

Čo je to WWW stránka

základné pojmy a technológie používané pre tvorbu WWW stránok

Každý podnikateľ, ktorý chce podnikat' v určitom odbore, sa stretáva s problémom, ako o svojej firme a o predmete svojho podnikania dať vedieť potenciálnym zákazníkom. Jedným zo súčasných moderných riešení, ako dať vedieť o svojej firme, je použitie Internetu, konkrétne webovej prezentácie (web stránky). Webová prezentácia tu môže plniť niekoľko úloh súčasne:

Úlohu prostriedku pre informácie o firme

Firma musí dať o sebe vedieť. Informácie, kde sídli, kde sa nachádza jej prevádzka (prevádzky), informácie o činnosti firmy, o jej hospodárení a úspechoch a o tom, na koho sa môže zákazník nakontaktovať, keď potrebuje viac informácií - to všetko sa môže nachádzať na web stránkach firmy.

Úlohu prostriedku pre vedenie reklamnej kampane (alebo súčasti reklamnej kampane)

Firma dáva o sebe vedieť prostredníctvom reklamy - či už v tlači alebo iných médiách. Tu však má čas na prezentáciu vymedzený svojimi prostriedkami. Preto je dobré ponúknuť potencionálnym záujemcom ďalšie, trvalé miesto, kde môžu hľadať podrobnejšie informácie. Web stránka je opäť výborným prostriedkom pre tieto účely.

Úlohu prostriedku pre vzťah zákazník - firma (komunikácia zo zákazníkom)

Komunikácia zo zákazníkom je dôležitá. Zákazník, ktorý sa o niečom potrebuje informovať, bude určite spokojný, pokiaľ bude reakcia na jeho otázky rýchla a vecná. Vtedy dokáže pritiahnuť ku firme aj ďalších zákazníkov už len tým, že bude vyjadrovať svoju spokojnosť s firmou medzi svojimi známymi. Prostredníctvom web stránky a komunikačných možností Internetu je tieto ciele možné dosiahnuť rýchlo a bez vynaloženia priveľkých finančných prostriedkov (ktoré si menšia firma väčšinou nemôže dovoliť).

Úlohu prostriedku pre priamy predaj tovaru zákazníkovi

Prevádzkovať „kamenný“ obchod je väčšinou veľmi drahé. Prenájom obchodných a skladových priestorov, elektrická energia, zamestnanci - to všetko stojí prostriedky. Prostredníctvom web stránky sa dá vybudovať e-obchod za podstatne nižšiu cenu a tento krok mnohokrát menšie, začínajúce firmy používajú ako prvý vstup do sveta obchodu. Platby prostredníctvom platobných brán overených bánk otvárajú túto cestu aj menším podnikateľom so zaujímavým podnikateľským zámerom, ktorí nemajú toľko prostriedkov pre vybudovanie „kamenného“ obchodu.

Dobrá a kvalitná web prezentácia musí zákazníka zaujať, musí udržať jeho pozornosť na web stránke a prinútiť ho, aby klikol na ďalšie odkazy, ktoré sú na web stránke prítomné.

Tu si musíme uvedomiť, že technológie, ktoré nám umožňujú prezentovať na webe informácie, zároveň môžu niekedy byť aj prekážkou. K web stránke pristupujeme prostredníctvom Internetu. Web stránka, ktorá je postavená s dôrazom na vzhľad (grafiku), sa bude nášmu potenciálnemu zákazníkovi s pomalým pripojením na Internet nahrávať neúmerne dlho, lebo jej obsah je príliš veľký a rozsiahly pre toto pomalé pripojenie. Náš potenciálny zákazník nemusí byť trpezlivý a radšej našu prezentáciu zruší a pôjde na inú web stránku. Tu musíme **vedieť robiť kompromisy. Úvodná stránka web prezentácie by mala obsahovať len najpodstatnejšie informácie, dôležité pre orientáciu návštevníka v obsahu web stránky.**

Takisto, keď používame technológie ako je napríklad FLASH, by sme si mali byť istí, že náš potenciálny zákazník uvidí na web stránke aspoň nejaké informácie, aj keď nemá k dispozícii prehrávač tohto formátu. Mnohé firmy z neskúsenosti robia chybu, že si nechajú urobiť web stránku „profesionálom“, ktorí síce vedia urobiť krásnu pohyblivú animovanú prezentáciu vo FLASHi, ale neuvedomujú si, že tým beznádejne stratia časť zákazníkov, ktorí jednoducho či už zo zdravotných (napríklad slabozrakí, či dokonca nevidomí) alebo technických (staršie prehliadače, či prehliadače so staršou verziou zásuvného modulu pre FLASH) príčin jednoducho tú krásnu FLASH prezentáciu neuvidia.

Test použiteľnosti webového sídla

Použiteľnosť je kvalitatívny atribút internetových stránok, ktorý určuje ako ľahko sa na nich používateľ orientuje, ako rýchlo pochopí ich usporiadanie a ovládanie a aký zážitok si z nich odnesie. Dobré použiteľné stránky sú prehľadné, zrozumiteľné a ľahko ovládateľné. Je radosť s nimi pracovať.

Webové stránky

K najdôležitejšími kvalitatívnym atribútom použiteľnosti webových sídiel, patria:

- nájditeľnosť
- prístupnosť
- kredibilita
- použiteľnosť

Nájditeľnosť webového sídla

Na nájditeľnosť stránky vplýva predovšetkým to, ako je indexovaná prieskumnými softvérmi, botmi. V prípade, že na stránke je príliš veľa grafiky, technológia flash, nepoužívajú sa popisy obrázkov a nerešpektujú sa štandardy pre tvorbu dokumentov v elektronickom prostredí, stránka je pre používateľa internetu stratená, pretože robot ju nedokáže zindexovať.

Väčšina používateľov prichádza na stránky z vyhľadávačov a väčšina používateľov si pozrie nájdené záznamy iba na prvej strane.

Na vyššie umiestnenie stránky vo vyhľadávačoch vplýva:

- * hierarchia webovej stránky a spojenia
- * mapa webového sídla
- * webová stránka s užitočnými a informačne bohatými informáciami
- * dôležitosť slov a ich používania
- * text verzus obrázky
- * značka Title a ALT
- * správny HTML kód

Nájditeľnosť je kvalitatívny atribút webových stránok. Nájditeľnosť znamená, že sme lokalizovateľní.

Prístupnosť webového sídla

Prístupnosť znamená pružné zabezpečenie prispôsobenia sa niečoho potrebám a výberu používateľa. V súvislosti s internetom hovoríme o rozšírení okruhu ľudí, pre ktorých budú počítačové technológie a internet užitočné. Webové stránky by mali byť pre každého jednoduché na používanie. Pri návrhu webovej stránky musíme brať do úvahy, že internet využívajú aj ľudia:

- ktorí nevidia, nepočujú, prípadne nedokážu spracovať určitý typ informácií,
- majú ťažkosti v čítaní alebo v porozumení textu,
- nedokážu používať klávesnicu alebo myš,
- môžu používať iba obrazovku s textom, môžu mať malú obrazovku,
- nehovoria alebo nerozumejú jazykom, v ktorom je napísaná stránka,
- môžu byť práve v situácii, keď ich oči, uši, prípadne ruky sú zaneprázdnené
- môžu mať staršiu verziu prehliadača, úplne inú verziu prehliadača

Prístupné stránky teda rešpektujú používateľa, jeho technické vybavenie, zdravotné dispozície, vedomosti, skúsenosti a vzdelanie. Z hľadiska prístupnosti webového sídla zohráva významnú úlohu WWW konzorcium, ktoré vydáva smernice a odporúčania pre budovanie prístupného webu.

V súčasnosti platia smernice pre budovanie bezbariérového webu Web Content Accessibility Guidelines 1.0, ako W3C Recommendation z piateho mája 1999. V tomto dokumente nájdeme 14 pravidiel (všeobecných zásad) dizajnu prístupných webových stránok:

Zásada 1. Poskytujte rovnocenné alternatívy ku zvukovým a vizuálnym informáciám.

Zásada 2. Nespoliehajte sa len na farbu.

Zásada 3. Náležite používajte značky a definície štýlov.

Zásada 4. Používajte bežný jazyk.

Zásada 5. Vytvárajte tabuľky, ktoré sa správne zobrazujú.

Zásada 6. Zabezpečte, aby sa správne zobrazovali stránky, ktoré používajú nové technológie.

Zásada 7. Zabezpečte, aby používateľ mohol riadiť časované zmeny obsahu.

Zásada 8. Zabezpečte prístupnosť pripojených používateľských rozhraní.

Zásada 9. Vytvárajte nezávisle na používateľom zvolenej technológii.

Zásada 10. Používajte dočasné riešenia.

Zásada 11. Používajte technológie a pravidlá konzorcia W3C.

Zásada 12. Poskytujte informácie o kontexte a orientácii.

Zásada 13. Poskytujte prehľadný navigačný mechanizmus.

Zásada 14. Zabezpečte, aby boli dokumenty zrozumiteľné a jednoduché.

Príklad zabezpečenia prístupnosti:

- Odkazy sú správne odlíšené od ostatného obsahu.
- Netextové prvky (napr. obrázky) majú textovú alternatívu.
- Titulok stránky informuje o jej obsahu.
- Prvky tvoriace nadpisy sú správne označené.
- Text na stránke si môže používateľ v prehliadači sám zväčšiť.

Prístupnosť web stránok.

Prístupnosť web stránok znamená, že musíme počítať z možnosťou, že nie všetci návštevníci našej web stránky majú technické či zdravotné možnosti, aby sa mohli oboznámiť s kompletným obsahom našej prezentácie. Technické možnosti sme už spomenuli. Pokiaľ niekto nemá FLASH, tak si našu stránku, ktorá používa túto technológiu jednoducho neprezrie.

Technológie zvädzajú k používaniu vždy toho najnovšieho, čo je k dispozícii. My, ako autori web stránky, však musíme počítať s tým, že ľudia sú konzervatívni - neradi si zvykajú na nové veci, keď tie staré sú osvedčené.

Preto používajú staršie prehliadače, nemajú nainštalovaný zásuvný modul pre FLASH, majú pomalšie pripojenie na Internet a podobne. Takisto v technickej oblasti musíme rátať s tým, že existujú prostriedky ako napríklad mobilné telefóny, vreckové počítače a podobne, ktoré nedokážu zobraziť web stránku rovnako ako plnohodnotný prehliadač na osobnom počítači.

Zdravotné možnosti sú rovnako pre niektorých návštevníkov veľkou prekážkou pri návšteve niektorých web stránok. Stačia nevhodne zvolené farby a človek s poruchou farebného videnia bude mať problémy s prečítaním textu na stránke. Zvolené farby by mali byť vždy kontrastné (čierna a biela, tmavá a svetlá). Slabší zrak dokáže niektoré nevhodne vytvorené web stránky znepriístupniť úplne. Takisto technológia FLASH dokáže pre takýchto návštevníkov znepriístupniť web stránku tým, že lupy, či čítačky textu, ktoré bežne používajú nevidomí, či slabozrakí návštevníci, nedokážu čítať texty zobrazované prostredníctvom FLASH technológie.

Pokiaľ chceme predávať pomôcky pre slabozrakých a nevidomých, tak predsa neurobíme web stránku s maličkým písmom, ktoré má problém prečítať aj človek so zdravým zrakom. Takisto ju neurobíme vo FLASHi - lebo FLASH ako technológia neumožňuje využiť čítačky textu, ktoré nevidomí používajú ako jediný prostriedok pre prístup k informáciám na Internete. Tiež by sme sa mali zamyslieť nad usporiadaním textov na web stránke. Nevidomí používajú na prezeranie stránok špeciálne pomôcky ako sú Brailov riadok alebo programy, ktoré transformujú písaný text na hlas. Pokiaľ však začiatok našej web stránky bude obsahovať nepodstatné informácie, človek s takouto pomôckou stratí veľa času, kým sa dostane k užitočným informáciám. Preto je napríklad vhodné umiestniť hneď na začiatok navigáciu (menu), ktoré umožní takýmto návštevníkom dostať sa rýchlo priamo k hľadanej informácii.

Aby sme na záver zhrnuli informácie z tejto kapitoly. Vhodne a správne použitá webová prezentácia môže byť pre firmu veľkým plus. Treba si však vždy dopredu premyslieť, pre akú cieľovú skupinu zákazníkov má byť naša web prezentácia určená.

Prístupnosť web stránky je aj o tom, ako ľahko sa človek na stránke zorientuje. Kde a ako sú umiestnené tlačidlá. Je na stránke čitateľné a dostatočne veľké písmo? Sú vhodne zvolené farby (tmavé písmo na tmavom podklade?)? atď.

Kredibilita webového sídla

Iba 53% používateľov internetu verí, že informácia na internete je spoľahlivá a presná, pričom každý rok tento počet používateľov klesá.

Kredibilitu si môžeme zdefinovať ako *dôveryhodnosť*. Tak ako možno povedať, že dôveryhodný človek je spoľahlivý človek, tak isto aj vierohodná informácia je spoľahlivá informácia. Kredibilita alebo dôveryhodnosť je subjektívne vnímaná kvalita. Nie je to teda objektívna vlastnosť webovskej stránky (koľko slov obsahuje, koľko odkazov je na stránke a pod.). Dôveryhodnosť je vnímaná zo subjektívneho hľadiska pozorovateľa. Ľudia vnímajú kredibilitu súčasným zhodnocovaním viacerých aspektov.

* spoľahlivosť

* odbornosť

Spoľahlivosť je definovaná termínmi ako pravdivosť, nezaujatosť, dobrý zámer, atď. Toto v sebe obsahuje vnímanie dobra alebo morálky, etiky zdroja.

Odbornosť možno charakterizovať termínmi ako kompetencia, erudícia, renomovanosť. Tento aspekt v sebe obsahuje vnímané poznatky a kvalifikáciu.

Kredibilita ako vlastnosť sídiel je veľmi dôležitá, pretože ak má webové sídlo u používateľov vysoký punc kredibility, používatelia sa neboja zaregistrovať osobné údaje, kúpiť vec online, vyplniť dotazník, dodávať vlastný obsah, downloadovať softvér, kliknúť na reklamy a čo je najdôležitejšie, vrátiť sa na sídlo.

- Stránka poskytne rýchlu odpoveď na otázku zákazníka
- Stránka poskytuje vyčerpávajúcu informáciu vzťahujúcu sa k danému problému.
- Ku každému článku sú uvedené autorove odporúčania.
- Ku každému článku sú uvedené citácie.
- Na stránke sú možnosti vyhľadávania.
- Stránka má viac jazykových mutácií.
- Na stránke sú aj názory a recenzie používateľov.
- Na stránke sú recenzie alebo ratingy obsahu.
- Na stránke je obsah podľa preferencií používateľa.
- Stránka zistí, že ju používateľ už navštívil.
- Na stránke sú uvedené ocenenia, ktoré získala.

Kreditibilitu znižuje

Na stránke je veľa informácií, ktoré sa nevzťahujú k žiadnemu zdroju.

Na stránke sú typografické chyby.

Stránka je niekedy nedostupná.

Na stránke sú nefunkčné spojenia.

Na väčšinu faktorov odbornosti majú vplyv projektanti stránky.

Faktory spoľahlivosti majúce vplyv na kredibilitu webového sídla.

So stránkou už mám spoľahlivé skúsenosti

Na stránke je uvedená fyzická adresa organizácie

Na stránke je uvedené telefónne číslo

Na stránke je uvedená e-mailová adresa

Na stránku sú odkazy zo stránky, ktorej dôverujem

Na stránke je definovaná politika ochrany súkromia

Zo stránky vedú odkazy na ďalšie užitočné zdroje

Stránka bola odporúčaná v novinách, časopisoch, v elektronickom spravodaji a pod.

Stránku mi odporučil priateľ

Reklama.

K veľmi negatívnym faktorom, ktoré majú vplyv na kredibilitu webového sídla je otázka reklamy

- viac reklamy na jednej stránke

- automaticky sa otvárajúce okná na stránke

- nejasné rozlíšenie reklamy od obsahu stránky

Ďalšie faktory ovplyvňujúce kredibilitu.

+

- sídlo patrí renomovanej spoločnosti
- sídlo bolo od poslednej návštevy aktualizované
- dizajn sídla je profesionálne navrhnutý
- navigácia na stránke dáva zmysel

-

- na stránke sa vyžaduje zalogovanie používateľa
- stránka je komerčne zameraná
- stránka je hostovaná ďalšou stranou
- na stránke sa vyžaduje platený prístup
- stránka sa pomaly naťahuje
- doména nekorešponduje s názvom firmy
- sídlo reprezentuje firmu, ktorá má finančné ťažkosti
- sídlo odkazuje na nedôveryhodnú stránku
- na stránke je zlá navigácia
- na stránke je neaktuálny obsah

Každá návšteva webovej stránky by mala byť pridanou hodnotou pre používateľa (nový aktualizovaný obsah, rýchla odpoveď na otázky používateľa, rýchle vyhľadávanie, kontaktné informácie, možnosť vytlačenia strán v používateľskom formáte, personalizácia stránky a pod.).

Na stránke by mala byť prísne dodržaná integrita obsahu (prísne oddelenie reklamy od obsahu, vyhýbanie sa otvárajúcim sa reklamným oknám, citácia článkov, odporúčania stránky a pod.).

Jednoduchá a efektívna navigácia (na základe prieskumov používateľa vysoko pozitívne hodnotia jednoduchú a intuitívnu navigáciu).

Profesionálny vzhľad stránky vrátane funkčných spojení a rýchleho načítavania stránky.

Používateľnosť

Používateľnosť je kvalitatívny atribút internetových stránok, ktorý určuje ako ľahko sa na nich používateľ orientuje, ako rýchlo pochopí ich usporiadanie a ovládanie a aký zážitok si z nich odnesie.

