

Введение в CSS

Глоссарий

- Каскадные таблицы стилей CSS (Cascading Style Sheets) – стандарт стилей, объявленный консорциумом W3C

Способы подключения таблиц стилей

- Встроенные (inline)
- Глобальные
- Внешние/связанные

Встроенные стили

```
<p style="color: #0000ff">Сегодня мы узнаем, что такое frontend.</p>
```

Плюсы:

- легко и быстро добавить
- не нужно беспокоиться о том, к какому элементу будет применен стиль

Минусы:

- нужно дублировать стили для нескольких элементов, даже если они одинаковые

Глобальные стили

Плюс :

- HTML и CSS в одном файле

Минусы:

- файл сильно разрастается
- тяжело ориентироваться

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8"/>
  <title>Frontend начало</title>
  <style>
    header,
    article,
    footer
    {
      width: 600px;
      margin: 0 auto;
    }
  </style>
</head>
```

Внешние стили

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8"/>
  <title>Frontend начало</title>
  <link href="css/style.css" rel="stylesheet">
</head>
```

Плюсы:

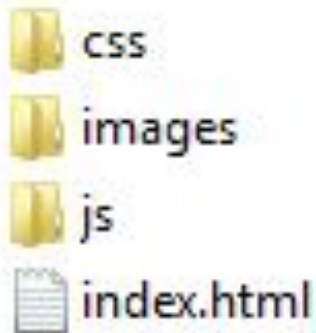
- можно подключать один файл на нескольких страницах
- стили можно подключать отложено, что позволит увеличить скорость загрузки страницы

Минусы:

- тяжело ориентироваться между файлами

В своей работе будем использовать внешние стили. Для того, чтобы было легче ориентироваться в файлах, будем использовать селекторы.

Структура вашего проекта будет такой (папки js пока не будет, добавим её, когда приступим к изучению js):



Глоссарий

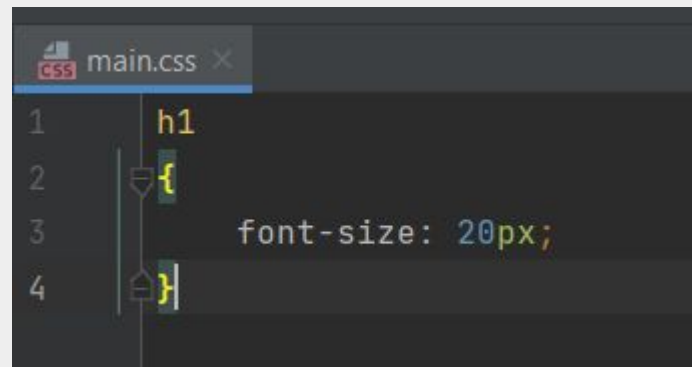
- Каскадные таблицы стилей CSS (Cascading Style Sheets) – стандарт стилей, объявленный консорциумом W3C
- Селекторы (от англ. select - выбирать) - это элементы каскадной таблицы стилей CSS, которые указывают на тот элемент на веб-странице, к которому должны будут применяться стили

Виды селекторов

1. Тэги
2. Классы
3. Идентификаторы
4. Атрибуты
5. Универсальный

Теги

В этом случае стили задаются
для тегов HTML

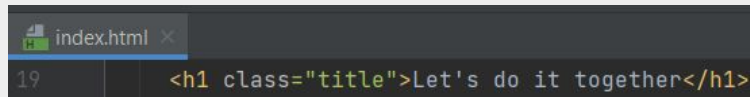


Классы

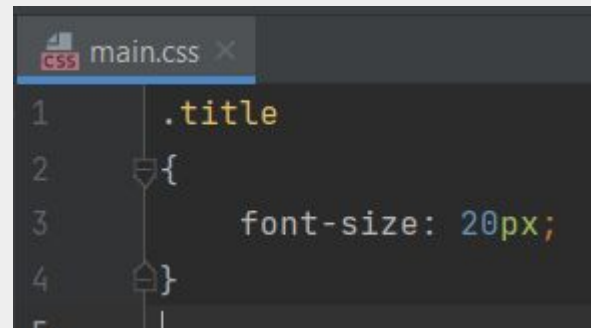
Для использования селектора "класс" нужно:

