

CSS

PRE GDM A MPS

SOŠE TRNAVA, ING. PETER BARANÍK

Obsah

CSS – Cascade Style Sheet	3
Hierarchia štýlov	3
BOX MODEL CSS	3
SYNTAX CSS PRAVIDLA.....	3
Selektory.....	4
<i>Rozdiel medzi ID a CLASS</i>	4
Viacnásobné selektory.....	4
Sila selektorov	5
Dedičnosť Inheritance	5
Jednotky veľkosti v CSS.....	6
Nastavenie fontov pre stránku	6
Ostatné vlastnosti textu.....	7
Tvorba vodorovného alebo zvislého menu	7
<i>Časť HTML</i>	7
<i>Časť CSS</i>	7
Správanie odkazov	8
Pozadie background.....	8
Tabuľky	9
V html:.....	9
V časti CSS:	9
Štýly formulárov	10
Blokové a riadkové prvky	11
Blokové prvky.....	11
Riadkové prvky.....	11
Normálny tok stránky.....	11
Pozicionovanie.....	11
Vrstvy	12
Parameter float	14
Collapsing Margins.....	15
Použitie webových fontov.....	15
.....	15
MediaQueries.....	16
Pre tlač	16
<i>Pre tablety a mobily</i>	16
CSS3	18
<i>Viacnásobné obrázky pozadia</i>	18
<i>Rozťahovanie pozadia</i>	18
<i>Zmena správania pozadia</i>	18
<i>Priehľadnosť</i>	18
<i>Prechody</i>	18
<i>Zaoblenie rohov elementu border radius</i>	18
<i>Tiene box-shadow</i>	19
<i>Tiene textu text-shadow</i>	19
<i>Použitie vlastných fontov @font-face</i>	19
<i>Viac stĺpcový text</i>	19
<i>RGBA farba</i>	19
<i>Text Overflow</i>	19
<i>Animácie = zmeny stavu = transitions</i>	19
Animácie	20
<i>Border zložený z obrázkov</i>	20
Postup pri tvorbe CSS stránky	20
Použitie viacerých CSS modulov pomocou importu.....	21
Zaujímavé stránky.....	21
Na control kodu:	21
<i>Pikošky</i>	21

CSS – Cascade Style Sheet

Je textová informácia pridaná k webovej stránke za účelom nastavenia jej vzhľadu.

Ak niečomu štýl nenastavíme, zobrazí sa podľa nastavení prehliadača, čiže sa môže stať, že bude vyzeráť inak v každom prehliadači!!!!

Môže sa nachádzať na 3 miestach:

1. Priamo v tagu pomocou slova style

```
<h1 style="color:red;">Nadpis bude cerveny</h1>
```

 - volá sa INLINE štýl, vyhýbať sa!!!
2. V hlavičke stránky medzi slovami

```
<style type="text/css">
  h1 {color:red;}
</style>
```

 - volá sa EMBEDDED, zriedka!!!!
3. V externom súbore s príponou css. Je to textový súbor, kde sú zapísané rovnaké pravidlá, ako v hlave (bod 2). V hlave stránky je odkaz na tento súbor zobrazený ako

```
<link rel="stylesheet" type="text/css" href="mojstyl.css" />
```

a v samotnom súbore sú len CSS pravidlá.

```
p {color: red;}
```

Hierarchia štýlov

=určuje, ktorý štýl je dôležitejší, ktorý vyhráva

! Čím bližšie k tagu, tým väčšia sila!

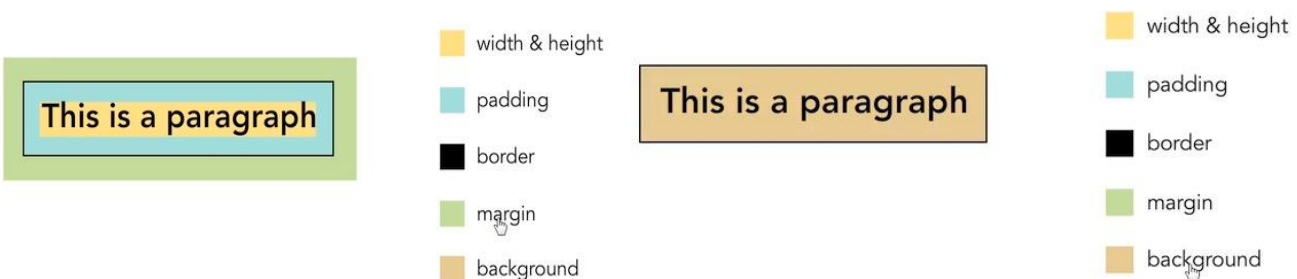
Sila podľa umiestnenia CSS:

Najsilnejšie je:

1. inline
2. v head=style
3. externý css
4. predvolené hodnoty prehliadača.

Ak sú rôzne pravidlá pre ten istý element v externom súbore, platí, že čím je pravidlo nižšie zapísané, tým má väčšiu silu.

BOX MODEL CSS



Ak nastavím divu šírku 300 px a výšku 100 px v skutočnosti je to viac, lebo sa k nim pripočítava padding = vnútorný okraj, border = orámovanie a margin = vonkajší okraj od strán

Ak ich nenastavíme prehliadač doplní pre margin a padding svoje hodnoty a border sa nezobrazí.

Príklad: odstavec šírka 300px margin 10 px, padding 5 px, border 2 px

dokopy šírka bude 334 px

Každá strana sa dá nastaviť samostatne pomocou:

margin-top, margin-right, margin-bottom, margin-left a rovnako aj padding a border.

!!!POZOR!!! Pozadie (farba alebo obrázok) sa "rozťahuje" iba POD elementom, paddingom a borderom, NIE POD MARGINOM.

Ak chceme vynútiť pevný rozmer elementu width a height včítane paddingu a borderu nastavíme mu

```
box-sizing: border-box;
```

```
body {
```

```
background-color: white;
```

```
color: black;
```

```
padding: 10px;
```

```
}
```

SYNTAX CSS PRAVIDLA

body je selektor = označiť, čoho sa pravidlo týka

2 - {

3 - vlastnosť (tu background-color, color a padding)

4 - : dvojbodka musí byť

5 - hodnota vlastnosti (tu biela, čierna a 10px) a za každou musí byť BODKOČIARKA;

6 - }

skrátenejší zápis

```
p {color:red; font: Arial;}
```

Selektory = to čo je pred {}

1. Jednoduché: TAGY:

- h1 {color:red;}, p, div, img, ul...

2. Triedy: meno triedy, trieda bude modrá

- .trieda {color:blue;}
- !Najprv treba triedu vyrobiť, potom použiť! v html takto:
<h1 class="trieda">Modrý nadpis</h1>
- !Dá sa použiť viac krát na stránke pre viac elementov!
- !Jeden element môže mať viac tried napr class="trieda vpravo tlacitko"!

3. ID štýly – pre elementy s ID, VYSKYTUJE SA LEN PRE 1!!!ELEMENT na stránke

```
#obsah {color:blue}
```

4. Pseudotriedy - rôzne. Začínajú dvojbodkou. Pri odkazoch sú štyri:

- a: link
- a: visited
- a: hover
- a: active

Ďalšie pseudotriedy sú:

- input:focus - na miesto formulára, do ktorého práve píšeme
- ::before h1 - je miesto PRED elementom h1; obsah sa pridáva s content: "bla"
- ::after h1 - je miesto ZA elementom h1

Rozdiel medzi ID a CLASS

Niektoré elementy na stránke môžeme pomenovať pomocou ID a potom im nastavovať CSS vlastnosti. Element ktorý má ID sa vyskytuje na stránke iba 1x, takže aj jeho CSS vlastnosti, sa týkajú len jeho. Zápis je

```
#meno {...;}
```

Preto ID používame len pre elementy, kde je podstatné ich ID, či už vo formulároch, alebo v JavaScripte.

