



**WEB ACADEMY**

# Front-end за Начинаещи



# За CSS

3

- CSS – съкратено от *Cascading Style Sheets*
- CSS е език за описание на стилове
- Създаден с цел да бъдат разделени съдържанието и структурата на уеб страниците отделно от тяхното визуално представяне
- Най-често се използва за описване представянето на документ написан на HTML или друг XML документ.
- CSS поддържа многоредови коментари във формат  
/\* Това е коментар, който няма да се интерпретира \*/



# 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. Стиловете на ниво елемент препокриват предишните дефиниции

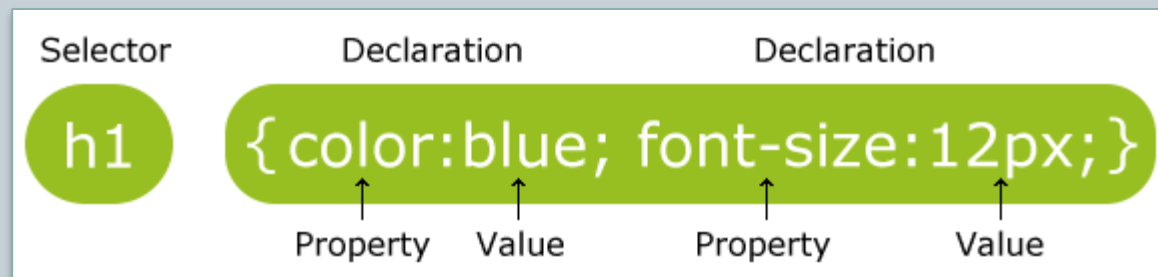


# Синтаксис



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 ":"



# 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!**



# 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...



- **Вижте още:**

- [информация за цветовете и примери](#)
- [списък с всички цветове групирани по нюанс](#)
- [списък с имена на цветове поддържани от всички браузъри](#)
- [списък с 216 "безопасни" цветове](#)



# 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>
```



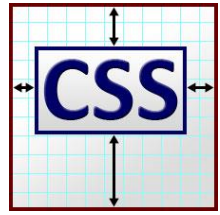
Това е така, защото:

- 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!"
- Стилосвете на ниво елемент препокриват предишните дефиниции. Те винаги печелят!

Повече информация: [тук](#)

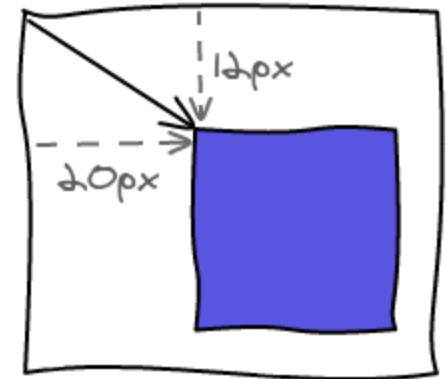


# Позициониране на елементи



CSS свойства за задаване позицията на елементите:

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



# Позициониране на елементи

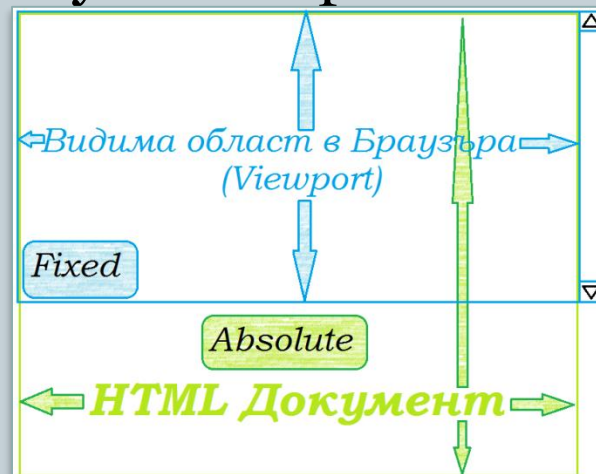


В CSS се използват следните видове позиции:

- static - стойност по подразбиране.
- relative - изчислява се спрямо нормалната си позиция
- absolute - изчислява се според размера на целия документ
- fixed - изчислява се според видимата област на браузъра, което означава, че местоположението му не се променя дори и при скролване на страницата

Опитайте сами:

```
#f_btn{position:fixed; bottom:11px; left:11px;}  
#a_btn{position:absolute; bottom:11px; left:51px;}  
...  
<button id="f_btn">Fixed</button>  
<button id="a_btn">Absolute</button>
```

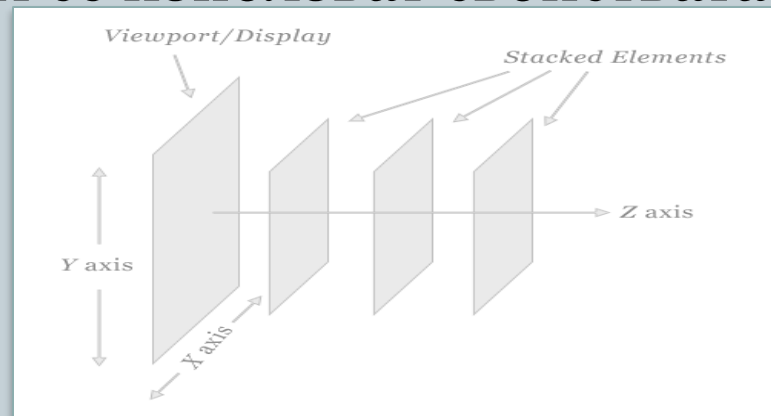


# Позициониране на елементи

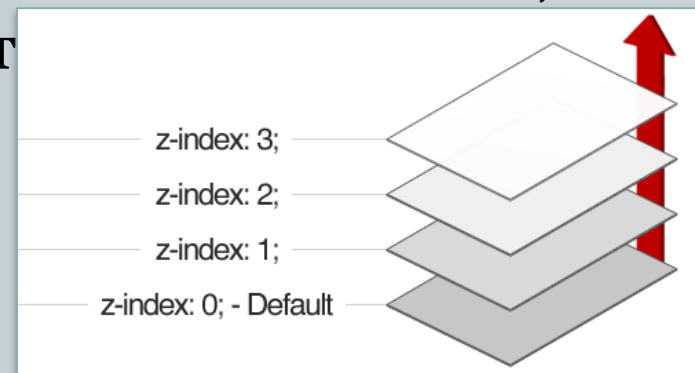


За задаване на конкретна позиция се използват свойствата:

- top - задава позиция отгоре
- right - задава позиция отдясно
- bottom - задава позиция отдолу
- left - задава позиция отляво
- z-index - указва дали даден елемент се намира над/под друг елемент. В случай на "застъпване" на елементи, отгоре е този с по-голяма стойност



Статичните елементи, не се влияят от свойствата: Top, Right, Bottom, Left



# 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



# Работа с Фон (Background)



- **background-color**
  - задава фонов цвят, чрез име на цвета или стойност.
  - възможни стойности, които приема: color|transparent|initial|inherit
  - повече информация за цветовете вижте в [слайда за цветовете](#)
- **background-image**
  - задава пътя до изображението, което да се използва като фон
  - възможни стойности: url('URL') | none
- **background-repeat**
  - указва дали/как да се повтаря изображението
  - възможни стойности: repeat|repeat-x|repeat-y|norepeat|initial|inherit



# Работа с Фон (Background)



- **background-attachment**

- указва дали изображението е фиксирано или се скролва с текста
- възможни стойности: `scroll|fixed|local|initial|inherit`

- **background-position**

- указва позицията на изображението
- по подразбиране изображението е горе вляво, и се повтаря по X и Y
- възможни стойности:

Value	Description
left top left center left bottom right top right center right bottom center top center center center bottom	If you only specify one keyword, the other value will be "center"
<code>x% y%</code>	The first value is the horizontal position and the second value is the vertical. The top left corner is 0% 0%. The right bottom corner is 100% 100%. If you only specify one value, the other value will be 50%. . Default value is: 0% 0%
<code>xpos ypos</code>	The first value is the horizontal position and the second value is the vertical. The top left corner is 0 0. Units can be pixels (0px 0px) or any other <u>CSS units</u> . If you only specify one value, the other value will be 50%. You can mix % and positions



# Работа с Фон (Background)

16

## Опитайте сами:

```
/* Пример за фиксиран background */  
body {  
    background-color: white;  
    background-image: url("img_tree.png");  
    background-repeat: no-repeat;  
    background-position: right top;  
    background-attachment: fixed;  
}
```

/\* Съкратен формат на горното.  
Двата примера са напълно еквивалентни \*/

