

## Obsah

1. Kaskádové štýly (Cascading Style Sheets) – CSS .....	3
1.1. Princíp fungovania CSS .....	4
1.2. Obaľujúce elementy .....	4
1.2.1. Riadkové a blokové elementy, vlastnosť display .....	5
1.3. Pravidlá CSS .....	5
1.4. Aplikácia štýlov .....	5
1.5. Dedičnosť .....	7
1.5.1. Hodnota „inherit“ .....	7
1.6. Triedy (class) .....	8
1.7. Identifikátory (ID selectors) .....	9
1.8. CSS Premenné .....	9
1.9. root selektor .....	9
1.10. Pseudo-triedy (pseudo-classes) .....	10
1.10.1. Pseudo-trieda :first-child .....	10
1.10.2. Linkové pseudo-triedy :link a :visited .....	11
1.10.3. Dynamické pseudo-triedy :hover, :active a :focus .....	11
1.10.4. Jazyková pseudo-trieda :lang .....	12
1.11. Pseudo-elementy (Pseudo-elements) .....	12
1.12. Zložená deklarácia .....	12
1.13. CSS kombinátory .....	13
1.14. Médiá .....	13
1.15. Jednotky .....	14
1.16. Farba popredia a pozadia, obrázok v pozadí .....	15
1.17. Písmo textu .....	15
1.18. Veľkosť písma .....	16
1.19. Šírka písma .....	17
1.20. Horizontálne zarovnanie textu .....	17
1.21. Odsadenie prvého riadku .....	18
1.22. Prevod textu .....	18
1.23. Textové dekorácie .....	18
1.24. Výška riadku textu .....	18
1.25. Medzery .....	18
1.26. Vertikálne zarovnanie .....	19
1.27. Zalomenie stránky .....	19

1.28. Testovanie štýlopisu .....	20
1.29. Písanie štýlov .....	20
2. Blokový model CSS .....	21
2.1. Vonkajšie okraje .....	22
2.2. Vnútorný okraj, odsadenie .....	22
2.3. Šírka hranice rámčeka .....	22
2.4. Farba hranice rámčeka .....	23
2.5. Štýl čiary hranice rámčeka .....	23
2.6. Súhrnná vlastnosť rámčeka .....	23
2.7. CSS Box sizing .....	24
3. Rozvrhnutie obsahu a obtekanie v CSS .....	25
3.1. Rozvrhnutie obsahu .....	25
3.2. Obtekanie elementov .....	26
3.3. Zrušenie obtekania .....	28
3.4. Os z .....	30
3.5. CSS Flexbox .....	31

## 1. Kaskádové štýly (Cascading Style Sheets) – CSS

Zatiaľ sme vytvárali „čisté“ (X)HTML dokumenty. Aj keď obsah elementov má svoju grafickú prezentáciu, elementy jazyka (X)HTML sme používali len na zvýraznenie sémantiky (významu) webu. Takéto dokumenty sú prehľadné, ľahko čitateľné a prístupné. Niečo im však chýba. Dizajn webu je chudobný, jednotvárný a rozmiestnenie objektov na stránke zatiaľ ovplyvniť nevieme (ak si odmyslíme tabuľky, ktoré sú aj tak určené na niečo iné). Obidva nedostatky môžeme vyriešiť pomocou kaskádových štýlov - Cascading Style Sheets (CSS). CSS okrem iného umožňujú:

- oddeliť samotnú prezentáciu (grafický vzhľad) informácií od obsahu dokumentu,
  - dôležitá vlastnosť pre výmenu informácií medzi rôznymi systémami,
- pre rôzne média alebo zariadenia môžeme definovať rôzne štýly (ten istý dokument môže vyzeráť na obrazovke monitora inak, ako po vytlačení na tlačiarňu a pod.),
  - takéto dokumenty sú prístupnejšie a flexibilnejšie,
- zmenšiť veľkosť dokumentov (grafická prezentácia pre všetky dokumenty webu sa môže definovať len raz, na jednom mieste webu),
  - táto vlastnosť je výhodná pri prenose dokumentov webu, pretože prehliadače si definíciu štýlov kešujú (ukladajú lokálne) a nemusia ju teda opakovane prenášať,
- jednou definíciou štýlov definovať vzhľad ľubovoľného počtu dokumentov,
  - užitočná vlastnosť najmä pre tvorcov rozsiahlejších webov.

O sile CSS sa môžete presvedčiť napr. na <http://www.csszengarden.com>. Celá prezentácia je definovaná len pomocou CSS. Zmenou CSS dostávame úplne inú prezentáciu, samozrejme s tým istým obsahom. Ak vypneme CSS (IE: Nástroje – Možnosti internetu – karta Všeobecné – Zjednodušenie ovládania, FF: Zobrazíť – Štýl stránky – Bez Štýlu), vidíme len samotný obsah stránky bez grafického formátovania.

Pomocou CSS vieme definovať vzhľad (vrátane umiestnenia) konkrétneho elementu (alebo jeho časti) na stránke, skupiny elementov alebo bloku elementov.

Kaskádové štýly majú špecifikáciu definovanú W3C<sup>1</sup>. Momentálne je aktuálna verzia CSS3. Jednotlivé špecifikácie nájdete na:

CSS1 - <https://www.w3.org/TR/CSS1/>

CSS2 - <https://www.w3.org/TR/CSS22/>

CSS3 - <https://www.w3.org/TR/2021/REC-css-cascade-3-20210211/>

alebo súhrnne: <https://www.w3.org/Style/CSS/current-work>

---

<sup>1</sup> Poznámka: Poznať všetky vlastnosti CSS je prakticky nemožné. Odporúčam vám pri definovaní štýlov používať nejaký editor CSS (napr. freeware TopStyle Lite, ktorý nájdete na webovej stránke v prostredí LMS Moodle alebo niektoré z komplexných vývojových prostredí ako VSC, PhpStorm a pod.)

## 1.1. Princíp fungovania CSS

Princíp fungovania CSS v súvislosti s (X)HTML dokumentom je pomerne jednoduchý. Pomocou CSS určíme, ako by mal vyzeráť vzhľad jednotlivých elementov na stránkach nášho webu, príp. kde by sa obsah týchto elementov mal na stránke zobraziť. Do samotného (X)HTML dokumentu umiestnime informáciu o tom, kde je definovaný vzhľad elementov. Samotný prehliadač sa postará o to, že obsah (X)HTML dokumentu zobrazí vo svojom okne tak, ako sme to pomocou CSS definovali.

[ukážka: [https://di.ics.upjs.sk/vyucba/pomocne\\_materialy/pws/css/c1.html](https://di.ics.upjs.sk/vyucba/pomocne_materialy/pws/css/c1.html)]

čistý  
(X)HTML  
dokument  
zobrazený  
v okne  
prehliadača



časť CSS  
pravidiel  
a prepojenie  
s (X)HTML  
dokumentom

```
/* CSS Document */
#kontajner {
  width: 800px;
  margin: auto;
  background-color: rgb(229, 229, 255);
  padding: 10px;
}

#menu {
  background-color: rgb(153, 153, 255);
  display: flex;
}

#menu li {
  display: inline-block;
  background-color: rgb(229, 229, 255);
  padding: 10px;
  margin: 0 5px 0 5px;
}

#menu li a {
  font-family: Verdana, Geneva, Tahoma, sans-serif;
  font-size: 2em;
  color: rgb(0, 0, 102);
}
```

```
<?xml version="1.0" encoding="iso-8859-2"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.d
">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="cs" lang=
">
<head>
  <meta name="generator" content="PSPad editor, www.pspad.co
  <meta http-equiv="content-type" content="text/html; charse
  <link rel="stylesheet" href="style.css" type="text/css"/>
  <title>Pes</title>
</head>
<body>
  <div id="kontajner">
    <div id="menu">
      <ul>
        <li><a href='c1.html'>pes</a></li>
        <li><a href='c2.html'>mačka</a></li>
        <li><a href='c3.html'>papagáj</a></li>
      </ul>
    </div>
    <h1>Verný a priateľský pes</h1>
    
    <p>Toto je nejaký text o verných a priateľských psoch. Tot
    nejaký text o verných a priateľských psoch. Toto je ne
    text o verných a priateľských psoch. Toto je nejaký te
    verných a priateľských psoch. Toto je nejaký text o ve
    a priateľských psoch. Toto je nejaký text o verných a
```

(X)HTML  
dokument  
zobrazený  
v okne  
prehliadača  
podľa  
definovaných  
CSS  
pravidiel



## 1.2. Obaľujúce elementy

Pri formátovaní pomocou CSS budeme často ovplyvňovať vlastnosti väčšieho množstva elementov. (X)HTML ponúka spôsob, ako „zoskupiť“ viacero elementov bez zásahu do ich formátovania. Slúžia na to dva elementy, `<div>` a `<span>`. Obidva majú svoj význam v spojení s CSS.

`<div>` je blokový element. Pred ním a za ním je automaticky pridaný koniec riadku. Nesmie byť uzatvorený v žiadnom riadkovom elemente (`<a>`, `<span>`, ...).

`<span>` je riadkový element. Nesmie uzatvárať žiadne iné blokové elementy (`<div>`, `<p>`, `<h1>`, ...).

Pozor, elementy `<div>` a `<span>` nenesú žiadnu sémantickú informáciu. Slúžia len na formátovanie vzhľadu dokumentu.

DIV element slúžil v (X)HTML na obalenie jednotlivých častí dokumentu (hlavička, obsah, menu, päta a pod). V HTML5 je takéto použitie kontraproduktívne, pretože HTML5 ponúka pre tento účel samostatné sémantické elementy (napr. header, main, nav, section, article, aside, footer)<sup>2</sup>.