Dobre použiteľné stránky sú prehľadné, zrozumiteľné a ľahko ovládateľné. Je radosť s nimi pracovať. Na zle použiteľných stránkach používateľia tápajú, nedokážu dosiahnuť svoj cieľ a odchádzajú sklamaní. Používateľnosť je však taktiež odborná disciplína, ktorá exaktnými metódami skúma príčiny používateľských úspechov i neúspechov pri práci s internetovými stránkami. Dokáže napríklad vysvetliť, prečo si mnoho používateľov (zákazníkov) vloží vybraný tovar do nákupného vozíka, ale iba málo z nich dokončí objednávku.

Používateľnosť charakterizuje päť kvalitatívnych komponentov

- **Naučiteľnosť** - používateľ rýchlo pochopí a dokáže používať web, ktorý navštívil po prvýkrát.
- **Efektívnosť** – ako sa používateľ naučí ovládať stránku, ako rýchlo splní požadované úlohy.
- **Zapamätateľnosť** - používateľ si zapamätá usporiadanie a ovládanie webu, keď sa naň po určitej dobe vráti.
- **Chybovosť** - používateľ robí minimum závažných chýb a z každej chyby sa rýchlo spamätá.
- **Spokojnosť** - používanie webu prináša používateľovi príjemný zážitok.

Používateľská priateľnosť sa dá dosiahnuť používateľsky orientovaným dizajnom. Je to filozofia, ktorá pri návrhu systému vychádza z požiadaviek používateľa. Je to proces, ktorý sa zameriava na faktory ako vnímanie, pamäť, učenie sa, riešenie problémov a pod., ktoré prichádzajú do úvahy počas interakcie ľudí so systémom.

Ak sú webové sídla nájdniteľné, prístupné a dôveryhodné, rastie aj ich návštevnosť a spokojnosť zákazníkov.

Termín použiteľnosť sa v poslednom čase u nás začína udomáčňovať ako synonymum anglického termínu usability. Zatiaľ čo v západných krajinách už existujú laboratória na testovanie použiteľnosti a väčšina webových sídiel sa pred spustením seriózne testuje s ohľadom na použiteľnosť, u nás môžeme hovoriť iba o začiatkoch.

Usability (použiteľnosť) môžeme vo všeobecnosti definovať ako ľahké používanie systému pre konkrétne účely.

Použiteľnosť je kvalitatívny atribút internetových stránok, ktorý určuje ako ľahko sa na nich používateľ orientuje, ako rýchlo pochopí ich usporiadanie a ovládanie a aký zážitok si z nich odnesie.

Ak je stránka dobre navrhnutá, používateľ robí minimum chýb a dokáže nájsť čo potrebuje.

Použiteľnosť je však taktiež odborná disciplína, ktorá skúma príčiny používateľských úspechov i neúspechov pri práci s internetovými stránkami. Dokáže napríklad vysvetliť, prečo si mnoho používateľov (zákazníkov) vloží vybraný tovar do nákupného košíka, ale iba málo z nich dokončí objednávku.

Prečo je dôležitá použiteľnosť webových sídiel?

V súčasnosti máme k dispozícii niekoľko miliárd webových stránok a ich počet neustále narastá. Každý deň vychádza vo svete v elektronickej forme okolo 7 miliónov kníh. Tlačený obsah reprezentuje iba 0,003% celého obsahu publikovaného vo svete. Znamená to, že každej jednej vete v tlačených médiách zodpovedá 30 000 viet v digitálnej forme. Keď sa v minulosti hovorilo o informačnom preťažení čitateľov, dnes to platí niekoľkonásobne.

Preto je dôležité, že toto množstvo obsahu treba určitým spôsobom zorganizovať a usporiadať tak, aby používatelia dokázali nájsť informácie na internete. Platí stále viac ako inokedy, obsah, ktorý nikto nenájde ani nikto neprečíta. Preto aplikovaním metodológie používateľsky orientovaného dizajnu a dodržiavaním zásad použiteľnosti sa môže významne zlepšiť spokojnosť zákazníkov s webovými sídlami.

Informácia je prezentovaná podľa priorít používateľov v dotazníku.

Oddelenie kľúčovej informácie od druhotnej informácie (najdôležitejšie informácie sú v strede stránky).

Všetky informácie vzťahujúce sa k určitej téme sú na jednom mieste.

Na každej stránke sa nachádza obsah stránky vo forme hypertextových spojení. Napríklad na stránke o informáciách o rakovine pľúc sú všetky informácie od spôsobu liečenia až po prevenciu, klinické prípady, výskum a zoznam literatúry.

K tej istej informácii sa môžeme dostať viacerými spôsobmi. Záleží to od typu používateľov a tiež ich úrovni vzdelania a informačnej gramotnosti.

Na stránke sa využívajú nadpisy, podnadpisy a zoznamy, ktoré sprehľadňujú stránku a zlepšujú orientáciu na stránke.

Ku všetkým úrovniam informácií by mal byť jednoduchý prístup.

Stránka by mala byť písaná jasným a zrozumiteľným štýlom.

Aj keď stránka prešla od pôvodného redizajnu viacerými zmenami, základné odporúčania na základe prieskumu používateľov ostali zachované.

Použiteľnosť webového sídla patrí k jeho najdôležitejším kvalitatívnym atribútom. Je zbytočné prevádzkovať internetový obchod, ak v ňom nikto nenakupuje, ponúkať služby, ak ich nikto nenájde. Preto je dôležité každé webové sídlo otestovať skutočnými používateľmi, aby sa odhalili problémy použiteľnosti webu.

Už aj test s iba 5 používateľmi odhalí až 85% problémov s použiteľnosťou webového sídla. Test s 15 používateľmi odhalí celých 100% problémov. Náklady investované do testovania sídla sa určite vrátia vo zvýšení návštevnosti sídla a tým aj vo zvýšení tržieb.

Príprava podkladov pre stránku - štruktúra a texty, grafické podklady

Príprava podkladov pre web je dôležitým medzistupňom ku kvalitnej web stránke. Návštevníka stránky musíme zaujať, nesmieme ho znechutiť predkladaním dlhého, nezáživného textu. Zoznam výrobkov by mal byť doplnený obrázkom a stručnou informáciou o cene výrobku a najdôležitejšími parametrami. K web stránke patrí aj vizuálna časť. Podrobné informácie (väčšia fotografia, podrobné parametre ...), môžu byť uložené na samostatnej web stránke.

Dôležité je mať dopredu premyslený základný obsah web stránky. Pokiaľ má web stránka plniť len informačný účel o firme, postačí nám rozdeliť si informácie do niekoľkých skupín:

Profil firmy

Základné informácie o firme, prípadne ďalšie informácie o štruktúre firmy a jej hospodárení (ktoré by mohli byť zaujímavé pre potenciálneho obchodného partnera).

Výrobný program/Produkty/Predmet obchodovania/Služby

Informácie o výrobkoch, ktoré firma vyrába či s ktorými obchoduje, prípadne o službách, ktoré firma poskytuje.

Kontakt

Informácie o možnostiach kontaktu - kde je možné nájsť firmu, jej pobočky, kontaktné telefónne či faxové čísla, e-mailu a podobne. Túto časť je vhodné doplniť mapkou.

Rozvrhnutie obsahu WWW stránky, umiestnenie navigačného menu, grafických prvkov a obsahu

Usporiadanie jednotlivých prvkov na web stránke je dôležité z hľadiska prehľadnosti a prístupnosti. Pred samotnou realizáciou web stránky by sme si mali urobiť niekoľko náčrtov ako si predstavujeme rozloženie obsahu na web stránke.

Príklady štandardne používaných rozvrhnutí web stránky.

Trojstĺpcové rozvrhnutie web stránky:

Hlavička web stránky (v angličtine header)		
Ľavé menu	Obsah web stránky (alebo tiež telo web stránky)	Pravé menu
Pätička web stránky (v angličtine footer)		

Dvojstĺpcové a jednoduché rozvrhnutie web stránky (menu môže byť aj napravo):

Hlavička web stránky (v angličtine header)	
Ľavé menu	Obsah web stránky (alebo tiež telo web stránky)
Pätička web stránky (v angličtine footer)	

Hlavička web stránky (v angličtine header)	
Menu	
Obsah web stránky (alebo tiež telo web stránky)	
Pätička web stránky (v angličtine footer)	

Fantázii sa samozrejme medze nekladú, väčšina stránok však vychádza z niektorého z týchto základných rozvrhnutí.

Podobný náčrt budúceho vzhľadu web stránky nazývame v angličtine **layout**. Toto samozrejme nie je jediný vzhľad, ktorý môže naša web stránka mať. Niekedy sa vynechá pravé menu, niekedy ľavé, prípadne sa obsah web stránky rozdelí na viac stĺpcov. To všetko závisí od toho, pre koho má byť web stránka určená a pre aký účel má byť vytvorená. Od takéhoto náčrtu sa dá rozvíjať samotný grafický vzhľad stránky.

Tiež je vhodné vybrať si veľkosti a typ písma pre menu (pozor na to, že miešať veľa druhov písma v jednom dokumente a teda aj na web stránke sa neodporúča), nadpisy, texty a ďalšie textové informácie. Vhodná farebná paleta, z ktorej sa bude vychádzať pri tvorbe grafického návrhu, sa dá získať napríklad z ilustračného obrázku, ktorý budeme chcieť na web stránke použiť.

Na Internete sa dá nájsť množstvo príkladov web stránok, na ktorých si môžeme názorne ukázať ako môžu dobre (ale aj zle) navrhnuté web stránky vyzerieť. Preto je potrebné tejto fáze návrhu venovať pozornosť a nezanedbať ju.

Ako spracovať grafický návrh do podoby web stránky

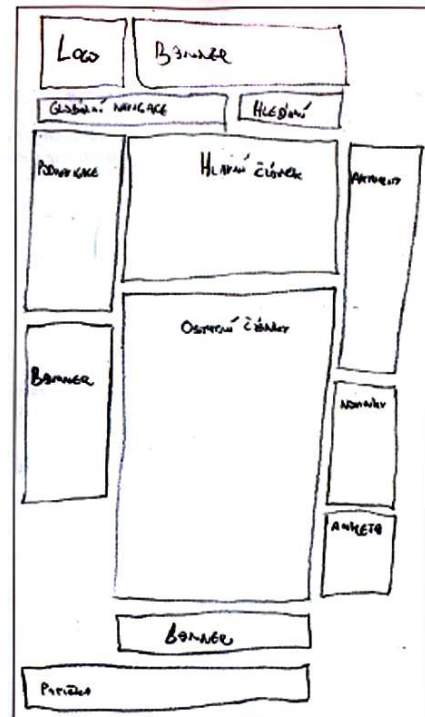
podoby (X)HTML a CSS (webdizajnér). Pre tvorbu grafického návrhu je nutný určitý talent

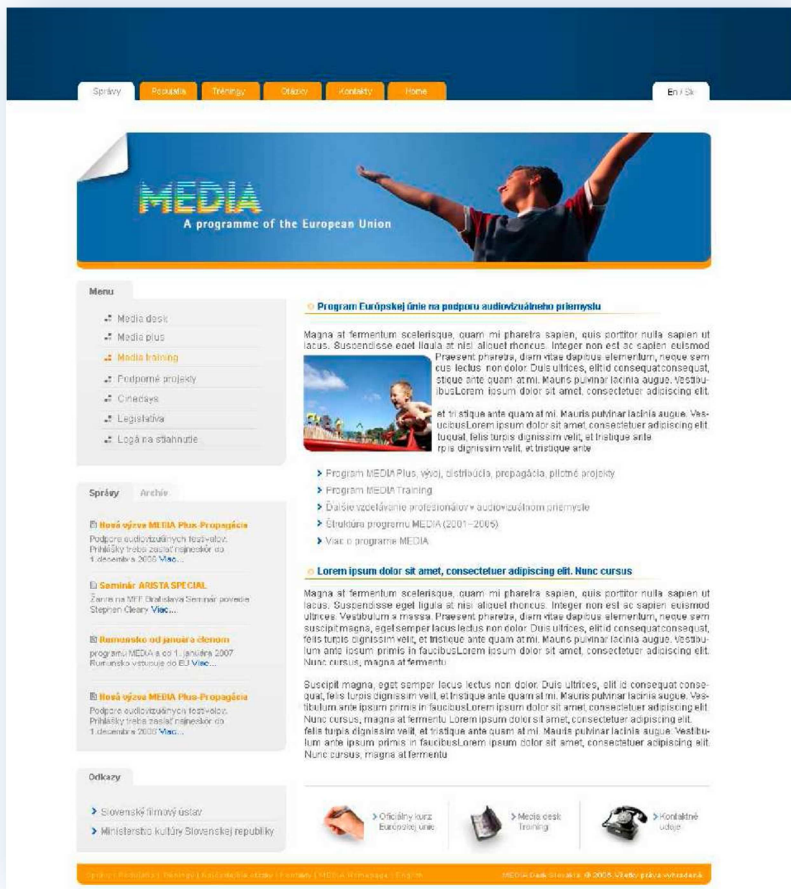
Človek, ktorý navrhuje grafický vzhľad (grafik), má odlišnú úlohu od človeka, ktorý grafický návrh spracúva do stránok. Predpokladá sa dobrá znalosť grafických editorov. Pre spracovanie grafického návrhu do výslednej podoby je nevyhnutná aspoň základná znalosť práce s niektorým z bitmapových grafických programov, umožňujúcich tvorbu a úpravu obrázkov.

Predložený grafický návrh slúži v podstate len ako vzor. Z neho môže webdizajnér podľa vlastnej úvahy vyberať farby a grafické prvky, ktoré sa na výslednej web stránke opakujú a zapracovať ich do (X)HTML a CSS.

Ideálnym riešením je spojenie role grafika a webdizajnéra do jednej osoby, toto však nie je možné vždy dosiahnuť. Bohužiaľ, je len málo dobrých grafikov, ktorí sú zároveň dobrými webdizajnérami.

Dobrý grafický návrh (spracovaný v niektorom bitmapovom alebo vektorovom grafickom programe) môže vyzeráť napríklad takto:





Z takéhoto návrhu sa dá vychádzať pri tvorbe HTML a CSS štýlov. Rozložením takéhoto obrázku na jednoduchšie grafické prvky (farba pozadia, obrázky hlavičky, malé doplňujúce ikony pri zoznamoch) sa dá zostaviť dynamická stránka, na ktorej sa budú meniť len texty a pri tom si stále zachová svoj vzhľad.

Čo je potrebné pre tvorbu WWW stránok - jazyk XHTML a štýlové predpisy CSS

tejto kapitole si povieme o jazyku XHTML a o jeho predchodcovi HTML. Jazyk HTML sa v priebehu histórie jeho používania vyvíjal. Dnes poznáme dve základné verzie jazyka pre tvorbu web stránok:

HTML - HyperText Markup Language [URL1] - posledná oficiálna špecifikácia jazyka HTML je dostupná vo verzii 4.1, na stránkach organizácie **WWW Consortium (W3C)**

XHTML - Extensible HyperText Markup Language [URL2] - v prípade XHTML máme verziu 1.0, ktorá sa ďalej delí na 3 základné vetvy (pre prechod z HTML na XHTML)

XHTML 1.0 Strict - Túto verziu použijeme, pokiaľ chceme mať web založený na značkách určujúcich štruktúru stránky, bez použitia značiek, ktoré sú používané pre úpravu vzhľadu stránky. Miesto týchto značiek budeme používať jazyk pre popis vzhľadu značiek – kaskádové štýly (CSS), za pomoci ktorých popisujeme typy písma, farby a celkové usporiadanie prvkov stránky. Použitie tejto verzie je dnes odporúčané.

XHTML 1.0 Transitional - Túto verziu používajú ľudia, ktorí potrebujú rýchlo modernizovať staršie web stránky a doplniť do nich novú funkčnosť, ako sú napríklad CSS štýly. Táto verzia je prakticky totožná z HTML vo verzii 4.1. Pokiaľ chcete mať web stránku použiteľnú aj v starších prehliadačoch a pritom napísanú v XHTML, môžete použiť aj značky popisujúce vzhľad stránky.

XHTML 1.0 Frameset - Túto verziu použijete, pokiaľ chcete vo svojej web stránke použiť rámy (používať rámy sa už dnes neodporúča, keďže jestvuje množstvo efektívnejších metód, ako dosiahnuť na web stránke podobnú funkčnosť.)

Jazyk XHTML je v súčasnosti odporúčeným pre tvorbu web stránok. Vo svojej podstate sú tieto značkové jazyky takmer identické. Rozdiely existujú hlavne v spôsobe zápisu a pravidlách, ktoré musíme pri písaní web stránok v jazyku XHTML dodržiavať.

XHTML nie je programovací jazyk. XHTML je jazyk, ktorý slúži na popis toho, ako má vyzerat' výsledné zobrazenie textu v prehliadači internetových stránok.