```
body {  
    background: #ffffff url("img_tree.png") no-repeat fixed right top;  
}
```

Result:

## Hello World!

The background-image is fixed. Try to scroll down the page.

The background-image is fixed. Try to scroll down the page.

The background-image is fixed. Try to scroll down the page.

The background-image is fixed. Try to scroll down the page.





# 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!

# Форматиране на текст



- **unicode-bidi**

- указва дали в документа ще се добавят различни езици
- възможни стойности, които приема: `normal|embed|bidi-override`
- използва се съвместно с **direction**

- **direction**

- указва посоката на писане - `ltr` (Left to Right) или `rtl` (Right to Left)
- използва се съвместно с `unicode-bidi`

- **text-align**

- задава хоризонтално подравняване на елемента
- възможни стойности: `left|right|center|justify|initial|inherit;`

- **vertical-align**

- задава вертикално подравняване на елемента
- често използвани стойности:  
`baseline|length|sub|super|top|middle|bottom|initial|inherit` и други



# Форматиране на текст



- **color**
  - задава цвят на текста, чрез име на цвета или стойност.
  - възможни стойности, които приема: color|initial|inherit
  - повече информация за цветовете вижте в слайда за цветовете
- **text-decoration**
  - задава подчертаване на текста.
  - възможни стойности: none|underline|overline|line-through
- **text-transform**
  - задава трансформиране на текста в главни или малки букви
  - възможни стойности: none|capitalize|uppercase|lowercase
- **text-indent**
  - задава отстъпа на първия ред на даден текстов блок
  - възможни стойности: % от width на родителския елемент, или цифрова стойност зададена чрез мерни единици



# Форматиране на текст



- **word-spacing**
  - задава разстояние между думите
  - възможни стойности: normal (стойност по подразбиране) или цифрова стойност зададена чрез мерни единици.
  - отрицателни стойности също са разрешени
- **letter-spacing**
  - задава разстояние между буквите
  - възможни стойности: normal (стойност по подразбиране) или цифрова стойност зададена чрез мерни единици.
  - отрицателни стойности също са разрешени.

Примери:

```
.arab_text { direction: rtl; unicode-bidi: bidi-override; }  
p { word-spacing: 16px; letter-spacing: normal; text-indent: 29px; }  
span { text-transform: uppercase; text-decoration: none; color: blue; }
```



# Форматиране на текст



- **line-height**

- задава междуредово разстояние
- възможни стойности: `normal|number|length|initial|inherit`
- подробна информация за стойностите в таблицата по-долу.

Value	Description
<code>normal</code>	A normal line height. This is default
<code>number</code>	A number that will be multiplied with the current font size to set the line height
<code>length</code>	A fixed line height in px, pt, cm, etc.
<code>%</code>	A line height in percent of the current font size
<code>initial</code>	Sets this property to its default value. <a href="#">Read about <i>initial</i></a>
<code>inherit</code>	Inherits this property from its parent element. <a href="#">Read about <i>inherit</i></a>

# Форматиране на текст



- **white-space**

- указва как да се обработва празното пространство
- възможни стойности:  
normal|nowrap|pre|pre-line|pre-wrap|initial|inherit
- подробна информация за стойностите в таблицата по-долу.

Value	Description
normal	Sequences of whitespace will collapse into a single whitespace. Text will wrap when necessary. This is default
nowrap	Sequences of whitespace will collapse into a single whitespace. Text will never wrap to the next line. The text continues on the same line until a <code>&lt;br&gt;</code> tag is encountered
pre	Whitespace is preserved by the browser. Text will only wrap on line breaks. Acts like the <code>&lt;pre&gt;</code> tag in HTML
pre-line	Sequences of whitespace will collapse into a single whitespace. Text will wrap when necessary, and on line breaks
pre-wrap	Whitespace is preserved by the browser. Text will wrap when necessary, and on line breaks
initial	Sets this property to its default value. <a href="#">Read about <i>initial</i></a>
inherit	Inherits this property from its parent element. <a href="#">Read about <i>inherit</i></a>



# Форматиране на текст



- **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 /* Бял текст с черна сянка */
```

```
{  
  color: white; text-shadow: 2px 2px 4px #000000;  
}
```

Примери:

```
h1 /* Пример за 2 сенки - червена и синя */
```

```
{  
  text-shadow: 0 0 3px #ff0000, 3 3 9px #0000ff;  
}
```