Triedy sa používajú vtedy, ak chceme nastaviť nejakú vlastnosť viacerým prvkom naraz. Zápis vytvorenie triedy pomocou bodky

```
.menoTriedy {vlastnosti..;}
```

Pričom nestačí triedu vytvoriť v CSS, ale ju treba aj použiť v každom elemente, ktorý má mať požadované vlastnosti.

Robí sa to v HTML pomocou class:

```
<h1 class="menoTriedy inaTrieda"> ...</h1>
```

Výhody:

- viac Elementov naraz
- možnosť použitia viacerých tried
- sú pripravené aj pre neexistujúce elementy
- ak chceme niečo zmeniť, stačí zmeniť parameter triedy, a zmenia sa všetky elementy, ktoré túto triedu majú

Viacnásobné selektory

1. * {margin:0; padding:0;}

- o VŠETKY ELEMENTY= !!! prvé pravidlo CSS na stránke!
- o vymaže okraje,treba potom nastaviť!
- o správny CSS súbor nájdete na meyerweb.com/eric/tools/css/reset/ alebo na <https://necolas.github.io/normalize.css/>

2. h1, h2, h3, li {font-family:Arial;}

3. Vnútorne selektory

- o div p {color:red;} = odstavec v dive
- o #obsah #clanok p {width:300px;height:200px;} odstavec v elemente s ID clanok a ten je v elemente s ID obsah
- o aside h1.velky {font-weight:bold;} nadpis 1 v elemente aside s triedou velky

- o `div > p` priamy potomok divu = všetky odstavce v dive nevnorené do iného tagu.
- o `div > h2 + p` najbližší súrodeneц h2 v dive = jason sibling
- o `p:first-line`, `p:first-letter` je od CSS1, v CSS3 sa píše `p::first-line`, `p::first-letter`
- o `li:first-of-type` a ...
- o `li:last-of-type` a ...
- o `p:only-of-type` ...
- o `h2 ~ p` general sibling
- o `ul li:not(:last-child) {}` vyberie všetky okrem posledného
- o `td:empty` vyberie prázdne bunky
- o `h2:target` na zvýraznenie miesta na stránke KOTVY kam nás hodil vnorený odkaz
- o `[id]` vyberie všetky elementy, čo majú nejaké ID.
- o `label[for=terminy]` vyberie label s for s názvom terminy

Selektory na označenie nejakého prvku v poradí sú:

Number, e.g., `li:nth-child(3)`

Keywords `odd` and `even`, e.g., `tr:nth-child(odd)`

Formula `an+b`, e.g., `tr:nth-child(4n+1)`

```
:nth-child()
:nth-last-child()
:nth-of-type()
:nth-last-of-type()
```

Vyzerá to v tabuľke:

```
tr:nth-child(4n+1)
(4 x 0) + 1 = 1
(4 x 1) + 1 = 5
(4 x 2) + 1 = 9
tr:nth-child(4n+2)
(4 x 0) + 2 = 2
(4 x 1) + 2 = 6
(4 x 2) + 2 = 10
```

	MAX TEMP °C (F)
Jan	7.2 (45.0)
Feb	7.5 (45.5)
Mar	10.1 (50.2)
Apr	12.5 (54.5)
May	16.3 (61.3)
Jun	19.1 (66.4)
Jul	21.7 (71.1)
Aug	21.6 (70.9)
Sep	18.5 (65.3)
Oct	14.5 (58.1)
Nov	10.3 (50.5)
Dec	8.1 (46.6)

Sila selektorov

Ak sa ten istý element vyskytne viac krát v CSS, platí pravidlo o SILE pravidla:

Selector	Specificity
Inline style	1, 0, 0, 0
ID	0, 1, 0, 0
Class, pseudo-class, attribute	0, 0, 1, 0
Type (tag), pseudo-element	0, 0, 0, 1
<code>#sidebar p</code>	0, 1, 0, 1
<code>blockquote p</code>	0, 0, 0, 2
<code>#sidebar blockquote p</code>	0, 1, 0, 2

Ak majú selektory rovnakú silu, vyhráva ten, čo je v CSS napísaný nižšie.

Ak si nevieme rady, použijeme výnimočne:

When all else fails

```
p em {
  color: #f00 !important;
}
```

Dedičnosť Inheritance

Vlastnosti elementov stránky sa dajú rozdeliť do dvoch skupín, podľa toho, či ich zdedia, aj elementy v jeho vnútri (napríklad odstavce v dive).

1. **Dedičné** vlastnosti sú v zozname. Uplatňujú sa hromadne. Môžu byť pretlačené aby sa neuplatnili = vyradené pomocou inheritance: none;
 - o `border-collapse`
 - o `border-spacing`
 - o `color`
 - o `cursor`
 - o `direction`
 - o `font-family`
 - o `font-size`
 - o `font-style`
 - o `font-variant`
 - o `font`
 - o `font-weight`
 - o `letter-spacing`

- *line-height*
- *list-style-image*
- *list-style-position*
- *list-style-type*
- *quotes*
- *text-align*
- *text-indent*
- *text-transform*
- *visibility*
- *volume*
- *white-space*
- *windows*
- *word-spacing*

2. Tie ktoré nededia vlastnosti rodičov = ostatné, hlavne nie sú marginy, paddingy, pozadia.
3. Z body color sa dedí ALE odkazy a formuláre nededia takmer nič.
4. Nadpisy nededia font-size. Ak chceme vynútiť dedenie napr. farby, dáme pre odkaz color: inherit.

Jednotky veľkosti v CSS

Absolútne:

- Pixely px
- Point (pt) je tlačový bod
- Dĺžkové- mm, un, inch (palce) – len na tlačenie
- *Percentá – napríklad dĺžka sa udáva z najbližšieho rodiča*
- *em – je určené pre všetky typy zobrazovacích zariadení a pri zobrazení na monitore platí, - IEM=16px. JE VIAZANÝ NA NADRIADENÝ PRVOK.*
- *rem = em –ALE JE viazaný na body a nie na nadriadený element. Používať tento. Sú najvýhodnejšie, lebo každé zariadenie má inú veľkosť podľa displeja.*
- vw = percentá zo šírky okna
- vh = percentá z výšky okna
- vmin = percentá z menšieho rozmeru okna (šírka / výška)
- vmax = percentá z väčšieho rozmeru okna (šírka / výška)
- *Bez jednotky = multiple – udáva násobok niečoho iného*

Nastavenie fontov pre stránku

- 1) Všetkým elementom nastav margin a padding na 0
- 2) Pre <body> nastavíme rodinu fontov- viac, ak by zariadenie nemalo nainštalované prvé fonty, posledný je vždy- serif alebo san-serif /{font family:"Verdana","Arial", san-serif;}
- 3) Pre body nastav veľkosť fontu v percentách 100%
- 4) Každému textovému tagu nastavíme font-size v „rem“ (1rem je 16px pre PC, každé zariadenie má svoju veľkosť rem)
- 5) Pre odstavce a nadpisy nastavíme výšku riadka „line-height“ v percentách alebo násobok napr 1.5
!v percentách alebo násobok sa počíta z veľkosti fontu! Line-height určuje vzdialenosť 2 riadkov od seba a zároveň vycentruje 1-riadkový text na výšku.

Pre bežný odstavec sa nastavuje výška riadka 1.4 pre obrazovky a 1.2 pre tlač.

- 6) Centrovanie textu :

a) Horizontálne=text-align :

- -center
- -inherit
- -justify
- -left
- -right

b) Vertikálne

- i) jednoriadkový text je vycentrovaný pomocou line-height väčší ako 1
- ii) viac riadkový text sa centruje:

Otcovi dáme

```
display: flex; * alebo grid
justify-content: center;
align-items: center;
```

Ostatné vlastnosti textu

- Farba písma
 - color: darkblue alebo
 - sa použije namiešanie farieb color: #AA00CC alebo
 - color: rgba(255(R),10(G),20(B),0.1(priehľadnosť))
- Font font-style: italic, normal, oblique
 - font-weight: 100-900, bold je 700, normal je 400
 - font-variant: normal, small caps,
 - text-transform: capitalize/lowercase/uppercase,
 - text-decoration: line-through, none, overline, underline
 - word-spacing: určuje vzdialenosť medzi slovami
 - letter-spacing: určuje vzdialenosť medzi písmenami
 - text-indent: určuje odsadenie textu prvého riadka

Väčšina vlastností textu je dedičná.

