

4. Jazyk HTML

Pri prezeraní webových stránok asi každého napadla myšlienka: Aké by to bolo, mať vlastnú stránku? To, že čítate tieto riadky, znamená, že Vy ste prešli od myšlienky k činu.

V tejto a v nasledujúcej kapitole sú informácie, ktoré Vám pomôžu pri vytváraní svojej stránky. Táto kapitola je zameraná na vytvorenie štruktúry stránky. V nasledujúcej kapitole je popísané použitie kaskádových štýlov, ktoré umožňujú vzhľad stránky ľubovoľne upravovať.

4.1. Vytvorenie prvej webovej stránky

Táto kapitola Vám ukáže, že je naozaj ľahké vytvoriť si jednoduchú stránku. Túto stránku budeme postupne v ďalších kapitolách upravovať a rozširovať. Po každej úprave si stránku pozrite v internetovom prehliadači. Nebojte sa experimentovať - tak si najrýchlejšie osvojíte, čo ktorý príkaz na stránke znamená.

4.1.1. Čo je dobré vedieť pred vytvorením prvej stránky

Obsah, ktorý sa zobrazuje na webových stránkach je uložený v súboroch s príponou htm alebo html. HTML je skratka pre Hyper Text Markup Language. V názve tohto jazyka sú slová, ktoré spôsob ako sú webové stránky vytvárané dosť dobre charakterizujú.

„Hyper Text“ je mechanizmus, ktorý zrejme stojí za úspechom webových stránok. Informácie v knihe alebo v časopise sú uložené sekvenčne – postupne za sebou. Čitateľ takéhoto textu je viac menej nútený prejsť celým textom, pokiaľ sa dostane k tomu, čo potrebuje. Na webových stránkach (v hyper texte) sa nachádzajú odkazy (linky), ktoré smerujú na iné stránky alebo na iné miesto na aktuálnej stránke. Tento spôsob je pre čitateľa výhodný, pretože nevyžaduje veľa úsilia a času aby sa dostal k informáciám, ktoré hľadá a nezdržoval sa čítaním toho čo ho nezaujímá. Podrobnejšie informácie o hypertexte a histórii webu sa nachádzajú v kapitole 1.16.

Ďalším slovným spojením je markup language – značkovací jazyk. Časti textu (nápisy, odseky, zoznamy, obrázky atď.), ktoré sa majú zobrazit' na stránke sú špeciálne označené aby prehliadač poznal štruktúru dokumentu a správne celý dokument zobrazil. Používaným značkám sa často hovorí tagy, čo pochádza z anglického slova tag = značka.

Tak ako ostatné veci, ktoré používa človek, prechádzajú vývojom a zlepšovaním, tak sa aj jazyk HTML vyvíjal a jeho posledná verzia je 4. 01. V súčasnosti sa používa čoraz viac jazyk XHTML – (x = eXtensible). HTML jazyk sa často používal nielen na popisovanie štruktúry dokumentu ale aj na jeho vizuálnu úpravu (nápisy zobrazit' veľkým a tučným písmom, odseky zarovnať do bloku atď.). Jazyk XHTML striktne vyžaduje označovať iba štruktúru textu. To ako sa majú zobrazovať jednotlivé časti dokumentu je popísané pomocou štýlov. Viac je o tejto problematike napísané v kapitole 5.

Takto dôjde k oddeleniu formy od obsahu. Výhodou je, že odborník na danú problematiku dodá obsah a dizajnér upraví vzhľad. Ďalšou výhodou je, že ak sa rozhodneme zmeniť vzhľad stránok, nie je nutné zasahovať do samotného obsahu. Ak sú štýly vhodne použité, tak je ľahké udržiavať jednotný vzhľad všetkých firemných stránok a prípadne ich vzhľad naraz zmeniť bez nutnosti úpravy jednotlivých stránok.

Jazyk HTML a XHTML sú podobné a v mnohých ohľadoch zhodné. HTML jazyk má ale mnoho značiek, ktoré sú v XHTML zbytočné alebo priamo zakázané. Najčastejšie sú to (ako sa dá z predchádzajúceho textu vytušiť) značky popisujúce vzhľad. My sa takýmito značkami zaoberať nebudeme.

Ďalšou zmenou je, že v jazyku HTML je jedno či je značka napísaná malými alebo veľkými písmenami, v XHTML je nutné písať všetky značky malými písmenami. Ostatné podstatné zmeny alebo odlišnosti sú uvedené postupne v tejto kapitole.

Stránky je možné vytvárať rôznymi postupmi, ktoré možno rozdeliť podľa toho, aký editor použijeme pri ich vytváraní:

- Nešpecializovaný textový editor. Voľba, ktorá znamená, že môžete okamžite začať s tvorbou stránok⁷⁵. Notepad (Poznámkový blok) máte v počítači. Nič iné okrem chuti a trocha vytrvalosti nepotrebujete.
- Editor, ktorý zvyrazňuje syntax – pozná značky používané v (X)HTML dokumentoch a farebne ich odlišuje od ostatného textu. Výhodou je jednak to, že ľahko od seba odlišíme samotný obsah a jednotlivé značky a tak ľahšie nájdeme prípadnú chybu. Oblíbené editory tohto typu sú napríklad PSPad alebo Crimson editor – oba sú zadarmo dostupné na Internete.
- Špecializované HTML editory Vám môžu priamo ponúkať jednotlivé značky, samozrejmosťou je farebné zvýraznenie značiek, umožňujú náhľad ako vyzera stránka bez nutnosti prezerania v internetovom prehliadači. Často ponúkajú možnosť úpravy stránky bez znalosti značiek – metódou WYSIWYG⁷⁶. Takýto postup pri tvorení stránok je často kritizovaný, pričom dôvodom kritiky je hlavne neprehľadnosť výsledného kódu a jeho prípadná následná úprava. Medzi najznámejšie editory z tejto skupiny bezpochyby patria Dreamweaver a MS FrontPage.

Pre úplnosť je potrebné uviesť, že na vytvorenie stránky bez znalosti (X)HTML, teda metódou WYSIWYG, je možné použiť aj MS Word. Stačí na stránku vložiť želaný obsah a vybrať si pri ukladaní (Uložiť ako) typ súboru html.

4.1.2. Prvá stránka

Stránky v knihe budeme vytvárať v textovom editore Poznámkový blok (Notepad). Otvorte nový súbor a napíšte kód z príkladu 1:

Príkazy `<html>` a `</html>` sú značky (tagy). Prvá je úvodná, druhá je ukončovacia. Dôležité je používať lomené zátvorky a nie iné. Značka `<html>` hovorí prehliadaču, kde sa nachádza začiatok HTML dokumentu a značka `</html>` kde je jeho koniec.

⁷⁵ Na tvorbu stránok a ich následné zobrazenie nie je potrebné pripojenie na Internet.

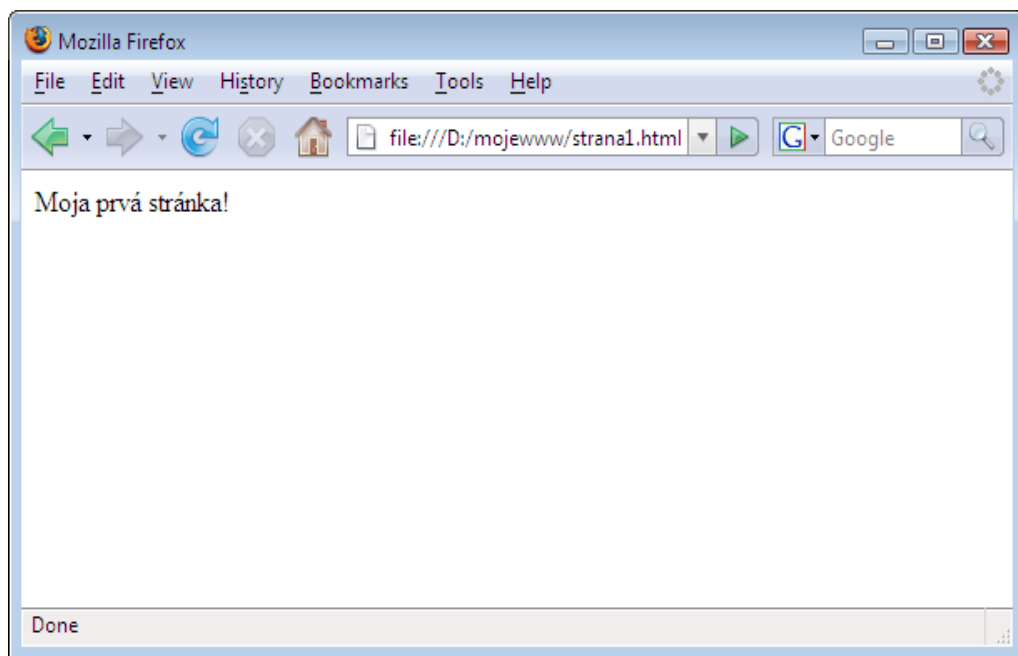
⁷⁶ What You See Is What You Get – čo vidíš, to dostaneš. Metóda spočíva v umiestňovaní objektov na stránku, podobným spôsobom ako sa vkladá obrázok alebo navrhuje tabuľka do dokumentu MS Word. Podobne ako vo Word je možné upravovať vlastnosti jednotlivých objektov na stránke (typ a farba písma, pozadie, rozmery obrázkov a pod.)

Príklad 1: Súbor strana1.html

```
<html>
  Moja prvá stránka!
</html>
```

Uložte súbor pod menom „strana1.html“ a zapamätajte si kde ste ho uložili. V knihe sme pre všetky ukážky vytvorili adresár „mojewww“ na disku D:, teda súbor je uložený ako „D:\mojewww\strana1.html“. Ak Notepad ponúkne možnosť „Uložiť vo formáte txt“, je nutné zmeniť spôsob uloženia na „Všetky súbory“, pretože inak by sa súbor mohol uložiť ako strana1.html.txt.

Pri pomenovaní súborov (a aj adresárov) nie sme viazaní skoro žiadnymi obmedzeniami ale je výhodné uspokojiť sa s malými písmenami a číslami, nepoužívať medzery a hlavne diakritiku. Neskôr, keď budete dávať stránky webový server na Internete, si tak môžete ušetriť trocha času pri zisťovaní, že server chce, aby ste napísali „moje%20www“, pričom Vy ste adresár pomenovali „moje www“.



Obr. 4.1. Stránka z príkladu 4.1 zobrazená v prehliadači Mozilla Firefox

Ak je súbor uložený, tak je niekoľko spôsobov ako ho zobraziť v internetovom prehliadači:

- Spustiť prehliadač a do okna s adresou napísať
 - „d:\mojewww\strana1.html“ alebo
 - file:///d:/mojewww/strana1.html.

- Spustiť prehliadač a do okna s adresou napísať „D:“, stlačiť kláves Enter a postupne sa preklikať až na súbor „strana1.html“.
- Spustiť prehliadač a v menu *súbor* vybrať *otvoriť* (často dostupné po stlačení Ctrl+O), nájsť a vybrať Vašu prvú stránku.
- V prieskumníkovi alebo niektorom správcovi súborov nájsť Váš súbor a dvakrát naň poklepať.

Výsledok je znázornený na obrázku 4.1.

Windows má pomerne nepríjemnú vlastnosť nezobrazovať v štandardnom nastavení prípony známych typov súborov. To môže viesť v našom prípade k dvom dôsledkom:

- V prieskumníkovi sa nezobrazí „strana1.html“ ale iba „strana1“.
- o niečo horšie je, ak je zobrazené len „strana1.html“ ale v skutočnosti sa súbor nazýva „strana1.html.txt“ a prípona „txt“ sa nezobrazuje. Tento problém nastane, ak ste pri ukladaní súboru nezmenili možnosť „Uložiť vo formáte txt“, na možnosť „Všetky súbory“.

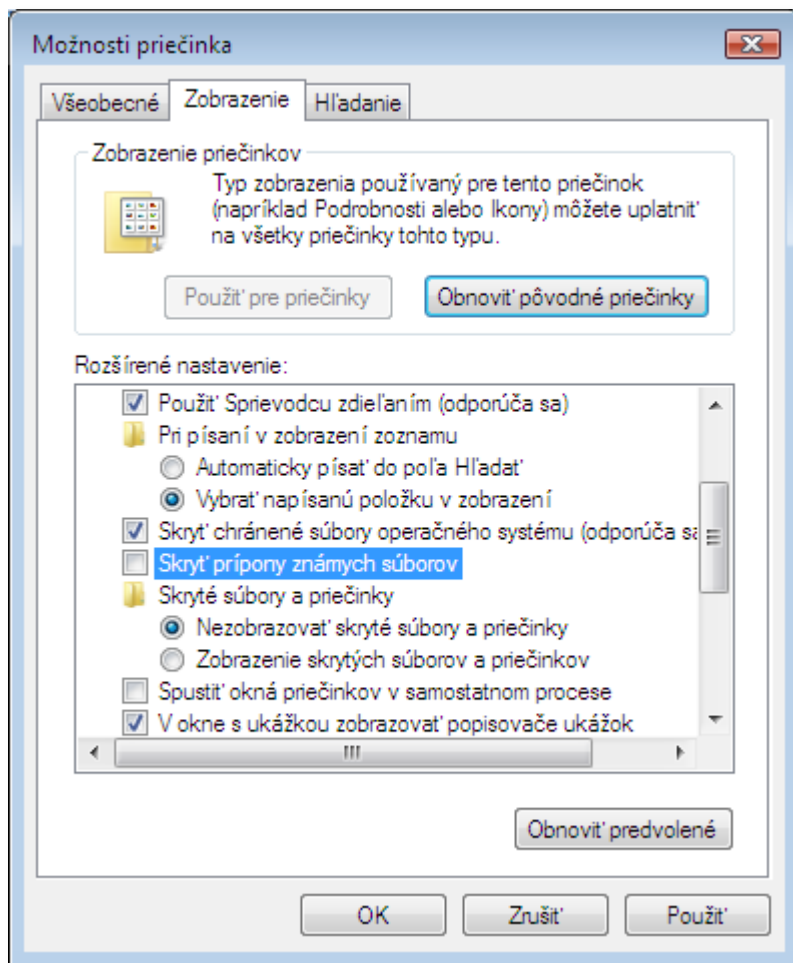
Tomuto problému sa je možné vyhnúť nastavením zobrazovania prípon známych typov súborov, tak ako je to znázornené na obrázku 4.2.

Toto nastavenie nám zároveň zabráni neželanému spusteniu programu, ktorý by sa mohol volať napríklad „ferdo.jpg.exe“, pričom nám sa zobrazuje iba „ferdo.jpg“. Takto sa maskujú niektoré druhy malware.

Poznámka: Je možné, že sa Vám niektoré znaky s diakritikou nezobrazia korektne. Zatiaľ túto skutočnosť budeme ignorovať. Vrátime sa k nej v kapitole 4.4. Pre rýchlu nápravu vyberte v menu prehliadača Zobrazit/Kódovanie (View/Character Encoding) a zvolte kódovanie: stredoeurópske jazyky - Windows 1250, prípadne vyskúšajte ostatné kódovania, kým sa stránka nebude zobrazovať správne. Bude to kódovanie, ktoré Váš editor použil pri ukladaní súboru.

Zdrojový kód ľubovoľnej stránky si môžete pozrieť cez menu prehliadača *Zobraziť/Zdrojový kód (View/Page source)*.

Náš zápis z prvého príkladu je veľmi zjednodušený a preto sa radšej hneď pozrieme na kosťu bežnej stránky z pohľadu značiek. Na nasledujúcom výpise je zachytená základná štruktúra (X)HTML dokumentu.



