



WEB ACADEMY

Front-end за Начинаещи



За CSS

3

- CSS – съкратено от *Cascading Style Sheets*

Какво ново има в CSS3?

- Селектори
- Backgrounds
- Рамки
- Текстови ефекти
- 2D/3D Трансформации
- Анимации
- Потребителски интерфейс



Вмъкване на CSS



- чрез деклариране на външен файл вътре в `<head>` тага
`<link rel="stylesheet" type="text/css" href="file.css" />`
- чрез използване на `<style>` таг вътре в `<head>` тага
`<style>`
`<!--`
`.content { width:425px; }`
`---->`
`</style>`
- чрез директно задаване на стил на ниво елемент
`<i class="content" style="color:red;">Text</i>`

Запомнете:

1. Добрата практика е да се използва първия начин
2. Стилите на ниво елемент препокриват предишните дефиниции

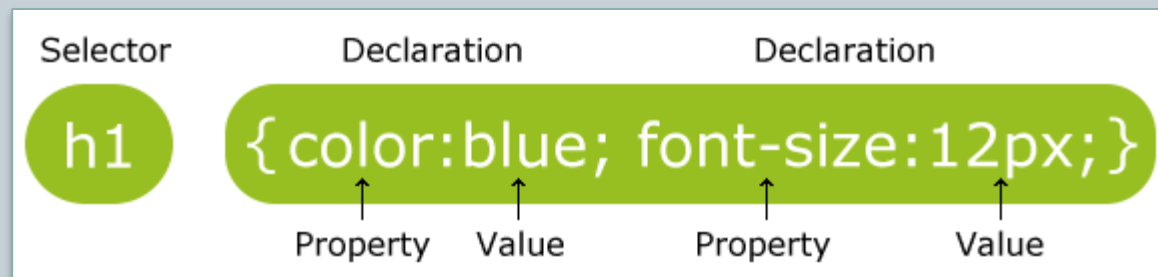


Синтаксис



Няколко основни понятия за синтаксиса:

- Селектор - указва елемента, който искаме да променим
- Декларация - съдържа една или повече дефиниции разделени с точка и запетая ";" помежду си
- Всяка декларация се състои от свойство и стойност, разделени с двуточие ":"



Мерни единици



В CSS се използват следните мерни единици:

- px - задава размера в пиксели.
- em - изчислява размера спрямо размера на шрифта.
- pt - задава размера в точки. Използва се често при печат.
- % - задава размера в проценти.
- Други мерни единици:
 - pc (picas)
 - cm (centimeters)
 - mm (millimeters)
 - in (inches)

Например:

```
a
{
  width: 50%;
  font-size: 2em;
  height: 15px; /* Няма празен интервал */
  margin: 0; /* Това е нула :-) */
}
```



Няма празен интервал между цифрата и мерната единица!



Цветовете



Цветовете в CSS се могат да се задават по няколко начина:

- шестнадесетично - #ff0000
- чрез RGB - rgb(255, 0, 0)
- чрез RGBA - rgba(255, 0, 0, 0.3)
- чрез HSL - hsl(120, 100%, 50%)
- чрез HSLA - hsla(120, 100%, 50%, 0.3)
- чрез името си - Red, Green, Blue, Maroon...



Задачи за самоподготовка върху цветовете:

- Задайте прозрачност посредством RGBA
- Задайте цвят посредством HSL
- Задайте прозрачност посредством HSLA
- Задайте прозрачност чрез цветови формат по избор



Многократни дефиниции?!



Нека помислим:

```
<html>
<head>
  <style>
    .c1{ color: green !important; }
    .c2{ color: blue; font-weight: bold; }
    .c3{ color: red; font-style: italic; }
  </style>
</head>
<body>
  <p class="c2">Text1</p><!-- Blue -->
  <p class="c3">Text2</p><!-- Red -->
  <!-- Ами сега... ??? -->
  <p class="c3 c2">Text3</p>
  <i class="c1" style="color:red;">T</i>
  <p class="c1 c3">Text5</p>
</body>
</html>
```



Това е така, защото:

- има значение кой клас е дефиниран последен
- няма значение реда на класовете при използване
- !important - служи да забрани предефинирането на дадено свойство, т.е. все едно казва: "Това е важно! Игнорирай ги!"
- Стилосете на ниво елемент препокриват предишните дефиниции. Те винаги печелят!