1. задать атрибут `class` для html-элемента
2. задать стили в css файле

Селектор класса указывается через точку - сначала точка, затем класс



```
index.html x
19 <h1 class="title">Let's do it together</h1>
```



```
main.css x
1 .title
2 {
3     font-size: 20px;
4 }
```

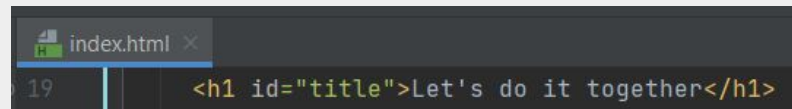
Идентификаторы

Для использования селектора "идентификатор" нужно:

1. задать атрибут `id` для html-элемента
2. задать стили в css файле

Селектор идентификатора указывается через символ `#` - сначала `#`, затем класс

PS: значение атрибута `id` должно быть уникальным на всей странице. То есть нельзя использовать один `id` для 2-х элементов



```
index.html x
19      <h1 id="title">Let's do it together</h1>
```



```
main.css x
1      #id
2      {
3          font-size: 20px;
4      }
```

Атрибуты

На самом деле можно использовать любой атрибут html-элемента. Для этого в css файле нужно использовать следующий шаблон:
[<название атрибута>=<значение атрибута>]

Значение атрибута необязательная часть. Также можно использовать различные правила (подробнее [тут](#))

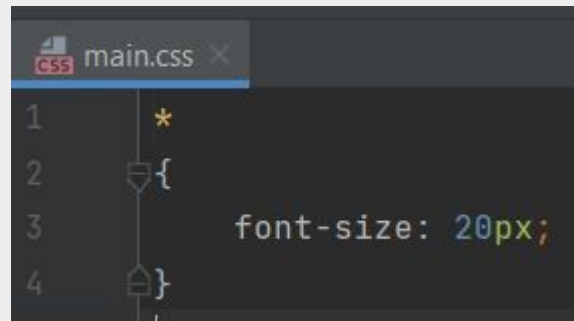
```
index.html x
19 <h1 lang="en">Let's do it together</h1>
```

```
main.css x
1 [lang="en"]
2 {
3     font-size: 20px;
4 }
```

Универсальный

Данный стиль используется для всех элементов

Для этого нужно использовать символ *



Псевдо

- **Псевдоэлемент** позволяет стилизовать определенную часть элемента. Например, первый или последний дочерний элемент. Подробнее [тут](#)
- **Псевдокласс** позволяет стилизовать определенное состояние элемента. Например, наведённое состояние. Подробнее [тут](#)

Сколько весят селекторы

0	0	0	0
style=""	#id	.class	<tag>

Все css-селекторы имеют свой вес. Для определения веса селектора нужно подставить "1" в соответствующую ячейку. Например, если мы зададим стиль через идентификатор (id), её вес будет 100 у.е., а если добавим стиль через селектор класса, то её вес будет 10 у.е.

Таким образом, чтобы переопределить стили, заданные через id, нужно задать минимум 10 раз стили через класс.

Что используем

Для задания стилей нужно использовать атрибут `class`, так как он является наиболее удобным для работы.