Tvorba vodorovného alebo zvislého menu

Časť HTML

1. Napíšeme texty odkazov
Hory
Lesy
More
2. Spravíme z nich linky

```
<a href="hory.html">Hory</a>
<a href="lesy.html">Lesy</a>
<a href="more.html">More</a>
```
3. Z každého odkazu spravíme položku nečíslovaného zoznamu

```
<ul>
  <li><a href="hory.html">Hory</a></li>
  <li><a href="lesy.html">Lesy</a></li>
  <li><a href="more.html">More</a></li>
</ul>
```
4. Skontrolujeme že sa začína a končí ul.

Časť CSS

1. Odstránime podčiarknutie odkazov pomocou

```
ul li a{text-decoration:none;}
```
2. Odstránime z nečíslovaného zoznamu odrážky pomocou

```
ul{list-style-type:none;}
```
3. Nastavíme farbu písma a farbu alebo obrázok pozadia pre odkaz

```
ul li a {
  color: ...;
  background-color:....;
}
```
4. Nastavíme odkazu vlastnosť ktorá zabezpečí že bude zaberáť celú šírku akú má mať nielen šírku textu

```
ul li a {display: block;}
```
5. Pri vodorovnom menu dáme odkazy vedľa seba pomocou – sú 2 techniky, pomocou
 - a.

```
ul li {display:inline;} alebo s float (presnejšie)
```
 - b.

```
ul li {float:left;} alebo right
```
6. Nastavíme vzdialenosti medzi odkazmi pomocou margin-left a margin-right.
7. Pri zvislom menu bod 5 netreba

Výhodou použitia zoznamu na tvorbu menu je:

1. veľmi ľahko pridáme alebo odstránime novú položku
2. v kóde toto menu vždy rýchlo nájdeme.

Na označenie aktívneho odkazu stránky na ktorej sa práve nachádzame, použijeme triedu .aktivna{...;} a v nej nastavíme kurzor aby sa nemenil pomocou
cursor:default;

Správanie odkazov

Odkazy môžeme nastaviť tak že keď sme na nad nimi myšou zmenia svoj stav. Robí sa to pseudotriedami, pričom je dôležité ich poriadie. Musia byť aj viaceré naraz ALE v poradí:

`link visited hover active`

a:link je základný stav odkazu, a:visited je stav už navštíveného odkazu, a:hover je stale odkazu v okamžiku keď je nad ním myš. Príklad: html časť hore pre menu pod sebou. CSS:

```
a {
    text-decoration:none;
    display: block;
}
a:link, a:visited {
    color: #fff;
    background-color: #0000dd;
    border: 1px solid black;
    border-collapse: true;
}
a:hover, a:focus {focus je pre posun s TABulátorom
    color: #0000dd;
    background-color: #fff;
    outline:none;
}
.active {
    color: #fff;
    background-color: #00dddd;
}
```

Pozadie background

Dá sa nastaviť pre každý element, prípadne pre celú – body

Môže byť:

- Farba background-color
- Obrázok (alebo aj viac) background-image url(adresa/cesta)
 - je silnejší než farba
 - aby sa obrázok neopakoval background-repeat: no-repeat
 - ak chceme aby obrázok prikryl celý objekt (stránku) **background-size: cover** alebo 100% (pomenej), ostatné možnosti sú rozmery napr. 300px 100px (šírka výška), alebo contain.
 - aby sa obrázok neskrolloval background-attachment: fixed
 - pozícia pozadia: background-position:
 - top left center right bottom (zľava doprava aj zhora nadol; prednastavená je center center)
 - 10px 2em = prvé číslo je zľava, druhé zhora
 - ak sú napr.obrázky 2 a meníme ich position, bude background-position: 20px 30px, 40px 20px; prvé dve pre prvý obrázok, a potom pre druhý, podobne repeat
 - v % pričom 0% 0% umiestni obrázok doľava hore a 100% 100% doprava dole
- Obrázkom pozadia môžeme nahradiť odrážku v nečíslovanom zozname, no-repeat a pohrať sa s pozíciou pozadia.
- Ak je viac obrázkov pozadia, prvý uvedený je vždy navrchu.
- Normálne je poloha obrázka vľavo hore a je pod borderom ;dá sa zmeniť s background-origin: padding = zmení ho tak, aby nebol pod borderom. background-clip:padding-box urobí to isté.
- Obrázkom pozadia môžeme nahradiť napríklad nadpis, ale v html nechajme aj jeho text napr. slovo kontakt `<h2>Kontakt</h2>` ALE aby sa tento text nebil s pozadím, posunieme ho doľava s h2 {text-indent:-1000em;} Tým zabezpečíme, že text bude na stránke viditeľný pre čítačky, a bude sa aj tlačiť. (obrázky pozadia sa netlačia).
- Farba alebo obrázok pozadia sa používa aj ako dynamický prvok = ak sme niekde s myšou nad niečím na zvýraznenie kde sme, alebo v akom stave je odkaz. Vždy sú na to pseudotriedy hover: alebo focus:
 - na odkaze a:hover {zameníme farby}
 - na zvýraznenie riadku tabuľky kde sme s kurzorom tr:hover {zameníme farby}
 - vo formulari na zvýraznenie poľa, ktoré vyplňame input:focus {background....}

- často použijeme pod odkaz obrázok, a aby sa ich nestáhalo veľa, spoja sa 3 obrázky do jedného, a ten sa pod odkazom posúva podľa stavu odkazu pomocou kľúčových slov


```
a:link, a:visited {background-position: left;}
a:hover {background-position: center;}
a:active {background-position: right;}
```
- Pozadie sa dá nastaviť aj skráteným zápisom:


```
background: url(img/a.gif) no-repeat 30px;
!!! záleží na poradí vlastností !!!
```

!!!! Obrázky pozadia sa NETLAČIA a nemajú iné atribúty, ako normálne obr.!!!!



Tabuľky

V html:

- 1) Dajme tabuľke atribúty id(meno) alebo triedu, summary (čoho sa týka) a vynulujme už v html border, cellpadding, cellspacing.
- 2) `<table border="0" cellpadding="0" cellspacing="0" id="kamery" summary="Porovnanie kamier">`. Pri dôležitých tabuľkách dajme aj popis tabuľky = býva nad ňou. !!!ZA `<table>!!!`
`<caption>Camera Specs</caption>`
- 3) Dôležité bunky = hlava tabuľky (býva 1.riadok alebo stĺpec) označíme nie `<td>` ale `<th>`. Bývajú tučným a zarovnané na stred.
- 4) Tabuľka je jediné miesto, kde sa dá nastaviť zvislé zarovnanie obsahu bunky s
 - o vertical-align: top, bottom, middle
- 5) Pre nevizuálne prehládanie môžeme bunke hlavičky dať atribút abbr čo je vlastne skrátená verzia obsahu bunky.
 `<th abbr="sony">Sony DSC-F717</th>`

V časti CSS:

1. Pre tabuľku nastavíme šírku, zarovnanie, okraj pozadie, fonty atd.

```
table #kamery {
width: 500px;
border: 1px solid #333;
margin: 0 auto;
border-collapse: collapse;
background: #574F3E;
}
```

2. Nastavíme bunky hlavičky th

```
table#kamery th {
font-family: ...;
font-size: ...;
color: ...;
background: ...;
height: ...;
vertical-align: bottom;
border: 1px solid #333;
}
```

3. Nastavíme popis caption

```
table#kamery caption {
font-family: .....;
font-size: ...;
font-weight: ...;
color: ...;
}
```

4. Nastavíme vlastnosti bunky td

```
table#kamery td {
padding: 1.5em 1em;
border: 1px solid #333;...
}
```

5. Na oddelenie riadkov tr striedame farby ich pozadia pomocou pseudotried `:nth-child(odd) /: nth-child(even)` alebo `nth-child(2n+1) ...`

```
table#kamery tr:nth-child(odd) {
    background: #8C7E62;
}
```

6. Na orientáciu v tabuľke dame zmenu farby riadka, nad ktorým je kurzor pseudotriedou `:hover`

```
table#kamery tr:hover {
    background: #333;
}
```

Pseudotriedy a pseudoelementy

The pseudo-class concept is introduced to permit selection based on information that lies outside of the document tree or that cannot be expressed using the other simple selectors.

