CSS Style Inheritance

**Inherited are:** Text-related and list-related properties

* color,
* font-size,
* font-family
* line-height
* text-align
* list-style, etc.

**Not inherited are** Box-related and positioning styles:

* width,
* height
* border
* margin
* padding
* position
* float, etc

**<a> elements do not inherit color and text-decoration**

Common Important TODO

**1. CSS file in HTML,< head>**

*<link rel="stylesheet" type="text/css" href="style/democss.css">*

2. С Ctrl+F5 се прави пълен refresh на страницата исе трие cash и се зарежда актуалният CSS file

**3. Reset Browswr settings**

*\* { margin: 0; padding: 0; }*

*body, h1, p, ul, li { margin: 0; padding: 0; }*

Ctrl+E+F ;$\*10 enum Shift+Alt+W - прави родител

**Removing bulets from ul unorderd list**

li {list-style-type: none;}

CSS Syntax

**Stylesheets** consist of rules, selectors, declarations, properties and values

* **Selectors** are separated by commas
* **Declarations** are separated by semicolons
* **Properties and values** are separated by colons

h1,h2,h3 {

color: green;

font-weight: bold;

}

***Primary kinds of selectors:***

* By tag (type selector):

h1 { font-family: verdana,sans-serif; }

* By element id:

#element\_id { color: #ff0000; }

* By element class name (only for HTML):

.myClass {border: 1px solid red**}**

Selectors can be combined with commas:

* All tags inside tag - nested

div p {text decoration: underline} all <p> inside <div>

Всички параграфи които се намират в дивове

child in child in.....parent div p ....a

* \*All tags inside tag

p \*{color:black} - all tags isida <p> не се използва

* + selector – used to match “next sibling”(братя и сестри):

img + .link {float:right}This will match all siblings with class name link that appear immediately after <img> tag

* > селектира директни наследници matches direct child nodes
* .class1.class2(no spase!) едновременно отговаря и на двете условия

<p class="class1 class2">

* ! important - cant overide better not to use

color: green !important;

* [ ] selects elements based on attributes

a[title] {color:black}

input[type=text] { font-family:Consolas

a[title\*=logo] {border: none}

* .class1 :not(p) селектира всички елементи с class="class1", които не са параграфи

Common Pseudo Selectors

* :hover - селектира елемент само когато мишката е отгоре

*a:hover { color: red; }*

* :visited -сменя цвета когато сле посетили даден линк
* :active - променя бутон когато сме го натиснали
* :lang - селектира елементите на какъв език са (lang=en)
* :first-line - селектира първата линия от даден текст
* :first-letter - селектира първата буква от даден текст
* :before - на текущия елемент преди това ми сложи някакъв content:

*a:before {content: "<<";}*

* :after - на текущия елемент след това ми сложи някакъв content

*a:after {content: ">>";}*

Структурни псевдо селектори

* **:root**

The root of the document -селектира елементи без родител <html> не се използва

* **E:nth-child(n)**

An E element, the n-th child of its parent селектира всяки n наследник от родителя (n=3, всеки трети наследник)

* **E:nth-last-child(n)** - същото но отзад напред

An E element, the n-th child of its parent, counting from the last onе

* **E:nth-of-type(n)**

An E element, the n-th sibling of its type

Ако имаме div селектира всеки трети див

* **E:nth-last-of-type(n)**

An E element, the n-th sibling of its type, counting from the last one

* **E:last-child**

An E element, last child of its parent

* **E:first-of-type**

An E element, first sibling of its type

* **E:last-of-type**

An E element, last sibling of its type

* E:**only-child**

An E element, only child of its parent

* **E:only-of-type**

An E element, only sibling of its type

* **E:empty**

An E element that has no children (including text nodes)

The UI Element States Pseudo-Glasses

* **E:enabled**

A user interface element E which is enabled

* **E:disabled**

A user interface element E which is disabled

* **E:checked**

A user interface element E which is checked (for instance a radio-button or checkbox). Currently supported only in Opera and IE10 !

* **E:target**

An E element being the target of the referring URI

* **E:not(s)**

An E element that does not match simple selector

* **E ~ F (tilda)**

An F element preceded (предшествано) by an E element

CSS Values

***Color***  *-* ***ColorZilla***

* **RGB**

#AACCBB(ABC)- шестнайсетична броина система

rgb(68,5,255) - десетична бройна система

rgba(255, 33, 0, 0.1)

a - alpha chanal = transperancy

* **Hue Saturation Light**

color: hsl(291, 85%, 89%);

color: hsl(291, 85%, 89%, 0.1)**;**

* **Size**

px, sm, inch

em- скалира спрямо размера на родителя

rem- скалира спрямо размера нa <body>

CSS Properties

Text-related CSS Properties

* **color** – specifies the color of the text
* **font-size** – size of font: xx-small, x-small, small, medium, large, x- large, xx-large, smaller, larger or numeric value
* **font-family** – comma separated font names

Example: verdana, sans-serif, etc.