### 1.2.1. Riadkové a blokové elementy, vlastnosť display

Spôsob zobrazenia elementu, riadkový (správa sa ako jeden znak textu) alebo blokový (je v samostatnom riadku, pred ním a za ním je vložený znak konca riadka), môžeme modifikovať definovaním vlastnosti `display`. Táto vlastnosť má niekoľko prípustných hodnôt. Medzi najčastejšie používané patria:

- inline: element sa zobrazí ako riadkový,
- block: element sa zobrazí ako blokový,
- inline-block: obsah tohto bloku je formátovaný ako normálny blok, ako celok je však zaradený v riadku ako jeden znak,
- none: element sa vôbec nezobrazí.

Vo väčšine prípadov spôsob zobrazenia elementov nemusíme modifikovať. Sú ale situácie, keď predefinované správanie elementov musíme upraviť (napr. zobraziť položky zoznamu v riadku).

## 1.3. Pravidlá CSS

Samotná definícia štýlov je pomerne jednoduchá. Pozostáva zo selektoru<sup>3</sup> a definície jeho vlastností, napr.:

```
selector { [vlastnosť : hodnota; ]* }
*        { [vlastnosť : hodnota; ]* }
```

[ukážka: [http://di.ics.upjs.sk/vyucba/pomocne\\_materialy/pws/index.php?id=css10](http://di.ics.upjs.sk/vyucba/pomocne_materialy/pws/index.php?id=css10)]

```
h2 {color: #FFFFFF; background-color: #000000}
```

V dokumentoch, v ktorých sa tento štýl aplikuje, budú nadpisy druhej úrovne zobrazené bielou farbou na čiernom pozadí. Na nadpis úrovne 1 sa aplikuje prednastavený štýl prehliadača (vysvetlíme si neskôr).

Pri definícii môžeme použiť aj univerzálny selektor „\*“. Vlastnosti definované pre univerzálny selektor sa použijú pre všetky elementy dokumentu.

## 1.4. Aplikácia štýlov

Existuje niekoľko spôsobov, ako sa štýly aplikujú na elementy dokumentu:

[ukážka: [http://di.ics.upjs.sk/vyucba/pomocne\\_materialy/pws/index.php?id=css2](http://di.ics.upjs.sk/vyucba/pomocne_materialy/pws/index.php?id=css2)]

<sup>2</sup> Popis štruktúrnych elementov <https://www.w3.org/TR/html52/sections.html#sections>

<sup>3</sup> Zoznam všetkých selektorov [https://www.w3schools.com/cssref/css\\_selectors.asp](https://www.w3schools.com/cssref/css_selectors.asp)

```
...
<link rel="stylesheet" type="text/css" href="css2.css" />
<style type="text/css">
  h1, h2 {font-family: serif; font-size: 2.5em;}
</style>
</head>
<body>
  <h1 style="font-size:3em;">Nadpis úrovne 1</h1>
  <h2>Nadpis úrovne 2</h2>
  <h3>Nadpis úrovne 3</h3>
  <h4>Nadpis úrovne 4</h4>
```

obsah súboru „css2.css“:

```
h1, h2, h3 {font-family: sans-serif; font-size:2em; color:#FF0000}
```

Súbor css2.css obsahuje definíciu štýlov pre nadpisy prvej až tretej úrovne. Pre tieto nadpisy bude použité bezpätkové písmo, dvojnásobnej veľkosti, červenej farby.

- Ak sa nedeklarujú žiadne štýly, aplikujú sa **štýly prehliadača** (doteraz sme nepoužívali žiadne štýly, napriek tomu naše stránky predsa len mali nejaký vzhľad). V našom prípade bude takto formátovaný nadpis štvrtej úrovne.

```
<h4>Nadpis úrovne 4</h4>
```

Nadpis štvrtej úrovne je formátovaný preddefinovaným štýlom prehliadača.

- **Pripojené štýly** sa definujú v externom dokumente a k danému dokumentu sa pripoja pomocou elementu link v hlavičke dokumentu.

```
<link rel="stylesheet" type="text/css" href="css2.css" />
```

Toto je zároveň najuniverzálnejší spôsob definície vzhľadu webu. V jednom dokumente (css2.css) vieme definovať vzhľad všetkých stránok nášho webu. Odporúča sa používať práve tento spôsob. V našom prípade budú pripojeným štýlopisom ovplyvnené nadpisy prvej až tretej úrovne.

- **Vložené štýly** sa definujú v hlavičke dokumentu a aplikujú sa na príslušné elementy konkrétneho dokumentu.

```
<style type="text/css">
  h1, h2 {font-family: serif; font-size: 2.5em;}
</style>
```

Pre nadpisy prvej a druhej úrovne v tomto dokumente bude použité pätkové písmo dva a pol násobnej veľkosti.

- **Individuálne štýly** sa používajú pre jednorazové účely, spravidla definujú vlastnosť jedného konkrétneho elementu.

```
<h1 style="font-size:3em;">Nadpis úrovne 1</h1>
```

Pre tento konkrétny nadpis v tomto konkrétnom dokumente bude použitá trojnásobná veľkosť textu.

V prípade konfliktu štýlov sa aplikuje ten štýl, ktorý je k elementu najbližšie (resp. bol načítaný ako posledný). Štýly prehliadača sa nahradia pripojenými štýlmi. Pripojené štýly sa nahradia vloženými a vložené štýly sú nahradené individuálnymi. Od tohto kaskádového spôsobu implementácie štýlov pochádza aj ich názov – kaskádové štýly. Konflikt štýlov nie je ničím výnimočným a bežne sa vyskytuje. Najskôr sa definujú štýly na globálnej úrovni (napr. všetky texty na webe) a potom sa v prípade potreby na lokálnej úrovni upravujú (napr. nadpisy).

štýly prehliadača	pripojené štýly	vložené štýly	individuálne štýly

Skúste sa zamyslieť, ako sa v našom príklade aplikujú konkrétne štýly na jednotlivé úrovne nadpisov.

Na tomto mieste je vhodné si uvedomiť fakt, že ak nenastavíme konkrétne hodnoty niektorých vlastností, použijú sa prednastavené vlastnosti definované prehliadačom<sup>4</sup>. Môže sa tak stať, že stránka vyzerá v rôznych prehliadačoch inak. Riešením je tzv. reset štýlov – nastavenie najčastejšie používaných vlastností na vopred vybrané hodnoty pomocou špeciálneho, na tento účel vytvoreného štýlopisu. Použiť môžeme niektorý z verejne dostupných resetovacích štýlopisov (napr. z <https://meyerweb.com/eric/tools/css/reset/>). CSS reset často odstráni formátovanie aj pri tých elementoch, kde by sme si to želali ponechať. Opätovne preto musíme formátovanie niektorým elementom definovať. V dôsledku toho potrebujeme preniesť väčšie množstvo dát, než je nutné a prehliadač potrebuje čas navyše, aby pravidlá v štýlopisoch aplikoval.

Dnes sa od samotného resetovania, ktoré „maže“ prednastavené hodnoty prehliadača, upúšťa a skôr sa uprednostňuje normalizácia štýlov (napr. <https://necolas.github.io/normalize.css/>). Pri normalizácii sa zachovávajú predvolené užitočné nastavenia štýlopisu prehliadača a opravujú sa chyby a nejednotnosť štýlov prehliadačov.

## 1.5. Dedičnosť

Pojmom dedičnosť označujeme vlastnosť, keď elementy preberajú štýl od im nadradených elementov.

[ukážka: [http://di.ics.upjs.sk/vyucba/pomocne\\_materialy/pws/index.php?id=css11](http://di.ics.upjs.sk/vyucba/pomocne_materialy/pws/index.php?id=css11)]

```
<div style="color:#0000FF">
  <p>Odsek textu. Odsek textu. Odsek textu.</p>
  <table border="1">
    <tr>
      <td>Text v tabuľke</td>
    </tr>
  </table>
</div>
```

Definovali sme modrú farbu písma elementu `<div>`. Farba písma v jeho podradených elementoch `<p>` a `<table>` je tiež modrá.

### 1.5.1. Hodnota „inherit“

Niektoré vlastnosti sa normálne nededia. Ak chceme dosiahnuť dedenie takejto vlastnosti z rodičovského elementu na potomka, pre požadovanú vlastnosť potomka

<sup>4</sup> Prednastavené hodnoty CSS vlastností pre jednotlivé prehliadače môžeme zistiť napr. na <https://browserdefaultstyles.com>.



definujeme hodnotu „inherit“. Hodnotu „inherit“ môžeme použiť aj v prípade, ak chceme zabezpečiť, aby sa vlastnosť určite zdedila.

[ukážka: [http://di.ics.upjs.sk/vyucba/pomocne\\_materialy/pws/index.php?id=css31](http://di.ics.upjs.sk/vyucba/pomocne_materialy/pws/index.php?id=css31)]

```
<p style="color:#FF0000">
  Odsek textu...
  <a href='http://google.com'>Google odkaz 2.</a>
  Odsek textu...
  <a style="color: inherit" href='http://google.com'>Google odkaz 1.</a>
  Odsek textu...
</p>
```

Text prvého odkazu je modrý (resp. formátovaný podľa štýlu prehliadača), druhý odkaz prebral farbu textu od nadradeného elementu <p>.

## 1.6. Triedy (class)

Ak potrebujeme definovať skupinu rovnakých vlastností pre rôzne elementy, môžeme definovať triedu vlastností, napr.:

[ukážka: [http://di.ics.upjs.sk/vyucba/pomocne\\_materialy/pws/index.php?id=css3](http://di.ics.upjs.sk/vyucba/pomocne_materialy/pws/index.php?id=css3)]

definícia:

```
.selektor { [vlastnost : hodnota; ]* }
```

napr.:

```
.farba {color: #FF0000}
```

použitie:

```
<h1 class="farba">Nadpis úrovne 1</h1>
<p class="farba">Text odseku.</p>
```

Selektor triedy v definícii štýlu vždy začína bodkou.

Často sa môžeme stretnúť so situáciou, keď sa niektoré vlastnosti opakujú vo viacerých prípadoch. Aby sme nemuseli opakovane definovať tie isté vlastnosti vo viacerých triedach, výhodnejšie je definovať si viacero tried a tie potom vhodne kombinovať.

