

Text sa vkladá do tela stránky alebo do bunky ako obyčajný, nadpis alebo odkaz.
Obyčajný text sa len napíše vtedy, ak nám stačia jeho vlastnosti zadané v prehliadači.
Ak nie, treba zadeinovať jeho formát:

písmo

 toto je text
znamená písmo Arial alebo Verdana veľkosti 5 a červené.

ostatné vlastnosti zmeny písma robia elementy:

<basefont size=4>	základná veľkosť textu na stránke	1 - 7
<big> .. </big>	zväčšenie písma o 1	
<small> .. </small>	zmenšenie písma o 1	
<body text=blue>	základná farba textu na stránke modrá, ! za body!!!	
 tučný 	text bude tučný	
<i> šikmý </i>	text bude kurzíva	
<u> podč </u>	text bude podčiarknutý	
<strike> preč </strike>	text bude prečiarknutý	
^{hore}	text bude horný index - na druhú	
_{dole}	text bude dolný index	
<p align=left>	text bude zarovnaný doľava	

Nadpisy

sa robia elementom

<h1> Nadpis </h1>	pričom za h môže byť 1 až 7. 1 je hlavný nadpis.	1 - 7
-------------------	--	-------

Odstavec

sa robí elementom

<p> .. </p>	Odstavec je viditeľne oddelený od ďalšieho.	
-------------	---	--

Zalomenie riadku

sa robí elementom

 	za textom, kde chceme riadok zalomiť.	
-------	---------------------------------------	--

Zoznamy - číslované

sa tvoria elementom

 ... 	medzi ktoré sa dávajú jednotlivé položky.	
----------------	---	--

 prvý

 druhý

 tretí

pričom ak chceme nie čísla, ale ABC abc I.II.III použijeme za elementom ol

<ol type="A">	pre zoznam A, B, C...	
---------------	-----------------------	--

<ol type="a">	pre zoznam a, b, c...	
---------------	-----------------------	--

<ol type="I">	pre zoznam I. II. III...	
---------------	--------------------------	--

<ol type="i">	pre zoznam i. ii. iii....	
---------------	---------------------------	--

odrážkový zoznam

sa tvorí elementom

 ... 	medzi ktoré sa dávajú jednotlivé položky.	
----------------	---	--

 prvý

 druhý

ak chceme iný typ odrážok, použijeme za ul:

<ul type=disk>	plná bodka	
----------------	------------	--

<ul type=circle>	prázdna bodka	
------------------	---------------	--

<ul type=square>	štvorec	
------------------	---------	--

Odkazy - typy

1. Odkaz na

- absolútny *napr. na www.pokec.sk*

`toto je odkaz na pokec`

- relatívny *na stránku na "mojom" - tom istom serveri, ako je stránka s odkazom*

`toto je odkaz na stránku moja`

ak je stránka v podadresári novy, bude za href="novy/moja.htm"

ak je odkaz o adresár vyššie, bude href="../moja.htm"

2. Odkaz Vnútorý *čiže na iné miesto v tej istej stránke*

`<p>toto je odkaz na novinky na tejto stranke</p>`

.... tu pokračuje stránka ...

`<h1>Novinky</h1>`

takýto odkaz po kliknutí na horný text hodí na text Novinky (s menom news)

2. Odkaz na mail

po kliknutí naň sa otvorí poštový program na písanie nového mailu s adresou

`pošlite mi mail`

s tým, že ak tam je aj subject, je predpísaný predmet mailu.

Možnosti

za odkazom okrem alternatívneho textu môže byť zadané aj:

accesskey=w ak užívateľ stlačí Alt a W prejde na tento odkaz

tabindex=2 pre užívateľov bez myši ovládanie po odkazoch pomocou TAB - číslo stlačení

target=_blank odkaz sa otvorí v novom okne

target=_self odkaz sa otvorí v tom istom okne

Vkladanie obrázkov

robí sa elementom ``

```

```

alt je alternatívny text, width a height je šírka a výška v pixeloch (nemusi byť rovnaká ako obrázku), okraj je 0. Elementy za obrázkom sú nepovinné.

Odkaz z obrázku sa urobí tak, že pred neho sa dá začiatok odkazu, t.j.

```
<a href="detail.html">
```

```

```

```
</a>
```

V tomto elemente sa dajú použiť aj ostatné vlastnosti, ako je okraj, zarovnanie, vzdialenosť od okraja a podobne.

border / align / vspace hspace

Vloženie vodorovnej čiary

často v minulosti, elementom:

```
<hr size="3" width="60%" align="right" noshade color="green"><p>
```

je zelená čiara cez 60% strany vpravo a bez tieňa.

