

КАСКАДНОСТЬ CSS

Исаева Кристина Валерьевна

студент, Ульяновский авиационный колледж – Межрегиональный центр компетенций, РФ, г. Ульяновск

Гнеушев Сергей Алексеевич

научный руководитель, преподаватель Ульяновский авиационный колледж – Межрегиональный центр компетенций, РФ, г. Ульяновск

Аннотация. В статье объясняется почему таблицы стилей называют каскадными, как происходит взаимодействие стилей между собой внутри документа.

Ключевые слова: CSS, каскадность CSS, приоритеты стилей, селекторы, объявление !important, конфликт сетей.

Как известно, стиль на веб-страницу можно добавить тремя способами:

- подключение внешней таблицы стилей;
- добавление внутренней таблицы стилей в HTML-документ через тег <style>;
- определение стиля элемента, применяя к тегу атрибут style с необходимыми значениями (inline-стиль).

Эти способы добавления стилей могут использоваться одновременно. Так, для тегов <p> может быть одновременно задан стиль как во внешней таблице стилей, так и во внутренней. Также несколько вариантов форматирования тегов <p> могут быть записаны в одной таблице стилей (рисунок 1).

```
1  p {color: green;}
2  .your_class p {color: yellow;}
3  #your_id p {color: red;}
```

Рисунок 1. варианты форматирования тегов <p>

Цель работы – изучение каскадности CSS. Для достижения поставленной цели, необходимо решить следующие задачи: изучить различные источники информации по данной теме, определить то, каким образом браузер расставляет приоритеты между стилями, какой стиль будет выбран и по какому принципу.

Для начала определим, что такое каскадность CSS. Это такой механизм, с помощью которого возможно применение более одного правила CSS к элементу HTML-документа. Правила могут исходить из различных источников:

- из внешней и внутренней таблицы стилей,
- от механизма наследования,
- от родительских элементов,
- от классов и ID,
- от селектора тега,
- от атрибута style и т. д.

Так как в этих случаях часто происходит конфликт стилей, была создана система приоритетов: в конечном результате будет применен тот стиль, который исходит от источника с более высоким приоритетом.

Таблица 1.

Таблица значимости селекторов (чем больше вес, тем выше приоритет)

Селектор тега:	1
Селектор класса:	10
Селектор ID:	100
Inline-стиль:	1000

В случае если селектор состоит из других селекторов, необходимо посчитать их общий вес. За каждый селектор добавляется 1 в соответствующую ячейку. В остальных ячейках стоят нули. Чтобы получить общий вес, необходимо совместить все числа в ячейках.

Таблица 2.

Пример получения общего веса селекторов

Селектор	ID	Класс	Тег	Общий вес
P	0	0	1	1
.your class	0	1	0	10
p.your class	0	1	1	11
#your id	1	0	0	100
#your id p	1	0	1	101
#your id .your class	1	1	0	110
p a	0	0	2	2
#your id #my id .your class p a	2	1	2	212

Когда вес двух селекторов совпадает, приоритет отдается стилю, находится в коде ниже. В случае, когда для одного элемента задан стиль и во внешней, и во внутренней таблицах, приоритет отдается стилю в таблице, находящейся в коде ниже.

Когда необходимо повысить значимость какого-либо свойства, можно добавить к нему объявление !important (рисунок 2).

```
1 | p {color: red !important;}
2 | p {color: green;}
```

Рисунок 2. Применение объявления *!important*

Также *!important* перекрывает inline-стили. Однако, данное объявление принято использовать только в тех случаях, когда конфликт стилей нельзя исключить иными способами.

Так как браузеры имеют большое количество отличий во встроенных стилях, при написании собственного стиля CSS, можно воспользоваться методом сброса встроенных стилей, чтобы не создавались помехи для кроссбраузерности.