Obr. 4.2 Nastavenie zobrazovania prípon známych typov súborov

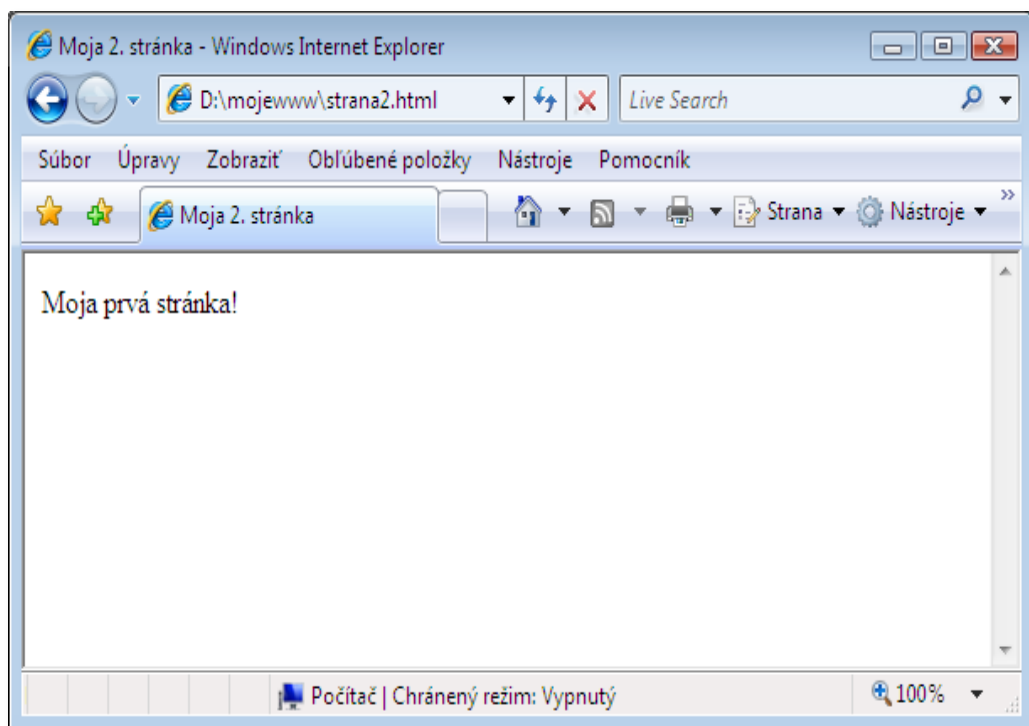
Príklad 2: Súbor strana2.html

```
<html>
  <head>
    <title>
      Moja 2. Stránka
    </title>
  </head>
  <body>
    Moja prvá stránka!
  </body>
</html>
```

Dokument z príkladu č. 2 sa skladá z dvoch častí. Prvou je časť hlavičiek (všetko medzi `<head>` a `</head>`), kde bývajú uložené informácie, ktoré sa priamo na stránke nezobrazujú ale môžu napríklad ovplyvniť spôsob zobrazenia (napríklad ako zobrazovať znaky s diakritikou), prípadne sú tu uvedené kľúčové slová, ktoré majú pomôcť internetovým vyhľadávateľom ako je napríklad Google, lepšie poznať danú stránku. V príklade č. 2 je v sekcii `<head>` značka `<title>`, podľa ktorej prehliadač vie, čo má zobraziť ako titulok v záhlaví okna. Výsledok je znázornený na obrázku 4.3.

Druhou časťou je sekcia `<body>`, v ktorej je umiestnený obsah stránky.

V príklade č. 2 sú zodpovedajúce si začiatkové a koncové značky pod sebou. Sú odsadené od okraja podľa toho, aké je ich postavenie na stránke. Značka `html` je najviac vľavo, pretože (priamo) obsahuje značky `head` a `body`, tak sú tieto značky odsadené. V rámci značky `head` je značka `title`, ktorá je odsadená od svojej nadradenej značky (`head`). A napokon samotný text titulu je odsadený od značky `title`. Analogicky je to aj v prípade značky `body` a textu „Moja prvá stránka!“.



Obr. 4.3 Súbor `strana2.html` zobrazený v MS Internet Explorer

Ako je ukázané na nasledujúcich príkladoch, na počte medzier a odsadení značiek nezáleží. Takáto úprava ale sprehľadňuje zdrojový kód a pomáha pri hľadaní jednej z najčastejších chýb, ktorou býva neukončenie značky. Preto aj špecializované editory (ale aj napríklad PSPad) ponúkajú automatické preformátovanie kódu takýmto spôsobom. V nasledujúcich príkladoch budeme odsadzovanie dodržiavať – s malou výnimkou. Úvodnú

4.2.1. Odsek

Internetové prehliadače ignorujú viacnásobné medzery a redukujú ich na jednu. Konce riadkov a tabulátory sa spracovávajú ako medzera⁷⁷. Ak chceme písať text do odsekov, a teda používať konce riadkov, je jednou z možností urobiť to pomocou značiek `<p>` a `</p>`, ktoré označujú začiatok a koniec odseku. Použitie je na ukázané na príklade č. 6 a obrázku 4.4.

Príklad 6: Súbor strana6.html

```
<html><head><title>Moja 6. stránka</title></head>
<body>
  <p> Je možné, že sa Vám niektoré znaky s diakritikou
  nezobrazia korektne. Zatiaľ túto skutočnosť budeme ignorovať.
  Vrátime sa k nej na konci kapitoly. </p>
  <p> Pre rýchlu nápravu vyberte v menu prehliadača
  Zobrazit/Kódovanie (View/Character Encoding) a zvolte
  kódovanie: stredoeurópske jazyky - Windows 1250, prípadne
  vyskúšajte ostatné kódovania, kým sa stránka nebude zobrazovať
  správne.</p>
  <p>Internetové prehliadače ignorujú viacnásobné medzery
  a redukujú ich na jednu. Konce riadkov a tabulátory považujú za
  medzeru. Ak chceme písať text do odsekov a teda používať konce
  riadkov, je nutné urobiť to pomocou značiek.</p>
</body>
</html>
```

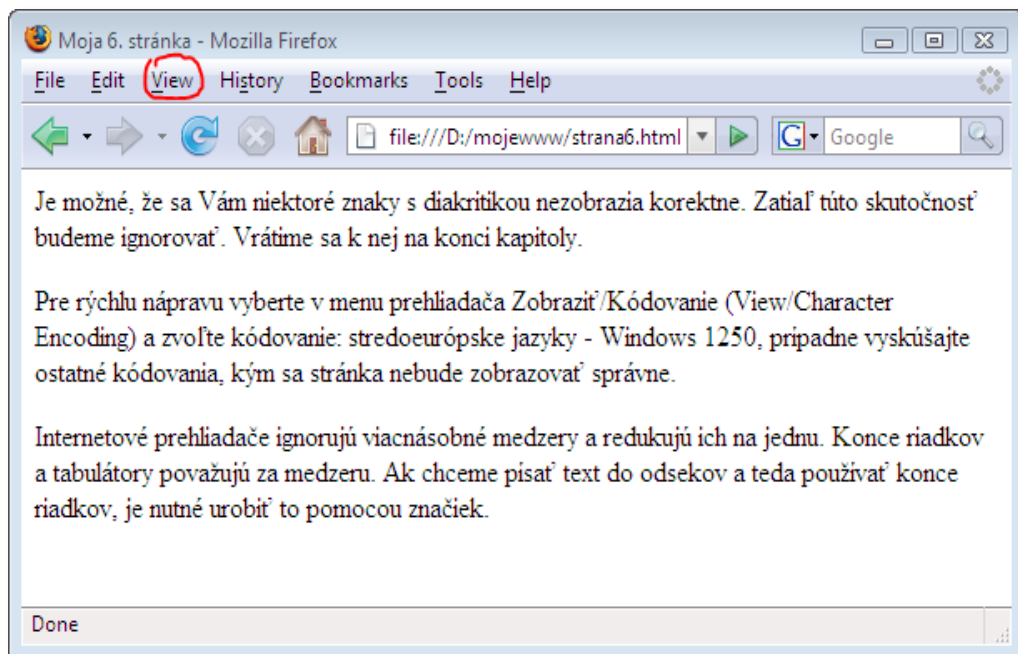
4.2.2. Koniec riadku

Niekedy je vhodné ukončiť riadok ale neukončiť odsek. Pre takéto prípady slúži značka `
`. Zatiaľ sa všetky značky skladali z dvoch častí: značky začiatku a značky konca, ktorá sa odlišovala od značky začiatku pridaním lomky (slash, /). Takéto značky sa nazývajú párové. Jazyk HTML pozná aj nepárové značky. Jednou z nich je zalomenie riadku `
`. Štandard XHTML ale vyžaduje aby všetky značky boli ukončené, čo sa rieši pridaním lomky pred pravú lomenú zátvorku. Použite značky `
` je znázornené na príklade č. 7.

Príklad 7: Súbor strana7.html

```
<html><head><title>Moja 7. stránka</title></head>
<body>
  A <br /> B <br /> C <br /> D <br /> E <br /> F
</body>
</html>
```

⁷⁷ Znak medzera, koniec riadku a tabulátor sa nazývajú WhiteSpace. Whitespace znaky sa v (X)HTML redukujú na jednu medzeru.



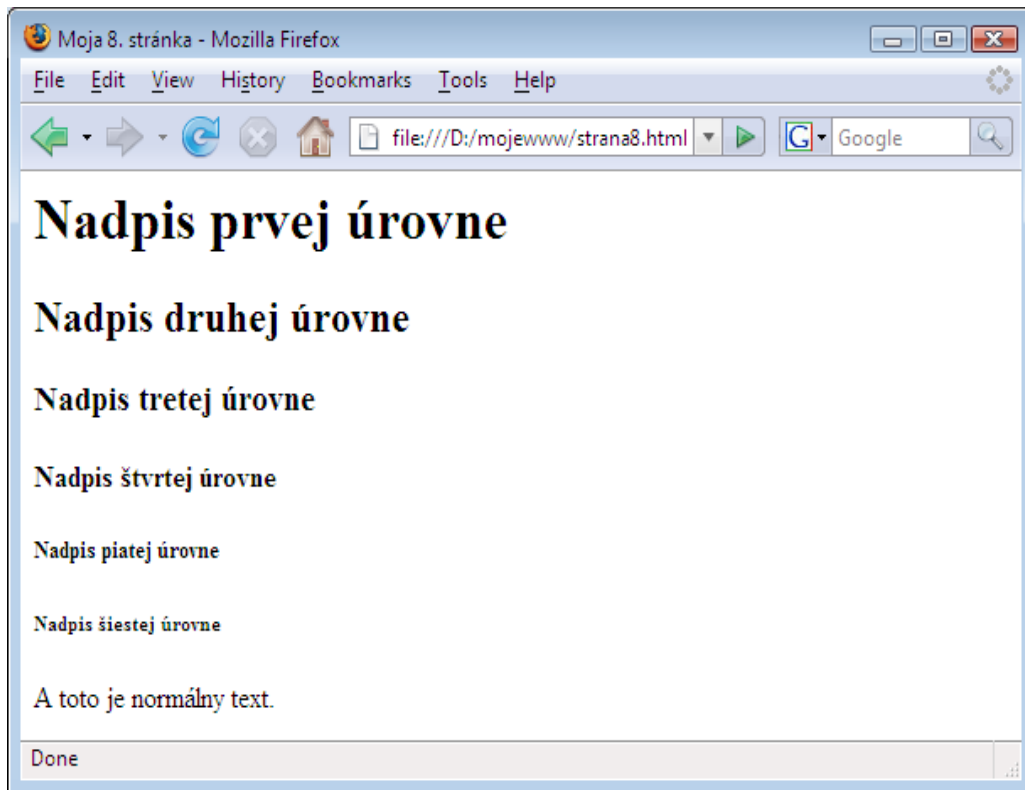
Obr. 4.4 Strana6.html v prehliadači Mozilla Firefox

4.2.3. Nadpisy

Snáď každý textový dokument obsahuje nadpisy. Nadpisy prvej úrovne sú väčšie a nápadnejšie ako nadpisy druhej úrovne, ktoré sú zasa výraznejšie ako nadpisy tretej úrovne atď. Nadpisy popisujú za nimi nasledujúci text v dokumente a členia ho na menšie celky. Použitie nadpisov prvej až šiestej úrovne na webovej stránke je znázornené na príklade č. 8 a výsledok na obrázku 4.5.

Príklad 8: Súbor strana8.html

```
<html><head><title>Moja 8. stránka</title></head>
<body>
  <h1> Nadpis prvej úrovne </h1>
  <h2> Nadpis druhej úrovne </h2>
  <h3> Nadpis tretej úrovne </h3>
  <h4> Nadpis štvrtej úrovne </h4>
  <h5> Nadpis piatej úrovne </h5>
  <h6> Nadpis šiestej úrovne </h6>
  A toto je normálny text.
</body>
</html>
```



Obr. 4.5. Nadpisy prvej až šiestej úrovně

Je dôležité upozorniť, že rôzne prehliadače na rôznych zariadeniach môžu jednotlivé úrovně nadpisov zobrazovať rozdielne. Napríklad prehliadač v PDA s malým displejom môže zobrazovať nadpis namiesto použitia veľkého písma použitím veľkých písmen alebo inej farby. Ako už bolo na začiatku kapitoly spomenuté, značky popisujú štruktúru textu, preto nie je vhodné využívať značky určené pre nadpisy na zvýraznenie obyčajného textu. Zmena formátu (farba, výška, rez písma a pod.) obyčajného textu ale aj napríklad nadpisov bude ukázaná v kapitole venovanej CSS.

4.2.4. Umiestnenie obrázka na stránke

Pre umiestnenie obrázka na stránke je určená značka *img*. Značka *img* je opäť príkladom nepárovej značky – preto je ukončená „`</>`“. Značka *img* má navyše oproti doteraz spomenutým značkám aj vlastnosti, ktorým priradíme určité hodnoty. Najlepšie je vysvetlenie na konkrétnom príklade.

Zápis `` znamená, že na pozíciu značky sa vloží obrázok, pričom samotný obrázok je uložený v súbore `pes.jpg`. Vlastnosť *src* (zdroj, source) hovorí nielen ako sa súbor nazýva, ale aj kde je uložený. V tomto prípade sa súbor `pes.jpg` musí nachádzať v rovnakom adresári ako súbor, v ktorom je značka napísaná. Všetky naše stránky sú uložené v adresári `d:\mojawww\`, preto sa aj obrázok musí nachádzať v tomto adresári.

Ak by sa obrázok nachádzal v nadradenom adresári (v našom prípade priamo v koreňovom adresári disku D:) zápis by vyzeral ``. Dve bodky a lomka v tomto prípade hovoria internetovému prehliadaču aby hľadal obrázok v adresári, ktorý je o úroveň vyššie.

Dôležité je použiť lomku (/) a nie spätnú lomku (backslash, \), čo by nemuselo vždy (hlavne po umiestnení stránky na web server) fungovať.

Ak je naopak potrebné zobrazit' obrázok v niektorom podadresári, použije sa zápis ``. V danom prípade sa obrázok nachádza v adresári d:\mojawww\zvieratka\.

Ak by sa zvieratká (adresáre) členili ešte na domáce a divé, zápis by mohol vyzerať nasledujúcim spôsobom: ``. Tento zápis predpokladá, že v aktuálnom adresári je podadresár zvieratka a v ňom ďalší podadresár domace, v ktorom sa nachádza náš obrázok.

Obrázok sa samozrejme môže nachádzať aj vo vedľajšom adresári ako je aktuálny súbor html. Ak by to bol napríklad adresár d:\zvieratka, v takom prípade by sme použili zápis ``. Zápis je možné prečítať: vyjde z aktuálneho adresára (to sú dve bodky, vyjde sa z adresára mojawww) a vojde do adresára zvieratka.

Ak sa obrázok nachádza v hlavnom adresári aktuálneho disku, je možné použiť aj zápis: ``. Kde počiatočná lomka znamená hlavný adresár aktuálneho disku. Na serveri, ktorý zrejme nebudete mať iba pre seba to znamená najvyššie umiestnenie, na ktoré máte práva.

Tieto spôsoby určenia súboru sa nazývajú relatívne adresovanie a zápisy podľa toho relatívne adresy. Relatívne (vzťahujúce sa na niečo) preto lebo sa odvíjajú od miesta, kde je uložená stránka, v ktorej sú tieto adresy zapísané.

