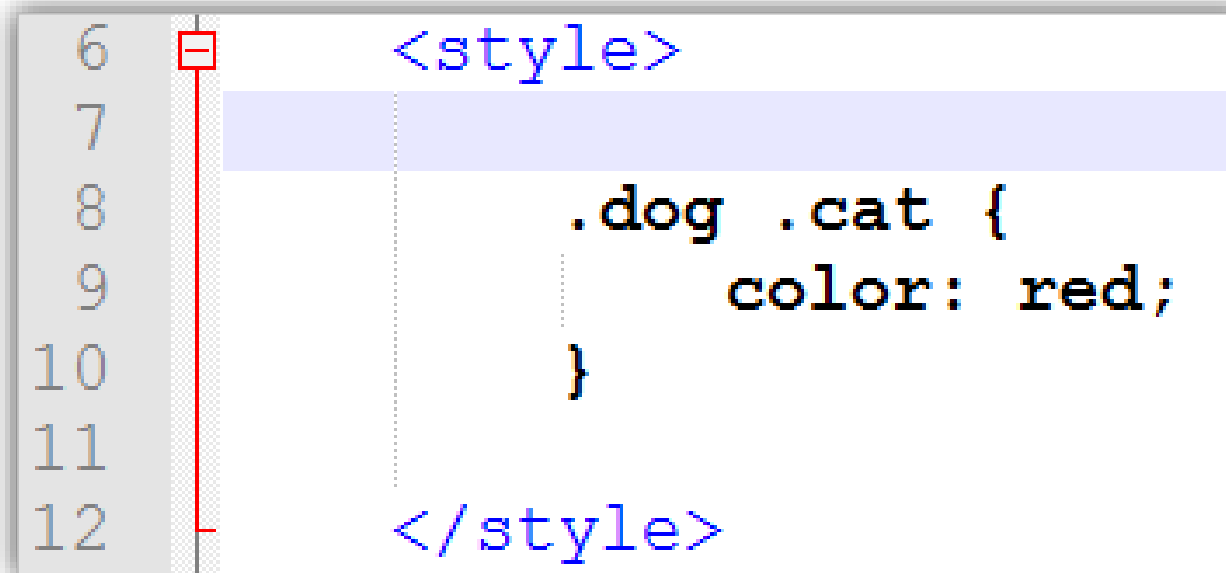
Это «соседний» селектор. Он поможет нам выбрать все теги с классом **.dog**, которые находятся на одном уровне (прямые потомки одного родителя) и следуют за тегом с **id** равным **tiger**.

# Селектор первого соседа

```
6  <style>
7
8      #tiger + .dog {
9          background: green;
10     }
11
12 </style>
```

Это соседний селектор. Он поможет нам выбрать первый тег с классом **.dog**, который находится на одном уровне (прямые потомки одного родителя) и следуют **сразу же** за тегом с **id** равным **tiger**.

## Селектор дочерних элементов (просто знак пробела) - когда нужно найти элемент вложенный в другой элемент



```
6 <style>
7
8     .dog .cat {
9         color: red;
10    }
11
12 </style>
```

Селектор говорит, что правило должно применится к тегу у которого есть класс **.cat** но только если у него среди родителей есть тег с классом **.dog** (вместо классов можно использовать id, название тегов или комбинированный селектор).



Прямой селектор дочерних элементов (знак '>') - когда нужно найти элемент – прямой потомок

```
6  <style>
7
8      .dog > .cat {
9          color: red;
10     }
11
12 </style>
```

Селектор говорит, что правило должно применится к тегу у которого есть класс **.cat** но только если он **прямой** потомок тега с классом **.dog** (вместо классов можно использовать id, название тегов или комбинированный селектор).

## 4. Порядковый селектор :nth-child(n)

# Порядковый селектор

```
p:nth-child(odd) {  
  background: red;  
}  
p:nth-child(even) {  
  background: blue;  
}  
p:nth-child(5) {  
  background: purple;  
}  
p:first-child {  
  background: orange;  
}  
p:last-child {  
  background: #green;  
}
```

```
<div>  
  <p>text 1</p>  
  <p>text 2</p>  
  <p>text 3</p>  
  <p>text 4</p>  
  <p>text 5</p>  
  <p>text 6</p>  
  <p>text 7</p>  
  <p>text 8</p>  
  <p>text 9</p>  
  <p>text 10</p>  
</div>
```

Если по селектору нашлось более одного элемента, то псевдокласс **:nth-child** позволяет уточнить порядковые номера элементов (среди потомков его родителя) которые нас интересуют. **:first-child** и **:last-child** указывают на элемент если он первый или последний потом своего родителя (соответственно).