Značky jazyka XHTML, zapísané v texte, číta prehliadač takto:

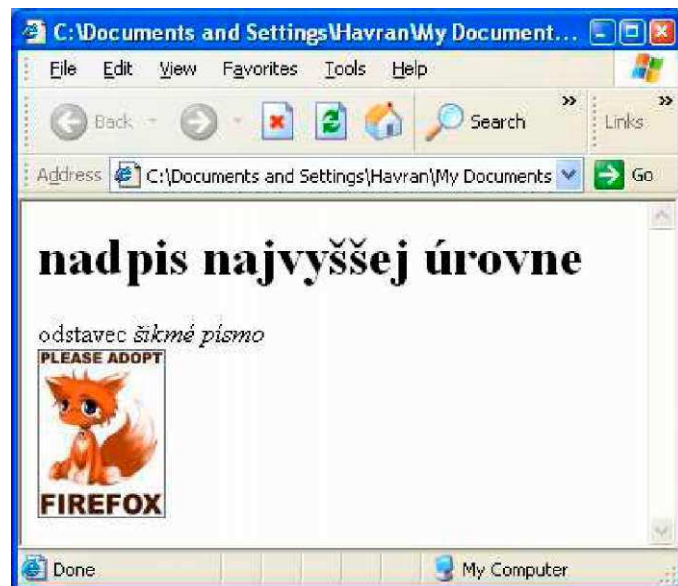
„tu začni písať nadpis najvyššej úrovne, tu ukonči písanie nadpisu najvyššej úrovne, tu začni písať odstavec, tu začni písať šikmé písmo, tu ukonči písanie šikmého písma, tu choď na ďalší riadok, tu vlož obrázok s alternatívnym textom "Obrázok", tu ukonči odstavec..."

A podobne pokračujeme ďalej.

V reálnom jazyku XHTML by zápis vyzeral takto:

```
<h1>nadpis najvyššej úrovne</h1>  
<p>odstavec <em>šikmé písmo</em><br />  
</p>
```

A zobrazený v prehliadači internetových stránok zasa takto:



Tomuto spôsobu spracovávaní jazyka XHTML hovoríme interpretácia (čítanie a spracúvanie), a tomu, ako prehliadač web stránku zobrazuje na svojej zobrazovacej ploche, hovoríme renderovanie (zobrazovanie).

Toto nám hneď napovedá, že výsledné zobrazenie web stránky závisí len a len od toho, akým spôsobom prehliadač jednotlivé značky interpretuje a zobrazí.

Každý prehliadač má vlastný spôsob, ktorým spracúva web stránku, a preto môžeme nájsť drobné rozdiely v zobrazení tej istej web stránky v rôznych prehliadačoch.

Jazyk XHTML vo verzii Transitional poskytuje vo svojich značkách veľa nastavení, ktoré pre prehliadač popisujú, ako má byť text zobrazený. Sú to vlastnosti ako typ písma, farba písma, farba pozadia, vzdialenosť textu od okrajov, hrúbka a farba okrajov a podobne.

Tieto nastavenia sú prebraté z jazyka HTML a dnes sa už prakticky nepoužívajú, keďže máme k dispozícii oveľa mocnejší nástroj na popis a nastavenie vzhľadu stránky - CSS.

Základné pravidlá pre zápis značiek jazyka XHTML (HTML)

Zápis párových značiek jazyka XHTML (HTML)

Značka jazyka XHTML **strong** (dôraz na označený text, vizuálne vyjadrený hrubým písmom) použitá vo web stránke vyzerá takto:

```
<strong>zdôraznený text</strong>
```

Každá značka jazyka XHTML vždy začína znakom < (menší) a končí znakom > (väčší). Použitá značka **strong** je značka **párová** (jazyk XHTML obsahuje párové a nepárové značky) - to znamená, že len text, ktorý je medzi začiatočnou značkou **** a zakončovacou značkou **** (všetky párové zakončovacie značky sa začínajú </ a končia >) bude zdôraznený - zobrazený hrubým písmom. Toto platí pre všetky párové značky! (všetky značky majú nejaký význam - strong, z angličtiny znamená silný; br, pochádza so skratky break (rozbiť, v tomto kontexte to znamená „rozbiť“ vetu na dva riadky)).

Zápis nepárových značiek jazyka XHTML (HTML)

V jazyku XHTML však existujú aj značky, ktoré sú **nepárové**. Sú to značky, ktoré neurčujú vzhľad nejakej časti textu, ale plnia inú úlohu. Tieto značky musíme v jazyku XHTML zapisovať trochu iným spôsobom. Príklad - značka **br** (break), ktorá určuje, kde má prejsť zobrazovanie textu v prehliadači na nový riadok:

```
<br />
```

Dalšími podobnými značkami sú **img** pre vloženie obrázku (image), **hr** pre vloženie vodorovnej oddeľovacej čiary (horizontal rule) a ďalšie.

Zápis parametrov (atribútov) značiek jazyka XHTML (HTML)

Ďalšou vecou, ktorú si musíme pri zápise značiek jazyka XHTML ukázať, sú **parametre** (atribúty) značky. Ako príklad si uvedieme značku pre vloženie obrázku - **img**:

```
<img src = "obrazok.jpg" alt = "Obrázok" title="PLEASE ADOPT FIREFOX!" />
```

Značka **img** hovorí - sem sa vloží obrázok, avšak sama o sebe nepovie prehliadaču, o aký obrázok ide a kde ho má prehliadač vziať. Na to práve slúži parameter **src** (source), v ktorom sa uvádza meno obrázku a prípadne celá cesta k obrázku (kde, v ktorom adresári na vzdialenom web serveri sa nachádza obrázok). Značka jazyka XHTML môže mať aj viac parametrov ako jeden - v našom prípade sú to parametre **alt** (alternatívny text, ktorý sa použije v prípade, keď nie je možné zobrazit' obrázok, alebo v prípade, keď stránku číta špeciálna čítačka pre zrakovo postihnutých) a **title** (text v tejto značke uvádza podrobnejší popis obsahu obrázku, ktorý sa zobrazí keď podržíme kurzor na obrázku).

Parameter musí byť vždy uvedený v úvodzovkách! (Toto neplatilo v HTML, parametre, pokiaľ parametrom nebol napríklad text s medzerami, nemuseli byť v úvodzovkách). Na druhú stranu je dobrým mravom používať úvodzovky v HTML pre parametre značiek vždy. Vyhnete sa tým chybám a obtiažnejšiemu neskoršiemu dopĺňaniu úvodzoviek.

Zanorenie párových značiek jazyka XHTML (HTML)

Značky jazyka XHTML je možné vkladať jednu do druhej (zanoriť ich), ale treba pri tom dodržiavať určité pravidlá.

Nesprávne!

```
<strong>  
<em>Tento text je zobrazený hrubým a šikmým písmom.</strong> </em>
```

Správne.

```
<strong>  
<em>Tento text je zobrazený hrubým a šikmým písmom.</em> </strong>
```

Znamená to, že sa značky nesmú navzájom krížiť. Párové značky zanorené do seba sa musia zakončovať v opačnom poradí, než v akom sa začínali (čiže posledná vložená značka sa musí zakončiť prvá).

Prehliadač sa síce podobný zápis pokúsi zobrazíť a väčšinou ho zobrazí správne, ale niekedy sa stáva, že podobný zápis znemožní správne zobrazenie obsahu web stránky.

Zle zapísané značky jazyka XHTML sťažujú prehliadača proces čítania a zobrazovania web stránky!

Základná štruktúra XHTML (HTML) dokumentu

HTML dokument

Každý HTML dokument má predpísanú základnú štruktúru - kostru dokumentu. HTML dokument je celý ohraničený (zanorený) v párovej značke `<html>.. ,</html>`, ktorá obsahuje hlavičku dokumentu `<head>.. ,</head>` a telo dokumentu `<body>...</body>`.

```
<!DOCTYPE html PUBLIC "-//IETF//DTD HTML 4.01//EN">
<html>
<head>
<title> ... titulka stránky ... </title> . . . ostatné prvky hlavičky ... </head> </body>
... viditeľný obsah dokumentu (telo dokumentu) ...
</body> </html>
```

Aby boli spracovateľné aj staršie HTML dokumenty, tak značky `<html>`, `</html>` `<head>`,`</head>` `<body>`,`</body>` nemusia byť zapísané vždy - prehliadače si dokážu poradiť aj s dokumentom bez týchto značiek. Samozrejme, pokiaľ sa vytvára nový HTML dokument, je vhodné dodržiavať túto základnú štruktúru.

Jazyk HTML nie je citlivý na veľkosť písma - to znamená, že všetky značky by mohli byť zapísané aj veľkými písmenami (`<html>` ako `<HTML>` a pod.). V jazyku XHTML však už toto neplatí, a preto je **jediný perspektívny spôsob**, ako zapisovať značky jazyka HTML **v použití malých písmen**.

XHTML dokument

Každý XHTML dokument má predpísanú základnú štruktúru - kostru dokumentu. XHTML dokument je celý ohraničený (zanorený) v párovej značke **<html>...</html>**, ktorá obsahuje hlavičku dokumentu

<head>. . .**</head>** a telo dokumentu **<body>**. . .**</body>**.

```
<?xml version="1.0" encoding="windows-1250"?>
<!DOCTYPE html PUBLIC "-//IETF//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd>
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="sk" lang="sk">
<head>
<meta http-equiv="content-type" content="text/html; charset=windows-1250" /> <meta http-equiv="content-language"
content="sk" />
<title> ... titulka stránky ... </title> . . . ostatné prvky hlavičky ... </head>
<body>
... viditeľný obsah dokumentu (telo dokumentu) ...
</body> </html>
```

Značky **<html>**,**</html>** **<head>**,**</head>** **<body>**,**</body>** musia byť zapísané vždy (to znamená že nesmú byť vynechané). Značky musia byť zapísané malými písmenami.

XHTML Úvod

26. januára 2000 bol organizáciou W3C doporučený nový nástupca dnes používaného formátovacieho jazyka HTML. Stal sa nim XHTML. Už z názvu vyplýva, že ide o jazyk, ktorý využíva výhody XML a HTML.

Prechod na XHTML by mal byť o to jednoduchší, že XHTML - až na niekoľko málo odlišností je spätne kompatibilný s HTML. Už dnes je bežná tvorba stránok v XHTML, ktoré by sa mali bezproblémovo zobrazit' v dnešných prehliadačoch. Kompletná špecifikácia jazyka XHTML je k dispozícii na adrese

<http://www.w3.org/TR/xhtml1/>

Príklad XHTML stránky:

V nasledujúcej tabuľke je uvedený zdrojový kód minimálnej XHTML stránky. Prakticky nijako sa neliší od bežného HTML dokumentu. Bol pridaný prvý riadok, ktorý hovorí, že ide o XML dokument verzie 1.0 s kódovaním UTF-8.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "DTD/xhtml11-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="sk" lang="sk">
  <head>
    <title>Moja prvá XHTML stránka</title>
  </head>
  <body>
    <p>Toto je jednoduchá XHTML stránka.</p>
  </body>
</html>
```


Najväčšie rozdiely XHTML oproti HTML, tak ako sú uvedené na stránkach konzorcia W3C:

Kríženie tagov v XHTML

V XHTML nie je možné kríženie jednotlivých tagov. **V XHTML ma kríženie tagov jediný výsledok - nezobrazenie celého dokumentu.**

Zlý postup:

```
<font face="arial"><b><i>Text</font></b></i></b></i>
```

Správny postup:

```
<font face="arial"><b><i>Text</i></b></font>
```

Ukončovacie tagy

V XHTML neexistujú tagy, ktoré nemusia byť ukončené nepárovým tagom. Všetky párové či nepárové tagy tak musia byť ukončené lomítkom. Musíte si zvyknúť na fakt, že v XHTML bude nutné ukončovať napríklad tagy `
` (zalomenie riadku), `<hr>` (čiara), či ukončiť dokonca tag pre vloženie obrázku `` na stránku.

Zlý postup:

```
<br>
```

```
<hr>
```

```
<img>
```

Správny postup:

```
<br></br> alebo iba <br />
```

```
<hr></hr> alebo iba <hr />
```

```
<img></img> alebo iba <img />
```

Ukončovanie všetkých tagov dnes prevažne používajú tvorcovia WML stránok. WML stránky sú určené pre zobrazovanie v mobilných telefónoch.

Hodnoty v úvodzovkách

V HTML je normálne, že hodnoty v parametroch tagov sú niekedy vložené do úvodzoviek a niekedy zase nie sú. XHTML každopádne definuje ako štandard iba jedno - **všetky hodnoty musia byť v úvodzovkách**.

Zlý postup:

```
<font face=arial size=2 color=black>
```

Správny postup:

