

**<hatmatilka>**

****

**<author>**Mgr. Marián Staňo**</author>**

**<title>**učebnica HTML a CSS**</title>**

**</hatmatilka>**

...že nerozumieš,  
čo to znamená?

Potom je  
táto knižka  
práve pre teba.

## Hatmatilka

HTML stránky vymyslel pán Tim Berners-Lee v roku 1989. Ako vedec v ústave CERN vo Švajčiarsku sa usiloval nájsť jednoduchý spôsob na publikovanie a výmenu informácií medzi vedcami.

Už v tom čase bolo v internete zapojených mnoho počítačov, ktoré sa líšili jednak hardvérovou konfiguráciou, ale aj softvérom a operačným systémom( atak je to v internete dodnes).

Niektoré systémy napríklad nedokážu zobrazovať grafiku. Pán Lee to musel vymyslieť tak, aby sa na každom zariadení zobrazilo, pokiaľ to bude možné, čo najviac podstatných informácií. Učene sa tomu hovorí, že stránky nie sú závislé na zobrazovacom zariadení.

A našiel takýto recept:

1. Vezmeme jazyk SGML od firmy IBM (Standard Generalized Markup Language = Štandardný zovšeobecnený značkovací jazyk) a kúsok z neho odkrojíme.
2. Tento kúsok premiesime a dobre rozvaľkáme.
3. Dostaneme jazyk veľmi jednoduchučký. Nazveme ho HTML, na počesť toho, že umožňuje, aby rôzne časti textu boli poprepájané cez odkazy. (HyperText Markup Language = Hypertextový značkovací jazyk).

Vďaka svojej jednoduchosti a šikovnosti sa HTML čoskoro prerazilo. Zároveň pán Lee vytvoril protokol HTTP, pomocou ktorého si počítače posielajú stránky medzi sebou. Protokol HTTP a značkovací jazyk HTML spolu tvoria službu, ktorá dostala poetické meno: *World Wide Web* (celosvetová pavučina, skrátene *Web*).

V tejto knižke nájdeš a naučíš sa:

- navrhnuť si základnú štruktúru webstránky
- zapísať stránku v jazyku HTML
- vytvárať pravidlá kaskádových štýlov (CSS), ktoré slúžia na prezentáciu stránky
- oživiť stránku jednoduchým skriptom.

## Čo budeme potrebovať

Nie je toho veľa. Niektoré programy máš v počítači, niektoré sú voľne dostupné v internete.

**Internetový prehliadač.** Dnes býva predinštalovaný v každom operačnom systéme. Okrem klasiky Internet Explorer existujú Mozilla Firefox\* a Opera (Windows aj Linux), Safari (Windows a Mac OS X), Konqueror (Linux) a mnoho ďalších.

**Textový editor.** HTML a štýly sú uložené v textových súboroch, t.j. v súboroch, ktoré obsahujú len znaky. Najjednoduchší program tohto typu je Poznámkový blok (pozor, nie Word). Zložitejšie sú napríklad Dreamweaver alebo NVU, ale tem my používať nebudeme, lebo častokrát v snahe pomôcť tvorcovi stránky, mu tam dopíše veci bez jeho vedomia.

Niekde medzi týmito dvoma extrémami sú editory PSPad, BlueFish, Quanta, HTML-kit a podobne. Takýto editor je pre nás najvhodnejší.

Samozrejme, môžeš použiť aj iný (tvoj obľúbený) editor, dôležité je, aby vedel zapísať obyčajný textový súbor.

**Grafický editor** v prípade, že si chceš stránku skrášliť grafikou alebo animáciou. My budeme používať voľne dostupný vektorový editor InkScape a rastrový editor GIMP.

---

\* *Firefox* je aglické pomenovanie pre *pandu červenú*, čo je druh pandy s ryšavou srstou. Toto zvieratko je maskotom projektu Mozilla Firefox.

# **O ■ O čom bude moja stránka? alebo „Z peknej, ale prázdnej misy sa nenaješ“.**

Prv ako začneš písať, mal by si dobre zvážiť, **o čom**. Možno ti to nepríde vôbec podstatné, ale téma stránky je najdôležitejšia vec zo všetkého.

Stránka nemôže byť *o všetkom naraz*, takú by asi nikto nezvládol napísať a už vôbec nie udržiavať. (Okej, možno ty si výnimka, ale aj tak ťa prosím, najprv to skúsme s jednou témou :-)

A naopak – nič nie je *trápnejšie* ako pekná stránka plná táranín. Čo na svojej ponúkneš ty? Máš niekoľko možností:

- **Osobná (domovská)** – predstav seba, svoju rodinu, školu, tvorbu a záujmy.
- **Denník (blog)** – podel' sa s inými o svoje myšlienky a postrehy, prípadne literárnu tvorbu.
- **Hobby** – Vlastníš zbierku? Venuješ sa nejakému športu? Chováš zvieratko? Alebo vo svojom voľnom čase robíš niečo zaujímavé? Podel' sa o skúsenosti, alebo sa pochváľ kamarátom!
- **Projekt** – Robíš projekt? Venuješ sa dakej téme naozaj hlbkovo? Takýmto spôsobom môžeš prezentovať najnovšie výsledky.
- **Náučná** – Chcel by su ostatných niečo naučiť? Máš doma kopu referátov a chcel by si sa o ne podeliť?
- **Fanúšikovská** – Máš idol: speváka, herca, športovca? Zhromažďuješ o ňom všetky dostupné informácie?
- **???** – samozrejme, stránka môže byť aj na inú tému. Záleží len na tebe a tvojej fantázii, čo pekné a nové si vymyslíš.

Tak čo, máš v tom už jasno? Áno? Potom je tu pre teba niekoľko úloh. Výsledky tejto úlohy si zapíšeš, je to už zárodok tvojho vlastného projektu. Hneď ako zvládneme najzákladnejšie základy, začneme s jeho vývojom.

**Úloha 1:** O čom bude tvoja stránka? .....

**Úloha 2:** Pre koho bude stránka určená? .....

**Úloha 3:** Zakresli jednoduchú schému, z akých častí bude tvoja stránka pozostávať a ako budú medzi sebou poprepájané. (Použi obdĺžničky, ktoré pomenuješ a prepojenia medzi nimi naznačíš čiarami).

## 1. lekcia: „Ahoj svet!“ alebo Zdravíme internet.

Prvý pokrik programátora, ktorý sa naučí základ nového jazyka, býva „**Hello, world!**“, teda „Ahoj, svet!“ Počas štúdia ho aj ty budeš kričať na plné pecky ešte niekoľkokrát. Znamená asi toľko: „Som tu a týmto pokrikom chcem dať o sebe vedieť!“

**Úloha 1:** Vo svojom domovskom priečinku vytvor podpriečink a nazvi ho napr. *html*. V program PSPad alebo Poznámkový blok napíš:

```
Hello, world!
```

A ulož to do vytvoreného priečinku pod názvom *hello.html*. Nezabudni zadať aj príponu, lebo programy ako Poznámkový blok k súborom bez prípony automaticky pridávajú príponu *.txt*!

### Základná štruktúra

Tak. Pozdraviť už vieme. Ale toto ešte nie je HTML. Každá dokument musí mať svoju štruktúru, aby sa v ňom počítač vyznal a vedel ho správne zobrazíť.

**Úloha 2:** Do súboru *hello.html* pridaj tieto riadky:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html>
<body>
    Hello, world!
</body>
</html>
```

Tie ďalšie zvláštne „zobákové“ slovíčka sa nazývajú **značky**. Každá značka pozostáva z dvoch častí. Prvá je **otvárajúca** a druhá **zatvárajúca**. Názvy značiek sa píše malými písmenami a líšia sa navzájom len znakom „/“ (lomka). Vo vnútri býva zvyčajne nejaký text alebo ďalšie značky.

Sú však aj značky, ktoré zostávajú prázdne, a také sa zapisujú v skrátenej podobe, napríklad značka **<br />**. Všimni si, že v tomto prípade *pred lomkou píšeme medzeru*.

V našom príklade máme hneď niekoľko značiek, tak sa im poďme pozrieť na zobáky :^)

**<!DOCTYPE**... hovorí prehliadaču o tom, v akej verzii jazyka je dokument napísaný, aby ho vedel správne vykresliť. Staršie dokumenty ju nepoznali, ale v najnovších verziách HTML je povinná. Keby ju náš dokument neobsahoval, prehliadač by kreslil stránku vo veľmi starom „quirks“ režime, ktorý nedodržiava totiž webové štandardy, a preto by naša stránka vyzerala v niektorých prehliadačoch veľmi zvláštne.

Ďalšia v poradí je značka **html**. Nachádza sa hneď za **!DOCTYPE** a vlastne hovorí, že to, čo je v nej zapísané, je dokument v HTML jazyku.

Značka **body** označuje viditeľnú časť (telo) dokumentu. Všetko, čo do nej napíšeme, sa objaví ako webstránka v okne prehliadača.

Nadpisy, odstavce, obrázky, odkazy... a mnohé iné veci vidíme na každej stránke. Sú to vlastne *základné kamene*, z ktorých je stránka vystavaná. Súhrnne ich nazývame slovom **elementy**. Aký je vlastne rozdiel medzi značkou a elementom?

Značka je len symbol, ktorý označuje začiatok a koniec elementu. Element je nielen značka, ale značka plus obsah v nej zapísaný. Čiže napr.

`<title></title>` je značka, ale `<title>Moja prvá stránka</title>` je element.

**Úloha 3.** Dopíšme ďalšie riadky.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html>
<head>
    <title>Moja prvá stránka</title>
</head>
<body>
    Hello, world!
</body>
</html>
```

Ak sa ti na stránke sa ti namiesto mäčkových zobrazili nejaké zmätené znaky, preštuduj si začiatok ### 8. lekcie o Meta-informáciách.

Značka **head** označuje hlavičku dokumentu. Do hlavičky vkladáme informácie o obsahu dokumentu. Prvou a najdôležitejšou značkou je **title**, čo je vlastne názov dokumentu. Keď si príklad zobrazíte v prehliadači, hore v titulkovom pruhu uvidíte práve obsah značky **title**.

Načo nám názov a informácie o dokumente vlastne sú? Aby ste to lepšie pochopili, stačí, keď si predstavíte zoznam stránok, ktoré vyhodí Google, keď niečo hľadáte – orientujete sa vo výsledkoch na základe názvov a popisov stránok – to sú vlastne informácie, ktoré si Google „vytiahol“ z hlavičky dokumentu.

Tak, to by sme mali základné HTML značky, ktoré musí obsahovať každý HTML dokument. V ďalšej lekcii si ukážeme ako písať zložitejší text.

## 2. lekcia: Text – nadpisy a odstavce

### Plníme stránku textom

Úloha 1. Dopíšme do našej stránky:

```
<body>
    Hello, world!
    Zdraví ťa Jožko Mrkvička.
</body>
```

Keď si stránku obnovíme v prehliadači, zistíme, že sa nám text zobrazí zosypaný na jeden riadku. Prehliadaču všetky konce riadkov statočne a vytrvalo ignoruje. Rovnako sa vykašle aj na to, keď mu napíšeme niekoľko medzier za sebou, napríklad Zdraví ťa      Jožko Mrkvička.

Ak trváme na tom, aby sa text zobrazil v odstavcoch, musíme prehliadaču začiatok a koniec odstavca vyznačovať:

```
<p>Hello, world!</p>
<p>Zdraví ťa Jožko Mrkvička.</p>
```

Značkou **p** (*paragraph* = *odstavec*) rozčleníme text na odstavce. V starších verziách malo HTML úľavy pre lenivých, takže sa nemusela písať zatvárajúca značka odstavca. Najnovšia verzia však vyžaduje zapísanie celej značky, keď zatváraciu časť vynecháme, tak sa to bude pokladať za chybu!

Niekedy, napríklad keď píšeme poéziu, môžeme vetu rozdeliť do veršov pomocou značky **br**:

```
<p>Zleteli orly z Tatry,<br />
tiahnu na podolia,<br />
ponad vysoké hory,<br />
ponad šíre polia.</p>
```

Značka **br** (*line break*) sa často *zneužíva* a používa sa aj tam, kde by bolo lepšie použiť **p**, teda na ukončovanie odstavcov. Ak členíš text na odstavce, pravdepodobne potrebuješ presne to, čo robí značka **p**. Keď budeš namiesto nej používať **br**, vypomstí sa ti to už pri písaní kaskádových štýlov pre odstavce.

Vo vnútri odstavcov používame ďalšie značky. Najmä, keď treba nejaké slovo *zdôrazniť*, prípadne *silne zdôrazniť*. Používa sa na značka **em** (*emphasize* = *dôraz*), na silné zdôraznenie značka **strong** (*strong emphasize* = *silný dôraz*). V texte to vyzerá napríklad takto:

Väčšina prehliadačov vypíše zdôraznené slová *kurzívou*, silne zdôraznené **tučným** písmom. Tieto vlastnosti môžeme pomocou CSS zmeniť na akékoľvek silnenviemepredstaviť.

Ak ste mali doteraz vo zvyku používať na to značky **i** a **b**, mal by ste namiesto nich používať **em** a **strong**, pretože namiesto toho, ako daný efekt vyzerá (kurzíva, tučné = prezentácia), udávajú význam textu (dôraz, silný dôraz), a prezentáciu nechávajú na konkrétny prehliadač alebo používateľa. Vezmite si napríklad nevidiaceho, ktorý si prezerá vašu stránku na braillovskom riadku – čo si má predstaviť pod pojmom *tučná kurzíva*?

## Nadpisy (a ich pocit dôležitosti)

Rovnako ako odstavce, svoju značku majú aj nadpisy. Nadpisy majú v rámci textu rôznu dôležitosť (Např. názov práce, názov kapitoly, podkapitola, podpodkapitola, ... atď.), preto aj existuje viacero značiek. Sú to značky **h1**, **h2**, **h3**, **h4**, **h5** a **h6**, čo je spolu 6 rôznych úrovní.

Nadpis **h1** je na stránke najväčší boss. Ostatné nadpisy sú jeho „podriadení“, platí: čím menšie číslo, tým menšia dôležitosť. Dvaja kohúti na jednom smetisku sa nezniesú a začnú na seba útočiť (to isté platí o dvoch bossoch v jednom rajóne), preto aj hlavný nadpis **h1** by mal byť na dobre štruktúrovanej stránke *iba jeden*.

Pre ostatné nadpisy platí, že by mali byť tam, kde je ich miesto. Teda, ak chceme nadpis v podkapitole na druhej úrovni (**h2**), na nadpisy použijeme značku **h3**.