# Форматиране на текст

24

Примери за работа с текст:

```
{  
  direction: rtl;  
  unicode-bidi: bidi-override;  
  word-spacing: 16px;  
  letter-spacing: normal;  
  text-indent: 29px;  
  text-transform: lowercase;  
  text-decoration: none;  
  color: #d7e7ff;  
  text-shadow: 2px 2px 4px #000000;  
  text-align:center;  
  vertical-align:middle;  
}
```





# 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**



# Работа с шрифт



- **font-family**

- задава името на шрифта, който искаме да използваме
- може да се задава повече от един шрифт, разделени с запетая ",". Ако браузъра, не поддържа първия, ще продължи с втория и т.н.
- ако името на шрифта съдържа празен интервал, цялото име трябва да се зададе с кавички (или апострофи ако сме в style атрибут).

Пример:

```
p { font-family: "Times New Roman", Georgia, Serif; }  
/*
```

~ За Ваша Информация ~

Шрифтовете са разделени в 3 групи по външността им:

"Serif" - те имат малки линии в краищата. (Times New Roman, Georgia и др.)

"Sans-serif" - те нямат линии в краищата. (Arial, Verdana и други)

"Monospace" - имат еднаква ширина. (Courier New, Lucida Console и други)

\*/

Sans-serif

Serif

днес

слушам сам въщи  
и пиша код

slushash?

1:19

да

1:20

kakvo

1:20

слушам го

1:20

пуснал съм си го и го слушам  
докато си пиша с теб

1:21

ne moga da razbera kak taka si pusnal da slushash  
i kakvo go

1:22

<http://vbox7.com/play:eeb118a3?p=playlist&id=1577664&order=date&sfx=87>

1:22

hahahahahahaha  
sega razbiram kolko sa vajni  
'Sam v kashti'

1:22

omg

1:23

parvite 7 pati  
cheta i vijdam smisala  
che nqma nikoi v staqta ti vse edno

1:23

Затова са важни "кавички" и 'апострофи'!

# Работа с шрифт



- **font-size**

- задава размера на шрифта
- стойност може да се задава по различни начини. Вижте таблицата:

Value	Description
medium	Sets the font-size to a medium size. This is default
xx-small	Sets the font-size to an xx-small size
x-small	Sets the font-size to an extra small size
small	Sets the font-size to a small size
large	Sets the font-size to a large size
x-large	Sets the font-size to an extra large size
xx-large	Sets the font-size to an xx-large size
smaller	Sets the font-size to a smaller size than the parent element
larger	Sets the font-size to a larger size than the parent element
<i>length</i>	Sets the font-size to a fixed size in px, cm, etc.
%	Sets the font-size to a percent of the parent element's font size



# Работа с шрифт



- **font-variant**

- указва дали малките букви да се изписват с малки или главни букви
- възможни стойности: `normal|small-caps|initial|inherit`
- да НЕ се бърка с **text-transform** свойството. Има съществена разлика!

- **font-style**

- задава дали шрифта да е наклонен или не
- възможни стойности: `normal|italic|oblique|initial|inherit`



# Работа с шрифт



- **font-weight**

- задава дебелината на шрифта
- възможни стойности: `normal|bold|bolder|lighter|number|initial|inherit`
- подробна информация в таблицата по долу:

Value	Description
normal	Defines normal characters. This is default
bold	Defines thick characters
bolder	Defines thicker characters
lighter	Defines lighter characters
100	Defines from thin to thick characters. 400 is the same as normal, and 700 is the same as bold
200	
300	
400	
500	
600	
700	
800	
900	

Нека помислим:

```
.c1 { font-weight: 400; }  
.c2 { font-weight: normal; }  
.c3 { font-weight: 700; }  
.c4 { font-weight: bold; }
```

c1 и c2 са еквивалентни.  
c3 и c4 са еквивалентни също.

Ами bolder и bold равни ли са?  
Ами semi-bold = ?  
Ами lighter = ?



