



WEB ACADEMY

Front-end за Начинаещи



CSS

3a 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

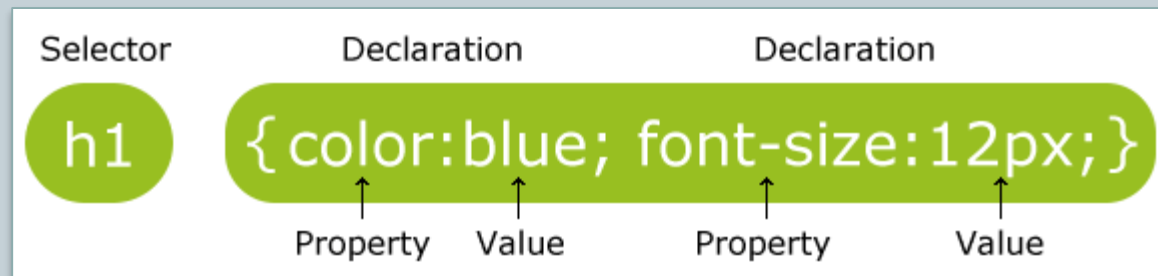


Синтаксис



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!"
- Стиливете 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**



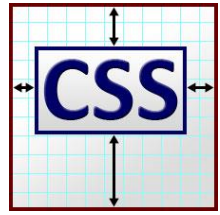
#whatever #whatever

2 x id selectors

Sith: **2, 0, 0**

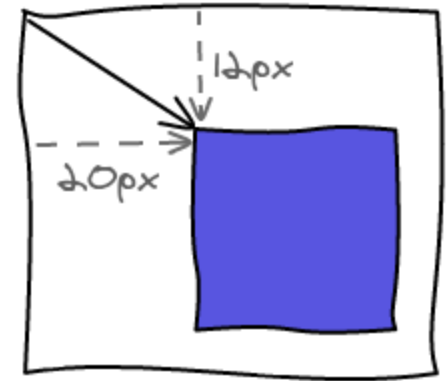


Позициониране на елементи



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

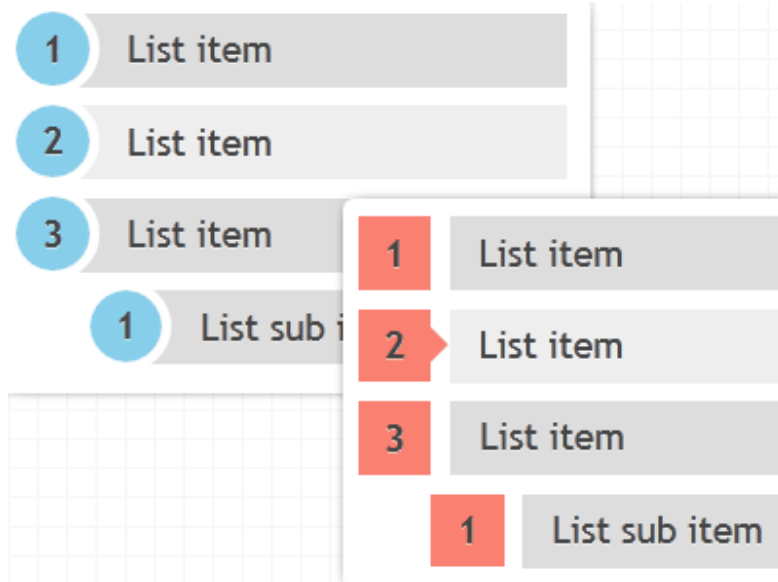


Благодарение на CSS можете:

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



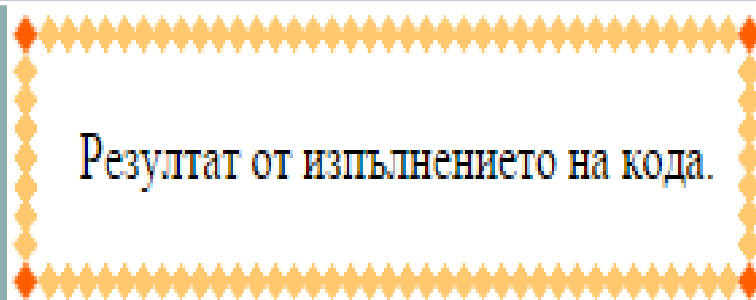
Рамки

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;  
}
```



Background

25

В CSS3 при работа с фонове има много новости:

- поддръжка на няколко фонови изображения
- възможност за задаване размер на изображението
- възможност за задаване на област на фона

Това се постига посредством следните свойства:

Property	Description
<u>background</u>	A shorthand property for setting all the background properties in one declaration
<u>background-clip</u>	Specifies the painting area of the background
<u>background-image</u>	Specifies one or more background images for an element
<u>background-origin</u>	Specifies where the background image(s) is/are positioned
<u>background-size</u>	Specifies the size of the background image(s)



Background

26

background-clip

- задава докъде се простира фона
- **ВЪЗМОЖНИ СТОЙНОСТИ:**
 - border-box - до края на рамката
 - padding-box - до края на padding-a
 - content-box - фонът се простира само в областта на съдържанието
- по подразбиране: border-box

```
.background_clip { /* Пример */  
  border: 10px dotted black;  
  padding: 15px;  
  background: yellow;  
  background-clip: content-box;  
}
```

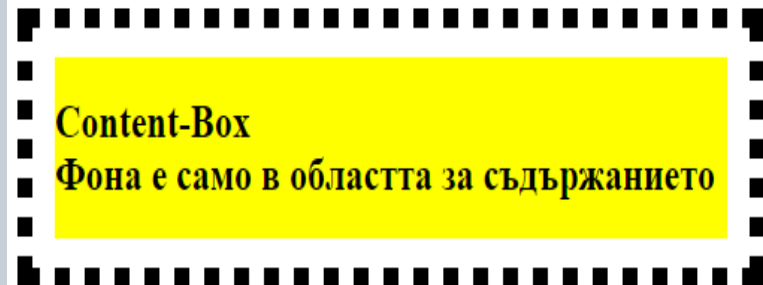
No background-clip (border-box is default):



background-clip: padding-box:



background-clip: content-box:



Background

27

- background-origin

- задава къде се позиционира фоновото изображение
- аналогично с **background-clip**
- възможни стойности: padding-box|border-box|content-box
- по подразбиране: padding-box
- игнорира се, ако **background-attachment** е зададено на fixed



Background

28

- **background-size**

- задава размера на фоновото изображение
- възможни стойности: `auto|length|cover|%|contain`
- възможно е да се задава повече от 1 изображение, разделени със ','
- стойностите за ширина и височина се разделят с празен интервал

Value	Description
<code>auto</code>	Default value. The background-image contains its width and height
<code>length</code>	Sets the width and height of the background image. The first value sets the width, the second value sets the height. If only one value is given, the second is set to "auto"