Príklad použitia nadpisov:

**<h1>Jožove stránky!!!</h1>**

**<h2>Hello, world!</h2>**

**<p>Zdraví ťa Jožko Mrkvička.</p>**

**<h2>Z básnickej tvorby slovenských klasikov</h2>**

**<h3>Samo Chalupka: Mor ho!</h3>**

**<p>Zleteli orly z Tatry,<br />**

**tiahnu na podolia,<br />**

**ponad vysoké hory,<br />**

**ponad šire polia.</p>**

Pre každú úroveň nadpisov má prehliadač nastavené istú veľkosť a typ písma. Toto neskôr budeme môcť zmeniť podľa našej chuti. Keďže to je vec prezentácie a nie štruktúry, venujeme sa tomu v lekciách o CSS.

## Vnáranie alebo Raz, dva tri... ponoriť!

Keď kombinujeme viacero značiek dohromady, hovorí sa tomu, že vnárame značky jednu do druhej. Pri tom treba dať pozor, aby boli značky do seba správne vnorené. To znamená, že ak je



nejaká značka začne vo vnútri inej, musí skončiť skôr ako tá iná. Príklad *nesprávneho* vnorenia značiek:

```
<p>Nezmyselná <em>kurzíva</p></em>
```

Správne by malo byť:

```
<p>Nezmyselná <em>kurzíva</em></p>
```

Prečo o tom hovoríme? Správne vnorenie je dôležité pri zápise zložitejších značiek, napríklad značky pre zoznamy.

## Zoznamy

**Zoznamy** slúžia na vymenovanie možností. Delia na 3 typy: **neusporiadané**, **usporiadané** a **definičné**. My si zatiaľ ukážeme prvé dva.

Ak nám nezáleží na poradí, v ktorom hodnoty vymenujeme, používame **ul** – **neusporiadaný zoznam** (unordered list):

```
<ul>
  <li>Kov</li>
  <li>Drevo</li>
  <li>Papier</li>
</ul>
```

Každá položka zoznamu je zapísaná značkou **li** (list item). Keď si zobrazíte tento príklad v prehliadači, uvidíte vedľa položiek aj odrážky. Ich vzhľad môžete meniť pomocou CSS.

**Usporiadaný zoznam** (ordered list) používame na vypísanie usporiadaných položiek. V hore uvedenom príklade stačí, keď vymeníte značku **ul** za **ol**. Hneď po obnovení stránky uvidíte vedľa položiek čísla 1., 2., 3... Samozrejme, nemusia to byť len čísla, prvky možno napríklad aj „opísmenkovať“.

Zoznamy sa dajú vnárať do seba a kombinovať usporiadané a neusporiadané:

```
<ul>
  <li>Kov
    <ol>
      <li>Železo</li>
      <li>Med</li>
      <li>Cín</li>
    </ol>
  </li>
```

```
<li>Drevo</li>
<li>Papier</li>
</ul>
```

V nasledujúcej lekcii sa pozrieme bližšie na **odkazy** a **obrázky**. Stretneme sa aj s naším prvým **atribútom**.

## Špeciálne znaky

Ak chceme v HTML zapísať **špeciálny znak** (znak, ktorý sa používa na html značky, je z „exotického“ jazyka alebo nie je na klávesnici), treba použiť špeciálny zápis – *html entitu*.

**Príklad.** Predstav si, že by si do html chcel zapísať nerovnicu:

$$2 < x < 4$$

Keby sme to zapísali presne takto, ako to vidíme, odkiaľ by prehliadač vedel, že to  $< x >$  si nemá vysvetľovať ako značku, ale ako znamienka nerovnosti? Preto mu to povieme práve pomocou html entít. Html entitu zapisujeme v špeciálnom tvare:

`&lt; x >`

Na začiatku je vždy *ampersand* (&), potom je *slovný zápis znaku* (bla) a napokon príde *bodkočiarka* (;). Podľa & a ; prehliadač rozpozná začiatok a koniec entity, a podľa slovného zápisu presne vie, ktorý znak sa má nakresliť.

Našu nerovnicu teda zapíšeme v html takto:

```
<p>2 + &lt; x > 4</p>
```

**Mnemotechnický zápis.** Slovný zápis znaku je mnemotechnický, t.j. taký, aby sa nám dobre pamätal. V našom prípade `&lt;` znamená *less than* – *menší ako*. Podobne `&gt;` znamená *greater than* – *väčší ako*. Je to vlastne význam znamienka nerovnosti, ktoré používame.

Každý znak zobraziteľný na počítači (sú ich tisíce) má podľa štandardu Unicode pridelený svoj kód. Niektoré potom majú pridelené aj mnemotechnické pomenovania. Kompletný zoznam html entít môžeš nájsť na internete (-> html entity), ale v praxi asi budeš väčšinou potrebovať len niekoľko:

`&amp;`      &      **ampersand**

`&lt;`      <      **menší ako**

`&gt;`      >      **väčší ako**

`&nbsp;`           **pevná (nerozdeliteľná) medzera – non breaking space**

`&#8212;`      —      **dlhá pomlčka** (niektoré entity majú iba číselný kód, žiadne slovo)

`&quot;`      "      **úvodzovky – quotation mark**

`&Auml;`      Ä      **veľké ä**

`&Ocirc;`      Ô      **veľké ô**

&copy; © **znak copyright**

&trade; ™ **obchodná značka**

&reg; ® **registrovaná (na patentovom úrade) obchodná značka**

**Úloha 5:** Zapiš, aby sa v okne prehliadača zobrazila ľubovoľný názov html entity a vedľa nej znak, ktorý táto entita predstavuje. V prehliadači by to malo vyzerat' takto:

&reg; ®

### 3. Odkazy a obrázky

Už vieme vyrobiť pomerne zaujímavo stavanú stránku. Stránka, podobne ako klasická knižka, obsahuje veľa textu a obrázkov. Oproti knihe však má jednu výbornú vlastnosť. Viacero stránok môže byť navzájom prepojených **odkazmi**. V takomto texte sa oveľa rýchlejšie dostaneme k informácii, ktorú hľadáme. Zväčša stačí jedno kliknutie. Preto ho aj nazývame **hypertext**.

#### Odkaz

Ako taký hypertextový odkaz vyrobíme? Používa sa na to značka **a** (anchor):

```
<a href="porsche911.html">Technické detaily Porsche 911</a>
```

Fiha, začína to byť troška zložitejšie. Ale nič nie je také zložité, aby sme to nezvládli. Značka **a** je zložitejšia tým, že má niekoľko povinných **atribútov**.

**Atribúty** sa zapisujú do otváraciej časti značky **a** majú vždy formu **atribút="hodnota"**. Hodnota atribútu sa vždy dáva do *úvodzoviek*. Ak má značka viacero atribútov, oddeľujeme ich od seba *medzerou*.

V našom prípade má značka **a** len jeden atribút **href** (hypertext reference). Hodnotou tohto atribútu je *adresa* stránky, ktorá sa má zobraziť v okne prehliadača po kliknutí. Ako odkaz sa označí text, ktorý je uzavretý v značke **a**. V tomto prípade je to text *Technické detaily Porsche 911*.

**Adresa** stránky môže byť **absolútna** alebo **relatívna**. V predchádzajúcom príklade vidíme použitie relatívnej adresy. Táto obsahuje iba cestu k stránke z aktuálneho priečinka. Keď sa stránka *porsche911.html*, nachádza tom istom priečinku ako *hello.html*, tak ako cestu píšeme len meno súboru.

Absolútna adresa zas obsahuje úplnú cestu k nejakému súboru na internete. Takouto adresou je napríklad

```
http://www.smnd.sk/stanom/hatmatilka.html.
```

Absolútne adresy používame zvyčajne v prípade, že odkazujeme niekde mimo nášho webu, relatívne adresy zas vtedy, keď odkazujeme niekde do našich stránok.

Cieľom odkazu nemusí byť vždy len HTML stránka. Môžeme odkázať na hocijaký súbor, ktorý sa v internete nachádza. Ak prehliadač tento súbor nedokáže sám zobraziť, ponúkne nám na výber: podhodíť inému programu (ktorý si s ním poradí), alebo ho uložiť na disk.

#### Záložky

Odkazy nemusia smerovať len na iné stránky alebo súbory. Môžu odkazovať aj na iné miesto v tej istej stránke – takéto odkazy nazývame **záložkami**. Stačí na to pridať atribút **id** do niektorej značky, napríklad do nadpisu:

```
<h2 id="kapitola2">2. Text - nadpisy a odstavce</h2>
```

Na inom mieste v tej istej stránke potom na záložku odkážeme takto:

```
<a href="#kapitola2">2. Kapitola: Text - nadpisy a odstavce</a>
```

Takých zložiek môže byť v dokumente koľko len chceme, každá sa však musí volať inak.

## Obrázky

Podobne ako odkazy vkladáme aj obrázky. Používa sa na to značka `img`.

```

```

Všimni si, že keďže je **img** prázdna značka (podobne ako **br**), píšeme ju s lomkou na konci!

Atribút **src** podobne ako **href** pri obrázkoch, obsahuje relatívnu alebo absolútnu cestu k obrázku. Atribút **alt** obsahuje povinný *alternatívny popis* obrázku pre tých, ktorí *nemôžu* alebo *nechcú* obrázky vidieť.

Je dobré uvádzať aj atribúty **width** a **height**, ktoré prehliadaču hovoria, koľko miesta si má pre obrázok vyhradiť. Inak si prehliadač musí zistiť jeho rozmery až v momente, keď sa obrázok načítava, čo pri veľkom počte obrázkov môže niekedy trvať dosť dlho a prejaví sa to „skákaním“ stránky.

Na internete sa používajú obrázky niekoľkých typov. Každý typ je určený na iné použitie, preto je dobré myslieť na to dopredu.

**Obrázky typu GIF a PNG** – používajú sa na kreslené obrázky, takzvanú čiarovú grafiku. Takéto obrázky vie formát GIF veľmi dobre komprimovať (zmenšiť). Používa len paletu s 256 farbami, preto sa nehodí na fotografie. Dnes sa namiesto GIF, ktorý bol dlhý čas licencovaný za poplatok, často používa slobodný alternatívny formát PNG.

**Obrázky typu JPG** – na zapamätanie obrazovej informácie používa matematické vzťahy. Navonok sa to prejavuje tak, že čím vyššiu úroveň kompresie nastavíme, tým menej detailov v obrázku zostane (= tým je kockatejší). A naopak: čím menšia je kompresie, tým bude súbor väčší. Tento formát sa mimoriadne hodí najmä na fotografie.

**Obrázky typu SVG** – vektorový formát (obrázky sa kreslia napríklad v programe InkScape). Vektorové formáty si nepamätajú zhľuky bodiek (ako GIF, PNG a JPG), ale matematické vyjadrenia objektov. Napríklad pre kružnicu si zapamätajú súradnice stredu a polomer, prípadne farbu čiary. Výhodou to má tú, že obrázok môžete zväčšiť ako chcete, vždy bude hladký, v najvyššej možnej kvalite. Vhodný je najmä na kreslenú grafiku ako ikonky, diagramy a podobne. Dnes už ho vie zobrazovať veľa prehliadačov, niektoré však ešte s chybami, to sa však o krátky čas určite zmení.

V ďalšej lekcii sa pozrieme na jednu z najzložitejších HTML tém, **tabuľky**.

## 4. Tabuľky

Tabuľky sa v minulosti často zneužívali na rozloženie elementov na stránke, na vytvorenie stĺpcov a grafického stvárnenia stránky. Bolo to však v časoch, keď ešte neexistovali kaskádové štýly CSS, a ani sa veľmi nemyslelo na to, aby boli stránky prístupné aj pre ľudí s obmedzeniami (zrakovo alebo sluchovo postihnutí, menej výkonné počítače a pomalé internetové pripojenie).

V ďalších lekcách si postupne ukážeme, ako vytvoriť dvoj- a trojstĺpcový dizajn stránok bez použitia tabuliek.

Pravý zmysel používania tabuliek je na prezentáciu tabuľkových údajov. Takže takto vyzerá jednoduchá tabuľka 3x3 políčka:

```
<table cols="3" rows="3">
  <tr>
    <td>Prvá</td>
    <td>Druhá</td>
    <td>Tretia</td>
  </tr>
  <tr>
    <td>Druhá prvá</td>
    <td>Druhá druhá</td>
    <td>Druhá tretia</td>
  </tr>
  <tr>
    <td>Tretia prvá</td>
    <td>Tretia druhá</td>
    <td>Tretia tretia</td>
  </tr>
</table>
```

Značka **table** ukazuje začiatok a koniec tabuľky. Nepovinné atribúty **cols** a **rows** udávajú počet stĺpcov (columns) a riadkov (rows). Každý riadok tabuľky je vyznačený osobitne značkou **tr** (table row) a v rámci každého riadku sú políčka tabuľky vyznačené **td** (table data).

V tabuľkách okrem buniek môžeme vyznačiť aj hlavičky stĺpcov, prípadne riadkov. Pridajme do vyššie uvedenej tabuľky hlavičky:

```
<table cols="4" rows="3">
  <tr>
    <th>Prvý stĺpec</th>
```

```
<th>Druhý stĺpec</th>
<th>Tretí stĺpec</th>
</tr>
<tr>
<td>Prvá</td>
<td>Druhá</td>
```

...

No a je to! Tak máme v tabuľke aj hlavičky. Pozrime sa ešte na to, ako možno viac políčok tabuľky zlúčiť do jedného. To si však ukážeme až v #### niektorej z ďalších kapitol.

```
<tr>
<td colspan="2">Prvá a druhá</td>
<td rowspan="2">Tretia a Druhá tretia</td>
</tr>
<tr>
<td>Druhá prvá</td>
<td>Druhá druhá</td>
</tr>
```

Okrem tabuliek sa v HTML nachádza ešte jedna zaujímavá skupina značiek: sú to značky na vytváranie **formulárov**. Sú to značky na vloženie elementov ako napríklad textové polia, rádio tlačidlá a zaškrŕavacie políčka, kombo boxy, tlačidlá a podobne.

Tieto značky sa používajú najmä pri práci so skriptami a databázou na strane servera. Bez skriptov ale formuláre veľký význam nemajú. Ak vás táto téma zaujíma, odporúčam vám štúdium Aninovej knižky *PHP a MySQL*, ktorá sa tvorbou HTML formulárov, PHP skriptov a databáz zaoberá do podrobností.

A tým sme prebrali to najnevyhnutnejšie minimum, ktoré potrebujete na písanie správnych a pekných HTML stránok. V ďalšej kapitole, na ktorú už isto nedočkavo čakáte, si ukážeme, *ako si našťylovať elementy pomocou CSS*.