```
<font face="arial" size="2" color="black">
```

Malé písmená pre všetko

Ďalší podstatný rozdiel. V HTML je dnes jedno akým písmom sú tagy písané. Je možné ľubovoľne striedať veľké a malé písmená, je to proste úplne jedno.

V XHTML to avšak jedno nie je - podľa definície **musí byť všetko písané malými písmenami**. Napríklad a sú v XHTML odlišné tagy.

Zlý postup:

Správny postup:

Základné otázky:

1. Prečo je potrebné XHTML? Nie je HTML dost' dobré?

HTML je najúspešnejší značkovací jazyk pre dokumenty na svete. Ale keď vzniklo XML, bolo na dvojdrňovej konferencii diskutované, či je potrebná nová verzia HTML v XML. Všeobecným názorom bolo „Ano“: s HTML založeným na XML môžu ostatné jazyky získať kusy XHTML a XHTML dokument môže obsahovať iné značkovacie jazyky. Zmeny návrhu boli využité k vyčisteniu niektorých častí HTML a pridali viac funkčnosti.

2. Aké sú výhody použitia XHTML oproti HTML?

Pokiaľ je dokument iba čisté XHTML 1.0 (neobsahuje žiadne ďalšie značkovacie jazyky) potom nezaznamenáme príliš mnoho zmien. Ale ako vznikajú ďalšie a ďalšie nástroje pre XML, ako je XSLT pre transformáciu dokumentov, začínajú sa objavovať výhody použitia XHTML. XForms napríklad umožňuje editovať XHTML dokumenty (alebo nejaký iný druh XML dokumentu) veľmi jednoducho. Aplikácie pre sémantický Web budú schopné využiť výhody XHTML dokumentov.

Pokiaľ je dokument viac než XHTML 1.0, napríklad obsahuje MathML, SMIL alebo SVG, potom výhody sú jasné: to by s HTML nebolo možné.

3. Je možné iba pridať XML deklaráciu na začiatok existujúceho HTML dokumentu? Je možné zmiešať HTML a XHTML dokumenty?

Nie. HTML nieje XML formát. Je potrebné urobiť zmeny potrebné k tomu, aby bol dokument korektný pred tým ako z neho urobíme XML dokument.

4. Ako najjednoduchšie je možné skonvertovať HTML dokumenty na XHTML?

[HTML Tidy](#) dáva možnosť transformovať akýkoľvek HTML dokument do XHTML 1. [Amaya](#) je prehliadač / editor, ktorý ukláda HTML dokumenty ako XHTML.

5. Prečo sú prehliadače k XML tak striktné? Pri HTML boli viac tolerantné.

Je to zámerne. HTML prehliadače akceptujú akýkoľvek vstup, aj nekorektný a snažia sa s tým urobiť niečo zmysluplné. Opravovanie chýb spôsobuje, že sú prehliadače veľmi ťažko programovateľné, hlavne pokiaľ sa očakáva, že by všetky prehliadače mali robiť rovnakú vec. Bolo tiež zistené, že mnoho HTML dokumentov je chybných, pretože keď sa v prehliadači zobrazia správne, autor si neuvedomí chyby. To spôsobuje, že je zložitá napísať nový software, keď dokumenty, ktoré sa tvária ako HTML sú často veľmi nekorektné.

6. Kde je možné overiť, či XHTML dokument je validný?

W3C ponúka službu na adrese <http://validator.w3.org/>. Prehliadač / editor [Amaya](#) taktiež umožňuje zistiť validitu stránok..

7. Prečo musíme používať menné priestory v XHTML?

V začiatkoch HTML rôzne skupiny a spoločnosti pridávali nové elementy a atribúty do HTML ako sa im zachcelo. To vyústilo v chaos spolupráce neschopných verzií HTML. XML (to X znamená eXtensible - rozšíriteľný) umožňuje každému použiť elementy z rôznych jazykov, ale prehliadaču, alebo inému užívateľskému softwaru musíme určiť, ktorý element patrí ku ktorému jazyku.

8. Prečo je povolené posilať dokumenty XHTML 1.0 ako text/html?

XHTML je XML formát; to znamená, že by malo byť odoslané s media typom z rodiny XML (application/xhtml+xml, application/xml, alebo text/xml). Avšak XHTML 1.0 bolo navrhnuté aby pracovalo taktiež aj na starších prehliadačoch. Pri dodržaní niekoľkých jednoduchých doporučení, nám dokážu aj staré prehliadače zobrazovať XHTML 1.0. Pričom si treba uvedomiť, že prehliadač použije HTML parser.

9. Ktoré prehliadače prijímajú media type application/xhtml+xml?

Medzi známe prehliadače patria prehliadače z rodiny Mozilly ako je Mozilla, Netscape 5+, Galeon a Firefox, taktiež Opera, Amaya, Camino, Chimera, DocZilla, iCab, Safari, a všetky prehliadače mobilných telefónov, ktoré pracujú s WAP 2. Pre detaily si pozrite stránky [XHTML Media-type test](#).

Zhrnutie rozdielov v zápise medzi HTML a XHTML

Keďže XHTML je implementáciou XML, oproti HTML má niektoré dodatočné požiadavky, ktoré zjednodušujú jeho preklad a uľahčuje zapamätanie pravidiel človekom.

Značky musia byť písané malými písmenami.

Párové značky musia byť ukončené. (Značka "option" sa v XHTML považuje za párovú.)

HTML: `<p>Toto je odsek, <tr><td>Prvá bunka<td>Druhá bunka, <option>Možnosť 1`

XHTML: `<p>Toto je odstavec.</p>, <tr><td>Prvá bunka</td><td>Druhá bunka</td></tr>, <option>Možnosť 1</option>`

Nepárové značky musia obsahovať koncovú lomku.

HTML: `
<input type="text" name="text1">`

XHTML: `
<input type="text" name="text1" />`

Vlastnosti musia byť uvedené v úvodzovkách alebo apostrofoch.

HTML: `<td colspan=3>`

XHTML: `<td colspan="3">`

Vlastnosti musia mať vždy uvedenú hodnotu.

HTML: `<textarea readonly>Nejaký text</textarea>`

XHTML: `<textarea readonly="readonly">Nejaký text</textarea>`

Koreňový element. Aby prehliadač vedel v XML dokumente rozlíšiť XHTML od iných jazykov (SVG, MathML), musí mať koreňový element (`<html>`) uvedený menný priestor.

HTML: `<html lang="sk">`

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

Tagy v XHTML 1.0

Tag	Funkcia	DTD
A		
<a>	Definuje prepojenie na iný dokument	S/T/F
<abbr>	Hovorí o tom, že text vo vnútri je skratka	S/T/F
<acronym>	Hovorí o tom, že text vo vnútri je zložené slovo	S/T/F
<address>	Hovorí o tom, že text vo vnútri je adresa	S/T/F
<applet>	Slúži k stiahnutiu a spusteniu apletu	T/F
<area />	Definuje oblasť v obrázkovej mape	S/T/F
B		
	Zobrazí tučný text vo vnútri tagu	S/T/F
<base />	Určuje základnú adresu pre relatívne odkazy	S/T/F
<basefont />	Určuje základný font dokumentu	T/F
<bdo>	Mení smer zobrazenia textu v tagu	S/T/F
<big>	Zväčší veľkosť písma o jednu úroveň	S/T/F
<blocquote>	Označuje citáciu bloku textu	S/T/F
<body>	Definuje telo dokumentu	S/T/F
 	Zalomí text	S/T/F
<button>	Zobrazí tlačítko formulára	S/T/F
C		
<caption>	Definuje titulok tabuľky	S/T/F
<center>	Zarovná text vo vnútri tagu na stred	T/F

<cite>	Text vo vnútri tagu je citácia	S/T/F
<code>	Text vo vnútri tagu je programový kód	S/T/F
<col />	Definuje stĺpec v skupine stĺpcov	S/T/F
<colgroup>	Definuje skupinu stĺpcov v tabuľke	S/T/F
<>		
D		
<dd>	Uzatvára popis definície v zozname definícií	S/T/F
	Označuje odstránenú časť textu	S/T/F
<dfn>	Označuje výskyt pojmu v dokumente	S/T/F
<dir>	Definuje zoznam súborov, alebo adresárov	T/F
<div>	Vytvorí nový oddiel dokumentu	S/T/F
<dl>	Definuje zoznam definícií	S/T/F
<dt>	Uzatvára definíciu v zozname definícií	S/T/F
E		
	Zvýrazní text vo vnútri tagu	S/T/F
F		
<fieldset>	Vytvorí skupinu prvkov vo formulári	S/T/F
	Nastaví vzhľad písma	T/F
<form>	Vymedzuje v dokumente formulár	S/T/F
<frame />	Definuje rámec	F
<frameset>	Definuje skupinu rámcov	F
H		
<head>	Definuje hlavičku dokumentu	S/T/F

<hr />	Vloží do dokumentu vodorovný oddělovač	S/T/F
<html>	Vymedzuje dokument	S/T/F
<h1> až <h6>	Text vo vnútri je nadpis úrovně 1 až 6	S/T/F
I		
<i>	Text vo vnútri tagu sa zobrazí kurzivou	S/T/F
<iframe>	Vloží do dokumentu plávající rámec	T/F
	Vloží do dokumentu obrázok	S/T/F
<input />	Vytvorí prvok formulára	S/T/F
<ins>	Označí text vo vnútri tagu ako vložený text	S/T/F
<isindex />	Zobrazí pole pre vyhľadávanie	T/F
K		
<kbd>	Označuje text zapísaný klávesnicou	S/T/F
L		
<label>	Definuje popisok ovládacieho prvku formulára	S/T/F
<legend>	Definuje legendu ovládacích prvkov formulára	S/T/F
	Vytvorí položku zoznamu	S/T/F
<link />	Pripojí iný dokument	S/T/F
M		
<map>	Definuje obrázkovú mapu	S/T/F
<menu>	Definuje zoznam ponúk	S/T/F
<meta />	Definuje doplňujúce informácie o dokumente	S/T/F
N		
<noframes>	Text vo vnútri tagu sa zobrazí ak nieje podpora ramcov	T/F

<noscript> Text vo vnútri tagu sa zobrazí ak nieje podpora skriptov S/T/F

O

<object> Vloží do dokumentu objekt S/T/F

 Definuje číslovaný zoznam S/T/F

<optgroup> Vymedzuje skupinu volieb ponuky S/T/F

<option> Definuje položku ponuky S/T/F

P

<p> Vymedzuje odstavce S/T/F

<param /> Definuje parameter značky S/T/F

<pre> Zobrazí text ako je zapísaný vo vnútri tagu S/T/F

Q

<q> Označuje citáciu S/T/F

S

<s> Vypíše preškrtnutý text vo vnútri tagu T/F

<samp> Označuje ukážku, alebo príklad S/T/F

<script> Definuje script v dokumente S/T/F

<select> Vytvorí ponuku v dokumente S/T/F

<small> Zmenší veľkosť písma o jednu úroveň S/T/F

 slúži pre určenie štýlu textu vo vnútri tagu S/T/F

<strike> Vypíše preškrtnutý text vo vnútri tagu T/F

 Vypíše tučný text vo vnútri tagu S/T/F

<style> Definuje štýl formátujúceho dokumentu S/T/F

<sub> Zobrazí text vo vnútri tagu ako dolný index S/T/F

| | | |
|------------|---|-------|
| <sup> | Zobrazí text vo vnútri tagu ako horný index | S/T/F |
| T | | |
| <table> | Vymedzuje v dokumente tabuľku | S/T/F |
| <tbody> | Definuje skupinu riadkov tabuľky | S/T/F |
| <td> | Vytvára bunku tabuľky | S/T/F |
| <textarea> | Vytvorí textové pole vo formulári | S/T/F |
| <tfoot> | Vymedzuje päť tabuľky | S/T/F |
| <th> | Vytvára bunku v záhlaví tabuľky | S/T/F |
| <thead> | Vymedzuje záhlavie tabuľky | S/T/F |
| <title> | Vymedzuje titulok dokumentu | S/T/F |
| <tr> | Definuje riadok buniek v tabuľke | S/T/F |
| <tt> | Zobrazí text neproporcionálnym písmom | S/T/F |
| U | | |
| <u> | Zobrazí text vo vnútri tagu podčiarknutý | T/F |
| | Definuje zoznam s odrážkami | S/T/F |
| V | | |
| <var> | Text vo vnútri tagu je premenná | S/T/F |

Vysvetlivky: S - XHTML 1.0 Strict, T - XHTML 1.0 Transitional, F - XHTML 1.0 Frameset

Elementy

Elementom nazývame celú sekvenciu začínajúcu sa počiatočným tagom a končiacu tagom ukončovacím. Existujú tri základné druhy elementov, blokové, inline a nahradzované. Podľa významu, ktorý textu priradzujú ich môžeme rozdeliť ešte na elementy pre štruktúrovanie dokumentu, textové elementy, elementy pre tvorbu odkazov, elementy pre tvorbu tabuliek, elementy pre tvorbu zoznamov atď.

Blokové elementy

Sú to elementy, ktoré tvoria blok. Zjednodušene to znamená, že po takomto elemente je text dokumentu zalomený, odriadkovaný. Blokovými elementami sú napr. elementy pre nadpis (h1 až h6), element pre odstavce (p).

Inline elementy

Inline elementy sú elementy, ktoré sa nachádzajú vnútri textu. Nedochoádza po nich k zalomeniu. Obvykle plnia funkciu zvýraznenia časti textu. Je to napr. element pre hypertextový odkaz (a).

Nahradzované elementy

Tieto elementy sú nahradené nejakým obsahom, pre začlenenie do dokumentu sú dôležité ich rozmery. Je to napr. element pre obrázok (img).

Atribúty

Elementy môžu mať svoje atribúty. Atribúty priradujú vlastnosti danému elementu. Atribúty sa zadávajú do počiatočného tagu, môže ich byť aj viac (oddeľujú sa medzerou), ale nemusí byť aj žiadny. Každý atribút má svoju hodnotu. Hodnota atribútu musí byť v XHTML zapísaná v úvodzovkách.

Príklad: Všeobecný zápis elementu s atribútmi

```
<element atribút="hodnota _atribútu">Text zobrazovaný na stránke.</element>
```

Príklad: Element s viacerými atribútmi

```
<input type="hidden" name="meno_premennej" value="hodnota_premennej" />
```

Pravidlá XHTML

Pred samotným dokumentom sa nachádza deklarácia XML:

```
<?xml version="1.0" encoding="windows-1250"?>
```

Povinná je deklarácia typu dokumentu (DTD):

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

alebo

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

alebo

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

Element *html* obsahuje atribút *xmlns*, ktorý určuje menný priestor dokumentu (namespace) a jazyk, ktorý je v dokumente použitý:

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

Element *html* vždy obsahuje dva elementy, *head* (záhlavie) a *body* (telo dokumentu). Záhlavie musí obsahovať element *title*.

```
<head>  
  <title>Titulok stránky</title>  
</head>
```

Ďalej môže obsahovať meta značky, linky ku kaskádovým štýlom, linky k javascriptom a pod. Kvôli starším prehliadačom by mala obsahovať najmä meta značku pre kódovanie

```
<head>
```

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

```
<meta http-equiv="Content-Language" content="sk" />
```

```
<meta http-equiv="ImageToolbar" content="no" />
```

```
<title>Titulok stránky</title>
```

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

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

```
<link rel="shortcut icon" href="favicon.ico" type="image/x-icon" />
```

```
<base href="http://www.stuba.sk" />
```

```
<script src="script.js" type="text/javascript"></script>
```

```
</head>
```

Meta značky

Meta značky nastavujú kódovanie dokumentu, opisujú web stránku, určujú autora stránok (alebo len niektorých častí), automaticky presmerovávajú, určujú kľúčové slová a je aj možnosť nastavenia si vlastných meta značiek. Toto všetko a ešte viac slúži nie len pre internetové vyhľadávanie a radenie stránok v katalógoch, ale aj pre komunikáciu s inými aplikáciami ako je internetový prehliadač.

Meta značky zapisujeme pomocou elementu `<meta />`, ktorý môžeme vložiť iba do záhlavia dokumentu.

Meta značka sa vo všeobecnosti skladá z dvoch povinných častí (atribútov). Prvým atribútom je buď `name` (všeobecná informácia) alebo `http-equiv` (systémová informácia). V tomto atribúte (`name`, `http-equiv`) je napísané, aký druh meta informácie bude poskytovaný (napríklad `name="keywords"` budú kľúčové slová). Druhým povinným atribútom je `content` (vlastný obsah meta informácie). Meta značka schématicky vyzerá nasledovne:

```
<meta name="druh informácie" content="jej obsah" />
<meta http-equiv="druh informácie" content="jej obsah" />
```

Meta značka pre nastavenie kódovania:

```
<meta http-equiv="Content-Type" content="text/html; charset=windows-1250" />
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-2" />
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
```

Meta značka pre automatické presmerovanie

```
<meta http-equiv="refresh" content="5; url=http://www.stuba.sk" />
```

Hodnota `refresh` znamená, že sa má otvorená web stránka obnoviť za dobu, ktorá je určená na 5 sekúnd a presmerovať na adresu `http://www.stuba.sk`.

Meta značka pre neukladanie stránky do cache pamäti

```
<meta http-equiv="pragma" content="no-cache" />
```

Meta značka pre opis stránky

```
<meta name="description" content="stránka o XHTML" />
```

Meta značka pre kľúčové slová

```
<meta name="keywords" content="Výuka, XHTML, CSS, ..." />
```

Meta značka pre autora stránky

```
<meta name="author" content="Meno Priezvisko" />
```

Meta značka pre vlastníka stránky

```
<meta name="owner" content="Meno Priezvisko" />
```

Meta značka pre copyright

```
<meta name="copyright" content="&copy; 2010 Firma a.s." />
```

Meta značka pre indexovanie vo vyhľadávačoch

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

Meta značka pre neindexovanie vo vyhľadávačoch

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

Meta značky – zhrnutie

Atribút	Hodnota	DTD
content	ľubovoľný text	Strict, Transitional, Frameset
http-equiv	content-type expires refresh set-cookie	Strict, Transitional, Frameset
name	author description keywords generator revised <i>iné</i>	Strict, Transitional, Frameset
scheme	ľubovoľný text	Strict, Transitional, Frameset

Kostra stránky

Dokument začína deklaráciou XML, nasleduje DTD, potom tag HTML obsahujúci záhlavie a telo dokumentu.

Príklad 1.7: Základná stránka - súbor s príponou htm alebo html

```
<?xml version="1.0" encoding="windows-1250"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="sk" lang="sk">
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=windows-1250" />
  <title>Titulok stránky</title>
</head>
<body>
  <h1>Nadpis</h1>
  <p>Text text text</p>
</body>
</html>
```

Nadpisy

Existuje 6 úrovní nadpisov, označujú sa *h1*, *h2*, *h3*, *h4*, *h5* a *h6*, kde 1 je najvyššia úroveň. Úrovne sa líšia veľkosťou písma, všetky nadpisy sú implicitne zarovnávané vľavo.

Príklad:

```
<h1>Nadpis 1. úrovne</h1>  
<h2>Nadpis 2. úrovne</h2>  
<h3>Nadpis 3. úrovne</h3>  
<h4>Nadpis 4. úrovne</h4>  
<h5>Nadpis 5. úrovne</h5>  
<h6>Nadpis 6. úrovne</h6>
```

Výsledok príkladu:

Nadpis 1. úrovne

Nadpis 2. úrovne

Nadpis 3. úrovne

Nadpis 4. úrovne

Nadpis 5. úrovne

Nadpis 6. úrovne

Odkazy

Odkazy sú základom každej web stránky. Bez odkazov by sa užívateľ nikam nedostal. V texte sú odkazy vizuálne odlišené, štandardne je to farebným odlišením a podtrhnutím. Kliknutím na odkaz je tento odkaz aktivovaný a dochádza k presmerovaniu.

Pre odkaz sa používa element `a`. Jeho atribútom je `href`. Ten udáva, kde je umiestnený cieľový dokument (teda na ktorú stránku odkaz smeruje). Samotné umiestnenie potom zadávame absolútnou, alebo relatívnou adresou.

Odkaz na inú stránku

Absolútny odkaz

Absolútny odkaz sa používa pre odkaz na dokument umiestnený na inom serveri, napr. ako odkaz na inú stránku. Odkaz na inú stránku musí vždy obsahovať `http://`.

Príklad: `STU v BA`

Výsledok príkladu: [STU v BA](http://www.stuba.sk)

Hodnotou atribútu `href` je adresa cieľového dokumentu (v tomto prípade `http://www.stuba.sk`). Zobrazovaný a klikateľný text bude `STU v BA`.

Relatívny odkaz

Relatívny odkaz sa používa, pokiaľ sa odkazujeme v rámci rovnakého servera. Pri použití relatívneho odkazu prehliadač sám doplňuje URL stránky, z ktorej je odkazované. Pokiaľ teda odkazujeme na dokument nachádzajúci sa v rovnakom adresári, stačí napísať iba názov súboru.

Príklad: `úvod`

Výsledok príkladu: [úvod](index.php)

Ak sa dokument nachádza v inom adresári, musíme do URL zahrnúť celú cestu od miesta, kde sa cesty k odkazovanému a odkazujúcemu dokumentu začali odlišovať.

Príklad: `odkaz na anchor`

Výsledok príkladu 1.11: [odkaz na anchor](#)

Keď sa budeme chcieť dostať v adresárovej štruktúre o úroveň vyššie, použijeme k tomu dve bodky (..). Pokiaľ by sme teda chceli zo stránky `anchor.html` (viď. vyššie) odkázať na súbor `anchor2.html`, ktorý je uložený v adresári *priklady*, použijeme zápis:

Príklad: Súbor `anchor.html` `odkaz na anchor2`

Príklad: Súbor `anchor2.html` `späť na kurz`

Relatívny odkaz na konkrétne miesto dokumentu

Niekedy je dobré odkazovať sa aj na časti stránky (napr. pokiaľ je stránka dlhšia — potom v úvode použijeme odkazy na jej jednotlivé časti a ďalej pokiaľ z inej stránky potrebujeme použiť odkaz na presne určenú časť stránky (aby užívateľ nemusel hľadať). Na označenie miesta, na ktoré chceme odkázať použijeme atribút `id`, v odkaze potom použijeme ako vždy atribút `href`, ktorý ale bude tentoraz začínať znakom `#`. Často sa toto používa tiež pri odkazovaní na začiatok stránky.

Príklad: `späť na začiatok`

Výsledok príkladu: [späť na začiatok](#)

Takýto odkaz môžeme dať niekam na koniec stránky, po kliknutí prehliadač sám „odroluje“ tak, že sa element označený príslušným názvom dostane do hornej časti okna prehliadača.

Poznámka: Odkazovaná časť dokumentu sa do hornej časti okna pochopiteľne nedostane v prípade, že pokračovanie stránky nie je dostatočne dlhé k zaplneniu celého okna prehliadača (potom bude niekde vnútri okna...).

Rovnakým spôsobom potom odkazujeme aj z inej stránky, len musíme pridať klasickým spôsobom odkaz na celý dokument a za nim odkaz na konkrétne miesto.

Príklad: Odkaz na `nadpis s návestím odkaz`

Výsledok príkladu: [Odkaz na nadpis s návestím odkaz](#)

Časť dokumentu, na ktorú takto odkazujeme označíme takto (v našom prípade budeme odkazovať na nadpis stránky druhej úrovne):

Príklad: `<h2 id="odkaz">Nadpis s návestím odkaz</h2>`

Výsledok príkladu: [Nadpis s návestím odkaz](#)

Titulok odkazu

Každý odkaz by mal mať svoj titulok. Tým je text, ktorý užívateľovi priblíži, kam vlastne odkaz smeruje. Titulok sa zobrazí pri nabehnutí kurzoru myši na odkaz. Titulky odkazov sú dôležité tiež napr. pre čítacie zariadenia, umožňujú lepšiu orientáciu na webe.

Príklad: `STU v BA`

Výsledok príkladu: [STU v BA](http://www.stuba.sk)

Zoznamy

Neusporiadaný (nečíslovaný) zoznam

Neusporiadaný (nečíslovaný) zoznam sa vytvára tagom *ul* (unordered list). Položka zoznamu je *li* (list item). Položka môže obsahovať aj viac odstavcov. Pred každou položkou sa štandardne vytvára odrážka.

Príklad:

```
<ul>
  <li>linux</li>
  <li>windows</li>
  <li>unix</li>
</ul>
```

Výsledok príkladu:

- linux
- windows
- unix

Usporiadáný (číslovaný) zoznam

Usporiadáný (číslovaný) zoznam sa vytvára tagom `ol` (ordered list). Položka zoznamu je opäť `li`. Pred položku sa automaticky vypisuje jej poradové číslo.

Príklad:

```
<ol>  
  <li>linux</li>  
  <li>windows</li>  
  <li>unix</li>  
</ol>
```

Výsledok príkladu:

1. linux
2. windows
3. unix

Zoznam definícií

Zoznam definícií sa používa obvykle v prípade, keď potrebujeme napr. vypísať nejaké termíny a k nim ich definície alebo popisy. Používa sa tag `dl` (definition list). Definovaný termín sa označuje tagom `dt` (term), definícia termínu tagom `dd` (definition).

Príklad:

```
<dl>
  <dt>OS</dt>
  <dd>operačný systém</dd>
  <dt>HDD</dt>
  <dd>pevný disk </dd>
  <dt>CPU</dt>
  <dd>procesor</dd>
</dl>
```

Výsledok príkladu:

OS

operačný systém

HDD/FDD

pevný/pružný disk

CPU

procesor

Tabuľky

Tabuľka sa vytvára pomocou elementu **table**. Riadok tabuľky sa definuje tagom **tr** (table row), pre bunky tabuľky sa používa tag **th** (table head; pre bunky v záhlaví tabuľky) a **td** (table data). Bunky sa vkladajú vnútri riadku tabuľky, koľko buniek, toľko bude mať tabuľka stĺpcov. Bunky záhlavia sú štandardne formátované tučným písmom, zarovnané na stred.

V tabuľke sa používa atribút **summary**, ktorý zhrňuje obsah tabuľky. Slúži pre lepšiu prístupnosť webu.

Príklad:

```
<table summary="skúšobná tabuľka">
<tr>
  <th>1</th>
  <th>2</th>
</tr>
<tr>
  <td>aaa</td>
  <td>bbb</td>
</tr>
<tr>
  <td>ccc</td>
  <td>ddd</td>
</tr>
</table>
```

Výsledok príkladu:

1	2
aaa	bbb
ccc	ddd

Nadpis tabuľky

Na nadpis tabuľky sa používa element **caption**. Ten by mal byť uvedený vnútri tagu `table` ešte pred prvým riadkom.

Príklad:

```
<table>
<caption>Nadpis tabuľky</caption>
<tr>
  <th>1</th>
  <th>2</th>
</tr>
<tr>
  <td>aaa</td>
  <td>bbb</td>
</tr>
<tr>
  <td>ccc</td>
  <td>ddd</td>
</tr>
</table>
```

Výsledok príkladu:

Nadpis tabuľky

1	2
aaa	bbb
ccc	ddd

Zlučovanie buniek

Atribút **colspan** zlúči dohromady niekoľko buniek v jednom riadku. Atribút **rowspan** zlúči bunky v stĺpci. Ako hodnotu týchto atribútov zapisujeme počet buniek, ktoré chceme takto zlúčiť.

Príklad:

```
<table>
<caption>Nadpis tabuľky</caption>
<tr>
  <th colspan="2">1</th>
</tr>
<tr>
  <td rowspan="2">aaa</td>
  <td>bbb</td>
</tr>
<tr>
  <td>ccc</td>
</tr>
</table>
```

Výsledok príkladu:

Nadpis tabuľky	
1	
aaa	bbb
	ccc

Obrázky

Na vloženie obrázku slúži nepárový element **img**. Povinnými atribútmi sú *src*, jeho hodnotou je názov obrázku, prípadne aj s cestou a *alt* obsahujúci alternatívny text, ktorý sa zobrazí, pokiaľ je obrázok prehliadača nedostupný, buď preto, že sa na zadanej adrese nenachádza alebo preto, že prehliadač obrázky nezobrazuje. Alternatívny text by mal teda nejakým spôsobom priblížiť čo je na obrázku tým, ktorí ho nevidia.

Príklad:

```

```

Výsledok príkladu:



Obrázky na webe by sme mali na webe používať opatrne. Okrem výberu vhodných obrázkov je treba dbať na ich dátovú veľkosť, zďaleka nie každý si môže dovoliť prehliadať stránky, ktorých veľkosť sa dá počítat v megabytoch. Obrázky, ktoré nemajú inú funkciu než „zlepšenie designu“ stránky by mali byť vkladané iba pomocou CSS štýlu a to ako obrázok na pozadí.

Veľkosť obrázku

Veľkosť obrázku nie je povinne zadávať, aj preto je veľkou chybou toto opomenúť. Pokiaľ totiž veľkosť obrázku nezadáte, prehliadač si na jeho zobrazenie nevyhradí dostatočný priestor a keďže text okolo sa samozrejme načíta skôr než obrázok (pretože je čo sa týka veľkosti dát menší), spôsobí to „poskakovanie“ stránky vo chvíli, keď sa začínajú načítavať obrázky, ktoré potrebujú viac priestoru než majú a text okolo im musí uhýbať.

Rovnako je vhodné zadávať obrázku jeho skutočnú veľkosť, nezmenšovať ani nezväčšovať pomocou prehliadača, ale pomocou grafického editora. Pokiaľ totiž obrázok o veľkosti napr. 100×100 pixelov zmenšíme v prehliadači na 50×50, docielime tak síce požadovanú veľkosť, ale užívateľ bude zbytočné sťahovať viac dát, pretože bude sťahovať samozrejme obrázok v pôvodnej veľkosti.

Pre určenie veľkosti obrázku sa používajú atribúty *width* (šírka) a *height* (výška) alebo vlastnosti s rovnakým menom v CSS.

Obrázková mapa

Obrázkové mapy využijeme napr. pre rôzne navigačné lišty. Je to obrázok, ktorého jednotlivé časti majú slúžiť ako odkazy na rôzne miesta. Mapa sa vytvára pomocou elementu **map**. Jeho povinným atribútom je *id*, čo je atribút slúžiaci ako identifikátor tejto mapy. Jeho hodnotou je teda nejaký textový reťazec, na ktorý sa neskôr bude obrázok tejto mapy odvolávať.

Element **map** obsahuje ľubovoľný počet (najmenej jeden) elementov **area**, ktoré vymedzujú priestor z obrázku, ktorý bude odkazom a určuje kam bude odkazovať. Tvar hranice aktívnej oblasti (t.j. miesto, na ktoré budeme môcť kliknúť a odkázať sa na daný dokument) určuje atribút *shape*. Jeho hodnotou je buď *rect* (obdĺžnik), *poly* (mnohouholník) alebo *circle* (kruh).

Konkrétne vymedzenie oblasti je dané v atribúte *coords*. Jeho hodnotou sú čísla oddelené čiarkou, ktorá definuje vrcholy n-uholníku. V prípade obdĺžnika je to v poradí: *súradnice x ľavého horného rohu*, *súradnice y ľavého horného rohu*, *súradnice x pravého dolného rohu*, *súradnice y pravého dolného rohu*. Obdobne po sebe budú nasledovať všetky vrcholy mnohouholníka. V prípade kružnice sú zadávané čísla v poradí: *súradnice x stredu*, *súradnice y stredu*, *polomer kružnice*.

Poznámka: Počiatok sústavy súradníc je v ľavom hornom rohu, teda ľavý horný roh má súradnice [0,0]. Súradnica *x* rastie v smere zľava doprava, súradnica *y* v smere zhora dole.

Povinným atribútom pri *area* je *alt* a nie menej dôležitým *href*, ktorý určuje cieľ odkazu. Takto môžeme teda rozdeliť obrázok na niekoľko aktívnych oblastí. Mapa je už vytvorená, stačí iba povedať obrázku, aby ju použil. Toho docielime pomocou atribútu *usemap* (atribút tagu *img*). Jeho hodnotou bude ten istý reťazec, aký sme použili v tagu *map* pri atribúte *id*. A teraz už len zostáva si to všetko ukázať na konkrétnom príklade:

Príklad:

```
<map id="mapa" name="mapa">  
  <area href="http://www.stuba.sk" alt="časť 1" shape="rect" coords="0,0,33,31" />  
  <area href="http://www.sme.sk" alt="časť 2" shape="rect" coords="34,0,66,31" />  
  <area href="http://www.php.sk" alt="časť 3" shape="rect" coords="67,0,98,31" />  
</map>
```

```

```

Výsledok príkladu:



Každá z troch častí obrázku skutočne odkazuje inam.

Úlohy

1. Vytvorte stránku, ktorá bude obsahovať nasledujúcu tabuľku:

		2	3	4
1		5	6	7
8	9			
10	11	12	13	

2. Vytvorte tri obsahom rovnaké web stránky s rôznym kódovaním:

- o kodovanie_cp1250.html (XHTML 1.0 Strict)
- o kodovanie_iso88592.html (XHTML 1.0 Strict)
- o kodovanie_utf8.html (XHTML 1.0 Strict)

Stránku rozdeľte na dve časti (ľavú a pravú) pomocou tabuľky. V ľavej časti sa zobrazí menu s linkami na jednotlivé stránky. Kliknutím na príslušnú linku sa zobrazí stránka s daným kódovaním. Text stránky sa zobrazí v pravej časti stránky.

Pri tvorbe stránok použite základné značky – <table>, <tr>, <td>, <p>, <a>, <h1>, ... s ich atribútmi, ktoré sa ale netýkajú formátovania.

Ochranári: Šúr bude upadať

Ak sa okamžite nezačne s čiastočnou obnovou vodného režimu v prírodnej rezervácii Šúr, zrýchlená mineralizácia pôdy sa nezastaví a rozpad vzácneho typu slatinného jelšového lesa bude pokračovať. Tvrdí to riaditeľ Správy chránenej krajiny oblasti Malé Karpaty Viliam Klescht.



[Ochranári \(kódovanie windows\)](#)

[Ochranári \(kódovanie iso\)](#)

[Ochranári \(kódovanie utf\)](#)

Obrázok: Odvodnenie Šúru počas druhej svetovej vojny zapríčinilo doteraz vyhynutie asi 33 druhov rastlín a 20 druhov živočíchov.

Hrubá vrstva rašeliny sa podľa jeho slov scvrkáva a mení na „obyčajnú“ pôdu. Hrozi, že vzácny slatinný jelšový les postupne nahradí iný, menej vzácny typ lužného lesa.

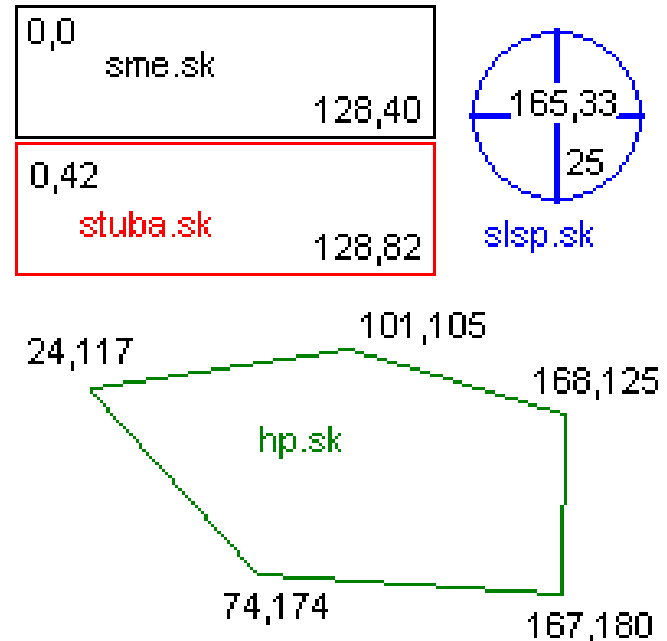
„Z lokality budú ubúdať a miznúť ďalšie vzácne a u nás ohrozené druhy rastlín a živočíchov. Pokračujúcu degradáciu slatinných biotopov bude sprevádzať nežiaduce rozširovanie nepôvodných invázných druhov rastlín, ako sú zlatobyľ obrovská, pohánkovec, viaceré druhy astry či dvojzub listnatý,“ hovorí Klescht.

Nedostatok pripovrchovej vody ohrozuje podľa neho aj slanomilné trávy v Panónskom háji a jeho okolí. „V prírodnej rezervácii Šúr majú osobitný význam, pretože tu rastú slanomilné druhy rastlín, ktoré sú zaradené v červenej knihe vzácných a ohrozených druhov. Kvôli nim bolo územie zaradené do európskej sústavy chránených území NATURA 2000.“

Obr. Ukážka jednej s web stránok (kliknutím na obrázok sa zobrazí v originálnej veľkosti)
Text a obrázok ku stránke podľa kódovania:

- Text stránky
 - [kódovanie windows-1250](#)
 - [kódovanie iso-8859-2](#)
 - [kódovanie utf-8](#)
- Text pre obrázok
 - [kódovanie windows-1250](#)
 - [kódovanie iso-8859-2](#)
 - [kódovanie utf-8](#)
- [Obrázok](#)

3. Vytvorte obrázkovú mapu. Ako obrázok použite nasledujúci obrázok:



Čo sú to štýlové predpisy (štýly) CSS

Kaskádové štýly alebo CSS [URL3] (skratka z angl. Cascading Style Sheets) je všeobecné rozšírenie (X)HTML o možnosti opisu vzhľadu textu základnými parametrami bežného DTP (skratka z angl. Desktop publishing je označenie pre celý proces spracovania grafického návrhu na počítači až po export súboru pre tlač). Štýly umožňujú oddeliť štruktúru HTML alebo XHTML od vzhľadu stránky.

Pomocou štýlov môžeme meniť tieto základné atribúty vzhľadu jednotlivých prvkov na stránke:

- okraj (**border**) (jeho vzhľad, farbu, hrúbku...),
- vnútorné odsadenie od okraja (**padding**),
- vonkajšie odsadenie od okraja (**margin**),
- písmo (**font**) (jeho typ, veľkosť, hrúbku...),
- pozadie (**background**) (farbu, obrázok a jeho umiestnenie na pozadí...),
- pozíciu prvku v stránke (**position**) (spolu s určením rozmerov prvku to poskytuje zaujímavé možnosti pre stavbu web stránky),
- ďalšie možnosti...

Základné pravidlá pre použitie CSS štýlov

Štýlový predpis (popis vzhľadu prvku na stránke) je možné vložiť do (X)HTML tromi rôznymi spôsobmi.

Vkladanie CSS ako parametru značky (X)HTML

Do každej značky jazyka (X)HTML môžeme za pomoci parametru style doplniť jej štýlový predpis.

```
<h1 style="color : blue;">Nadpis odstavca</h1> <p style="color : green;">Odstavec...</p>
```

príklade sme použili CSS pre nastavenie farby nadpisu na modrú a pre nadstavenie farby písma v odstavci na zelenú. CSS sa dá používať aj týmto spôsobom, ale je to ten najmenej efektívny spôsob. Ďalej si popíšeme efektívnejšie (a správnejšie) spôsoby použitia CSS.

Vkladanie CSS do hlavičky dokumentu (X)HTML

```
<head>
```

```
<style type="text/css"> h1 { color : blue; } p { color : green; } </style> </head>
```

príklade sme použili CSS pre rovnaký účel ako v predchádzajúcom príklade (vkladanie CSS ako parametru značky (X)HTML) s tým rozdielom, že takýto zápis predpisuje **všetkým nadpisom h1** a **všetkým odstavcom p**, ktoré sa nachádzajú v tele stránky rovnaký štýl - farbu písma.

Vkladanie CSS z externého súboru

Obsah externého súboru **styl_stranky.css**:

```
/1 externy CSS subor */ h1 { color : blue; } p { color : green; }
```

Aby sme tento štýlový predpis mohli v stránke použiť, musíme v hlavičke web stránky zapísať:

```
<head>
```

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

Tento spôsob je podobný predošlému (vkladanie CSS do hlavičky dokumentu (X)HTML) s tým rozdielom, že externý súbor môžeme rovnakým spôsobom vložiť do súboru každej (X)HTML stránky. Potom sa každá zmena, ktorú v tomto súbore štýlového predpisu urobíme, prejaví na zobrazení každej (X)HTML stránky.

Toto je najvýhodnejší spôsob, ako CSS môžeme v našich stránkach používať. Predchádzajúce dva sú vhodné na testovanie, prípadne tam, kde nie je nutné používať rovnaký štýl na viacerých web stránkach. Znaky */** a */** sú použité pre ohraničenie odstavca s poznámkou, ktorá nemá vplyv na nastavenie štýlov (X)HTML dokumentu.

Použitie CSS tried (class) a selektorov

Keďže táto oblasť je veľmi rozsiahla, uvedieme len niekoľko najčastejšie používaných prípadov.

Typový selektor X - platí pre každú značku X na stránke

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

```
<h1>Hocijaký nadpis h1 bude mať modré písmo.</h1> <p>Hocijaká značka p na stránke bude mať zelené písmo.</p>
```

Tento prípad sme si ukázali v časti Vkladanie CSS do hlavičky dokumentu (X)HTML.

ID selektor

```
h1#nadpis { color : blue; } #odstavec { color : green; }
```

```
<h1 id="nadpis">Len nadpis h1 s použitým ID selektorom „nadpis“ bude mať modré písmo.</h1>
```

```
<p id="odstavec">Hocijaká značka na stránke s použitým ID selektorom „odstavec“ bude mať zelené písmo.</p>
```

Pritom platí, že ID musí byť v rámci stránky jedinečné, žiadne dva prvky nesmú mať rovnaké ID. Toto sa používa na označenie hlavných oblastí na stránke, ako sú napríklad hlavička stránky, oblasť s menu, hlavná oblasť pre obsah stránky atď.

CSS Triedy (class)

```
h1.nadpis { color : blue; } .odstavec { color : green; }
```

```
<h1 class="nadpis">Všetky nadpisy h1 s použitou triedou „nadpis“ budú mať modré písmo.</h1>
```

```
<p class="odstavec">Hocijaká značka na stránke s použitou triedou „odstavec“ bude mať zelené písmo.</p>
```

Na rozdiel od ID selektora platí, že trieda nemusí byť jedinečná, môže sa v rámci stránky opakovať.

V značke je možné použiť aj niekoľko tried súčasne.