# Работа с шрифт



- **font**
  - позволява задаване на всички свойства на шрифта с 1 декларация
  - ред на задаване:  
font-style font-variant font-weight font-size/line-height font-family

## Например:

```
p.ex1 {font: italic bold 12px/30px Georgia, serif; }
```

/\* Горния ред е кратък запис на долните няколко реда. Еднакви са! \*/

```
p.ex1
```

```
{
```

```
  font-style:italic;
```

```
  font-weight:bold;
```

```
  font-size:12px;
```

```
  line-height:30px;
```

```
  font-family: Georgia, serif;
```

```
}
```



# 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
```





# Списъци

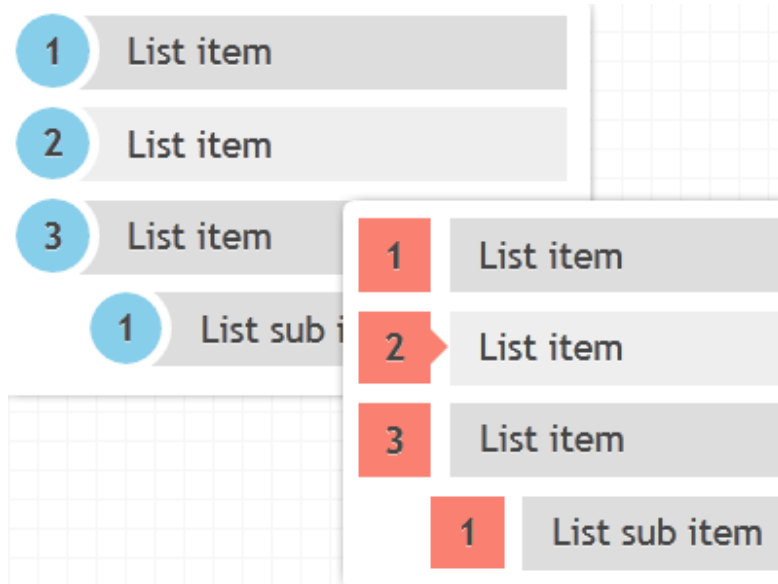


Благодарение на CSS можете:

- 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



# Списъци



- **list-style-image**
  - указва изображението, което да се ползва като маркер
  - възможни стойности: none|url|initial|inherit
  - използвайте го в комбинация с **list-style-type**. Дефиницията от него, ще се ползва в случай, че изображението не може да се зареди.
- **list-style-type**
  - задава типа на използвания маркер
  - често използвани стойности: armenian|upper-roman|decimal и други
  - ПЪЛЕН СПИСЪК С ВСИЧКИ ВЪЗМОЖНИ СТОЙНОСТИ
- **list-style-position**
  - указва дали маркерите трябва да бъдат вътре в съдържанието
  - възможни стойности: inside|outside|initial|inherit
  - по подразбиране, маркерите са извън съдържанието (outside)

# Списъци



- **list-style**

- кратка форма за задаване на всички свойства с една декларация
- ред на задаване: list-style-type, list-style-position, list-style-image
- ако дадено свойство липсва, ще се използва неговата default стойност

```
ul { list-style: square url("sqpurple.gif"); }
```

Примери:

```
/* Горния код е еквивалентен на долния! Защо?! */
```

```
ul  
{  
  list-style-type: square;  
  list-style-position: outside;  
  list-style-image: url(' sqpurple.gif ');  
}
```



# Таблицы



- **table-layout**

- задава алгоритъма, който се използва за изчертаване на таблицата
- възможните стойности: auto|fixed
- auto - широчината на колоната зависи от най-дългото непрекъснато съдържание в колоната. Възможно е да се визуализира по-бавно.
- fixed - широчината зависи изцяло от дефинирания размер за таблица и колони, и не зависи от съдържанието. Визуализира се по-бързо. Ако не е зададен размер за таблица/колони тогава размера зависи от съдържанието в първия ред на таблицата.



Пример:

```
table
{
  table-layout: fixed;
}
```



# Таблици



- **empty-cells**

- указва дали да се показват празните клетки в таблицата
- възможни стойности: show|hide|initial|inherit
- игнорира се при **border-collapse: collapse**

- **border-spacing**

- задава разстоянието между клетките подобно на **cellspacing** атрибута
- възможни стойности: length|initial|inherit
- игнорира се при **border-collapse: collapse**

- **border-collapse**

- указва дали рамките се сливат или са отделни
- възможни стойности: separate|collapse|initial|inherit
- при стойност **collapse** игнорира **border-spacing** и **empty-cells**



# Таблицы



- **caption-side**

- указва къде да се показва заглавието, дефинирано чрез <caption> таг
- възможни стойности: top|bottom|initial|inherit

- **border**

- задава рамка около таблицата
- повече информация в следващата секция "Рамки"