## 4. lekcia: Šminky a farby

V predchádzajúcich lekciiach sme zvládli základy písania HTML, dnes si povieme, ako priradíme HTML elementom jednoduché štýly. Inými slovami, doteraz sme sa zaoberali *obsahom* a *štruktúrou* dokumentu, teraz sa budeme chvíľu zaoberať jeho *prezentáciou*. Prezentácia sa týka vecí ako veľkosť a rozmery, farba, písmo, umiestnenie, (ne)viditeľnosť a podobne.

Každému elementu môžeme určiť jeho štýl. A každý štýl sa skladá z pravidiel. Ak chceme zmeniť určitú vlastnosť nejakého elementu, napíšeme preň jednoduché pravidlo:

vlastnosť: hodnota

Vidíme, že takýto zápis toho s HTML veľa spoločného nemá. Aby sme ich nejako dali dohromady, pôjdeme na to cez atribút **style** (píšeme do súboru *hello.html*):

```
<h1 style="color: red">Jožove stránky!!!</h1>
```

Tomuto zápisu hovoríme aj **priamy štýl**, pretože sa vkladá priamo do značky. Je to však veľmi nevhodný spôsob, pretože miešame obsah dokumentu s jeho prezentáciou, preto by sme sa mu mali *zdáleka vyhýbať*.

Ďalšia nevýhoda priamych štýlov je to, že ak chceme meniť vlastnosti viacerých elementov naraz, musíme *ten istý štýl* zapísať do každej značky zvlášť. Také dokumenty sa potom ťažko udržiavajú, a preto dávame prednosť hromadným zápisom pravidiel:

```
h1 {  
    color: red  
}  
  
p {  
    color: navy  
}
```

Meno značky píšeme pred kučeravými zátvorkami a nazýva sa **selektor**. Do „kučeravých“ zátvoriek píšeme pravidlá. Selektor spolu s pravidlami v zátvorkách vytvára definíciu štýlu. Definície sa už nepíšu do atribútov značiek, ale do špeciálnej značky **style**:

```
<html>  
<head>  
    <title>Mojá prvá stránka</title>
```



```
<style>
    h1 {
        color: red
    }

    p {
        color: navy
    }
</style>
</head>
<body>
...
```

Všimni si, že značka **style** sa nachádza v hlavičke dokumentu!

Hm, lenže to stále ešte miešame jabĺčka s hruškami... takto zapísané definície nám HTML súbor značne zneprehľadnia! Sú rozťahané cez tri riadky, a keď ich bude v dokumente viac (a to určite bude!), budú nám tam prekážať.

Okrem toho, predstavte si, že by váš web pozostával z viacerých (napr. 5 alebo aj 555) stránok. Keby boli definície štýlov v každej z nich, bolo by veľmi ťažké napr. zameniť farbu všetkých nadpisov **h1** vo všetkých 555 stránkach za nejakú inú, pretože by sme to museli prepisovať v každom súbore... a nedá sa to robiť nejakو centrálnе? Máte pravdu – dá.

To isté napadlo aj tvorcom CSS a to je vlastne dôvod, prečo vlastne CSS vzniklo: aby sa jednou zmenou štýlu mohlo zmeniť vzhľad celého webu, nech je tam tých stránok aj 5 555 555.

Riešením je, že definície štýlov vložíme do samostatného súboru, napríklad *styl.css*:

```
h1 {
    color: red
}

p {
    color: navy
}
```

V hlavičke HTML dokumentu nám namiesto tých rozťahaných definícií zostane iba jedna nová veľmi sympatická značka **link**:

```
...
<head>
```

```
<title>Moja prvá stránka</title>
<link rel="stylesheet" type="text/css" href="styl.css" />
</head>
...
```

Značka `link` je prázdna, preto sa píše s lomkou na konci!

Po značke `title` je to ďalšia dôležitá súčiastka, ktorá patrí do *hlavičky dokumentu*. Je však nepovinná, čiže keď nechceme štýly, nemusíme ju písať.

Atribút `rel` hovorí, že pripájaný súbor obsahuje definície štýlu (stylesheet). Atribút `type` udáva typ štýlu, v našom prípade je to textový súbor s CSS pravidlami (okrem CSS existujú napríklad aj sofistikovanejšie XSL). A napokon, atribút `href`, ktorý už poznáme z odkazov, obsahuje cestu k súboru so štýlmi. Odteraz prehliadač vie, že sa má pri vykresľovaní (okrem vlastných pravidiel) riadiť aj našimi definíciami.

Podme si nastaviť farbu pozadia. Do súboru *styl.css* pridajme tieto riadky:

```
body {
    background: navy;
    color: yellow
}

h1 {
    ...
```

Všimnime si: jedna definícia môže obsahovať *viacero pravidiel*, oddeľujú sa od seba *bodkočiarkou*.

Touto definíciou sme nastavili farbu pozadia na žltú a farbu textu na modrú. Rovnako by sme mohli vymyslieť aj iné. Prehliadače poznajú dokopy asi 16 rôznych preddefinovaných slovných pomenovaní – **farebných konštánt** (niektoré prehliadače aj viac, ale navzájom si s inými nerozumejú, takže nepoužívať!). Sú to farby: `###` doplniť názvy farieb.

Naše farby sú veľmi sýte, zle sa na ne pozerá. Skúsme ich teda trochu „skrotiť“:

```
body {
    background: #99f;
    color: #ff9;
}
```

`#99f` a `#ff9` sú tiež **farebné konštanty**. Každý počítačový monitor či televízor mieša farby z troch základných farieb (rgb – red, green, blue – červená, zelená, modrá). Písmenká a číslice za mriežkou udávajú intenzitu každej základnej farby. Znak mriežky hovorí, že číslice sú zapísané

v šesnástkovej (hexadecimálnej) sústave.

My bežní smrteľníci si pod pojmom šesnástková sústava predstavíme veľké kakavkovonebíckové nič z vesmíru. Ale pre počítač je veľmi vhodná a lepšie sa mu v nej pracuje ako v desiatkovej. Na rozdiel od nej sa v šesnástkovej sústave používa... 16 číslic. Keďže my máme číslic len 10, ako ďalšie číslice berieme prvé písmená abecedy A–F.

Farebnú konštantu **#99f** vieme zapísať ešte zložitejšie: **#9999ff**. Načo? Tento zápis je síce dlhší, ale dá sa v ňom zapísať viac farieb, ktoré by sme v skrátenej forme zapísať nemohli. Napríklad konštanta **#fe0ba5** predstavuje veľmi zaujímavý odtieň krikľavoružovej ;)

## Rodič a jeho (poslušný) potomkovia

Elementy majú spoločnú veľkú väčšinu vlastností, ktoré možno pomocou CSS zmeniť. Preto môžeme meniť nielen farbu nadpisov a odstavcov, ale napríklad aj elementu **body**.

Keďže element **body** predstavuje celú viditeľnú časť HTML dokumentu, zmena jeho farby spôsobí vyfarbenie celej stránky. Všetky elementy, ktoré sú vnorené do **body**, automaticky získavajú (zdedia) nielen jeho farby, ale aj ostatné vlastnosti. Preto sa takéto elementy aj nazývajú **rodič** a **potomkovia**.

Prečo ale u nás sú niektoré texty červené alebo modré? Dôvodom je to, že vlastnosti, ktoré potomkovia zdedili od rodiča **body**, sme u elementov **h1** a **p** už predefinovali ďalšími štýlmi, a preto sa vykresľujú podľa vlastných pravidiel.

Dedenie vlastností je súčasťou zložitého procesu, ktorý sa volá **kaskáda**. Na tomto princípe fungujú celé kaskádové štýly (CSS). Viac o kaskáde niekedy nabudúce v iných častiach tejto knižky.

V ďalšej lekcii konečne príde na rad písmo.

## 5. lekcia: Fontotéka

Nejeden zvedavec asi hneď na začiatku skočí na túto lekciiu. Predsa mu však odporúčam najprv si preštudovať tú predchádzajúcu (####Šminky a farby) – sú tam totiž dôležité veci o tom, ako vyzerá kaskádový štýl a ako ho vôbec dostať do stránky. Bez týchto znalostí je pochopenie nasledujúcich lekcii oveľa náročnejšie.

Od tejto lekcie ďalej už predpokladám, že rozumiete pojmom *definícia štýlu*, *selektor*, *pravidlo*, *vlastnosť* a *hodnota* a že poznáte značky **style** a **link** a ich atribúty.

### Škaredé kačiatko sa mení na labuť

Základné písmo, ktoré používajú prehliadače, je pre mnohých z nás veľmi nudné a škaredé. Bodlo by, keby sme ho vedeli zmeniť. Preto sa dnes zameriame čisto na písmo a vlastnosti s ním spojené.

### Základná veľkosť písma

Takto nastavíme základnú veľkosť písma:

```
body {  
    font-size: 12pt;  
}
```

Stupeň (veľkosť) písma sa uvádza bodoch (**pt**) alebo pajkách (**p**). Takto zadané písmo sa zobrazuje rovnako veľké na obrazovkách s rôznym rozlíšením. Samozrejme, nik vám nebráni používať aj iné jednotky (napr. **px** – obrazovkové body alebo aj **mm** či **cm**) ale tu veľkosť písma závisí od rozlíšenia, prípadne správneho nastavenia dpi. Ďalšie jednotky sú **em** – veľkosť základného písma alebo % - percentá:

```
h1 {  
    font-size: 200%;  
    font-weight: bold;  
}
```

Pozor, názov jednotky od hodnoty neoddeľujeme medzerou!

Odteraz sa všetky nadpisy prvej úrovne budú vypisovať dvojnásobne veľkým písmom ako základný text. V pohode by sme namiesto **font-size: 200%** mohli napísať aj **font-size: 2em**, čo v tomto prípade znamená to isté (dvjnásobok základnej veľkosti písma).

Vlastnosť **font-weight** určuje tučnosť písma. Možné hodnoty sú: #### **normal**, **semi-bold**, **bold**, **bolder**.

A čo takto kurzíva? OK, tu ju máme:

```
h2 {  
    font-size: 150%;  
    font-weight: bold;  
    font-style: italic;  
}
```

No a je to. Všimnite si, že kurzívu dosiahneme zmenou vlastnosti **font-style**. Možné hodnoty sú **normal** a **italic**, **small-caps**. Keďže sme nastavili aj vlastnosť **font-weight**, výsledkom bude tučná kurzíva.

Dodefinujme ešte ostatné veľkosti titulkov:

```
h3 {  
    font-size: 120%;  
    font-weight: bold;  
}
```

```
h4, h5, h6 {  
    font-size: 100%;  
    font-weight: bold;  
}
```

Novinka v tomto prípade spočíva v zoskupení selektorov **h4**, **h5**, **h6**: týmto spôsobom nastavíme vlastnosti pre všetky elementy **h4**, **h5** a **h6** naraz.

## No a teraz už tie fóóónty, plz!

Dobre teda:

```
body {  
    font-size: 12pt;  
    font-family: "Trebuchet MS";  
}
```

No a to je všetko? Nie je, ešte nie sme na konci! Nezabúdajme, že našu stránku sa budú prezerať na

rôznych počítačoch s rôznymi operačnými systémami. Písma, ktoré máme nainštalované my, nemusia mať nainštalované ostatní. Napríklad písmo *Trebuchet* je bežné v novších verziách Windows, ale užívatelia Linuxu, Mac či starších Windows ho v systéme bežne nemávajú. Preto sa odporúča uviesť aj alternatívy, ktoré sa použijú, keď dané písmo nie je poruke:

```
font-family: "Trebuchet MS", Arial, Helvetica, sans-serif;
```

Všimnite si, že *jednoslovné* názvy písiem nemusíme dávať do úvodzoviek!

V tomto prípade sa prehliadač najprv pozrie, či má nainštalované písmo *Trebuchet MS*. Ak ho nenájde, pohládá písmo *Arial* (bežné na starších Windows), potom *Helvetica* (bežné v Mac OS). Až napokon, keď nenájde žiadne z uvedených písiem, použije písmo, ktoré má prehliadač nastavené ako štandardné bezpätkové (**sans-serif**).

## Odstavce neodstavte !

S odstavcami je veľmi ľahká robota. Všetko, čo potrebujeme, je nastavenie zarovnania a odsadenia prvého riadku:

```
p {  
    text-indent: 2em;  
    text-align: center;  
}
```

Vlastnosť **text-indent** udáva odsadenie prvého riadku. Vo webových stránkach sa nepoužíva veľmi často, odstavce bývajú zväčša oddelené jedným voľným riadkom. Omnoho častejšie sa používa vlastnosť **text-align**, teda zarovnanie textu. Možné hodnoty sú **left**, **center**, **right**, **justify**.

## Ďalšie vlastnosti písma a textu

Na písmo majú vplyv aj ďalšie vlastnosti, uveďme ich v krátkosti.

```
line-height  
text-decoration
```

## ***Linky šminky***

Ako šminkovať linky

stavy

## 6. lekcia: Okraje, odstup a rámiky

### Blokové a riadkové elementy

HTML elementy podľa správania delíme na dva typy: **riadkové** a **blokové**. Rozdiel medzi nimi je v tom, ako sa správajú k ostatným elementom na stránke. Niektoré elementy sa zobrazujú ako úseky textu (napr. **em**, **strong** alebo **a**.) Iné elementy sa zobrazia ako odstavce textu, oddelené od zvyšku stránky (**h1**, **u1**, **body** alebo **p**).

Už poznáme vlastnosť **background**, ktorá nastavuje farbu pozadia. Táto patrí medzi **blokové vlastnosti**. No a tieto vlastnosti, ktorým sa dnes ideme pozrieť na zúbky, sa prejavujú trochu inak na blokových a trochu inak na riadkových prvkoch.

### Rámik

Vezmime si nejaký blokový prvok:

```
p {  
    border-width: 1px;  
    border-style: solid;  
    border-color: red;  
}
```

Keď si pozrieme dokument, výsledkom bude červený rámik (**border-width: 1px**) neprerušovanou čiarou (**border-style: solid**) hrúbky 1 pixel (**border-color: red**) okolo každého odstavca.

Všimnite si, že rámiky okolo odstavcov zaberajú celú šírku okna!

