



Frontender[1.0] Приоритеты селекторов. HTML, CSS

YouTube	https://youtu.be/ULjnqf8vayo
Telegram	https://t.me/Dmitry_Kolotilshikov
# Номер урока	20



Приоритетность селекторов - это механизм, используемый для определения, какой стиль должен быть применен к элементу на странице, когда существуют несколько стилей, которые могут быть применены к одному и тому же элементу.

Приоритетность селектора определяется его специфичностью. Чем более специфичный селектор, тем выше его приоритет. Например, селектор по **ID** имеет более высокий приоритет, чем селектор по **классу**, который в свою очередь имеет более высокий приоритет, чем селектор по типу **тега**.

Если несколько стилей или скриптов имеют **одинаковую специфичность**, то приоритет отдается тому, который **объявлен позже**.

Приоритетность селекторов может быть также изменена с помощью модификатора `!important;`, который может быть использован для установки максимального приоритета для стиля. Например, если стиль `color: red !important;` будет применен к элементу, то этот стиль будет иметь максимальный приоритет и изменит цвет элемента на красный, даже если другие стили имеют более высокую специфичность.

Важно помнить, что использование `!important` слишком часто может привести к непредсказуемым результатам и сложности в отладке, поэтому его следует использовать с осторожностью.

Простыми словами принцип можно описать так — чем конкретнее задан селектор правила, тем больше его «вес».

Выбор конкретного правила зависит от трёх составляющих (в порядке возрастания значимости):

1. Порядок правила в коде.
2. Значение специфичности селектора.
3. Модификатор `!important`.

Таблица для расчёта специфичности селекторов выглядит так:

Тип селектора	Пример	Специфичность
Инлайновый стиль	<code><p style="color: red;"></code>	1000
Идентификатор элемента	<code>#button { color: green; }</code>	100
Класс, селектор по атрибуту, псевдокласс	<code>.active { color: white; }</code> <code>[name="address"] { color: black; }</code> <code>:hover { color: red; }</code>	10
Тег, псевдоэлемент	<code>a { text-decoration: none; }</code> <code>::placeholder { color: gray; }</code>	1

Исходя из этой таблицы, каждое правило в таблице стилей будет иметь свою специфичность, например:

Селектор	Типы селекторов	Специфичность
<code>.menu a { color: white; }</code>	Класс и тег	$10 + 1 = 11$
<code>#menu a:hover { color: blue; }</code>	Идентификатор, тег и псевдокласс	$100 + 1 + 10 = 111$
<code>form.large input[type="checkbox"]::after { color: #555; }</code>	Тег, класс, тег, селектор по атрибуту и псевдоэлемент	$1 + 10 + 1 + 10 + 1 = 23$
<code>.thumb > a:hover img { transform: scale(1.3); }</code>	Класс, тег, псевдокласс, тег	$10 + 1 + 10 + 1 = 22$

Комбинаторы `>`, `+`, `~`, селектор `*` и псевдокласс `:not` не влияют на специфичность.

Если два разных правила имеют одинаковую специфичность и влияют на один и тот же элемент, то перевесит правило, которое стоит ниже в коде. Свойство с модификатором `!important` перевешивает все прочие свойства.

Чтобы пересилить `!important`, придётся добавить новый `!important` в правило с большей специфичностью, или поставить селектор с такой же специфичностью и `!important` позже в коде. Злоупотребление флагом `!important` часто приводит к тому, что весь каскад правил начинает выходить из под контроля разработчика.

Специфичность селекторов — это внутренний механизм и алгоритм работы ЦСС, который описан в стандарте, реализован в браузерах, но напрямую не виден разработчикам. Специфичность нельзя прочитать и нельзя задать напрямую, но полезно знать о том как она работает, чтобы применять ЦСС более эффективно и добиваться решения задачи минимальными средствами.

Статья по приоритетам селекторов: <https://bureau.ru/soviet/20200723/>

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>styling</title>

  <style>
    p {
      border-width: 5px;
      border-color: black;
      border-style: solid;
      height: 200px;
      color: red;
    }

    #text {
      color: blue;
    }
    .text {
      color: orange;
    }
  </style>
</head>
<body>
```

```
<p id="text" class="text">
  Lorem ipsum dolor sit, amet consectetur adipisicing elit. Nulla, et.
  Lorem ipsum dolor sit, amet consectetur adipisicing elit. Nulla, et.
  Lorem ipsum dolor sit, amet consectetur adipisicing elit. Nulla, et.
</p>
</body>
</html>
```