<code>percentage</code>	Sets the width and height of the background image in percent of the parent element. The first value sets the width, the second value sets the height. If only one value is given, the second is set to "auto"
<code>cover</code>	Scale the background image to be as large as possible so that the background area is completely covered by the background image. Some parts of the background image may not be in view within the background positioning area
<code>contain</code>	Scale the image to the largest size such that both its width and its height can fit inside the content area

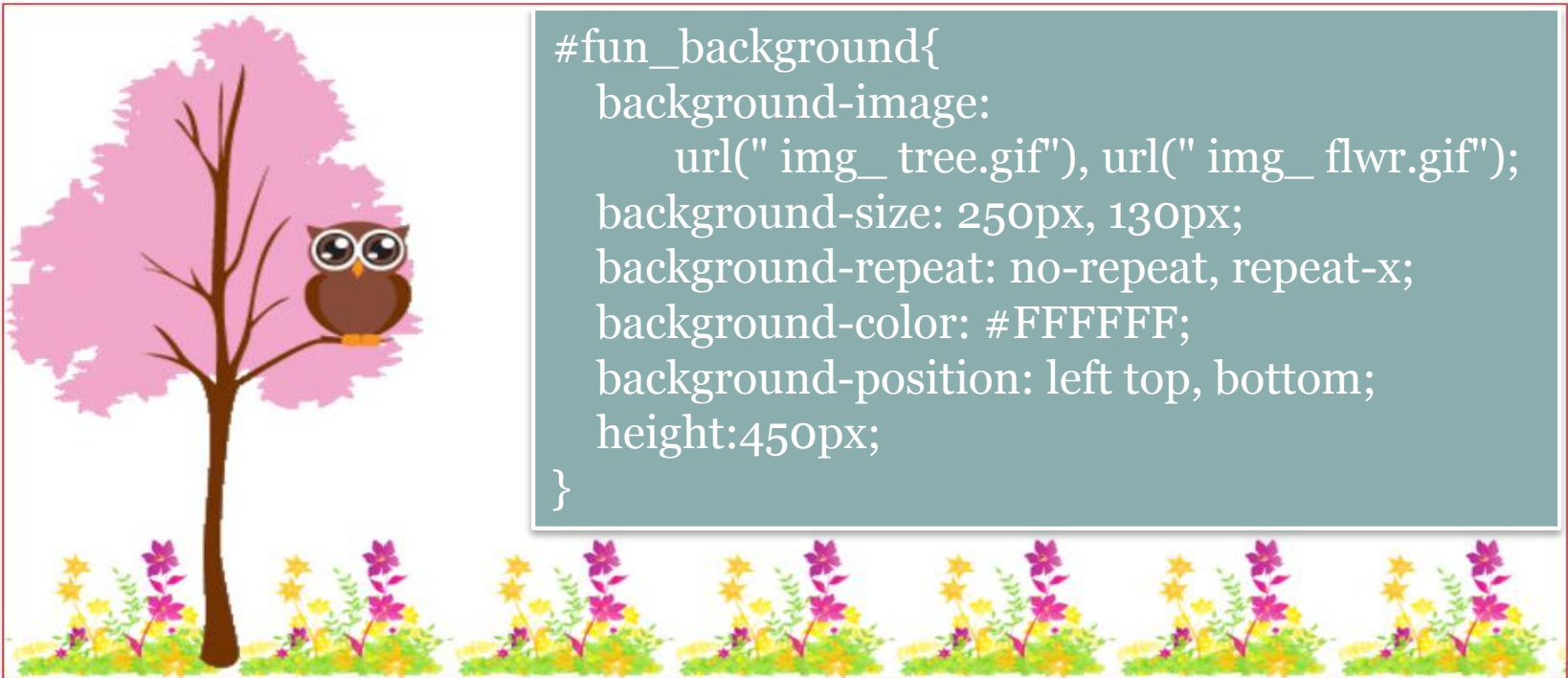


Background

29

- **background-image**

- задава пътя до изображението, което да се използва като фон
- възможни стойности: `url('URL')` | `none`
- възможно е да се задава повече от 1 изображение, разделени със ','



Background

30

Задачи за самоподготовка:

- добавете второ фоново изображение
- променете размера на фоновото изображение
- променете размера на фоновото изображение, по такъв начин, че да покрива цялата страница
- задайте фоновото изображение да започва от левия горен ъгъл на областта на съдържанието
- променете фона, за да бъде до вътрешния отстъп

Текст



- Свойства за работа с текст

Property	Description
<u>text-align-last</u>	Specifies how to align the last line of a text
<u>text-emphasis</u>	A shorthand for setting text-emphasis-style and text-emphasis-color in one declaration
<u>text-justify</u>	Specifies how justified text should be aligned and spaced
<u>text-overflow</u>	Specifies how overflowed content that is not displayed should be signaled to the user
<u>word-break</u>	Specifies line breaking rules for non-CJK scripts
<u>word-wrap</u>	Allows long words to be able to be broken and wrap onto the next line

Сенки



- **text-shadow**

- задава сянка около текста
- възможни стойности:
h-shadow v-shadow blur-radius color|none|initial|inherit;
- ако е зададен цвят, h-shadow и v-shadow са задължителни
- възможно е добавянето на 2 сенки едновременно - това става със ','

Horizontal Offset Hex Color

text-shadow: 1px 1px 2px #000;

Vertical Offset Blur Radius