Ak sa obrázok nachádza na inom serveri je nutné uviesť absolútnu (celú) adresu a aj protokol akým ho je možné získať. V ukážke je použitý najčastejší protokol http. Napríklad:

```

```

Uvedenie protokolu je povinné, pri jeho vynechaní:

```

```

 by sa súbor pes.jpg hľadal v podadresári s názvom google.com.

Nezapisujte cestu k súboru s použitím názvu disku: ``. Ak sa obrázok nachádza v hlavnom adresári na disku D:, tak Vám ho internetový prehliadač (v závislosti od druhu a verzie) na Vašom počítači zobrazí. Ale v prípade, že dáte takúto stránku na web server a niekto si ju bude pozerať, tak sa mu obrázok pes.jpg nezobrazí. Internetový prehliadač nebude hľadať súbor na jeho disku.

Okrem určenia zdroja odkiaľ zobrať obrázok, vyžaduje štandard XHTML aj určenie veľkosti obrázka (*width* - šírka, *height* - výška) a uvedenie alternatívneho textu popisujúceho obrázok (*alt*) – to je dobré pre používateľov, ktorí majú vypnuté zobrazovanie obrázkov.

Použitie je znázornené v príklade č. 9. Rozmery obrázku sú v príklade uvedené v pixloch, t. j. v bodoch. Je možné použiť aj iné jednotky. Ich spresnenie a použitie je rozobrané v kapitole o kaskádových štýloch.

Príklad 9: Súbor strana9.html

```
<html><head><title>Moja 9. stránka</title></head>
<body>
  <h1> Ukážka použitia obrázkov na stránke </h1>
  
  
</body>
</html>
```

4.2.5. Odkazy

Odkazy alebo tiež linky sú soľou webových stránok. V tejto podkapitole sú popísané spôsoby práce s nimi. Vzhľadom na to, že odkazy často smerujú z aktuálnej stránky na iné stránky, je vhodné oboznámiť sa s relatívnym a absolútnym adresovaním. Oba spôsoby boli rozobrané v predchádzajúcej podkapitole.

Najjednoduchšie je vytvoriť odkaz na už existujúcu stránku na Internete. V príklade č. 10 je ukážka stránky z odkazom umožňujúcim po kliknutí naň dostať sa na stránku www.google.com.

Podobne ako značka *img* má aj značka *a* vlastnosti, ktorým je potrebné priradiť hodnoty. Základnou vlastnosťou je *href*. Táto vlastnosť udáva, na ktorú stránku nás prehliadač presmeruje po kliknutí na text (alebo obrázok), ktorý je medzi značkou `` a ``.

V príklade č. 10 sa zobrazí na stránke pod nadpisom slovo Google, na ktoré je možné kliknúť a prejsť na stránku [google.com](http://www.google.com). V prípade, že by sme zmenili slovo Google na Yahoo, zmenilo by sa zobrazované slovo, ale kliknutím naň by sme sa presunuli opäť na [google.com](http://www.google.com), pretože hodnota *href* je ostala nezmenená.

Príklad 10: Súbor strana10.html

```
<html><head><title>Moja 10. stránka</title></head>
<body>
  <h1> Ukážka použitia odkazov </h1>
  <a href="http://www.google.com" > Google </a>
</body>
</html>
```

Stránka z príkladu č. 11 využíva, že už máme hotovú stránku č. 9 (strana9.html) a umožňuje na ňu vytvoriť odkaz a prejsť. Predpokladá sa, že stránky č. 9 a 11 sa nachádzajú v rovnakom adresári. Ak sú v rozdielnych adresároch, tak je nutné použiť relatívne adresy - postup ako to urobiť je popísaný v predchádzajúcej podkapitole.

Príklad 11: Súbor strana11.html

```
<html><head><title>Moja 11. stránka</title></head>
  <body>
    <h1> Ukážka použitia odkazov v rámci servera </h1>
    <a href="strana9.html" > Choď na stránku č. 9 </a>
  </body>
</html>
```

Zatiaľ odkazy v oboch ukázkach vykonali presun na začiatok stránky. Pomocou vlastnosti *name* môžeme na stránke zdefinovať miesta, na ktoré potom možno odkazovať z iných alebo z pôvodnej stránky. Postup odkazu v rámci jednej stránky je znázornený v príklade č. 12. Na začiatku stránky je odkaz na jej koniec:

` Choď na koniec stránky `. Konštrukcia odkazu sa podobá na odkaz na inú stránku. Medzi značkami je text, na ktorý je možné kliknúť. Miesto, na ktoré sa odkazuje, je tiež uvedené pomocou vlastnosti *href*.

Na konci stránky je vlastnosťou *name* definované miesto, kam je možné smerovať odkazy. ` koniec stránky `. Vlastnosť *name* pomenuje časť dokumentu, na ktoré je možné odkazovať.

Všimnite si, že pri odkaze sa pred meno miesta, na ktoré sa odkazuje, vloží znak #. Je to pre odlišenie, že nejde odkaz na stránku s názvom „dole“, ale o odkaz v rámci stránky. Súbor nemusí mať v názve typ (príponu) a „dole“ je regulárne pomenovanie súboru.

Príklad 12: Súbor strana12.html

```
<html><head><title>Moja 12. stránka</title></head>
  <body>
    <h1> Ukážka použitia odkazu do stránky </h1>
    <a href="#dole" > Choď na koniec stránky </a>
    Veľa textu <br /> Veľa textu <br /> Veľa textu <br />
    Veľa textu <br /> Veľa textu <br /> Veľa textu <br />
    Veľa textu <br /> Veľa textu <br /> Veľa textu <br />
    Veľa textu <br /> Veľa textu <br /> Veľa textu <br />
    <br /><br /><br /><br /><br /><br /><br /><br />
    Veľa textu <br /> Veľa textu <br /> Veľa textu <br />
    Veľa textu <br /> Veľa textu <br /> Veľa textu <br />
    Veľa textu <br /> Veľa textu <br /> Veľa textu <br />
    Najlepšie, keď bude textu viac ako na jednu obrazovku <br
  />
    <a name="dole" > koniec stránky </a>
  </body>
</html>
```

Odkaz do vnútra inej stránky je možné vykonať nasledujúco: Predpokladajme, že nasledujúci odkaz sa nachádza v súbore stranaxy.html, ktorá sa nachádza v rovnakom adresári ako stránka strana12.html

```
<a href="strana12.html#dole"> Chod' na koniec stránky č. 12</a>
```

Na príklade č. 13 je ukážka ako je možné skombinovať odkaz a obrázok, tak aby sa po kliknutí na obrázok otvorila nová stránka. V príklade je zobrazený obrázok pes.jpg z podadresára zvieratka. Pretože ho obklopuje značka `<a>`, tak po kliknutí naň sa zobrazí stránka uvedená vo vlastnosti `href`, t. j. strana10.html.

Príklad 13: Súbor strana13.html

```
<html><head><title>Moja 13. stránka</title></head>
<body>
  <h1> Ukážka skombinovania odkazu a obrázku</h1>
  <a href="strana10.html">
    
  </a>
</body>
</html>
```

4.2.6. Zoznamy

Zoznamy sú analógiou odrážok ako ich poznáme textových editorov. Slúžia na vymenovanie jednotlivých bodov z nejakej množiny. Môžu byť číselované alebo nečíselované. V prípade vytvorenia zoznamu nutné použiť dva druhy značiek.

1. Prvý druh určuje, že ide o zoznam a zároveň popisuje či sa jednotlivé body:
 - a) číslujú, vtedy použije sa značka ``...`` alebo
 - b) nečíslujú, v takom prípade sa používa značka `` ...``.
2. Druhým druhom sú značky popisujúce každý jednotlivý bod zoznamu: ``...``.

Použitie je znázornené v príklade č. 14 a obrázku 4.6.

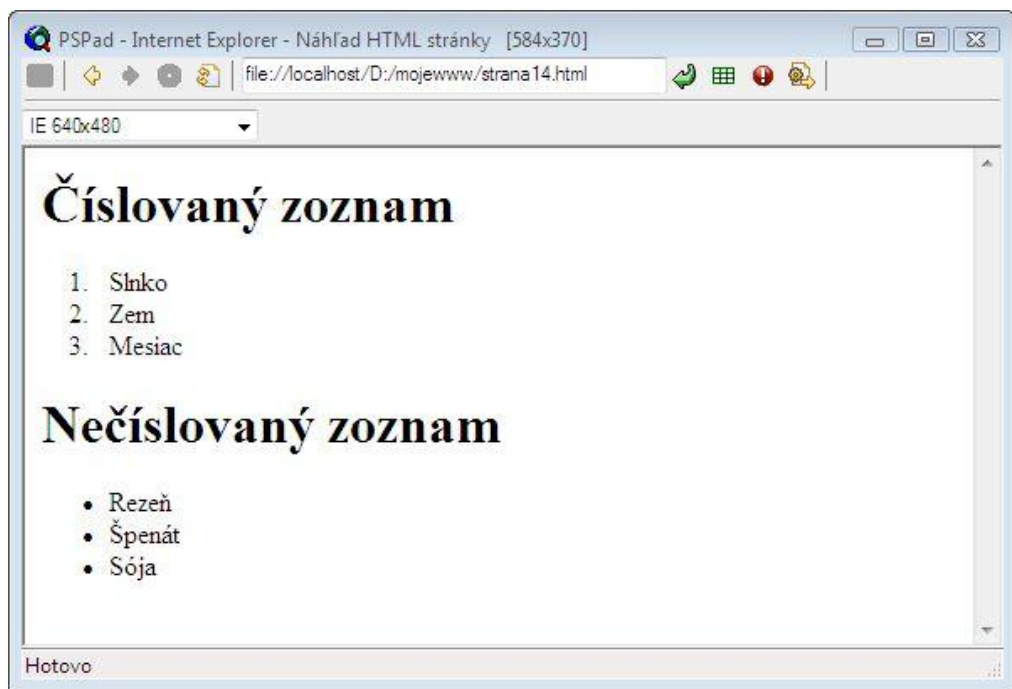
Príklad 14. Súbor strana14.html

```
<html><head><title>Moja 14. stránka</title></head>
<body>
  <h1> Číslovaný zoznam</h1>
  <ol>
    <li>Slnko</li>
    <li>Zem</li>
    <li>Mesiak</li>
  </ol>
  <h1> Nečíselovaný zoznam</h1>
  <ul>
    <li>Rezeň</li>
    <li>Špenát</li>
  </ul>
</body>
</html>
```

```

        <li>Sója</li>
    </ul>
</body>
</html>

```



Obr. 4.6 Zoznamy v prevedení náhľadovej funkcie editoru PSPad

Je možné vytárať aj viac úrovňové zoznamy vnorením značiek *ol* alebo *ul* do značky *li*. Typy zoznamov možno ľubovoľne kombinovať. Použitie je znázornené na príklade č. 15. a obrázku 4.7.

Príklad č. 15. Súbor strana15.html – Viac úrovňové zoznamy.

```

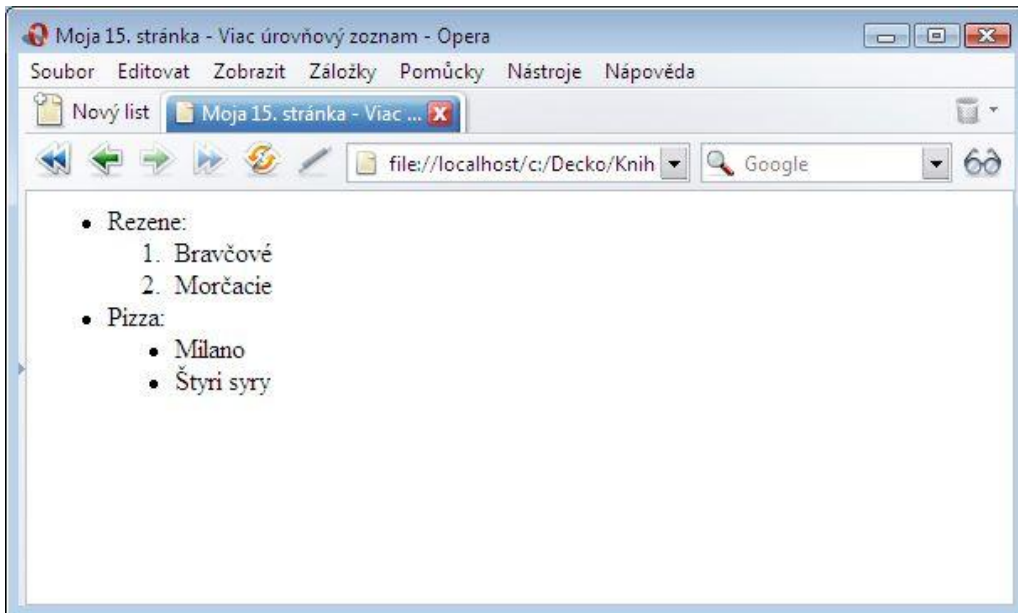
<html>
  <head> <title>Moja 15. stránka - Viac úrovňový zoznam</title>
</head>
  <body>
    <ul>
      <li>Rezene:
        <ol> <li>Bravčové</li> <li>Morčacie</li>
        </ol>
      </li>
      <li> Pizza:
        <ul> <li>Milano</li> <li>Štyri syry</li>
        </ul>
      </li>
    </ul>

```

```

</ul>
</body>
</html>

```



Obr. 4.7 Viacúrovňové zoznamy z príkladu č. 15, zobrazené v Opere

V HTML existuje pre značky zoznamu niekoľko vlastností, ktoré umožňujú zobraziť rôzne typy číslovania alebo rôzne symboly. Vzhľadom na to, že v XHTML nie sú povolené, nebudeme sa nimi zaoberať. Zoznamy je samozrejme možné upravovať ale tak, ako pri ostatných značkách sa tým budeme zaoberať pre kaskádových štýloch v ďalšej kapitole.

4.2.7. Tabuľky

Tabuľky sú výborným prostriedkom na zobrazenie rôznych druhov dát. Tabuľky v (X)HTML môžu mať ľubovoľný počet riadkov a stĺpcov, je možné susedné riadky alebo stĺpce zlúčiť a je možné tabuľky do seba vnárať. Aj keď je možné nastaviť pre tabuľku alebo aj jednotlivé bunky rôzne okraje, pozadie a pod., túto možnosť opäť preskočíme. Vzhľadom na to, že tabuľky bez formátovania môžu vyzeráť neprehľadne, tak v tomto prípade urobíme malú výnimku a nastavíme hrúbku okrajov na dva pixely (*border = 2px*). Úpravou tabuliek sa zaoberá kapitola o kaskádových štýloch.

Podobne ako pri zoznamoch, tabuľky využívajú viacero značiek. V najjednoduchšom prípade sú to tri páry:

- Značky `<table>` a `</table>` určujú odkiaľ a pokiaľ je kód pre danú tabuľku.
- Značky `<tr>` a `</tr>` (table row) popisujú (definujú) riadok tabuľky.

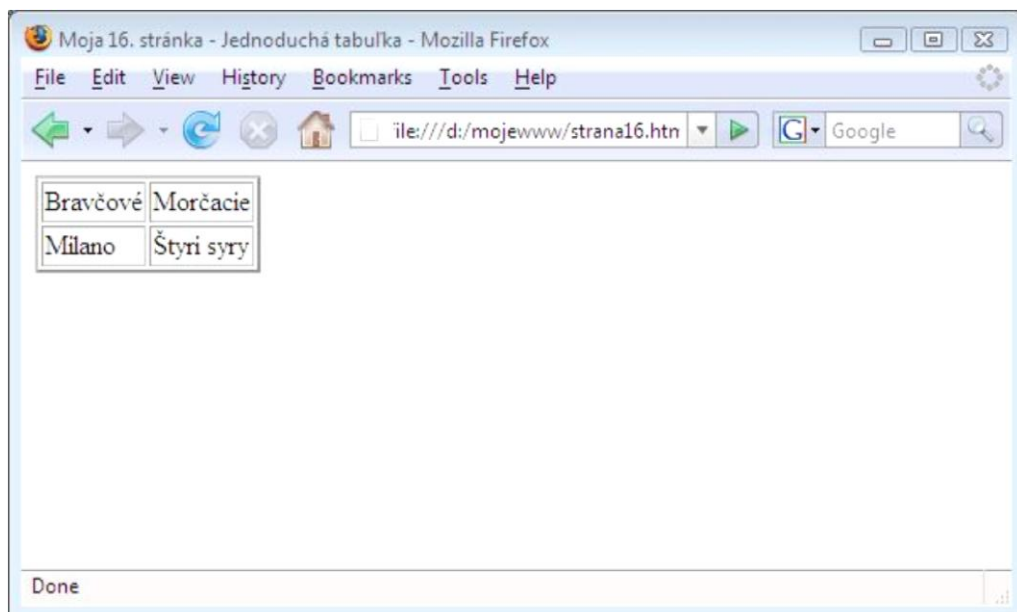
- V každom riadku sa nachádza pre každú bunku pár značiek `<td>` a `</td>` (table data).