Vkladanie multimédií

Zvukové a videosúbory sa dajú vložiť do stránky ako odkaz, alebo priamo ako prehrávač zvuku alebo videa.

1. Ako odkaz

```
<a href="video/video.avi" title="toto sa zobrazí pri myši nad odkazom">Toto je odkaz na video</a>
```

title je alternatívny text. Podobne aj mp3, wav, mp4, mpeg a podobne.

2. V prehrávači

```
<object classid="clsid:22D6F312-B0F6-11D0-94AB-0080C74C7E95" width="320" height="240">
```

```
<param name="FileName" value="3d.wmv" />
```

```
</object>
```

alebo aj jednoducho

```
<object data="media/video.avi" type="video/avi" />
```

3. Aby sa prehral po načítaní stránky na pozadí

```
<bgsound src="zvuk.wav" loop="5"/bgsound>
```

len wav a midi. Loop je počet opakovaní, infinite je nekonečno.

4. Elementom embed

```
<embed src="zvuk.wav" autostart="true" height="22px" loop="1" align="left"></embed>
```

Autostart je automatické prehratie, loop je opakovanie, align a height sú zarovnanie a výška ovládačov prehrávania.

Dnes sa s výhodou používa programovací jazyk JAVA, a prehrávanie je robené práve pomocou javascriptu.

Vkladanie flashu - animácie

```
<object classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000"
```

```
<param name="movie" value="Mid/year.swf">
```

```
<param name="quality" value="high">
```

```
<embed src="Mid/year.swf" width="250" height="30" align="right" quality="high"
```

```
</object>
```

dôležitá je len šírka, výška zarovnanie, cesta a názov flash súboru a kvalita prehratia. Ostatné je najlepšie skopírovať.

Zobrazenie webstránky

```
<object type="text/html" height="100%" width="100%"  
data="http://www.herbie.szm.com">  
</object>
```

Celkom sa pri vkladaní multimédií ešte môžete stretnúť s týmito vlastnosťami a ich hodnotami:

clicktoplay	true/false (či treba kliknúť na prehratie)
displaybackcolor	color (farba za)
displayforecolor	color
showcontrols	true/false (či sú zobrazené ovládače prehratia)
showdisplay	true/false
showtracker	true/false

Tabuľky

sa vytvárajú elementom `<table>` a na konci `</table>`

`<table> .. </table>`

Vlastnosti tabuľky

sa dajú za slovo `table` a upresňujú jej vlastnosti

<code>border</code>	hrúbka rámu tabuľky v pixeloch	<code>border=5</code>
<code>bordercolor</code>	farba okraja	<code>bordercolor=blue</code>
<code>bordercolorlight</code>	farba pravého a spodného okraja	<code>bordercolorlight=yellow</code>
<code>bordercolordark</code>	farba ľavého a horného okraja	<code>bordercolordark=black</code>
<code>bgcolor</code>	farba pozadia tabuľky	<code>bgcolor=orange</code>
<code>background</code>	pozadie ako obrázok, treba zadať cestu k nemu	<code>background=" ../obr/sat.jpg"</code>
<code>width</code>	šírka tabuľky v pixeloch	<code>width=500</code>
	šírka tabuľky v percentách z otvoreného okna prehliadača	<code>width=80%</code>
<code>height</code>	výška tabuľky v pixeloch	<code>height=300</code>
<code>align</code>	zarovnanie tabuľky: naľavo .. stred .. napravo	<code>align=right</code>
<code>valign</code>	zarovnanie tabuľky na výšku: hore .. stred .. dole	<code>valign=center</code>
<code>cellpadding</code>	medzera medzi bunkami tabuľky v pixeloch	<code>cellpadding=5</code>
<code>cellspacing</code>	vzdialenosť obsahu bunky a okraja bunky v pixeloch	<code>cellspacing=2</code>