[ukážka: [http://di.ics.upjs.sk/vyucba/pomocne\\_materialy/pws/index.php?id=css32](http://di.ics.upjs.sk/vyucba/pomocne_materialy/pws/index.php?id=css32)]

```
<style type="text/css">
  .dolezite {
    color: #000000;
    background-color: #FFCCCC;
  }
  .velkePismo {
    font-size: 2em;
  }
  .vacsiePismo {
    font-size: 1.2em;
  }
</style>
</head>
<body>
<h1 class="dolezite velkePismo">Nadpis</h1>
<p>Odsek textu a <strong class="dolezite vacsiePismo">dôležité slová</strong> v
ňom.</p>
```



## 1.7. Identifikátory (ID selectors)

Identifikátory fungujú podobne ako triedy ale s tým rozdielom, že v dokumente môže byť len jeden element označený daným identifikátorom. Obvykle sa toto riešenie používa pre potreby skriptov (skript takto vie „osloviť“ konkrétny element).

[ukážka: [http://di.ics.upjs.sk/vyucba/pomocne\\_materialy/pws/index.php?id=css4](http://di.ics.upjs.sk/vyucba/pomocne_materialy/pws/index.php?id=css4)]

definícia:

```
#selektor {[vlastnost : hodnota; ]*}
```

napr.:

```
#nadpis {color: #FF0000}
```

použitie:

```
<h1 id="nadpis">Nadpis úrovne 1</h1>
```

Selektor identifikátora v definícii štýlu vždy začína znakom mriežka #. Ak by sme sa pokúsili aplikovať štýl `nadpis` na ďalší element, bola by to chyba.

## 1.8. CSS Premenné

Ak v štýlope potrebujeme opakovane používať niektorú z hodnôt, môžeme využiť premenné. Premennú definujeme pre ten element (alebo selektor), v ktorom ju chceme používať. Premenná je prístupná aj pre potomkov tohto elementu (alebo elementu na ktorý sa selektor vzťahuje). Ak potrebujeme premennú používať pre elementy v celom dokumente, je výhodné ju definovať v selektore `:root`. Názov premennej musí začínať dvoma znakmi pomlčka „--“ a v názve premennej záleží na veľkosti písmen. Ak hodnotu premennej chceme použiť ako hodnotu nejakej vlastnosti, sprístupníme ju pomocou funkcie `var()`.

[ukážka: [http://di.ics.upjs.sk/vyucba/pomocne\\_materialy/pws/index.php?id=css37](http://di.ics.upjs.sk/vyucba/pomocne_materialy/pws/index.php?id=css37)]

```
:root {
  --farba-svetla: #CCCCCC;
  --farba-tmava: #333333;
}
.svetle_na_tmavom {
  color: var(--farba-svetla);
  background-color: var(--farba-tmava);
}
.tmave_na_svetlom {
  color: var(--farba-tmava);
  background-color: var(--farba-svetla);
}
```

## 1.9. root selektor

`:root` selektor označuje koreňový element dokumentu. V (X)HTML je týmto elementom element `html`.

```
:root {[vlastnost : hodnota; ]*}
/* štýl pre koreňový element dokumentu */
```

[ukážka: [http://di.ics.upjs.sk/vyucba/pomocne\\_materialy/pws/index.php?id=css34](http://di.ics.upjs.sk/vyucba/pomocne_materialy/pws/index.php?id=css34)]

```
...
<style type="text/css">
  :root {
    background-color: #000000;
```

```

        color: #00FF00;
        font-family: "Courier New", Courier, monospace
    }
</style>
</head>
<body>
<p>Monochromatický displej</p>
<ul>
    <li>zelený text</li>
    <li>čierne pozadie</li>
</ul>

```

## 1.10. Pseudo-triedy (pseudo-classes)<sup>5</sup>

Pseudo-triedy vytvára prehliadač pri zobrazení dokumentu. Rozlišujeme ich ako triedy pri definíciách, ale nerozlišujeme ich v dokumente (rozlíši ich automaticky prehliadač). Pseudo-triedy sa pri definovaní označujú s prefixom dvojbodka `:pseudotrieda`.

### 1.10.1. Pseudo-trieda `:first-child`

Pseudo-trieda `:first-child` označuje element, ktorý je prvým potomkom iného elementu (je do neho vnorený ako prvý). Pri definovaní tejto pseudo-triedy môžeme určiť dvojicu elementov rodič - potomok, alebo určíme len potomka, resp. len rodiča.

```

rodic > potomok:first-child {[vlastnost : hodnota; ]*}
/* štýl pre prvý element (potomok) v elemente rodic */
* > potomok:first-child {[vlastnost : hodnota; ]*}
/* štýl pre prvý element (potomok) v ľubovoľnom elemente */
rodic > *:first-child {[vlastnost : hodnota; ]*}
/* štýl pre ľubovoľný prvý element v elemente rodic */

```

[ukážka: [http://di.ics.upjs.sk/vyucba/pomocne\\_materialy/pws/index.php?id=css5\\_1](http://di.ics.upjs.sk/vyucba/pomocne_materialy/pws/index.php?id=css5_1)]

```

<style type="text/css">
    div:first-child {border: #666666 solid 1px;}
    div > p:first-child {font-weight: bold;}
</style>
</head>
<body>
<div>
    Prvý div v body. Aplikuje sa štýl pseudo-triedy :first-child pre div.
</div>

<div>
    Druhý div v body. Neaplikuje sa žiaden z nami definovaných štýlov.
</div>

<div>
    <p>Prvý odsek v div-e. Aplikuje sa štýl pseudo-triedy :first-child pre
    odsek.</p>
    <p>Druhý odsek v div-e. Neaplikuje sa štýl pseudo-triedy :first-child pre
    odsek.</p>
</div>

```

Na prvý element `<div>` sa aplikoval štýl orámovania, pretože je prvým elementom v rodičovskom elemente (`<body>`). Na druhý element `<div>` sa tento štýl neaplikoval.

<sup>5</sup> Ďalšie pseudo-triedy nájdeme na <https://www.w3.org/TR/2018/REC-selectors-3-20181106/#pseudo-classes>

Na prvý element, odsek textu (`<p> ... </p>`) v obaľujúcom elemente `<div>` bude aplikovaný definovaný štýl. Písmo odseku bude tučné. Na ostatné odseky, resp. elementy v elemente `<div>`, sa štýl aplikovať nebude.

### 1.10.2. Linkové pseudo-triedy `:link` a `:visited`

Linkové (vytvárajú sa pri linkovom elemente `<a>...</a>`) pseudo-triedy `:link` a `:visited` sa aplikujú na odkazy. Týmto spôsobom ich dokážeme lepšie odlišiť od okolitého textu.

```
a:link      {[vlastnosť : hodnota; ]*}      /* nenavštívená linka */
a:visited   {[vlastnosť : hodnota; ]*}      /* navštívená linka */
```

[ukážka: [http://di.ics.upjs.sk/vyucba/pomocne\\_materialy/pws/index.php?id=css5\\_2](http://di.ics.upjs.sk/vyucba/pomocne_materialy/pws/index.php?id=css5_2)]

```
...
<style type="text/css">
  a:link {color:#FF0000}
  a:visited {color:#00FF00}
</style>
</head>
<body>
  <a href="http://di.ics.upjs.sk/" title="Didaktika informatiky PF
UPJŠ">di.ics.upjs.sk</a>
```

Linky, ktoré sme ešte nenavštívili, sú zobrazené červenou farbou. Navštívené linky sú zobrazené farbou zelenou.

### 1.10.3. Dynamické pseudo-triedy `:hover`, `:active` a `:focus`

Dynamická pseudo-trieda `:hover` je vytvorená, pokiaľ sa ukazovateľ (najčastejšie kurzor myšky) nachádza nad daným elementom (element nemusí byť aktivovaný).

Dynamická pseudo-trieda `:active` je vytvorená, keď užívateľ aktivuje daný element (napr. čas medzi tým ako užívateľ stlačí a uvoľní tlačidlo myšky).

Dynamická pseudo-trieda `:focus` je vytvorená, pokiaľ je element vybraný používateľom (napr. tabulátorom vybraný odkaz).

```
a:hover     {[vlastnosť : hodnota; ]*}      /* linka pri nájazde myši */
a:focus     {[vlastnosť : hodnota; ]*}      /* vybraná linka */
a:active    {[vlastnosť : hodnota; ]*}      /* aktívna linka */
```

[ukážka: [http://di.ics.upjs.sk/vyucba/pomocne\\_materialy/pws/index.php?id=css5\\_3](http://di.ics.upjs.sk/vyucba/pomocne_materialy/pws/index.php?id=css5_3)]

```
...
<style type="text/css">
  a:hover {color:#FF0000}
  a:focus {color:#00FF00}
  a:active {color:#FF00FF}
</style>
</head>
<body>
  <a href="http://di.ics.upjs.sk/" title="Didaktika informatiky PF
UPJŠ">di.ics.upjs.sk</a>
```

Linka, nad ktorou sa nachádza kurzor myšky, bude červená. Ak ju vyberieme, ale neaktivujeme (napr. tabulátorom), bude zelená. Ak ju aktivujeme (klepneme na ňu myškou, ale tlačidlo nepustíme), zmení sa na fialovú.

### 1.10.4. Jazyková pseudo-trieda :lang

Jazykovú pseudo-triedu `:lang` používame na rozlíšenie častí dokumentu, ktoré sú písané v konkrétnom jazyku.

```
selektor:lang(jazyk) {[vlastnosť : hodnota; ]*}
/* štýl jazyka v danom selektore*/
```

[ukážka: [http://di.ics.upjs.sk/vyucba/pomocne\\_materialy/pws/index.php?id=css5\\_4](http://di.ics.upjs.sk/vyucba/pomocne_materialy/pws/index.php?id=css5_4)]

```
...
<style type="text/css">
  span:lang(en) { font-style:italic;}
</style>
</head>
<body>
  Ústav informatiky (<span lang='en'>Institute of Computer Science</span>).
```

Texty v angličtine obalené elementom `<span>` budú zvýraznené kurzívou.

