

Cascading Style Sheets (CSS)

«каскадные таблицы стилей»

– это средство, позволяющее задавать различные визуальные свойства HTML-элементам.

Таблицы стилей - текстовые файлы, обычно имеющие расширение *.css.

HTML задает основную структуру веб-страницы, а также указывает, какие элементы на ней присутствуют.

Оформление веб-страницы, положение и вид элементов возложен на стили (CSS).

Добавление стилей на страницу

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
<head>
```

```
  <meta charset="UTF-8">
```

```
  <title>Css</title>
```

```
  <link rel="stylesheet" href="css/style.css">
```

Связывание

```
  <style>
```

```
    h3 {
```

```
      color: red;
```

```
      font-weight: bold;
```

```
    }
```

```
  </style>
```

Вложение

```
</head>
```

```
<body>
```

```
  <p style="color: green; font-weight: bold;">Встраивание</p>
```

Встраивание

```
  <h3>Вложение</h3>
```

```
</body>
```

```
</html>
```

Cascading Style Sheets (CSS)

Подключение через **встраивание** повышает время загрузки стилей, **затрудняет редактирование кода.**

Встраивание переопределяют вложенные и связанные стили.

Вложенные стили переопределяют связанные стили.

Правильнее всего добавлять стили из внешнего файла

К каждой веб-странице можно присоединить несколько таблиц стилей, добавляя последовательно несколько тегов <link>.

Преимущества использования таблиц стилей:

- ❖ разграничение кода и оформления;
- ❖ разное оформление для разных устройств;
- ❖ ускорение загрузки сайта;
- ❖ единое стилевое оформление множества документов;
- ❖ централизованное хранение

Правило @import

Позволяет загружать внешние таблицы стилей.

Чтобы директива @import работала, она должна располагаться в таблице стилей перед всеми остальными правилами, например:

```
@import url(other_style.css);
```

Правило @import также используется для подключения веб-шрифтов:

```
@import url('https://fonts.googleapis.com/css?family=Notable');
```

CSS - селектор

- это инструкция, которая указывает браузеру какой элемент выбрать для применения к нему стиля.

Браузер выбирает элемент и применяет к нему указанный стиль именно по СЕЛЕКТОРУ.



Селектор и блок объявлений образуют правило.

Таблица стилей состоит из набора правил.

Виды CSS селекторов

Селектор тегов (имя_тега) позволяют **форматировать все элементы** указанного типа.

h1 { **color**: **green**; } - задаст общий цвет всех заголовков h1

Селектор класса (.имя_класса) позволяют задавать **стили для элементов с одинаковым именем класса**.

Например, для создания абзаца с классом center необходимо добавить атрибут class со значением center в открывающий тег <p> и задать стиль для указанного класса.

<p class="center">Применяются стили класса center**</p>**

```
.center {  
  text-align: center;  
}
```

Виды CSS селекторов (продолжение)

Селектор идентификатора (#id_элемента) - позволяет форматировать один конкретный элемент, тк id должны быть уникальными

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

```
#container{  
    width: 700px;  
    background: yellow;  
}
```

Дочерний селектор (родитель > дочерний элемент) - позволяет применить стили только если дочерний элемент идёт сразу за родительским элементом.

p > em { font-size: 20pt; } - для всех тегов em, являющихся дочерними по отношению к элементу p установит указанный размер текста.

Виды CSS селекторов (продолжение)

Селектор потомка - применяют стили к элементам, расположенным внутри элемента-контейнера.

Например, `ul li { list-style-type: none; }` - уберет маркеры всех li, которые являются потомками ul.

Если нужно отформатировать **потомки определенного элемента**, этому элементу **нужно задать класс**:

`p.green a { color: green; text-decoration: none; }` - данный стиль применится ко всем ссылкам, потомкам абзаца с классом green;

`p .green a { color: green; text-decoration: none; }` - если добавить пробел, то будут стилизованы ссылки, расположенные внутри любого тега класса .green, который является потомком элемента <p>;

`.green a { color: green; text-decoration: none; }` - данный стиль применяется к любой ссылке, расположенной внутри другого элемента с классом .green.

Очередность применения CSS-стилей, наследование и каскад

1. Стили могут наследоваться от родительского элемента;
2. Стили, расположенные в таблице стилей ниже, отменяют стили, расположенные в таблице выше;
3. К одному элементу могут применяться стили из разных источников.

Наследование

- механизм, с помощью которого определенные свойства передаются от предка к его потомкам.

Предусмотрено **наследование свойств, относящихся к текстовому содержимому** страницы (color, font, letter-spacing, line-height, list-style, text-align, text-indent, text-transform, visibility, white-space и word-spacing).

Свойства, **относящиеся к форматированию блоков, не наследуются** (background, border, display, float и clear, height и width, margin, min-max-height и -width, outline, overflow, padding, position, text-decoration, vertical-align и z-index).

С помощью ключевого слова **inherit** можно принудить элемент наследовать любое значение свойства родительского элемента. Это работает даже для тех свойств, которые не наследуются по умолчанию.

Каскадирование

- механизм, который управляет конечным результатом в ситуации, когда **к одному элементу применяются разные CSS-правила.**

Критерии, которые определяют порядок применения свойств:

- ❖ правило `!important` будет иметь приоритет над всеми остальными правилами
`span { font-weight: bold!important; }`
- ❖ порядок, в котором подключены таблицы стилей. Если в разных таблицах будут встречаться разные значения свойств одного элемента, то в результате к элементу применяется правило, находящееся в таблице стилей, идущей в списке ниже;

Каскадирование

Критерии, которые определяют порядок применения свойств: (продолжение)

- ❖ специфичность. Для каждого правила браузер вычисляет специфичность селектора, и если у элемента имеются конфликтующие объявления свойств, во внимание принимается правило, имеющее наибольшую специфичность.

Значение специфичности состоит из четырех частей: 0, 0, 0, 0.

Специфичность селектора определяется следующим образом:

для id добавляется 0, 1, 0, 0;

для class добавляется 0, 0, 1, 0;

для каждого элемента и псевдоэлемента добавляется 0, 0, 0, 1;

для встроенного стиля -1, 0, 0, 0;

универсальный селектор (*) не имеет специфичности.

см. <https://htmlacademy.ru/courses/66/run/15>