```
h1
{
  text-shadow: 0 0 3px #FF0000, 0 0 5px #0000FF;
}
```

```
h1 {
  color: white;
  text-shadow: 1px 1px 2px black, 0 0 25px blue, 0 0 5px darkblue;
}
```

Примери:

Йордан Енев

WebAcademy



Web Fonts

With CSS3, web designers are no longer forced to use only "web-safe" fonts.

Уеб шрифтовете (Web Fonts) позволяват зареждане на шрифтове през от уеб, т.е. дизайнера не е ограничен само до шрифтове инсталирани на потребителския компютър.

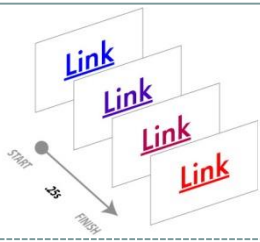
- задават се чрез @font-face декларираща клауза
- съществуват различни файлови формати
 - TrueType Fonts (TTF)
 - OpenType Fonts (OTF)
 - Web Open Font Format (WOFF)
 - Embedded OpenType Fonts (EOT)

```
@font-face { /* Декларираме */  
    font-family: myFirstFont;  
    src: url(sansation_light.woff);  
}  
/* Използваме зададеното име */  
div { font-family: myFirstFont; }
```

Съвет: URL на шрифта задавайте с малки букви, понеже главните букви могат да дадат неочаквани резултати в IE



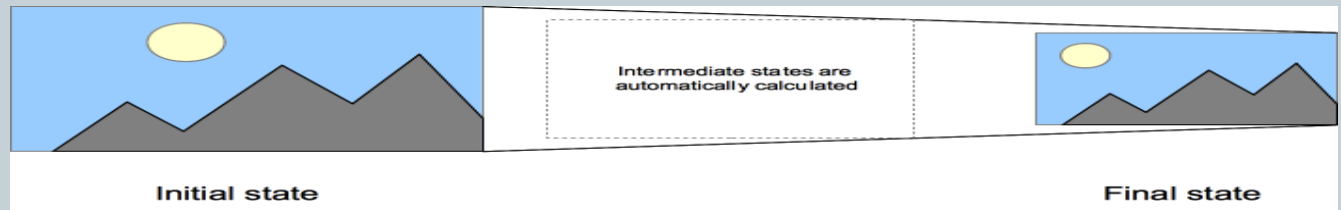
Descriptor	Values	Description
font-family	<i>name</i>	Required. Defines a name for the font
src	<i>URL</i>	Required. Defines the URL of the font file
font-stretch	normal condensed ultra-condensed extra-condensed semi-condensed expanded semi-expanded extra-expanded ultra-expanded	Optional. Defines how the font should be stretched. Default is "normal" <div> <h1>Web Fonts</h1> <p><i>With CSS3, web designers are no longer forced to use only "web-safe" fonts.</i></p> </div>
font-style	normal italic oblique	Optional. Defines how the font should be styled. Default is "normal"
font-weight	normal bold 100 200 300 400 500 600 700 800 900	Optional. Defines the boldness of the font. Default is "normal" <div> <p>При използването на @font-face шрифта може да бъде детайлно описан посредством показаните тук, т.нар. font descriptors</p> </div>
unicode-range	<i>unicode-range</i>	Optional. Defines the range of UNICODE characters the font supports.



Transitions

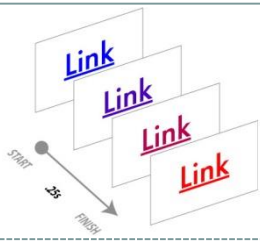
35

Преходите (Transitions) позволяват да се промени плавно стойността на дадено свойство, за даден период от време.



Запомнете:

- нужно е да се зададе свойството, което ще се променя и продължителността от време, за което това да се случи
- ако продължителността не е задена, преход няма да има, понеже продължителността по подразбиране е 0 (нула)
- могат да се задават по няколко свойства, разделени със ','
- може да задава забавяне чрез свойството `transition-delay`
- преходите могат да се комбинират с трансформациите



Transitions

36

- Списък със свойства за задаване на преходи:

Property	Description
<u>transition</u>	A shorthand property for setting the four transition properties into a single property
<u>transition-delay</u>	Specifies a delay (in seconds) for the transition effect
<u>transition-duration</u>	Specifies how many seconds or milliseconds a transition effect takes to complete
<u>transition-property</u>	Specifies the name of the CSS property the transition effect is for
<u>transition-timing-function</u>	Specifies the speed curve of the transition effect

```
div {  
  Пример 1: Подробно задаване  
  transition-property: width;  
  transition-duration: 2s;  
  transition-timing-function: linear;  
  transition-delay: 1s;  
}
```

```
div {  
  Пример 2: Кратко задаване  
  transition: width 2s linear 1s;  
  /* property duration function delay */  
}  
  
/* Пример 1 и Пример2 са еднакви */
```



Media Queries

37

Media Queries позволяват да имаме различни CSS дефиниции за различните устройства и резолюции

- Могат да се използват за:
 - проверка широчина и височина на видимата област на браузъра
 - проверка широчина и височина на устройството
 - ориентация (landscape/portrait)- при таблети / телефони
 - проверка на резолюцията
- Синтаксис:

/ Вътрешен синтаксис използван при <style> таг или вътре в CSS файл */*