Nadpis tabuľky

je text ešte nad tabuľkou, dáva sa vždy za `</table>` ešte pred riadky tabuľky

`<caption>`Toto je popisok tabuľky `</caption>`

Vloženie riadku

element `<tr> ... </tr>`

Vloženie bunky do riadku

element `<td> ... </td>`

Nadpisy stĺpcov a riadkov

element `<th> .. </th>`

všade, kde chceme mať hrubým zvýraznený text v bunke (ako

Vlastnosti riadkov a buniek

sa zadávajú rovnako ako u tabuľky hneď za element riadka

nie všetky sú použiteľné

Spájanie buniek

2 vedľa seba

`<td colspan=2>`toto su bunky vedlajsie spojene do jednej`</td>`

2 nad sebou

`<td rowspan=2>`toto su bunky nad spojene do jednej`</td>`

kombinácia = 2 vedľa a 3 nad, čiže 6 buniek do jednej

`<td rowspan=3 colspan=2>`toto je 6 buniek v jednej`</td>`

Vlastnosti prvých 3 stĺpcov

ešte pred prvý riadok tabuľky dáme element `<col />`

`<col span=3 align=left valign=top width=100/>`

Příklad

```
<table width="500" border="1" align="center" cellpadding="1" cellspacing="5">
  <caption>Elektronické publikovanie 1.E </caption>
  <tr>
    <td width="78">Obsah</td>
    <td colspan=2>Linky</td>
  </tr>
  <tr>
    <td>Tabuľky</td>

</tr>
<tr>

</tr>
</table>
```

Formátovanie prvkov stránky

je vlastne určovanie vzhľadu jednotlivých prvkov stránky, ako napr. veľkosť a farba písma, zarovnanie, farba pozadia, obrázok pozadia, obtekanie textu, okraje, vzhľad tabuľky - farby, okraje, medzery a podobne.

Ak v našej stránke nezádáme pravidlá pre písmo a vzhľad stránky, o jej zobrazení rozhodne typ a nastavenie web prehliadača. Teda tú istú stránku môžeme zobrazit' na rôznych počítačoch rôzne. Aby sa to nestalo, je potrebné zadávať vlastnú podobu stránky = formátovanie.

Dá sa urobiť 4 spôsobmi:

1. Priamo v tele stránky na mieste formátovaného prvku - individuálne štýly

už sme sa s ním stretli - pri texte a tabuľkách, napr.:

```
<h1 style="color: gray; font-size: 24px;">Toto je nadpis 1!</h1>
```

nastavujú len ten element, v ktorom sú uvedené. Je veľmi prácne a nevýhodné
Používajú sa IBA ak chceme urobiť výnimku z pravidla, t.j. napr. jeden nadpis

2. Vložením do hlavičky stránky - pred element </head>

Vkladajú sa do hlavičky a majú vplyv len na dokument, v ktorom sú uložené.

Je dobré ich použiť pri príprave stránky a na konci práce ich vystrihnúť a vložiť do prázdneho .css dokumentu, čím vytvoríme najvýhodnejší typ 3.

Sú vždy v hlavičke, a najvýhodnejšie je ich dať pred jej koniec </head>

Začínajú sa <style type="text/css"> a končia </style>

```
<head>
```

```
<title> Práca so štýlmi </title>
```

```
<style type="text/css">   v nasl. riadku sa zadá pozadie stránky na čierne a farba písma na bielu  
body {background-color: black; color:white;}
```

```
h1 {font-size: 24px;}    tu sa zadá veľkosť písma nadpisu 1 na 24 pixelov
```

```
p {font-size: 12px;}    tu sa zadá veľkosť písma na 12 pixelov
```

```
</style>
```

```
</head>
```

3. Pripojením externého súboru (v hlavičke) typu .css v ktorom sú všetky pravidlá formátovania

je najvýhodnejšie, lebo:

1. Akúkoľvek zmenu urobíme rýchlo v prehľadne BEZ otvorenia samotnej stránky
2. Zmena sa prejaví NARAZ vo všetkých stránkach pripojených na tento štýl
3. Samotné stránky sú menšie a tým sa rýchlejšie načítajú
4. Stránky sú prehľadnejšie a ľahšie sa upravujú
5. Súbor .css ostávajú načítané v medzipamäti prehliadača, takže ich stále nanovo nest'ahuje, takže stránky sú rýchlejšie zobrazené.

V hlavičke stránky sa urobí takýto odkaz:

```
<link rel="stylesheet" type="text/css" href="styly.css" media="all" />
```

príčom styly.css je názov nášho súboru so štýlmi. Ten si dajte aký chcete, len musí byť správne uvedený aj v tomto riadku.