```
.odstavec { color : green; } .hrubepismo { font-weight : bold; }
```

```
<div class="odstavec hrubepismo">Tieto dve triedy hovoria že písmo ohraničené touto značkou bude zelenej farby a hrubé.</div>
```

Takto je možné kombinovať určité vlastnosti dohromady, bez toho aby sme museli rovnakú definíciu CSS opakovať viackrát.

Zápis vlastností CSS

Keď chceme pre značku, či triedu zmeniť súčasne viac vlastností, môžeme ich kombinovať takto:

```
.odstavec {  
font-size : 12px; /* veľkosť písma */ font-weight : bold; /* hrubka písma */ color : green; /* farba písma */  
}
```

Ako je vidieť, veľkosť písma (font-size) sa dá určiť za pomoci čísla. Konkrétne v tomto prípade sme použili jednotku pixel (px - bod na obrazovke). Je možné použiť viac typov jednotiek.

Absolútne jednotky (tie, ktoré by mali byť rovnako veľké napríklad po vytlačení na papier aj na obrazovke):

mm - milimetre,

cm - centimetre (1cm = 10mm),

in - inch (palec 1in = 2,54cm),

pt - point (typografický bod 1pt = 1/72in),

pc - pica (1pc - 12pt).

Relatívne jednotky (odvodené od určitej základnej veľkosti):

em - stupeň aktuálneho písma (v podstate percentuálne vyjadrenie veľkosti - napríklad 1.2em znamená 120%),

ex - stredná výška aktuálneho písma (x-výška),

px - pixel (obrazovkový bod).

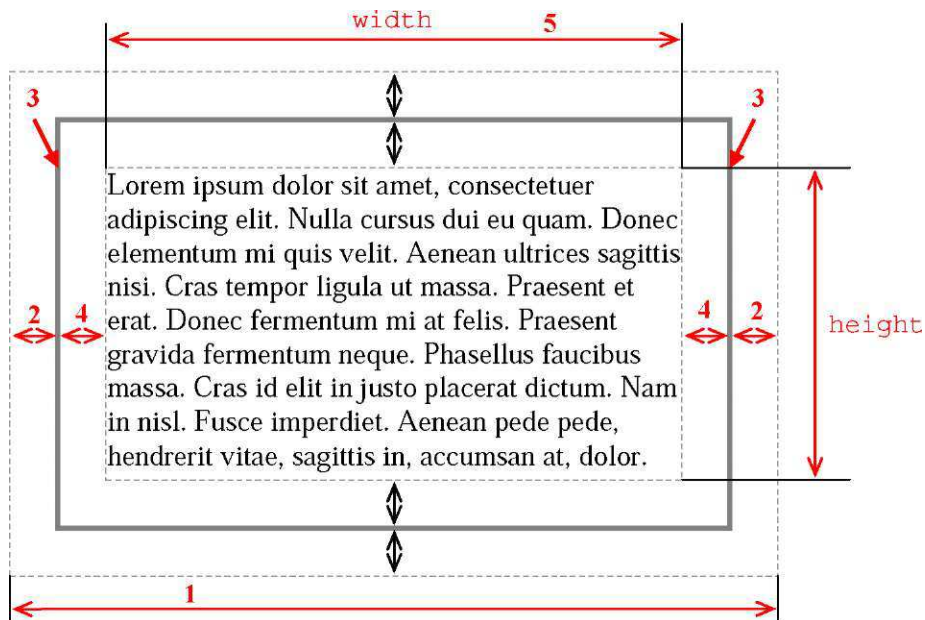
Ďalej platí pravidlo, že vždy platí posledný zápis štýlového predpisu. Toto sa používa v prípade, keď existujúci súbor zo štýlovým predpisom nechceme prepisovať. Jednoduchým vložením nového štýlového predpisu, ktorý bude v druhý v poradí, môžeme prepísať hodnoty nastavené predchádzajúcim zápisom. Príklad:

```
.nadpis { font-size: 14pt; } /* povodny styl */  
.nadpis { font-size: 16pt; } /* upraveny styl */
```

Blokový model (box model)

Základom celého spôsobu práce so štýlovými predpismi je tok dokumentu - spôsob, akým za sebou nasledujú (zobrazujú sa) jednotlivé oštylované (alebo neoštylované) značky (X)HTML dokumentu a tzv. box model. Box model určuje, akým spôsobom sa zobrazujú na stránke tzv. blokové prvky. Základným zobrazením textu je tzv. **inline** (za sebou) spôsob. Všetky prvky sú v toku dokumentu zobrazované postupne, za sebou. Čistý text, odkazy (značka **a**) hrubé písmo (**strong**) atď., sa zobrazujú v jednom riadku za sebou, pokiaľ nenarazia na okraj prehliadača (alebo nadradeného bloku). Vtedy automaticky dôjde k odriadkovaniu - presunu nasledujúceho celého slova na nový riadok.

Bloky (alebo boxy) (**block**) sú prvky, ktoré môžu mať určenú výšku a šírku prostredníctvom parametrov CSS **width** a **height** a majú aj ďalšie parametre (vonkajší okraj (**margin**), okraj (**border**), vnútorný okraj (**padding**) a pozadie (**background**)). Pokiaľ tieto parametre určené nemajú, platí pravidlo, že blok je široký na maximálnu možnú šírku okna prehliadača (alebo jeho nadradeného bloku, ktorý šírku určenú má) a zároveň sa takýto blok automaticky prispôsobí výške obsahu. Keď máme dva bloky za sebou, každý sa zobrazuje na samostatnom riadku (pokiaľ nepoužijeme vlastnosť CSS **float**, ktorou môžeme určiť, akým spôsobom sa majú jednotlivé bloky skladať za sebou (pomocou tejto vlastnosti sa dá napríklad zabezpečiť, aby text „obtekal“ nejaký blok zľava (**left**), či sprava (**right**)). Príklad bloku:



Šírku bloku (boxu) ⁽¹⁾ vyrátame ako súčet nastavení vonkajších okrajov **margin-left + margin-right** ⁽²⁾, okrajov **border-left + border-right** ⁽³⁾ vnútorných okrajov **padding-left + padding-right** ⁽⁴⁾ a šírky **width** ⁽⁵⁾. Výšku rátame rovnakým spôsobom - **margin-top + margin-bottom + border-top + border-bottom + padding-top + padding-bottom + height**. (left, z anglického vľavo, - right, z anglického vpravo, - top, z anglického hore, - bottom, z anglického dole).

Pozor - prehliadač Internet Explorer v niektorých prípadoch nesprávne ráta šírku boxu - v tomto prehliadači je šírka boxu nastavená len parametrom **width**, vonkajší okraj, okraj aj vnútorný okraj sú vo vnútri boxu (rovnako je to aj pre výšku bloku).

Pokiaľ je to možné, je vždy najvhodnejšie ponechať prispôsobenie rozmerov bloku na prehliadač. Pokiaľ nastavíme jeho šírku a výšku, tak sa nám môže stať, že text, alebo vložený blok, ktorý bude v bloku napísaný „pretečie“ mimo blok (čo je v prípade, ak použijeme pre blok okraj či zvýraznené pozadie okamžite viditeľné). Či je prvok typu **inline**, **block** a či sa má vôbec zobrazovať (**display: none;**), sa dá nastavovať pomocou CSS vlastnosti **display**. To znamená, že aj prvok, ktorý je štandardne **inline**, sa dá za pomoci nastavenia vlastnosti **display** zmeniť na prvok typu **block**. Pre vlastnosť display existujú ešte ďalšie hodnoty, ale tie nebudeme na úvod potrebovať.

Zmena pozície prvku (bloku)

Ďalšou vlastnosťou, ktorá je pri práci z CSS často využívaná, je možnosť zmeny pozície bloku. Túto vlastnosť voláme **position** a má štyri nastavenia.

static - základné nastavenie pre všetky prvky na stránke, pozícia sa nemení, prvky sa zobrazujú na mieste, kde sú normálne vykreslené,

relative - pozícia prvku je vypočítaná relatívne od miesta, kde by bol prvok normálne vykreslený, ostatné prvky nie sú týmto posunom ovplyvnené,

absolute - pozícia prvku je určená vzhľadom k jeho nadradenému prvku, prvok s **position: absolute**; je vyňatý z normálneho toku dokumentu,

fixed - pozícia prvku je určená rovnako ako v prípade **position: absolute**; ale zároveň je prvok uzamknutý na rovnakej pozícii v okne prehliadača (prvok sa nebude hýbať keď rolujeme stránkou).

Samotnú pozíciu prvku potom meníme pomocou niektorej z vlastností **top**, **left**, **bottom**, **right**. Pekný príklad, ako sa po zmene vlastnosti **display** správa zobrazovaný blok, nájdete na stránke [QuirksMode.org](http://quirksmode.org) [URL4].

Na nasledujúcom obrázku je zobrazený prípad, keď je blok z obsahom posunutý relatívne (**relative**) o 100 pixelov zvrchu (**top**) a zľava (**left**) voči svojmu predchodcovi:

- Home
- Contact
- Resources
- Sitemap
- Preferences
- Junior webontwikkelaars gezocht
- The book
- The blogs
- JavaScript
- **CSS**
 - Table of contents
 - General
 - Selectors
 - Pseudo-classes
 - **Declarations**
 - **position**
 - display
 - overflow
 - min and max
 - cursor
 - List styles
 - outline

The position declaration

The **position** declaration lets you declare what the position of an element should be.

Test

Test **position** by changing its value for the test element.



position: fixed is not supported by Explorer 6 and lower on Windows; nor by Explorer 7 in Quirks Mode. In Strict Mode Explorer 7 supports it fine, though.

The four values

This is the test element. It has the following style:

```
top: 100px;
left: 100px;
padding: 1em;
border: 1px solid #cc0000;
```

position takes four values: **static**, **relative**, **absolute**, and **fixed**. **static** is the default value, for any other value you have to apply a CSS declaration.

This element is an absolutely positioned child of the test element.

2. Kaskádové štýly

2.1. Úvod do CSS

2.1.1. História

CSS (Cascading Style Sheets) resp. kaskádové štýly vznikli ako súhrn metód pre úpravu vzhľadu stránok. Prvý návrh normy bol zverejnený v roku 1994, v roku 1996 bola vydaná špecifikácia CSS 1, v roku 1998 CSS 2, v súčasnosti sa pracuje na verzi CSS 3.

2.1.2. Na čo je to dobré?

CSS sa využíva na formátovanie obsahu HTML, XHTML a XML dokumentov. V porovnaní s formátovaním pomocou atributov v HTML formátovacie schopnosti rozširuje. Štýly umožňujú presne určiť, ako bude ktorý element vypadáť. Narozdiel od atributov štýlom môžeme definovať jednotný vzhľad elementu pre celý dokument (napr. že všetky nadpisy úrovne 1 budú červené) a to jediným zápisom pre príslušný element. Rovnako tak môžeme pomocou štýlu určiť odlišné formátovanie pre napr. len jediný výskyt určitého elementu. Tým sa jednak zbavíme veľkého množstva kódu, jednak sa tento kód stane oveľa prehľadnejší. Navyše pokiaľ sa raz rozhodneme zmeniť napríklad farbu písma všetkých odstavcov, bude to pre nás otázka niekoľko málo sekúnd, meniť každý atribút v každom elementu v HTML by bola katastrofa. Jeden štýl môžeme ľahko použiť pre ľubovoľné množstvo stránok.

Ukážka možností CSS

Nasledujúce linky odkazujú na stránky, ktoré majú rovnaký obsah. Rozdiel je iba v pripojenom súbore CSS, ktorý ovplyvňuje vzhľad celej stránky.

- [Stránka bez štýlov - základ pre všetky stránky](#)
- [Jednoduché formátovanie 1](#) [\[zobraziť súbor CSS\]](#)

- [Jednoduché formátovanie 2](#) [\[zobraziť súbor CSS\]](#)
- [Formátovanie s menu](#) [\[zobraziť súbor CSS\]](#)
- [Komplexné formátovanie 1](#) [\[zobraziť súbor CSS\]](#)
- [Komplexné formátovanie 2](#) [\[zobraziť súbor CSS\]](#)

2.1.3. Začínáme

Štýl sa skladá z pravidiel pre jednotlivé elementy, ktoré majú byť formátované. Každé takéto pravidlo má dve časti, **selektor** (názov elementu, pre ktorý má toto pravidlo platiť) a **deklaráciu** (čo má pre neho platiť). V deklarácii určujeme vlastnosť a jej hodnotu, deklarácia je uzavrená do zložených zátvoriek. Celé to zapisujeme takto:

```
selektor {vlastnosť: hodnota_vlastnosti;}
```

Príklad 2.1:

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

Selektorom, teda elementom, ktorý formátujeme je *h1* (nadpis 1. úrovne). Deklarácia je *{color: blue}*. Tá určuje, že vlastnosť *color* bude mať hodnotu *blue*. Celé dokopy to teda znamená, že všetky nadpisy 1. úrovne v dokumentu budú mať modrú farbu.

Pokiaľ budeme chcieť určiť elementu viac než jednu vlastnosť, jednotlivé vlastnosti od seba oddelíme bodkočiarkou. Takto môžeme definovať ľubovoľné množstvo vlastností.

```
selektor {
```

```
vlastnosť1: hodnota_vlastnosti1;  
vlastnosť2: hodnota_vlastnosti2;  
}
```

Pokiaľ budeme chcieť určiť dvom elementom ich spoločnú vlastnosť, oddelíme od seba jednotlivé selektory čiarkou.

```
selektor1, selektor2 {  
  vlastnost: hodnota_vlastnosti;  
}
```

2.1.4. Dedičnosť

Väčšina vlastností sa dedí. To znamená, že element, ktorý nemá vlastnosť definovanú, dedí ju po nadradenom elemente. Týka sa to predovšetkým vlastností písma — farby, veľkosti, štýlu atď. Pokiaľ teda chceme definovať nejakú vlastnosť, ktorú budú mať všetky elementy spoločnú, definujeme ju pre element *body*.

2.1.5. Komentáre

Pokiaľ si chceme k štýlu písať nejaké poznámky pre lepšiu orientáciu, píšeme ich do komentárov. Tie sa v CSS tvoria pomocou */** a **/*. Medzi hviezdičky potom môžeme umiestniť aj niekoľkoriadkový komentár, ten sa samozrejme vo výslednom zobrazení neobjaví.

```
body {  
  color: blue; /* komentár: všetky texty budú modré*/  
}
```

2.1.6. Pripojenie štýlu k dokumentu

Štýl môžeme k dokumentu pripojiť rôznymi spôsobmi, môžeme definovať priamo v dokumente alebo v externom súbore, spôsoby môžeme aj kombinovať.

2.1.7. Externý súbor

Pokiaľ chceme mať štýl uložený v externom súbore (čož je veľmi výhodné pri používaní jedného štýlu pre viac dokumentov), v nejakom textovom editore uložíme nami definovaný štýl do súboru s príponou `css`. Ten potom pripojíme k dokumentu zápisom v hlavičke (t.j. medzi tagmi `<head>` a `</head>`) buď v tagu *link*:

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

alebo v tagu *style*:

```
<style type="text/css">@import "styl.css";</style>
```

Pokiaľ je štýl umiestnený na inom serveri, tak môžeme použiť zápis:

```
<style type="text/css">@import url("http://www.neco.cz/styl.css");</style>
```

Zápisom `@import "styl.css";` môžeme tiež vložiť jeden štýl do druhého štýlu.

Uloženie formátovacích informácií do samostatného súboru je výhodné aj na zníženie množstva prenášaných dát zo servera smerom ku klientom, pretože ak sú formátovacie dáta súčasťou každého webového dokumentu, sú prenášané vždy, keď je dokument vyžiadaný. V prípade ich uloženia do samostatného súboru

sú prenesené raz, následne sa čítajú z vyrovnávacej pamäte počítača používateľa. V niektorých prípadoch sa takto darí znížiť množstvo prenášaných dát o desiatky percent.

Definovanie štýlu vnútri dokumentu

To môžeme vykonať opäť v tagu *style* kam tentokrát namiesto odkazu na externý štýl umiestníme priamo definíciu štýlu.

```
<style type="text/css">body {color: blue}</style>
```

Alebo môžeme definovať štýl priamo nejakému elementu, čo sa hodí zvlášť v prípade, kde máme definovaný jednotný štýl, ale pre napríklad jedno konkrétne slovo chceme použiť iné pravidlo. Potom použijeme v príslušnom tagu atribút *style*.

