

# HTML/CSS

Путь, классы и идентификаторы, стили

Файл стилей

Стили по редактированию текста

# Тег <img>

- ▶ Тег <img> предназначен для отображения на веб-странице изображений в графическом формате GIF, JPEG или PNG
- ▶ Тег непарный
  - ▶ Основные атрибуты тега:
    - ▶ src = “путь к файлу”
    - ▶ alt = “невидимое описание”
    - ▶ title = “описание, которое видно при наведении”

Подсказка: не забывайте прописывать alt для картинок, это очень важный атрибут для поисковых систем

# Атрибут src

- ▶ Атрибут src предназначен для подключения к нашей веб-странице внешних файлов (картинок, файлов JS и т.д.)
- ▶ Используется:
  - ▶ Для подключения мульти-медиа файлов (картинок, видео, GIFF анимации)
  - ▶ Для подключения фалов с кодом JavaScript и многим другим
- ▶ Значением атрибута “src” является путь к файлу. Путь к файлу бывает:
  - ▶ Относительный - относительно расположения главного файла (img/img.jpg)
  - ▶ Абсолютный - относительно корневой папки (C:\Work\Site\img\img.jpg)

Подсказка: старайтесь использовать относительный путь к файлу, т.к. при переносе сайта возникнут огромные проблемы

# Атрибут класса (class)

```
<div class="wrapper"></div>
```

- ▶ Задаёт стилевой класс, который позволяет связать определённый тег со стилевым оформлением. В значении допускается указывать сразу несколько классов, разделяя их между собой пробелом.
- ▶ Имена классов могут содержать в себе латинские буквы (A-Z, a-z), цифры (0-9), символ дефиса (-) и подчёркивания (\_). Использование русских букв в классах недопустимо.
- ▶ Имена классов у разных объектов могут быть одинаковыми, что очень расширяет возможности наложения стилей. Для того, чтобы не повторять написания одинаковых стилей для объектов их можно применять к разным объектам.

Подсказка: если видите одинаковые объекты, с одинаковыми стилями - создайте общий стиль и используйте его для всех похожих объектов

# Атрибут идентификатора (id)

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

- ▶ Идентификатор (называемый также «ID селектор») определяет уникальное имя элемента, которое используется для изменения его стиля и обращения к нему через скрипты.
- ▶ ID должен быть УНИКАЛЬНЫМ и может быть назначен всего один раз одному объекту.
- ▶ Объект не может содержать несколько ID
- ▶ Все свойства ID имеют больший приоритет нежели свойства класса.

Подсказка: значение ID используется так же для анкорных ссылок

# Атрибут стилей Style

```
<div style="color:red">Lorem</div>
```

- ▶ Атрибут style применяется для определения стиля элемента с помощью правил CSS.
- ▶ В качестве значений указываются стилевые правила: вначале следует имя стилевого свойства, затем через двоеточие его значение. Силевые свойства разделяются между собой точкой с запятой.
- ▶ Имеет наивысший приоритет среди всех объявлений стилей.

Подсказка: используйте этот атрибут максимально редко, т.к. его свойства ничем не перебиваются за исключением JS

# Подключение стилей

- ▶ Стили можно подключать как в <HEAD> тегом <STYLE>
- ▶ Так же стили можно подключать через внешний файл (рекомендуется использовать именно такое подключение)
- ▶ Примеры подключение стилей:

```
<head>
  <meta charset="UTF-8">
  <style>
    p {
      color: white;
    }
  </style>
  <title>Достоевский: мальчик у христа на елке</title>
</head>
```

```
<head>
  <meta charset="UTF-8">
  <link rel="stylesheet" href="style.css">
  <title>Достоевский: мальчик у христа на елке</title>
</head>
```

Подсказка: хорошим тоном считается подключение стилей внешним файлом. Так вы можете расширять зоны действия свойств стилей.