### Odstup

Ak už nie sýte kriľavé farby, určite nám na tomto bloku prekáža, že písmo je natlačené až na rámiku. Dajme mu preto trochu vzduchu:

```
p {  
    border-width: 1px;  
    border-style: solid;  
    border-color: red;  
    padding: 0.5em;  
}
```



## Okraj

Ách, hneď sa nám tu lepšie dýcha! Nenulová hodnota padding (odstup) totiž spôsobuje medzeru medzi vnútrom bloku a rámikom. No a keď už sme vo švungu, skúsme to ešte trochu vylepšiť:

```
p {  
    margin: 3em;  
    border-width: 1px;  
    border-style: solid;  
    border-color: red;  
    padding: 0.5em;  
}
```

## Šírka a výška

Pozrime sa, ako sa nám všetky odstavce od zvyšku stránky vydelili! Teraz je toho vzduch azda aj priveľa. Ak sa nám zdajú bloky príliš široké, môžeme si nastaviť rozmery:

```
p {  
    width: 33%;  
    height: 3em;  
    margin: 3em;  
    border-width: 1px;  
    border-style: solid;  
    border-color: red;  
    padding: 0.5em;  
}
```

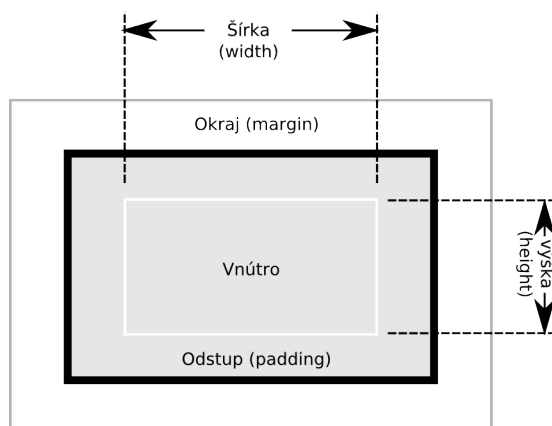
A tak máme blok široký na tretinu okna a vysoký na tri riadky textu. Hrúbka odstupu, rámika a okraja sa do tejto šírky *nezapočítavajú*, preto ak chceme poznať naozajstné rozmery elementu, celkovú treba k rozmerom pripočítať pripočítať dvojnásobky týchto troch. Teda

$$\text{Šírka} = 33\% + 1\text{em} + 2\text{px} + 6\text{em};$$

$$\text{Výška} = 3\text{em} + 1\text{em} + 2\text{px} + 6\text{em};$$

## Zhrnutie

Aby sme z toho nemali guláš, zhrňme si to na obrázku:



Posledná vec, ktorú dnes ešte vyskúšame, je naformátovať týmto štýlom ľubovoľný riadkový element, povedzme napr. **em**:

```
p, em {  
    ...  
}
```

Použili sme chmat z minulej lekcie: zoskupovanie selektorov (Ved' načo písať viac, keď sa to dá vyriešiť tromi znakmi?). Keď si stránku obnovíme (najprv treba skontrolovať, či sa v dokumente nachádza aspoň jeden element **em**), uvidíme, aké vlastnosti sa zmenili a aké nie.

Zmenilo sa toho viac či menej oproti blokovému elementu **p**? Ktoré vlastnosti sa zmenili a ktoré nie?

## 7. lekcia: Ako sa Div divil a Span pospával (v tej istej triede!)

Naučili sme sa toho už pomerne veľa. Vieme správne zapísať stránku s jednoduchou štruktúrou. Vieme stránku vyfarbiť a niektoré prvky aj vložiť do vlastnej orámovanej farebnej krabičky.

Ale ako dosiahneme, aby bola v jednom rámičku viac elementov naraz? Povedzme napríklad, že chceme zarámovat' celé *Mor ho* aj s titulkom. To zatiaľ nevieme.

Čo keby sme celú báseň (nadpis **h3** a odstavce **p**) vnorili do ďalšieho odstavca?

`<p>`

`<h3>Samo Chalupka: Mor ho!</h3>`

`<p>Zleteli orly z <em>Tatry</em>,<br />`

`tiahnu na <em>podolia</em>,<br />`

`ponad vysoké hory,<br />`

`ponad šíre polia.</p>`

`</p>`

A teraz odstavce orámujeme:

`p {`

`border-width: 3px;`

`border-style: dotted;`

`border-color: red;`

`}`

Hm, lenže sa nám naformátovali úplne všetky odstavce v stránke! To nie je presne to, čo sme chceli. Našťastie, existuje jednoduché riešenie, ako z toho von. Volá sa **triedy**. Mierne upravíme selektor pre odstavce:

`p.basen {`

`border-width: 3px;`

`...`

`}`

Selektor **p.basen** znamená, že každý odstavce, ktorý patrí do triedy **basen**, bude mať uvedené vlastnosti. V HTML ale musíme tento odstavce označiť, a to pomocou atribútu **class** (trieda).

<p class="basen">

<h3>Samo Chalupka: Mor ho!</h3>

...

Je to ako v obyčajnej triede: všetci žiaci (aj učiteľ :-), ktorí navštevujú triedu, učia sa to isté a musia dodržiavať tie isté triedne pravidlá (s väčšími či menšími odchýlkami ;-).

## Lenže...

Má to ešte jeden háčik. Pokračujme v prirovnaní k triede a každú slohu básne (značka **p**) prirovnajme k žiakovi. Nadpis prirovnajme k učiteľovi. Značka odstavca, ktorá to všetko „obaľuje“ bude školská trieda. A teraz by sme o tej školskej triede, so žiakmi aj s učiteľom povedali, že je to „človek“. Pekný nezmysel, však?

No ale podobne sme sa zachovali pred chvíľou HTML. Zneužili sme značku **p**, ktorá má význam *odstavec* na to, na čo sa používať nemá. *Odstavcom sme totiž nazvali celú báseň*, čo nie je pravda. Tým sme porušili **sémantiku** (###význam) značkovania. „Hm, veď to nevádi,“ povieť si. „Hlavne, že sa to vykreslí ako má.“ Máte pravdu: Keď to číta človek, veľmi to neprekáža. Ale keď tú stránku prečíta *autmat* (napr. Google), nedokáže sa v zmätenej štruktúre zorientovať, a tým pádom nám nepomôže. Načo je taký automat, ktorý napríklad, keď mu prikážeme vypísať prvý odstavec básne, vypíše ju celú? A tomu sme sa mohli dodržaním sémantiky vyhnúť.

## Bezvýznamná HTML značka?

Ak si predchádzajúci text preskočil, vedz, že sme zistili, že automat sa pri čítaní stránky môže veľmi *diviť* (dokonca až *zdivieť*), ak sme pri písaní zneužívali HTML značky v protiklade s tým, čo značia (znamenajú).

Ako sme videli v príkladoch, k takejto *HTMeLe* dochádza vtedy, keď sa snažíme HTML stránku pekne prezentovať, a pritom používame značky, ktoré neslúžia na prezentáciu.

Ako ale zariadiť, aby sme automat nedoplietli, ale zároveň aby naša stránka bola pekná? Vyzerá to na peknú bitku medzi rozumom a krásou. Chcelo by to nejaký element, ktorým by sme nemohli nič pokaziť, keď ho dopíšeme do HTML. A takýto element existuje, ba dokonca sú až dva.

## Div a Span zasahujú

Prosím potlesk, na scénu prichádzajú elementy **div** a **span**! Volajú sa **bezvýznamové** (pozor, nie bezvýznamné!) alebo aj **nešpecifické**, pretože nie su nositeľmi významu, a teda nemenia štruktúru. Akoby ste pekný obrázok oblepili priesvitnou páskou. Automaty ich obchádzajú akoby ich nevideli. Pomôžu nám však pri písaní štýlov. Prepíšeme teda náš príklad pomocou **div**:

<div class="basen">

<h3>Samo Chalupka: Mor ho!</h3>

...

</div>

A ešte mierne pozmeníme CSS:

```
div.basen {
```

```
....
```

Výsledok je ten istý, ale tentoraz sme na to použili správnu značku. **Div** nemení význam dokumentu, a to je dobre, pretože rámkovanie básne súvisí s prezentáciou, nie s jej štruktúrou. Namiesto rámika by sme rovnako dobre mohli nechať báseň poskakovať a robiť kotrmelce, alebo svietiť dúhovými hviezdikami (pravda, keby sme to vedeli...:-), jej štruktúru by to nezmenilo.

## Dobre, ale načo sú nám dve?

Z minulej lekcie vieme, že element môže fungovať ako blokový a riadkový. Blokový sa od ostatného textu vydeľuje, riadkový ostáva na riadku (hovoríme, že **neprerušuje tok textu**). No a vlastne to je rozdiel medzi **div** a **span**. Div sa používa všade tam, kde formátujeme blok, a span sa používa v rámci riadku. Aké jednoduché!

## 8. Nová META dobre metie

Keď už sme našu stránku napísali, aj sme ju vyfarbili, ostáva nám v krátkosti opísať, aké informácie a v akej forme stránka obsahuje.

### Meta-informácie

Najdôležitejším účelom takýchto informácií (volajme ich **meta-informácie**) je, aby nám prehliadače a vyhľadávače čo najrýchlejšie dodali správne výsledky. A to veru nie je jednoduché! Na internete sa povalujú stovky miliárd stránok. Veľa z nich nie je správne označených, a tak sa v nich musí vyhľadávač dlho prehŕňať, kým pre nás nájde to, čo potrebujeme. Nám sa to síce zdá ako chvíľa, ale ktovie, čo všetko by pre nás našiel, keby mal viac času.

Meta-informácie, ktoré sú vlastne heslovito zapísané najpodstatnejšie informácie o obsahu dokumentu, to strojom pri hľadaní výrazne uľahčujú. Preto je užitočné ich do stránky napísať.

Preto si my niekoľko takýchto meta-informácií ukážeme.

### Prvá je diakritika

Určite sa ti už stalo, že si napísal stránku, ale po zobrazení v prehliadači si namiesto mäkčeňov a dlžňov (diakritiky) videl iba rôzne skupiny neznakov. Je to preto, lebo prehliadač pri zobrazení použil nesprávnu tabuľku znakov.

Keďže počítač si pamätá iba čísla, programátori vynasli riešenie, aby bolo možné zapamätať text pomocou postupnosti čísel. Každý znak dostal svoje číslo a vznikla tabuľka znakov, alebo kódovanie znakov.

V ďalekej minulosti boli počítače niekoľkokusové originály, a každý originál používal vlastnú tabuľku znakov. Až sa prišlo na to, že keď majú rôzne počítače medzi sebou komunikovať, musia si najprv porozumieť, a tak vznikla ASCII tabuľka, ktorá sa tak rozšírila, že ju po istom čase používali všetci. Problém však bol, že tento kód obsahuje len tie znaky, ktoré sa používajú v anglicky hovoriacich krajinách a v západnej Európe.

Na mäkčene a dlžne by sme mohli zabudnúť, keby osvietené hlavy nerozšírili ASCII tabuľku o ďalšie znaky. Ale takýchto ľudí bolo na svete viacero. Každý si ASCII rošíril po svojom, a tak vzniklo niekoľko rozšírení, každé z nich má znaky poprehadzované nejako inak.

### Du jú spík uft-8?

Keď počítač s jednou tabuľkou (napr. UTF-8) posiela čosi druhému počítaču, v záujme používateľa je slušné oznámiť, s akou tabuľkou má pracovať. V HTML to urobíme takto:

```
<html>
```

```
<head>
```

```
<meta http-equiv="content-type" content="text/html; charset=UTF-8" />
```

```
...
```

</head>

Pomocou značky **meta** vkladáme väčšinu meta-informácií. Atribút **content** dáva jasne najavo, že súbor je v HTML a že na kódovanie znakov sa používa tabuľka **utf-8**, a tak sa prehliadač môže podľa toho zariadiť a znaky správne zobrazíť. Atribút **http-equiv** hovorí, čím sa meta-informácia zaoberá, v tomto prípade je to typ obsahu (content-type).

Všimnite si, že **meta**, sa vždy píše do hlavičky dokumentu (na stránke sa nezobrazuje), a že je to prázdna značka, takže má na konci lomku! Ďalej si všimnite, že atribútom je **content**, ale nie charset, pretože sa nachádza vo vnútri úvodzoviek! Ak to napíšete zle, prehliadač to nepochopí!

## Ďalšie užitočné info pre FBI

Meta-informácie môžu veľmi ovplyvniť postavenie vašej stránky v indexoch vyhľadávačov. Čím presnejšie informácie zadáte, tým väčšiu šancu máte, že čitatelia nájdu práve tú vašu stránku. Správny výber a písanie týchto informácií je pomerne zložitá vec a nazýva sa SEO – *Search Engine Optimization* (optimalizácia pre vyhľadávače).

Takže čo v hlavičke môžeme ešte nájsť?

```
<meta name="author" content="Jožko Mrkvička, JojoM, 2008, Bratislava" />
```

Obsahuje meno autora stránky, prípadne rok a miesto.

```
<meta name="copyright" content="Jožko Mrkvička, (c) 2008" />
```

Vlastník autorských práv a druh licencie, rok.

```
<meta name="description" content="Popis stránky" />
```

Krátky popis stránky, ktorý ju čo najlepšie vystihuje. Odporúča sa najviac 250 znakov.

```
<meta name="keywords" content="Chalupka, Samo, Mor ho!" />
```

Kľúčové slová, ktoré sa viažu k obsahu stránky. Zadávame ich oddelené čiarkami. Odporúča sa zadávať len slová, ktoré sú v dokumente zapísané, v tom istom tvare vrátane diakritiky.

###Popisy pri obrázkoch

## Táto stránka je už neaktuálna. O 10 sekúnd budete presmerovaní

No a na záver jedna pikoška, ktorú môžete použiť, pokiaľ chcete návštevníka automaticky presmerovať na inú stránku (napr. nová verzia vašej stránky):

```
<meta http-equiv="refresh" content="10; url=www.smnd.sk" />
```

Číslo udáva počet sekúnd, po ktorých sa má nahrať nová stránka, url udáva adresu stránky. Znova si všimnite, že **url** nie je atribútom značky **meta**!



## Tabuľky a definičné zoznamy

Ostali nám dve veci, ktoré by sme si mali dovysvetľovať. V predchádzajúcich lekciách sme ich pre jednoduchosť obišli a odložili na časy, keď už budeme nacvičenejší.

Pamätáte sa na tabuľku? Dnes sa naučíme, ako ju jednoducho naštýlovať, a ako vyrobiť výraznejšie hlavičky stĺpcov. Napokon si ukážeme ešte jeden zvláštny typ zoznamov, takzvaný **definičný** (slovníkový) **zoznam**.

###tabuľky

###štýlovanie tabuľky

###definičné zoznamy

V podstate sme sa už naučili všetko nevyhnutne dôležité, čo budeme pri písaní a štýlovaní stránok potrebovať. Hm, všetko? Už počujem tých skúsenejších frflať, že sme nebrali formuláre, rámce a podobné veci. HTML toho samozrejme obsahuje ešte trochu viac, to však nie je nevyhnutne potrebné na písanie jednoduchých stránok a dozvedieť sa to môžeš z iných lekcí (napr. o formulároch veľa hovoria Aninove lekcie MySQL a PHP, o rámcoch, ktoré sa používajú dnes už len pri niekoľkých špeciálnych prípadoch, si môžeš nájsť niečo na webe, napr. [www.jakpsatweb.cz](http://www.jakpsatweb.cz)).