Použité značiek je znázornené na príklade č. 15 a obrázku 4.8.

Skúste zväčšiť obsah niektorej bunky, uvidíte, že šírka stĺpca sa automaticky prispôsobuje šírke najširšej bunky v stĺpci.

Príklad č. 16. Súbor strana16.html – Jednoduchá tabuľka

```
<html> <head> <title>Moja 16. stránka - Jednoduchá
tabuľka</title> </head>
<body>
  <table border="2px">
    <tr>
      <td>Bravčové</td>
      <td>Morčacie</td>
    </tr>
    <tr>
      <td>Milano</td>
      <td>Štyri syry</td>
    </tr>
  </table>
</body>
</html>
```



Obr. 4.8 Jednoduchá tabuľka z príkladu č. 16

V prípade, že bunka má funkciu záhlavia, používa sa namiesto značky *td*, značka `<th>` a `</th>` (table header).

Často je potrebné zlúčiť niektoré bunky v riadku alebo stĺpci. V (X)HTML sa to dočieľa pridaním vlastnosti *colspan* alebo *rowspan*, do značky *td* alebo *th*. Hodnotou týchto vlastností je počet riadkov (stĺpcov), ktoré sa majú zlúčiť.

Napríklad zápis `<td colspan="2" rowspan="3">` znamená, že sa zlúčia dve bunky v riadku (inak povedané zlúčia sa dva stĺpce v riadku) a tri bunky v stĺpci (tri riadky v stĺpci). Použité je znázornené v príklade č. 17 a na obrázku 4.9.

Príklad č. 17. Súbor strana17.html – Tabuľka so záhlavím a zlúčenými bunkami

```
<html> <head> <title>Moja 17. stránka - Tabuľka so záhlavím a
zlúčenými bunkami</title> </head>
<body>
  <table border="2px">
    <tr>
      <th></th> <th colspan="2">Sobota</th> <th
colspan="2">Nedeľa</th>
    </tr>
    <tr>
      <th>Variant 1</th>
      <td rowspan="2"> Kapustová <br />polievka</td>
      <td>Bravčový rezeň</td>
      <td rowspan="2"> Hovädzia<br />polievka</td>
      <td>Morčací rezeň</td>
    </tr>
    <tr>
      <th>Variant 2</th>
      <td>Pizza Milano</td>
      <td>Pizza Štyri syry</td>
    </tr>
  </table>
</body>
</html>
```

	Sobota		Nedeľa	
Variant 1	Kapustová	Bravčový rezeň	Hovädzia	Morčací rezeň
Variant 2	polievka	Pizza Milano	polievka	Pizza Štyri syry

Obr. 4.9. Tabuľka s použitím značiek *th*, *colspan* a *rowspan* z príkladu č. 17

4.2.8. Zapisovanie špeciálnych znakov

V situácii, keď je nutné zobrazit' na stránke napríklad znak < dôjde k problémom. Prehliadač bude po znaku < očakávať značku a nemusí zobrazit' stránku správne. Preto sa namiesto znakov, ktoré majú špeciálny význam používajú určité kombinácie znakov. Začínajú znakom ampersand (&) a končia bodkočiarkou (;). To ale znamená, že aj znak ampersand sa musí zapísať špeciálnym spôsobom.

Ďalšou špeciálnou skupinou sú znaky s diakritikou a pomocou špeciálnych znakov sa zapisujú aj znaky ako sú €, Ł, ©, ® a pod.

Tabuľka 4.1 a príklad č. 18 znázorňujú použitie špeciálnych znakov⁷⁸.

Príklad č. 18. Súbor strana18.html – špeciálne znaky

```
<html> <head> <title>Moja 18. stránka - Špeciálne znaky</title>
</head>
<body>
  <table border="2px">
    <tr>
      <th>Znak</th> <th>HTML kód</th> <th>Význam</th>
    </tr>
    <tr>
      <th>&amp;</th>
      <td>&amp;&amp;</td>
      <td>ampersand</td>
    </tr>
    <tr>
      <th>&lt;</th>
      <td>&amp;lt;</td>
      <td>ľavá ostrá zátvorka, znak menší</td>
    </tr>
    ...
    ...
    <tr> <th colspan="3">Znaky s diakritikou</th> </tr>
    <tr>
      <th>&Aacute;</th>
      <td>&amp;Aacute;</td>
      <td>dlhé veľké písmeno A - acute znamená dĺžeň</td>
    </tr>
    ...
    ...
  </table>
</body>
</html>
```

⁷⁸ Na stránke <http://www.w3.org/MarkUp/Guide/Advanced> si je možné pozrieť ostatné prípustné znaky

Tabuľka 4.1. Špeciálne znaky

Znak	HTML kód	Význam
&	&	ampersand
<	<	ľavá ostrá zátvorka, znak menší
>	>	pravá ostrá zátvorka, znak väčší
"	"	úvodzovky
©	©	copyright
£	£	libra
€	€	euro
®	®	registrovaná ochranná známka
™	™	registrovaná obchodná známka
‰	‰	promile
½	½	jedna polovica
¼	¼	jedna štvrtina, analogicky ¾
²	²	druhá mocnina, analogicky ³
	 	nedeliteľná medzera
Znaky s diakritikou		
Á	Á	dlhé veľké písmeno A - acute znamená dĺžku
á	á	dlhé malé písmeno a - analogicky je to aj pre ostatné písmená
Ô	Ô	O s vokáľom
ä	ä	prehlasované písmeno a
š	š	písmeno s s mäkčeňom

4.2.9. Ostatné bežné značky

Do súboru html je možné zapísať si komentáre, ktoré prehliadač ignoruje. Komentáre sa dajú využiť aj pri ladení stránky, keď je nutné dočasne odstrániť (znefunkčniť) časť kódu, aby sa našla prípadná chyba. Značka komentár má nasledujúcu syntax `<!-- Tento text prehliadač ignoruje -->`.

Na oddelenie jednotlivých častí na stránke je možné využiť aj vodorovnú čiaru `<hr />`.

Na zdôraznenie časti textu je možné použiť dvojicu značiek `` a `` alebo `` a ``. Pre časť textu, ktorý bol dodatočne vložený do pôvodného textu sa používa `<ins>` a `</ins>`. Naopak, pre zmazaný text, keď má byť vidno, že pôvodný text bol zmenený sa používa `` a ``.

Na stránkach sa môže vyskytnúť nutnosť zapísať horný alebo dolný index (m^2 , n_4). Okrem možností uvedených v predchádzajúcej tabuľke je možné použiť značky `^{` a `}` pre mocninu alebo horný index a značky `_{` a `}` pre dolný index.

V príklade č. 18 je spomenutá nedeliteľná medzera (` `). V prípade dlhšieho textu sa text v okne prehliadača zalamuje tam, kde je medzera (white space). Ak nechceme aby sa riadok končil napríklad predložkou, je možné napríklad namiesto `v prípade` napísať `v prípade`.

4.3. Document Type Declaration, DTD a validácia stránky

Na počiatku webových prehliadačov bol Netscape, ktorý, keďže bol prakticky jediný (90% podiel), určoval aké značky sa majú používať a ako sa majú zobrazovať. Neskôr pribudli ďalšie prehliadače (hlavne Internet Explorer), ktoré postupovali rovnako. A bol to IE, ktorý preto, že bol zadarmo, zakrátko predbehol Netscape. Výsledkom bola situácia, keď sa v rôznych prehliadačoch zobrazovali tie isté stránky rôzne. Preto vznikla nutnosť štandardizovať jazyk HTML. Prvá štandardizovaná verzia bola HTML verzia 2, čo malo odzrkadľovať to, že jazyk, čo sa používal pred tým, nebol štandardom. Spoločnosti vyrábajúce prehliadače sa dohodli, že budú štandard dodržiavať. O vývoj jazyka sa stará World Wide Web Consortium (W3C, www.w3c.org), založená Tom Berners-Lee, človekom, ktorý vynášiel HTML.

To bol pokrok ale napriek tomu prehliadače podporovali aj neštandardné (svoje vlastné) značky a navyše sa jazyk HTML začal čoraz viac používať aj na formátovanie dokumentu. To len zvyšovalo rozdiely v zobrazovaní jednotlivých prehliadačov. Preto bol vývoj jazyka HTML vo verzii 4.01 zastavený a pokračuje sa jazykom XHTML, ktorého najvyššia verzia je v súčasnosti 1.1.

Skutočnosť, že existuje viacero štandardov komplikuje situáciu tvorcom stránok ale aj výrobcom prehliadačov. Aby bolo možné určiť, podľa ktorého štandardu je stránka vytvorená, vkladá sa na začiatok stránky popis určujúci príslušný štandard – DTD (Document Type Declaration/Definition, tiež Doctype).

Ak prehliadač DOCTYPE nenájde (alebo je chybný), tak zobrazuje stránky v takzvanom quirk móde. To je spôsob, keď si prehliadač sám určí ako bude jednotlivé značky zobrazovať – dá sa povedať, že zobrazuje stránky ako za starých čias.

Ak je na stránke chyba, prehliadač sa ju snaží opraviť – napríklad, ak je na stránke iba začiatočná značka, prehliadač niekde doplní jej ukončenie. Rôzne prehliadače (prípadne ich rôzne verzie) môžu doplniť ukončujúcu značku na rôzne miesto, čo môže viesť k rôznemu zobrazeniu stránky na rôznych prehliadačoch.

Čím vyšší štandard je zadeklarovaný, tým je stránka interpretovaná presnejšie.

V súčasnosti je možné sa stretnúť s nasledujúcimi DTD.

HTML 4.01 DTD

DTD HTML 4.01 existuje v troch verziách strict, transitional a frameset.

HTML 4.01 strict nepovoľuje použiť formátovacie značky, namiesto toho vyžaduje použiť CSS. Definícia vyzerá nasledujúco:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
```

HTML 4.01 transitional povoľuje formátovacie značky a niektoré staré značky, ktoré sú už zakázané:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
```

HTML 4.01 frameset povoľuje použitie rámcov. Rámce sú spôsob, ktorý sa v minulosti s obľubou používal. Princíp spočíval v rozdelení okna na takzvané rámy (frames), čo boli obdĺžnikové oblasti. Napríklad na rám menu (vľavo), rám záhlavia, rám obsahu a rám pätičky. Každý rám bol tvorený html stránkou. Výhodou je, že po zobrazení stránky a klikaní na menu sa mení (a teda sťahuje) iba rám obsahu, čo znižuje prenosové nároky. Druhou výhodou je, že nie je nutné pri tvorbe stránok dávať menu, hlavičky a pätičky do každej stránky (menu, hlavička aj pätička ostávajú stále rovnaké), čo zjednodušuje údržbu stránok.

Nevýhodou je, že pri vyhľadávaní sa môže nájsť stránka, ktorá patrí do rámu obsahu a po jej zobrazení chýba menu, záhlavie aj pätička. To zabraňuje ďalšiemu prezeraniu firemného webu. Inou nevýhodou je napríklad nemožnosť jednoducho pridať si niektorú stránku medzi obľúbené položky alebo na plochu, pretože sa v riadku pre adresu stále zobrazuje rovnaká pôvodná stránka (mení sa iba obsah niektorého rámu).

Pre tieto nevýhody sa rámce dnes už prakticky nepoužívajú.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN"
"http://www.w3.org/TR/html4/frameset.dtd">
```

XHTML 1.0 DTDs

XHTML 1.0 DTD existujú rovnako v troch verziách strict, transitional a frameset.

XHTML 1.0 strict DTD je najstriktnejší DTD, nesmú byť použité žiadne zakázané značky a kód musí byť napísané bez chýb.

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

XHTML 1.0 transitional DTD je podobný svojmu striktnému súrodencovi ale sú povolené niektoré značky zakázané v XHTML.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

XHTML 1.0 frameset DTD je jediný XHTML DTD podporujúci rámce.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">
```

XHTML 1.1 DTD

Najaktuálnejší DTD, ktorý je založený na XHTML 1.0 Strict.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
```

Úplný výpis DTD je možné nájsť na stránkach www.w3.org. Pri XHTML DOCTYPE je nutné upraviť začiatočnú značku `html` nasledujúcim spôsobom.

```
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
```

Príklady v knihe sú vytvorené tak, aby spĺňali XHTML 1.1.

To či vaša stránka spĺňa štandard je možné otestovať na validator.w3.org.

4.4. Značky meta

Značky meta sa vyskytujú v časti *head*. Je ich viacero druhov a autor stránky si môže pridať vlastné. V tejto kapitole sú uvedené len tie najdôležitejšie a najčastejšie používané.

V *http-equiv* alebo *name* je názov vlastnosti a v *content* je hodnota alebo hodnoty vlastnosti.

Pretože sa pri ukladaní súborov používa mnoho kódovaní je nutné oznámiť prehliadaču aké kódovanie je použité. V prípade slovenčiny je často používané *windows-1250*. Inou častou možnosťou je *iso-8859-2*. Hodnota sa musí nastaviť podľa toho ako ukladá zdrojový kód Váš editor.

```
<meta http-equiv="Content-Type" content="text/html;
charset=windows-1250" />
```

Pre prehliadače môže byť dobrá informácia v akom jazyku je stránka. Túto informáciu môžu využívať aj vyhľadávače.

```
<meta http-equiv="content-language" content="Slovak" />
```

Veľmi dobré je umiestniť do meta značiek aj popis stránky a kľúčové slová vystihujúce obsah stránky. Môže to pomôcť pri „nájditeľnosti“ stránky inými.

```
<meta name="description" content="Stránky sú venované..." />
<meta name="keywords" content="Pár kľúčových slov oddelených
čiarkov" />
```

V metaznačke môže byť uvedený aj autor stránky:

```
<meta name="author" content="Webmaster webmasterovity" />
```

Roboti vyhľadávačov prechádzajú stránky ako pavúky. Využívajú pritom odkazy, ktoré na stránkach nájdu. Autor stránky im môže prehľadávanie umožniť alebo zakázať.

All znamená umožnenie indexovania stránky (stránka bude nájdiiteľná) a aj pokračovať v skúmaní stránok, na ktoré smerujú odkazy na danej stránke. *Noindex* neumožňuje indexovanie. *Nofollow* zakazuje pokračovať prehľadávaním odkazov na stránke. Analogicky fungujú *index* a *follow*. Je možné použiť viacero hodnôt oddelených čiarkou.

