

Раздел

[CSS для JavaScript-разработчика](#)

Навигация по уроку

Каждое свойство – на отдельной строке

Каждый селектор – на отдельной строке

Свойства, сильнее влияющие на документ, идут первыми

Организация CSS-файлов проекта

Комментарии

Поделиться

[Редактировать на GitHub](#)[🏠](#) → [CSS для JavaScript-разработчика](#)

📅 4-го октября 2020

# Правила форматирования CSS

Для того, чтобы CSS легко читался, полезно соблюдать пять правил форматирования.

## Каждое свойство – на отдельной строке

Так – неверно:

```
1 /*+ no-beautify */
2 #snapshot-box h2 { padding: 0 0 6px 0; font-weight: bold;
```

Так – правильно:

```
1 /*+ no-beautify */
2 #snapshot-box h2 {
3     position: absolute;
4     left: 0;
5     top: 0;
6     padding: 0 0 6px 0;
7     font-weight: bold;
8 }
```

Цель – лучшая читаемость, проще найти и поправить свойство.



## Каждый селектор – на отдельной строке



Неправильно:

```
1 /*+ no-beautify */
2 #snapshot-box h2, #profile-box h2, #order-box h2 {
3     padding: 0 0 6px 0;
4     font-weight: bold;
5 }
```

Правильно:

```
1 /*+ no-beautify */
2 #snapshot-box h2,
3 #profile-box h2,
4 #order-box h2 {
5     padding: 0 0 6px 0;
6     font-weight: bold;
7 }
```

Цель – лучшая читаемость, проще найти селектор.

## Свойства, сильнее влияющие на документ, идут первыми

Рекомендуется располагать свойства в следующем порядке:

1. Сначала положение элемента относительно других: `position`, `left/right/top/bottom`, `float`, `clear`, `z-index`.
2. Затем размеры и отступы: `width`, `height`, `margin`, `padding`...
3. Рамка `border`, она частично относится к размерам.
4. Общее оформление содержимого: `list-style-type`, `overflow`...

Раздел

CSS для JavaScript-разработчика

Навигация по уроку

Каждое свойство – на отдельной строке

Каждый селектор – на отдельной строке

Свойства, сильнее влияющие на документ, идут первыми

Организация CSS-файлов проекта

Комментарии

Поделиться



Редактировать на GitHub



5. Цветовое и стилевое оформление: background, color, font ...

**Общая логика сортировки: «от общего – к локальному и менее важному».**

При таком порядке свойства, определяющие структуру документа, будут видны наиболее отчётливо, в начале, а самые незначительно влияющие (например цвет) – в конце.

Например:

```
1  /*+ no-beautify */
2  #snapshot-box h2 {
3    position: absolute; /* позиционирование */
4    left: 0;
5    top: 0;
6
7    padding: 0 0 6px 0; /* размеры и отступы */
8
9    color: red;          /* стилевое оформление */
10   font-weight: bold;
11 }
```

**Свойство без префикса пишется последним.**

Например:

```
1  -webkit-box-shadow:0 0 100px 20px #000;
2  box-shadow:0 0 100px 20px #000;
```

Это нужно, чтобы стандартная (окончательная) реализация всегда была важнее, чем временные браузерные.

## Организация CSS-файлов проекта

Стили можно разделить на две основные группы:

1. **Блоки-компоненты имеют свой CSS.** Например, CSS для диалогового окна, CSS для профиля посетителя, CSS для меню.

Такой CSS идёт «в комплекте» с модулем, его удобно выделять в отдельные файлы и подключать через @import .

Конечно, при разработке будет много CSS-файлов, но при выкладке проекта система сборки и сжатия CSS заменит директивы @import на их содержимое, объединяя все CSS в один файл.

2. **Страничный и интегрирующий CSS.**

Этот CSS описывает общий вид страницы, расположение компонент и их дополнительную стилизацию, зависящую от места на странице и т.п.

```
1  /*+ no-beautify */
2  .tab .profile { /* профиль внутри вкладки */
3    float: left;
4    width: 300px;
5    height: 200px;
6  }
```

Важные страничные блоки можно выделять особыми комментариями:

```
1  /** =====
2   *   Профиль посетителя
3   *   =====
4  */
5
6  .profile {
7    border: 1px solid gray;
8  }
9
```

Раздел

[CSS для JavaScript-разработчика](#)

Навигация по уроку

Каждое свойство – на отдельной строке

Каждый селектор – на отдельной строке

Свойства, сильнее влияющие на документ, идут первыми

Организация CSS-файлов проекта

Комментарии

Поделиться



Редактировать на GitHub



```
10 .profile h2 {  
11   color: blue;  
12   font-size: 1.8em;  
13 }
```

CSS-препроцессоры, такие как [SASS](#), [LESS](#), [Stylus](#), [Autoprefixer](#) делают написание CSS сильно удобнее.

Выберите один из них и используйте. Единственно, они добавляют дополнительную предобработку CSS, которую нужно учесть, и желательно, на сервере.

Проводим [курсы по JavaScript и фреймворкам](#). ✕



## Комментарии

перед тем как писать...