В целом, инструмент для сброса стилей – это та же таблица CSS, где описаны правила, которые сбрасывают встроенные стили браузеров, устанавливая базовые значения свойств. Такая таблица называется *reset.css* и служит для того, чтобы была возможность начать создавать стиль «с нуля».

```
1  html, body, div, span, applet, object, iframe,
2  h1, h2, h3, h4, h5, h6, p, blockquote, pre,
3  a, abbr, acronym, address, big, cite, code,
4  del, dfn, em, img, ins, kbd, q, s, samp,
5  small, strike, strong, sub, sup, tt, var,
6  b, u, i, center,
7  dl, dt, dd, ol, ul, li,
8  fieldset, form, label, legend,
9  table, caption, tbody, tfoot, thead, tr, th, td,
10 article, aside, canvas, details, embed,
11 figure, figcaption, footer, header, hgroup,
12 menu, nav, output, ruby, section, summary,
13 time, mark, audio, video {
14     margin: 0;
15     padding: 0;
16     border: 0;
17     font-size: 100%;
18     font: inherit;
19     vertical-align: baseline;
20 }
21 /* HTML5 display-role reset for older browsers */
22 article, aside, details, figcaption, figure,
23 footer, header, hgroup, menu, nav, section {
24     display: block;
25 }
26 body {
27     line-height: 1;
28 }
29 ol, ul {
30     list-style: none;
31 }
32 blockquote, q {
33     quotes: none;
34 }
35 blockquote:before, blockquote:after,
36 q:before, q:after {
37     content: '';
38     content: none;
39 }
40 table {
41     border-collapse: collapse;
42     border-spacing: 0;
43 }
```

Рисунок 3. Стандартная таблица сброса

Такой вариант таблицы Reset.css был разработан Эриком Мейер –американским веб-дизайнером, который широко известен благодаря своей работе над веб-стандартами.

Что же можно сделать с помощью таблицы представленной выше?

- в первом селекторе собраны наиболее популярные теги, для которых удаляются все отступы, рамки и поля, а также назначается размер шрифта 100%;
- второй групповой селектор предназначен для правильного отображения тегов HTML5 в старых браузерах;
- с помощью свойства line-height селектора body устанавливается одинаковый интерлиньяж (межстрочный интервал) для текста;
- для списков ol, ul убраны маркеры;
- отменены кавычки и другой контент перед и после содержимого тегов blockquote, q;
- упрощено добавление рамок для ячеек таблиц.

Таким образом, конечный стиль элемента, который отображается в браузере – это комбинация нескольких последовательно примененных стилей. В случае конфликта нескольких стилей включается правило приоритетов. Предпочтение отдается более значимому стилю. И чтобы создавать грамотные таблицы стилей, необходимо знать, как работают правила распределения приоритетов между стилями.

Список литературы:

1. Каскад и наследование : сайт – 07.11.2022 – URL: https://developer.mozilla.org/ru/docs/Learn/CSS/Building_blocks/Cascade_and_inheritance (дата обращения 20.10.2022) - Текст: электронный.
2. Каскадность в CSS : сайт – 2016-2023 – URL: https://basicweb.ru/css/css_kaskadnost.php (дата обращения 20.10.2022) - Текст: электронный.
3. Построение правила css. Виды селекторов Построение правила css : сайт – 2018-2019 – URL: <https://studfile.net/preview/7475742/page:15/> (дата обращения 20.10.2022) - Текст: электронный.
4. Приоритеты в Css и их повышение за счет Important, комбинация и группировка селекторов, пользовательские и авторские стили : статья, автор: Дмитрий Петров – 07.01.2021 – URL: <https://ktonanovenkogo.ru/html/uroki-css/prioritety-css-important-kombinacii-gruppirovka-selektorov-polzovatelskie-avtorskie-stili.html> (дата обращения 20.10.2022) - Текст: электронный.
5. Современный учебник CSS : сайт – 2017 – URL: <http://m.siterx.ru/prodvizhenie/sozдание-sajtov/blog/uchebnik-css/azy-css/kaskadnost.html> (дата обращения 20.10.2022) - Текст: электронный.