V ďalších kapitolách, s názvom *Stránka od piky* si ukážeme, ako prebieha taká tvorba webu od začiatku až do jeho zverejnenia.

## **Stránka od piky: Úvod**

V ďalších kapitolách si postupne prejdeme cestu od návrhu webu a k jeho vytvoreniu a zverejneniu a optimalizáciu pre vyhľadávače.

Pre správne pochopenie predpokladám, že už máte základy HTML a CSS (preberané v predchádzajúcich kapitolách) šťastne za sebou. V nasledujúcej časti už nebude priestor a ani čas na opakovanie a vysvetľovanie základov.

## Stránka od piky: Mapa

### Priemysel štruktúry

Aby sme mohli začať s grafikou, najskôr si treba ujasniť, čo všetko má stránka obsahovať a ako to bude organizované. Postačí malá (prípadne viacúrovňová) osnova zapísaná do odrážok.

**Príklad.** *Osnova budúcej stránky hasičského zboru.*

Predstavme si, že si vedúci hasičského zboru objednal u nás peknú stránku. Nič o tvorbe stránok netuší, tak len povedal, že by mala byť pekná a že by na nej malo byť to, to a to. K tomu nám dal jeden pokrčený papier formátu A4, na ktorom sú v jednom dlhom odstavci zapísané informácie, ktoré by na stránke chcel mať.

{citácia: papier A4 s informáciami}

Na papieri je tých informácií veľa naraz. Väčšinou návštevník stránok nemá veľa času ani chuti na čítanie. Chce nájsť nejakú konkrétnu informáciu, a to čo najrýchlejšie. Našou webdizajnérskou úlohou je teda čo najviac mu toto hľadanie uľahčiť. Je veľký rozdiel, keď mu predhodíme podsatzné informácie napríklad v logicky rozdelenej stromovej štruktúre alebo ho zavalíme niekoľkými tonami obkecov nabitých do jednej strany bez ladu a skladu. Treba preto informácie rozčleniť a zbaviť balastu.

**Zásada:** *Informácie treba dávať stručne a v malých dávkach. Najdôležitejšie informácie idú najskôr.*

Vráťme sa k našim hasičom. Po preštudovaní papiera vieme jeho obsah rozdeliť do týchto oblastí:

#### **história zboru, činnosť, ľudia, autá, hasičská technika, galéria fotografií**

Výskumami sa zistilo, že dočasná pamäť človeka je obmedzená. Netrénovaný človek si v dočasnej pamäti nedokáže udržať viac ako 7 *navzájom nesúvisiacich* vecí – telefónnych čísel, mien, alebo trebárs domov. Z toho dôvodu by malo byť položiek v každom menu čo najmenej.

**Zásada:** *Položiek ľubovoľného menu by malo byť čo najmenej (maximálne 7).*

Keď toto zistenie aplikujeme na náš príklad, zistíme, že *Autá* a *Hasičská technika* by sa dali zlúčiť do jedného pojmu: *Technika*:

#### **história zboru, činnosť, ľudia, hasičská technika, galéria fotografií**

Niektoré pomenovania sa dajú ešte zjednodušiť. Napríklad položka *História zboru* sa dá zostručiť na *História*, lebo čitateľovi by malo byť jasné, že sme na stránke hasičského zboru, a že teda ide o jeho históriu. Po zostručnení dostaneme konečne podobu hlavného menu.

#### **história, činnosť, ľudia, technika, galéria**

Každá z položiek menu môže mať pod sebou ďalšie položky – pokiaľ potrebujeme informáciu ďalej členiť:

- **história**
- **činnosť**
  - zásahy

- výcvik
- reprezentácia
- **ľudia**
  - výbor
  - členovia
- **technika**
  - autá
  - čerpadlá
  - hadice
  - iná
- **galéria**

Dostali sme sa k 2-úrovňovej hierarchickej štruktúre, ktorá prehľadne zhŕňa informácie na našej budúcej stránke. Nazýva sa to aj *mapa stránok* alebo *informačná štruktúra*. Z tejto štruktúry bude vychádzať grafická podoba nášho webu. Ale o tom až v budúcej lekcii.

## **Stránka od piky: Grafický návrh**

Keď máme informačný návrh hotový, zdefinujeme si, ako bude stránka vyzerat' po grafickej stránke. Ako vidíte, ešte sme nenapísali ani riadok html, ale aj na to príde.

Keď si najprv navrhujeme grafiku, má to tieto výhody:

- stránka bude vyzerat' graficky jednoliata;
- ušetríme si nervy pri zmenách grafiky, lebo grafika bude v jednom súbore.

## **Skica**

Rukou si nakreslíme malú skicu budúcej stránky. Prečo rukou? Je to rýchlejšie a najmä – rukou vieme robiť lepšie ako myšou.

Pri kreslení myslíme polohu hlavičky, hlavného a vedľajšieho menu, kde sa bude vypisovať text a ďalšie informácie a pätička s doplnkovými informáciami.

###ručná skica stránky

Pri návrhu dbáme na zásadu č. 1, aby najdôležitejšie informácie boli čitateľné a najmä na to, aby sme informácie a ich dôležitosť vedeli rozoznať na prvý pohľad.

###popisy k obrázkom Využívame pritom rôzne farby a veľosti a písma (24, 18, 12, 9-bodové), štýl písma (normálne, tučné, kurzíva, kapitálky). Nezabudnime aj na voľné miesto, kde si oko čitateľa môže oddýchnuť.

Samotnú grafiku vytvoríme v programe InkScape, neskôr ju spracujeme programom GIMP.



# DHZ Horná Dolná

**História Činnosť Ľudia Technika Galéria**

**Autá**  
**Čerpadlá**  
**Iná**

## Čerpadlá sú nevyčerateľnou súčasťou

Čerpadlá sú nevyčerateľnou súčasťou výzbroje každého dobrého hasičského zboru v Slovenskej republike. Čerpadlá sú nevyčerateľnou súčasťou výzbroje každého dobrého hasičského zboru v Slovenskej republike. Čerpadlá sú nevyčerateľnou súčasťou výzbroje každého dobrého hasičského zboru v Slovenskej republike. Čerpadlá sú nevyčerateľnou súčasťou

### Výzbroje každého

dobrého hasičského zboru v Slovenskej republike. Čerpadlá sú nevyčerateľnou súčasťou výzbroje každého dobrého hasičského zboru v Slovenskej republike. Čerpadlá sú nevyčerateľnou súčasťou výzbroje každého dobrého hasičského zboru v Slovenskej republike. Čerpadlá sú nevyčerateľnou súčasťou výzbroje každého dobrého hasičského

ho zboru v Slovenskej republike. Čerpadlá sú nevyčerateľnou súčasťou výzbroje každého dobrého hasičského zboru v Slovenskej republike. Čerpadlá sú nevyčerateľnou súčasťou výzbroje každého dobrého hasičského zboru v Slovenskej republike. Čerpadlá sú nevyčerateľnou

**Stránka od piky: Stílpce**

**Stránka od piky: Navigácia**



***Stránka od piky: Upload***

**Stránka od piky: Animácia**

**Stránka od piky: Validácia**

## ***Stránka od piky: Flash a multimédiá***

## **Stránka od piky: JavaScript**

## Stránka od piky: Optimalizácia (SEO a iné)

Farba písma

**Príklad:** Chceli by sme vyfarbiť nadpis Atlantída na modro. Tento nadpis je vyznačený značkou `h1`. Pridáme tejto značke nový **atribút** `style`:

```
<h1 style="color: blue">Atlantída</h1>
```

Atlantída

Všimni si, ako vyzerá zápis atribútu `style`: najprv sa píše **názov vlastnosti**, ktorú treba zmeniť. V našom prípade je to `color`. Potom je **dvojbodka** (`:`) a za ňou nasleduje **hodnota** (`blue`).

**Názvy farieb** sú v angličtine, často sa používajú napr. *green, red, blue, yellow, magenta, cyan, black, white, pink* a podobne. Pri niektorých farbách možno použiť aj predponu *light* a *dark*, napr. *lightblue, darkgreen*. Kompletný zoznam farieb nájdeš na internete ( $\Rightarrow$  html farby).

**Pre machrov:** Ak slovne pomenované farby nestačia, môžeš si namiešať vlastnú. Takáto farba sa zapisuje v tvare `#rrggbb`, kde `rr`, `gg` a `bb` sú dvojčiferné čísla v šestnástkovej sústave (ako číslce sa používajú 0—9, a—f), a určujú intenzitu farebných zložiek pre červenú (`r`), zelenú (`g`) a modrú (`b`) farbu. Ak sú číslce po dvojiciach rovnaké (napr. `#5500ff`), tak môžeme takúto farbu zapísať v skrátenej tvare (v našom prípade `#50f`).

**Úloha 1:** Vyfarbi každý nadpis vo výsledkovej listine z predchádzajúcej lekcie inou farbou.

## Farba pozadia

Pamätáte sa na značku `<body></body>`, ktorá v sebe uzatvára všetko, čo sa na stránke zobrazuje? Pridáme jej **atribút** `style`. Nech má napríklad bledomodré pozadie:

```
<body style="background-color: lightblue">
```

**Atribút `style`.** Tento atribút sa dá vložiť takmer do každej html značky (do tých, ktoré sa zobrazujú). Pomocou neho určíme štýl, akým sa značka zobrazí.

**Farba pozadia.** Musí byť kontrastná s farbou písma, aby boli obrysy znakov dobre čitateľné. Čím sú farby *podobnejšie* (podobný farebný tón, jas...), tým *menší* kontrast medzi sebou majú. Takýto text sa síce môže dať čítať, ale je to náročnejšie a odnesú to naše oči -- rýchlejšie sa unavia.

**Viacslovné pomenovania vlastností.** Všimni si, že vlastnosť `background-color` sa píše so spojovníkom! Píšeme to pri viacslovných pomenovaniach vlastností.

**Úloha 2:** V stránke o Atlantíde nastav farbu pozadia a odstavcom daj farbu písma, ktorá dobre kontrastuje s pozadím.

## Zarovnanie

Keď prehliachu nepovieme ako má text zarovnávať, zarovná ho najjednoduchším spôsobom – vľavo. Každý riadok bude začína na ľavom kraji okna a končí slovom, ktoré sa ešte zmestí celé.

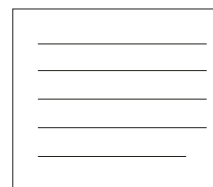
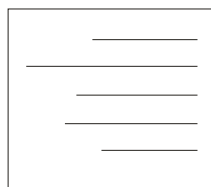
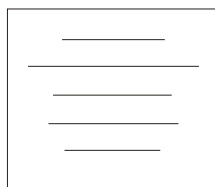
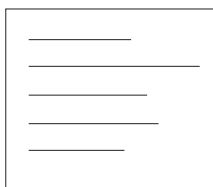
Nasledujúce slovo sa *zalomí* už na ďalší riadok. Na rozdiel od textového procesora prehliadač slová nerozdeľuje (na to by potreboval slovník).

**Úloha 3:** Vyskúšaj na texte o Atlantíde, ako prehliadač láme riadky, keď sa myšou zmení šírka okna.

Ak chceme zarovnať na pravý okraj, na stred alebo nebodaj do bloku (*justify*), urobíme to znova pomocou štýlu. Vezmime si napríklad odstavce zarovnaný vpravo:

```
<p style="text-align: right">Po chodníku sa chodí vpravo!</p>
```

Možné zarovnania:



```
text-align: left    text-align: center    text-align: right    text-align: justify
```

**Keď meníš viac vlastností**, treba ich oddeliť bodkočiarkou. Inak by to prehliadač nepochopil. Teda štýl na modrý nadpis zarovnaný vľavo bude vyzeráť:

```
<h1 style="color: blue; text-align: left">Atlantída</h1>
```

**Úloha 4:** V texte o Atlantíde zarovnaj nadpis na pravý okraj a oba odstavce do bloku.

## Zmena vlastností pre celú stránku naraz

Nevrela ti krv v žilách, keď si v predchádzajúcich cvičeniach musel pridávať atribút style do každej značky? Nečudo, mne by bolo tak isto. Po chvíli to je tých značiek už akosi veľa, v texte je riadny zmätok. Tu treba rýchlo čosi urobiť, kým nestratíme orientáciu

Vypisovať farbu a zarovnanie do každej značky je zdĺhavé, preto sa to teraz pokúsime dať všetky informácie o štýle na jedno miesto na stránke. To miesto je v **hlavičke**:

```
<style type="text/css">
  body { background-color: lightgreen }
  p     { color: navy; text-align: justify }
  h1, h2 { color: blue; text-align: right }
</style>
```

**Značka `<style></style>`.** Súhrnne sú v nej zapísané definície štýlov. Všimni si atribút `type`. Hovorí, že sa jedná o kaskádové štýly. Naprv ide názov značky (bez lomených zátvoriek), a potom v zložených zátvorkách nasledujú zápisy štýlov, ako ich už poznáme. Ak pre jednu značku meníme viac vlastností, oddeľujeme ich od seba bodkočiarkou. Podrobnosti niekedy nabadúce.

**h1, h2** znamená, že vlastnosti sa zmenia pre nadpis prvej aj druhej úrovne.

**Úloha 5:** Naformátuj celý dokument o Atlantíde pomocou značky `style` tak, ako to vidíš vyššie. Nezabudni odstrániť atribúty `style` zo značiek.



## 5. Písmo

Písmo je samostatnou kapitolou internetových stránek. Hrá tu důležitou úlohu velikost, čitelnost, farba, typ — font.

Internetové prohlídače vedí zobrazit jen několik druhů písma. Zdaleka neplatí, že ak máš nějaké písmo nainstalované vo svojom počítači, môžeš ním napísať webstránku. Ono to síce technicky ide (prohlídač sa nebude brániť keď použiješ písmo *Bukvice*), na iných počítačoch však môže nastať problém (Písmo *Bukvice* nemusí byť nainštalované, a teda sa nezobrazí :-).

Ako tvorcovia stránok sa nemôžeme spoliehať na to, že určité písmo bude v počítači nainštalované, pretože na každom počítači môžu byť tieto písma iné. Zabudni na to, že čitateľ si kvôli našej stránke bude inštalovať nové písmo (ak to vôbec vie)!

### Zmena písma na celej stránke

Postup je podobný, ako keď sme menili farbu pozadia. Začneme tým, že **do značky style v hlavičke stránky** pridáme túto definíciu štýlu:

```
body {  
    font-family:    Verdana, Arial, "Helvetica CE", sans-serif;  
    font-size:      10pt;  
    font-style:     normal;  
}
```

Toto zmení vlastnosti pre značku body, teda pre celú stránku.