CSS свойства

Стилизация текстов

- `font-size` - размер текста
- `font-family` - шрифт
- `font-weight` - толщина текста
- `line-height` - межстрочный интервал (в относительном значении)
- `font-style` - стиль текста (например, курсив)

Сокращенно пишется с помощью свойства `font`

<http://htmlbook.ru/html/font>

Подключение шрифтов

Нестандартные стили нужно загружать из <https://fonts.google.com/>

```
11  
12 <link href="https://fonts.googleapis.com/css2?family=Montserrat:wght@100&display=swap" rel="stylesheet">  
13
```

ФОН

Для отображения фона нужно использовать свойство background

<http://htmlbook.ru/css/background>

display

max-width; max-height;

border

border-radius

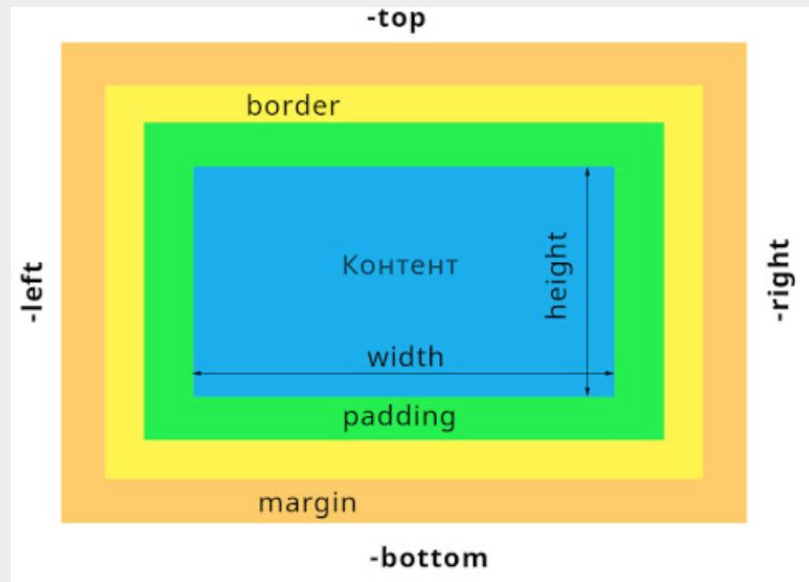
text-align

Отступы

Для задания рамки вокруг элемента используется свойство `border`
Отступ внутри рамки задается с помощью свойства `padding`
Отступ вне рамки задается с помощью свойства `margin`

Чтобы внутренний отступ (`padding`) входил в ширину элемента, элементу нужно задать свойство `box-sizing: border-box`

PS: чтобы свойство `padding` работало, отображение `border` не обязательно. Оно есть всегда, просто толщина его 0px



Еще больше про css свойства в <http://htmlbook.ru/css>

БЭМ

БЭМ

БЭМ – методология, в основе которой лежит принцип разделения интерфейса на независимые блоки.

БЭМ расшифровывается как Блок, Элемент, Модификатор

Глоссарий

- Каскадные таблицы стилей CSS (Cascading Style Sheets) – стандарт стилей, объявленный консорциумом W3C
- БЭМ – методология, в основе которой лежит принцип разделения интерфейса на независимые блоки

Блок

Блок – это независимый элемент страницы, который может использоваться повторно

Элемент

Элемент – это часть страницы, который не может использоваться без своего родителя (блока)

При этом элемент может быть самостоятельным блоком в одном случае и элементов внутри другого блока в другом случае

Модификатор

Модификатор – это сущность, которая используется для добавления стилей, отличающих блок или элемент от других блоков или элементов.

Например, если на странице есть 5 одинаковых блоков по всем стилям, только у одного из них другой background

Именование классов

В именовании классов мы будем использовать блоки

- Название блока указывается просто по выбранному названию, например, `main`
- Название элемента указывается через двойное подчёркивание после названия блока, например `main__text`
- Название модификатора указывается через одинарное подчёркивание после названия блока или элемента, например `main__text_bold`

Порядок всегда соответствует названию методологии Блок->Элемент->Модификатор

Можно писать Блок->Модификатор (`main_bold`), но нельзя писать
Блок->Модификатор->Элемент (`main_bold__text`)

Больше про БЭМ можно прочитать в

<https://ru.bem.info/methodology/quick-start/>

Соглашение о кодировании

https://docs.google.com/document/d/1yD83BfsYanvfLKhetA5lM1hxw_GukEDw/edit?usp=sharing&ouid=117714276428562136793&rtpof=true&sd=true