

HTML&CSS 2

уеб разработка с PHP/2020

Милена Томова
Vratsa Software

<https://vratsasoftware.com/>

Table of Contents

1. CSS
2. CSS правило
3. CSS селектори
4. Добавяне на CSS - варианти
5. Стилизиране на
 - текстове,
 - Граници,
 - Фонове,
 - Варианти за задаване на цвят
6. CSS box-model



CSS

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

- ★ Със CSS може да предоставим и стилове
 - за различна резолюция на екрана,
 - за различни среди за визуализация на веб-страницата –
 - екран,
 - печат на принтер...
 - дори и за брайлов четец.

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

Да - child-елементите наследяват стиловете на родителите си, свързаните със оформлението на текст и списъци -

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

Не - child-елементите НЕ наследяват стиловете на родителите си, отнасящи се до Box-модела, позициониране и големина :

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

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

The background of the slide features a light gray, stylized illustration of a mountain range with sharp peaks and a dense forest of evergreen trees in the foreground. The illustration is rendered in a sketch-like style with fine lines.

CSS ПРАВИЛО

Anatomy of a CSS rule

h1 - selector

```
h1 {
```

```
  color: #ccc;  
  text-align: center  
}
```

color - property

color: #ccc;
declaration = property + value

center -
value

Anatomy of a CSS rule

more than one selectors

comma
separated

```
h1, h2,  
h3, h4 {  
    color: #ccc;  
    text-align: center  
}
```



Selectors

- Селекторите определят за **кой елемент** се отнася **CSS-правилото**
 - елементите се посочват - **по tag**

```
div, p {  
    font-size: 2rem  
}
```

- Селекторите определят за кой елемент се отнася CSS-правилото
 - елементите се посочват - **id**, **class**

```
div#id-name, p.class-name {  
    font-size: 2rem  
}
```

```
<div id="class-name">
```

A div to style

```
<p class=""> A paragraph to style </p>
```

```
</div>
```


Selectors

index.html

```
<div id="id-name">
```

A div to style

```
</div>
```

```
<p class="class-name">
```

A paragraph to style

```
</p>
```

style.css

```
div#id-name p.class-name {
```

```
    font-size: 2rem
```

```
}
```


- Може да зададем

Или да изброим няколко селектора, разделени със запетая.

```
a {  
  едителеелектор  
  decoration:none  
  color: #ccc  
}
```

```
div, p, a {  
  margin: 0;  
  padding: 0;  
}
```

множество елементи с еднакви CSS
правила

- 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/*

```
<p id="unique" class = "class1 class-name2  
class-different-name-3 ...">
```

paragraph content

```
</p>
```

**В документа id стойностите
не трябва да се повтарят!**



Добавяне на CSS

INLINE

като атрибут на елемента

```
<div style="color: #ccc">
```

....

```
</div>
```

Не се препоръчва!!!

- труден за управление при големи проекти
- с най-висок приоритет - презаписва всички останали варианти на задаване на стил

Вграден стил в HTML документа

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

Не се препоръчва!!!

- труден за управление при големи проекти
 - втори по приоритет - презаписва правилата зададени в **.css файл**



inline CSS

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 /снимките винаги се зареждат по-бавно от текста/.

Добавяне на CSS

Външен .css файл.

- Стиловите правила на сравнително голям проект, могат да бъдат обединени в един файл.
- Така може да променяме външния вид на целия уебсайт чрез промени само в един единствен 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 **.css** файла



Допълнителни ресурси

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



Приоритет на стиловете

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

The background of the slide features a light gray, stylized illustration of a mountain range with sharp peaks and a dense forest of evergreen trees in the foreground. A large, solid dark blue circle is centered over the image, containing the title text in white.