### 1.11. Pseudo-elementy (Pseudo-elements)<sup>6</sup>

Pseudo-elementy tiež nemusíme špeciálne vyznačovať. Prehliadač ich deteguje automaticky ako časť elementu. Pseudo-elementy sa v definícií označujú s prefixom dve dvojbodky `::pseudoelement`. Prístupné sú tieto pseudo-elementy:

```
blokovy_element::first-line {[vlastnosť : hodnota; ]*}
/* ovplyvní prvý riadok blokového elementu, napr. odseku */
Element::first-letter {[vlastnosť : hodnota; ]*}
/* ovplyvní prvý znak elementu, napr. odseku */
```

[ukážka: [https://di.ics.upjs.sk/vyucba/pomocne\\_materialy/pws/index.php?id=css6](https://di.ics.upjs.sk/vyucba/pomocne_materialy/pws/index.php?id=css6)]

```
<style type="text/css">
  p::first-letter {font-size:4em}
  p::first-line   {text-transform: uppercase}
</style>
</head>
<body>
  <p>Odsek textu. Odsek textu. ...
```

Prvý znak odseku má štvornásobnú veľkosť oproti okolitému textu. Prvý riadok odseku je zobrazený veľkým písmom. Všimnime si, že pri zmene veľkosti zobrazenia (zoom stránky, zmena veľkosti okna) sa táto zmena aplikuje len na prvý riadok.

### 1.12. Zložená deklarácia

Ak definujeme rovnaké vlastnosti niekoľkým elementom súčasne, môžeme to spraviť naraz zloženou deklaráciou (nezabudnite na čiarky oddeľujúce jednotlivé selektory):

```
selektor1, selektor2, .. selektorn {[vlastnosť : hodnota; ]*}
/* definícia vlastností pre selektor1, selektor2 až selektorn */
```

[ukážka: [http://di.ics.upjs.sk/vyucba/pomocne\\_materialy/pws/index.php?id=css7](http://di.ics.upjs.sk/vyucba/pomocne_materialy/pws/index.php?id=css7)]

```
<style type="text/css">
  h1, p, a {font-family:Arial, Helvetica, sans-serif}
</style>
</head>
```

<sup>6</sup> Ďalšie pseudo-elementy nájdeme na <https://www.w3.org/TR/2018/REC-selectors-3-20181106/#pseudo-elements>.

```
<body>
  <h1>Nadpis úrovne 1</h1>
  <p>Odsek textu. Odsek textu. Odsek textu. ... </p>
  <a href="http://di.ics.upjs.sk/" title="Didaktika informatiky PF
UPJŠ">di.ics.upjs.sk</a>
```

## 1.13. CSS kombinátory

Kombinátory použijeme v situácii, keď potrebujeme špecifikovať vzťah medzi selektormi. Kombinátor môžeme vložiť medzi dva jednoduché selektory. Rozlišujeme štyri rôzne kombinátory (medzera, >, +, ~):

```
/* štýl pre element2, ktorý je vnorený v element1 */
element1 element2 {[vlastnosť : hodnota; ]*}

/* štýl pre element2, ktorý je vnorený priamo v element1 */
element1 > element2 {[vlastnosť : hodnota; ]*}

/* štýl pre element2, ktorý je ihneď za element1 */
element1 + element2 {[vlastnosť : hodnota; ]*}

/* štýl pre element2, ktorý je niekde za element1 */
element1 ~ element2 {[vlastnosť : hodnota; ]*}
```

[ukážka: [http://di.ics.upjs.sk/vyucba/pomocne\\_materialy/pws/index.php?id=css8](http://di.ics.upjs.sk/vyucba/pomocne_materialy/pws/index.php?id=css8)]

```
<style type="text/css">
  div {border: #999999 solid 2px;}
  p {border: #DDDDDD solid 1px;}
  /*styl pre span v div, nemusí byť priamo v div*/
  div span {background-color: #DDD}
  /*styl pre span, ktorý je priamo v div*/
  div > span {color: #DD0000}
  /*styl pre span priamo za div*/
  div + span {text-decoration: underline}
  /*styl pre span niekde za div*/
  div ~ span {font-weight: bold}
</style>
</head>
<body>
<div>
  <p>Odsek textu. Odsek textu. Odsek textu. <span>span v odseku v div</span> Odsek
textu. Odsek textu.</p>
  <span>span priamo v div</span>
</div>
<span>span priamo za div</span>
<p>Odsek textu. Odsek textu. Odsek textu. Odsek textu. Odsek textu.</p>
<span>span niekde za div</span>
```

## 1.14. Médiá

Pomocou CSS môžeme definovať vzhľad dokumentu v závislosti od zobrazovacieho zariadenia, na ktorom je dokument zobrazený. K dispozícii máme niekoľko typov zobrazovacích zariadení<sup>7</sup>:

<sup>7</sup> V predchádzajúcich verziách boli definované aj iné zariadenia. Autorom sa odporúča, aby namiesto už nepodporovaných konkrétnych zariadení využili ich vlastnosti a tak lepšie prispôsobili vzhľad konkrétnemu zariadeniu. V budúcnosti budú pravdepodobne aj tu uvedené zariadenia označené ako zastarané.

- all – všetky zariadenia,
- print – náhľad dokumentu, vytlačený dokument, zariadenia určené na reprodukciu alebo tlač obrazu,
- screen – bežná obrazovka počítača, zariadenia, ktoré nie sú zahrnuté v časti print

Definovanie štýlu pre vybrané zariadenie:

```
@media typ_media { selektor1 {[vlastnost : hodnota; ]*}
                  selektor2 {[vlastnost : hodnota; ]*}
                  ...
                }
```

[ukážka: [http://di.ics.upjs.sk/vyucba/pomocne\\_materialy/pws/index.php?id=css26](http://di.ics.upjs.sk/vyucba/pomocne_materialy/pws/index.php?id=css26)]

```
...
<style type="text/css">
  @media screen {p {color: #0000FF} }
  @media print {p {color: #FF0000} }
</style>
</head>
<body>
  <p>Tento text je na obrazovke modrý, po vytlačení na farebnej tlačiarni (a v
náhľade tlače) bude červený</p>
```

## 1.15. Jednotky

Pri definícii rozmerov elementov (veľkosť písma, veľkosť boxu, vzdialenosti a pod.) je potrebné použiť okrem samotnej hodnoty aj príslušnú jednotku (okrem prípadu keď hodnotou je 0). Jednotky môžu byť relatívne alebo absolútne:

**relatívne jednotky:**

- em: veľkosť písma vzhľadom na okolitý text,
- ex: výška písmena 'x' okolitého textu,
- ch: šírka znaku '0' okolitého textu,
- rem: veľkosť písma rootovského elementu,
- vw: 1% z viditeľnej šírky zobrazovacieho zariadenia,
- vh: 1% z viditeľnej výšky zobrazovacieho zariadenia,
- vmin: 1% z viditeľnej časti menšieho rozmeru zobrazovacieho zariadenia,
- vmax: 1% z viditeľnej časti väčšieho rozmeru zobrazovacieho zariadenia,
- % - relatívne k rodičovskému elementu (niekedy môže byť problém zistiť, čo sa berie za základ: jeden z rozmerov elementu, nadradený element, šírka okna a pod.)

**absolútne jednotky:**

- in: palce (1in = 2.54cm),
- cm: centimetre,
- mm: milimetre,
- pt: bod, 1pt = 1/72 z 1in,
- pc: pica, 1pc = 1/6 z 1in,
- px: pixel, 1px = 1/96 z 1in.

Pri tvorbe by sme mali uprednostniť relatívne jednotky pred absolútnymi. Dokument používajúci relatívne jednotky je flexibilnejší a prístupnejší. Napr. ak je veľkosť písma definovaná pomocou relatívnych jednotiek, jeho veľkosť na obrazovke si vieme prispôbiť podľa potrieb (zmena mierky zobrazenia - lupa).

## 1.16. Farba popredia a pozadia, obrázok v pozadí

Väčšine objektov môžeme nastaviť farbu popredia a pozadia. Zaujímavé efekty možno dosiahnuť s obrázkom v pozadí (pozor však na čitateľnosť textu na takomto pozadí).

farba popredia

```
selektor { color: farba }
```

farba\_pozadia

```
selektor { background-color: farba }
```

obrazok\_pozadia

```
selektor {
  background-image: url(url obrázku);
  background-repeat: repeat | repeat-x | repeat-y | no-repeat | inherit;
  background-attachment: scroll | fixed | inherit;
  background-position:
    [
      [ left | center | right | top | bottom | <percentage> | <length> ]
      |
      [ left | center | right | <percentage> | <length> ]
      [ top | center | bottom | <percentage> | <length> ]
      |
      [ center | [ left | right ] [ <percentage> | <length> ]? ] &&
      [ center | [ top | bottom ] [ <percentage> | <length> ]? ]
      |
      inherit
    ]
}
```

[ukážka: [http://di.ics.upjs.sk/vyucba/pomocne\\_materialy/pws/index.php?id=css12](http://di.ics.upjs.sk/vyucba/pomocne_materialy/pws/index.php?id=css12)]

```
<p style="color:#000066;
  background-color:#FFFFCC";
  background-image:url(pozadie1.gif);
  background-repeat:repeat-y;
  background-position:left;">
Text odseku. Text odseku. ...
</p>
<p style="background-image:url(pozadie2.gif);
  background-repeat:repeat-x;
  background-position:top;">
Text odseku. Text odseku. ...
</p>
```

Farbu ako takú môžeme definovať rôznymi spôsobmi. Nasledujúce texty sú zobrazené tou istou farbou:

```
<p style="color: #D2691E">Toto je text čokoládovej farby.</p>
<p style="color: chocolate">Toto je text čokoládovej farby.</p>
<p style="color: rgb(210, 105, 30)">Toto je text čokoládovej farby.</p>
```

