



## Уеб разработка

# CSS или как да разкрасим HTML

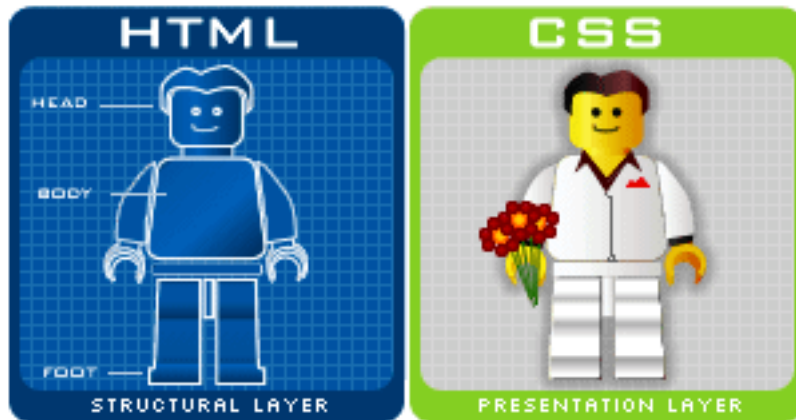
# Съдържание

- CSS
- CSS правило
- Inline стилизиране
- CSS селектори
- `<style></style>` - вграден стил
- външен CSS файл
- Стилизиране на текстове, Граници, Фонове
- Варианти за задаване на цвят
- Други – width/height, display

CSS

# CSS ...?

- **CSS - Cascading Style Sheets**
- CSS описва как **HTML** елементите да изглеждат върху екрана, хартия или др. среда за визуализация.
- **CSS спестява много работа** – С един файл може да контролираме как ще изглеждат множество уеб-страници.
- Препоръчително е стилизирането на уеб-страниците да бъде описано в **един външен CSS файл.**



# CSS ...?

## Задаваме

- големина,
- разстояния,
- шрифтове,
- цветове,
- разположение HTML на елементите...

*Създаден е за да раздели презентирането от съдържанието.*

Заради CSS, всички HTML тагове и атрибути, свързани с презентиране са забранени за използване - font, center, etc

# CSS ...?

## *info*

Със CSS може да представим и стилове

- за различна резолюция на екрана,
- за различни среди за визуализация на уеб-страниците –
  - екран,
  - печат на принтер...
  - дори и за брайлов четец.

# Cascading(?) style sheets

**Важно** Child-елементите наследяват почти всички стилови правила на родителските елементи. Можем да зададем правила за конкретния/те child-елементи за да изменим родителските стилове.

## Наследяване

**Да** - свързаните със оформлението на текст и списъци -

- Color, font, size, font-family, line-height, text-align, list-style, и др.

**Не** - отнасящи се до Box-модела и позициониране, големина :

- Width, height, border, margin, padding, position, float и др.
- <a> елементите не наследяват color и text-decoration

# Cascading(?) style sheets

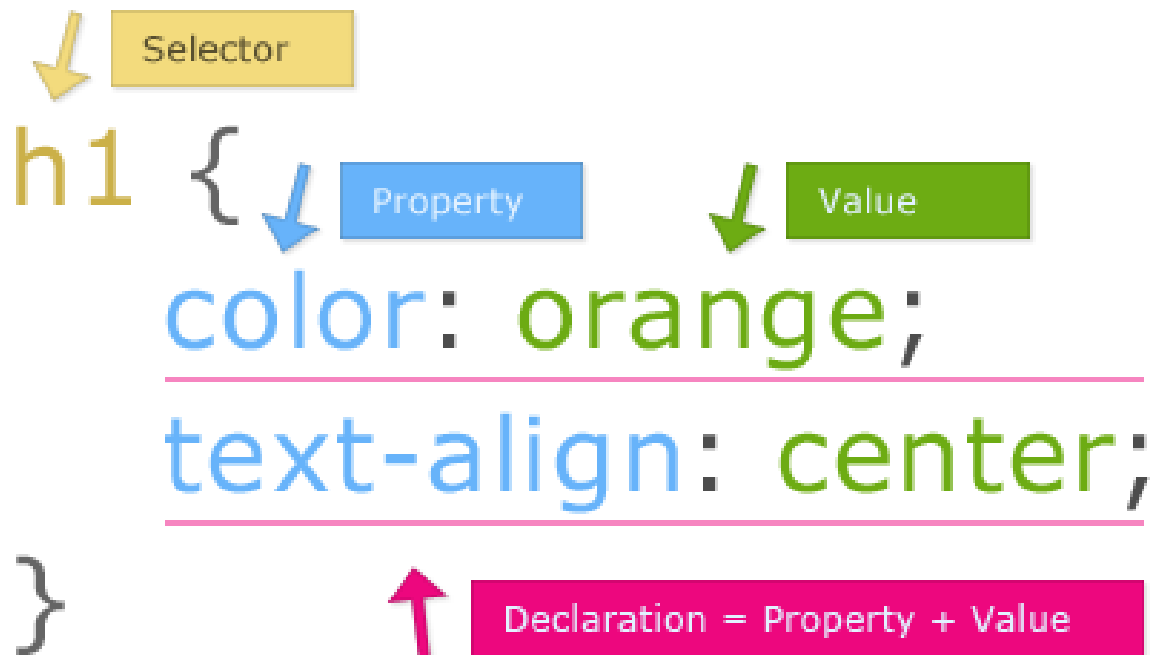
***Важно** style.css се чете отгоре надолу – всяко следващо правило презаписва предишното, ако се отнасят за един и същи елемент!*