* **font-weight** can be normal, bold, bolder, lighter or a number in range [100 … 900]
* **font-style** – styles the font

Values: normal, italic, oblique

* **font-variant-** specifies whether or not a text should be displayed in a small-caps font:
* normal -normal fonnt by default
* small-caps - all lowercase letters are converted to uppercase letters
* initial - set the property to its defalt value
* inherit - inherit the property from his parent
* **text-decoration** – decorates the text

Values: none, underline, line-trough, overline, blink

* **text-align** – defines the alignment of text or other content

Values: left, right, center, justify

* **padding** - разстоянието от текста до рамката
* **margine -**разстояние отвън между две рамки или два блокови елемента

Shorthand Font Property

*font: italic normal bold 12px/16px veranda*

font-style: italic

font-variant : normal

font-weight:bold

font-size: 12px

line-height:16px

font-family:veranda

More Fonts

За красиви и интересни букви -ART

*@font-face {*

*font-family: SketchRockwell;*

*src: url('SketchRockwell-Bold.ttf');*

*}*

*.my\_CSS3\_class {*

*font-family: SketchRockwell;*

*font-size: 3.2em;*

*}*

Sorce: Google Fonts

Other text effects

* Text shadow

Syntax: text-shadow: <horizontal-distance> <vertical-distance>  
 <blur-radius> <shadow-color>;

*text-shadow: 2px 2px 7px #000000;*

* Text Overflow - не се подържа на всички браузъри като Firefox,IE

*text-overflow: <value>;*

*overflow:hidden*

*witdth:400px*

ellipsis - текста минава на другия ред



clip - by default - cut the text



* word wraping: <value> : break and break-word

*-ms-word-break:break all;*

*word-break:break-all;*

Border Properties

* **border-width:** thin, medium, thick or numerical value (e.g. 10px)
* **border-color:** color alias or RGB value
* **border-style:** none, hidden, dotted, dashed, solid, double, groove, ridge, inset, outset

*Each property can be defined separately for left, top, bottom and right*

*border-top-style, border-left-color,....*

Border Shorthand Properties

**border:1px solid red**

border-style:solid

border-width:1px;

border -color:red

*Specify different borders for the sides via shorthand rules: border-top, border-left, border-right, border-bottom*

border:none or border:0?

Cool colored borders

* Only Firefox supports this type of coloring

*border: 8px solid #000;  
-moz-border-bottom-colors: #555 #666 #777 #888 #999 #aaa #bbb #ccc;  
-moz-border-top-colors: #555 #666 #777 #888 #999 #aaa #bbb #ccc;  
-moz-border-left-colors: #555 #666 #777 #888 #999 #aaa #bbb #ccc;  
-moz-border-right-colors: #555 #666 #777 #888 #999 #aaa #bbb #ccc*;

* **box shadow** - same as text shadow

*-moz-box-shadow: 10px 10px 5px #888;  
-webkit-box-shadow: 10px 10px 5px #888;  
box-shadow: 10px 10px 5px #888;*

* **border-radius-** Rounded corners are a part of CSS 3

*border-radius: [<length>|<%>][<length>|<%>]?*

*border-radius: 15px;*

*border-radius: 15px 15px 15px 10px*

*border-radius: 15px 20px;*

Backgraund Properties

* **background-image**

URL of image to be used as background, e.g.:

*background-image:url("back.gif");*

* **background-color**

Using color and image and the same time

* **background-repeat**

repeat-x, repeat-y, repeat, no-repeat

* **background-attachment**

fixed / scroll

* **background-position:** specifies vertical and horizontal position of the background image:
* Vertical position: top, center, bottom
* Horizontal position: left, center, right
  + Both can be specified in percentage or other numerical values

*background-position: top left;*

*background-position: -5px 50%;*

За да позиционираме къде точно да се намира фоновата картинка (например в td или bullet пред li ) се използват property;

*background-position :0 5px;*

*padding-left:=15px;*

Background Shorthand Properties

*background: #FFF0C0 url("back.gif") no-repeat fixed top****;***

*background-color: #FFF0C0;*

*background-image: url("back.gif");*

background-repeat: no-repeat;

*background-attachment: fixed;*

*background-position: top;*

Background Gradient

* Взимаме 1от картинка и я запазваме като нов файл. После й даваме repeat-x и се получава приливане на цветове. Колкото е по малка картинката толкова по бързо се предава по интернет
* Gradient с цветове - наизуст