4. Importovaním externého súboru .css do stránky

Je to kombinácia bodu 2 a 3:

1. musíte mať súbor so štýlmi typu .css, napr. styly.css
2. v hlavičke potom dáte príkaz

```
<style type="text/css"> @import url(styly.css); </style>
```

3. príkazy súboru styly.css budú potom importované priamo do našej stránky tak, ako keby ste ich tam napísali (priamo do hlavičky, ako v bode 2)

Pravidlá CSS

sú jednoduché. Skladajú sa zo selektorov a aspoň jednej deklarácie. Príklad:

```
h1 {background-color: black; color:white;}
```

h1 je selektor - určuje čoho sa štýl týka (teraz nadpisu 1)

v zátvorkách sú 2 deklarácie. Najprv sa píše vlastnosť (farba pozadia) a za dvojbodku a medzeru hodnota (black). Jednotlivé deklarácie sú oddelené bodkočiarkou a medzerou!! (alebo aj novým riadkom)

Poznámky píšeme:

```
/* poznámka */
```

Vytvorenie vlastnej triedy - vlastného štýlu

Používa sa stále častejšie. Ak chceme vytvoriť vlastné štýly s vlastným názvom, a tie potom ľubovoľne v dokumente použiť, postupujeme takto:

1. V hlavičke zadefinujeme jeho názov a vlastnosti

```
.detailText {  
    názov je detailText  
font:11px arial;    typ a veľkosť písma  
color:#ffffff;     farba  
line-height:16px;  výška riadka  
letter-spacing:.1em;  medzera medzi písmenami  
}
```

2. Priamo v dokumente na mieste, kde chceme štýl použiť dáme class="nazov"

```
<p class="detailText">Dodatky</p>
```


Zadávanie farieb

je možné viacerými spôsobmi, pričom je na vás, ktorý použijete:

1. V hexadecimálnom tvare

keď sa zadá množstvo R, G, B v šestnástkovej sústave, vždy je ale predtým krížik

color: #ffcc80;

2. V skrátrenom hexadecimálnom tvare

dá sa použiť len ak dvojice znakov RRGGBB sú rovnaké

color: #fc8

je vlastne farba ffcc88

3. Zadaním skutočných hodnôt RGB

bez mriežky sa napíše rgb a v zátvorke množstvá v poradí RGB od 0 po 255

{color: rgb(101,22,122);

4. Percentami RGB

podobne ako 3 len v zátvorke sú percentá od 0 po 100

color: rgb(10%,22%,99%);

5. Názvom farby

white	lime
silver	olive
grey	green
black	teal
yellow	cyan
magenta	blue
red	navy
maroon	purple

color: purple;

5. Pehľadná farba background-color: transparent;

Zadávanie pozadia

1. Farba

Farbu pozadia môžeme nastaviť úplne inú pre každý element, ako je nadpis, tabuľka, telo, odkaz, odstavec atd. Pre celý dokument je to selektor body. Syntax je background-color:

body: {background-color: #999};

a: {background-color: #fff};

table: {background-color: magenta};

2. Obrázok pozadia

Platí rovnako, ako pre farbu (každý element môže mať vlastné pozadie). Navyše však môžeme zadať veľa parametrov pre obrázok, ako či sa má opakovať a v ktorom smere, ako má byť umiestnený (zarovnanie), priehľadnosť atd. Syntax je background-image: Hodnota je url(cesta a názov obrázku), pričom ak je obrázok v adresári, používame lomítko / (napr. url(obrazky/pozadie.jpg). Ak chceme ísť o adresár vyššie, použijeme ../obr.jpg. Ak nič nezádáme, obrázok sa opakuje po stránke.

body: {background-image: url(pozadie.jpg)};

opakovanie *background-repeat upresní či a v ktorom smere sa má obrázok opakovať. Hodnoty sú repeat-x, repeat-y, no repeat.*

body: {background-image: url(pozadie.jpg); background-repeat: repeat-y;}

umiestnenie *background-position mi určí polohu obrázka pozadia a to v percentách (0% 0% je ľavý horný roh), v pixeloch, alebo pomocou slov. POZOR!!! Pred týmto musíte vždy nastaviť opakovanie na no-repeat!!!*

body: {background-image: url(pozadie.jpg); background-repeat: no-repeat; background-position: center bottom;}

iné príklady:

background-position: 0% 10%;

background-position: 20px 20px;

background-position: right;

background-position: bottom right;

uchytenie na posun *Pomocou background-attachment: scroll (/fixed) zabezpečíme, že sa obrázok pri posôvaní stránky dole bude / nebude posúvať. ! Platí len pre no-repeat!*

body: {background-image: url(pozadie.jpg); background-repeat: no-repeat; background-attachment: scroll;}