```
<meta name="robots" content="index, nofollow" />
```

Autor stránky si môže vytvoriť vlastné dvojice vlastnosť - hodnota. Napríklad:

```
<meta name="righttocopy" content="All" />
```


5. Kaskádové štýly

CSS (Cascading Style Sheets, kaskádové štýly) umožňujú efektívne formátovať webové stránky. Veľkou výhodou CSS je to, že neovplyvňuje obsah stránok. Pomocou CSS môžeme nastaviť formátovanie pre každý objekt na stránke zvlášť ale je možné nastaviť rovnaké formátovanie pre skupinu objektov alebo aj všetky objekty na stránke. Podobne ako pri stránkach, aj CSS je možné dať si skontrolovať. Jednou z možností je <http://jigsaw.w3.org/css-validator/>.

5.1. Pripojenie CSS v dokumentoch XHTML

Formátovacie predpisy CSS pre dokumenty XHTML sú uložené v samostatnom súbore a pripájajú sa k dokumentu v sekcii *head* pomocou značky *link*⁷⁹. Ak by sa súbor so štýlmi volal napríklad *mojestyly.css* a bol uložený v tom istom adresári ako formátovaný dokument, tak by pripojenie štýlov vyzeralo nasledujúco:

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

V prípade, že sa súbor so štýlmi nachádza v inom adresári ako formátovaný dokument je potrebné uviesť k nemu cestu. Viac je o problematike relatívnej a absolútnej cesty napísané v kapitole 4.2.4.

⁷⁹ Existujú ešte dva spôsoby použitia štýlov: priama definícia štýlov a vloženie tabuľky štýlov priamo do dokumentu (X)HTML v sekcii *head* pomocou značky `<style>`. Prvá možnosť je v XHTML 1.1 zakázaná (odporuje myšlienke kaskádových štýlov) a druhá sa neodporúča. Pre úplnosť uvedieme príklady použitia oboch možností. Priama definícia:

```
<p style="color:blue"> text tohto odseku bude modrý </p>
```

Definície vložené priamo v (X)HTML dokumente:

```
<html>
  <head>
    <style type="text/css">
      p:{color:blue}
      h1:{color:red}
    </style>
  </head>
  <body>
    <h1>Tento nadpis je písaný červenou farbou</h1>
    <p> text tohto odseku bude modrý </p>
  </body>
</html>
```

Výhodou CSS je aj to, že je možné nastaviť pre rôzne médiá rôzne spôsoby formátovania. Tabuľka 5.1 obsahuje zoznam všetkých možných cieľových zariadení, pre ktoré je možné definovať formátovanie. Nasledujúci príklad popisuje situáciu, keď je jeden štýl určený pre PC monitory, v prípade terminálu a vreckového počítača sa použije iný štýl a v prípade tlače sa použije tretí štýl.

Príklad použitia rôznych štýlov podľa média, pre ktoré sú určené

```
<head>
  <link href="mojestyly1.css" type="text/css" rel="stylesheet"
media="screen" />
  <link href="mojestyly2.css" type="text/css" rel="stylesheet"
media="handheld, tty" />
  <link href="mojestyly3.css" type="text/css" rel="stylesheet"
media="print" />
</head>
```

Tabuľka 5.1. Typy CSS médií

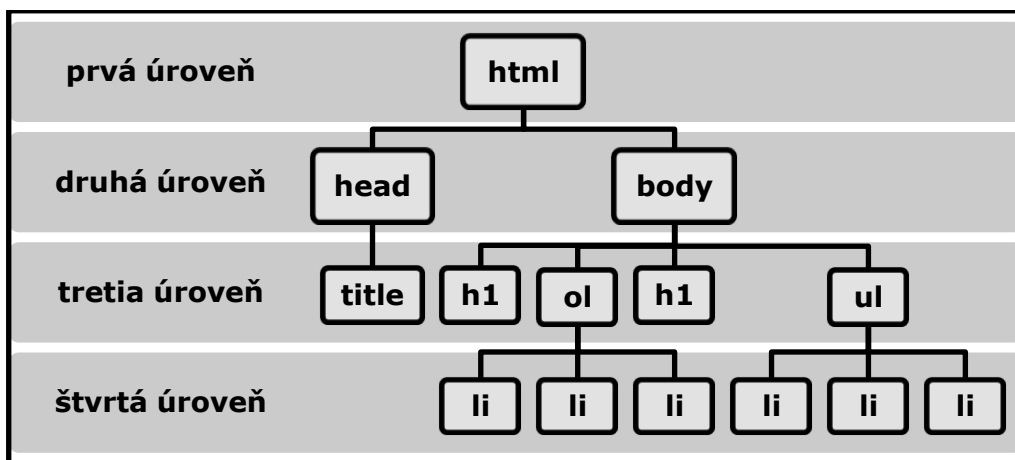
Medium	Význam
all	Všetky typy zariadení – v XHTML prednastavená hodnota.
aural	Zvukový výstup.
braille	Dotykové zariadenia.
embossed	Tlačiareň s 3D tlačou – braillova tlačiareň.
handheld	Vreckový počítač – menšia obrazovka, pomalší prenos dát.
print	Tlačiareň.
projection	Projektory, prezentácie.
screen	PC monitor.
tty	Monitor s neproporcionálnym písmom, terminál.
tv	Televízna obrazovka, nižšie rozlíšenie ako monitor

5.2. Štruktúra dokumentu a dedičnosť

Značkovacie jazyky (X)HTML pracujú so štruktúrou dokumentu, ktorú popisujú pomocou stromovej štruktúry. Pomocou vnorených značiek (napríklad značka *title* je vnorená v značke *head*) vytvárajú v dokumente celky, medzi ktorými sú určité vzťahy. Na popisovanie týchto vzťahov sa používa genealogická terminológia (potomok, súrodenec, rodič a pod.). Ako ukážku zoberieme stránku z príkladu č. 14 na strane 133. Jej stromová štruktúra je znázornená na obrázku 5.1.

Každý dokument začína z jediného (koreňového) prvku, ktorý obsahuje všetky ostatné prvky. V dokumentoch (X)HTML je to značka *html*. Prvok obsahujúci aspoň jeden ďalší prvok sa nazýva rodič alebo rodičovský prvok a prvky priamo v ňom vnorené sa označujú ako potomkovia. Potomkovia potomkov sa vzhľadom k pôvodnému rodičovi nazývajú následníci a pôvodný rodič sa volá predok.

Na obrázku má prvok *body* niekoľko potomkov: *h1*, *ol*, *h1* a *ul*. Tieto prvky sú navzájom súrodencami. Prvok *li* (ľubovoľný) je následníkom prvku *body*. Prvok *ul* má predchádzajúceho súrodencu prvok *h1* (iba ten, čo je na obrázku vedľa neho). Prvok je nasledujúcim súrodencom prvku *h1*.



Obr. 5.1 Štruktúra dokumentu z príkladu č. 14

Hierarchickým usporiadaním dokumentu sa uľahčuje okrem iného nastavovanie vlastností pre jednotlivé prvky. Niektoré vlastnosti CSS sú dedičné. To znamená, že ak nie sú definované prevezmú sa z najbližšieho nadradeného prvku. Postup, akým sa priradujú prvkom vlastnosti sa dá zhrnúť do troch bodov:

1. Ak je hodnota vlastnosti objektu explicitne určená, berie sa do úvahy táto hodnota.
2. Ak sa explicitne neurčí hodnota, potom sa skúma či sa hodnota nedá zdediť z rodičovských prvkov alebo z prvkov predkov. Rodičovský prvok má pri dedení prednosť pred ľubovoľným iným predkom, obdobne bližší predok má prednosť pred vzdialenejším. Ak v obrázku 2.1 zdefinujeme modrú farbu písma pre *body* a červenú pre *ol*, tak nadpisy prvej úrovne (*h1*) budú modré a takisto text v odrážkach nečíslovaného zoznamu (*ul* → *li*). Text v číslovanom zozname (*ol* → *li*) bude červený.

3. Ak hodnota nie je ani nastavená ani sa nedá zdediť, použije sa preddefinovaná hodnota. Preddefinovaná hodnota môže závisieť od použitého DTD, prehliadača alebo zariadenia na ktorom sa dokument zobrazuje.

Vo všeobecnosti sa dá povedať, že vlastnosti textu sú dedičné a ostatné vlastnosti sa nededia.

5.3. Definícia štýlov

Štýly sa definujú vo forme pravidiel. Pravidlo sa skladá z takzvaného selektora a deklarácie. Selektorom sa určuje, pre ktoré prvky sa použije formátovanie. Deklarácia popisuje samotné formátovanie a je umiestnená v zložených zátvorkách. Deklarácia sa skladá (podobne ako pri niektorých značkách HTML) z vlastnosti a jej hodnoty. Symbolický zápis vyzerá nasledujúco:

```
selektor {vlastnosť: hodnota;}
```

V konkrétnom prípade by sme mohli použiť (všetky nadpisy prvej úrovne budú modré):

```
h1 {color: blue;}
```

V rámci jednej deklarácie možno použiť viacero dvojíc „vlastnosť – hodnota“, ktoré sa od seba oddeľujú bodkočiarkou, a tak priradiť prvku naraz viacero formátovaní:

```
selektor {vlastnosť1:hodnota1; vlastnosť2: hodnota2; ...}
```

Napríklad (všetky odseky budú písané žltou farbou na modrom podklade):

```
p {color: yellow; background-color: blue;}
```

V prípade dlhších deklarácií sa používa zápis, v ktorom sa každá deklarácia uvedie na samostatnom riadku (význam je rovnaký ako v predchádzajúcom prípade + šírka odseku bude 150 bodov).

```
p {
color: yellow;
background-color: blue;
width: 150px;
}
```

V CSS sa podobne ako v (X)HTML môžu používať komentáre ale označujú sa ináč. Začiatok komentára sa označuje `/*` a koniec `*/`. V nasledujúcom príklade sa pre text v odseku nenastaví žltá farba.

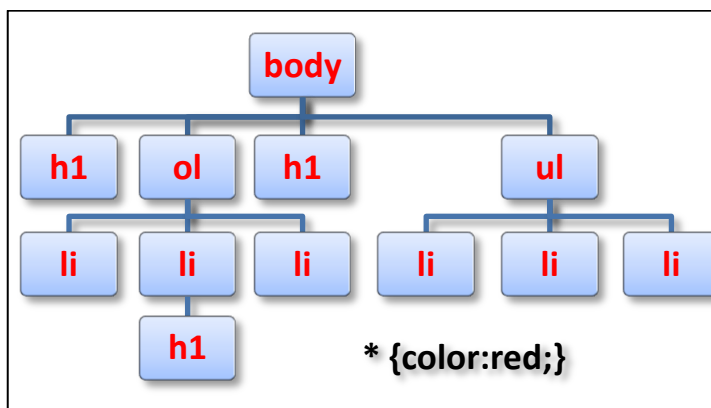
```
p {/* color: yellow; */ background-color: blue; }
```

5.3.1. Typy selektorov

CSS rozlišuje štyri typy selektorov. Prvý typ sme použili v už uvedených príkladoch. Je to **selektor typu**, ktorý využíva názov (X)HTML značky, na ktorú sa aplikuje formátovanie. V prípade `h1 {color: blue;}` sme použili ako selektor značku *h1*, preto budú všetky nadpisy prvej úrovne modré.

Tento typ selektorov je možné rozšíriť v závislosti od vlastnosti, ktorá je použitá v značke v (X)HTML dokumente. Napríklad: `table[border] {color: black;}` znamená, že sa formát použije iba na tabuľky, ktoré majú definovaný atribút *border*.

Druhým typom je **univerzálny selektor** - symbol *, ktorý znamená, že dané formátovanie bude použité na všetky prvky dokumentu. Príklad: `* {color: red;}`.



Obr. 5.2. Použitie univerzálného selektora

Tretím typom je výber prvku podľa mena – *id*, **selektor id**, tiež sa nazýva **selektor mena**. V súbore CSS sa definuje pomocou znaku mriežka - #. Napríklad:

```
#hlavnyodsek {color: green;}
```

Na rozdiel od selektora typu, ktorý platí pre všetky prvky danej značky, *id* (v tomto prípade *#hlavnyodsek*) môže byť na stránke (X)HTML iba jedenkrát. Na stránke sa používa nasledujúcim spôsobom:

```
<p id= "hlavnyodsek"> text odseku bude zelený</p>
```

Prvok s *id* je možné využiť na stránke v značke *a* ako keby bol definovaný pomocou ``. Napríklad: ``.

Štvrtým typom selektoru je **selektor triedy**. V dokumentoch (X)HTML sa tento typ používa asi najčastejšie. Umožňuje si zdefinovať triedu (class, skupinu prvkov), ktorej sa

následne priradí formátovacie pravidlo. Selektor triedy začína znakom bodka. V súbore kaskádových štýlov môže definícia triedy vyzerat' napríklad takto:

```
.vyrazne {color: red;}
```

V stránke sa trieda definuje pomocou slova *class*. Bodka sa už nepíše. Napríklad: `<p class="vyrazne">tento text je písaný výrazne</p>`. Na jednej stránke sa môže nachádzať viacero rôznych tried a triedy sa môžu opakovať. Napríklad na stránke s už uvedeným odsekom (*p class="vyrazne"*) môže byť aj:

```
<li class="vyrazne">aj táto odrážka je písaná výrazne</li>
```

Na tomto príklade vidno krásu a silu CSS. Ak niekto povie, že zvýraznenie nemá byť červené ale modré, stačí zmeniť definíciu:

`.vyrazne {color: red;}` na `.vyrazne {color: blue;}` a samotných stránok sa netreba ani dotknúť – všetky zvýraznenia budú modré.

Pomocou zápisu `p.vyraznyodsek {color: yellow}` je možné vytvoriť triedu použiteľnú iba v odsekoch (*p*).

V stránke možno použiť naraz priradenie viacerých tried, napríklad:

```
<p class="vyrazne vpravo">text je písaný výrazne a vpravo</p>
```

V príklade sa predpokladá, že existujú dve definície tried *vyrazne* a *vpravo*.

Zostáva už iba povedať, čo sa stane ak je jednému prvku priradených viacero protirečivých formátov. Napríklad (príklad by platil aj keby sme použili namiesto selektora *p* selektor *triedy* alebo *id*):

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

V danom prípade sa použije pravidlo, že platí najpresnejšie určenie prvku. Keďže selektor typu (*p*) je presnejší ako univerzálny selektor, odseky budú modré.

Ak by k dvom uvedeným pribudlo pravidlo `#mojodsek {color: green}` a v stránke by sa vyskytol zápis `<p id="mojodsek"> text text text </p>`, bol by tento odsek zelený, pretože *id* je jedinečné a tým najpresnejšie určenie.

V prípade, že sú si selektory rovnocenné, použije sa posledný. Napríklad:

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

V tomto prípade bude odsek písaný čiernou farbou.

5.3.2. Kombinovanie selektorov

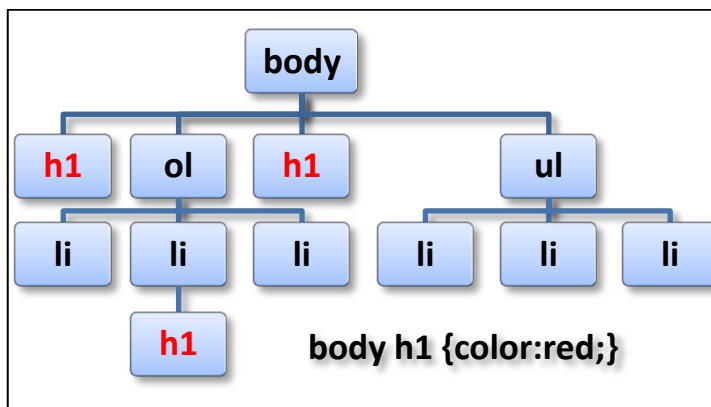
Najjednoduchším spôsobom je **spoločná definícia** pre viac selektorov. Selektory sú navzájom oddelené čiarkou. Napríklad nadpisy úrovne jeden až tri majú byť zobrazené modrou farbou: `h1, h2, h3 {color: blue;}`.

Často je vhodné naformátovať iba prvky, ktoré sú vnorené do nejakého nadradeného prvku. Napríklad v súbore strana14.html z príkladu č. 14 (hierarchická štruktúra je znázornená na strane 145) naformátovať odrážky iba číslovaného zoznamu na modro. V prípade použitia selektora typu: `li {color: blue;}` by sa vybrali a naformátovali aj odrážky nečíslovaného zoznamu. Preto sa zaviedli takzvané **následnícke selektory**. Pri ich definovaní sa používa **operátor medzera**. V našom prípade by zápis vyzeral nasledujúco:

`ul li {border:1px solid black;}` a číta sa: vyber všetky prvky *li* nachádzajúce sa niekde vo vnútri *ul* a nastav im rámček (presnejšie je o vlastnosti *border* napísané v kapitole 5.12). Asi je dobré zdôrazniť, že výber (a následné formátovanie) sa týka iba prvku *li*. Je možné vytvárať aj zložitejšie konštrukcie, napríklad: `td li h1{color: red;}`. V tomto príklade sa vyberie a následne naformátuje prvok každý *h1*, ležiaci niekde v prvku *li*, ktorý sa nachádza niekde v bunke tabuľky.

Týmto spôsobom je možné kombinovať aj selektory *id* a *tried*, napríklad: `ol .vyrazne {color: red;}`. Tento prípad treba odlišovať od

`ol.vyrazne {color: red;}` (bez medzery).

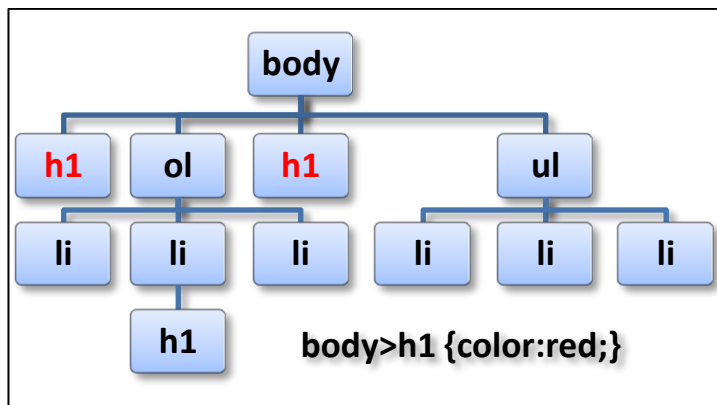


Obr. 5.3. Použitie operátora medzera

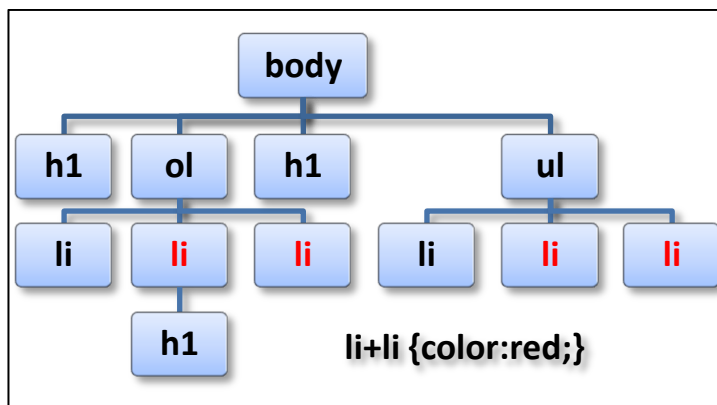
V prvom prípade sa vyberú všetky triedy *.vyrazne*, nachádzajúce sa niekde v *ol* a v druhom sa vyberie `<ol class="vyrazne">`.

Podobné následníckym selektorom sú **selektory potomkov**. Zapisujú sa pomocou **operátora väčší**. Príklad: `ul>li{border:1px solid black;}`. Výber a formátovanie

prvku *li* sa uskutoční, iba ak je *li* potomkom (priamym následníkom) prvku *ul*. Ak by nečíslovaný záznam obsahoval číslovaný záznam (tak ako je to v príklade v kapitole 4.2.6), tak v prípade selektora `>` by sa naformátovali len prvky *li* nečíslovaného zoznamu. Použitím následníckeho selektora (medzera) by sa naformátovali aj vnorené prvky *li* patriace číslovanému zoznamu.



Obr. 5.4. Použitie operátora `>`



Obr. 5.5. Použitie operátora `+`

Zaujímavou možnosťou je vybrať prvok v prípade, že jeho predchádzajúcim súrodencom je nejaký iný konkrétny prvok. Napríklad `table + p {color:red;}` vyberie a naformátuje všetky odseky, ktoré sú nasledujúcim súrodencom tabuľky (nie celkom presne povedané: každý odsek ležiaci priamo pod tabuľkou). Iným príkladom je naformátovanie všetkých odrážok v každom zozname okrem prvej odrážky:

`li + li {border-top: 1px solid black; width:50px;}`, všetky odrážky okrem prvých, budú mať nad sebou tenkú čiaru dlhú 50 pixlov.

5.3.3. Značky `div` a `span`

(X)HTML značky `<div> </div>` a ` ` majú zvláštne postavenie. Používajú sa iba spolu so štýlmi.

`Span` je takzvaný riadkový (inline) prvok. Bez štýlov je jeho použitie zbytočné, v kóde `<p> mama má Emu </p>` sa nijako vizuálne neprejaví. `Span` sa spoločne so štýlmi používa na úpravu textu v riadku. Napríklad:

```
<p> mama <span class="vyrazne"> má </span> Emu </p>
```

V prípade blokového prvku `div` sa obdobné použitie:

```
<p> mama <div> má </div> Emu </p>
```

prejaví „rozlomením“ riadku na tri časti, pretože slovo `má` obalí do obdĺžnikového bloku, ktorý pred sebou a za sebou ukončuje riadky. `Div` sa spoločne so štýlmi používa na úpravu blokov a prípadné zoskupovanie blokov a textov na stránke. Príklad použitia:

```
<html><head><title>Moja 1. css stránka</title></head>
<body>
  <div class="uzke vpravo">
    <p> ... text .. </p>
    <img src= .... />
    <a href ...> ... </a>
  </div>
</body>
</html>
```

5.4. Jednotky v CSS

Ešte pred samotným popisom jednotiek je vhodné poznamenať, že veľkosť prvku sa určuje tak ako každá iná hodnota nejakej vlastnosti. Postup pozostáva z rovnakých krokov ako sú uvedené v predchádzajúcej kapitole 5.2. Štruktúra dokumentu a dedičnosť.

Veľkosť objektov je možné v CSS vyjadrovať v rôznych jednotkách. Jednotky, v ktorých sa udávajú rozmery objektov v CSS sa dajú rozdeliť do dvoch skupín: relatívne a absolútne. Medzi relatívne patria *em*, *ex*, percentá a *px*. Absolútne sú palec (*in*, *inch*), *cm*, *mm*, *pica*, *pt*. Tabuľka 5.2 uvádza všetky jednotky so stručným popisom.

Absolútne jednotky by sa mali používať hlavne pri tlači. Nastavenie veľkosti písma, napríklad na 2 cm, vyznie inak na 22 palcovom monitore a inak malom displeji mobilného telefónu.

Obrazový bod *px* sa často považuje za absolútnu jednotku ale to nie je pravda, pretože závisí od rozlíšenia a typu zobrazovacieho zariadenia. V rámci jedného zobrazovacieho zariadenia (štandardne monitora) je ho možné považovať za jednotku absolútnu.

1em zodpovedá veľkosti písma bez úpravy. To ale neznamená, že jednotku *em* je možné použiť iba na určenie veľkosti písma. Pomocou *em* je možné nastaviť výšku ľubovoľného prvku na stránke. Dá sa použiť aj desatinné vyjadrenie, napríklad *1.5em*, pričom sa používa desatinná bodka a nie čiarka.

Veľkosť *1ex* zodpovedá výške malého písmena *x*. Prehliadače ale aproximujú túto hodnotu hodnotou *0.5em*.

Percento sa vždy musí viazať k nejakej hodnote. Pri každej vlastnosti, ktorá pripúšťa použitie percent, je štandarde CSS uvedené k čomu sa percentá vzťahujú.

Tabuľka 5.2. Jednotky používané v CSS

Jednotka	Význam
em	1em = veľkosť písma v danom prvku
ex	výška písmena „x“
%	percentá
px	pixel (bod na obrazovke, tv, tlačiarňi a pod.)
in	palec, (inch = 2,54 cm)
cm	centimeter
mm	milimeter
pt	bod (1 pt = 1/72 inch)
pc	pica (1 pc = 12 point)

5.5. Farby v CSS

Pre používanie farieb na webových stránkach sa používa farebný model RGB. Tento model je definovaný tromi základnými farbami – červenou (Red), zelenou (Green) a modrou (Blue). Je to prirodzené, pretože už samotný pixel na obrazovke je vytvorený týmito jednotlivými farbami a väčšina stránok sa prezerá na nejakej obrazovke. V CSS sa farby zapisujú pomocou kľúčových slov alebo číselného zadania. Tabuľka 5.3 popisuje všetky možnosti.

Pri farbách, ktorých jednotlivé zložky sa skladajú z dvoch rovnakých číslíc, sa môžeme stretnúť so skráteným zápisom `#rgb` namiesto `#rrggbb`. Napríklad farba `#2299cc` sa zapíše `#29c`.

Jednotlivé prehliadače môžu farby prispôsobiť možnostiam zariadenia, preto treba mať na pamäti, že farby sa nemusia zobrazit' presne tak, ako sú zadané. Istú mieru farebnej

presnosti poskytujú takzvané bezpečné farby pre web (web safe colors), ktoré sú podmnožinou farebného priestoru RGB ale obsahujú len farby, ktorých zložky sú tvorené len pomocou hexadecimálnych čísel 00, 33, 66, 99, cc, ff. Táto podmnožina tvorí 216 farieb, ktoré by mali prehliadače prednostne zobrazovať čo najpresnejšie. Ale ani tak nie je zaručené verné podanie farby, pretože používatelia môžu mať svoje monitory nastavené rôzne.

Tabuľka 5.3. Zápis farieb v CSS

Zápis	Význam
#rrggbb	Hexadecimálne číslo R, G, B – každé v rozsahu 00-ff
ColorName	Anglický názov farby (red, green, blue, magenta, ...)
rgb(r, g, b)	Číselné hodnoty jednotlivých farieb RGB 0-255
rgb(r%, g%, b%)	Percentuálne hodnoty jednotlivých farieb RGB 0-100%

5.6. Vlastnosti popredia a pozadia prvku

V tejto kapitole je popísané ako nastaviť farbu popredia a pozadia ľubovoľného prvku na stránke a tiež ako nastaviť obrázok na pozadie. Ukážeme si, aké máme možnosti pri umiestňovaní obrázka na pozadí. V kapitole sú rozoberané nasledujúce vlastnosti:

- color
- background-color
- background-image
- background-repeat
- background-attachment
- background-position
- background

Farba popredia - color

Vlastnosť *color* popisuje farbu popredia prvku. Na stránkach je to farba písma. Táto vlastnosť už bola viackrát použitá v príkladoch v predchádzajúcich kapitolách.

Táto vlastnosť sa dedí.

Ak napríklad chceme, aby všetky nadpisy prvej úrovne boli modré, použijeme nasledujúce pravidlo:

```
h1 {color: #0000ff;}
```

Namiesto zápisu #0000ff je možné použiť aj názov farby (*blue*). Viac je o používaní farieb uvedené v kapitole 5.5. Farby v CSS.

Farba pozadia - background-color

Vlastnosťou *background-color* je možné, podobným spôsobom ako vlastnosťou *color*, nastaviť farbu pozadia prvku.

Hodnota tejto vlastnosti sa nededí a prednastavená hodnota je *transparent* – priesvitnosť. Okrem tejto hodnoty je možné použiť farbu.

Príklad použitia (farba pozadia odseku bude zelená):

```
p {color: green;}
```

Obrázok na pozadí - background-image

Na umiestnenie obrázka na pozadie slúži vlastnosť *background-image*.

Táto vlastnosť sa nededí a prednastavená hodnota je *none*.

V príklade je na pozadie číslovaného zoznamu umiestnený obrázok *pes.jpg*. Obrázok sa v príklade nachádza v tom istom adresári ako stránka. V prípade iného umiestnenia je potrebné použiť relatívnu alebo absolútnu cestu, tak ako bola popísaná v kapitole 4.2.4. Cesta k obrázku a jeho názov sú zapísané pomocou kľúčového slova *url* a zátvoriek.

```
ol {background-image: url("pes.jpg");}
```

Opakovanie obrázka na pozadí - background-repeat

Opakovanie obrázka na stránke je možné ovplyvniť pomocou vlastnosti *background-repeat*. Táto vlastnosť môže nadobúdať hodnoty: *repeat* (opakovať vo všetkých smeroch, prednastavená hodnota), *no-repeat* (neopakovať), *repeat-x*, *repeat-y* (opakovať iba vodorovne/zvisle).

V nasledujúcom príklade sa obrázok opakuje iba vo vodorovnom smere, tam kde nie je obrázok je farba pozadia modrá.

```
body {
  background-color: blue;
  background-image: url("pes.jpg");
  background-repeat: repeat-x;
}
```

Ukotvenie obrázka na pozadí - background-attachment

Vlastnosť *background-attachment* určuje či je obrázok na pozadí pevne na svojej pozícii alebo sa pohybuje spolu s ostatným obsahom. Táto vlastnosť môže nadobúdať dve hodnoty: *scroll* (prednastavená hodnota, pozadie sa pohybuje spolu s ostatným obsahom), *fixed* (pozadie sa nepohybuje).

Táto vlastnosť sa nededí.

V nasledujúcom príklade je ukázané použitie nepohybujúceho sa pozadia.

```
body {
    background-image: url("pes.jpg");
    background-repeat: repeat-x;
    background-attachment: fixed;
}
```

Umiestnenie obrázka na pozadí - background-position

Vlastnosť *background-position* definuje pozíciu obrázka na pozadí. Bez uvedenia tejto vlastnosti je obrázok umiestnený do ľavého horného rohu.

Je viacero ciest ako nastaviť hodnoty pre tento atribút. Všetky ale vychádzajú zo súradnicovej sústavy, ktorá má počiatok (0, 0) v ľavom hornom rohu. Tento atribút má dve hodnoty – súradnicu *x* a súradnicu *y*. Súradnice sa oddeľujú medzerou. Napríklad:

background-position: 100px 50px; znamená, že obrázok pozadia bude umiestnený 100 bodov vpravo a 50 bodov dole od ľavého horného bodu okna.

Súradnice môžu byť zadávané aj pomocou percent. Percentá sa vzťahujú na oblasť, ktorú zaberá samotný prvok, pre ktorý nastavujeme pozíciu pozadia. Pozícia 0% 0% je totožná s ľavým horným rohom a znamená pozícia 100% 0% znamená pravý horný roh. Taktiež je možné použiť kľúčové slová: *top*, *bottom*, *center*, *left* and *right*. Napríklad: *top right* znamená pravý horný roh.

Spoločná vlastnosť pozadia - background

Spoločná vlastnosť *background* je skratka pre nastavenie všetkých vlastností pozadia v jednej definícii. Napríklad pravidlo:

```
background: blue url("pes.jpg") no-repeat fixed right bottom;
```

nastaví farbu pozadia na modro a obrázok *pes.jpg* umiestni bez opakovania fixne do pravého dolného rohu.

Ak niektorá hodnota chýba, tak sa použije na prednastavená hodnota, napríklad:

```
background: blue url("pes.jpg") no-repeat;
```

umiestni obrázok do ľavého horného rohu a vlastnosti *background-attachment* nastaví mu hodnotu *scroll*.

5.7. Vlastnosti písma

V tejto kapitole sú popísané možnosti štýlov pri formátovaní písma. V kapitole sú popísané nasledujúce vlastnosti:

- font-family
- font-style

- font-variant
- font-weight
- font-size
- font

Typ písma - font-family

Vlastnosť *font-family* definuje zoznam fontov, oddelených čiarkami, pre daný prvok. Fonty sa snažia aplikovať postupne zľava doprava. Ak nie je font nainštalovaný na počítači (zariadení), tak sa skúsi ďalší - alternatívny. Ako prvé by sa mali uvádzať konkrétne názvy fontov a na konci by mal byť všeobecný názov pre celú rodinu písiem (*serif*, *sans-serif*, *mono-space*, *fantasy*, *cursive*). To zabezpečí, že aj keď cieľový počítač nepozná požadované typy písma – zobrazí aspoň písmo, ktoré sa typovo zhoduje s požadovaným. Ak sa názov skladá z viacerých slov je ho potrebné dať do úvodzoviek.

Použitie je znázornené na nasledujúcom príklade:

```
h1 {font-family:"Arial CE", "Helvetica CE", Arial, sans-serif;}
h2 {font-family: "Times CE", "Times New Roman", serif;}
```

Prednastavená hodnota závisí od prehliadača. Vlastnosť sa dedí.

Štýl písma - font-style

Atribút *font-style* umožňuje vybrať z nasledujúcich štýlov písma: *normal*, *italic* alebo *oblique*. *Italic* a *oblique* bývajú často nahradzované rovnakým štýlom.

Prednastavenou hodnotou je *normal* a vlastnosť sa dedí.