Повече информация: [тук](#)



Нека Силата бъде с Вас!



a
1 x element selector
Sith: **0, 0, 1**



p a
2 x element selectors
Sith: **0, 0, 2**



.whatever
1 x class selector
Sith: **0, 1, 0**



a.whatever
1 x element selector
1 x class selector
Sith: **0, 1, 1**



p a.whatever
2 x element selectors
1 x class selector
Sith: **0, 1, 2**



.whatever .whatever
2 x class selectors
Sith: **0, 2, 0**



p.whatever a.whatever
2 x element selectors
2 x class selectors
Sith: **0, 2, 2**



#whatever
1 x id selector
Sith: **1, 0, 0**



a#whatever
1 x element selector
1 x id selector
Sith: **1, 0, 1**



.whatever a#whatever
1 x element selectors
1 x class selector
1 x id selector
Sith: **1, 1, 1**

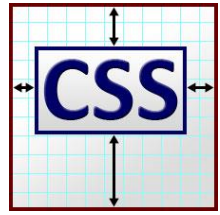


.whatever .whatever #whatever
2 x class selectors
1 x id selector
Sith: **1, 2, 0**



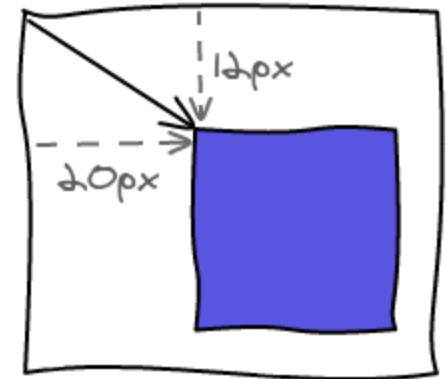
#whatever #whatever
2 x id selectors
Sith: **2, 0, 0**

Позициониране на елементи



CSS свойства за задаване позицията на елементите:

- position
- top
- left
- right
- bottom
- z-index



Работа с Фон (Background)



Свойства за манипулиране на фона:

Property	Description
<u>background</u>	Sets all the background properties in one declaration
<u>background-attachment</u>	Sets whether a background image is fixed or scrolls with the rest of the page
<u>background-color</u>	Sets the background color of an element
<u>background-image</u>	Sets the background image for an element
<u>background-position</u>	Sets the starting position of a background image
<u>background-repeat</u>	Sets how a background image will be repeated



Форматиране на текст



Свойства за работа с текст:

- color
- direction
- letter-spacing
- line-height
- text-align
- text-decoration
- text-indent
- text-shadow
- text-transform
- unicode-bidi
- vertical-align
- white-space
- word-spacing

Йордан
Енев Web Academy
Избери бъдеще в твой град! С
нами Богъ! България ще пребъде! Ако
не намираш път, създай път! Боже пази
България! Web Academy Избери бъдеще в
твой град! WebAcademy.bg EnevSoft.net Работи и
живей достойно в България! Ние сме като свита
пружина. Ние сме опъната тетива. Ние сме стрела
преди полет. Ние сме българи. Йордан Енев Web
Academy Избери бъдеще в твой град! С нами Богъ!
България ще пребъде! Ако не намираш път, създай
път! Боже пази България! Web Academy Избери
бъдеще в твой град! WebAcademy.bg EnevSoft.net
Работи и живей достойно в България! Ние сме като
свита пружина. Ние сме опъната тетива. Ние сме
стрела преди полет. Ние сме българи. Йордан
Енев Web Academy Избери бъдеще в твой
град! С нами Богъ! България ще
пребъде! Станете част от Уеб
Академи - заедно ще
променим България!
Амин!