Google - CSS gradient generator, colorzilla

/\* Firefox 3.6+ \*/

background: -moz-linear-gradient(100% 100% 90deg,

#FFFF00, #0000FF);

/\* Safari 4-5, Chrome 1-9 \*/

background: -webkit-gradient(linear, 0% 0%, 0%

100%, from(#0000FF), to(#FFFF00));

/\* Safari 5.1+, Chrome 10+ \*/

background: -webkit-linear-gradient(#FFFF00,

#0000FF);

/\* Opera 11.10+ \*/

background: -o-linear-gradient(#2F2727, #0000FF);

Multiple Background

Simple comma-separated list of images

Comma separated list for the other properties

background-image: url(sheep.png), url(grass.png);

background=position: center bottom, left bottom;

background-repeat: repeat-x, no repeat

Opacyty

Определя прозрачност на елементите.

-moz-opacity: 0.5;

opacity: 0.5;

filter: alpha(opacity=50);

zoom: 1;

CSS Layout

**Control the arrangement of the HTML elements**

Width and Height

* **width (height)** – defines numerical value for the width of element, e.g. 200px, отнася се само за размера на самото съдържание, не са включени margine, padding,border

width applies **only for block elements**

Their with is 100% by default

The width of inline elements is always the width of their content, by concept

* **min-width** **(height)** - defines the minimal width

min-width**(height)** overrides width if (width<min-width)

* max-width**(height)** - defines the maximal width

max-width **(height)** overrides width**(height)** if (width>max-width)

*Мерни единици*

* Centimeters (cm) - по рядко
* Рercentages - да се избягна, 100% by default. Оразмерява спрямо родителя или дефолтната стойност
* Pixels ( px) - препоръчително е да се използват

Min i Max се използва когатио сме стнали% за w h .

Overflow

Defines the behavior of element when content needs more space than the available

***overflow values:***

* visible (default) – content spills out of the element
* auto - show scrollbars if needed
* scroll – always show scrollbars
* hidden – any content that cannot fit is clipped

Display

Controls the display of the element and the way it is rendered and if breaks should be placed before and after the element - как елементите се подреждат на стрницата и какво влияние оказват върху другите елементи

* **inline:** no breaks are placed before and after (<span> is an inline element) за вмъкване не нещо в текст, било то маркировка или картинка

The element's height and width depend on the size of the content

* **block:** breaks are placed before AND after the element (<div> is a block element) Слага нов ред преди и след себе си. Все едно е <br/>

Height and width may not depend on the size of the content

* **none:** element is hidden and its dimensions are not used to calculate the surrounding elements rendering (differs from visibility: hidden!) Този елемент го има в ХТМЛ но не се визуализира и не се задава място за него ==0.
* **inline-block:** : no breaks are placed before and after (like inline)

Height and width can be applied (like block) Отива на нов ред но след него може да има и други елементи - основната му функция е да се оразмери елемента

* **table, table-row, table-cell** : the elements are arranged in a table-like layout - от <div> става таблица, table-row но search engines го индексират, вътре слагаме няколко <span> -table-sell
* grid - ново за HTML CSS, в web, но се използва за други UI приложения които се пошат с CHTML CSS and JS

Vizibility

Determines whether the element is visible

* hidden: element is not rendered (се рисува ), but still occupies place on the page (similar to opacity:0)
* visible: element is rendered normally

Margines and Рaddings

**Margine**- показва на какво разтояние даден елемент отблъсква елементите около него Spacing outside of the border

**Padding** - показва на какво разстояние се намира съдържанието от рамкатаma на елемента. Spacing between the border and the content

Мargine и margine се сливат

Рadding и margine не се сливат, избутват се взаимно, т.е ако родителя (външният елемент) има padding 20px а наледника (вътрешният елемент ) има margine 50px - разстоянието между рамката на родителя и насленика ще бъде 20+50=70px

Margin and Pading Short rules

* margin: 5px;

Sets all four sides to have margin of 5 px;

* margin: 10px 20px;

top and bottom to 10px, left and right to 20px;

