



# CSS. Каскадные таблицы стилей: основные свойства



Easy  
UM

IT ОБУЧЕНИЕ  
В МОСКВЕ

Умным быть легко!

# Темы модуля



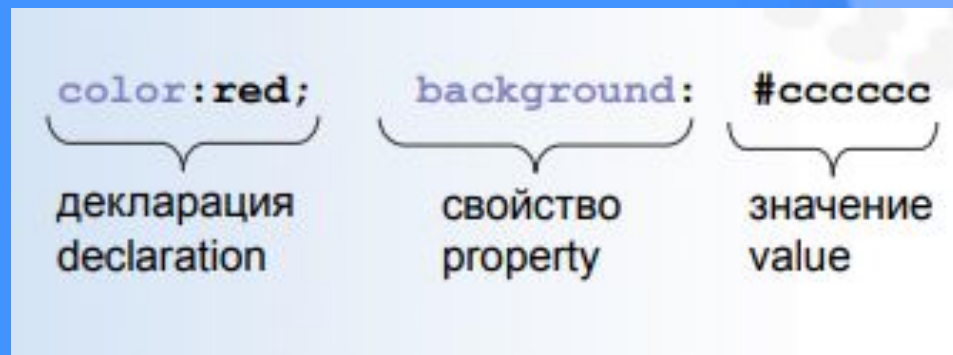
- Основные цели и задачи CSS.
- Способы добавления стилей на web-страницу.
- Модульная структура CSS3
- Наследование
- Селекторы
- Принципы каскадирования и принципы группировки
- Псевдоклассы (псевдоселекторы)
- Псевдоэлементы
- Работа с цветом и фоном, со шрифтами
- Работа с текстом
- Инструменты разработчика

# CSS – это...



CSS — это язык стилей, который определяет отображение HTML-документов. Актуальная версия – CSS 3, релиз которой был в 2012 году. Периодически появляются новые функции, которые поддерживаются не во всех браузерах.

## Правила декларирования стилей:



# Способы размещения CSS



- Вложение (inline)
  - `<p style="color:red; background:#FF0000">`
- Встраивание (embedding)
  - `<style type="text/css">`  
`p {color:red; background:#FF0000}`  
`</style>`
- Связывание (linking)
  - `<link rel="stylesheet" type="text/css"`  
`href="style.css"/>`

# Селекторы



**Селекторы** – правила выбора элементов на странице для изменения их CSS-свойств.

**Выделяют селекторы:**

- Селекторы по элементу
- По идентификатору (атрибуту id)
- По классу (атрибуту class)
- Селекторы по атрибутам
- Селекторы псевдо-классов
- Селекторы псевдо-элементов

Селектор элемента <div>

```
div{  
  color: #ffc  
}
```

# Декларация класса и id



```
<p class="text">Текст</p>
```

```
<style>  
  .text {  
    color: red;  
  }  
</style>
```

```
<p id="text">Текст</p>
```

```
<style>  
#text {  
  color: red;  
}  
</style>
```

Помните, что стили через id труднее перезаписать, а также то, что один id может использоваться только один раз на всю страницу. Не рекомендуется прибегать к стилизации по id без крайней необходимости.

# Каскадирование



Под каскадированием понимается одновременное применение разных стилевых правил к элементам документа — с помощью подключения нескольких стилевых файлов, наследования и перезаписи свойств.

Согласно этому принципу каждая более приоритетная запись декларации CSS заменяет значения менее приоритетной.

# Приоритеты по возрастанию



\* /\*все элементы\*/

**div** /\*тег\*/

**div:first-of-type** /\*тег с псевдоселектором\*/

**div p** /\*родитель — вложенный элемент\*/

**div span+p** /\*родитель + соседний элемент\*/

**.red** /\*класс\*/

**div p.red** /\*родитель и тег с классом\*/

**p.red.level** /\*тег с 2+ классами\*/

**#id1** /\*id (идентификатор)\*/

**#id1 #id2** /\*вложенные id\*/



# Максимальный приоритет



В CSS есть своего рода “kozyрной туз”, который перекрывает все остальные правила, заданные ранее – это запись `!important`, заданная сразу после значения свойства.

```
<style>
  p {
    color: red !important; /*перебивает любые ранее
    заданные записи для цвета тега <p>*/
  }
</style>
```

# Чем плох !important



Эту запись не рекомендуется применять по следующим причинам:

- 1) огромная сложность в перезаписи (вплоть до того, что перезаписать стили не представляется возможным в принципе);
- 2) захламление кода ненужными декларациями.

Для любых манипуляций гораздо лучше использовать принципы каскадирования и специфичности.