Работа с шрифт



Какво ни е нужно:

- font
- font-family
- font-size
- font-style
- font-variant
- font-weight

Ad Lib BT
Alako-Bold
Bernard

Freehand 575 BT
Galaxy BT
Garmond

Poster Bodoni
Script MS Bold
Secret Recipe

Edwardian Script

Dobkin

Lucinda calligraphy

Amaze

Brook Script

Freehand591 BT

Kristen ITC

Adler

BernardTango BT

BradleyHand ITC

Park Avenue

Pristina

French Script MS

Harrington

T4C Beaulieux *Ghostly*

Black Chancery

Rockwell *Americana BT*

Elphinstone

Blackadder JTC **Patrick**

Celtic Garmond

Gothikka

Cloister Black BT

Chalk

Childish Alpha

BoysRGross

Curlz MT

Burton's Nightmare

Comic Sans MS

Times New Roman

Haunt

DRACULA

Psycho Poetry

Black Jack

Old English



Списъци

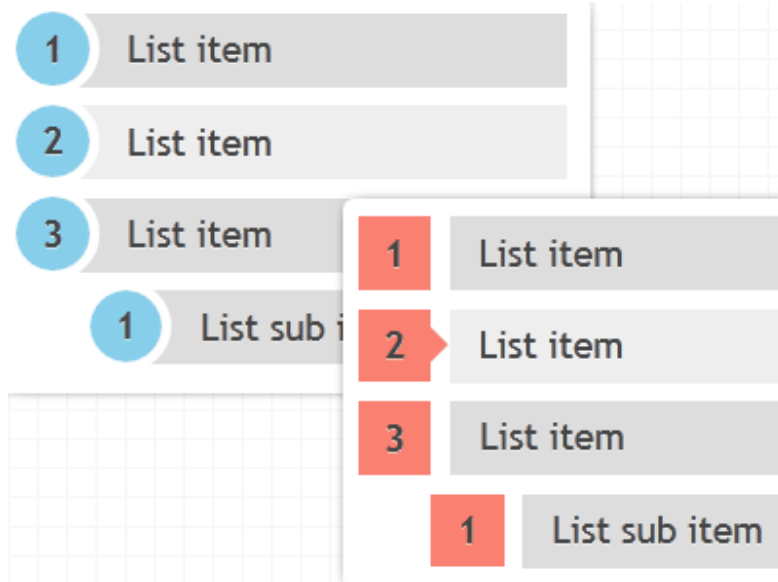


Благодарение на CSS можете:

- задавате маркер за списъците
- задавате къде да се показват маркерите
- задавате изображение, което да ползвате като маркер

И всичко това само с 4 свойства:

- list-style-image
- list-style-position
- list-style-type
- list-style



"Кутиен" модел



Подредба от вътре навън:

- content
- padding
- border
- margin



```
/* Имаме следната дефиниция */  
div {  
  width: 320px;  
  padding: 10px;  
  border: 5px solid gray;  
  margin: 0;  
}
```

```
/* Колко е крайната ширина?! */  
Проста математика:  
320px (width)  
+ 20px (left + right padding)  
+ 10px (left + right border)  
+ 0px (left + right margin)  
= 350px
```


Хипервръзки



За да се манипулират хипервръзките се използват техните 4 състояния:

- link - непосетена хипервръзка
- visited - посетена хипервръзка
- hover - "посочен" с курсора линк
- active - линк в момента на кликане

 Запомнете че:

- последователността им на деклариране има значение
- винаги се декларира точно в този ред
- този въпрос често се задава при интервюта, тестове и т.н.