# Создание и подключение файла стилей

- ▶ Файл стилей имеет расширение .css (.sass, .scss - препроцессоры)
- ▶ Файлов со стилями можно подключить сколько угодно, главное потом не запутаться в них ;)
- ▶ Файл стилей подключается внутри тега <head> тегом <link>, который имеет 2 атрибута:
  - ▶ href - путь к нашему файлу стилей;
  - ▶ rel = "stylesheet", который подсказывает интерпретатору, что в этом файле подключаются стили.

```
<head>  
  <meta charset="UTF-8">  
  <link rel="stylesheet" href="style.css">  
  <title>Достоевский: мальчик у христа на елке</title>  
</head>
```

Подсказка: не забывайте, что интерпретатор идет сверху вниз и если в 2 файлах попадутся одинаковые селекторы стилей, то сработает тот, который был подключен последним



# Селекторы

- ▶ **Селектор** - это часть CSS-правила, которая сообщает браузеру, к какому элементу (или элементам) веб-страницы будет применён стиль.
- ▶ Основные селекторы:
  - ▶ `. myClass` - выбирает все элементы с классом `myClass` (`class="myClass"`).
  - ▶ `#main` - выбирает элемент с идентификатором `main` (`id="main"`).
  - ▶ `*` - выбор всех элементов.
  - ▶ `span` - выбор всех элементов `<span>`.
  - ▶ `[title="cost"]` - выбирает все элементы с атрибутом `title`, значение которого в точности совпадает со значением указанным в селекторе (`title="cost"`).

# Практика создания и написания фала стилей

- ▶ Создадим в папке с нашим index.html файл style.css
- ▶ Откроем style.css через SubLime и попробуем написать первый стиль для нашего текста.
- ▶ Давайте зададим всему, что обернуто у нас в тег <p> красный цвет текста, а так же поэкспериментируем с классами и id;
- ▶ В нашем css файле должна появиться строка:

```
p {  
    color: red;  
}
```
- ▶ Это означает, что всему контенту, который обернут в тег <p> будет задан цвет текста - красный. (если, конечно наш файл подключен в нашем index.html)

Подсказка: для написания стилей так же хорошим тоном является вложенность, со временем вы поймете на сколько это облегчает задачу и возможность управления стилями