A pseudo-class always consists of a "colon" (:) followed by the name of the pseudo-class and optionally by a value between parentheses.

Pseudo-classes are allowed in all sequences of simple selectors contained in a selector. Pseudo-classes are allowed anywhere in sequences of simple selectors, after the leading type selector or universal selector (possibly omitted). Pseudo-class names are case-insensitive. Some pseudo-classes are mutually exclusive, while others can be applied simultaneously to the same element. Pseudo-classes may be dynamic, in the sense that an element may acquire or lose a pseudo-class while a user interacts with the document.

Pseudo-elements create abstractions about the document tree beyond those specified by the document language. For instance, document languages do not offer mechanisms to access the first letter or first line of an element's content. Pseudo-elements allow authors to refer to this otherwise inaccessible information. Pseudo-elements may also provide authors a way to refer to content that does not exist in the source document (e.g., the `::before` and `::after` pseudo-elements give access to generated content).

A pseudo-element is made of two colons (:) followed by the name of the pseudo-element.

This `::` notation is introduced by the current document in order to establish a discrimination between pseudo-classes and pseudo-elements. For compatibility with existing style sheets, user agents must also accept the previous one-colon notation for pseudo-elements introduced in CSS levels 1 and 2 (namely, `:first-line`, `:first-letter`, `:before` and `:after`). This compatibility is not allowed for the new pseudo-elements introduced in this specification.

Only one pseudo-element may appear per selector, and if present it must appear after the sequence of simple selectors that represents the subjects of the selector.

Note: A future version of this specification may allow multiple pseudo-elements per selector.

The biggest difference is that they actually create a new virtual element on which rules and even pseudo-class selectors can be applied to. They don't filter elements, they basically filter content (`::first-line`, `::first-letter`) and wrap it in a virtual container, which the author can style however he want (well, almost).

For example the `::first-line` pseudo-element cannot be reconstructed with JavaScript, as it heavily depends on the current used font, the fonts size, the elements width, floating elements (and probably the time of the day). Well, that's not entirely true: one could still calculate all those values and extract the first line, however doing so is a very cumbersome activity.

```
q:lang(de)::after{
    content: " (German) ";
}
```

```
q:lang(en)::after{
    content: " (English) ";
}
```

```
q:not (:lang(de)):not(:lang(en))::after{
    content: " (Idi v pyzdu nemertem) ";
}
```

Štýly formulárov

Nastavujeme hlavne šírku, farby písma a pozadia pre label, legend, a ich rozostupy pomocou margin. V html kóde treba spárovať popisok label pomocou `for="meno"` s `id="meno"` v input elemente.

Je výhodné používať triedy na nastavenie rovnakej šírky input. Do tejto triedy môžeme priradiť príkaz `display:block` pre input typu text, čím ho dáme do samostatného riadku.

Na zaškrťavacie alebo výberové políčka sa táto trieda neaplikuje, takže zostanú vedľa seba.

Na zvýraznenie toho poľa, ktoré práve vypisujeme, použijeme pseudotriedu `focus`
`input.text:focus, textarea:focus {...;}`

Na rozostupy NEPOUŽÍVAŤ `
` ale MARGINY!!!!

input atribút `for` sa viaže na ID v inpute, alebo sa dá použiť stratégia:

```
<label>
    <input type="checkbox" value="" checked="" /> patriaci k text labels inputu
</label>
```

a daj mu v css

```
label {cursor:pointer;}
```

Rozdiel je v tom, že teraz funguje aj kliknutie na text v labeli ako na input, hneď sa zaškrtnie a podobne. Pridané atribúty:

- o `checked` znamená že bude vybratý hneď
- o pre `select` je to `selected`

odosiela sa value hodnota po kliknutí na button. Toto sa dá použiť na input aj `checkbox`, `select`, `radio`.

```
<input type="file" .../>
```

v css mu daj

```
input[file]:focus {outline:none;}
```

Pre označení niečoho napr. v formulári dáme

```
::selection {background-color, color, text-shadow:none;}
```

Blokové a riadkové prvky

Blokové prvky

Sú tie, ktoré akoby končili zalomením riadka, t.j. každý prvok za ním začína na novom riadku. Patria sem skoro všetky tagy: `h1` až `h6`, `p`, `q`, `div`, `ul`, `ol`, `li`, `hr`, `br`, `address`, `form`, `table`, `tr`.

Dá sa im nastaviť šírka a výška aj všetky marginy.

Riadkové prvky

Sú tie, ktoré nekončia zalomením riadka = ďalší prvok sa zobrazí hneď vedľa neho. Patria sem iba `img`, `a`, `span`, `input`, `td`.

NEMAJÚ možnosť nastaviť šírku výšku ani `margin-bottom/top`.

Ich správanie sa dá zmeniť pomocou css vlastnosti:

`h3 {display:inline;}` zmení nadpis 3 na riadkový

`img, a {display: block;}` zmení obrázky a odkazy na blokové.

`display:inline-block` zmení riadkový element tak, že ostane v riadku ALE môžeme mu dať `width`, `height` aj všetky marginy.

Normálny tok stránky

Je taký, ako je stránka písaná v kóde = zobrazuje sa v tom poradí zhora zľava doprava dole.

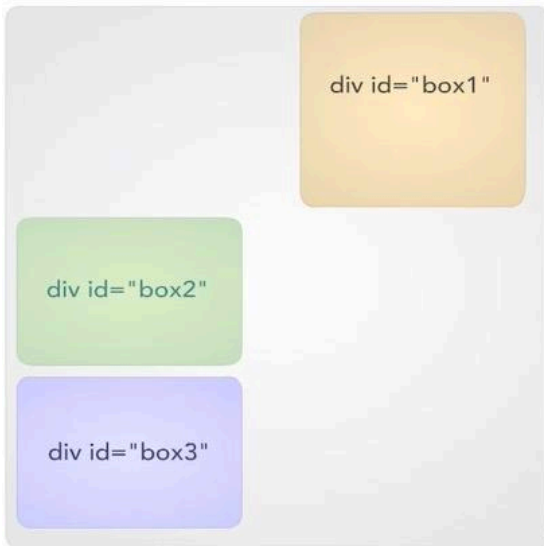
Dá sa narušiť s css vlastnosťami ako je `float` a `position`, `z-index`.