**Typ písma.** Ako font-family sa zvykne uvádzať viacero fontov. Názvy oddeľujeme čiarkami. Tento zápis zabezpečí, že ak nie je nejaké písmo v počítači, použije sa podobné. V našom prípade sa najprv hľadá písmo Verdana. Ak sa nenájde, hľadá sa Arial. Vo Windows býva väčšinou jedno z týchto písma veľmi bežné. Ale na počítačoch Apple je namiesto Arialu písmo Helvetica CE.

**Úvodzovky.** Všimni si, že ak názov písma obsahuje medzeru, píše sa do úvodzoviek!

**Všeobecné meno fontu.** Ak v počítači nenájde ani Helvetica, použijeme písmo, ktoré má prehliadač štandardne nastavené na zobrazovanie bezpätkového písma (*sans-serif*). Toto písmo si každý môže zmeniť v nastavení prehliadača. Takýchto všeobecných písma je niekoľko:

- serif            pätkové (serifové) písmo (napr. Times New Roman)
- sans-serif      bezpätkové písmo (napr. Arial alebo Helvetica, Verdana, Trebuchet)
- cursive          ručné písmo
- fantasy          rozšafné, voľne písané písmo
- monospace      písmo ako na písacom stroji, znaky majú rovnakú šírku

**Veľkosť písma.** Veľkosti zadávame v rôznych jednotkách. Pri písme sú to najčastejšie typografické body – *pt*, alebo percentá. **Všimni si**, že jednotky sa píše hneď za číslom (bez medzery).

**Štýl písma.** Môže byť normal (obyčajné) alebo italic (*kurzíva*).

## Nadpisy

Zmenili sme písmo na celej stránke. Samozrejme, písmo môžeme zmeniť aj jednotlivým značkám. Napríklad písmo nadpisov býva často odlišné od písma ostatného textu:

```
h1, h2, h3 {  
    font-family: "Times New Roman", "Times CE", serif;  
    font-weight: bold;  
    font-variant: small-caps;  
}  
  
h1 { font-size: 200% }  
h2 { font-size: 160% }  
h3 { font-size: 120%; font-style: italic }
```

**Tučné písmo.** Pomocou `font-weight` môžeme nastaviť tučnosť písma. Normal (obyčajné) alebo bold (**tučné**).

**Variant.** Niekedy sa na stránkach môžete stretnúť s textom, napísaným kapitálkami (small-caps). Čo sú kapitálky? No, zjednodušene povedané, sú to písmená veľkej tlačenej abecedy, ktoré používame aj namiesto malých. Malý príklad: JOŽO A KATKA ŠLI NA VÝLET DO TREŇČINA.

**Percentá.** Jednotky percentá sú výhodné v prípade, keď potrebujeme nejaký rozmer znásobiť. Písmo titulu `h1` má dvojnásobnú veľkosť ako ostatné písmo na stránke, teda 20 bodov. To isté platí aj pre `h2` a `h3`, písmo však bude 1,6- a 1,2-násobok základného písma, teda 16 a 12 bodov.

**Spoločné a jednotlivé definície.** Všimni si, že najprv meníme vlastnosti, ktoré majú `h1`, `h2`, `h3` spoločné: Písmo, váhu a variant. Na ďalších riadkoch pridáme veľkosť písma a štýl pre každé písmo zvlášť.

## Nakoniec malé zopakovanie

<code>Font-family:</code>	názov písma alebo viac názvov oddelených čiarkou
<code>font-size:</code>	veľkosť písma (jednotky pt, %, px... )
<code>font-style:</code>	normal / italic
<code>font-weight:</code>	normal / bold
<code>font-variant:</code>	normal / small-caps

**Úloha 1 (2b):** Nastav svojej stránke základné písmo a písmo pre nadpisy. Nastav farby stránky a zarovnanie odstavcov a nadpisov.

**Úloha 2:** Zariad', aby sa všetky odstavce vypisovali kurzívou, ale nadpisy normálnym písmom.

**Úloha 3 (2b):** V texte o Atlantíde zmeň štýl značiek `em` a `strong` tak, aby sa zdôraznenie kreslilo **ŠIKMÝMI KAPITÁLKAMI** a silné zdôraznenie **TUČNÝMI KAPITÁLKAMI S O DVA BODY VÄČŠÍM PÍSMOM**.

## 6. lekcia: Obrázky

Hurá, prišiel čas vložiť do našej stránky obrázky! Ale nie je to také jednoduché, ako by sa mohlo na prvý pohľad zdať. Musíme najprv vedieť niekoľko dôležitých vecí.

Určite vieš, ako do textu vložíme obrázok vo Word. Tento obrázok sa potom prenáša spolu s dokumentom. Do html stránky sa obrázok nedá takto „zabalit“. Obrázok sa nachádza v samostatnom súbore. Keď ho chceme vložiť do stránky, musíme sa naň **odkázat**.

Prečo je to tak? Nuž, možno sa ti to bude zdať čudné, ale niektoré počítače nevedia zobrazovať grafiku. Ďalšie zasa grafiku zobrazovať nepotrebujú (napríklad keď číta nevidiaci človek, prípadne keď ju „číta“ automat). V tomto prípade by bolo zbytočné „sťahovať“ obrázky, ktoré sú väčšinou niekoľkonásobne objemnejšie ako zvyšok stránky.

**Typ obrázku.** Prehliadač vie sám od seba zobraziť iba obrázky s príponou .gif, .jpg, .png. Niektoré vedia zobrazovať aj .bmp, ale to sa neodporúča, lebo ho nemusia vidieť všetky prehliadače. A ešte je tu aj iný dôvod, prečo je .bmp nevhodný: obrázky tohto typu bývajú veľmi veľké a ich prenos po pomalej linke trvá dlho. A každý kilobajt navyše sa napokon ukáže na vašom účte za internet...

### Príprava

Obrázok, ktorý mienime včleniť do stránky, sa nemôže váľať len tak hocikde na disku. Nakopírujeme ho do priečinka s našou html stránkou, alebo do nejakého jeho podpriečinka. Najlepšie by bolo, keby sme si vytvorili na disku priečinok, kde bude naša stránka, a všetky potrebné súbory (stránku aj obrázky) preniesli doň. Vyhneme sa tak neskorším problémom s prepisovaním adries pri prenose stránky na internet. Ako taká štruktúra priečinkov môže vyzeráť, vidíš vpravo.



### Vloženie obrázku

Predpokladajme, že už máme obrázky pripravené v priečinkoch. Stránku, do ktorej chceme obrázok dať, máme v priečinku `moj_web`. Pomocou atribútu `src` (*source*) ukážeme prehliadaču **cestu**:

```

```

Týmto prehliadaču povieme: „Sem vlož obrázok `nahrдельник.png` z toho istého priečinku, v ktorom sa nachádza stránka (voláme ho aj **aktuálny priečinok**).“

```

```

„Z aktuálneho priečinku choď do podpriečinka obrazky/ a tam nájdeš diagram07.gif.“

Ked' aj sama stránka leží v nejakom podpriečinku, obrázok jej ukážeme pomocou dvoch bodiek, čo znamená „vráť sa do nadpriečinku“:

```

```

„Z aktuálneho priečinku choď najprv do nadpriečinku a potom v priečinku fotky nájdeš súbor fotografia.jpg.“

**Všimni si**, že značka `img` je nepárová, teda má pred pravou zátvorkou lomku.

**Všimni si tiež**, že každá z týchto značiek obsahuje atribút `alt="popis"`, v ktorom je napísané, čo vidíme na obrázku. Atribút `alt` (**alternatívny popis obrázku**) je dôležitý najmä pre nevidiacich a pre vyhľadávače (google), ktoré podľa toho indexujú obrázky.

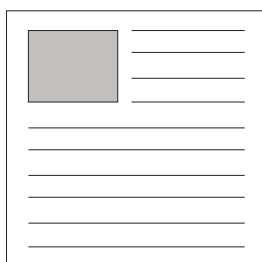
Popis sa zobrazí namiesto obrázku v prípade, že sa obrázok nedá zobrazit'. V niektorých prehliadačoch sa ukáže aj vtedy, ak na obrázok prideme šipkou a chvíľu počkáme. Najlepšie je dať ho ku každému obrázku, pričom popíšeme, čo na obrázku vidíme. Ak je na obrázku tvoja fotka, alternatívny popis môže byť napr. „Moje foto“.

**Úloha 1:** Vytvor si vo svojom domovskom priečinku priečinky, podobné tým z nášho príkladu. Skopíruj do nich stránku a obrázky.

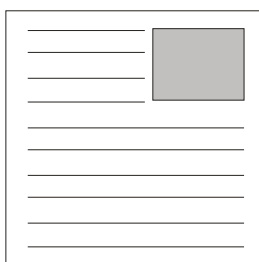
**Úloha 2:** Vlož do stránky aspoň 3 rôzne obrázky s alternatívnymi popismi.

## Vlastnosti obrázka

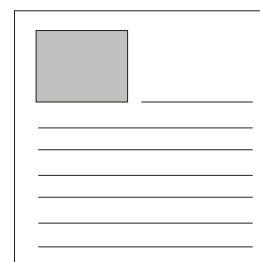
Ďalšie vlastnosti obrázka (veľkosť, zarovnanie a pod.) môžeme zmeniť pomocou štýlu.



`float: left`



`float: right`



`float: none`

Vlastnosť `float` nastavuje „obtekanie obrázku“ v texte. Keď je `left`, obrázok **pláva** (*float*) na ľavom okraji textu, keď je `right`, je to presne naopak. Text okolo obrázku **obteká**. Keď však dáme `float: none`, obrázok bude sedieť na riadku spolu s písmom!

**Úloha 3:** Jeden obrázok v stránke zarovnaj tak, aby ho text obtekal. Vyskúšaj aj ostatné spôsoby zarovnania.

```
height: 100px; width: 150px
```

**Výška** (*height*) a **šírka** (*width*) obrázka. Tieto dve vlastnosti sú najlepšie kamarátky, budeme ich vídať často spolu. Pomocou nich môžeme obrázok ľubovoľne naťahovať a stláčať. Dôležité však je to, že prehliadač pozná jeho rozmery, takže mu môže dopredu (ešte pred načítaním stránky, čo pri veľkých obrázkoch môže trvať veľa) vyhradiť miesto. Keď rozmery nepozná, musí stránku prekresliť v momente, keď má načítaný celý obrázok.

**px** (pixel, obrazovkový bod) sú jednotky dĺžky. Pišu sa vždy hneď za číslo (bez medzery!)

**Úloha 4:** Jednému z obrázkov nastav šírku na 50px a výšku na 500px. Čo urobí obrázok, ak mu nastavíme len **jeden** z rozmerov?

```
margin: 10px
```

**Okraj** (*margin*) určuje veľkosť medzery medzi obrázkom a okolitým textom.

```
border-width: 1px;
border-style: solid;
border-color: red
```

**Rámik** (*border*) je čiara okolo obrázka. Má viacero vlastností: **hrúbka** (*border-width*) sa zadáva v px; **štýl** (*border-style*) môže byť plný (*solid*), bodkovaný (*dotted*), čiarkovaný (*dashed*), dvojité (*double*). **farba** (*border-color*) môže byť anglický názov farby alebo číselný zápis.

Nastavenie okraja môžeme napísať aj skráteno. Vyššie uvedený príklad by sme zapísali takto:

```
border: 1px solid red
```

Vlastnosti okraja musia byť v tomto poradí a musia byť všetky.

```
Padding: 10px
```

**Vsadenie** (*padding*). Určuje vzdialenosť obrázku od okraja. Týmto spôsobom môžeme vytvoriť napr. efekt bieleho okraja, ako to často vidíme na fotografiách:

```
img {
    background-color: white;
    margin: 10px;
    padding: 10px;
    border: 1px solid gray
}
```

## Tapeta

Obrázok do pozadia sú jednoduchá záležitosť:

```
body {
    background-image: url(cesta_obrazku)
}
```

A je to! Obrázok sa odteraz bude opakovať na stránke v smere osi x aj y. Bude to vyzeráť ako tapeta v obývačke. Niekedy sa však môže hodiť opakovanie len v jednom smere, prípadne aby sa obrázok vôbec neopakoval. Potom zmeníme vlastnosť *background-repeat*:

```
background-repeat: repeat-x /* vodorovne opakovanie */
```

```
background-repeat: repeat-y      /* zvisle opakovanie */
background-repeat: no-repeat      /* neopakovat pozadie */
```

**Úloha 5:** Vytapetuj si stránku nejakým obrázkom. Pozri sa na stránku a povedz, či obrázok, ktorý si zvolil, je vhodný ako tapeta alebo nie. Prečo? Ako ovplyvňuje obrázok v pozadí čitateľnosť?

**Pre machrov.** Pozadie môžeš nastaviť nielen značke body, teda celej stránke. Ide to v podstate pre každú značku, ktorá definuje nejaký blok (zatiaľ poznáme napr. odstavec, nadpisy, blockquote). Obrázok sa potom kreslí ako pozadie len v určitom bloku.



## 6. Lekcia: Odkazy

Najzaujímavejšou vlastnosťou webstránok sú asi odkazy. Umožnia nám skákať vpred a vzad v texte, ba aj zo stránky na stránku bez toho, že by sme museli prácne hľadať. Tiež riziko, že sa stratíme, je menšie, lebo prehliadač si pamätá históriu navštívených stránok. A práve tieto dve vlastnosti sa dajú popísať jediným slovom: **hypertext**.

### Jednoduchý odkaz

Predpokladajme, že z našej stránky chceme po zakliknutí odkazu odskočiť na nejakú adresu, napr. školy. Stačí napísať značku `<a></a>`:

```
<a href="http://www.smnd.sk">Mojá škola</a>
```

**Značka `<a></a>`.** Táto značka sa nazýva *anchor* (kotva). Asi preto, že ukazuje miesto, kde sa na našu stránku „ukotví“ nejaká iná, takže sú spojené ako loď s pevninou.

**Adresa.** Adresa, na ktorú sa má skočiť, sa zadáva do atribútu `href`. Tento atribút je povinný, teda musíme ho napísať. Inak by prehliadač nevedel, kam skočiť po zakliknutí. Všimni si, že pred adresu píšeme `http://`, čo je názov protokolu, ktorým sa stránky prenášajú. Niekedy tu môže byť aj iný protokol, možno poznáš protokol `https` (šifrovaný `http`), `ftp`... Každý so súbormi pracuje trochu inak.

**Text odkazu.** Kus stránky, ktorý vyznačíme značkou `<a></a>`, sa vykreslí zvýraznený (podčiarknutý). Po kliknutí prehliadač skočí na adresu, ktorú sme zadali v atribúte `href`. Na tomto mieste nemusí byť iba text. Môže to byť takmer čokoľvek, vrátane obrázku. Tak sa dá vyrobiť obrázkový odkaz (klikací obrázok).