5. ~~Каскадирование~~

~~Специфичность~~

Как браузер разрешает  
противоречия?

**К одному тегу могут применяться  
несколько правил, но что  
если они противоречат друг другу?**

**В случае противоречия браузер отдаёт предпочтение одному из стилевых свойств. У правил есть приоритеты.**

### **Порядок приоритетов такой:**

1. Наибольший приоритет имеют стили описанные прямо в теге в атрибуте **style="..."**;
2. Далее следует правила у которых есть селектор по атрибуту **id** т.е. вида **#report { ... }**;
3. После этого следуют правила с селекторами по любым другим атрибутам (в том числе и атрибуту **class** т.е. вида **.sometype { ... }**);
4. Правила в селекторе которых просто название тега, например **h1 { ... }**.

*Если у нескольких правил одинаковый приоритет, то применяется то, которое встречается последним (т.е. идёт ниже, в коде, чем другие).*

<https://css-live.ru/css/nikto-ne-znaet-css-specifichnost-ne-kaskad.html>

Команда **!important** записанная после какого-либо из стилевых правил делает его наиболее приоритетным из всех, независимо от того какой селектор применяется.

## 6. Немного практики



### Кухонный комбайн KX567-T

Этот кухонный комбайн KX567-T оснащен мощным мотором с 2 скоростями и импульсным режимом, что обеспечивает мощность и точное управление для приготовления любых блюд.

~~100000~~ €

7499 €

- 25% !!!

Купить сейчас!

Воспроизведём оформление  
компонента на базе готовой  
HTML-разметки

Необходимо  
воспроизвести стили  
макета. Без внесения  
изменений в файл  
**index.html**

Проект с макетом вы можете скачать:

<https://filebase.xyz/storage/web/selectors/v1/component.zip>



Будет полезным

# Узнать о селекторах по атрибуту

`[attr] { ... }`

`[attr=value] { ... }`

`[attr~=value] { ... }`

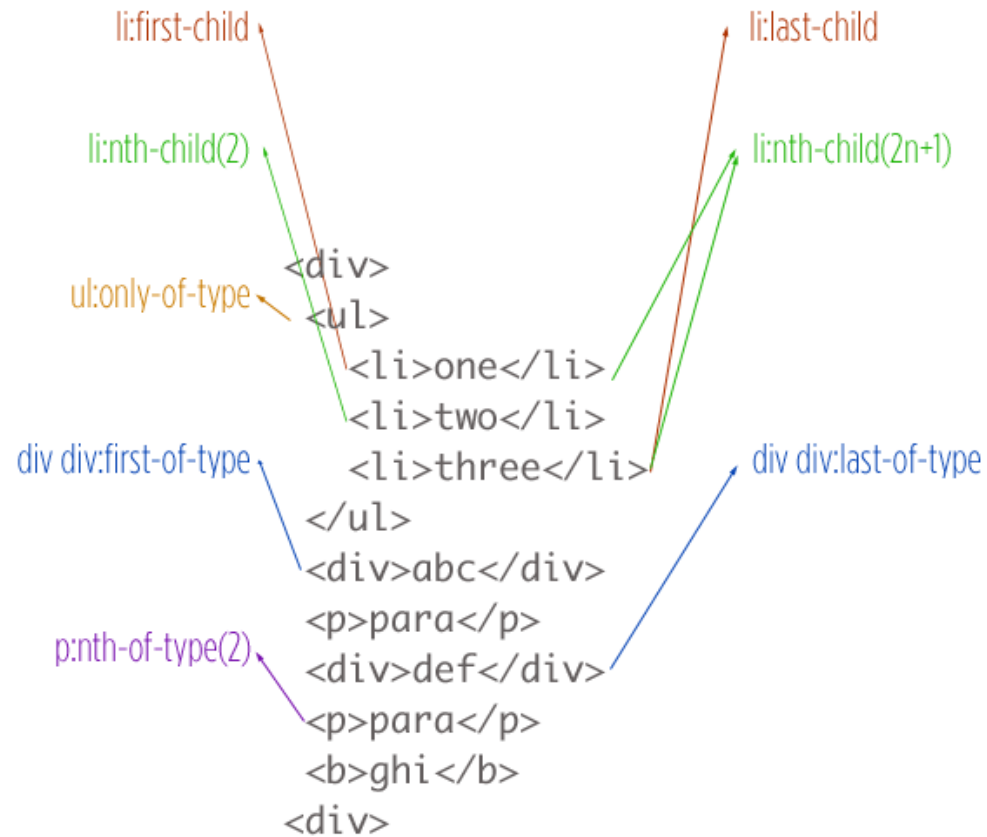
`[attr^=value] { ... }`

`[attr$=value] { ... }`

`[attr*=value] { ... }`

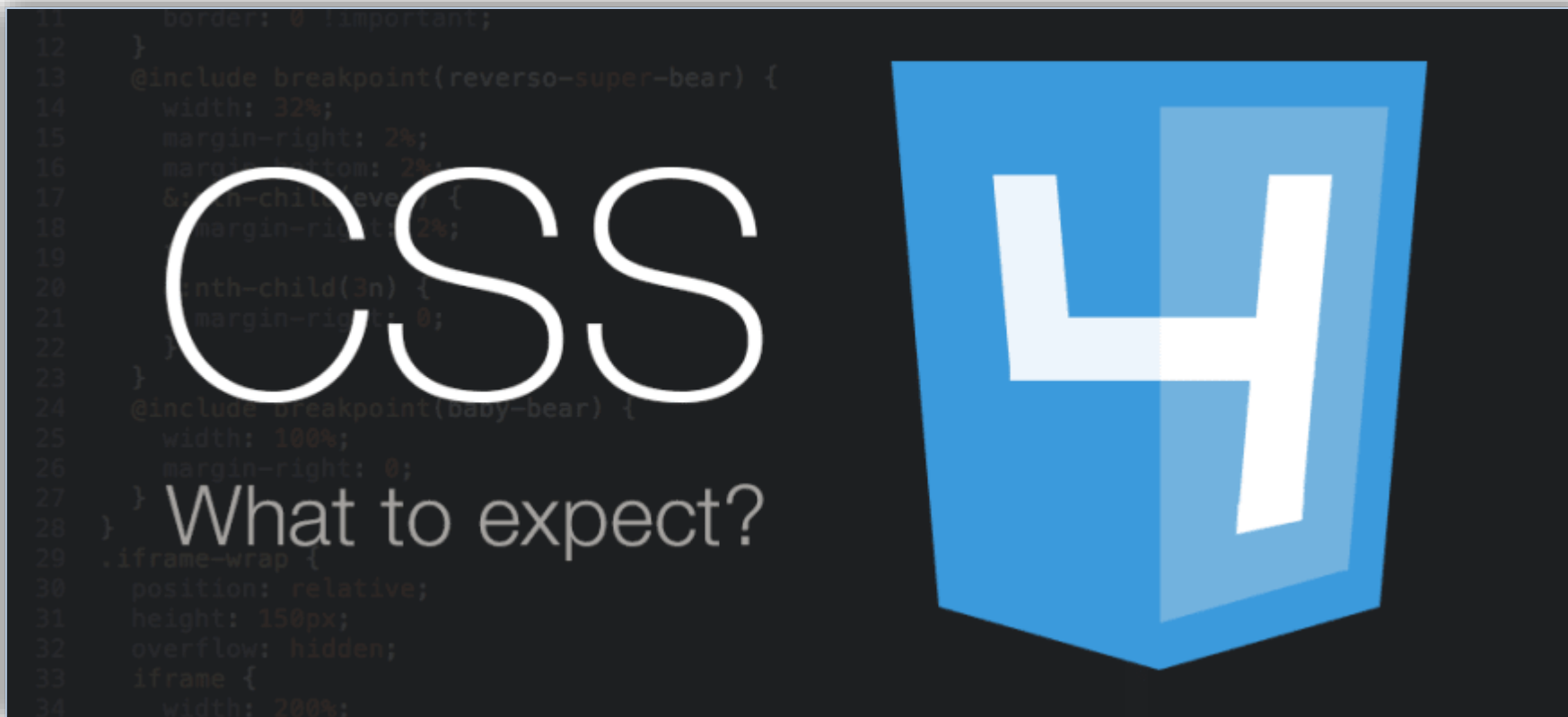
<https://webref.ru/css/selector/attr>