Pozicionovanie

Position- slúži nastavenie umiestnenie prvku a môže byť:

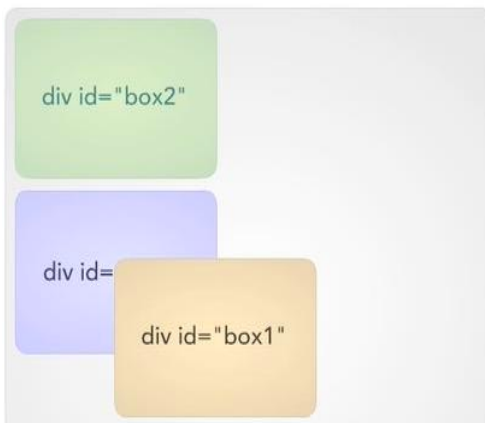
1) **static**-východiskové-normálny tok dokumentu, nemožno nastaviť súradnice ;

2) **relative**-prvok je v normálnom toku, ale je posunutý o zadané súradnice: **top**, **left**. Ostatné prvky sa zobrazia tak, ako by posunutý nebol; Pôvodné miesto prvku ostáva prázdne.



```
#box1 {
  position:relative;
  width: 100px;
  left: 120px;
}
```

3) **absolute**-prvok sa zobrazí v polohe podľa súradníc vzhľadom na rohy nadriadeného elementu/tagu, ten je body alebo najbližší ktorý má nastavenú : position- absolute/relative. Ostatné elementy ho nevidia, a zaujmú jeho pôvodné miesto. Posúva sa slovami **top**, **left**, **bottom**, **right**.



```
#box1 {
  position:absolute;
  width: 100px;
  left: 50px;
  top: 200px;
}
```

4) **fixed**-zostáva na mieste ktoré má určené voči oknu prehliadača (neskroluje sa, zostáva na tom istom mieste). Posúva sa slovami **top**, **left**, **bottom**, **right**.

5) **sticky** = kombinácia relative a fixed, skroluje sa so stránkou až do polohy nastavenej s top: a tam ostane natrvalo.

!3 a 4 vytrhne prvok z normálneho toku, ostatné ho nevidia !

Hodnoty posunu možno zadať ako:

- vzdialenosť - v bežných jednotkách
- v percentách - z rozmerov obmedzujúceho bloku

Ak chceme centrovať element čo má absolute, nedá sa s margin auto ale dáme mu:

```
left: 50%;
```

```
margin-left: -polovica šírky elementu;
```

Vrstvy

Poradie objektu sa dá určiť vlastnosťou

z-index: auto- prvok je vo vrstve ako rodičovský prvok

(číslo)-čím je väčšie, tým vyššie sa prvok umiestni

Uplatní sa LEN pre element s vlastnosťou position. Ak nie je uvedené, je navrchu ten prvok, ktorý bol neskôr v kóde.

Vlastnosť display- slúži na nastavenie typu (spôsobu zobrazenia) prvku

- None: prvok sa nezobrazí
- Inline: prvok je riadkový
- Block: prvok je blokový
- Inline-block: ostane riadkový, ale dá sa mu nastaviť výška šírka a všetky marginy
- List-item: prvok je zobrazený ako položka zoznamu
- hidden; spôsobí, že element nevidno, ALE ostatné sa správajú, akoby tam bol

Parameter float

Určuje obtekanie objektu a môže mať tieto hodnoty:

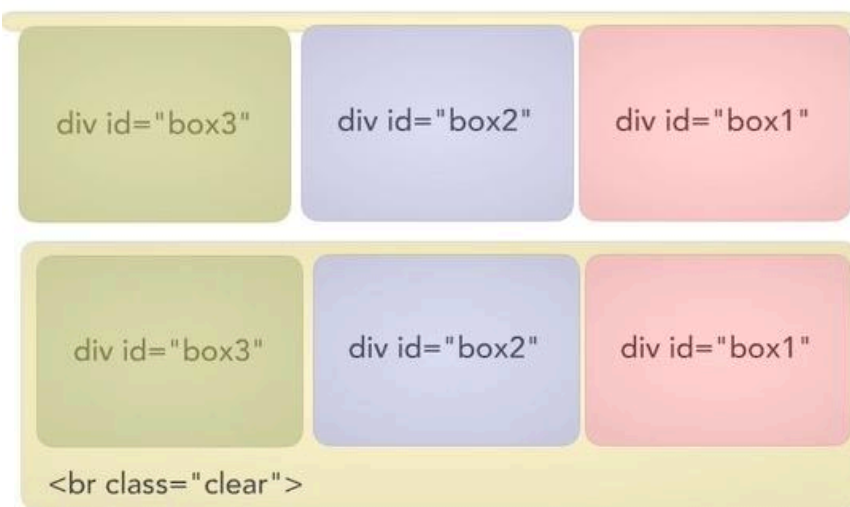
- **none**-prvok nie je obtekaný
- **right/left**-doprava/dola

Ak dám viac elementom za sebou, skladajú sa vedľa seba, a keď už v riadku nie je miesto, skladajú sa do ďalšieho ale TAK, ABY boli čo najvyššie!!! POZOR NA TO.

Všetkým elementom s float nastav aspoň width!

!float ruší normálny tok stránky, vytrhne element z toku a stáva sa že nadriadený element ani elementy za ním ho nevidia!

Na obnovenie normálneho toku stránky sa vyrába trieda (napr.: obnov, cistic,...nie anglické názvy) .obnov{clear:both;} a triedu použijeme na mieste kde chceme obnoviť tok stránky (prázdny znak, zlom riadka,...)



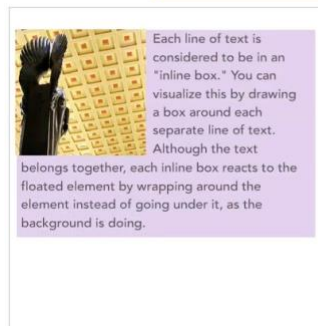
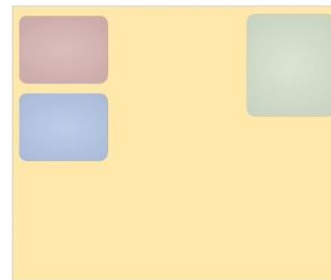
Add a
 element after the last floated element (any element will do, spans and br elements are often used due to their neutral effect on layout)

Možnosti na obnovenie normálneho toku stránky po použití float:

1. dáme float aj nadriadenému = problémy
2. nadriadenému presnú výšku a overflow:hidden vyreže čo sa nezmesť alebo overflow:auto a vyrobí scrollbar naboku
3. za elementy float dáme: <br class="clear">.clear {clear:both;}
4. použi vyrobenú triedu .group = obrázok NAPRAVO.

Snažíme sa použiť float čo najmenej, ak je viac elementov na strane pod sebou dáme ich do spoločného dovu a tomu dáme float.

float: left
float: right



img {float:left;}

```
#box1, #box2, #box3 {  
  float:right;  
}
```

```
#box1, #box2, #box3 {  
  float:right;  
}
```

```
br.clear {  
  clear:both;  
}
```

Strategies for containing floats

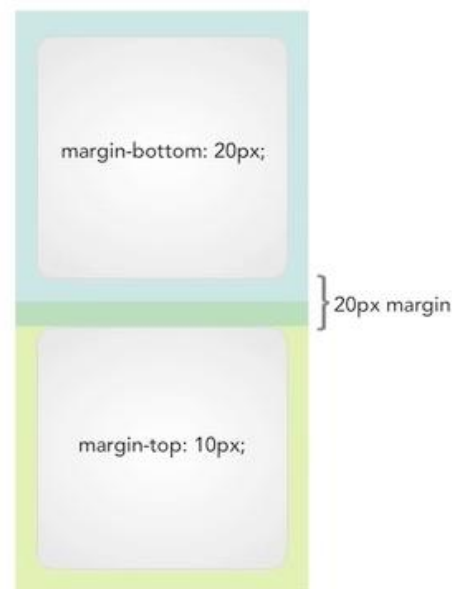
1. Float the container
> may cause layout issues
2. Clear contained element
> may require non-semantic markup

```
.group:before,  
.group:after {  
  content: " ";  
  display: table;  
}  
  
.group:after {  
  clear: both;  
}  
  
.group {  
  *zoom: 1;  
}
```

Collapsing Margins

Je zlievanie vonkajších okrajov objektov ktoré sú nad sebou v zvislom smere. Nenastáva v týchto prípadoch:

- Pri elementoch s float
- Pri position absolute
- Pri elementoch inline/block
- elementy s overflow inom ako visible
- elementy s parametrom clear



Since nothing separates the two margins, they collapse into one 20px margin.