### Odkaz na podstránku

**Príklad:** Chceme vyrobiť odkaz:

Obyvateľsvo Atlantídy

Po kliknutí na odkaz by sa mala zobrazit' stránka nášho webu s informáciami o Atlantíd'anoch.

```
<a href="obyvatelstvo.html">Obyvatelstvo Atlantídy</a>
```

**Pozor, pozor!** Pri názvoch súborov veľký pozor na **veľké a malé písmená**. Stránka s názvom `stranka.htm` **nie je** to isté ako `Stranka.htm`, webservery, na ktorých sa stránky nachádzajú totiž väčšinou robia rozdiely medzi malými a veľkými písmenami.

**Pozor, pozor 2!** Ak v názve súborov používaš **medzery alebi diakritiku (mäkčene a dĺžne)**, rýchlo sa to oduč, pretože si môžeš vyrobiť pekne problémy. Štandardné webservery totiž medzery v adresách netolerujú. Často majú problém aj s diakritikou. Namiesto medzery môžeš použiť znak `_` alebo `-`, aleb špeciálny zápis `%20`, čo je vlastne zápis ASCII kódu medzery v šetnástkovej sústave. Písmená píš bez diakritiky.

## Odkaz na e-mail

Niekedy môžeš vidieť odkaz, na ktorý sa klikne a otvorí sa poštový klient (napr. Outlook), a v ňom je hneď vyplnená adresa, prípadne aj predmet mailu. Robí sa to tak, že ako adresa odkazu sa použije e-mailová adresa a k tomu čarovné slovíčko `mailto:`:

```
<A href="mailto:fusakel@smnd.sk?subject=tvoja_stranka">Pošli mi mail</a>
```

**Pozor!** Ak nechceš, aby sa ti zvýšilo množstvo spamu (nevyžiadanej pošty) v tvojej mailovej schránke, *nikdy* na stránku nepíš svoju e-adresu priamo (tak ako vyššie uvedenom príklade). Problém je v tom, že spameri používajú špeciálne programy (crawlers), ktoré preliezajú webstránky a vyberajú z nich e-adresy, a tak si budujú zoznam obetí, ktoré potom „bombardujú“ reklamou.

## Odkaz v rámci jednej stránky

Keď obsahuje tvoja stránka viac textu, alebo má na začiatku obsah, pre zlepšenie orientácie je dobré zadať skoky na jednotlivé kapitoly, prípadne skoky na začiatok stránky.

**Príklad:** ak chceme skočiť na určité miesto na stránke, najprv toto miesto označíme špeciálnou **nepárovou** formou značky a:

```
<a name="zalozka" />
```

**Atribút name.** Týmto sme do stránky vložili „záložku“. Týchto záložiek môže byť viac a na rôznych miestach v tele stránky, musia však mať rôzne mená. Tu tiež platí, že **meno záložky nesmie obsahovať medzeru**. Keď sme potom niekde v texte (napr. na obsahu), a chceme sa odkázať na určité miesto (napr. na začiatok konkrétnej kapitoly), napíšeme odkaz :

```
<a href="#kapitola1">1. kapitola: Plníme stránku textom</a>
```

Môžeš skákať na záložku aj do inej stránky. Potom napíšeš:

```
<a href="ina_stranka.html#zalozka">slok na zalozku na inej stranke</a>
```

**Úloha 1:** Vyskúšaj všetky uvedené príklady. Prípadne si vyrob vlastné.

**Úloha 2:** Podľa svojho návrhu projektu z 0. lekcie vytvor html stránky, pre každú sekciu tvojej stránky jednu a ktomu aj úvodnú stránku, ktorú pomenuj `index.html`. **Do úvodnej stránky** vlož Odkazy na všetky podstránky tvojho webu.

## Štýl odkazov

Na stránkach často vidíme farebné premenlivé odkazy, ktoré sa menia pri príchode myši, pri klinutí, po navštívení.

Skúsime to teda aj my. V nasledujúcom príklade použijeme novinku – **pseudotriedy**, po našom by sa dalo napísať **stavy**. Hovoria o tom, ako sa má kresliť značka, ktorá sa nachádza v určitom stave. Naše odkazy budú mať rovnaké písmo, ale bude sa im meniť farba a podčiaknutie:

```
a:link { color: green; text-decoration: none; }  
a:hover { color: green; text-decoration: underline }  
a:active { color: red; text-decoration: underline }  
a:visited { color: gray }
```

Pseudotriedy sú súčasťou kaskádových štýlov, a môžeme nimi jednoducho oživiť a rozhybať stránku. Názov pseudotriedy je hneď za názvom značky, oddelený dvojbodkou. Pre náš odkaz sme použili niekoľko pseudotried (stavov):

<b>:link</b>	hovorí o tom, ako bude odkaz vyzerat' v kludovom stave;
<b>:hover</b>	nad odkazom je práve kurzor myši (šípka);
<b>:active</b>	odkaz v čase kliknutia;
<b>:visited</b>	navštívený odkaz.

**Úloha 3:** Vlož tento štýl do stránky. Zobraz ho a pozoruj, čo sa stane pri prechode myši. Pridaj do štýlu pravidlo, ktoré pri prechode myši písmo stuční.

Pseudotriedy nemusia byť len pri odkazoch. Niektoré z nich, napríklad **:hover** sa dajú použiť aj pri iných elementoch. Potom napríklad odstavec alebo riadok tabuľky môže zmeniť farbu, keď sme nad ním myšou. Žiaľ, nemusí to fungovať vo všetkých prehliadačoch. Napríklad v Internet Exploreri fungujú pseudotriedy **:hover** IBA pre odkazy. (Opera, Firefox a iné prehliadače nemajú problém).

**Úloha 4:** Napíš štýl, ktorý pri prechode myši zmení farbu pozadia odstavca, nad ktorým sa práve myš nachádza. Skús vylepšiť štýl tak, že odstavec bude mať navyše i jednobodový rám. (Ak zvyčajne používaš Internet Explorer, použi tentoraz radšej iný prehliadač).

**Úloha 5:** Vo svojej stránke nastav správanie odkazov podľa svojej fantázie. Daj pozor na dobrú čitateľnosť.

## 7. lekcia: Zoznamy a tabuľky

Pri písaní stránok často potrebujeme vypísať zoznam, či už s odrážkami, číslovaný a podobne. Aj hrml má značky špeciálne na písanie takýchto zoznamov.

**Značka <li>.** Je spoločná pre všetky druhy zoznamov. Označuje položku zoznamu (*list item*), ktorá sa zobrazí s odrážkou, prípadne číslovaním. Môže obsahovať text, odstavce textu, obrázky, dokonca ďalšie zoznamy. Je to párová značka, píše sa dovnútra zoznamu <ul>, <ol>.

**Odrážky – neusporiadaný zoznam (unordered list):** kreslí sa s odrážkami.

```
<ul>
  <li>Broskyňa</li>
  <li>Čerešňa</li>
  <li>Ananás</li>
</ul>
```

- ◆ Broskyňa
- ◆ Čerešňa
- ◆ Ananás

**Číslovanie – usporiadaný zoznam (ordered list):** namiesto odrážky sa kreslí poradie.

```
<ol>
  <li>Ananás</li>
  <li>Broskyňa</li>
  <li>Čerešňa</li>
</ol>
```

1. Ananás
2. Broskyňa
3. Čerešňa

**Definičný zoznam (definition list):** používa sa napríklad na slovníky a encyklopédie. Stretávame sa s ním však minimálne.

```
<dl>
  <dt>Jabĺko</dt>
  <dd>Druh ovocia, plod jablone.
  Jednoduchý dužinatý plod, zaraďuje sa
  medzi malvice.</dd>
  <dt>Hruška</dt>
  <dd>Je to plod ovocného stromu hruška.
  Patrí medzi malvice.</dd>
</dl>
```

<b>Jabĺko</b>	Druh ovocia, plod jablone. Jednoduchý dužinatý plod, zaraďuje sa medzi malvice.
<b>Hruška</b>	Je to plod ovocného stromu hruška. Patrí medzi malvice.

Zoznamy môžu byť do seba **vnorené**. Treba však dať pozor, aby sa značky neprekrývali!

```
<ol>
  <li>Jabĺko
    <ol>
      <li>Golden delicious
      <li>Rubin
    </ol>
  </li>
  <li>Hruška
```

1. Jabĺko
  - ◊ Golden delicious
  - ◊ Rubin
2. Hruška

```
<ul>
  <li>Golden delicious</li>
  <li>Rubín</li>
</ul>
</li>
<li>Hruška</li>
</ol>
```

**Úloha 1 (1 b):** Vyskúšaj všetky uvedené príklady.

Vzhľad zoznamov. Obyčajný usporiadaný alebo neusporiadaný zoznam sa kreslí s medzerou pred aj za sebou, a tiež je odsadený od okraja. Vnorený zoznam sa odsadzuje ďalej.

## Štýl pre zoznamy

Na formátovanie zoznamov používame niekoľko vlastností:

**Vlastnosť `list-style-type`.** Táto vlastnosť určuje, aký znak sa bude zobrazovať ako odrážka alebo čím sa bude číslovať. Záleží to na tom, či je zoznam neusporiadaný alebo usporiadaný.

**Odrážky:** `list-style-type: disc | circle | square;`

**Číslovanie:** `list-style-type: decimal | upper-roman | lower-roman | upper-alpha | lower-alpha;`

**Nič:** `list-style-type: none;`

Príklad:

```
<ul style="list-style-type: square">
  <li>Táto odrážka bude štvorček</li>
</ul>
```

**Vlastnosť `list-style-position`.** Určuje, či sa bude zobrazovať odrážka „zapustená“ v zozname, alebo „vytrhnutá“ von.

`list-style-position: outside | inside.`

**Úloha 2 (2 b):** Vyskúšaj si tieto možnosti pre značku `ul` i `ol`.

**Úloha 3:** Nastav **pre všetky usporiadané zoznamy naraz** číslovanie veľkými rímskymi číslicami, zapustenými do položiek zoznamu.

**Vlastnosť `List-style-image`.** Namiesto odrážky môžeme nastaviť ľubovoľný (najlepšie malý) obrázok. Príklad:

```
list-style-image: url (obrazky/odrazka.gif);
```

**Úloha 4:** Vytvor si malý obrázok typu `.jpg`, `.gif` alebo `.png` a použij ho namiesto odrážok pre všetky neusporiadané zoznamy na stránke.



## 8. lekcia: Tabuľky

Tabuľky sú jedna z najzložitejších štruktúr v html. Ak zvládneš túto lekciiu, môžeš si povedať, že si začínajúci borec a v html sa už nestratíš.

**Príklad:** Chceme vyrobiť takúto jednoduchú tabuľku:

ovocie	1. miesto	2. miesto	3. miesto
domáce	jablká	hrušky	slivky
cudzokrajné	ananás	banán	mango

V html to bude vyzerat' takto:

```
<table cols="4" rows="3">
  <tr>
    <th>ovocie</th><th>1. miesto</th><th>2. miesto</th><th>3. miesto</th>
  </tr>
  <tr>
    <th>domáce</th><td>jablká</td><td>hrušky</td><td>slivky</td>
  </tr>
  <tr>
    <th>cudzokrajné</th><td>ananás</td><td>banán</td><td>mango</td>
  </tr>
</table>
```

**Značka <table></table>.** Táto značka označuje, že to bude tabuľka. Jej atribúty cols a rows udávajú počet stĺpcov (columns) a riadkov (rows), vrátane hlavičky (tučným písmom).

**Značka <tr></tr>.** Označuje riadok tabuľky (table row). Malo by ich byť toľko, koľko je riadkov v tabuľke.

**Značka <th></th>.** Označuje hlavičku stĺpca alebo riadku. Hlavička sa kreslí odlíšená, tu na obrázku je zvýraznená tučným písmom na sivom pozadí.

**Značka <td></td>.** Označuje políčko tabuľky (table data). To, čo do nej dáme, sa zobrazí v políčku. Malo by ich byť v každom riadku toľko, koľko máme stĺpcov. V našom prípade sú v každom riadku po tri. Keď dáme menej, nevykreslí sa ani prázdne políčko.

**Úloha 1:** Prepíš túto tabuľku do stránky.



Zatiaľ sa nám tabuľka zobrazuje „nenašminkovaná“, a nevyzerá veľmi pekne. Pridáme preto kaskádové štýly a nastavíme farby a rámiky. Nezabudnite, že tieto štýly patria do značky `style` v hlavičke html stránky!

```
table      { border: 3px solid green; border-collapse: collapse }
th, td     { border: 1px solid blue; text-align: center }
th         { background-color: lightgreen }
```

**Okraj celej tabuľky** (table) sme nastavili na hrúbku 3px a farbu zelenú. Urobili sme to pomocou skráteného zápisu štýlov – zlúčili sme dohromady vlastnosti `border-width`, `border-style` a `border-color`. Vznikla z toho vlastnosť `border`, ktorej hodnoty zapíšeme do jedného riadku, ale vždy v pevnom poradí. Pre okraj je toto poradie vždy hrúbka štýl\_okraja farba. Hodnoty oddeľujeme medzerami.

**Vlastnosť `border-collapse: collapse`** hovorí, že rámiky políček budú spojené.

**Vlastnosti hlavičky a políčka.** Jednobodový modrý okraj, zarovnanie textu na stred políčka sme nastavili pre obidve značky naraz. Názvy značiek, pre ktoré nastavujeme spoločnú vlastnosť, sa píše spolu, oddelené čiarkou. **Všimni si**, že zarovnáva sa nie na stred stránky, ale na stred značky, ktorú formátujeme.

**Vlastnosti len pre hlavičku:** Podfarbenie políčka (`background-color`) nastavujeme len pre značku `th`, preto ju píšeme ako osobitné pravidlo. Ostatné vlastnosti, ktoré sme nastavili predtým, ostanú zachované.