```
h1 {font-style: italic;}
```

Variant písma - font-variant

Zmenu medzi normálnym písmom a kapitálkami (malé znaky sú nahradené zmenšenými veľkými písmenami) umožňuje vlastnosť *font-variant*. Možné hodnoty sú *normal* alebo *small-caps*.

Ak nie sú kapitálky k dispozícii, môže ich nahradiť prehliadač veľkými písmenami. Použitie znázorňuje nasledujúci príklad:

```
h1 {font-variant: small-caps;}
```

Predefinovanou hodnotou je *normal* a vlastnosť sa dedí.

Hrúbka písma - font-weight

Hrúbku písma popisuje vlastnosť *font-weight*. Vlastnosť môže nadobúdať hodnoty *normal* a *bold*. Okrem toho možno použiť hodnoty *bolder*, resp. *lighter*, ktoré použijú najbliž-

šie tučnejšie alebo štíhlejšie písmo. Hodnotu je možné zadať aj číslami 100, 200, 300, ..., 900. Príklad použitia:

```
p { font-weight: bold; }
td { font-weight: 900; }
```

Prednastavenou hodnotou je *normal*. Hodnota sa dedí.

Veľkosť písma - font-size

Veľkosť písma určuje vlastnosť *font-size*. Veľkosť písma je možné určovať v rôznych jednotkách – ich prehľad je uvedený v kapitole 5.4. Jednotky v CSS. Príklad použitia:

```
h1 { font-size: 27px; }
li { font-size: 120%; }
p { font-size: 1.1em; }
```

Relatívne jednotky *em*, *ex* a *percentá* sa vzťahujú na veľkosť písma rodičovského prvku. Okrem toho je možné použiť aj slová: *xx-small*, *x-small*, *small*, *medium*, *large*, *x-large*, *xx-large*, *smaller*, *larger*.

Tí, ktorí budú pozerat' vašu stránku môžu mať problémy so zrakom alebo môžu zobrazovať stránky na staršom monitore alebo naopak na mobilnom zariadení, ktoré má menšie rozmery. Nastavením veľkosti písma v absolútnych jednotkách sa im skomplikuje prezeranie stránok. Preto sa odporúča nastavovať veľkosť písma v *em* alebo percentách

Prednastavenou hodnotou je *medium*. Hodnota sa dedí.

Spoločná vlastnosť písma - font

Podobne ako pri pozadí existuje aj pri písme spoločná vlastnosť. Vlastnosť *font* umožňuje nastaviť vlastnosti na jedenkrát

```
div { font: italic bold 30px arial, sans-serif; }
```

5.8. Vlastnosti textu

Formátovanie textu patrí ku kľúčovým nástrojom pri navrhovaní každého dokumentu. V tejto časti sú popísané nasledujúce vlastnosti CSS použiteľné pri formátovaní textu:

- *text-indent*
- *text-align*
- *text-decoration*
- *letter-spacing*
- *text-transform*

Odsadenie prvého riadka odseku - text-indent

Odsadenie prvého riadka sprehľadňuje text, a tak zlepšuje jeho čitateľnosť. V prípade, že sa veľkosť odsadenia udáva v percentách, vzťahuje sa hodnota k šírke nadradeného prvku. Príklad použitia:

```
p {text-indent: -1.5em;}
```

Prednastavená hodnota je 0 a hodnota sa dedí.

Zarovnanie textu - text-align

Vlastnosť text-align určuje ako bude zarovnaný text. Možné hodnoty sú *left*, *right*, *center* a *justify* (do bloku).

V príklade je text odseku zarovnaný doprava:

```
p {text-align: right;}
```

Prednastavená hodnota závisí od prehliadača a toku textu. Hodnota sa dedí.

Dekorácia textu - text-decoration

Vlastnosť *text-decoration* umožňuje pridať do textu rôzne efekty. Možné hodnoty sú *none*, *underline* (podčiarknutie), *overline* (čiara na textom), *line-through* (prečiarknutie) alebo *blink*. Vlastnosť môže mať zároveň viacero hodnôt, použitie je znázornené na príklade:

```
h1 { text-decoration: underline overline line-through;}
```

Prednastavenou hodnotou je *none*. Hodnota sa dedí.

Vzdialenosť medzi znakmi - letter-spacing

Medzeru medzi znakmi je možno nastaviť pomocou vlastnosti *letter-spacing*. Ako hodnotu možno použiť slovo *normal* alebo číselnú hodnotu (nemožno použiť percentá).

```
p {letter-spacing: 3px;}
```

Prednastavenou hodnotou je *normal*. Hodnota sa dedí.

Vzdialenosť medzi slovami - word-spacing

Podobne ako predchádzajúca vlastnosť, vlastnosť *word-spacing* umožňuje nastaviť medzeru medzi slovami. Ako hodnotu možno použiť slovo *normal* alebo číselnú hodnotu (nemožno použiť percentá).

```
body {word-spacing: 30px;}
```

Prednastavenou hodnotou je *normal*. Hodnota sa dedí.

Zmena veľkosti znakov - text-transform

Vlastnosť text-transform riadi veľkosť písmen v texte. Možno zadať *none* (bezo zmeny), *capitalize* (prvý znak každého slova bude veľké písmeno), *uppercase* (všetko veľkými písmenami) alebo *lowercase* (všetko malými písmenami). Príklad použitia:

```
h1 { text-transform: uppercase; }
```

Prednastavenou hodnotou je *none*. Hodnota sa dedí.

5.9. Vlastnosti zoznamu

CSS umožňuje upravovať zoznamy pomocou nasledujúcich vlastností: *list-style-type*, *list-style-image*, *list-style-position* a spoločnej vlastnosti *list-style*.

list-style-type

Táto vlastnosť umožňuje definovať, aká odrážka sa má použiť. Jednotlivé odrážky sa dajú rozdeliť do troch skupín:

- Symboly - *disc*, *circle*, *square*.
- Čísla - *decimal*, *decimal-leading-zero*, *lower-roman*, *upper-roman*.
- Písmena - *lower-alpha*, *lower-greek* a mnoho ďalších abecied. Podobne existujú aj hodnoty pre veľké písmená (*upper-*).

Prednastavená hodnota je *disc*. Hodnota sa dedí.

list-style-image

Táto vlastnosť umožňuje používať ako odrážku ľubovoľný obrázok. Použitie:

```
ul {list-style-image: url ("../obrazky/mojaodrazka.gif") }
```

Namiesto odrážky sa použije obrázok *mojaodrazka.gif* zo susedného adresára.

Prednastavená hodnota je *none*. Hodnota sa dedí.

list-style-position

Táto vlastnosť definuje umiestnenie značky, povolené hodnoty sú *inside* a *outside*.

Prednastavená hodnota je *outside*. Hodnota sa dedí.

list-style

Spoločná vlastnosť list-style umožňuje nastaviť všetky vlastnosti v jednej definícii.

Použitie:

```
ul {list-style: circle url ("mojaodrazka.gif") inside;}
```

5.10. Pseudo-triedy

V predchádzajúcej kapitole je popísané ako možno jednotlivým prvkom zmeniť farbu, font a pod. V CSS je možné definovať formát odkazu nielen staticky ale aj dynamicky v závislosti od toho, či je odkaz navštívený, nenavštívený alebo či je nad ním kurzor. To umožňuje vylepšiť úpravu stránky. Vlastnosti, pomocou ktorých je to možné dosiahnuť sa nazývajú pseudo-triedy.

Definovanie pseudo-tried je podobné ako definovanie tried. Pri triedach sa používa bodka a zápis `a.vyrazne {color:red;}` znamená použitie triedy *vyraz*, ak sa nachádza v značke *a*. Pri pseudo-triedach sa používa namiesto znaku bodka znak dvojbodka a namiesto definovanej triedy (v predchádzajúcom príklade *vyrazne*) sa použije slovo *link*, *visited*, *active* alebo *hover*.

V nasledujúcom príklade sa nastaví farba nenavštíveného odkazu na modro a farba navštíveného na červeno. Ak je kurzor nad odkazom nastaví sa farba na čiernu a ak je odkaz aktívny na zeleno.

```
a:link {color: blue;}
a:visited {color: red;}
a:active {color: green;}
a:hover {color: black;}
```

Pretože odkaz môže byť naraz vo viacerých stavoch (napríklad navštívený, aktívny a môže byť nad ním kurzor) je dôležité definovať pravidlá v poradí od najvšeobecnejšieho po konkrétnejšie.

Ďalšie obľúbené triedy sú *:first-line* a *first-letter*. Používajú sa na formátovanie prvého riadku alebo prvého znaku. Použitie je znázornené na nasledujúcom príklade:

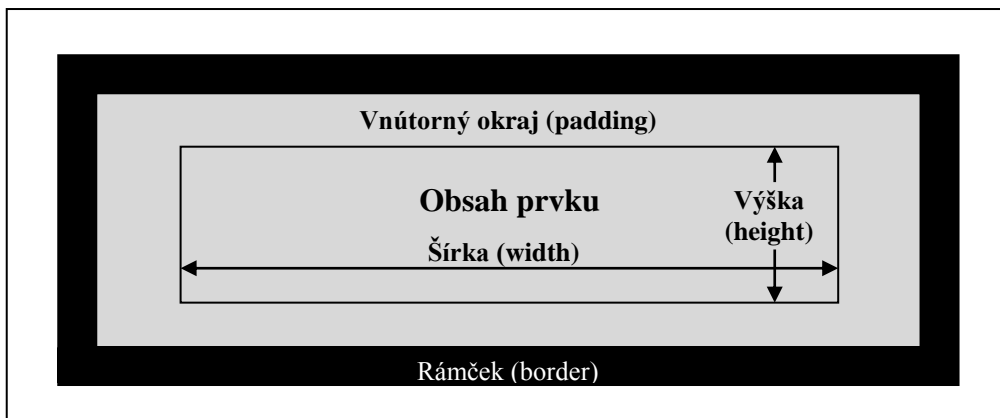
```
p:first-letter{font-size:3em}
p:first-line{font-variant:small-caps}
```

CSS umožňuje používať pseudo-triedy aj pre iné prvky. Bohužiaľ takýto postup podporujú len najnovšie prehliadače. Použitie môže vyzeráť tak, ako v nasledujúcom príklade (pri ukázaní kurzorom na odsek sa zmení farba písma na modrú):

```
p:hover {background-color: blue;}
```

5.11. Blokový model (Box model)

Blokový model v CSS popisuje bloky tak, ako ich generujú prvky na stránke. Tento model detailne popisuje jednotlivé časti bloku. Blok sa skladá z vonkajšieho okraja (*margin*), rámčeka (*border*), vnútorného okraja (*padding*) a samotného obsahu prvku. Na obrázku 5.6 je znázornený blok aj s vlastnosťami, ktoré je možné nastavovať pomocou CSS:



Obr. 5.6. Blokový model

Nastavenie vonkajších okrajov - margin

Prvok má štyri okraje: horný (*top*), pravý (*right*), spodný (*bottom*) a ľavý (*left*). Vonkajší okraj určuje vzdialenosť (z každej strany zvlášť) k susedným objektom na stránke. Ako príklad možno uviesť nastavenie vonkajších okrajov pre odsek. V nasledujúcom príklade sú nastavené rôzne vonkajšie okraje – vzdialenosti od okraja okna prehliadača. Zmenou jednotlivých hodnôt je možné zmeniť vzdialenosť odsekov od seba, prípadne od okrajov iných objektov. V prípade, že sa na stránke nachádzajú len odseky, možno pomocou vonkajších okrajov nastaviť vzdialenosť od okrajov okna prehliadača.

```
p {
margin-top: 100px;
margin-right: 0px;
margin-bottom: 20px;
margin-left: 50px;
}
```

CSS umožňuje nastavenie vonkajších okrajov jedinou definíciou:

```
p {
margin: 100px 0px 20px 50px;
}
```

Význam čísel postupne: *top*, *right*, *bottom*, *left*. Pri vonkajšom okraji môže byť hodnota aj záporná.

Nastavenie vnútorných okrajov - padding

Padding je vzdialenosť obsahu prvku od rámčeka (*border*). Zápis vlastnosti *padding* je podobný ako pri vlastnosti *margin*. Opäť možno použiť nastavenie pre každý okraj zvlášť

(*padding-top*, *padding-right*, *padding-bottom*, *padding-left*) alebo použiť spoločnú definíciu (*padding*).

Pri demonštrovaní vlastnosti *padding* využijeme vlastnosť *background-color*. Farba pozadia sa nastavuje pre časť obsahu bloku a vnútorných okrajov. Nasledujúci príklad predpokladá, že na stránke je nadpis prvej úrovne:

```
h1 {
background: green;
padding: 30px 40px 10px 50px;
}
```

Zmenou jednotlivých hodnôt je možné meniť vzdialenosť textu nadpisu od rámečka (ten nie je vidieť ale nachádza sa hneď za vnútornými okrajmi, t. j. tam kde končí zelená výplň).

Pre vlastnosť *margin* aj *padding* môžu byť použité skrátené zápisy:

margin: X = margin: X X X X

margin: X Y = margin: X Y X Y

margin: X Y Z = margin: X Y Z Y

Rovnako aj pre *padding*. Napríklad (horný okraj odseku bude 1em, pravý a ľavý budú 0,5 em a dolný okraj bude 10 px)

```
p {margin: 1em .5em 10px}
```

Prednastavená hodnota pre *margin* aj *padding* je 0. Vlastnosť sa nededí. Percentuálne hodnoty sa pre oba okraje vzťahujú k šírke objektu, v ktorom sa prvok nachádza.

Šírka a výška – **width, height**

Vlastnosťou *width* sa nastavuje šírka prvku. Výšku je možné nastaviť vlastnosťou *height*. Nasledujúci príklad nastavuje rozmery triedy *uzko*. Pre lepšie pozorovanie zmien pri modifikovaní hodnôt je v príklade pridaná vlastnosť farby pozadia.

```
.uzko {
height: 350px;
width: 150px;
background-color: green;
}
```

Prednastavená hodnota pre *height* aj *width* je *auto*, t. j. rozmery sa vypočítajú podľa obsahu prvku. Vlastnosť sa nededí. Percentuálne hodnoty sa pre *width* vzťahujú k šírke a pre *height* k výške objektu, v ktorom sa prvok nachádza.

V CSS sa dajú použiť aj vlastnosti *min-height*, *min-width* a *max-height*, *max-width*, ktoré nastavujú hraničné možnosti rozmerov. Maximálne vlastnosti majú prednastavenú hodnotu *none* (môže byť ľubovoľná veľkosť). Prednastavená hodnota pre minimálnu šírku závisí od prehliadača a pre minimálnu výšku je 0.

5.12. Rámček

Rámčeky môžu byť použité na ozdobu alebo na oddelenie dvoch susedných objektov. Rámčeku je možné nastaviť šírku (*border-width*), farbu (*border-color*), štýl (*border-style*).

Šírka rámčeka - *border-width*

Šírku rámčeka je možné zadať slovami (*thin*, *medium*, *thick*) alebo číslom. Šírku možno zadať pre každú stranu zvlášť (*border-top-width*, *border-right-width*, *border-bottom-width*, *border-left-width*) alebo spoločnou definíciou *border-width*. V prípade spoločnej definície je možné uviesť jeden až štyri argumenty, s analogickým významom ako je uvedené v predchádzajúcej kapitole pre vlastnosť *margin*. Príklad použitia (všetky odseky budú mať hore a dole tenký a po bokoch hrubý rámček)

```
p {border: thin thick}
```

Prednastavená hodnota pre *border* je *medium*. Vlastnosť sa nededí.

Farba rámčeka - *border-color*

Farbu rámčeka je možné nastaviť pre každú stranu rámčeka zvlášť (*border-top-color*, *border-right-color*, *border-bottom-color*, *border-left-color*) alebo spoločnou definíciou *border-color*. Okrem konkrétnej farby možno určiť, že rámček alebo jeho časť bude priehľadná (*transparent*).

Prednastavená hodnota je hodnota vlastnosti *color*. Vlastnosť sa nededí.

Štýl rámčeka - *border-style*

Aj tvar rámčeka je možné určiť pre každú stranu zvlášť (*border-top-style*, *border-right-style*, *border-bottom-style*, *border-left-style*) alebo pomocou spoločnej definície *border-style*. Hodnotou pre každú stranu rámčeka môže byť jeden z nasledujúcich štýlov: *none*, *hidden*, *solid*, *double*, *dashed*, *dotted*, *groove*, *ridge*, *inset*, *outset*.

Prednastavená hodnota je *none*. Vlastnosť sa nededí.

Združené vlastnosti

Rámčeky je možné definovať aj pomocou spoločných vlastností: *border-top*, *border-right*, *border-bottom*, *border-left*, a *border*. Hodnotami sú postupne šírka, farba, štýl. V nasledujúcom príklade sa pre odsek nastavi rámček široký 2px, *solid*, červenej farby a pre nadpis prvej úrovne sa nastavi tenká, modrá a bodkovaná horná strana rámčeka.