Riešenie:

Jednému z elementov dáme požadovaný margin

Pri vnorených elementoch dáme nenulový padding

Pri elementoch nad sebou nastavíme border pre jeden z nich – ten medzi nimi

Použitie webových fontov

Patrí do CSS3. Do googla dáme google web fonts a nájdeme si, ktorý sa nám hodí.

Pozrite si vždy sadu EXTENDED, a skontrolujte znaky č š ž ť ú ô, či ich zobrazí správne.

Priamo na stránke je tlačítka na skopírovanie kódu do našej stránky, a ten dáme do nášho CSS, najlepšie do úvodnej časti.

```
<link href="https://fonts..." rel="stylesheet">
```

Ak vložíme odkaz z tohto kódu do adresy prehliadača, otvorí sa kód CSS pre import tohto fonu.

Ak mám font stiahnutý, môžeme ho vložiť do CSS priamo(ak je v priečinku fonts):

```
@font-face {  
    font-family: jano;  
    src: url(fonts/str.ttf);  
}
```

Nie každý font má kurzívu, vtedy sa použije rez oblique, čo je trocha zaklonené písmo.

Typy web fontov:

Embedded OpenType (.eot)
Internet Explorer
TrueType/OpenType (.ttf & .otf)
Firefox, Opera, Safari, Chrome
Scalable Vector Graphics (.svg)
iPhone, iPad
Web Open Font Format (.woff)
W3C standard—supported by IE 9 and all other browsers, except Safari

Hrúbka fontov

font-weight: 400 je normal, 700 je bold, 100 až 300 je tenké.

MediaQueries

Patria do CSS3!!! Vytvárajú iné správanie stránky pri zobrazení na mobile, tablete aj pri tlači. Rozhoduje slovo `media="screen/projection/print"`, potom rozmery displeja resp. prehliadača a jeho orientácia.

1.možnosť – link na ext. súbor

2.možnosť – dnes používaná:

```
@media print {..}
```

```
<link href="styles/tozai.css" rel="stylesheet"
type="text/css" media="screen">
<link href="styles/print.css" rel="stylesheet"
type="text/css" media="print">
```

Pre tlač

- nezobrazujeme pozadia – farby ani obrázky
- veľkosti písma sú v pt (tlačené body)
- farby písma sú čierne
- nezobrazujeme hlavičky, boč.panely, vodor. ani zvislé menu, päťu, nepotrebné obrázky
- obrázky zmenšíme na veľkosť v mm (`max-width:...`)
- ak chceme sami zalomiť stránku na nejakom mieste (nadpise), dáme mu triedu `.zlom {page-break-before: always;}`
- ak sú v texte dôležité oznamy, môžeme ich vytlačiť pomocou:
 - `a[href^=http://]:after {content="pozri online na: " attr(href);}`
 - o v preklade to znamená že absolútny odkaz (začína http..) bude pridaný za odkaz textu s textom pozri online na ... adresa.

Pre tablety a mobily

- môžeme vytvoriť samostatné css súbory a naimportovať ich pomocou `@import url(phone.css) only screen and (min-width: 361px);`
- alebo umiestniť do hlavného css

```
@media only screen and (max-width: 1360px)and (min-width:600px ) {
  header h1 {
    font-size: 36px;
  }
}
```
- použitie môžeme rozmery displeja, maximum aj minimum, aj ich kombinácie pomocou slov `and` a `or`, prípadne orientáciu na výšku – šírku. Podpora v prehliadačoch je na obrázku.
- Spájanie viac podmienok je pomocou slov `and` alebo `or`

```
@media ((max-width: 1360px) and (min-width:600px)) {
  ...
}
```
- Orientáciu displeja zadávame pomocou `orientation: portrait` alebo `landscape`

Feature	Value	Min/Max	Relates to
width	Length	Yes	Display area
height	Length	Yes	Display area
device-width	Length	Yes	Rendering area
device-height	Length	Yes	Rendering area
orientation	portrait landscape	No	Device orientation

CSS3 Media Queries - Candidate Recommendation

Method of applying styles based on media information. Includes things like page and device dimensions

Resources: [IE demo page with information](#) [Demo page for page width](#)

	IE	Firefox	Safari	Chrome	Opera	iOS Safari
Two versions back	7.0	3.5	3.2	9.0	10.6	3.2
Previous version	8.0	3.6	4.0	10.0	11.0	4.0-4.1
Current		4.0		11.0	11.1	4.2-4.3
Near future	9.0	5.0	5.0	12.0	11.5	
Farther future	10.0	6.0	6.0	13.0	12.0	

Note: Incomplete support by older webkit browsers refers to only acknowledging different media rules on page

CSS3

Pridalo oproti predošlej verzii niekoľko možností. Staršie prehliadače to nemusia všetko podporovať, a tak takmer každú vlastnosť by bolo treba napísať viac x a to najprv s predponami:

```
-webkit-text-shadow: 1px pre safari chrome a edge
```

```
-moz-text-shadow: 1px pre mozilla
```

```
-o-text-shadow: 1px pre operu
```

```
-ms-text-shadow: 1px pre explorer
```

a až potom všeobecný zápis pre moderné prehliadače

```
text-shadow: 1px
```

Na uľahčenie to za vás urobí

<http://output.jsbin.com/gufoko> doplní váš CSS3 kód o riadky tak aby bol kompatibilný aj so staršími prehliadačmi.

Do CSS3 patria:

Viacnásobné obrázky pozadia

Viac farieb aj viac obrázkov pozadia. Vyhráva ten nižšie napísaný. Zápis je každá vlastnosť oddelená čiarkou napr.

```
background-image: url(img_flwr.gif), url(paper.gif);
```

```
background-position: right bottom, 10px top;
```

```
background-repeat: no-repeat, repeat;
```

Rozťahovanie pozadia

```
background-size: (cover/contain) _vertikalny_rozmer_ _horizontalny_rozmer_;
```

Ak chceme obrázok pozadia vidieť vždy celý:

```
background-size: cover;
```

```
background-attachment: fixed;
```

```
background-position: center;
```

Zmena správania pozadia

```
box-sizing: content-box | border-box | padding-box
```

Zmení odkiaľ a kam sa umiestni pozadie.

Priehľadnosť

opacity: 0 až 1 !!! pozor, platí pre celý element nielen pre jeho pozadie, ale aj text.

Prechody

Na vytvorenie prechodu farieb už môžeme použiť CSS.

```
background: linear-gradient(45deg, lightgreen, darkgreen);
```

45 deg je uhol, nepovinné.

```
.box-2 {
```

```
background: radial-gradient(ellipse 50px 30px, lightgreen, darkgreen);
```

```
}
```

ellipse alebo circle je tvar, nepovinné.

Dá sa určiť stred prechodu

```
.box-3 {
```

```
background: radial-gradient(at top left, lightgreen, darkgreen);
```

```
}
```

Najjednoduchšie použiť online generátor

<https://cssgenerator.org>

<http://www.cssmatic.com/>

Na tejto stránke sú aj ostatné ako zaoblenie rohov, tieň atd.

Zaoblenie rohov elementu border radius

```
border-radius: 10px; border-radius: 15% 15% 0 0;
```

Dá sa nastaviť aj samostatne:

```
border-top-left-radius: 4em;
```

```
border-top-right-radius: 4em;
```

```
border-bottom-right-radius: 4em;
```

```
border-bottom-left-radius: 4em;
```

TIP: zo štvorca urobíme kruh s border-radius:50%;

Tiene box-shadow



`box-shadow: 5px 5px 10px rgba(0, 0, 0, .5);`
kde hodnoty znamenajú: posun doprava, dole, rozmazanie a farba tieňa. Ak pred farbou dáme slovo INSET znamená vnútorný tieň.

Môžeme ich vrstviť na seba, prvý je ten najviac hore.