## 1.17. Písmo textu

Pri voľbe písma textu je potrebné byť opatrný. Môže sa stať, že nami vybrané písmo (font) sa na počítači užívateľa nenachádza. Je preto dobré použiť pri definícii písma aj všeobecnú rodinu písiem. Uvádza sa ako posledná z množiny písiem. Ak sa nenachádza v počítači užívateľa ani jeden z uvedených konkrétnych fontov, použije sa niektorý zo všeobecnej rodiny písiem.

pätkové:



```
{ font-family: Georgia, 'Times New Roman', Times, serif }
```

bezpätkové:

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

neproporcionálne:

```
{ font-family: 'Courier New', Courier, monospace }
```

dekoratívne:

```
{ font-family: 'Alpha Geometrique', Critter, fantasy }
```

ručné:

```
{ font-family: 'Caflisch Script', 'Adobe Poetica', cursive }
```

[ukážka: [http://di.ics.upjs.sk/vyucba/pomocne\\_materialy/pws/index.php?id=css13](http://di.ics.upjs.sk/vyucba/pomocne_materialy/pws/index.php?id=css13)]

```
<span style="font-family: Georgia, 'Times New Roman', Times, serif">
pätkové<br />
</span>
<span style="font-family: Verdana, Arial, Helvetica, sans-serif">
bezpätkové<br />
</span>
<span style="font-family: 'Courier New', Courier, monospace">
neproporcionálne<br />
</span>
<span style="font-family: 'Alpha Geometrique', Critter, fantasy">
dekoratívne<br />
</span>
<span style="font-family: 'Caflisch Script', 'Adobe Poetica', cursive">
ručné
</span>
```

Jedným z možných riešení problému, keď sa v počítači nenachádza požadovaný font je možnosť použiť externé fonty. Jednu z možností ponúka služba Google Fonts (<https://www.google.com/fonts#>).

[ukážka: [http://di.ics.upjs.sk/vyucba/pomocne\\_materialy/pws/index.php?id=css13\\_1](http://di.ics.upjs.sk/vyucba/pomocne_materialy/pws/index.php?id=css13_1)]

```
<link href='https://fonts.googleapis.com/css?family=Oswald&display=swap'
rel='stylesheet' />
</head>
<body>
<p style="font-family: 'Oswald', sans-serif;">Tento text je zobrazený písmom
Oswald.</p>
<p>Tento text je zobrazený preddefinovaným písmom</p>
```

## 1.18. Veľkosť písma

Veľkosť písma textu možno zadávať niekoľkými spôsobmi. Absolútna veľkosť, relatívna veľkosť, dĺžka v jednotkách alebo v percentách.

- absolútna veľkosť, medium = 12pt. škálovací faktor dvoch susedných hodnôt je 1.2

```
{ font-size: [xx-small|x-small|small|medium|large|x-large|xx-large] }
```

- relatívna veľkosť vzhľadom na rodičovský element, škálovací faktor je 1.2

```
{ font-size: [larger|smaller] }
```

- veľkosť v jednotkách

```
{ font-size: hodnota<jednotky> }
```

- veľkosť v percentách. vzhľadom na rodičovský element

```
{ font-size: hodnota<percento> }
```

[ukážka: [http://di.ics.upjs.sk/vyucba/pomocne\\_materialy/pws/index.php?id=css14](http://di.ics.upjs.sk/vyucba/pomocne_materialy/pws/index.php?id=css14)]

```
<p><span style="font-size: xx-small">xx-small</span>
  <span style="font-size: x-small">x-small</span>
  <span style="font-size: small">small</span>
  <span style="font-size: medium">medium</span>
  <span style="font-size: large">large</span>
  <span style="font-size: x-large">x-large</span>
  <span style="font-size: xx-large">xx-large</span>
</p>
<p><span style="font-size: smaller">smaller</span>
  normálne
  <span style="font-size: larger">larger</span>
</p>
<p><span style="font-size: 0.5em">0.5em</span>
  <span style="font-size: 1em">1em</span>
  <span style="font-size: 2em">2em</span>
</p>
<p>
  <span style="font-size: 5mm">5mm</span>
  <span style="font-size: 10mm">10mm</span>
  <span style="font-size: 15mm">15mm</span>
</p>
<p><span style="font-size: 50%">50%</span>
  <span style="font-size: 100%">100%</span>
  <span style="font-size: 150%">150%</span>
</p>
```

## 1.19. Šírka písma

Šírka písma je vlastnosť, ktorá určuje ako „tučné“ písmo bude. Platí, že normal = 400, bold = 700.

```
{ font-weight: normal | bold | bolder | lighter | 100 | 200 | 300 | 400 | 500 |
600 | 700 | 800 | 900 | inherit }
```

[ukážka: [http://di.ics.upjs.sk/vyucba/pomocne\\_materialy/pws/index.php?id=css15](http://di.ics.upjs.sk/vyucba/pomocne_materialy/pws/index.php?id=css15)]

```
<span style="font-weight: lighter">lighter</span>
<span style="font-weight: normal">normal</span>
<span style="font-weight: bold">bold</span>
<span style="font-weight: bolder">bolder</span>
```

## 1.20. Horizontálne zarovnanie textu

Zarovnanie textu v blokovom elemente (napr. v odseku) nastavíme vlastnosťou `text-align`.

```
{ text-align: left | right | center | justify | inherit }
```

[ukážka: [http://di.ics.upjs.sk/vyucba/pomocne\\_materialy/pws/index.php?id=css16](http://di.ics.upjs.sk/vyucba/pomocne_materialy/pws/index.php?id=css16)]

```
<p style="text-align:left">Text odseku zarovnaný na ľavý okraj. ...
<p style="text-align:right">Text odseku zarovnaný na pravý okraj. ...
<p style="text-align:center">Text odseku zarovnaný na stred. ...
<p style="text-align:justify">Text odseku zarovnaný podľa okrajov. ...
```

## 1.21. Odsadenie prvého riadku

Na sprehľadnenie textu je vhodné odseky oddeliť opticky od seba. Jednou z možností je odsadenie prvého riadku odseku. Odsadenie riadku môžeme použiť v ľubovoľnom blokovom elemente.

```
{ text-indent: hodnota<jednotky> }
```

[ukážka: [http://di.ics.upjs.sk/vyucba/pomocne\\_materialy/pws/index.php?id=css17](http://di.ics.upjs.sk/vyucba/pomocne_materialy/pws/index.php?id=css17)]

```
<p style="text-indent:20mm">Odsadenie prvého riadku v texte. ...
```

## 1.22. Prevod textu

Pomocou prevodov textu môžeme dosiahnuť zaujímavé efekty. Prvé písmeno každého slova textu obaleného v danom elemente môže byť veľké, všetky písmená môžu byť veľké alebo malé.

```
{ text-transform: capitalize | uppercase | lowercase | none | inherit }
```

[ukážka: [http://di.ics.upjs.sk/vyucba/pomocne\\_materialy/pws/index.php?id=css18](http://di.ics.upjs.sk/vyucba/pomocne_materialy/pws/index.php?id=css18)]

```
<p style="text-transform:capitalize">prvé písmená slov sú veľké</p>
<p style="text-transform:lowercase">VŠETKY PÍSMENÁ SÚ MALÉ</p>
<p style="text-transform:uppercase">všetky písmená sú veľké</p>
```

## 1.23. Textové dekorácie

Textové dekorácie nastavujeme vlastnosťou `text-decoration`.

```
{ text-decoration: none | [ underline || overline || line-through || blink ] | inherit }
```

[ukážka: [http://di.ics.upjs.sk/vyucba/pomocne\\_materialy/pws/index.php?id=css19](http://di.ics.upjs.sk/vyucba/pomocne_materialy/pws/index.php?id=css19)]

```
<span style="text-decoration:underline">podčiarknuté</span>
<span style="text-decoration:overline">čiara nad textom</span>
<span style="text-decoration:line-through">prečiarknuté</span>
<span style="text-decoration:blink">blikajúce</span>
```

## 1.24. Výška riadku textu

Výšku riadku textu (riadkovanie) nastavíme pomocou `line-height`. Texty ktorých riadky sú príliš husté/riedke sú ťažko čitateľné.

```
{ line-height: normal | <number> | <length> | <percentage> | inherit }
```

[ukážka: [http://di.ics.upjs.sk/vyucba/pomocne\\_materialy/pws/index.php?id=css20](http://di.ics.upjs.sk/vyucba/pomocne_materialy/pws/index.php?id=css20)]

```
<p style="line-height:2">Odsek - riadkovanie 2. ...
<p style="line-height:3em">Odsek - riadkovanie 3em. ...
<p style="line-height:200%">Odsek - riadkovanie 200%. ...
```

## 1.25. Medzery

Medzerou myslíme medzeru medzi písmenami a slovami (napr.: pre zvýraznenie textu). Príliš veľké medzery medzi slovami a písmenami v texte narušujú súvislosť textu. Takéto texty sú ťažko čitateľné. Zvýrazniť takto môžeme len krátke úseky textu alebo nadpisy.