```
@media not|only mediatype and (expressions) {  
    CSS-Code;  
}
```

/ Синтаксис за зареждане на външни CSS файлове при дадено условие */*

```
<link rel="stylesheet" media="mediatype and|not|only (expressions)" href="f.css">
```



Media Queries

38

- Възможни стойности за mediatype:

Value	Description
all	Used for all media type devices
print	Used for printers
screen	Used for computer screens, tablets, smart-phones etc.
speech	Used for screenreaders that "reads" the page out loud

- Пример:

```
@media screen and (min-width: 480px) {  
  body {  
    background-color: lightgreen;  
  }  
}
```

- Повече информация: [тук](#)



Има ли още?

39

Разбира се! Светът на CSS е наистина необятен!

Вижте още:

- [FlexBox](#) и [Box Sizing](#)
- [CSS 2D Трансформации](#)
- [CSS 3D Трансформации](#)
- [създаване на странициране](#)
- [създаване на красиви бутони](#)
- [преоразмеряване от потребителя](#)
- [примери върху Media Queries](#)



Проблем?! реши го сам!



Тази страница, ще ви научи, как да решите често срещани CSS проблеми. Нека Силата бъде с Вас!

- Не ми приема дадено свойство (дефиниция) ?
- Как да центрирам елемент без <center> таг ?
- Как да подравня елемент в дясно? `text-align` не помага :(
- Как да стилизирам бутон за прикачване на файл?
- Използвам `float` и съдържанието излиза извън елемента

Не ми приема дадено свойство?

41

Наръчник за оцеляване в CSS свят:

- проверете дали няма синтактични грешки може да има липсващи ":" , ";" незатворени низове и др.
- проверете дали въпросното свойство, е изписано правилно
- низовете трябва да бъдат оградени правилно в "кавички" или 'апострофи' (виж задаване на фон и работа с шрифт)
- при задаване на стойност в мерни единици, между цифрата и мерната единица не трябва да има разстояние.
- проверете дали дадена буква, не е изписана на кирилица, вместо на латиница.

```
.class{                                /* Коректен код */  
    font-size:16px;  
    font-family:'Open Sans';  
    background-image: url("path/to/image.jpg");  
}
```



Как да центрирам елемент?

42

В CSS `<center>` тага няма готов еквивалент.

За да създадем наш вариант трябва да помним, че:

- за да центрираме елемент е нужно да зададем стойност `auto` на свойствата `margin-left` и `margin-right`
- за да центрираме текста на даден елемент трябва да използваме свойството `text-align` със стойност: `center`
- елемента, който искаме да центрираме трябва да бъде `block` елемент

Имайки в предвид горното, се нуждаем от следния код:

```
.center { margin-left: auto; margin-right: auto; text-align: center; display: block; }
```



Как да подравня елемент в дясно?

43

Позициониране на елемент в дясно става чрез:

- задаване на позиция на елемента (показания пример)
- свойството `float` (виж ТОЗИ евентуален проблем)
- други подобни техники

Не забравяйте, че:

- `text-align` подравнява текста на даден елемент
- `float` подравнява самият елемент

```
<div>
  Normal Text <span style="position: absolute; right:5px;">Absolute Text</span>
</div>
<div>
  <span style="float:left;">Left Float Text</span>
  <span style="float:right;">Right Float Text</span>
</div>
```



Стилизиране на file upload бутон

44

Начина, по който се визуализира file upload бутона зависи от браузъра.

Можете да го "промените" по няколко начина:

- чрез <label> таг (виж долупосочения пример)
- чрез прозрачен file upload бутон
- други подобни техники

/* CSS Code */

```
input[type=file] { display:none; }  
.fake_btn{  
  border: 2px solid black;  
  background-color:silver;  
  font-style:italic;  
  padding:3px 6px;  
}
```

<!-- HTML Code -->

```
<label>  
  <!-- оригиналия бутон е скрит -->  
  <input type="file" name="u_file" />  
  <span class="fake_btn">Upload</span>  
  <!-- този бутон е направен по наш вкус,  
  а дефинициите в ляво, са примерни -->  
</label>
```



Проблем с float?!

45

- Ако изображението е по-високо от елемента, който го съдържа, то ще излезе извън него. Решение?

/* CSS Code */