# Примеры использования селекторов класса и идентификатора

Элемент с данным селектором будет иметь цвет текста - белый

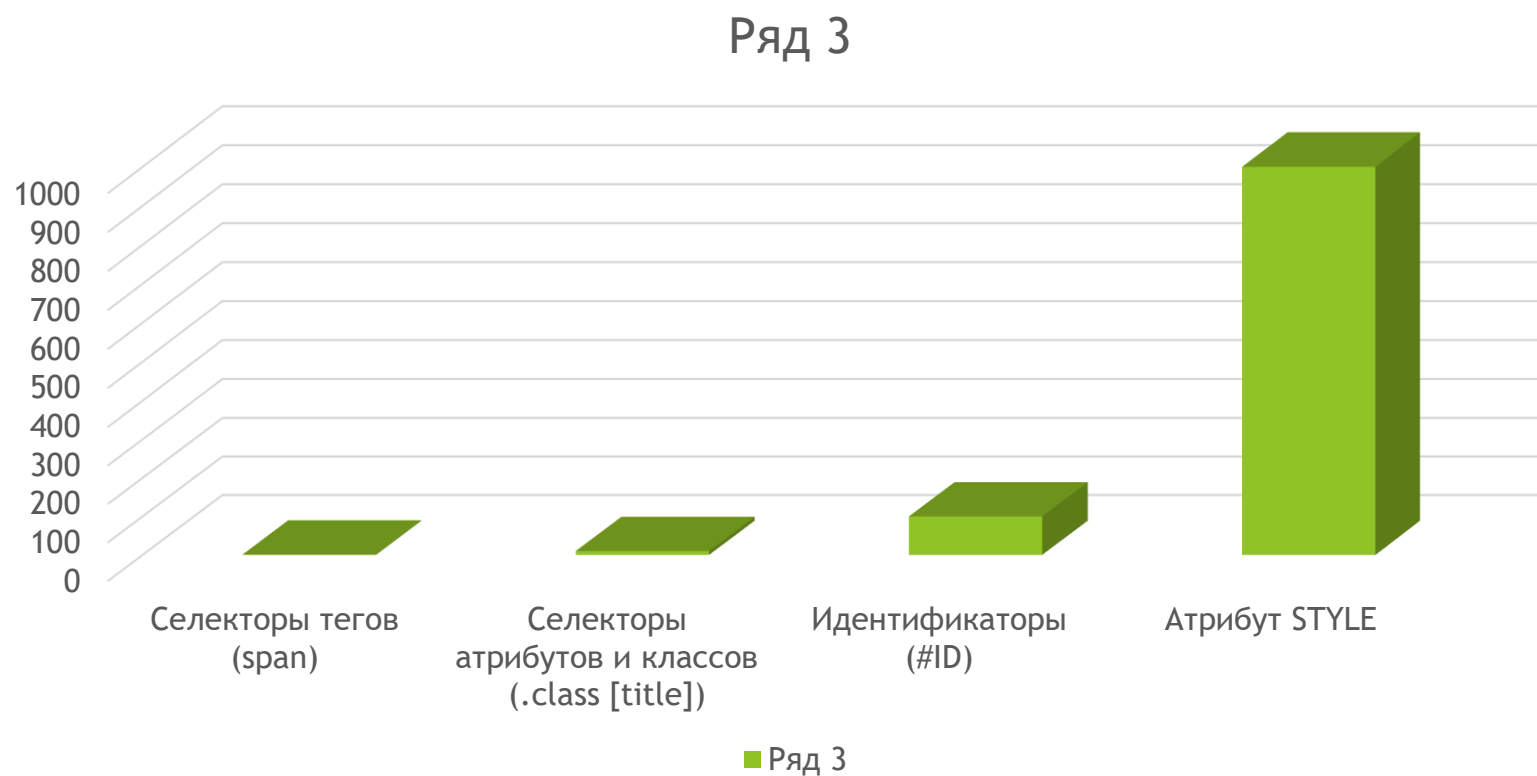
#ID

```
untitled
1 #myID {
2   color: white;
3 }
```

.Class

```
untitled
1 .myClass {
2   color: white;
3 }
```

# Приоритеты селекторов



Подсказка: есть правило !important, оно имеет самый высокий приоритет из вышеперечисленных, желательно данное правило использовать редко, т.к. его становится очень тяжело перебить

# Normalize.css

- ▶ Все браузеры имеют разные стили по умолчанию. Где-то больше отступ слева, где-то больше межстрочный интервал и т.д..... Для того, что бы Ваш сайт отображался на всех браузерах одинаково, существует библиотека normalize.css. Она приводит к общему знаменателю все стили браузера. И Ваш сайт будет отображаться одинаково на всех браузерах.
- ▶ Скачать ее можно по адресу: <http://necolas.github.io/normalize.css/> - здесь вы всегда найдете свежую и последнюю версию данной библиотеки.

Подсказка: normalize.css необходимо подключать самой первой, для того, что бы она (библиотека) не перебивала подключенные нами стили.

# Единицы измерения объектов и цвета

- ▶ Единицы измерения значений бывают абсолютные и относительные:
  - ▶ Относительные: px, %, em, rem
  - ▶ Абсолютные: in, cm, mm, pt, pc
- ▶ Единицы измерения цвета:
  - ▶ Константы (white, black, blue)
  - ▶ HEX (#FFF, #F0C3B4)
  - ▶ RGBA (255,255,255,0.5)

# Основные свойства селекторов для написания стилей текста.

- ▶ Основные стили, которые вы будете использовать для форматирования текста:
  - ▶ Color
  - ▶ Font-size
  - ▶ Text-transform
  - ▶ Text-align
  - ▶ Text-decoration
  - ▶ Line-height
  - ▶ Letter-spacing

# Свойство Color

- ▶ Color - css свойство, которое задает цвет текста.
- ▶ Принимает значения в разных системах исчисления: Hex, RGBA, а так может задаваться встроенными значениями браузера (white, blue, green).