```
table
{
  border-collapse: separate;
  border-spacing: 11px 55px;
  empty-cells: hide;
  caption-side: bottom;
  border: 5px ridge red;
}
```

Примери:



# "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
```

# Рамки



- **Основни свойства за задаване на рамки**
  - border - кратък формат за задаване на всички свойства с 1 декларация
  - border-color - задава цвят на цялата рамка
  - border-style - задава стил на цялата рамка
  - border-width - задава дебелина на цялата рамка
- **Детайлни (псевдонимни) свойства:**
  - border-top, border-right, border-bottom, border-left - подобно на border, задава всички свойства, но само за конкретна част от рамката
  - други още по детайлни свойства:  
border-top-color , border-top-style, border-top-width,  
border-right-color , border-right-style, border-right-width,  
border-bottom-color , border-bottom-style, border-bottom-width,  
border-left-color , border-left-style, border-left-width,



# Рамки



- **border-color**
  - задава цвят на рамката
  - възможни стойности: color|transparent|initial|inherit
- **border-style**
  - задава стил на рамката
  - възможни стойности:  
none|hidden|dotted|dashed|solid|double|groove|ridge|inset|outset
- **border-width**
  - задава дебелина на рамката
  - възможни стойности: medium|thin|thick|length|initial|inherit
- **border** - задава всички свойства с 1 дефиниция
  - ред на задаване: border-width border-style border-color

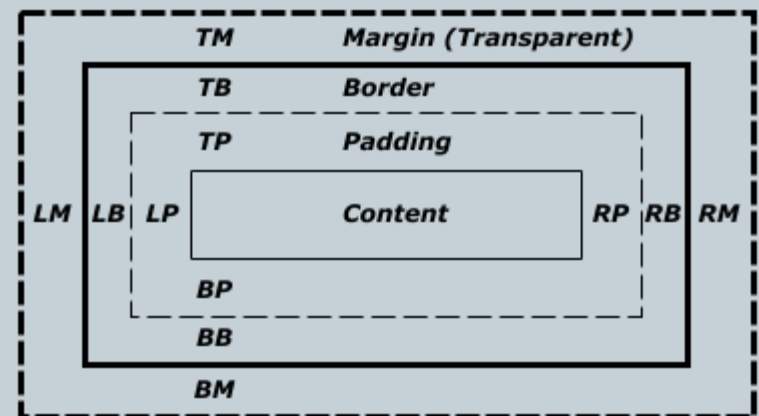
Например:      p { border: 5px ridge red; } /\* Двата реда са идентични \*/  
                  p { border-width:5px; border-style:ridge; border-color:red }



# Вътрешен отстъп



- Главно свойство за задаване на вътрешен отстъп
  - padding - кратък формат за задаване на всички свойства
  - задава отстъп между рамката на елемента и съдържанието му
  - може да приема от 1 до 4 стойности в мерни единици или проценти
- Детайлни свойства:
  - padding-top
  - padding-right
  - padding-bottom
  - padding-left



--- Margin edge  
— Border edge  
- - - Padding edge  
—— Content edge

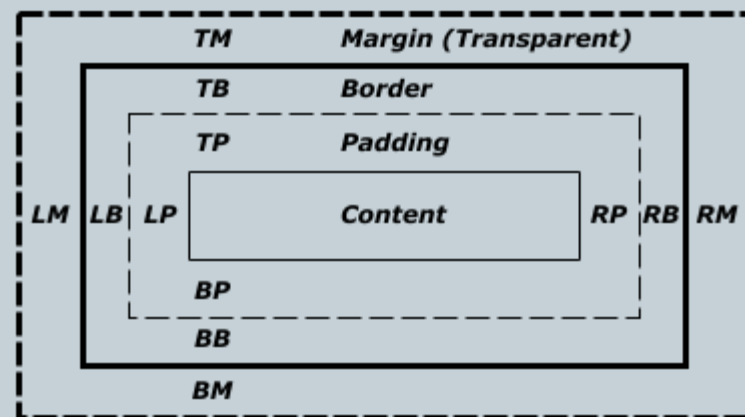
Примери:  
`p { padding: 5px; }`  
`p { padding: 5px 3px; }`  
`p { padding: 5px 4px 3px; }`  
`p { padding: 5px 4px 3px 2px; }`