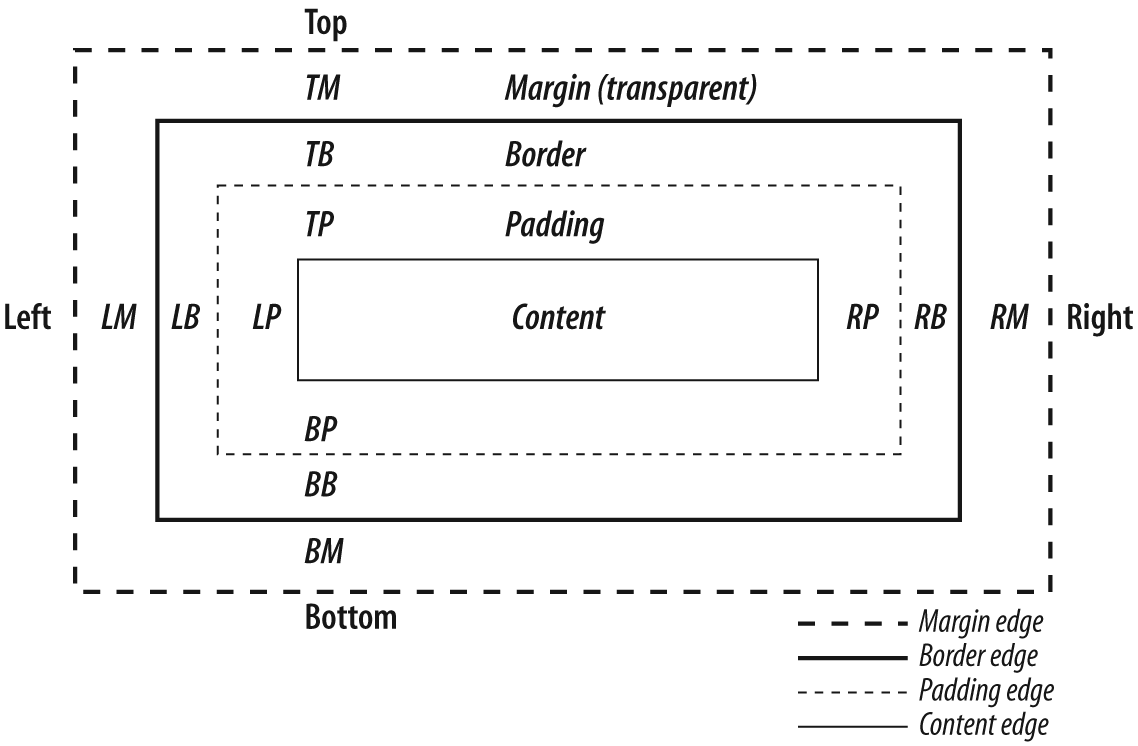
* margin: 5px 3px 8px;

top 5px, left/right 3px, bottom 8px

* margin: 1px 3px 5px 7px;

top, right, bottom, left (clockwise from top)

**Box Model**



Determine whether you want an element to render its' borders and padding within its specified width, or outside of it.

* **box-sizing: content-box (default)** размера на съдържанието  
  box width: 288 pixels + 10 pixels padding and 1 pixel border on each side = 300 pixels
* **box-sizing: border-box** = content + padding+borderbox width: 300 pixels, including padding and borders IE<=8(p 29 PPP)

better to set for positioning because include the width of whole element

/\* Firefox \*/

-moz-box-sizing: border-box;

/\* WebKit \*/

-webkit-box-sizing: border-box;

/\* Opera 9.5+, Google Chrome \*/

box-sizing: border-box;

Positioning

Defines the position of elements in the page content flow , допълнително изместване на елемента

Values:

* **static** - default
* **relative** - according to where the element would appear with static position, изместването е спрямо позицията където би се намирал елемента, като самата позиция остава (съседните елементи не се сливат и не заемат мястото на изместения елемент - то стои празно) Може да се разположи върху съществуващ вече елемент като го закрие изцяло или частично
* **absolute** - position according to the innermost positioned parent element спрямо първият позициониран над него елемент или body(ако няма позициониран родител). Ако родителя е позициониран релативно или абсолютно (дори условно с 0,0) абсолутното позициониране е спрямо родителя а не спрамо body. Съседните елементи се сливат като заемат мястото на позционираният елемент
* **fixed** - same as absolute, but ignores page scrolling- дори да скролваме съдържнието на сайта този елемент си остава статично на сайт.

Relative влияе върху подрдбата на другите елементи

position: relative(absolute

top (bottom) :(-)12px

left(right): (-)100px

Релативно позициониране не използваме ако можем да си позиционираме елементите с padding и margin един спрямо друг.

Релативно + абсолютно при наседника позициониране ползваме при dropdown menu

* Margin VS relative positioning
* Fixed and absolutely positioned elements do not influence the page normal flow and usually stay on top of other elements( не влиаят върху съдържанието на страницата и тяхното съдържание е върху другите елементи - за тях няма заделено място.
* Their position and size is ignored when calculating the size of parent element or position of surrounding elements
* Overlaid according to their z-index
* Inline fixed or absolutely positioned elements can apply height like block-level elements

Z-index - Определя кой елемент върху кой се намира

най-горният елемент е с най голям z-index

Float - дава се място на картинка или блоков елемент и останалият текст го обикаля

:left

:right