

# Сколько весят селекторы?

03/01/2014

Содержание:

[Вес селектора](#)

[Варианты решений](#)

Все CSS-селекторы имеют свой вес, который определяет как взаимодействуют одинаковые свойства, заданные в разных местах кода одному и тому же элементу.

Иногда это может создавать трудности, когда свойство, объявленное ниже в коде, перекрывается тем, что объявлено выше, потому что селектор первого более специфичен (имеет больший вес).

Вот пример проблемы. Есть див с `id="container"`, внутри него некоторый текст и список ссылок.

```
<div id="container">
  <p><a href="#">link in P</a></p>

  <ul class="list">
    <li><a href="#">Link1</a></li>
    <li><a href="#">Link2</a></li>
  </ul>
</div>
```

Сначала задаём всем ссылкам внутри `#container` оранжевый фон:

```
#container A {
  background: orange;
}
```

А потом, чтобы в списке `.list` внутри контейнера ссылки имели зелёный фон, ниже дописываем такое:

```
.list A {
  background: mediumspringgreen;
}
```

Казалось бы, ссылки в тексте должны получить оранжевый фон, а ссылки в списке — зелёный, но нет:

Почему так? Потому, что первый селектор содержит ID и перевешивает второй, то есть:

```
#container A > .list A
```

## Вес селектора

Специфичность селектора рассчитывается по 4-м позициям:

Для каждого из них подсчитывается количество подходящих элементов в селекторе, и это число помещается в соответствующую позицию.

**Пример:**

**Вес селекторов (по убыванию):**

`style=""1,0,0,0`

`#id0,1,0,0`

`.class0,0,1,0`

`[attr=value]0,0,1,0`

`LI0,0,0,1`

`*0,0,0,0`

У стилей, заданных в атрибуте `style`, на первой позиции будет единица — `1,0,0,0`. Это самая высокая специфичность, которая перевешивает свойства, заданные другими способами.

Переопределить стили, заданные в `style`, можно дописав `!important` к значению свойства в таблице стилей.

Обратный вариант — универсальный селектор `*`, он не имеет веса: `0,0,0,0`.

**Примеры:**

`LI0,0,0,1` — селектор по тегу

`UL LI0,0,0,2` — селектор с двумя тегами весит больше, чем с одним.

`.orange0,0,1,0` — селектор с классом весит больше, чем селектор с тегом.

`.orange A SPAN0,0,1,2` — селектор перевесит предыдущий, потому что помимо класса содержит два тега.

`#page .orange0,1,1,0` — селектор с ID перевесит всё, кроме `inline`-стилей.

Теперь сравним селекторы из исходного примера:

`#container A0,1,0,1`

`.list A0,0,1,1`

`0,1,0,1 > 0,0,1,1` — хорошо видно, что селектор с ID весит больше, чем селектор с классом, поэтому все ссылки имеют оранжевый фон, хотя ниже в коде им задан зеленый.

## Варианты решений

1. Добавить `!important`

```
#container A {
```

```

background: orange;
}

.list A {
background: mediumspringgreen !important;
}

```

Ссылки получают зеленый фон, быстро и легко. Но это плохой способ, потому что код запутывается ещё больше. Со временем для переопределения `!important` в одном месте может потребоваться добавить его в других местах. Иерархичность начнет работать не сверху вниз и от общего к частному, а как попало. В конце-концов поддерживать такой код будет весьма проблематично.

В общих случаях использовать `!important` не рекомендуется, но может пригодиться, если нужно, чтобы часто используемый блок на всех страницах выглядел одинаково, независимо от окружения. В любом случае нужно всегда четко понимать зачем вы его используете.

2. Следующий очевидный способ — добавить `#container` ко второму селектору, чтобы увеличить его вес:

```

#container A {
background: orange;
}

#container .list A {
background: mediumspringgreen;
}

```

Это тоже сработает, но решение так себе: удлиняется цепочка селекторов (что может отразиться на скорости отрисовки страницы) и ухудшается читаемость кода. Так тоже делать не стоит.

1-й и 2-й способ могут использоваться, если у вас нет доступа к разметке, а в ней нет нужных классов. Если же вы можете редактировать разметку либо классы у элементов таки есть — используйте последний способ, самый правильный:

3. Просто не используйте в стилях селекторы с ID, используйте классы.

Посмотрим на разницу между `#container` и `.container`:

`#container A0,1,0,1` — селектор с ID перевешивает всё вне зависимости от своего расположения в коде.

Заменяем в разметке страницы `id` на `class`:

`.container A0,0,1,1` — селектор с классом весит меньше, он менее специфичен.

Селектор ссылок в списке весит столько же:

```
.list A0,0,1,1
```

Это означает, что при равном весе селекторов применяются стили, объявленные ниже в коде. То есть достаточно будет просто написать стили, следуя от общего к частному, сверху вниз.

В итоге разметка может быть такой:

```
<div class="container">
  <p><a href="#">link in P</a></p>

  <ul class="list">
    <li><a href="#">Link1</a></li>
    <li><a href="#">Link2</a></li>
  </ul>
</div>
```

А стили — такими:

```
.container A {
  background: orange;
}

.list A {
  background: mediumspringgreen;
}
```

И код работает так, как ожидается:

*Если id в вашей разметке уже используется в Js, логичнее будет добавить элементу класс и перевесить стили на него. Если же id участвует только в разметке — лучше заменить его на class.*

В качестве общих рекомендаций так же следует упомянуть, что нужно как можно меньше использовать селекторы по тегу и как можно больше — селекторы по классу. Это поможет избежать проблем при повторном использовании блоков сайта, а при использовании "умных" классов — может значительно сократить цепочки селекторов, увеличить читабельность кода и скорость отрисовки страницы.

Спецификации:

- [w3.org/TR/CSS2/cascade.html#specificity](http://w3.org/TR/CSS2/cascade.html#specificity)
- [w3.org/TR/css3-selectors/#specificity](http://w3.org/TR/css3-selectors/#specificity)

Ссылки по теме:

[Specifics on CSS Specificity](#)

[CSS Specificity: Things You Should Know](#)

[Специфичность в комиксах](#)

[Специфичность на примере "Звёздных войн"](#)