Примери:

```
/* unvisited link */  
a:link { color: red; }  
/* visited link */  
a:visited { color: green; }  
/* mouse over link */  
a:hover { color: hotpink; }  
/* selected link */  
a:active { color: blue; }o
```


Селектори



В CSS се използват следните основни селектори:

- елемент (таг) - селектира всички тагове от този тип
- идентификатор (id) - селектира по атрибута id
- клас - селектира по атрибута class
- групов селектор - селектира група от елементи
- Повече информация и списък с всички селектори: [ТУК](#)

Примери:

```
p { color:red; } /* селектира всички параграфи */  
#main { color:red; } /* селектира само 1 елемент с id="main" */  
.bgo { border:1px solid red; } /* селектира елементите с class="bgo" */  
h1, h2, p { float:right; color:blue } /* селектира изброените елементи */
```



Името на клас, или ID НЕ бива да започва с цифра!



Селектори



В CSS също така често се използват и други селектори:

Selector	Example	Example description
<u><a>element,element</u>	div, p	Selects all <div> elements and all <p> elements
<u><a>[attribute]</u>	[target]	Selects all elements with a target attribute
<u><a>[attribute=value]</u>	[target=_blank]	Selects all elements with target="_blank"
<u><a>[attribute~=value]</u>	[title~=flower]	Selects all elements with a title attribute containing the word "flower"
<u><a>[attribute =value]</u>	[lang =en]	Selects all elements with a lang attribute value starting with "en"
<u><a>[attribute^=value]</u>	a[href^="https"]	Selects every <a> element whose href attribute value begins with "https"
<u><a>[attribute\$=value]</u>	a[href\$=".pdf"]	Selects every <a> element whose href attribute value ends with ".pdf"
<u><a>[attribute*=value]</u>	a[href*="WA"]	Selects every <a> element whose href attribute value contains the substring "WA"
<u><a>:not(selector)</u>	:not(p)	Selects every element that is not a <p> element
<u><a>:nth-child(n)</u>	p:nth-child(2)	Selects every <p> element that is the second child of its parent
<u><a>:nth-of-type(n)</u>	p:nth-of-type(2)	Selects every <p> element that is the second <p> element of its parent
<u><a>:only-of-type</u>	p:only-of-type	Selects every <p> element that is the only <p> element of its parent
<u><a>:only-child</u>	p:only-child	Selects every <p> element that is the only child of its parent
<u><a>:optional</u>	input:optional	Selects input elements with no "required" attribute
<u><a>:required</u>	input:required	Selects input elements with the "required" attribute specified



Псевдо класове



Псевдо класовете се използват за да селектират дадено състояние на елемент. Задават се чрез ':'

Най-често се използват за:

- промяна на стила при `hover` или `focus` събитие
- промяна стила на посетени/непосетени линкове

Най-често използвани псевдо класове:

Selector	Example	Example description
<u>:active</u>	<code>a:active</code>	Selects the active link
<u>:checked</u>	<code>input:checked</code>	Selects every checked <code><input></code> element
<u>:disabled</u>	<code>input:disabled</code>	Selects every disabled <code><input></code> element
<u>:empty</u>	<code>p:empty</code>	Selects every <code><p></code> element that has no children
<u>:enabled</u>	<code>input:enabled</code>	Selects every enabled <code><input></code> element
<u>:first-child</u>	<code>p:first-child</code>	Selects every <code><p></code> elements that is the first child of its parent



Псевдо елементи



Псевдо елементите се използват за да селектират част от даден елемент. Задават се чрез '::'

Най-често се използват за:

- вмъкване на съдържание преди/след елемента
- задаване стил на първа буква, първи ред и други

Selector	Example	Example description
<u>::after</u>	p::after	Insert something after the content of each <p> element
<u>::before</u>	p::before	Insert something before the content of each <p> element
<u>::first-letter</u>	p::first-letter	Selects the first letter of each <p> element
<u>::first-line</u>	p::first-line	Selects the first line of each <p> element
<u>::selection</u>	p::selection	Selects the portion of an element that is selected by a user

- Повече информация за псевдо елементи и класове: [тук](#)



Комбинатори



Комбинаторите се използват за да зададат отношенията между отделните селектори.