# CSS ПРАВИЛО

# CSS правило

## Anatomy of a CSS Rule



# Как да стилизираме ...

Inline

```
<h1 style="color:blue;margin-left:30px;">This is a heading.</h1>
```

Не се препоръчва, освен в случаите, когато се задава размер на снимка

```

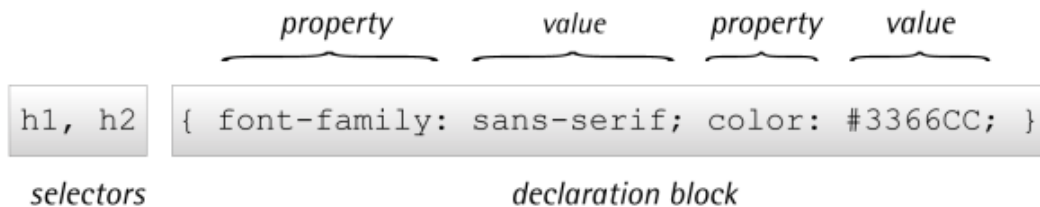
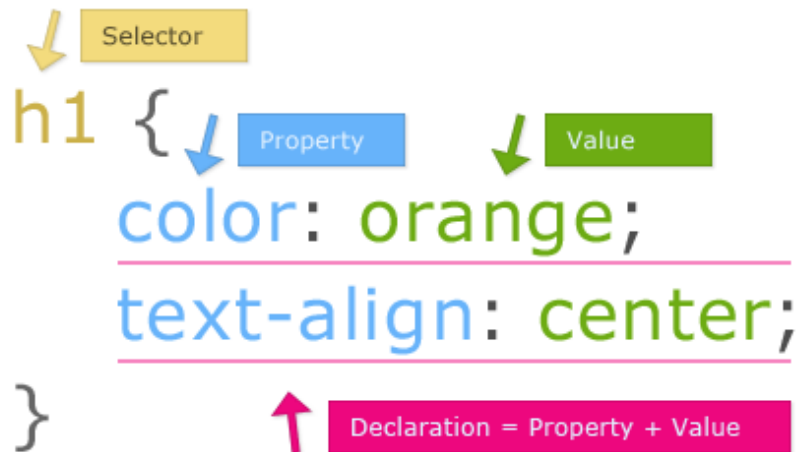
```

Браузърът заделя място за снимката и така при зареждане на страницата, текста не “подскача” при зареждане на снимките

# CSS правило с включен селектор



## Anatomy of a CSS Rule



# Селектори



# Селектори

- Селекторите определят за кой елемент се отнася CSS-правилото
- Посочват елементи
  - от тип /по таг/
  - тези, които имат специфичен атрибут – id, class
  - отговарящи на конкретно влагане на HTML- елементи /описват пътя до елемента, който селектираме/

```
p {  
    background-color: red;  
}  
div, p, ul, li {  
    margin: 0;  
    padding: 0;  
}  
  
.plain, #nice, #ugly {  
    color: blue;  
}  
  
div#header div#nav ul li {  
    margin-left: 20px;  
}
```

# Селектори - 2

- Може да зададем един селектор
- Или да изброим няколко селектора - разделяме ги със запетая.
- Един HTML елемент може да притежава един или няколко класа и едно единствено id. Класовете и id се задават според конкретния дизайн, който трябва да постигнем чрез CSS или по друг повод /напр. id за label при input елементите на формите/
- `<tag-name id="unique" class="class1 class2 ...">`
- *За целия документ имаме едно единствено id!*

# Как да стилизираме ...

Вграден стил в HTML документа

```
<head>  
  <style>  
    body {  
      background-color: linen;  
    }  
  
    h1 {  
      color: maroon;  
      margin-left: 40px;  
    }  
  </style>  
</head>
```



# CSS STYLESHEETS

# Как да стилизираме ...

## 3. Външен **.css** файл.

По този начин може да променяме външния вид на целия уебсайт чрез промени само в един файл.