**Úloha 2:** Vlož do svojej stránky tabuľku. Pripíš do stránky aj horeuvedený kaskádový štýl.

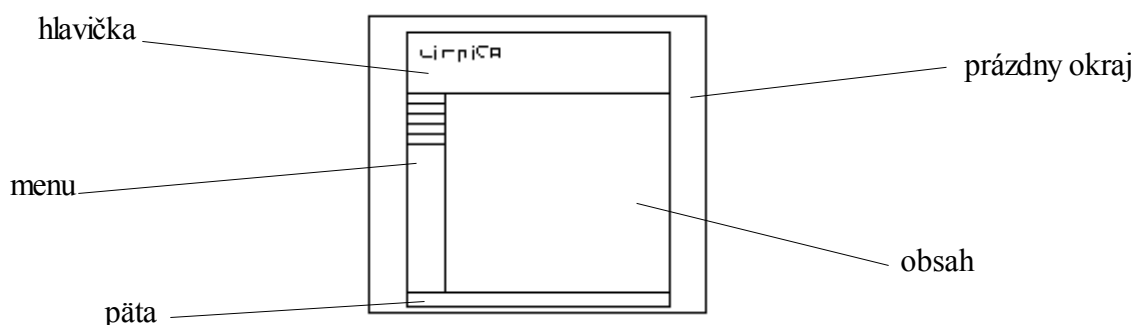
**Úloha 3:** Naformátuj tabuľku podľa vlastnej fantázie. Neboj sa používať farby a štýly okraja.

**Úloha 4:** Ukáž, ako by tabuľka vyzerala, keby sme nenastavili vlastnosť `border-collapse`.

**Úloha 5:** Vyskúšaj a popíš čo sa stane, keď do niektorej `td` alebo `th` pripíšeš atribút `colspan="2"` alebo `rowspan="2"`, prípadne obidve. Vyskúšaj použiť rôzne čísla a rôzne bunky.

## 9. lekcia: Bloky

V nasledujúcich dvoch lekciách skúsime vyrobiť stránku, akú bežne môžeme vidieť na internete. Vyzerá asi takto:



**Všimni si**, že celá stránka sa to skladá z obdĺžnikov – blokov. To je základný postup, ktorý použijeme. Stránku si rozdelíme do blokov, ktoré potom naformátujeme tak, aby sa poskladali do výsledného tvaru.

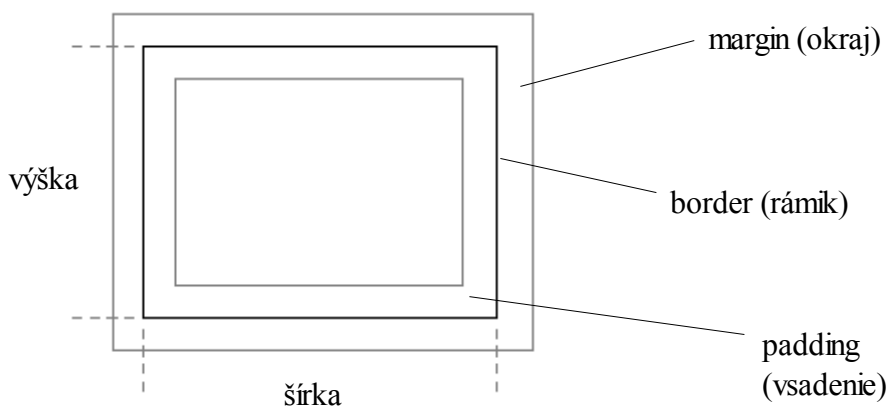
Bloky môžeme rozdeliť na dva druhy: špecifické a nešpecifické. So **špecifickými blokmi** sme sa stretávali doteraz, napríklad odstavce, nadpis, obrázok. Ich spoločnou črtou je, že tieto značky sa oddelia a odlišia od ostatného textu. Každý špecifický blok má nejakú špeciálnu vlastnosť.

**Nešpecifický blok** nemá žiadne vlastnosti. Na stránke ho spočiatku ani nezbadáte.

**<div></div>**. Značka nešpecifického bloku. Názov pochádza z anglického *divisor* – oddeľovač. Okrem toho, že označený text oddelí od ostatného, nemá žiadne vlastnosti. Tie jej musíme nastaviť pomocou štýlu.

**Úloha 1:** Do ľubovoľnej stránky, kde je viac ako 2 odstavce textu. Do stránky vpiš značku div tak, aby mala v sebe nejaký text. Pozri si výsledok v prehliadači.

Každý blok má niekoľko vlastností. Čiastočne sme sa s nimi stretli už pri obázkoch (príklad špecifického bloku). Tieto vlastnosti možno nastavovať pre ľubovoľný blok na stránke.



**Výška a šírka bloku:** určujú, aký veľký bude blok na stránke. Zadávaajú sa v rôznych jednotkách: napr. px, pt, em, alebo %.

width: 10px;  
height: 1em;

Ak výšku a šírku bloku nezadáme, budú mať hodnotu auto, čo znamená, že sa prispôbia šírke stránky a obsahu, ktorý je do nich vložený. Šírka bude najväčšia možná a výška taká, aby sa do bloku vošlo všetko, čo je v ňom uzavreté.

**Úloha 2:** Nastav bloku z úlohy 1 šírku 50%.

**Okraj, rámik a vsadenie.** Tieto vlastnosti určujú, ako bude blok vyzerat', a aký bude mať vzťah k svojmu obsahu a k svojmu okoliu. **Okraj (margin)** hovorí, aký bude odstup bloku od ostatného textu, ktorý ho môže napríklad obtekať. **Rámik (border)** určuje hrúbku, štýl a farbu čiary, ktorou bude rámik nakreslený. **Vsadenie (padding)** zas udáva, či bude obsah rámika nalepený celkom na čiare, alebo bude mať okolo seba aj nejaký „vzduch“.

**Najčastejší spôsob použitia:**

```
margin: 10px;          /* okraj okolo celého bloku na 10px */
border-width: 1px;     /* rámik hrúbky 1px */
border-style: solid;    /* štýl rámika plná čiara (solid) */
border-color: red;      /* červená farba */
padding: 0.5em;        /* vsadenie na polovicu šírky písmena „M“ (jednotka em) */
```

Štýl okraja môže byť rôzny: solid (plná čiara), dotted (bodkovaný), dashed (čiarkovaný) a pod. Podrobne si to môžeš nájsť na internete (=> border-style).

**Okrem toho všetky vlastnosti môžeme nastaviť aj pre každú stranu bloku zvlášť.**

```
margin-top: 0;         /* horný a ľavý okraj na nulu, */
margin-right: 1em;     /* pravý a dolný okraj na 1 a 2 em */
margin-bottom: 2em;
```

margin-left: 0;

Podobne môžeme nastaviť aj border a padding (border-top, border-right, border-bottom, border-left, padding-top, padding-right, padding-bottom, padding-left).

**Tieto vlastnosti môžeme zapísať do jedného riadku, dokonca sa dajú v istých prípadoch aj zjednodušiť:**

margin: 0 1em 2em 0; /\* to isté, čo vyššie, ale v jednom riadku – pozor na poradie \*/

padding: 0 10%; /\* horný a dolný okraj žiadny, pravý a ľavý 10% šírky nadradeného bloku \*/

**Pozor:** pri zjednodušenom zápise okraja nastavujeme len jeho šírku. Ak chceme na každej strane iný okraj, treba to nastaviť jednotlivo.

**Úloha 3:** Do stránky vlož jednu alebo viac značiek div s textom vo vnútri. Tipni si, čo z nich vyrobí nasledujúci štýl:

```
div {  
    width: 80px; padding: 2px; background-color: lightgray; text-align: center;  
    border-bottom: 2px solid #555; border-right: 2px solid #555;  
    border-left: 2px solid #aaa; border-top: 2px solid #aaa;  
}
```

```
div:hover {  
    padding: 3px 1px 1px 3px; text-align: center;  
    border-bottom: 2px solid #aaa; border-right: 2px solid #aaa;  
    border-left: 2px solid #555; border-top: 2px solid #555;  
}
```

**Úloha 4:** (pre machrov, 2b) Aká bude celková šírka bloku, keď mu nastavíme šírku na 100px, okraj na 10px, vsadenie 10px a rámik 2px. (Porovnaj šírku v MSIE a iných prehliadačoch. Je rovnaká?)

## 11. lekcia: Stilllllpce!

No a môžeme začať s rozmiestňovaním blokov. Stránku pomocou značky `div` rozdelíme do blokov., podľa obrázku na začiatku tejto lekcie. Každý blok bude mať iné vlastnosti, preto ich musíme od seba odlišiť. Použijeme na to nový atribút `id` (identifikátor).

```
<div id="stranka">

  <div id="hlavicka">
    <h1>Jožinkov projekt o Atlantíde</h1>
  </div>

  <div id="menu">
    <a href="atlantida.html">Atlantída</a>
    <a href="geografia.html">Geografia</a>
    <a href="obyvatelstvo.html">obyvatelstvo</a>
  </div>

  <div id="obsah">
    <h2>Novinky</h2>
    <p><strong>30. 1. </strong></p>Pridaný prvý článok.</p>
    <p><strong>22. 1. </strong>Nový dizajn stránok.</p>
  </div>

  <div id="pata">
    (c)2007 Jožinko. | <a href="kontakty.html">Kontakt na webmastera</a>
  </div>

</div> <!-- stranka -->
```

**Atribút `id`.** Tento atribút nazývame identifikátor, alebo jednoznačné pomenovanie html značky. Môže ho mať ľubovoľná html značka, ale nesmie sa rovnaký názov vyskytovať viackrát. To preto, lebo sa používa na rozlíšenie viacerých rovnakých prvkov. V našom prípade na rozlíšenie viacerých značiek `div`.

**Html poznámka.** Občas, hlavne v rozsiahlejších stránkach sa stane, že si potrebujeme niečo v kóde poznačiť (okomentovať), napríklad pre ľahšiu orientáciu. Vtedy použijeme špeciálnu značku poznámka, ktorá vyzerá takto: `<!-- text poznámky -->`.

## Každému čo jeho je

Teraz pridáme každému bloku kaskádový štýl:

```
body {  
    background-color: lightblue;  
}  
  
div#stranka {  
    background-color: white;  
    margin: 1em 20%; border: 1px solid navy; padding: 10px;  
}
```

Toto je základ stránky. Pozadie celého okna nastavíme značke body (ako vždy). Potom nastavíme vlastnosti **pre div s menom stránka – zapisujeme to div#stranka** (to je presne to meno, ktoré sme použili v atribúte id pre daný div). Tento blok bude mať biele pozadie, jednobodový modrý okraj a vsadenie na 10px.

**Všimni si**, že nastavenie okraja nastaví hodnoty symetricky, to znamená, že blok **stranka** bude v okne prehliadača vždy vycentrovaný. A nech sa stane čokoľvek, jeho okraje budú vždy 20% zo šírky okna.

```
div#hlavicka {  
    height: 5em;  
    background-color: lightblue;  
    padding: 1em;  
}
```

Toto bude hlavička stránky, bude sa zobrazovať vždy navrchu, na všetkých stránkach bude rovnako graficky spracovaná – podľa toho bude návštevník vedieť, že sa nachádza na našej stránke. Tu sme nastavili výšku hlavičky na 5em (šírka písmena M v písme, ktoré používa tento blok), pozadie na svetlomodré, vsadenie na 1em. Okraj a rámik nie je nastavený.

```
div#menu {  
    float: left;  
    width: 200px;  
    border-right: 1px solid lightblue;  
}
```

**Plávajúci blok.** Aby sme mohli umiestniť menu vedľa textu, povieme mu, že má „plávať“. Keď blok pláva, všetko, čo nasleduje v stránke za ním, ho bude „obtekať“. To znamená, že sa to vykreslí vedľa neho, podľa toho, na ktorej strane náš blok plávať. Vykreslí sa to samozrejme len v prípade, že sa to do medzery, ktorá vedľa bloku zostala, zmestí.

Náš blok pláva naľavo (`float: left`) a všetko ostatné (blok **obsah** a blok **pata**) budú plávať vedľa neho.

```
div#obsah {  
    padding-left: 10px;  
}
```

Tento blok bude obsahovať **hlavný text stránky**. Aby nebol text „nalepený“ až na okraj, nastavíme teda aj vsadenie (`padding: 1em`).

Všimni si, že ak do bloku dáme viac textu a menu nie je príliš vysoké, text začne po skončení menu „tiecť“ aj pod ním. To asi nie je veľmi estetické, my sme zvyknutí na to, že menu je v samostatnom stĺpci. Preto bloku nastavíme ešte vlastnosť `margin`:

```
div#obsah {  
    margin-left: 200px;  
    padding-left: 10px;  
}
```

Vďaka širokému, 200px okraju nám pod menu zostane biele miesto až potiaľ, kým neskončí hlavný obsah stránky. Je to akoby sme umiestnili menu do okraja druhého bloku.

```
div#pata {  
    background-color: lightblue;  
}
```

**Päta.** Tento blok bude mať opäť bledomodré pozadie. Hlavička sa kreslí na konci stránky. Keď bude menu príliš dlhé a hlavný text krátky, môže sa stať, že hlavička sa nám namiesto naspodku stránky vykreslí vedľa menu. Aby sme predišli tejto chybičke, povieme päte, že sa má vykresliť až vtedy, keď bude mať po oboch stranách „prázdno“, teda žiadny blok:

```
div#pata {  
    background-color: lightblue;  
    clear: both;  
}
```

Vlastnosť **clear** pripúšťa okrem hodnoty **both** aj hodnoty **left** a **right**.

A o tom, ako nastaviť hlavičku, a päť, ako vyrobiť pekné menu, povieme si podrobnejšie v nasledujúcej lekcii.

**Úloha 1 (3b):** Sformátujte svoju stránku do takéhoto stĺpcového výzoru.

**Úloha 2 (2b):** Pridajte do stránky tretí stĺpec (vpravo), ktorý bude sformátovaný rovnako ako stĺpec s menu, ale na pravej strane obsahu.

## **12. lekcia: Vnútro blokov, hlavička menu a päta**

Externé štýly v súbore

CSS formát menu – display: inline a block

Previazanie viacerých stránok medzi sebou





## **Menu**

animované efekty, CSS rollovery

Galéria obrázkov

- lo-res a hi-res bitmapy?, rozmery obrázkov, náhľady a ich význam

**klikacie mapy** špecialitka pre machrov

**advanced: optimalizácia** GIF, PNG (podľa root.cz?), JPG

Odrážky a číslovanie

## **Galéria obrázkov**

-- vlastnosť float a radenie obrázkov



## **Tabuľky**

- motivácia: odprezentovať niečo v tabuľke
- návrh jednoduchkej tabuľky s popismi stĺpcov
- spájanie políčok cez riadky/stĺpce
- CSS: nastavenie vzhľadu tabuľky
- CSS: vyfarbovanie riadkov/políčok
- nastavenie obrázka do pozadia



## **Formuláre**

– form, input (rôzne...), textarea...





## **Meta značky**

## **Validácia kódu**



## **JavaScript**

– základná syntax, deklarácia a umiestnenie v rámci stránky

**Udalosti a zavesenie Javascriptu na ne** – zmena farby/pozadia jedným kliknutím

DOM: ako vyzerá webstránka znútra

Ako sa dostať k prvku na stránke?

OnChange, onClick a ako sa používajú