```
p {border: 2px solid red;}
h1 {border-top: thin dotted blue;}
```

5.13. Obtekanie

Vlastnosť *float* určuje či bude objekt obtekaný a ak áno, tak z ktorej strany. Povolené hodnoty pre túto vlastnosť sú *none* (nemožno obtekať), *left* (obtekanie sprava) a *right* (obtekanie zľava). Napríklad hodnota *left* posunie objekt k ľavému okraju objektu, v ktorom sa formátovaný prvok nachádza a nasledujúce objekty ho obtekajú sprava. Analogicky je to v prípade hodnoty *right*.

Použitie na stránke (dôležité je aby bol obtekaný objekt uvedený pred objektmi, ktoré ho majú obtekať):

```
<p class="vlavo"> Toto je text, ktorý je obtekaný nejakým iným
textom. </p>
<p>Lorem ipsum dolor sit amet consectetur lacinia hendrerit
ipsum vitae condimentum. Mauris at Vestibulum interdum rutrum
leo pharetra dignissim eros montes pede. Congue consequat lorem
Curabitur congue pede Nunc orci habitant Donec lacus. Mattis
leo convallis sollicitudin tincidunt morbi lacus auctor et
semper justo. Eu Vestibulum feugiat at in magnis neque nascetur
lacinia Sed facilisis. Et Donec ornare interdum Proin dui a
nisl convallis ac ut. Elit.</p>
```

Použitie v súbore kaskádových štýlov:

```
.vlavo {
float:left;
width: 100px;
}
```

Vlastnosť *float* má prednastavenú hodnotu *none* a nededí sa.

Vlastnosť *clear*

Vlastnosť *clear* ukončuje obtekanie. Môže nadobúdať vlastnosti *left*, *right*, *both* or *none*. Napríklad hodnota *left* ukončuje obtekanie prvkov s vlastnosťou *float* nastavenou na *left*. Analogicky je to aj pre *right* a *both* ruší obtekanie úplne.

Vlastnosť *clear* má prednastavenú hodnotu *none* a nededí sa.

5.14. Umiestňovanie prvkov na stránke

Pomocou CSS je možné umiestniť prvok na ľubovoľné miesto na stránke. Pri umiestňovaní máme na výber z niekoľkých režimov, ktoré sa určujú pomocou vlastnosti *position*. Spoločne s vlastnosťou *position* sa používajú **súradnice** - vlastnosti *top*, *right*, *left*, *bottom* a prípadne aj vlastnosť *z-index*. Umiestnenie prvku na stránke ovplyvňujú aj vlastnosti obtekania, ktoré sú popísané v predchádzajúcej kapitole.

Povolené hodnoty pre vlastnosť *position* sú:

- *static* - Objekt sa formátuje normálne. Súradnice *top*, *right*, *left*, *bottom* sa ignorujú.
- *relative* - Objekt sa najskôr naformátuje v režime *static* a potom sa posunie zo svojho miesta o hodnotu súradníc. Posúva sa teda vzhľadom na miesto, na ktoré by normálne patril.
- *absolute* - Prvok sa umiestni podľa súradníc, pričom sa súradnice vzťahujú k najbližšiemu nadradenému prvku. Súradnice 0 0 označujú ľavý horný roh.
- *fixed* - Prvok sa formátuje rovnako ako pri *absolute* ale sa navyše ešte aj upevní (zafixuje). Pri posúvaní stránky sa nepohybuje. Ostatné objekty sa pohybujú pod ním.

Vzhľadom na to, že vďaka vlastnosti *position* sa môžu na jednom mieste ocitnúť viaceré prvky, je nutné zabezpečiť ich poradie (ktorý z nich je na vrchu). To zabezpečuje vlastnosť *z-index*. Hodnotou je celé číslo alebo *auto* (je v rovnakej vrstve ako jeho rodičovský prvok). Čím vyššie číslo, tým vyššie sa prvok nachádza - prvok s vyššou hodnotou *z-index* prekryva prvok s nižšou hodnotou.

Príklad (posun smerom hore zo svojej regulárnej pozície, zmenšenie veľkosti):

```
span.mocnina{
position: relative;
font-size: 50%;
bottom: 0.5em;
}
```

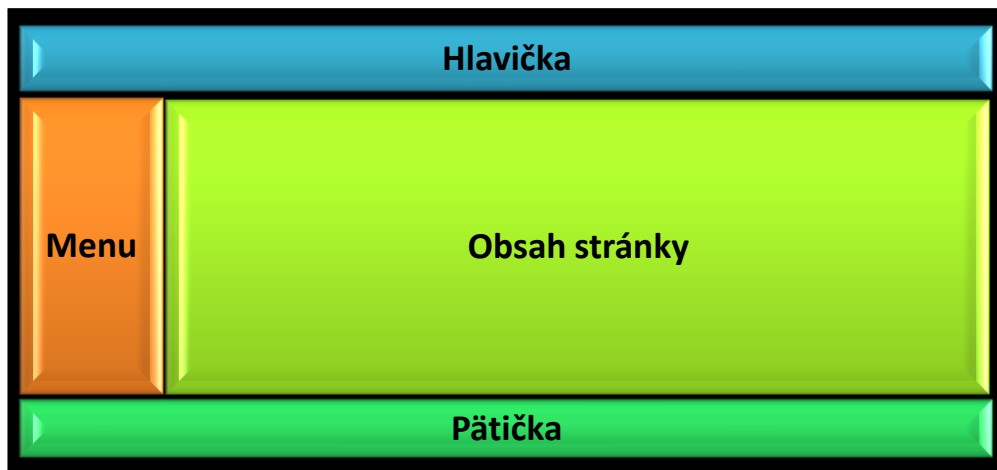
Vlastnosť *position* má prednastavenú hodnotu *static* a nededí sa.

Súradnice majú prednastavenú hodnotu *auto* (pozícia sa určuje automaticky), vlastnosť sa nededí. Percentuálne hodnoty sa viažu k šírke (*left*, *right*) a k výške (*top*, *bottom*) nadradeného objektu.

Vlastnosť *z-index* má prednastavenú hodnotu *auto* a nededí sa.

5.15. Rozvrhnutie stránky pomocou štýlov

V tejto kapitole je ukázané ako rozvrhnúť stránku pomocou štýlov. Na internete je množstvo stránok, ktoré sa zaoberajú vytvorením rozvrhnutia (layout) pre rôzne typy stránok. V príklade v tejto kapitole sa zoberáme stránkou so záhlavím, pätičkou, menu na ľavej strane a hlavnou časťou. Štruktúra takejto stránky je znázornená na obrázku 5.7.



Obr. 5.7. Rozvrhnutie (layout) stránky

Základný HTML kód takéhoto rozvrhnutia môže byť takýto:

```
<div id="stranka">
  <div id="hlavicka">Hlavička</div>
  <div id="menu">Menu</div>
  <div id="obsah">Obsah stránky</div>
  <div id="paticka">Pätička</div>
</div>
```

Každá zo štyroch častí stránky je zabalená do značky *div*. Tieto značky sú zabalené v značke s *id = stranka*, ktorú použijeme na vycentrovanie. Je to jednoduchá štruktúra a zatiaľ nič nenasvedčuje tomu, že na stránke budú dva stĺpce.

To ako bude vyzerat' každá z častí je samozrejme popísané v CSS. Jednotlivé časti sú popísané postupne.

Div stranka

Tento div v sebe zahŕňa všetky prvky na stránke. Nastavíme mu čierny okraj. Zarovnanie na stred možno docieľiť dvoma spôsobmi. Jedným je nastavenie šírky a zároveň nastaviť ľavý a pravý vonkajší okraj na hodnotu *auto*. Vyzeralo by to nasledujúco:

```
div#stranka {
  border:5px solid black;
  width:90%;
  margin-left: auto;
  margin-right: auto;
}
```


Druhý spôsob (nastavenie vonkajších okrajov na rovnakú veľkosť bez určenia šírky) je uvedený v nasledujúcom príklade:

```
div#stranka {
  border:5px solid black;
  margin-left: 5%;
  margin-right: 5%;
}
```

Div hlavička

Nastavenie hlavičky nie je ničím výnimočné:

```
div#hlavicka {
  border:5px outset blue;
  height:1.5em;
  background-color:blue;
}
```

Text „Hlavička“ by sa mal objaviť približne na tom mieste ako je nakreslený na obrázku na strane 165. Pretože sa text nezobrazuje v strede, pridáme ešte centrovanie textu a zároveň môžeme zvýšiť hrúbku písma. Formátovanie pre všetky prvky *div*:

```
div {
  text-align:center;
  font-weight:bold;
}
```

Div menu

Ak chceme docieľiť, aby dva objekty *div* boli vedľa seba môžeme použiť obtekanie jedného druhým alebo absolútne umiestnenie pomocou vlastnosti *position: absolute* a nastavením príslušnej pozície napríklad pomocou *top* a *left*. V príklade je častejšie využívaný spôsob a tým je obtekanie. Realizácia je znázornená na nasledujúcom príklade.:

```
div#menu {
  border: 5px outset maroon;
  width: 20%;
  float:left;
  margin:10px 0 10px 5px;
  height:350px;
  background-color: maroon;
}
```

Možno sa stretnúť aj s nastavením šírky v pixloch. Takto naformátovaná stránka má stále rovnakú šírku a na širších monitoroch zaberá iba časť obrazovky. V príklade bude zabe-

rať menu skoro 20 percent šírky monitora (je to 20% zo šírky nadradeného objektu a ten nezaberá celú šírku monitora).

Div obsah

Tak ako pri menu, aj pri formátovaní obsahu sú použité percentá. Všimnite si, že aj ľavý vonkajší okraj je nastavený pomocou percent.

```
div#obsah {  
  border:5px outset lime;  
  width:75%;  
  margin:10px 0 10px 20%;  
  min-height:350px;  
  background-color: lime;  
}
```

Div paticka

V pätičke je nutné urobiť iba jedinú vec - aby náhodou aj ona nezačala obtekať, je nutné nastaviť vlastnosť *clear* na *left* (alebo *both*). Nastavenie formátovanie pre pätičku je zachytené v nasledujúcom príklade:

```
div#paticka {  
  border:5px outset green;  
  height:1.5em;  
  clear:left;  
  background-color:green;  
}
```

5.16. Záver

Kaskádové štýly umožňujú používať viacero vlastností, ktoré nie sú popísané v tejto knihe. Jedným z najlepších zdrojov pre ďalšie štúdium (CSS aj XHTML) je stránka w3schools.com.

Pri skúšaní a experimentovaní so štýlmi je výhodné zobrazit' si stránku v rôznych prehliadačoch. Dobré je si zapamätať, že niektoré prehliadače zobrazia stránku podľa toho, či (a aký) má DOCTYPE. Po vyskúšaní si základných značiek je najlepšie snažiť vytvárať stránky s verziou XHTML 1.1 a hneď si ich aj otestovať validátormi HTML a CSS na stránke w3c.org.

Záver

Niet pochýb o tom, že súčasný fenomén Internetu mení životný štýl, prácu, spôsob vzdelávania a zábavy. A nie len to. Don Tapscott a Anthony Williams vo svojom diele, nazvanom Wikinomics, tvrdia, že Internet podmienil vznik nového druhu obchodovania, ktorý firmám otvára dvere doslova do celého sveta. Využíva silu masovej spolupráce a firme umožňuje správať sa skutočne ako globálna spoločnosť.

Internet sprístupňuje a evokuje nové modely spolupráce. Napríklad spoločnosť Procter and Gamble je presvedčená, že do roku 2010 polovicu nových výrobkov a služieb budú tvoriť používatelia prostredníctvom Internetu. Budú to oni, ktorí budú určovať napríklad, akú farbu budú mať prášky na pranie, či v akom balení sa budú predávať.

V prostredí Internetu vznikalo napríklad aj mamutie dopravné lietadlo Boeing 787. Výrobca prizval k spolupráci 100 dodávateľov zo šiestich krajín. Plány lietadla zverejnil na svojej Internetovej stránke, na ktorej sa mohli budúci potenciálni cestujúci vyjadriť, ako sa im páči. Celý dizajn vrátane návrhov a pripomienok budúcich používateľov vznikol prostredníctvom nástroja Global Collaborative Environment v prostredí Internetu.

Postupom času narastá uvedomovanie si firmi, že deliaca čiara medzi zábavou a biznisom na Internete nemá zmysel. Mnohé z firiem sa stávajú súčasťou aj virtuálnych svetov. Stačí si uvedomiť, že napríklad on-line digitálny svet Second Life mal v júni 2007 takmer sedem miliónov virtuálnych „obyvateľov“. Títo Avatarovia, za identitou ktorých sa skrývajú skutoční ľudia, utratili vo virtuálnom svete za 24 hodín cca 1,5 milióna reálnych dolárov. To je taký obrovský spotrebiteľský trh, ktorému nemôže konkurovať ani Čína.

S naberajúcim tempom Internetu sa pomerne rýchlo menia aj názory na jeho ďalší vývoj. V roku 2006 zástupcovia jednej z vedúcich firiem na trhu Internetu, firmy Cisco tvrdili, že Internet ovplyvňujú štyri megatrendy. Sú to digitalizácia, virtualizácia, mobilita a personalizácia. V roku 2007 zástupcovia tej istej firmy hovoria o konvergencii týchto trendov. Digitalizácia má na svedomí, že všetok obsah, ktorý existoval v analógovej podobe, ako text, video, obrázky či hudba, možno posielat' už po globálnej počítačovej sieti Internet. Digitálna forma obsahu je navyše mimoriadne efektívna, lebo obsah možno lacno a jednoducho vyrábať, modifikovať a distribuovať. Denne sa na Internete uskutoční 200 miliónov downloadov. Táto skutočnosť je veľmi významná.

Virtualizácia umožňuje preniesť fyzické veci na sieť. Napríklad japonská automobilka Toyota uviedla na trh auto, ktoré vznikalo na Internete. Jej model Avalon je prvým vozidlom na svete, ktoré sa začalo vyrábať bez prototypov. Všetky testy prebiehali virtuálne. Prinieslo to úžasnú úsporu nákladov a otvára sa priestor pre ďalšie, netušené možnosti.

Mobilita sprístupňuje obsah a komunikáciu z ktoréhokoľvek miesta na planéte Zem, v ktoromkoľvek čase. Rôzne produkty, rôznych firiem umožňujú v súčasnosti robiť cez mobil-

né zariadenie všetko čo za počítačom v kancelárii. Navyše je k dispozícii, instant messaging, videohovor, konferenčný hovor, zdieľanie dokumentov a elektronický podpis.

Z hľadiska obchodovania je mimoriadne dôležitá aj personalizácia, pretože „zákazníci sú rôzni“. Každý má iný vkus, iné prania. Internet poskytuje prostredie, ktoré môže týmto individuálnym praniam vyhovieť. Napríklad automobilka BMW poskytuje záujemcom o kultový model Mini Cooper možnosť, aby si poskladali prostredníctvom možností Internetu a firemných aplikačných programov auto podľa svojho „gusta“. Potom im toto auto za patričnú cenu vyrobí. Takto vyrobené auto zaručene nebude mať nikto iný.