`box-shadow: 5px 5px 10px 10px rgba(0, 0, 0, .5),`

`inset 5px 5px 10px 10px rgba(0, 0, 0, .5);`

Tiene textu text-shadow

```
.tien{
  text-shadow: 1px 2px 5px #444444;
}
```

rovnako ako u box-shadow

Použitie vlastných fontov @font-face

```
@font-face {
  font-family: _nazov_rodiny_;
  src: url(_cesta_k_suboru_s_pismom_) format(_format_suboru_);
}
h1 {
  font-family: _nazov_rodiny_;
}
```

Viac stĺpcový text

```
column-width: _sirka_stlpca_;
column-count: _pocet_stlpcov_;
column-gap: _sirka_odsadenia_mezi_;
column-rule: _vlastnosti_ciary_mezi_;
```

Stačí zadať počet.

RGBA farba

```
color: rgba(255, 0, 0, 0.5);
```

Text Overflow

je označenie textu, ktorý sa nezmestil do svojich rozmerov.

```
text-overflow: ellipsis;
overflow: hidden;
white-space: nowrap;
```

This is some long text that ...

Animácie = zmeny stavu = transitions

Umožňujú jednoduché animácie Bez javascriptu. Treba zadať:

1. Vlastnosť, ktorá sa mení = animuje
`transition-property: width, border, left;`
2. Ako dlho trvá zmena v sekundách, prípadne oneskorenie `transition-delay`
`transition-duration: 0.8s;`
3. Spôsob ako sa to deje = rovnomerne, zrýchlene atd.
`transition-timing-function: linear, ease-in, ease-out, ease-in-out, step-start, step-end,`

Pozn. transform-origin: bod, okolo ktorého sa zmena deje.

Môžeme použiť aj keyframes a určiť kedy sa aká zmena udeje:

```
@keyframes nase_animace {
  0%{vlastnost: hodnota;}
  20%{vlastnost: hodnota;}
  40%{vlastnost: hodnota;} 60%{vlastnost: hodnota;}
  80%{vlastnost: hodnota;}
  100%{vlastnost: hodnota;}
}
```

Použiť sa dajú 2D a 3D transformácie.

2D sú

Zväčšenie

```
.scale {
    transform: scale(1.5);
}
```

Skosenie

```
.skew {
    transform: skew(-15deg);
}
```

Posun

```
.translate {
    transform: translate(0, 50%);
}
```

Otočenie

```
.rotate {
    transform: rotate(-15deg);
}
```

Kombinácie:

```
.skoszvacis {
    transform: scale(1.5) skew(-15deg);
}
.scale-2 {
    transform: scale(1.5); transform-origin: top left;
}
```

3D sú hlavne rotácia = okolo osi XYZ

čiže rotateX(15deg)

Animácie

= prechodu medzi stavmi, napríklad pri polohe myši nad odkazom – stav hover.

```
.box {
    background: green;
    transition: 300ms;
}
.box:hover {
    background: blue;
}
```

Udáva sa zmena aká sa má vykonať a čas v sekundách, prípadne typ zrýchľovania.

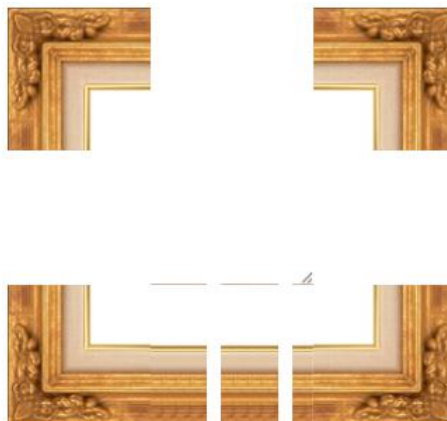
Border zložený z obrázkov

border-image:

- border-top-image
- border-right-image
- border-bottom-image
- border-left-image
- border-image-source
- border-image-slice
- border-image-width
- border-image-outset
- border-image-repeat

border-corner-image:

- border-top-left-image
- border-top-right-image
- border-bottom-left-image
- border-bottom-right-image



Skrátene: border-image: zdroj obrázku, rez, šírka, začiatok, opakovanie;

Rez ani počiatok a nezadáva.

Postup pri tvorbe CSS stránky

V head a v ňom:

1. link na externý font, cudzie fonty: nedávaj všetky rezy, veľmi to spomalí stránku.
2. normalizácia CSS

3. moje štýly lebo v nich je odkaz na font tak musí byť nižšie
4. jquery = javascript knižnica, sú 2 verzie.
5. ostatné ako
 - o favicon.ico stačí aby bol v roote = kde je html. Pomôže favicomatic.com .
 - o Pre odkaz na plochu apple zariadenia daj do head
`<link rel="apple-touch-icon" url= />`
 - o metatagy
`<meta name="description" content="tu napis co ukaze google ked najde tvoju stranku" />`
 - o ostatné meta

Príklad:

```
<link
rel="stylesheet"href="http://fonts.googleapis.com/css?family=Source+Sans+Pro:400,600">
<link rel="stylesheet" href="font-awesome.css">
<link rel="stylesheet" href="normalize.css">
<link rel="stylesheet" href="style.css">
```

fontawesome.io má v sebe kopu fontov aj s ikonkami, ktoré môžeme použiť miesto niektorých obrázkov.

Odpadne problém so zväčšovaním obrázka. Sú to vektory. Do head dáme odkaz aj na to.

V CSS si vytvoríte s komentami bloky ako GENERAL, KONTAKT, NAV, PATA, OBSAH atd...

V general budú:

body, písma, triedy pre float a clear.

Použitie viacerých CSS modulov pomocou importu

Ak chceme rôzne štýly zaradiť do samostatných cez súbory podľa ich použitia (tabuľky, formuláre...), použijeme príkaz import v hlavnom CSS súbore:

```
@import url('nulovanie.css');
@import url('stranka.css');
@import url('form.css');
@import url('tabulka.css');
```

Zaujímavé stránky

Na control kodu:

codepen.io

sbin.com

wufoo.com

meyerweb.com/eric/tools/css/reset/ na reset css

<https://necolas.github.io/normalize.css/> tiež

<https://dojotoolkit.org/> dojo toolkit

<http://prototypejs.org>

Pikošky

Ak je pozadie na celú šírku a obsah nie:

jeden div bude na celú šírku, a v ňom druhý vycentrovaný s margin.

html5 používa elements strong i b em takto:

em strong = šikmé hrubé za účelom dôležitosti textu

i b = šikmé hrubé

Format času dáme s

```
<time datetime="..."> datum </time>
```

bodku lebo znak pred - za niečo dáme v css:

```
time:before {content:' . '};
```

Úvodné pravidlá CSS

```
:root {  
  --prva: #047aed;  
  --druha: #1c3fa8;  
  --dark: #002240;  
  --light: #f4f4f4;  
  --padding-zaklad: 1rem;  
}  
* {  
  box-sizing: border-box;  
  padding: 0;  
  margin: 0;  
}  
body {  
  font-family: 'Verdana', sans-serif;  
  font-size: 100%;  
  line-height: 1.6;  
}  
ul {  
  list-style-type: none;  
}  
a {  
  text-decoration: none;  
}
```