# Специфичность



Если к одному элементу одновременно применяются противоречивые стилевые правила, то более высокий приоритет имеет правило, у которого значение специфичности селектора больше.

```
<style>
  div p {
    color: red; /*эта запись применится к тегу p внутри div*/
  }

  p {
    color: blue; /*эта - не применится к тегу p внутри div*/
  }
</style>
```

# Дополнение



Также всегда более приоритетны стили, написанные строчно в HTML (однако это не значит, что этим способом стоит пользоваться)

```
<p style=""></p>
```

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

```
<link href="style2.css" rel="stylesheet">
```

А в данном случае значения из файла style2.css будут перекрывать аналогичные из style.css

# Фоновое изображение



Фоновое изображение можно разместить у любого элемента при помощи CSS

```
background-image:url(path/to/image.jpg)
```

# Другие частоиспользуемые правила



background-repeat: no-repeat (не делать паттерн из картинки)

background-size: contain (вместить) | cover (растянуть) | 100% (любые величины – проценты, пиксели и т.д.)

background-position: top | bottom | center | left | right

(позиционирование фона по одному или двум краям, также может быть в пикселях или других величинах)

# Цвет



- Есть следующие способы декларации цвета:
- название цвета (red, green, white...)
- #00cc00, #ffffff, #000000 - шестнадцатеричное представление
- #0c0, #fff, #000 - сокращенное шестнадцатеричное представление
- rgb(0,240,125) - в формате RGB
- rgba(0,240,125, .5) - в формате RGB с прозрачностью в 50%
- rgb(0%,80%,25%) - в формате RGB, относительное

# Селекторы класса и идентификатора, контекстные селекторы



```
.main{color: #555}  
#back{color:red}  
div#back{color:black}  
div b{color:green}  
td td td{color:blue}
```

**Селектор по классу .main**

**Селектор по идентификатору**

**Контекстные селекторы  
(выбор в зависимости)**



# Группировка селекторов



```
h1 { font-family: sans-serif }  
h2 { font-family: sans-serif }  
h3 { font-family: sans-serif }  
=   
h1, h2, h3 {  
font-family:  
sans-serif }
```

# Единицы измерения



- **Относительные** (широко используются в Вебе)
  - **px - pixels**, пикселы, самые универсальные единицы
  - **em (16px) - ems**, высота используемого элементом шрифта. Применяется для текстов, обладает уникальным свойством сложения значений у элементов со вложенностью.
  - **rem (16px) – rems**, аналогично em, но без сложения значений при наследовании
  - **%** - значение относительно родительского элемента в процентах
  - **vw, vh** – viewport-width, viewport-height (процент от видимой ширины или высоты экрана)

# Пиксель — это не пиксель



- В вебе 1px не равно одному физическому пикселю.
- Количество пикселей разнится от экрана к экрану. Все экраны различаются по плотности пикселей, диагонали и т.п., и поэтому установить пиксель, равный физическому, невозможно. Браузеры используют свою модель построения.
- Согласно спецификации W3 1 дюйм = 96px, однако такая модель измерения не дает нам реального понимания размеров. Поэтому все браузеры обрабатывают пиксели только с учетом разрешения экрана и его диагонали.

# Практические примеры



- Таким образом, в iPhone 5 или iPhone SE браузер видит только 320 пикселей в ширину, а в iPad – 768px. Общепринято, что ширина контейнера на десктопах обычно около 1100-1280px.
- Эти моменты мы будем подробнее разбирать на следующих занятиях, а сейчас важно запомнить, что пиксель – это именно относительная величина, а не абсолютная, и браузер может определять такие величины, как полпикселя или 1/10 пикселя. По этой причине запись 16.5px, или даже 16.05px – абсолютно нормальна.

# Малоиспользуемые и неиспользуемые



- **Абсолютные** (использовать в очень частных случаях)
- **in - inches**, дюймы
  - **cm - centimeters**, сантиметры
  - **mm - millimeters**, миллиметры
  - **pt - points**, пункты( $1\text{pt} = 1/72\text{in} = 0,35\text{mm}$ )
  - **pc - picas**, пики( $1\text{pc} = 12\text{pt}$ )

Также есть такие относительные величины, как **ex** и **ch**.

**Ex** – это высота символа **x**, а **ch** – ширина символа **0** (ноль). Они тоже используются в частных случаях из-за сложности в понимании таких величин.

# Псевдоклассы

a:link (до клика)	a:link{color:blue}
:visited (после клика)	a:visited{color:blue}
:active (при клике)	a:active{color:red}
:hover (при наведении)	a:hover{color:green}
:focus	input:focus{color:red}



# Псевдоэлементы



Псевдоэлементы позволяют задать стиль элементов не определённых в дереве элементов документа, а также генерировать содержимое, которого нет в исходном коде текста.

```
p:before {  
    content: "Текст перед тегом <p>" /*обязательное свойство*/  
}
```

```
p:after {  
    content: "Текст после тега <p>" /*обязательное свойство*/  
}
```

# Домашняя работа



Создайте файл `style.css` и подключите стилевой файл во всех HTML файлах своего домашнего задания, не прибегая к редактированию стилей внутри HTML

Сделаем тёмную тему. Присвойте на всех страницах следующие цвета:

- фону страницы – чёрный
- тексту страницы – белый

**Также:**

Присвойте произвольные цвета для ссылок в навигации.

Присвойте произвольные цвета для ссылок в тексте.