Всяка страница на уебсайта трябва да има добавена връзка към .css файла

```
<head>  
    <link rel="stylesheet" type="text/css" href="mystyle.css">  
</head>
```

href="..."

- посочва пътя към файла

# Важно!!!

*Важно* Приоритет на стиловете

- Браузър стиловете се
- презаписват от външния файл, който се
- презаписва от вътрешния стил, който се
- презаписва от inline стила

# CSS - ресурси

- Допълнителни ресурси
- <http://www.w3schools.com/css/default.asp>
- <https://www.codecademy.com/>
- CSS – tips and tricks
- Murach's HTML5 and CSS3 (Training & Reference), 2015



# ВАЖНО!!!!!!

Организирайте файловете си по следния начин

Папка – проект

вложени папки	– img/images – тук съхраняваме всички снимки
	- js – тук съхраняваме всички JavaScript файлове
	- styles – тук съхраняваме .css файловете
файлове -	index.html – зарежда се по подразбиране
	other.html
	other.html

'''

Имената на файловете само с latinski\_bukvi

Ако държите имената да са съставени от няколко думи, свързвайте ги с долно тире – dolno\_tire.

# СТИЛИЗИРАНЕ НА ТЕКСТОВЕ

# Задаване свойства на текстовете

- `color` – цвят на текста – като `blue`, `#000000(#000)` или `RGB (num1, num2, num3)`  
/sublime color picker – избирате цвета директно в програмата/
- `font-size` – големина на шрифта – в `px` /засега/
- `font-family` – разделени със запетая имена на шрифтове.
  - Например: `verdana, sans-serif`, и т.н.
  - браузъра зарежда първия наличен шрифт.
- Винаги трябва да има поне един по-общ шрифт като например – `sans-serif`
- `font-weight` - `normal`, `bold`, `bolder`, `lighter` или число `[100 ... 900]` /не винаги има разлика в начина, по който изглеждат буквите при различните числа/

# Задаване свойства на текстовете

- `font-style` – `normal`, `italic`, `oblique`
- `text-decoration` – `none`, `underline`, `line-through`, `overline`, `blink`

/най-честа употреба за стилизиране на `<a>`

- `text-align` – дефинира разположението на текста или друго съдържание - `left`, `right`, `center`, `justify`



**Border & background**

# background

1. Задаваме цвят за фон —

```
body {  
  background-color: #000;  
}
```

[Изображение за фон](#) — разгледайте самостоятелно и пробвайте за домашно

```
body {  
  background-image: url("paper.gif");  
}
```

# border

1. [w3s borders info](#)
2. border-style – вида на линията, която ще очертае границата, solid – най-често използваната
3. border-width – ширината на линията в px /засега/
4. border-color
5. Описваме всичките свойства на един ред

Или всяко свойство отделно – според ситуацията

```
p.one {  
    border-style: solid;  
    border-width: 5px;  
    border-color: red;  
}
```

```
td {  
    border: 1px solid #541A1A;  
}
```

# Border

Може да стилизираме горната, долната, лявата и дясната граница по различен начин.

```
td {  
    border-top: 1px solid #541A1A;  
    border-left: 2px dotted #D20909;  
    border-bottom: 3px outset #138220;  
    border-left: 2px inset #1F42D9;  
}
```

Може да стилизираме една или повече от границите /не е задължително да задаваме свойствата и на четирите граници/

# ВАЖНО!!!!

При писане на `style.css` се стремим с минимално количество правила да обхванем максимално количество елементи.

- ако дадено правило се отнася за повече от един елемент избираме селектор, с който да обхванем максимален брой елементи.

- ако прилагаме правило към повече от един селектор – изброяваме селекторите със запетая и прилагаме правилото ...

- обединяваме всички правила, отнасящи се за даден селектор и ги изброяваме, по възможност, еднократно в `{}` , разделени с точка и запетая и т. н.

**Обобщение**

# Обобщение

## Cascading Style Sheets

- с един файл контролираме външния вид на множество уеб страници
- приоритет на стиловете – стилове по подразбиране за всеки браузър, външен файл, вграден стил в документа, inline – с най-висок приоритет /препоръчва се само за задаване размер на снимка/
- стиловите правила се наследяват – част от тях – за оформление на текст и списъци – от child – елементите.
- стиловите правила могат да бъдат презаписвани – селектираме необходимия елемент/и и задаваме правила за него.
- всяко следващо правило презаписва предишното, ако пред скобите стои селектор, таргетиращ един и същи елемент/и

# Обобщение

- Използваме съкратен или пълен вариант за изписване на правила
  - border: .... или border-style

Винаги се стремим с минимален брой правила да стилизираме максимален брой елементи, като се стараем да не повтаряме правила и селектори.

CSS отговаря за презентацията на html елементите. HTML тагове, свързани с презентацията на елементите не се използват.