# Външен отстъп



- Главно свойство за задаване на външен отстъп
  - margin - кратък формат за задаване на всички свойства
  - задава отстъп между рамката на елемента и съседните елементи
  - може да приема от 1 до 4 стойности в мерни единици или проценти
- Детайлни свойства:
  - margin-top
  - margin-right
  - margin-bottom
  - margin-left



--- Margin edge  
— Border edge  
- - - Padding edge  
—— Content edge

Примери:

```
p { margin: 5px; }  
p { margin: 5px 3px; }  
p { margin: 5px 4px 3px; }  
p { margin: 5px 4px 3px 2px; }
```



# Размери



- **Задаване на дължина и ширина**

- width - задава ширина в мерни единици или %
- height - задава височина в мерни единици или %

```
div {      Пример:  
width: 525px;  
height: 161px;  
max-height: 55px;  
max-width: 77px;  
}
```

- **Детайлни свойства:**

- min-width - задава минимална широчина в мерни единици или %
- min-height - задава минимална височина. Аналогично с min-width
- max-width - задава максимална широчина в мерни единици или %
- max-height - задава максимална височина. Аналогично с max-width

- **Запомнете че:**

- тези свойства, не работят с inline елементи.
- max-width препокрива дефиницията на width
- max-height препокрива дефиницията на height

# Видимост



- **display**

- указва как да се визуализира елемента
- често използвани стойности: inline|block|inline-block|none
- при зададена стойност **none**, не се заделя място за елемента

- **visibility**

- указва дали елемента е видим или не.
- заделя се място за елемента, независимо дали е видим или не
- възможни стойности: visible|hidden|collapse|initial|inherit

```
.ninja{  
  color: black;  
  visibility: hidden;  
}
```

```
h1.hidden {  
  display: none;  
}  
/* ~~ Еднакви ли са? ~~ */  
h1.hidden {  
  visibility: hidden;  
}
```



# Видимост



- **overflow**

- указва какво се случва със съдържанието, което не се побира
- възможни стойности: visible|hidden|scroll|auto|initial|inherit

- **Детайлни свойства:**

- overflow-x
- overflow-y

```
/* Пример за overflow */
```

```
div {  
  width: 150px;  
  height: 150px;  
  overflow: scroll;  
}
```

```
div.ex1 {  
  width: 500px;  
  margin: auto;  
  border: 3px solid #73AD21;  
}
```

```
div.ex2 {  
  max-width: 500px;  
  margin: auto;  
  border: 3px solid #73AD21;  
}
```

# Други свойства

47

- **quotes**

- задава елементите използвани за цитат

- **cursor**

- задава използвания курсор
- често използвани стойности: alias|help|move|pointer|wait
- СПИСЪК С ВСИЧКИ СТОЙНОСТИ

- **float**

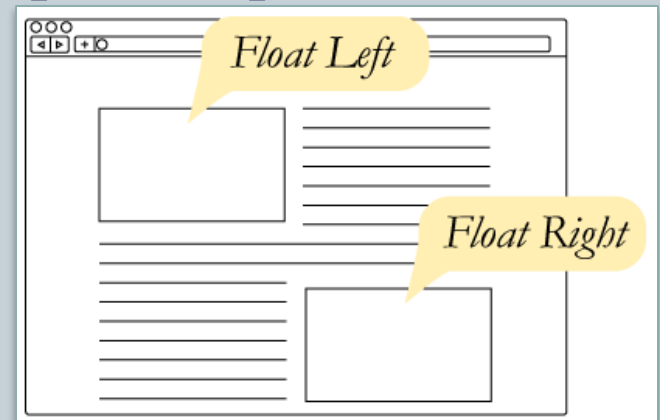
- задава подравняване на **елемента**
- възможни стойности: none|left|right

- **clear**

- задава отместване спрямо съседните елементи
- използва се съвместно с **float**

Пример:

```
q{  
  quotes: "«" "»";  
  cursor: alias;  
}
```



# 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?

49

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?

50

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?

51

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

52

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?!

53

- 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

54



# Използвана литература

55

- <http://www.htmldog.com/guides/css/beginner/selectors/>
- [http://www.w3schools.com/css/css\\_syntax.asp](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!

56



- Този модул приключи!
- Желаяем Ви късмет със следващия модул!
- Вече знаете, какво е HTML & CSS!

*HTML = How To Meet Ladies  
&  
CSS = Countless Sex Styles*