Zadávanie pozadia - súhrn

Sú 2 spôsoby, ako v css súhrnne zadať vlastnosti:

1. Plný

každá vlastnosť sa píše do samostatného riadka, na konci je ;

```
body {  
    background-color: white;  
    background-image: url(images/lemon.gif);  
    background-repeat: no-repeat;  
    background-attachment: scroll;  
    background-position: right bottom;  
}
```

2. Skrátený

za sebou, pričom sa vynechajú ; ako aj kľúčové slovo background je len raz

```
body {background: white url(images/lemon.gif) no-repeat scroll right bottom;}
```

Samozrejme takto zadáme aj tabuľku, odkaz, odstavec, text, odrážky - miesto body je table, a, p atd.

Typy písma

Pri zadávaní typov písma (fontov) treba použiť hlavne tie, u ktorých je predpoklad, že ich všetci používatelia budú mať nainštalované, teda najlepšie tie vstavané do najčastejších operačných systémoch. Je 5 rodín písom:

1. serif

pätkové písma vhodné na všetko, ako Times a Georgia

2. sans-serif

bezpätkové písma viac zaoblené vhodné na všetko, okrem malého textu a kurzívy (šikmé). Sú to napr. Arial, Helvetica a Verdana.

3. Monospace

všetky písmená majú rovnakú šírku - Courier. Väčšinou na ukážky kódu.

4. Fantasy

ozdobné písma, možno pre nadpisy, napr. Western.

5. Cursive

kurzívy, teda ručne písané typy, používané zriedka, ako Lucida Handwriting.

V praxi typ písma

Pre požadovaný element sa zadá buď názov fontu, alebo typ rodiny (body 1 až 5).

body {font-family: Arial, Helvetica, Verdana, sans-serif;}

znamená, že v tele bude použité Arial, ak nie je nainštalovaný tak Helvetica, ak ani tá - Verdana, a ak ani jeden, tak klasický prednastavený typ sans-serif. Nie je nutné používať toľko typov, ale je to výhodné.

h1, h2 {font-family: Georgia, serif;}

nadpisy 1. a 2. úrovne majú písmo Georgia, alebo serif.

Veľkosť písmen

Môžeme použiť jeden z 3 postupov:

1. Absolútne

Zadáme priamo veľkosť v:

22pt

bodoch

11mm / 2cm

milimetroch a centimetroch

1in

palcoch

Výhodnejšie pre obrazovku je použiť pixely a em:

12px

2em

2. Relatívny spôsob

Kľúčové slová sú xx-small, x-small, small, medium, large, x-large, xx-large.

p {font-size: medium;}

znamená, že odstavec bude mať stredne veľké písmo.

! Tento spôsob je relatívny, t.j. treba zdefinovať hlavnú veľkosť prvým spôsobom, najlepšie v elemente body.

3. V percentách

je tiež relatívny, takže nadradený element musí byť nastavený absolútne!

body {font-size: 12px;}

h1 {font-size: 150%;} nadpis bude o polku väčší, než telo.

Šírka, štýl, farba, zarovnanie, dekorácia a odsadenie textu

1. Šírka *definuje sa pomocou font-weight. Hodnoty sú od 100 (veľmi tenké) po 900 (najtučnejšie). Použiť môžeme aj slová normal (400), bold (700).*

h1 {font-size: 150%; font-weight: bold;}

h2 {font-size: 130%; font-weight: lighter;}

Ak použijeme bolder alebo lighter, zmení sa písmo oproti predchádzajúcemu (stučnie alebo schudne).