СТИЛИЗИРАНЕ НА ТЕКСТ



**!important
DEMO**

Стилизиране на текст

- **color** – цвят на текста – като blue, #000000(#000) или RGB (num1, num2, num3)
/sublime color picker – избирате цвета директно в програма/
- **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/

Стилизиране на текст

- **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

//Цвет за фон

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

//Изображение за фон – разгледайте самостоятелно

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

- [w3s borders info](#)
- **border-style** – вида на линията, която ще очертае границата, solid – най-често използваната
- **border-width** – ширината на линията в px /засега/
- **border-color**
- Описваме всичките свойства на един ред
- Или всяко свойство отделно – според ситуацията

```
p.one {  
    border-style: solid;  
    border-width: 1px;  
    border-color: #fff  
}
```



```
p.one {  
    border: 1px solid #fff  
}
```


четирите страни на рамката около елемента, могат да бъдат стилизирани различно

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



CSS box-model

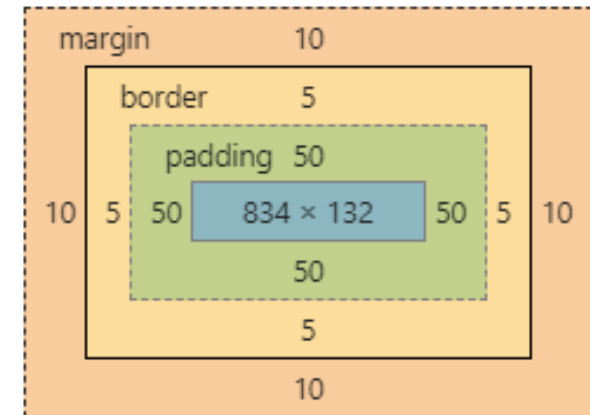
CSS box-model

index.html

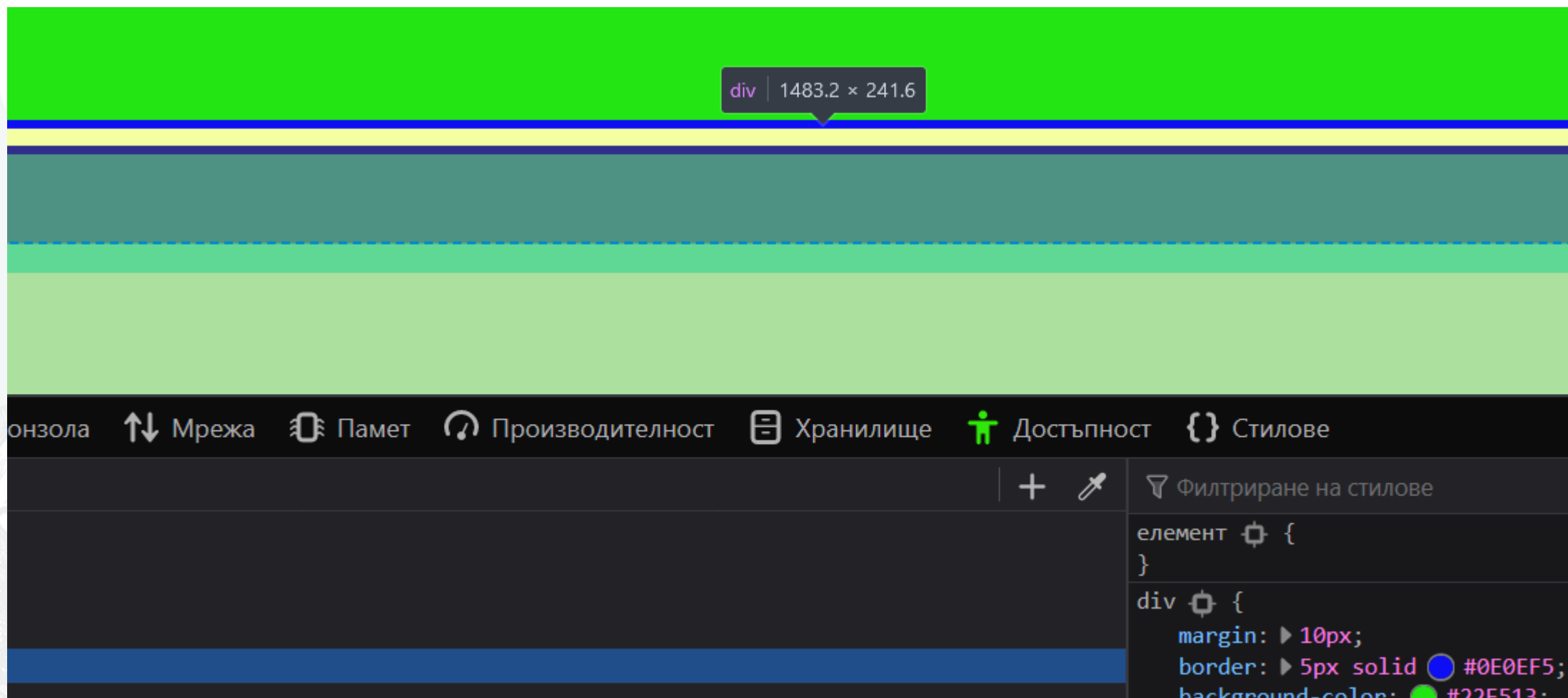
```
<div>  
    <p>Text</p>  
</div>  
<div>  
    <p>Text</p>  
</div>
```