- medzery medzi písmenami

```
{ letter-spacing: normal | <length> | inherit }
```

- medzery medzi slovami

```
{ word-spacing: normal | <length> | inherit }
```

[ukážka: [http://di.ics.upjs.sk/vyucba/pomocne\\_materialy/pws/index.php?id=css21](http://di.ics.upjs.sk/vyucba/pomocne_materialy/pws/index.php?id=css21)]

```
<p style="letter-spacing:1em">Zväčšené medzery medzi písmenami.</p>
<p style="word-spacing:3em">Zväčšené medzery medzi slovami.</p>
```

## 1.26. Vertikálne zarovnanie

Objekty (obrázky, text, ...) môžeme zarovnať aj vertikálne.

```
{vertical-align: baseline | sub | super | top | text-top | middle | bottom |
text-bottom | <percentage> | <length> | inherit }
```

[ukážka: [http://di.ics.upjs.sk/vyucba/pomocne\\_materialy/pws/index.php?id=css22](http://di.ics.upjs.sk/vyucba/pomocne_materialy/pws/index.php?id=css22)]

```
<p><span style="vertical-align:sub">Text dole. </span>
Normálny text.
<span style="vertical-align:super">Text hore.</span>
</p>
<p>Zarovnanie obrázku na osnovu riadku.

</p>
<p>Zarovnanie obrázku na stred textu.

</p>
<p>Zarovnanie obrázku hore.

</p>
<p>Zarovnanie obrázku dole.

</p>
```

## 1.27. Zalomenie stránky

Ak si chceme vynútiť zalomenie stránky (napr. dosiahnuť aby sa pri tlačení text tlačil na ďalšom liste papiera), kaskádové štýly nám ponúkajú celkom tri možnosti. Zalomenie pred daným elementom `page-break-before`, za daným elementom `page-break-after` a vo vnútri elementu `page-break-inside`.

- zalomenie stránky pred elementom

```
{ page-break-before : auto | always | avoid | left | right | inherit }
```

- zalomenie stránky za elementom

```
{ page-break-after : auto | always | avoid | left | right | inherit }
```

- zalomenie stránky vo vnútri elementu

```
{ page-break-inside : avoid | auto | inherit }
```

Jednotlivé hodnoty majú nasledovný význam:

- auto – automaticky, bez vynútenia
- always - vždy

- avoid – zabráni zalomeniu stránky
- left – zalomí stránku a obsah nasledujúcej strany umiestni tak, aby bola formátovaná ako ľavá strana
- right - zalomí stránku a obsah nasledujúcej strany umiestni tak, aby bola formátovaná ako pravá strana
- inherit – zdedí vlastnosť od nadradeného elementu

[ukážka: [http://di.ics.upjs.sk/vyucba/pomocne\\_materialy/pws/index.php?id=css25](http://di.ics.upjs.sk/vyucba/pomocne_materialy/pws/index.php?id=css25)]

```
<p style="page-break-after:always;" >Strana 1</p>
<p>Strana 2</p>
<p style="page-break-before:always;" >Strana 3</p>
<p>Pozrite si ukážku pred tlačou.</p>
```

## 1.28. Testovanie štýlopisu

CSS sa postupne vyvíja a tak nie je nič výnimočné, že niektoré vlastnosti nie sú podporované v nejakom prehliadači. Otestovať štýlovanie webu môžeme urobiť nasledovne:

- skontrolujeme, či štýlopis vyhovuje špecifikácii CSS, využiť môžeme napr. online validátor na adrese: <https://jigsaw.w3.org/css-validator/>,
- otestujeme, ako sa stránka zobrazí v prehliadačoch, ktoré máme k dispozícii,
- otestujeme stránku aj v iných prehliadačoch (minoritné prehliadače, staršie verzie prehliadačov, prehliadače na iných operačných systémoch a pod.), využiť môžeme napr. službu <http://browsershots.org/>.

## 1.29. Písanie štýlov

Pri písaní (definovaní) štýlov narážame na problém nepreberného množstva vlastností a ich hodnôt<sup>8</sup>. Vedieť aké vlastnosti môžeme pri tom ktorom elemente použiť je prakticky nemožné. Jednou z možností je použiť špeciálne editory, ktoré nám túto prácu značne uľahčia (napr.: TopStyle Lite). Riešením môžu byť aj online generátory CSS štýlov. Aj štýly však majú definované jasné pravidlá. Skontrolovať si ich môžete napr.: CSS validátorom (<http://jigsaw.w3.org/css-validator/>).

Pri vytváraní dokumentu štýlov je vhodné rozumne komentovať jeho jednotlivé časti.

```
.nadpis {font-family:Arial, Helvetica, sans-serif}
/* globálny štýl nadpisov */
```

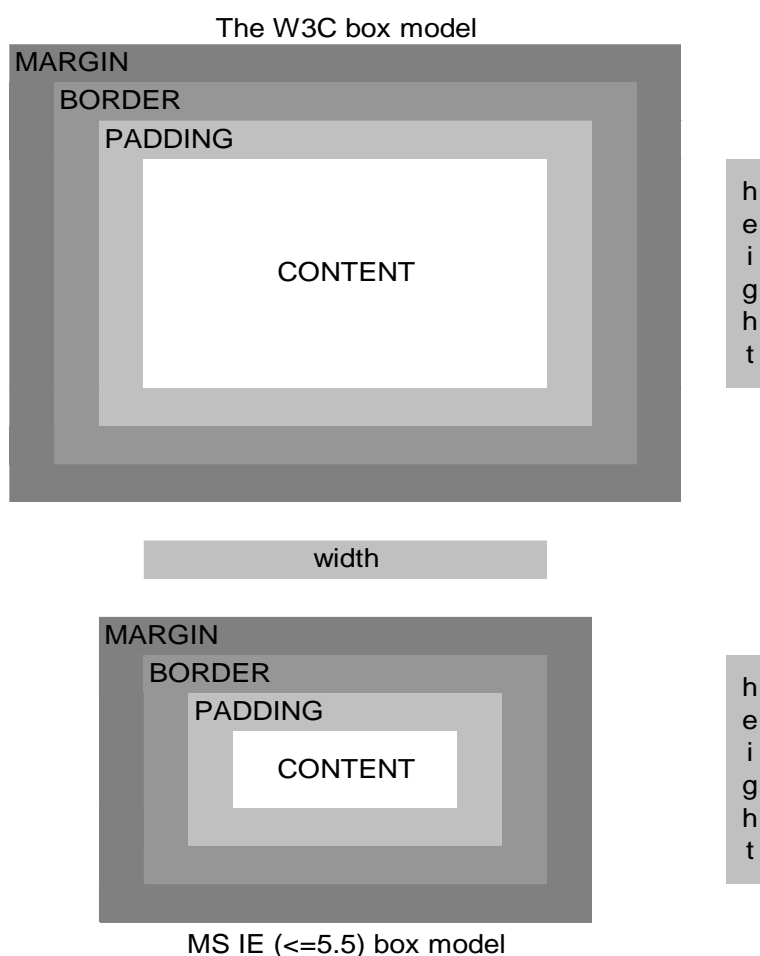
---

<sup>8</sup> Zoznam CSS vlastností nájdeme okrem oficiálneho štandardu napr. aj na <https://www.w3.org/Style/CSS/all-properties.en.html> alebo aj s príkladmi na <https://www.w3schools.com/cssref/>

## 2. Blokový model CSS

Vo vizuálnom CSS modeli majú všetky elementy vonkajší rámček, jeho vonkajšie a vnútorné odsadenie (aj keď to na prvý pohľad nie je viditeľné). Každý element je „teda“ vložený do obdĺžnika (boxu). Existujú dva druhy obdĺžnikov, blokové a vložené. Blokový obdĺžnik (napr. nadpis) zaberá celú dostupnú šírku (okna, stránky, ...), vložený najmenšiu možnú.

W3C definovala blokový model CSS tak, ako ho vidíte na obrázku nižšie. Ako sa to občas stane, jeho implementácia nebola už taká jednoduchá. MS so svojim prehliadačom Internet Explorer (do verzie 5.5) v snahe dosiahnuť spätnú kompatibilitu s už existujúcimi stránkami interpretoval box model po starom. Zatiaľ čo W3C rozmer (šírku, výšku) boxu definovala ako veľkosť samotného obsahu, IE do rozmeru boxu zarátava aj vnútorné odsadenie a šírku hranice boxu (viď. obrázok).<sup>9</sup>



Existuje niekoľko riešení tohto problému. Asi najznámejšie z nich sa nazýva Matrijoška hack. Spočíva v obalení bloku ďalším blokom, ktorý má nulové okraje, nulové orámovanie a nulové odsadenie a jeho šírka (výška) je pevne definovaná. Blok, ktorý sa nachádza vo vnútri takto definovaného bloku majú svoje maximálne rozmery určené práve týmto

---

<sup>9</sup> Poznámka: príklad s rôznou interpretáciou veľkosti rámčeka chápme skôr ako ukážku toho, že aj definícia v štandarde nie je zárukou 100% implementácie. Konkrétne tento problém dnes prakticky neexistuje (kto ešte používa IE 5.5?).

elementom (bez ohľadu na to, či sa do ich rozmeru započítavajú aj okraje a odsadenie alebo nie)

V nasledujúcej ukážke je len kvôli názornosti pozadie obaľujúceho elementu šedé. Pri praktickom použití je tento element „neviditeľný“.

[ukážka: [http://di.ics.upjs.sk/vyucba/pomocne\\_materialy/pws/index.php?id=css9](http://di.ics.upjs.sk/vyucba/pomocne_materialy/pws/index.php?id=css9)]

```
<div style="width:15cm; height:10cm;
        margin:0; border:0; padding:0;
        background-color: #DDDDDD">
  <div style="margin:1cm; border:1cm; padding:1cm;
        border-color:#FF0000; border-style:solid">
    Text v blokovom elemente. ...
  </div>
</div>
```

## 2.1. Vonkajšie okraje

Vonkajší okraj (margin) sa definuje naraz pre celý box, alebo pre každú stranu boxu zvlášť.

- vonkajšie okraje globálne

```
{ margin: hodnota<jednotky> | inherit }
```

- vonkajšie okraje jednotlivo

```
{ margin-top: hodnota<jednotky> | inherit;
  margin-right: hodnota<jednotky> | inherit;
  margin-bottom: hodnota<jednotky> | inherit;
  margin-left: hodnota<jednotky> | inherit }
```

## 2.2. Vnútný okraj, odsadenie

Vnútný okraj (odsadenie, padding) sa definuje naraz pre celý box alebo pre každú stranu zvlášť.