2. Štýl *Pomocou font-style. Hodnoty sú normal, italic (šikmé).*

body {font-size: 12px; font-style: italic;}

3. Farba *Jednoducho slovom color a hodnoty sú farby ľubovoľným spôsobom.*

color: black;

4. Zarovnanie textu

pomocou text-align a hodnotami left, center, right, justify (obe strany sú zarovnané).

body {font-size: 12px; font-style: italic; color: black; text-align: center;}

5. Dekorácia

pomocou text-decorate a hodnoty sú:

none *používa sa na odstránenie podčiarknutia odkazov*

underline *podčiarknutie*

overline *čiara nad text*

line-through *prečierknutie*

blink *blikajúci text*

6. Odsadenie prvého riadku textu

Ak chceme, aby prvý riadok odstavca bol posunutý, urobí to text-indent s hodnotami v pixeloch alebo percentách.

p {text-indent: 45px;}

Varianty písma, výška riadka, medzery medzi písmenami a slovami

1. Prevod textu a varianty písma

Prevod textu je text-transform s hodnotami:

capitalize *prvé písmeno každého slova je veľké*

uppercase *všetky písmená veľké*

lowercase *všetky písmená malé*

none *žiadne*

Varianta písma je font-variant a hodnoty sú:

small-caps *kombinácia uppercase a capitalize, všetko veľké a prvé písmená ešte väčšie*

normal *je normálne písmo*

2. Výška riadka

pomocou line-height s hodnotami v pixeloch, percentách z výšky písma, alebo len číslom - násobkom výšky písma.

<p style="line-height: 2;"> text Na viac riadkov. </p>

znamená, že ak text zadaný vyššie bol napr. 12, vzdialenosť riadkov bude 2x12=24.

<p style="line-height: 80%;"> text Na viac riadkov. </p>

znamená, že ak text zadaný vyššie bol napr. 12, vzdialenosť riadkov bude 80% z 12, takže 10.

3. Medzery medzi písmenami a slovami

Písmená letter-spacing a slová word-spacing. Hodnoty sú pixeloch alebo em.

<p style="letter-spacing: 10px; word-spacing: 2em;"> text text. </p>

4. Prvé písmená a riadky textu

Ak chceme, aby prvé písmeno odstavca alebo prvý riadok mal samostatný vzhľad, použijeme :first-letter alebo "first-line.

p:first-line {font-weight: bold; color: #333;}

p:first-letter {font-weight: italic; color: #999;}

týmito 2 riadkami zabezpečíme, že prvé písmená odstavca (p na začiatku riadka) budú šikmé s farbou 999999 a celé prvé riadky (okrem 1. písmena) budú tučné a s farbou 333333.

Písmo súhrnne

Spoločnou vlastnosťou písma je font, takže v skrátenej verzii bubde napr.:

p {font: italic small-caps bold 14/15px Arial, sans-serif;}

Stavy odkazov

Sú 4 a z nich vychádzajú aj selektory:

a:link *pred jeho aktivovaním*
a:visited *keď už bol aspoň raz použitý*
a:hover *keď je nad ním myš*
a:active *pri kepnutí na neho*

!!! Aby funfovali správne musia byť v tomto poradí!!!

Príklad pre odkazy v css:

```
a {color: orange; text-decoration: none;}  
a:link {color: orange;}  
a:visited {color: yellow;}  
a:hover {color: fuchsia; font-style: italic; backgroundcolor: white;}  
a :active {color: orange;}
```

! V prvom riadku sme odstránili podčiarknutie odkazov!!!

Ak chceme mať v stránke viac typov odkazov, zadefinujeme základný (príklad hore), a okrem neho vytvoríme novú triedu (aj viac) s naším názvom (tu je názov new), a v stránke ju použijeme kde chceme pomocou class="new". Za každé a dáme bodku a meno triedy.

Takže okrem horeuvedených a bude v css aj:

```
a.new {color: blue;}  
a.new:link {color: blue;}  
a.new:visited {color: red;}  
a.new:hover {color: black; backgroundcolor: white;}  
a:active {color: red;}
```

a na mieste stránky, kde chcem použiť tento štýl odkazu, bude

```
<a class="new" href="http://www.pokec.sk/>pokec</a>
```

t.j. Pred odkaz href dáme class= a meno triedy. Dá sa viackrát.

Zoznamy

1. Číslované

Zadáva sa len typ číslovania rímskych číslic (malé lower-roman a veľké upper-roman), číslovanie od nuly - decimal-leading-zero. To všetko pomocou list-style-type:

```
ol {list-style-type: lower-roman;}
```

2. Nečíslované

Zadáva sa len typ odrážky disc, circle, square pomocou list-style-type:

```
ul {list-style-type: circle;}
```

!!! Pre oba typy môžeme zadať, že miesto čísla alebo odrážky bude náš vyrobený obrázok, napr cislo.jpg, z adresára pcs. Potom pomocou list-style mu dáme cestu:

```
ul {list-style-type-image: url(pcs/cislo.jpg) inside;}
```

inside znamená, že odrážka je vnorená do zoznamu (opak je outside).