В CSS се използват следните основни комбинатори:

- [интервал] - комбинатор за наследници (descendant)
- > - комбинатор за връзка дете-родител (child)
- + - комбинатор за съседен събрат (adjacent sibling)
- ~ - комбинатор за общ събрат (general sibling)

Примери:

```
div p { color: red;} /* селектира всички <p>, които са някъде в <div> */  
div > p{ color: red;} /* селектира всички <p>, които са деца на <div> */  
/* Всички деца са наследници, но всички наследници НЕ са деца ;-) */  
div + p{ color: red;} /* селектира всички <p>, веднага след <div> */  
div ~ p{ color: red;} /* селектира всички <p>, които братя на <div> */
```



Комбинатори



```
<!DOCTYPE html>
<html>
<head>
<style>
div { border:1px solid blue; background-color:lightblue; }
p { border:1px solid red; background-color:white; }
div p { font-weight: bold; }
div > p { font-style: italic; }
div + p { background-color:lightblue; font-weight: bold; color:white;}
div ~ p { background-color:lightblue; }
p ~ div { border-style:dashed; }
</style>
</head>
<body>
<p>Параграф 0 е преди div-a. Р НЕ Е брат на Div-a, но Div-a е брат на Р, и дори негов
съсед!</p>
<div>
  <p>Параграф 1 е в div. Той е наследник И дете едновременно!</p>
  <p>Параграф 2 е в div. Той също е наследник И дете едновременно!</p>
  <span><p>Параграф 3 е в span, който е в div. Р е наследник на Div, НО НЕ Е дете!</p>
</span>
</div>
<p>Параграф 4 е на едно ниво с div-a. Той е негов брат И и дори съсед!</p>
<p>Параграф 5 е на едно ниво с div-a. Той е негов брат, НО НЕ Е негов съсед</p>
</body>
</html>
```

Параграф 0 е преди div-a. Р НЕ Е брат на Div-a, но Div-a е брат на Р, и дори негов съсед!

Параграф 1 е в div. Той е наследник И дете едновременно!

Параграф 2 е в div. Той също е наследник И дете едновременно!

Параграф 3 е в span, който е в div. Р е наследник на Div, НО НЕ Е дете!

Параграф 4 е на едно ниво с div-a. Той е негов брат И и дори съсед!

Параграф 5 е на едно ниво с div-a. Той е негов брат, НО НЕ Е негов съсед

Още задачи върху комбинатори:

- Задача 1
- Задача 2
- Задача 3
- Задача 4

Рамки



- Свойства за заобляне на рамки (Rounded Corners)

Property	Description
<u>border-radius</u>	A shorthand property for setting all the four border-*-*-radius properties
<u>border-top-left-radius</u>	Defines the shape of the border of the top-left corner
<u>border-top-right-radius</u>	Defines the shape of the border of the top-right corner
<u>border-bottom-right-radius</u>	Defines the shape of the border of the bottom-right corner
<u>border-bottom-left-radius</u>	Defines the shape of the border of the bottom-left corner

```
border-radius: 2em 1em 4em / 0.5em 3em;  
/* Двата записа са еквивалентни. */  
border-top-left-radius: 2em 0.5em;  
border-top-right-radius: 1em 3em;  
border-bottom-right-radius: 4em 0.5em;  
border-bottom-left-radius: 1em 3em;
```

Стойностите преди / са отстъпа по X,
а след наклонената черта са по Y

Рамки

24

- Свойства за задаване на изображение като рамка:

Property	Description
<u>border-image</u>	A shorthand property for setting all the border-image-* properties
<u>border-image-source</u>	Specifies the path to the image to be used as a border
<u>border-image-slice</u>	Specifies how to slice the border image
<u>border-image-width</u>	Specifies the widths of the border image
<u>border-image-outset</u>	Specifies the amount by which the border image area extends beyond the border box
<u>border-image-repeat</u>	Specifies whether the border image should be repeated, rounded or stretched

```
#borderimg {  
  border: 10px solid transparent;  
  padding: 15px;  
  border-image: url(border.png) 30 round;  
}
```