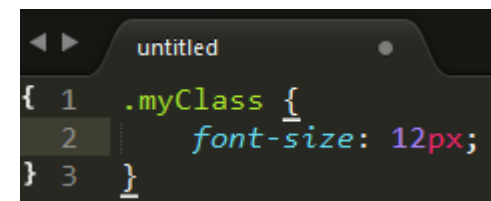
```
p {  
  color: red;  
}
```

Подсказка: лучше использовать систему счисления HEX или RGBA



# Font-size

- ▶ Определяет размер шрифта элемента.
- ▶ Также разрешается использовать любые допустимые единицы CSS: em (высота шрифта элемента), ex (высота символа x), пункты (pt), пиксели (px), проценты (%) и др. За 100% берется размер шрифта родительского элемента. Отрицательные значения не допускаются.

A screenshot of a code editor window titled 'untitled'. It shows a CSS class definition: 

```
{ 1  .myClass {  
 2      font-size: 12px;  
 3  }
```

 The text is color-coded: the class name is green, the opening brace is blue, the property name is blue, the value is red, and the closing brace is blue. Line numbers 1, 2, and 3 are visible on the left.

Подсказка: для адаптивной верстки во многих случаях очень уместно использовать единицу счисления rem

# Text-transform

- ▶ Управляет преобразованием текста элемента в заглавные или прописные символы. Когда значение отлично от none, регистр исходного текста будет изменен.
- ▶ Значения:
  - ▶ capitalize - первый символ каждого слова в предложении будет заглавным. Остальные символы свой вид не меняют.
  - ▶ lowercase - все символы текста становятся строчными (нижний регистр).
  - ▶ uppercase - все символы текста становятся прописными (верхний регистр).
  - ▶ none - не меняет регистр символов.
  - ▶ inherit - наследует значение родителя.

# Text-align

- ▶ Определяет горизонтальное выравнивание текста в пределах элемента.
- ▶ Значения:
  - ▶ Left - Выравнивание текста по левому краю.
  - ▶ Right - Выравнивание текста по правому краю.
  - ▶ Justify - Выравнивание по ширине - одновременно по левому или правому краю.
  - ▶ Center - выравнивает текст по центру.
  - ▶ Start / End - задает направление текста

Подсказка: в основном вы будете использовать первые четыре свойства, но и про остальные 2 надо знать ;)

# Text-decoration

- ▶ Добавляет оформление текста в виде его подчеркивания, перечеркивания, линии над текстом и мигания.
- ▶ Значения:
  - ▶ line-through - Создает перечеркнутый текст (~~пример~~).
  - ▶ Overline - Линия проходит над текстом.
  - ▶ Underline - Устанавливает подчеркнутый текст (пример).
  - ▶ None - Отменяет все эффекты, в том числе и подчеркивания у ссылок, которое задано по умолчанию.

# Line-height

- ▶ Устанавливает интерлиньяж (межстрочный интервал) текста, отсчет ведется от базовой линии шрифта.
  - ▶ Синтаксис:
    - ▶ Line-height: множитель | значение | проценты | normal | inherit
  - ▶ Значения:
    - ▶ Normal - значение по умолчанию
    - ▶ Inherit - наследует у родителя

Подсказка: очень часто line-height используют для отцентровки маленького текста по середине блока

# Letter-spacing

- ▶ Определяет интервал между символами в пределах элемента.
- ▶ Синтаксис:
  - ▶ letter-spacing: значение | normal | inherit
- ▶ Значения:
  - ▶ Normal - Задаёт интервал между символами как обычно.
  - ▶ Inherit - Наследует значение родителя.