Ak použijeme list-style: none môžeme vytvoriť menu = žiadne odrážky.

Vertikálne menu

Ak chceme urobiť ponuku odkazov pod sebou, vytvoríme vlastne nečíslovaný zoznam s

```
ul #mojemenu {list-style-type: none; width: 100px; background-image: url(ja.gig); border: 2px orange;}
```

```
#mojemenu li {border: bottom každý odkaz má horný okraj}
```

```
#mojemenu a {color: orange; display: block; text-decoration: none;}
```

```
#mojemenu a:link {color: orange;}
```

```
#mojemenu a:visited {color: yellow;}
```

```
#mojemenu a:hover {color: fuchsia; font-style: italic; background-image: url(jojo.gif);}
```

```
#mojemenu a :active {color: orange;}
```

V tretom riadku je display: block, čo zabezpečí, že odkazy budú v bloku pod sebou.

Horizontálne menu

Urobí sa podobne, len do li dáme display: inline, takže:

```
ul #mojemenu {background-image: url(ja.gig);}
```

```
#mojemenu li {display: inline; list-style-type: none;}
```

```
#mojemenu a {color: orange; display: block; text-decoration: none;}
```

```
#mojemenu .....
```

!!! Pozor !!! Takto sme vytvorili jedno menu, preto sme ho pomenovali za krížikom mojemenu, a v stránke ho potom začneme vytvárať pomocou

```
<ul id="mojemenu">
```

```
a tu už budú jednotlivé odkazy pomocou li ...
```

Okraje a rámčeky

Pre každý element (telo, nadpis, odstavce atd) sa dajú nastaviť:

1. Vonkajší okraj

- a) pre body udáva v pixeloch vzdialenosť elementu od okrajov okna prehliadača
b) pre ostatné udáva v pixeloch vzdialenosť elementu od predchádzajúceho - vyššieho elementu (nadpis h2 od h1, odstavce od predchádzajúceho atd)

Zadáva sa slovom margin s pomlčkou a polohou (top, right, bottom, left):

```
body {margin-top: 30px; margin-right: 30px; margin-bottom: 30px; margin-left: 130px;}
```

Alebo ešte lepšie naraz v presnom poradí horný, pravý, spodný a ľavý. Ak zadáme len dva, sú to tie prvé - horný a ľavý, ostatné sa zrkadlia (pravý=ľavý):

```
body {margin: 30px 30px 30px 130px;}
```

```
body {margin: 30px 20px;}
```

```
p {margin: 30px;}
```

ak sú všetky rovnaké, stačí aj jeden, za nulou netreba dať px a dajú sa aj v percentách:

```
h1 {margin: 20%;}
```

2. Rámik

Udáva aký rámik je okolo elementu. Nastavuje sa šírka, štýl a farba každej strany rámčeka. Slovo je border.

- a) šírka strany - width *udáva sa v pixeloch alebo slovom. Slová sú thin, medium a thick.*

```
border-top-width: 2px; border-right-width: thick;
```

- b) štýl rámčeka - style *udáva sa slovom: dotted (bodky), dashed (čiarky), solid (čiara), double (2 čiary), inset (prepadnutý), outset (vystúpený), hidden (skrytý), none (bez).*

```
border-right-style: dotted;
```

- c) farba okraja *color - klasicky*

```
border-right-color: #808080;
```

Zjednodušený zápis je pre stranu okraja vždy v poradí a,b,c takto:

```
body {border-top: 1px dotted red;border-left: .....;}
```

Ak použijeme IBA slovo border, nastava sa tak všetky strany rovnako:

```
h1 {border: 1px dotted red;}
```

3. Vnútorý okraj

Umožňuje nastaviť priestor medzi rámčekom a obsahom. Zadáva sa tiež pre jednotlivé strany, ako 1 - vonkajší - v px alebo percentách.