style.css

```
div {  
    margin: 10px;  
    border: 5px solid #0E0EF5;  
    background-color: #22E513;  
    padding: 50px;  
}  
  
p {  
    background-color: #E0FB2B;  
    height: 100px;  
}
```



CSS box-model



CSS box-model

Когато блоков елемент няма зададени

width и height

заема място =

размер на съдържанието му

+ padding

+ border

+ margin

CSS box-model

Инлайн елемент -

трябва да промените поведението му с

display: inline-block или

display: block

за да управлявате мястото, което ще заема
според **box-model**



display

display

CSS display

Променяме поведението на елементите -

display: inline блоковите елементи стават инлайн
елементи

display: block инлайн елементите стават блокови

display: inline-block

елементите са инлайн /редят се един до друг, ако
има място и блокови /управляват размера им чрез
width/height, padding/

display

CSS display

Променяме поведението на елементите -

display: hidden елементът се скрива, но мястото му не се освобождава

display: none елементът се скрива и мястото му се заема от следващите елементи

с **display: block/inline** или **inline-block/**

възстановявате елемента - той се вижда отново в страницата/



position

position

CSS
position
static

Елементите се редят отляво надясно и отгоре надолу, по реда, на записване в html документа.

position: static

Позиция по подразбиране на всеки елемент.

position

CSS position relative

position: relative;

top: 10px;

left: 20px;

Елементът е изместен спрямо позицията си по подразбиране

Когато имате няколко елемента е рисковано да ги позиционирате с **position: relative;** точно.

Нямате контрол върху поведението им.

position

CSS
position
absolute

position: absolute;

top: 10px;

left: 20px;

Елементът е позициониран спрямо рамката на родителския си елемент.

За да работи коректно, родителския елемент трябва да има позициониране различно от static.

**Не задавайте размери и координати с точност до пиксел.
Оставяйте 2-5 пиксела свободни!**

position

CSS
position
fixed

position: fixed;

top: 10px;

left: 20px;

Елементът е позициониран на определено място на екрана, което не се променя при скрол.

- 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

Questions?



Гнездото
Coworking

Цялостен
курс по
програми
ране

Дизайн
курс

Курс по
дигит.
маркетинг

MindHub



Partners



**Telerik
Academy**



MindHub

ПРОМЯНАТА

Trainings @ Vratsa Software



- Vratsa Software – High-Quality Education, Profession and Jobs
 - www.vratsasoftware.com
- The Nest Coworking
 - www.nest.bg
- Vratsa Software @ Facebook
 - www.fb.com/VratsaSoftware
- Slack Channel
 - www.vso.slack.com