# Ознакомиться с вариантами псевдоклассов



<https://webref.ru/css/type/pseudoclass>

# Новые селекторы в CSS 4?!

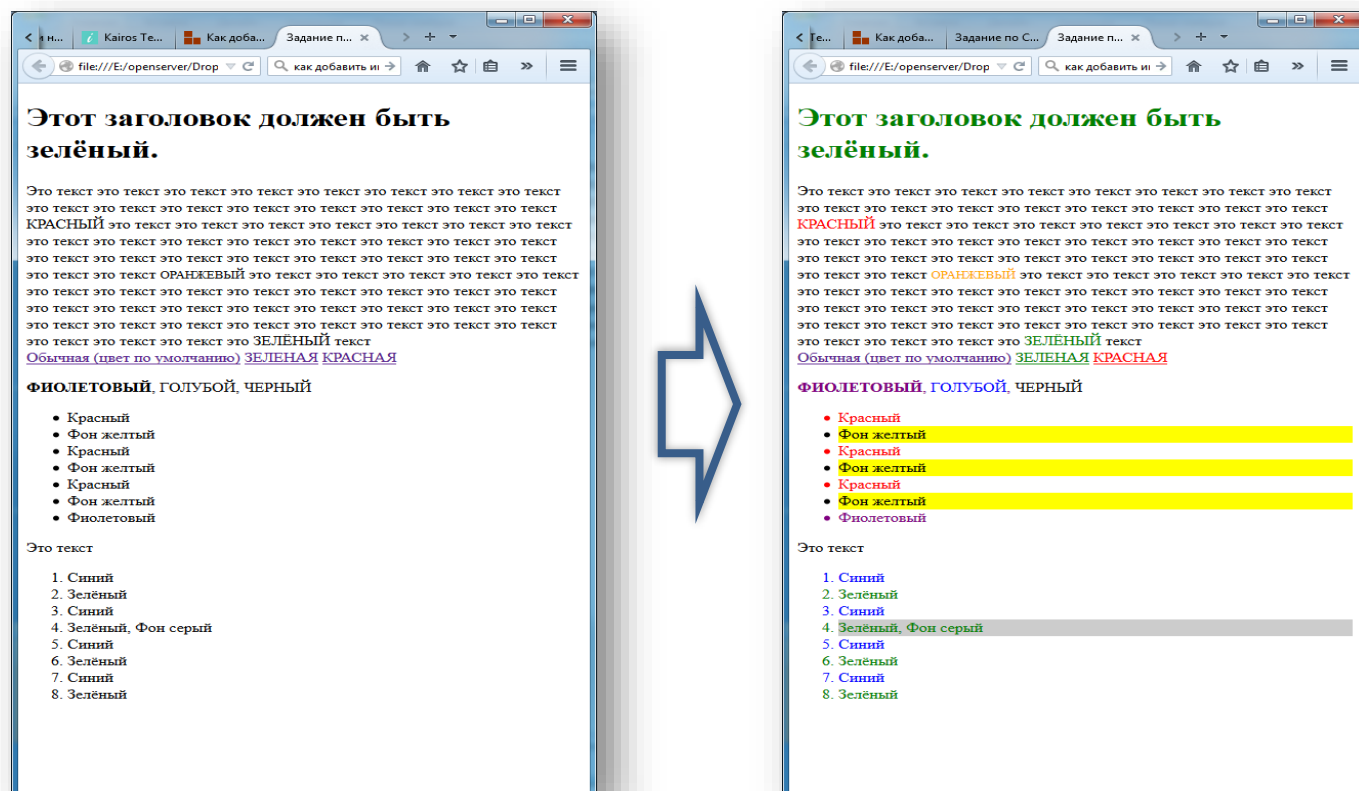


<https://css4-selectors.com/selectors/>

# Домашнее задание

# Задача на применение CSS селекторов

Возьмите заготовленный шаблон, и раскрасьте его в соответствии с инструкцией в разметки.