Zadáva sa slovom padding v rovnakom poradí:

```
p {border: thin solid orange; padding-top: 5px; padding-right: 2px;}
```

alebo skrátene

```
p {border: thin solid orange; padding: 5px 2px 12px 2px;}
```


Nastavenie pozície objektov a vrstvy

Ak nie je zadané inak, je stránka vykresľovaná prehliadačom v tom poradí a s takými parametrami, ako je napísaná. Záleží aj o typu prehliadača ako bude potom vyzerat'. Takého jednoduché stránky sa pri zmene veľkosti okna automaticky preusporiada. Ak ale chceme, aby sa stránky správala inak, musíme použiť pozicionovanie.

Vrstvy - kontajnery

Ak chceme využiť silu pozicionovania, používajme vrstvy / kontajnery. Vrstva sa vytvorí na mieste štýlov (hlavička alebo samostatný súbor css) pomocou znaku # a jej mena, za ktorým udáme polohu, poradie, obtekanie a iní vlastnosti vrstvy.

Pr:

```
#meno1 {position: fixed; left: 0px; top: 0px; background: #ccc; width=100%; border: 1px red; padding-left: 20px; z-index: 2;}
```

toto je vrstva s menom meno1 s vlastnosťami ako poloha, pozadie, šírka, okraj, obtekanie

```
<div id="meno1"> ..... </div>
```

Môžu byť samostatné = za sebou plynúce, ako stránka, ale aj jeden v druhom. To bude vyzerat' takto - v hlavičke:

```
#vrstva1 { ..... }
```

```
#vrstva2 { ..... }
```

A potom na stránke v tele

```
<div id="vrstva1"><div id="vrstva2"> ..... </div> </div>
```

Druhy pozícií

1. Absolútne

position: absolute

umiestni vrstvu do presne určených súradníc zľava a zhora a nadradí ju nad normálny beh stránky. Umiestňuje voči nadradenému elementu (väčšinou html = okraj stránky), ale ak sú vnorené, tak potom poloha vnútorného voči vonkajšiemu.

2. Relatívne

position: relative

umiestni vrstvu do normálneho poradia stránky s tým, že vzdialenosti sú v pixeloch voči predchádzajúcemu elementu (nadpis, odstavec, obrázok atd).

3. Pevné

position: fixed

je vlastno absolútne, ale vždy voči oknu prehliadača!!! Čiže ak máme takúto pevnú vrstvu na stránke a stránku posúvame v zmenšenom okne prehliadača, táto vrstva ostáva VŽDY na tom istom mieste!!!

Obtekanie

Udáva, ako je element (čiže aj obrázok) alebo vrstva obtekaná ostatnými - textom, inými vrstvami atd.

Zadáva sa pomocou float a padding:

float: right

umiestni element napravo

padding-right: 20px

zabezpečí, že 20 pixelov zľava nebude okolo vrstvy nič

padding-top: 15px

zabezpečí, že 15 pixelov zdola nebude okolo vrstvy nič

Ak chceme obtekanie zrušiť, použijeme clear: a smer

clear: left

clear: right

clear: both

Poradie vrstiev

Ak umiestnime viac vrstiev na seba, o ich poradí rozhoduje z-index. Čím vyššie číslo, tým je vrstva viac navrchu.

z-index: 3

Nastavenie pozívie objektov a vrstvy

A. 3 stĺpce - ľavý a pravý s pevkou šírkou a stredný s plávajúcou šírkou

```
#lavy {position: absolute; left: 10px; top: 50px; width: 200px;}  
#stredny {margin-left: 200px; margin-right: 200px; margin-top: 10px;}  
#pravy {position: absolute; left: 10px; top: 10px; width: 200px;}
```

toto znamená, že stredný stĺpec sa bude prispôsobovať oknu prehliadača.

B. 3 stĺpce - so spoločnou hlavou a päťou

```
#hlava{width: 768px;}  
#lavy {float: left; width: 200px;}  
#stred {float: left; width: 368px;}  
#pravy {float: left; width: 200px;}  
#pata{width: 768px; clear: both;}
```

toto je s pevnou šírkou v pixeloch, clear v poslednom zabezpečí, že päta bude až dole po celej šírke. Ak zmeníme px na percentá, bude stránka prispôsobená oknu prehliadača.

C. Vnorený plávajúci element

```
#obsah {margin: 10px; border: 1px orange;}  
#navigacia {float: right; width: 150px; border: 1px orange; margin-left: 5px;}
```

obsah je celý stránka, ktorá sa posuva a prispôbuje, a navigacia je vnútri obsahu. Navigacia ostáva potom na mieste a je vždy zafixovaná na svojom mieste.