CSS/flexmat.html

```
<!DOCTYPE html>
<html>
  <head>
    <title>Display FLEX</title>
    <style>
      * {
        box-sizing: border-box;
      }
      html, body {
        background-color: #333;
        height: 100%;
        margin: 0;
      }
      .box {
        background-color: #379AD6;
        color: #222;
        border: 1px solid black;
        margin: 5px;
        padding: 5px;
      }

      .kontainer {      vlastnosti otca
        background-color: magenta;
        padding: 1rem;
        width: 100%;
        display: flex;      vytvorenie flexu
        flex-wrap: wrap;    povolenie zalomenia do noveho riadka
        height: 400px;
      }

      .box-1 {      vlastnosti deti
        flex-shrink: 0;    zakazanie zmensovania
        order: 3;          poradie - cim vyssie cislo tym je neskor v rade
      }

      .box-2 {
        flex-grow: 1;      povolenie rastu
        flex-basis: 0;      minimalna sirka dietata, ak je 0 su vsetky rovnake
        order: 1;
      }

      .box-3 {
        flex-grow: 1;
        flex-basis: 0;
        order: 0;
      }

      .kontainer2 {
        display: flex;
        flex-wrap: nowrap;
        flex-direction: column;      zmena smeru na smer zhora dole
        background-color: magenta;
        padding: 1rem;
        width: 100%;
      }

      .box-4 {
```

```

    flex-grow: 1;
}
.centrovanie {
    padding: 1rem;
    margin: auto;
    width: 80%;
    height: 400px;
    display: flex;
    justify-content: center;    vycentrovanie
    align-items: center;      na sirku aj vysku
}
</style>
</head>
<body>
    <div class="kontainer">
        <div class="box box-1"><p>box je 1</p></div>
        <div class="box box-2"><p>box je 2</p></div>
        <div class="box box-3"><p>box je 3</p></div>
    </div>
    <hr>
    <div class="kontainer2">
        <div class="box box-4">Lorem ipsum dolor sit, amet consectetur adipisicing
elit. Impedit error, accusantium fugit labore expedita tempora. Quas harum
accusantium atque placeat! Sapiente vitae corporis explicabo? Magni?</div>
        <div class="box box-4">Lorem ipsum dolor sit amet consectetur adipisicing
elit. Perspiciatis inventore fuga ducimus nulla repellat? Magni accusamus totam quas
dolore debitis nostrum, nemo doloremque esse atque culpa facilis nam perspiciatis
nihil quidem, ex blanditiis earum velit impedit sequi laudantium. Id nobis pariatur
reprehenderit deserunt iusto consectetur molestias iure corporis mollitia vero
delectus ducimus sit, asperiores exercitationem omnis perspiciatis ab distinctio sunt
quod soluta odit et voluptates ipsam! Voluptas cupiditate soluta labore dolore sunt
eos blanditiis minima tempora deleniti fuga voluptate rerum at, impedit
exercitationem dolorem tempore! Et ipsa eius nemo rem quia consequatur tempora quod,
ipsam repudiandae accusantium nihil iure eaque.</div>
        <div class="box box-4">Lorem ipsum dolor sit amet consectetur adipisicing
elit. Dolorem ea repellendus necessitatibus ipsam ducimus obcaecati temporibus libero
harum quo impedit ullam quam eos animi velit aperiam soluta esse quis tempora, illo,
delectus consequatur et! Explicabo suscipit nulla enim recusandae libero nostrum
possimus molestias ea porro, saepe sed voluptatibus! Provident dolores inventore
dolor neque. Dolorem, porro.</div>
    </div>
<hr>
    <div class="centrovanie box">
        <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Aperiam
laudantium quos porro officiis dolores? Ullam mollitia consequuntur accusantium?
Facilis unde et incidunt minus nesciunt, tenetur quis? Dignissimos ratione quia eum
iusto soluta excepturi quae rerum et. Cum illo at numquam impedit quae, voluptatem
consectetur modi natus blanditiis eos dignissimos odio? Hic similique porro aliquam
molestias!</p>
    </div>
</body>
</html>

```

CSS/gridmat.html

```
<!DOCTYPE html>
<html>
  <head>
    <title>Display GRID</title>
    <style>
      * {
        box-sizing: border-box;
      }
      html, body {
        background-color: #333;
        height: 100%;
        margin: 0;
      }
      .box {
        background-color: #379AD6;
        color: #222;
        padding: 10px;
      }

      .kontainer {      vlastnosti pre otca
        padding: 1rem;
        width: 100%;
        height: 100%;

        display: grid;      vytvorenie mriezky
        grid-template-columns: repeat(3, 150px);      vytvorenie 3 rovnako velkych stlpcov
        grid-auto-rows: minmax(150px, auto);      vytvorenie riadkov ked neviem kolko ich bude
        grid-gap: 20px;      medzera medzi detmi
        justify-content: center;      centrovanie na
        align-content: center;      vysku aj sirku
      }

      .box-1 {
        grid-column: span 3;      3 bunky vedla seba spojene do jednej
      }
      .box-2 {
        grid-row: span 2;      2 bunky pod sebou spojene do jednej
      }
      .box-3 {
        grid-row: span 2;
        grid-column: span 2;      2 bunky vedla seba a 2 pod sebou spojene do jednej
      }
      .box-4 {
        grid-column: 1 / -1;      box 4 zacina na stlpci 1 a konci poslednym = je cez celu sirku okna
        grid-row: 4 / 5;      box 4 zacina na riadku 4 a konci na piatom
      }

      .otec {
        grid-template-areas:      iny sposob umiestnenia deti:
          "hore hore hore"      po riadkoch zlava
          "bok main main"      doprava dam mena deti
          "bok main main"      ktore som pomenoval
          "dole dole dole";      svojimi nazvami
      }
      .a1 {
```

```

    grid-area:hore;
}
.a2 {
    grid-area: bok;   pomenoval som box a2 menom bok
}
.a3 {
    grid-area: main;
}
.a4 {
    grid-area: dole;
}
</style>
</head>
<body>
    <div class="kontainer">
        <div class="box box-1"><p>Lorem ipsum dolor, sit amet consectetur adipisicing
elit. Quis praesentium eum, ex explicabo cupiditate placeat minus mollitia doloremque
possimus saepe enim tenetur atque accusamus laudantium pariatur, impedit voluptate
numquam assumenda reprehenderit! Quisquam, natus.</p></div>
        <div class="box box-2"><p>Lorem ipsum dolor, sit amet consectetur adipisicing
elit. Quis praesentium eum, ex explicabo cupiditate placeat minus mollitia doloremque
possimus saepe enim tenetur atque accusamus laudantium pariatur, impedit voluptate
numquam assumenda reprehenderit! Quisquam, natus.</p></div>
        <div class="box box-3"><p>Lorem ipsum dolor, sit amet consectetur adipisicing
elit. Quis praesentium eum, ex explicabo cupiditate placeat minus mollitia doloremque
possimus saepe enim tenetur atque accusamus laudantium pariatur, impedit voluptate
numquam assumenda reprehenderit! Quisquam, natus.</p></div>
        <div class="box box-4"><p>Lorem ipsum dolor, sit amet consectetur adipisicing
elit. Quis praesentium eum, ex explicabo cupiditate placeat minus mollitia doloremque
possimus saepe enim tenetur atque accusamus laudantium pariatur, impedit voluptate
numquam assumenda reprehenderit! Quisquam, natus.</p></div>
    </div>
    <hr>
    <div class="kontainer otec">
        <div class="box a1">
            <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Quas ducimus ut,
est quae enim ipsum ipsam numquam beatae molestias nesciunt sed magni nam consectetur
eligendi deleniti blanditiis assumenda iure dolor. Quasi, eos sit?</p>
        </div>
        <div class="box a2">
            <p>Exercitationem ex placeat temporibus perferendis quaerat quos est ducimus
tempore, illum consequuntur velit perspiciatis ipsa iusto incidunt ad cum officiis
quia reprehenderit voluptas nobis debitis nam pariatur vitae. Rem cumque nam
repudiandae adipisci?</p>
        </div>
        <div class="box a3">
            <p>Unde magni consequuntur ipsum ad molestias, temporibus tempora distinctio
harum repudiandae placeat, mollitia totam repellat. Voluptatibus nam, animi tempore
sapiente incidunt voluptas modi explicabo veniam qui architecto alias nisi laboriosam
optio error veritatis.</p>
        </div>
        <div class="box a4">
            <p>Voluptatem est ex cum dolorum. Itaque harum tempora rerum illo? Magnam
dolorem eum in doloremque culpa, quam obcaecati ratione dolore, optio velit sunt quis
commodi molestias maxime expedita consequuntur, nostrum adipisci. Eveniet, sint.</p>
        </div>
    </div>
</body>
</html>

```