```
<h1 "style=color: green">nadpis</h1>
```

2.1.8. Váha štýlov

Pokiaľ v štýle definujeme pre rovnaký element rovnakú vlastnosť dvakrát, vyššiu váhu má deklarácia, ktorá bola definovaná neskôr (myslené na neskoršom riadku) a tá sa tiež vykoná. Pokiaľ by sme chceli niektorú deklaráciu priradiť väčšiu dôležitosť, použijeme *!important*.

```
h1 {  
    color: blue !important  
}
```

Poznámka: staršie prehliadače štýly vôbec nepodporujú. Pokiaľ teda používate zápis pomocou tagu *style* a chcete tento prípad ošetriť, stačí celý štýl vložiť do komentárov: *<!-- tu bude definovaný štýl -->*.

2.2. Farby

2.2.1. Použitie

Na stránke sa farba používa na text, odkazy, pozadie a ohraničenie. Farbiť môžeme buď celý dokument selektorom *body* alebo jednotlivé elementy.

Farbu pozadia volíme tak, aby text bol dobre čitateľný, teda kontrastný k textu, farba by nemala byť príliš jasná, vyslovene nevhodné je použitie jasno červenej, opatrne aj s modrou. Signálne farby sú dobré pre zvýraznenie textu alebo odkazov. Pokiaľ predpokládame, že užívateľia si budú stránku tlačiť, je treba dať pozor na podobnú sýtosť farieb (aby sa farby „nezliali“).

2.2.2. Vznik farieb

Pre zobrazenie farieb na monitore sa používa systém farieb **RGB** (Red Green Blue). Každá farba na monitore sa skladá z troch základných farieb (červenej, zelenej a modrej) v rôznom pomere. Farby sa vytvárajú aditívne – pridávaním k čiernej, sčítaním farieb. Čím viac farieb, tým bližšie k bielej. Čím menej farieb, tým bližšie k čiernej. Odstiene šedej farby vznikajú, keď sú všetky tri zložky majú rovnakú hodnotu.

2.2.3. Zápis farby

Pre zápis farby môžeme použiť:

- Názov farby (v anglickom jazyku)

```
body {color: blue}
```

Poznámka: Pri použití zápisu menom farby sú validné iba [základné farby](#) (16 VGA farieb).

- RGB zápis:

- Percentuálne

Intenzitu každej farby určujeme percentami 0% (minimálna intenzita) až 100% (maximálna intenzita)

```
body {color: rgb(100%,0%,0%)}
```

- Desiatkovo

Čísla 0 až 255

```
body {color: rgb(255,0,0)}
```

- šestnástkovo

00 až FF (t.j. 0-9, a-f), zápis začína hash-markom #

```
body {color: #FF0000}
```

Pokiaľ sú obe číslice odpovedajúce jednej farebnej zložke rovnaké, môžeme použiť skrátený zápis

```
body {color: #F00}
```

2.2.4. Farbenie dokumentu

Vo väčšine elementov môžeme farbiť pozadie pomocou vlastnosti *background-color*. Napríklad:

```
body {background-color: #00FF00}
```

Text farbíme vlastnosťou *color*. Napríklad:

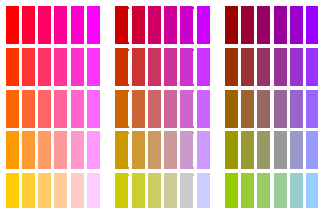
```
h1 {color: #FFFFFF}
```

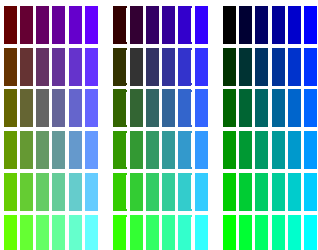
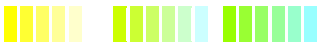
Pokiaľ nedefinujeme žiadne farby, budú použité pôvodné farby prehliadača, t.j. transparentná pre pozadie, čierna pre text, modrá pre odkazy, fialová pre navštívené odkazy a červená pre aktívne odkazy.

2.2.5. Bezpečné farby

Bezpečné farby sú také farby, ktoré by mal podporovať akýkoľvek WWW klient na akomkoľvek operačnom systéme, podporujúci aspoň 256 farieb. Inými slovami, pokiaľ použijete na svojich stránkach tieto farby, máte zaručené, že každý, kto si vaše stránky bude pozerať, si ich bude pozerať vo farbách, v akých ste ich vytvorili. V prípade, že WWW klient nemá k dispozícii vami definovanú farbu, snaží sa ju „poskladať“ z iných farieb (tzv. dithering).

Bezpečných farieb je 216 a 16 odstieňov šedej farby. Sú to farby, pri ktorých hodnota každej zložky (červená, zelená a modrá) nadobúda iba hodnoty 00, 33, 66, 99, CC alebo FF. Pri všetkých teda môžeme použiť zkrátený zápis farby (napr. #F09).





















Poznámka: Pri posunutí kurzoru myši na farbu sa zobrazí jej RGB kód (pokiaľ nemáte vypnuté zobrazovanie popiskov).

2.2.6. Pomenované farby

Základných 16 VGA farieb

00FFFF		Aqua	000000		Black	0000FF		Blue	FF00FF		Fuchsia
808080		Gray	008000		Green	00FF00		Lime	800000		Maroon
000080		Navy	808000		Olive	800080		Purple	FF0000		Red
C0C0C0		Silver	008080		Teal	FFFFFF		White	FFFF00		Yellow

2.3. Písmo

[Rodina písma](#) — [Štýl písma](#) — [Varianta písma](#) — [Veľkosť písma](#) — [Duktus písma](#)

Písmo na stránke by malo byť predovšetkým čitateľné. Je zbytočné, používať kdejaké zložité nepodporované písmo, keď si potom návštevník text nebude môcť ani prečítať. Čím jednoduchšie a obyčajnejšie písmo bude, tým lepšie. Je tiež zbytočné, používať rôzne písmo v rôznych častiach webu.

2.3.1. Rodina písma

Na určenie rodiny písma slúži vlastnosť *font-family*. V nej definujeme názvom písma aké konkrétne písmo bude použité (napr. Arial). Ďalej definujeme písmo, ktoré bude použité v prípade, že prvé definované písmo nemá prehliadač k dispozícii. Takto môžeme určiť aj niekoľko alternatívnych písiem. Pre prípad, že by prehliadač nemal k dispozícii žiadné z určených písiem ešte použijeme obecnú rodinu (tu by sme mali použiť vždy a to na konci výpisu písiem).

serif

pätkové písmo (napr. Times New Roman)

sans-serif

bezpätkové písmo (napr. Arial, Arial CE, Arial Narrow, Verdana, Helvetica)

monospace

neproporciálne písmo (napr. Courier, Courier New)

cursive

ozdobné kurzíva (napr. *Comic Sans*,)

FANTASY

ozdobné písmo (napr. *WESTERN*)

Celý zápis potom môže vypadáť napr. takto:

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

2.3.2. Štýl písma

Vlastnosť *font-style* určí, či sa bude jednať o písmo normálne, kurzívu, alebo sklonené písmo. Rozdiel medzi kurzívou a skloneným písmom je aj v tom, že kurzíva je jedinečným rezom písma. Sklonené písmo vzniká transformáciou normálneho rezu.

normal

normálne

italic

kurzíva

oblique

sklonené písmo

```
body {font-style: italic}
```

2.3.3. *Varianta písma*

Sú dve varianty písma: normálne písmo a kapitálky. Určujú sa vlastnosťou *font-variant*.

normal

normálne

SMALL-CAPS

kapitálky

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

2.3.4. *Veľkosť písma*

Veľkosť písma určíme vlastnosťou *font-size*. Takto môžeme určiť 7 rôznych veľkostí:

xx-small

x-small

small

medium

large

x-large

xx-large

```
h1 {font-size: large}
```

Veľkosť písma môžeme určiť tiež relatívne vzhľadom k aktuálnej veľkosti písma.

larger

väčšie než aktuálne písmo

smaller

menšie než aktuálne písmo

```
h2 {font-size: larger}
```


Ďalej môžeme veľkosť určiť napr. v pixeloch (*px*) alebo iných jednotkách (*pt, em, ex, cm, mm, in*).

```
p {font-size: 16px}
```

Relatívne vzhľadom k aktuálnej veľkosti písma môžeme veľkosť určiť v percentách.

```
h1 {font-size: 150%}
```

2.3.5. *Duktus písma*

Duktus písma sa určuje vlastnosťou *font-weight*

normal

normálne

bold

tučné

```
h1 {font-weight: bold}
```

Tato vlastnosť môže mať ešte hodnotu *bolder* (o niečo silnejšie písmo ako normálne) a *lighter* (o niečo slabšie ako normálne).

2.4. Zarovnanie

Vertikálne zarovnanie — Horizontálne zarovnanie

2.4.1. Vertikálne zarovnanie

Vertikálne zarovnanie textu sa nastavuje vlastnosťou *vertical-align*. Ako jej hodnota sa používajú *baseline*, *sub*, *super*, *top*, *text-top*, *middle*, *bottom*, *text-bottom*, číslo, percento, *inherit*.

top		
		middle
	bottom	

vertical-align: 30px

text text

text **vertical-align: -150%** text

text **vertical-align: super** text

text **vertical-align: sub** text

2.4.2. Horizontálne zarovnanie

Nastavenie horizontálneho zarovnania riadkových prvkov sa vykoná pomocou vlastnosti *text-align*. Ako jej hodnota sa používajú *left*, *right*, *center*, *justify*, *inherit*.

text-align: left

text-align: right

text-align: center

text-align: justify

2.5. Odkazy

[Farba odkazu](#) — [Ozdobenie textu](#) — [Pseudotriedy](#)

Odkazy by mali byť od okolitého textu odlíšené tak, aby boli na prvý pohľad viditeľné. Návštevník stránok sa totiž na webe orientuje predovšetkým podľa odkazov, text treba ani nečíta, sleduje iba kam smerujú odkazy. Preto je dobré, keď odkazy z okolitého textu jasne vyčnievajú.

2.5.1. Farba odkazu

Defaultne sa odkazy v prehliadači zobrazujú modrou farbou a podtrhnuté. Navštívené odkazy sú fialové a aktívne odkazy sú červené. Pri prechode myši cez odkaz sa jeho vzhľad nijak nemení. Toto všetko ide samozrejme ľahko zmeniť.

Farbu odkazu zmeníme rovnako ako farbu akéhokoľvek iného elementu, teda pomocou vlastnosti *color*.

a {color: green}

2.5.2. Ozdobenie textu

Býva dobrým zvykom odkazy podtrhávať. Nie pre každého užívateľa je farebné odlišenie odkazov od okolitého textu dostačujúci. Ďalším zvykom je nepočíarkovať žiadny iný text. Na definovanie štýlu ozdobenia slúži vlastnosť *text-decoration*. Má vlastnosti *none* (bez ozdobenia), *underline* (podčiarknutie), *overline* (nadčiarknutie), *line-through* (preškrtnutie). Použiť môžeme aj všetky hodnoty naraz.

```
a {text-decoration: underline overline}
```

2.5.3. Pseudotriedy

Pseudotriedy slúžia na odlišné definovanie vlastností odkazov (*link*), navštívených odkazov (*visited*) a aktívnych odkazov (*active*). Zapisujú sa s dvojbodkou za tag *a*.

```
a:visited {color: yellow}
```

Pokiaľ chceme, aby odkazy svoj vzhľad zmenili pri prechode myšou, použijeme *hover*.

```
a:hover {text-decoration: none}
```

Vyzerať to potom môže treba takto: [odkaz](#).

2.6. Zoznamy

[Odrážky a číslovanie](#) — [Umiestnenie odrážky](#) — [Obrázok ako odrážka](#)

Zoznamy môžeme v CSS formátovať ako akékoľvek iné elementy, navyše však môžeme určiť druh odrážky číslovaného aj nečíslovaného zoznamu. S použitím obrázku môžeme vytvoriť aj odrážky vlastné.

2.6.1. Odrážky a číslovanie

Druh odrážky aj číslovanie určujeme vlastnosťou *list-style-type*. Túto vlastnosť môžeme priradiť tak selektoru zoznamu *ul* alebo *ol*, ako aj položke zoznamu *li*.

```
ul {list-style-type: square}
```

2.6.2. Nečíslované seznamy

- *disc* – vyplnené koliečko v IE a Opere; vyplnený kosoštvorec v Mozille a Netscape
- *circle* – prázdne koliečko v IE a Opere; prázdny kosoštvorec v Mozille a Netscape
- *square* – štvorec vyplnený

2.6.3. Číslované seznamy

1. *decimal* – arabské číslice (1.)
 - ii. *lower-roman* – malé rímske číslice (i.)
 - III. *upper-roman* – veľké rímske číslice (I.)
 - d. *lower-alpha* – malé písmená (a.)
 - E. *upper-alpha* – veľké písmená (A.)
- *none* – bez odrážky

2.6.4. *Dalšie možnosti číslovania zoznamov, ktoré ale nefungujú vo všetkých prehliadačoch*

1. *decimal-leading-zero* – arabské číslice, pri jednomiestných číslach začínajúce nulou (01.); nefunguje v IE a Opere
2. *lower-greek* – malé písmená grckej abecedy (α .); nefunguje v IE
3. *lower-latin* – malé písmená (a.); nefunguje v IE
4. *upper-latin* – veľké písmená (A.); nefunguje v IE
5. *georgian* – nefunguje asi nikde
6. *armenian* – arménština, nefunguje v IE, v Opere iba pokiaľ je dostupný font

2.6.5. *Umiestnenie odrážky*

Odrážku umiestňujeme pomocou vlastnosti *list-style-position*. Môže byť buď vnútri textu, tomu odpovedá hodnota *inside*, alebo zvonka textu s hodnotou *outside*.

2.6.6. *Obrázok ako odrážka*

Ako odrážku môžeme tiež použiť obrázok. Vlastnosť *list-style-image* má ako hodnotu URL obrázka.

```
ul {list-style-image: url("adresar/obrazok.gif");}
```

2.7. Rámčeky a ohraničenie

[Možnosti ohraničenia](#) — [Bez ohraničenia](#) — [Samostatný zápis](#)

Ohraničenie elementov tvorí dosť často veľkú časť designu. Pomocou ohraničenia sa vytvárajú rôzne zvislé a vodorovné čiary, rámčeky pre obrázky a podobne.

Možnosti ohraničenia

Spôsob ohraničenia elementov určuje vlastnosť **border** – pre ohraničenie elementu „zo všetkých strán“ a vlastnosti **border-top**, **border-right**, **border-bottom** a **border-left** pre ohraničenie zhora, zprava, zdola a zľava. Tieto vlastnosti majú tri hodnoty:

hrúbka ohraničenia

napr. 1px

štýl ohraničenia

- *solid* – jednoduchá čiara
- *double* – dvojité čiara
- *dotted* – bodkovaná čiara

farba ohraničenia

napr. black (viac vid'. [Farby v CSS](#))

```
border-top: 2px solid green;
```

Bez ohraničenia

Hodnotou vlastnosti **border** môže byť tiež *none*. Element teda nebude mať žiadne ohraničenie (čo je štandardné nastavenie). Túto vlastnosť využijeme napríklad pokiaľ máme obrázkový odkaz, keď sa okolo obrázku vytvorí spravidla modrý rámček.

```
img {border: none}
```

Samostatný zápis

Jednotlivé hodnoty vlastnosti *border* môžeme zapisovať tiež zvlášť pomocou vlastností **border-width** (hrúbka ohraničenia), **border-style** (štýl ohraničenia) a **border-color** (farba ohraničenia). Takto rozpísaný zápis bude totožný z prvým zápisom.

```
border-width: 2px;  
border-style: solid;  
border-color: green;
```

Rozpísanie po jednotlivých vlastnostiach je vhodné napr. v prípade, že chceme všetky nadpisy ohraničiť tenkou čiarou, ale každý inou farbou. Potom definujeme hrúbku a štýl ohraničenia pre všetky nadpisy naraz a farbu definujeme zvlášť pre každú úroveň nadpisu.

2.8. Pozadie

Pozadie určené farbou — Obrázok ako pozadie

Pozadie elementu a celej stránky môžeme vytvoriť buď prostým určením farby pozadia alebo použitím obrázku.

Pozadie určené farbou

Pre vytvorenie farebného pozadia použijeme CSS vlastnosť **background-color**. Jej hodnotou bude požadovaná farba (viac o farebných hodnotách viď. [farby](#))

```
body {background-color:#aaa}
```

Obrázok ako pozadie

Pozadie elementu a stránky môže byť akýkoľvek obrázok. Je však nutné mať na zreteli, že je iba pozadím a dá sa predpokladať, že cez neho bude nejaký text, ktorý musí byť čitateľný, a to vo všetkých častiach obrázku.

Samotný obrázok určíme vlastnosťou **background-image**, jej hodnotou je cesta k obrázku, zapísaná v zátvorke za parametrom *url*.

```
body {background-image:url(obrazky/pozadie.gif)}
```

Opakovanie obrázku

Obrázok na pozadí sa môže opakovať (napr. ako nejaká vzorka) alebo môže byť umiestnený iba raz. Štandardne sa obrázok opakuje v oboch smeroch (x a y) až do zaplnenia celej plochy elementu.

Pre určenie spôsobu opakovania sa používa vlastnosť **background-repeat**. Pokiaľ chceme, aby sa obrázok opakoval len v smere *x* (t.j. horizontálne), použijeme hodnotu *repeat-x*, v smere *y* (t.j. vertikálne) *repeat-y*. Pokiaľ nechceme, aby sa obrázok opakoval (napr. ak ide o logo či iný grafický prvok, ktorý nemá vytvárať vzorku), použijeme hodnotu *no-repeat*.