Воспользуйтесь разметкой:

<https://filebase.xyz/storage/web/selectors/v1/homework.html>

# Пройти игру!

Select the apple directly on a plate

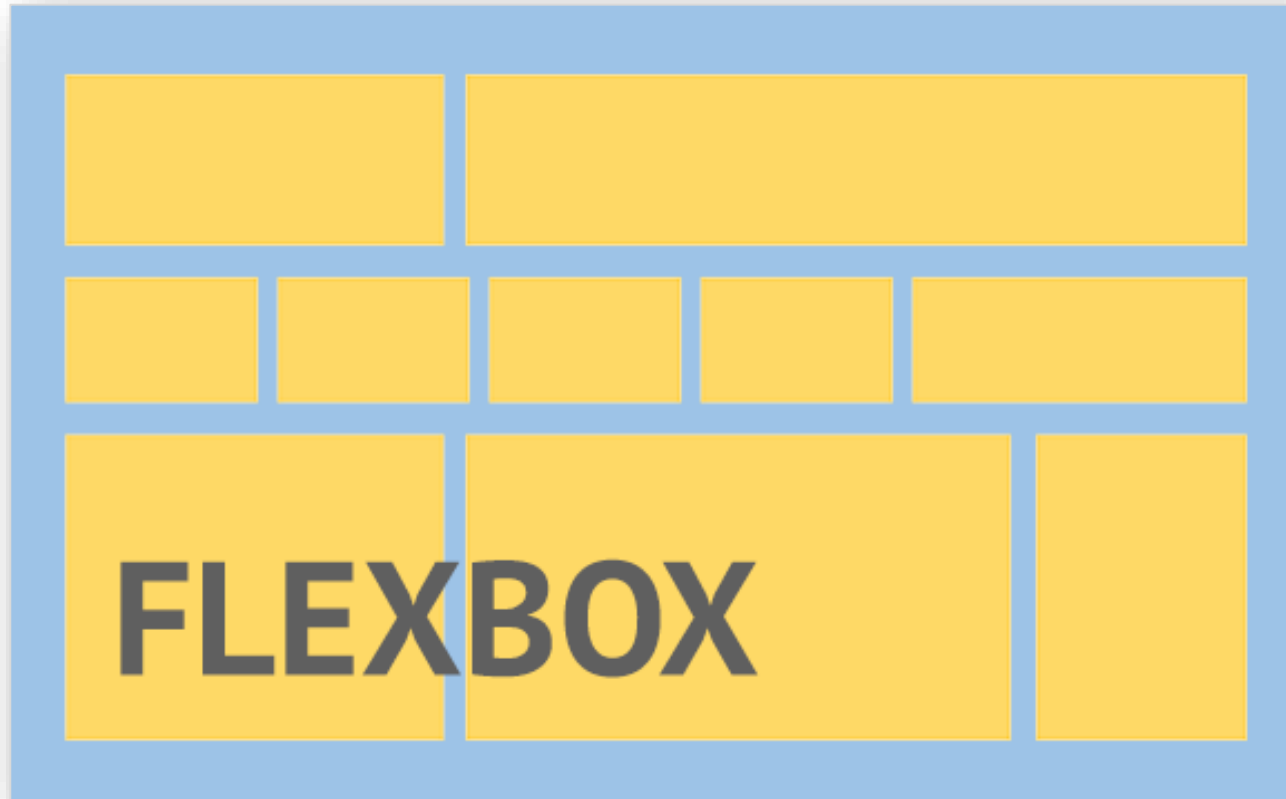


<https://flukeout.github.io/>

К следующему  
занятию...



# FlexBox – управление размещением элементов



Предварительные знания – лучший помощник в обучении, поэтому к следующему занятию жду, что **посмотрите небольшой ролик о FlexBox.**

<https://youtu.be/CDWMSF0nI2A?t=11>