```
div {  
  border: 3px solid #73AD21;  
}
```

```
.img1 {  
  float: right;  
  width: 100px;  
  height: 140px;  
}
```

```
.clearfix {  
  overflow: auto;  
}
```

<!-- HTML Code -->

```
<div>  
    
  Картинката излиза извън елемента :(  
</div>
```

```
<p style="clear:right">  
  Без clear, подравняването се омазва :(  
</p>
```

```
<div class="clearfix">  
    
  Вижте clearfix. Той реши проблема! :)  
</div>
```



ВЪПРОСИ

46



Използвана литература

47

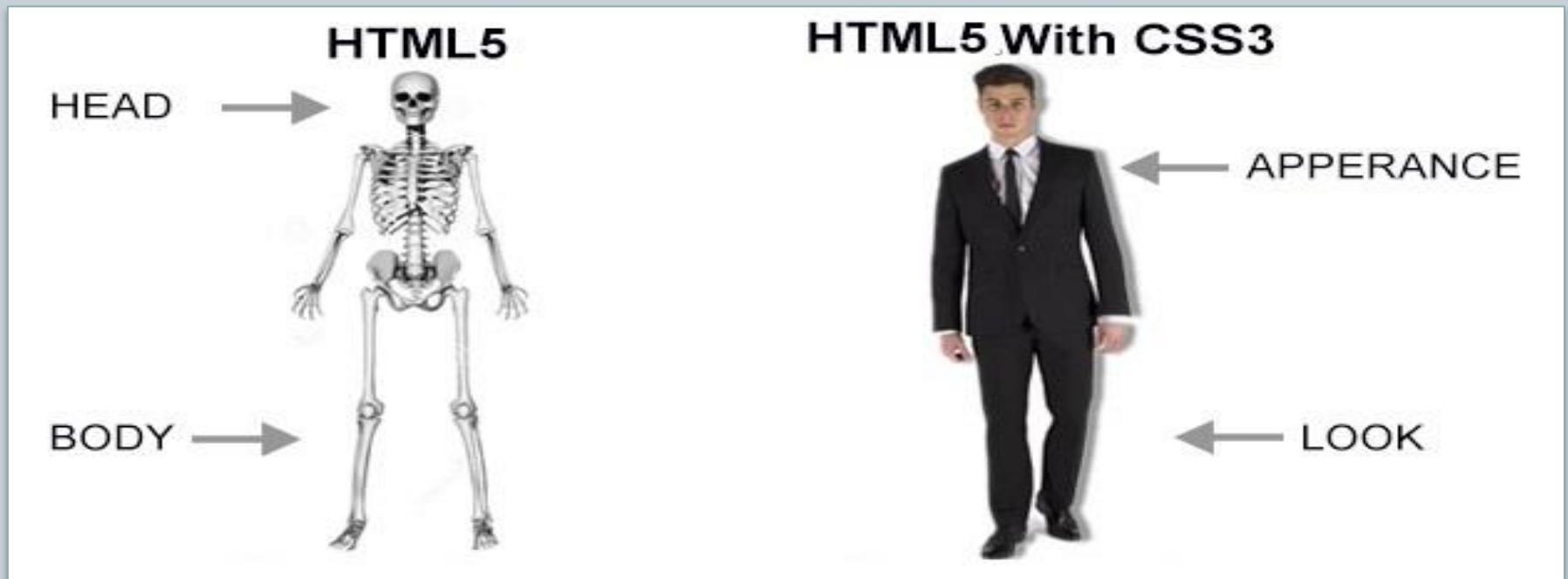
- <http://www.htmldog.com/guides/css/beginner/selectors/>
- http://www.w3schools.com/css/css_syntax.asp
- <https://bg.wikipedia.org/wiki/CSS>
- <https://css-tricks.com/specifcs-on-css-specificity/>
- <http://www.csstextwrap.com/>



Поздравления!

48

- Този модул приключи!
- Желаем Ви късмет със следващия модул!
- Вече знаете, какво е HTML5 & CSS3!



Това може да ми потрѣбва



HTML & CSS



```
a.foo {  
  padding: 5px 10px;  
  background: #69f;  
  color: #000;  
  -webkit-transition-pr  
  -webkit-transition-du  
  -webkit-transition-tj  
}
```

```
a.foo:hover {  
  background: #33f;  
  color: #fff;  
}
```

```
a.foo {  
  padding: 5px 10px;  
  background: #69f;  
  color: #000;  
  -webkit-transition-t  
}
```

CSS3

Transit