- vnútorné okraje globálne

```
{ padding: hodnota<jednotky> | inherit }
```

- vnútorné okraje jednotlivo

```
{ padding-top: hodnota<jednotky> | inherit;
  padding-right: hodnota<jednotky> | inherit;
  padding-bottom: hodnota<jednotky> | inherit;
  padding-left: hodnota<jednotky> | inherit }
```

## 2.3. Šírka hranice rámčeka

Šírka rámčeka (hranica, border-width) sa definuje naraz pre celý box alebo pre každú stranu zvlášť.

- šírka rámčeka globálne

```
{border-width: hodnota<jednotky> | inherit}
```

- šírka rámčeka jednotivo

```
{ border-top-width: hodnota<jednotky> | inherit;
  border-right-width: hodnota<jednotky> | inherit;
  border-bottom-width: hodnota<jednotky> | inherit;
  border-left-width: hodnota<jednotky> | inherit }
```



## 2.4. Farba hranice rámčeka

Farba rámčeka (border-color) sa definuje naraz pre celý box alebo pre každú stranu zvlášť.

- farba rámčeka globálne

```
{ border-color: farba | inherit }
```

- farba rámčeka jednotlivito

```
{ border-top-color: farba | inherit;
  border-right-color: farba | inherit;
  border-bottom-color: farba | inherit;
  border-left-color: farba | inherit }
```

## 2.5. Štýl čiary hranice rámčeka

Štýl čiary rámčeka sa definuje naraz pre celý box alebo pre každú stranu zvlášť. Prípustných je niekoľko typov:

- none - žiaden
- hidden - skrytý
- dotted - bodkovaný
- dashed - čiarkovaný
- solid - plná čiara
- double - dvojité čiary
- groove – 3D žliabok
- ridge – 3D rám
- inset – osvetlený 3D žliabok
- outset – osvetlený 3D rám

- štýl rámčeka globálne

```
{ border-style: <typ> | inherit }
```

- štýl rámčeka jednotlivito

```
{ border-top-style: <typ> | inherit;
  border-right-style: <typ> | inherit;
  border-bottom-style: <typ> | inherit;
  border-left-style: <typ> | inherit }
```

[ukážka: [http://di.ics.upjs.sk/vyucba/pomocne\\_materialy/pws/index.php?id=css23](http://di.ics.upjs.sk/vyucba/pomocne_materialy/pws/index.php?id=css23)]

```
<div style="margin:20mm;
           padding-left:10mm;
           border-left-width:5mm;
           border-left-style:double;
           border-left-color:#FF0000;">
  Text umiestnený do boxu. ...
</div>
```

## 2.6. Súhrnná vlastnosť rámčeka

Ak chceme použiť rovnaké vlastnosti pre všetky štyri strany rámčeka (border), môžeme to spraviť skrátenou formou.

- štýl rámčeka

```
{ border: šírka štýl farba }
```

[ukážka: [http://di.ics.upjs.sk/vyucba/pomocne\\_materialy/pws/index.php?id=css24](http://di.ics.upjs.sk/vyucba/pomocne_materialy/pws/index.php?id=css24)]

```
<div style="border:2mm dotted #FF0000">  
  Text umiestnený do boxu. ...  
</div>
```

## 2.7. CSS Box sizing

Vlastnosť box-sizing umožňuje zahrnúť padding a border elementu do celkovej šírky a výšky elementu. Tento prístup je užitočný najmä v situáciách, keď potrebujeme s blokmi pracovať v celku (napr. pri vzájomnom rozmiestňovaní) a neriešiť že ku skutočnej veľkosti je potrebné ešte pripočítať vnútorné odsadenie a hranicu.

[[http://di.ics.upjs.sk/vyucba/pomocne\\_materialy/pws/index.php?id=css33](http://di.ics.upjs.sk/vyucba/pomocne_materialy/pws/index.php?id=css33)]

```
<div style="width:10cm; height:5cm; background-color:#FF0000; padding: 1cm;  
border: #00FF00 solid 1cm">šírka (10cm) a výška (5cm) je podľa definície w3c box  
modelu</div>  
<br />  
<div style="width:10cm; height:5cm; background-color:#FF0000; padding: 1cm;  
border: #00FF00 solid 1cm; box-sizing: border-box">v šírke (10cm) a výške (5cm)  
je zahrnuté aj padding a border-width</div>
```

### 3. Rozvrhnutie obsahu a obtekanie v CSS

Pod pojmom rozvrhnutie obsahu webovej stránky chápeme schému v CSS, ktorá umožňuje používať elementy k vytvoreniu blokov a ich umiestnenie v dokumente alebo okne prehliadača.

Pod pojmom obtekanie chápeme techniku v CSS, ktorá umožňuje umiestniť element vpravo alebo vľavo tak, aby text (okolitý obsah) obtekal okolo elementu.

#### 3.1. Rozvrhnutie obsahu

Rozvrhnutie obsahu môže byť:

- absolútne – pozícia k explicitnému kontajneru alebo koreňovému elementu, bloky sú vyňaté z „normálneho“ toku elementov
- relatívne – pozícia voči normálnemu toku elementov, bloky nie sú vyňaté z „normálneho“ toku elementov

```
{ position: relative | absolute | inherit }
```

Okrem toho môžeme definovať vzdialenosti od hrany boxu:

- left - určuje vodorovnú vzdialenosť ľavej hrany kontajnera od ľavej hrany obsahujúceho kontajnera
- right - určuje vodorovnú vzdialenosť pravej hrany kontajnera od pravej hrany obsahujúceho kontajnera
- top - určuje zvislú vzdialenosť hornej hrany kontajnera od hornej hrany obsahujúceho kontajnera
- bottom - určuje zvislú vzdialenosť spodnej hrany kontajnera od spodnej hrany obsahujúceho kontajnera

[ukážka: [http://di.ics.upjs.sk/vyucba/pomocne\\_materialy/pws/index.php?id=css27](http://di.ics.upjs.sk/vyucba/pomocne_materialy/pws/index.php?id=css27)]

```
.obsah {
  position: relative;
  left: 1cm;
  top: 1cm;
  width: 7cm;
  height: 7cm;
  border-color: #000000;
  border-style: solid;
}

.red {
  position: absolute;
  left: 1cm;
  top: 1cm;
  width: 2cm;
  height: 2cm;
  border: 1px;
  border-color: #FF0000;
  border-style: solid;
}

.green {
  position: relative;
```

```

left: 1cm;
top: 1cm;
width: 2cm;
height: 2cm;
border: 1px;
border-color: #00FF00;
border-style: solid;
}

```

```

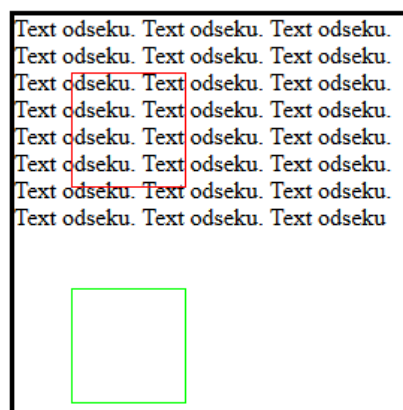
<link rel="stylesheet" type="text/css" href="css27.css" />
</head>
<body>
  <div class="obsah">
<p style="margin:0">...</p>
<div class="red"></div>
<div class="green"></div>
</div>

```

Kontajner (čierny box - obsah) je polohovaný relatívne. Svoju pozíciu (odsadenie 1cm zhora a zľava) si odvodil od nadradeného elementu – zobrazenej časti okna.

Červený box je polohovaný absolútne. Svoju pozíciu si odvodil od nadradeného čierneho boxu „obsah“. Jeho pozícia je vyňatá z normálneho toku elementov. Vôbec neberie do úvahy odsek textu pred ním.

Zelený box je polohovaný relatívne. Svoju pozíciu si odvodil od nadradeného čierneho boxu „obsah“. Jeho pozícia rešpektuje normálny tok elementov. Svoje odsadenie si odvodil nie len od nadradeného elementu, ale aj od predchádzajúceho odseku a červeného boxu.



### 3.2. Obtekanie elementov

Pomocou obtekania vieme nastaviť spôsob obtekania. Je potrebné si uvedomiť, že **nastavujeme, ako bude daný element (blok) obtekať ostatné**, nie to, ako budú obtekať oni jeho. Obtekanie nastavujeme vlastnosťou float. Použiť môžeme hodnoty:

- left - obtekanie zľava, element bude obtekať ostatné elementy zľava
- right - obtekanie sprava, element bude obtekať ostatné elementy sprava

```

{ float: left | right | none | inherit }

```

[ukážka: [http://di.ics.upjs.sk/vyucba/pomocne\\_materialy/pws/index.php?id=css28](http://di.ics.upjs.sk/vyucba/pomocne_materialy/pws/index.php?id=css28)]

```

...
<style type="text/css">
<!--
  .obsah { position:relative;
           left:1cm; top:1cm;
           width:7cm;
           height:7cm;
           border-style:solid;

```

```

    }

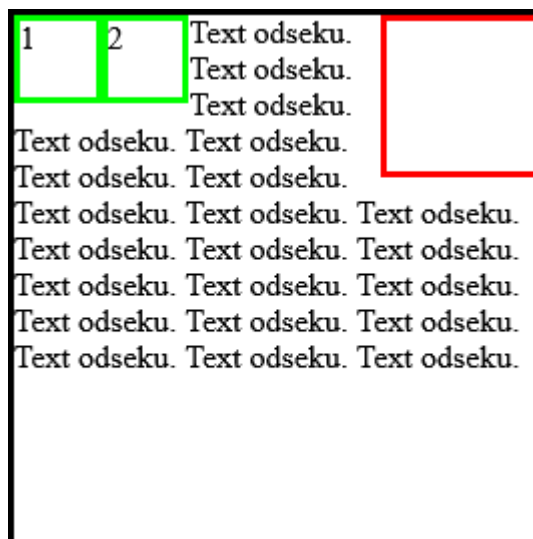
    .green { float:left;
             width:1cm; height:1cm;
             border-style:solid;
             border-color:#00FF00;
            }

    .red { float:right;
           width:2cm; height:2cm;
           border-style:solid;
           border-color:#FF0000;
          }
//-->
</style>
<body>
  <div class="obsah">
    <div class="red"></div>
    <div class="green">1</div>
    <div class="green">2</div>
    <p style="margin:0">Text odseku. ...</p>
  </div>

```

Zelené boxy majú nastavené obtekane zľava. Sú teda umiestnené tak, aby ostatné elementy obtekal zľava. Všimnime si, že aj zelený box 1 obteká zelený box 2 tiež zľava.

Červený box má nastavené obtekanie sprava. Je teda umiestnený tak, aby ostatné elementy obtekal sprava.



### 3.3. Zrušenie obtekania

Zrušenie obtekania znamená zrušenie obtekania okolitého obsahu. Element (blok) bude stále zarovnaný na danú stranu, ale už nebude obtekať iné elementy. Zrušenie obtekania je možné vlastnosťou `clear`.

- `left` – zrušenie obtekania zľava
- `right` - zrušenie obtekania sprava
- `both` - zrušenie obtekania z oboch strán

```
{ clear: none | left | right | both | inherit }
```

Ak zrušíme obtekanie odseku textu v našom príklade, výsledok bude nasledovný:

[ukážka: [http://di.ics.upjs.sk/vyucba/pomocne\\_materialy/pws/index.php?id=css29](http://di.ics.upjs.sk/vyucba/pomocne_materialy/pws/index.php?id=css29)]

```
...
<style type="text/css">
<!--
  .obsah { position:relative;
           left:1cm; top:1cm;
           width:7cm;
           height:7cm;
           border-style:solid;
         }

  .green { float:left;
           width:1cm;
           height:1cm;
           border-style:solid;
           border-color:#00FF00;
         }

  .red { float:right;
         width:2cm;
         height:2cm;
         border-style:solid;
         border-color:#FF0000;
       }
//-->
</style>
<body>
  <div class="obsah">
    <div class="red"></div>
    <div class="green">1</div>
    <div class="green">2</div>
    <p style="margin:0; clear:left">Text odseku. ...</p>
  </div>
```

Odseku sme zrušili obtekane zľava. Zelené boxy už neobtekajú odsek. Keďže obtekanie sprava je stále v platnosti, červený box je obtekaný naďalej.

1	2	
Text odseku.	Text odseku.	
Text odseku.	Text odseku.	
Text odseku.	Text odseku.	
Text odseku.	Text odseku.	Text odseku.
Text odseku.	Text odseku.	Text odseku.
Text odseku.	Text odseku.	Text odseku.
Text odseku.	Text odseku.	Text odseku.
Text odseku.		



### 3.4. Os z

Aj keď pracujeme v rovine, CSS nám ponúka možnosť navzájom (v priestore) prekryvať elementy. Pozíciu elementu vieme nastaviť aj vzhľadom na os z. Slúži na to vlastnosť `z-index`. Čím vyššie číslo `z-index-u`, tým je objekt vyššie.

```
{ z-index: auto | <integer> | inherit }
```

[ukážka: [http://di.ics.upjs.sk/vyucba/pomocne\\_materialy/pws/index.php?id=css30](http://di.ics.upjs.sk/vyucba/pomocne_materialy/pws/index.php?id=css30)]

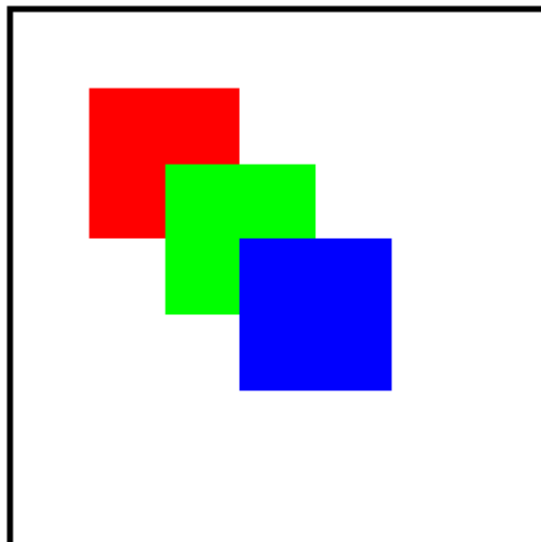
```
...
<style type="text/css">
<!--
  .obsah { position:relative;
           width:7cm;
           height:7cm;
           border-style:solid;
         }

  .red { position:absolute;
         z-index:1;
         left:1cm;
         top:1cm;
         width:2cm;
         height:2cm;
         background-color:#FF0000;
       }

  .green { position:absolute;
           z-index:2;
           left:2cm;
           top:2cm;
           width:2cm;
           height:2cm;
           background-color:#00FF00;
        }

  .blue { position:absolute;
          z-index:3;
          left:3cm;
          top:3cm;
          width:2cm;
          height:2cm;
          background-color:#0000FF;
        }
//-->
</style>
<body>
  <div class="obsah">
    <div class="blue"></div>
    <div class="green"></div>
    <div class="red"></div>
  </div>
```

Aj keď sme elementy uviedli v poradí blue-green-red, ich zobrazenie akceptuje hodnotu z-index.



### 3.5. CSS Flexbox

Pomerne komplikované rozmiestňovanie elementov (layout) v okne prehliadača je možné riešiť aj jednoduchšie. Slúži k tomu Flexbox layout modul<sup>10</sup>. Ide o flexibilný nástroj pre vytváranie layoutu bez využitia obtekania (float) a pozicionovania (position). Jeho použitie je pomerne jednoduché. Potrebujeme definovať rodičovský element (kontajner) s vlastnosťou `display: flex`. Do rodičovského elementu potom umiestnime prvky, pri ktorých vyžadujeme nejaké rozmiestnenie v rámci rodičovského elementu.

[ukážka: [http://di.ics.upjs.sk/vyucba/pomocne\\_materialy/pws/index.php?id=css35](http://di.ics.upjs.sk/vyucba/pomocne_materialy/pws/index.php?id=css35)]<sup>11</sup>

Pre kontajner môžeme definovať rôzne vlastnosti:

- `display: flex`  
vygeneruje sa flexibilný blok, flex kontajner,
- `flex-direction: row|row-reverse|column|column-reverse`  
spôsob zobrazenia položiek v kontajneri, v riadku | v riadku sprava doľava | v stĺpci | v stĺpci zdola hore,
- `flex-wrap: wrap|nowrap|wrap-reverse`  
položky v kontajneri sa môžu | nemôžu zalamovať | zalomia sa reverzne,
- `justify-content: flex-start|flex-end|center|space-between|space-around|space-evenly`  
horizontálne zarovnanie položiek v kontajneri, od začiatku | od konca | na stred | rovnaký priestor medzi položkami | každá položka má okolo seba rovnaký priestor | medzi položkami a krajnou položkou a okrajom kontajnera je rovnaký priestor,
- `align-items: flex-start|flex-end|center|baseline|stretch`  
horizontálne zarovnanie položiek v kontajneri, hore | dole | na stred | na základnú čiaru textu položiek | na celú výšku kontajnera,
- `align-content: space-between|space-around|stretch|center|flex-start|flex-end`

<sup>10</sup> <https://www.w3.org/TR/css-flexbox-1/>

<sup>11</sup> Meňte šírku okna, aby ste daný efekt lepšie postrehli.

horizontálne zarovnanie obsahu kontajnera (ako celku), rovnaký priestor medzi položkami | rovnaký priestor okolo každej položky | položka sa natiahne na maximálnu výšku | položky ako celok sú centrované | položky ako celok sú zarovnané hore | položky ako celok sú zarovnané dole,

Aj pre jednotlivé položky kontajnera môžeme definovať, ako sa majú správať voči ostatným položkám kontajnera.

[ukážka: [http://di.ics.upjs.sk/vyucba/pomocne\\_materialy/pws/index.php?id=css36](http://di.ics.upjs.sk/vyucba/pomocne_materialy/pws/index.php?id=css36)]<sup>12</sup>

- `flex-grow: cislo`  
ako rýchlo sa má položka zväčšovať vzhľadom na ostatné položky,
- `flex-shrink: cislo`  
ako rýchlo sa má položka zmenšovať vzhľadom na ostatné položky,
- `flex-basis: šírka`  
definuje základnú šírku položky.

---

<sup>12</sup> Meňte šírku okna, aby ste daný efekt lepšie postrehli.

## Použité a ďalšie zdroje

- Cascading Style Sheets, level 2 revision 1  
<http://www.w3.org/TR/CSS21/>
- Appendix F. Full property table  
<http://www.w3.org/TR/CSS21/propidx.html>
- CSS Tutorial,  
<http://www.w3schools.com/css/default.asp>
- CSS - Kaskádové styly  
<http://www.jakpsatweb.cz/css/>
- Stručný opis kaskádneho štýlu CSS1  
<http://www2.fiit.stuba.sk/~kapustik/SPECIF/CSS1/index.html>
- CSS Properties  
<http://www.htmlhelp.com/reference/css/properties.html>
- QrONE CSS Designer  
<http://www.qrone.org/cssdesigner.html>
- TopStyle Lite  
<http://www.newsgator.com/NGOLProduct.aspx?ProdId=TopStyle&ProdView=lite>
- The W3C CSS Validation Service:  
<http://jigsaw.w3.org/css-validator/>
- Přehled vlastností CSS  
<http://www.jakpsatweb.cz/css/css-vlastnosti-hodnoty-prehled.html>
- CSS | WebTvorba  
<http://www.webtvorba.cz/css/>
- Properties : CSS 2.1 Reference : Cultured Code:  
<http://www.culturedcode.com/css/reference.html>  
(download offline:  
[http://www.culturedcode.com/css/Cultured\\_Code\\_CSS\\_Reference.zip](http://www.culturedcode.com/css/Cultured_Code_CSS_Reference.zip))
- css Zen Garden: The Beauty in CSS Design  
<http://www.csszengarden.com>