```
body {background-repeat:repeat-y}
```

Pri opakovaní obrázku je vhodné použiť obrázok, ktorý „sám na seba nadväzuje“.

Umiestnenie pozadia (obrázku)

Pokiaľ chceme obrázok umiestniť inak, než s začiatočnou pozíciou (t.j. ľavý horný roh elementu), použijeme vlastnosť **background-position**. Umiestnenie určuje horizontálna a vertikálna hodnota. Hodnoty horizontálneho umiestnenia môžu byť *left* (vľavo), *center* (na stred) a *right* (vpravo). Hodnoty vertikálneho umiestnenia sú *top* (hore), *center* (na stred) a *bottom* (dole).

```
h1 {background-position:left center}
```

2.9. Triedy a identifikátory

Triedy a identifikátory v CSS slúžia k tomu, aby sme mohli rôzne elementy formátovať rôzne. Napríklad odkazy na stránke. Každý z nás asi chce mať na stránke rôzne druhy odkazov, nie iba jeden. Inak sa obvykle robia odkazy v menu, inak odkazy v texte.

2.9.1. Trieda *class* v CSS

Triedy vytvoríme ľahko tak, že k elementu v HTML pridáme atribut *class*. Jeho hodnotou bude nejaký reťazec písmen, rovnaký potom budeme používať v CSS štýle ako selektor.

```
<p class="poznamka">Nejaký text</p>
```

Týmto hovoríme, že tento odstavec bude formátovaný podľa pravidiel triedy *poznamka*, na formátovanie ostatných odstavcov sa tieto pravidlá neprejavia. Teraz musíme ešte tie pravidlá určiť v CSS štýle.

```
.poznamka {font-size: x-small; color: black}
```

Teraz teda budeme mať všetky odstavce rovnaké, iba odstavec s triedou *poznamka* bude vypadáť inak (malým čiernym písmom). Resp. inak budú vypadáť všetky odstavce s triedou *poznamka*, pretože rovnakú triedu môžeme použiť pri ľubovoľnom množstve elementov. Dokonca aj pri rôznych elementoch.

```
<p class="poznamka">Nejaký odstavec</p>  
<li class="poznamka">Položka zoznamu</li>
```

```
.poznamka {color: black}  štýl sa aplikuje na všetky elementy s triedou poznamka  
/* štýl sa aplikuje na všetky elementy s triedou poznamka */
```

```
li.poznamka {color: blue}
```

```
/* štýl sa aplikuje iba na elementy li s triedou poznamka */
```

Dedičnosť

Trieda každého elementu bude dediť všetky vlastnosti daného elementu a navyše bude mať svoje vlastné, teda napr.:

```
p {text-align: center; color: blue}  
.poznamka {font-size: x-small; color: black}
```

Týmto zápisom je naznačené, že všetky odstavce budú zarovnané na stred a ich text bude mať modrú farbu (prvá deklarácia). Toto obecné platí pre všetky odstavce stránky. Odstavec s triedou *poznamka* bude mať navyše ešte menšie písmo (druhá deklarácia). Farba je určená v oboch deklaráciach, v takomto prípade má vyššiu prioritu deklarácia zadaná neskôr, teda odstavce s triedou *poznamka* bude mať čierny text. Všetky odstavce teda budú modré, zarovnané na stred, okrem odstavca s triedou *poznamka*, ten bude čierny, s malým písmom, zarovnaný bude tiež na stred.

Kontextová deklarácia s triedou

```
<p class="poznamka"><a href=" ... ">Odkaz</a> Nejaký text</p>  
a.poznamka {color: red}
```

Možnosť kontextovej deklarácie platí aj pre triedu. Tento zápis hovorí, že odkazy v odstavci triedy *poznamka* budú červené.

2.9.2. Identifikátor id v CSS

Identifikátor sa od triedy líši tým, že sa jedná vždy o **jednoznačný identifikátor**. To znamená, že ho na každej stránke môžeme použiť iba raz. Oproti tomu triedu sme mohli použiť viackrát na každej stránke webu.

Identifikátory sa teda používajú práve tam, kde je isté, že sa daný element objaví v stránke iba raz. Ideálne sa teda hodí pre veci ako je box celej stránky, menu, záhlavie alebo zápatie. Identifikátory sa označujú dvojkřížkom (#). Inak je ich zápis rovnaký ako zápis triedy.

```
<div id="menu"> ... </div>
#menu {width:14em; background-color:black}
#menu a {color: white}
```

2.9.3. Absolútna a relatívna pozícia

```
<html>
<head>
  <style>
    #lavy {
      position:absolute;
      width:176px;
      top:0px;
      left:0PX;
      height:100%;
      float: left;
      background-color: #c0c0c0;
    }
    #stred {
      position:absolute;
      top:0px;
      width:550px;
      height:100%;
      left:177px;
      background-color: #ffff99;
    }
  </style>
```

```
</head>
<body>

<!-- Menu -->
<div id="lavy">
<a href="linka1.html" >LINKA 1</a><br />
<a href="linka1.html" >LINKA 1</a><br />
<a href="linka1.html" >LINKA 1</a><br />
</div>
<!-- Menu -->

<!-- Hlavna stranka -->
<div id="stred">
<p>Text Text Text</p>
</div>
<!-- Hlavna stranka -->

</body>
</html>
```

2.10. CSS vlastnosti - stručný prehľad

[Vlastnosti písma](#) — [Vlastnosti farby a pozadia](#) — [Vlastnosti textu](#) — [Vlastnosti boxů](#) — [Klasifikační vlastnosti](#) — [Vlastnosti pozicování](#)

Tu nájdete kompletný prehľad vlastností [CSS1](#) a niektoré vlastnosti a hodnoty [CSS2.1](#) (označené *). V tabuľkách je zapísaný názov CSS vlastnosti a jej opis, možné hodnoty (tučne vyznačená hodnota je pre danú vlastnosť implicitná), význam hodnoty, elementu, pre ktorý je možné danú vlastnosť použiť a dedičnosť vlastnosti.

Vlastnosti písma

Vlastnosť	Hodnoty	Hodnoty	Elementy	Dedenie
font-family	<ul style="list-style-type: none"><family-name><generic-family>	<ul style="list-style-type: none">meno rodiny písmaobecná rodina písma	všetky	áno
rodina písma				
font-style	<ul style="list-style-type: none">normalitalicoblique	<ul style="list-style-type: none">normálnekurzívasklonené	všetky	áno
štýl písma				
font-variant	<ul style="list-style-type: none">normalsmall-caps	<ul style="list-style-type: none">normálnekapitálky	všetky	áno
varianta písma				
font-weight	<ul style="list-style-type: none">normalboldbolder	<ul style="list-style-type: none">normálnetučnétučnejšie	všetky	áno
duktus písma				

Vlastnosť	Hodnoty	Hodnoty	Elementy Dedenie	
	<ul style="list-style-type: none"> • lighter • 100 200 300 400 • 500 600 700 800 900 	<ul style="list-style-type: none"> • slabšie • číselné vyjadrenie 		
font-size	<ul style="list-style-type: none"> • <absolute-size> • <relative-size> • <length> • <percentage> 	<ul style="list-style-type: none"> • absolutna veľkosť • relatívna veľkosť • dĺžka • percentuálne zväčšenie 	všetky	áno
veľkosť písma				
font	<ul style="list-style-type: none"> • <font-style> • <font-variant> • <font-weight> • <font-size> • <font-family> • caption * • icon * • menu * • message-box * • small-caption * • status-bar * 	<ul style="list-style-type: none"> • štýl písma • varianta písma • duktus písma • veľkosť písma • rodina písma • pre ovládacie prvky • pre označenie ikon • pre menu • pre dialogové okna • pre malé ovládacie prvky • pre stavový riadok 	všetky	áno
písmo				

Vlastnosti farby a pozadia

Vlastnosť	Hodnoty	Hodnoty	Elementy	Dedenie
-----------	---------	---------	----------	---------

Vlastnosť	Hodnoty	Hodnoty	Elementy	Dedenie
color farba	<ul style="list-style-type: none"> • <color> 	<ul style="list-style-type: none"> • farba 	všetky	áno
background-color farba pozadí	<ul style="list-style-type: none"> • <color> • transparent 	<ul style="list-style-type: none"> • farba • priehľadná 	všetky	nie
background-image obrázok na pozadí	<ul style="list-style-type: none"> • <url> • none 	<ul style="list-style-type: none"> • URL obrázku • žiadny 	všetky	nie
background-repeat smer opakovania pozadia	<ul style="list-style-type: none"> • repeat • repeat-x • repeat-y • no-repeat 	<ul style="list-style-type: none"> • opakovanie všetkými smermi • opakovanie v smere x • opakovanie v smere y • bez opakovania 	všetky	nie
background-attachment pripevnenie pozadia	<ul style="list-style-type: none"> • scroll • fixed 	<ul style="list-style-type: none"> • roluje sa • je fixné 	všetky	nie

Vlastnosť	Hodnoty	Hodnoty	Elementy	Dedenie
background-position umiestnenie obrázku na pozadí	<ul style="list-style-type: none"> • <percentage> • <length> • top • center • bottom • left • center • right 	<ul style="list-style-type: none"> • percentuálne • dĺžka • hore • vertikálny stred • dole • vľavo • horizontálny stred • vpravo 	<ul style="list-style-type: none"> • blokové • nahradzované 	nie
background pozadie	<ul style="list-style-type: none"> • <background-color> • <background-image> • <background-repeat> • <background-attachment> • <background-position> 	<ul style="list-style-type: none"> • farba pozadia • obrázok na pozadí • opakovanie pozadia • pripevnenie pozadia • umiestnenie obrázku na pozadí 	všetky	nie

Vlastnosti textu

Vlastnosť	Hodnoty	Hodnoty	Elementy Dedenie	
word-spacing	<ul style="list-style-type: none"> • normal • <length> 	<ul style="list-style-type: none"> • bez zväčšenia • zväčšenie o dĺžku 	všetky	áno
zväčšenie medzislovnej medzery				
letter-spacing	<ul style="list-style-type: none"> • normal • <length> 	<ul style="list-style-type: none"> • bez zväčšenia • zväčšenie o dĺžku 	všetky	áno
<ul style="list-style-type: none"> • zväčšenie medzery medzi znaky 				
text-decoration	<ul style="list-style-type: none"> • none • underline • overline • line-through • blink 	<ul style="list-style-type: none"> • žiadne • podtrhnutie • nadtrhnutie • preškrtnutie • blikanie 	všetky	nie
ozdobenie textu				
vertical-align	<ul style="list-style-type: none"> • baseline • sub • super • top • text-top • middle • bottom • text-bottom 	<ul style="list-style-type: none"> • na riadok • horný index • dolný index • k najvyššiemu elementu • na vrch rodičovského • na stred rodičovského • na spodok rodičovského 	inline	nie
vertikálne zarovnávanie				

Vlastnosť	Hodnoty	Hodnoty	Elementy Dedenie	
	<ul style="list-style-type: none"> <percentage> 	<ul style="list-style-type: none"> k najnižšiemu elementu percentuálne 		
text-transform transformace textu	<ul style="list-style-type: none"> capitalize uppercase lowercase none 	<ul style="list-style-type: none"> prvé písmena slov veľké všetky na veľké všetky na malé žiadna zmena 	všetky	áno
text-align zarovnávanie	<ul style="list-style-type: none"> left right center justify 	<ul style="list-style-type: none"> vľavo vpravo na stred do bloku 	blokové	áno
text-indent odsadenie 1. riadku odstavca	<ul style="list-style-type: none"> <length> <percentage> 	<ul style="list-style-type: none"> dĺžka percentuálne 	blokové	áno
line-height výška riadku	<ul style="list-style-type: none"> normal <number> <length> <percentage> 	<ul style="list-style-type: none"> normálne číslo dĺžka percentuálne 	všetky	áno

Vlastnosti boxov

Vlastnosť	Hodnoty	Hodnoty	Elementy	Dedenie
margin-top	<ul style="list-style-type: none">• <length>• <percentage>	<ul style="list-style-type: none">• dĺžka• percentuálne	všetky	nie
vonkajší horný okraj	<ul style="list-style-type: none">• auto	<ul style="list-style-type: none">• automaticky		
margin-right	<ul style="list-style-type: none">• <length>• <percentage>	<ul style="list-style-type: none">• dĺžka• percentuálne	všetky	nie
vonkajší pravý okraj	<ul style="list-style-type: none">• auto	<ul style="list-style-type: none">• automaticky		
margin-bottom	<ul style="list-style-type: none">• <length>• <percentage>	<ul style="list-style-type: none">• dĺžka• percentuálne	všetky	nie
vonkajší spodný okraj	<ul style="list-style-type: none">• auto	<ul style="list-style-type: none">• automaticky		
margin-left	<ul style="list-style-type: none">• <length>• <percentage>	<ul style="list-style-type: none">• dĺžka• percentuálne	všetky	nie
vonkajší ľavý okraj	<ul style="list-style-type: none">• auto	<ul style="list-style-type: none">• automaticky		
margin	<ul style="list-style-type: none">• <length>• <percentage>	<ul style="list-style-type: none">• dĺžka• percentuálne	všetky	nie
vonkajší okraj	<ul style="list-style-type: none">• auto	<ul style="list-style-type: none">• automaticky		
padding-top	<ul style="list-style-type: none">• <length>• <percentage>	<ul style="list-style-type: none">• dĺžka• percentuálne	všetky	nie
vnútorný horný				

Vlastnosť	Hodnoty	Hodnoty	Elementy	Dedenie
okraj padding-right	<ul style="list-style-type: none"> • <length> • <percentage> 	<ul style="list-style-type: none"> • dĺžka • percentuálne 	všetky	nie
vnútorný pravý okraj padding-bottom	<ul style="list-style-type: none"> • <length> • <percentage> 	<ul style="list-style-type: none"> • dĺžka • percentuálne 	všetky	nie
vnútorný spodný okraj padding-left	<ul style="list-style-type: none"> • <length> • <percentage> 	<ul style="list-style-type: none"> • dĺžka • percentuálne 	všetky	nie
vnútorný ľavý okraj padding	<ul style="list-style-type: none"> • <length> • <percentage> 	<ul style="list-style-type: none"> • dĺžka • percentuálne 	všetky	nie
vnútorný okraj border-top-width	<ul style="list-style-type: none"> • thin • medium • thick • <length> 	<ul style="list-style-type: none"> • tenká • stredná • tlstá • dĺžka 	všetky	nie
šírka horného rámčeka border-right-width	<ul style="list-style-type: none"> • thin • medium • thick 	<ul style="list-style-type: none"> • tenká • stredná • tlstá 	všetky	nie
šírka pravého				

Vlastnosť	Hodnoty	Hodnoty	Elementy	Dedenie
rámčeka	<ul style="list-style-type: none"> • <length> 	<ul style="list-style-type: none"> • dĺžka 		
border-bottom-width	<ul style="list-style-type: none"> • thin • medium • thick • <length> 	<ul style="list-style-type: none"> • tenká • stredná • tlstá • dĺžka 	všetky	nie
šírka spodného rámčeka				
border-left-width	<ul style="list-style-type: none"> • thin • medium • thick • <length> 	<ul style="list-style-type: none"> • tenká • stredná • tlstá • dĺžka 	všetky	nie
šírka ľavého rámčeka				
border-width	<ul style="list-style-type: none"> • thin • medium • thick • <length> 	<ul style="list-style-type: none"> • tenká • stredná • tlstá • dĺžka 	všetky	nie
šírka rámčeka				
border-color	<ul style="list-style-type: none"> • <color> 	<ul style="list-style-type: none"> • farba 	všetky	nie
farba rámčeka				
border-style	<ul style="list-style-type: none"> • none • dotted • dashed • solid 	<ul style="list-style-type: none"> • žiadny • bodkovaný • čiarkovaný • jednoduchý 	všetky	nie
štýl rámčeka				

Vlastnosť	Hodnoty	Hodnoty	Elementy	Dedenie
	<ul style="list-style-type: none"> • double • groove • ridge • inset • outset 	<ul style="list-style-type: none"> • dvojitý • ryha • chrbát • vsadený • vysadený 		
border-top horný rámček	<ul style="list-style-type: none"> • <border-top-width> • <border-style> • <color> 	<ul style="list-style-type: none"> • šírka horného rámčeka • štýl rámčeka • farba 	všetky	nie
border-right pravý rámček	<ul style="list-style-type: none"> • <border-right-width> • <border-style> • <color> 	<ul style="list-style-type: none"> • šírka pravého rámčeka • štýl rámčeka • farba 	všetky	nie
border-bottom spodný rámček	<ul style="list-style-type: none"> • <border-bottom-width> • <border-style> • <color> 	<ul style="list-style-type: none"> • šírka spodného rámčeka • štýl rámčeka • farba 	všetky	nie
border-left	<ul style="list-style-type: none"> • <border-left-width> • <border-style> 	<ul style="list-style-type: none"> • šírka ľavého rámčeka 	všetky	nie

Vlastnosť	Hodnoty	Hodnoty	Elementy	Dedenie
ľavý rámček	<ul style="list-style-type: none"> • <color> 	<ul style="list-style-type: none"> • štýl rámčeka • farba 		
border	<ul style="list-style-type: none"> • <border-width> • <border-style> • <color> 	<ul style="list-style-type: none"> • šírka rámčeka • štýl rámčeka • farba 	všetky	nie
rámček				
width	<ul style="list-style-type: none"> • <length> • <percentage> • auto 	<ul style="list-style-type: none"> • dĺžka • percentuálne • automaticky 	<ul style="list-style-type: none"> • blokové • nahradzované 	nie
šírka				
height	<ul style="list-style-type: none"> • <length> • auto 	<ul style="list-style-type: none"> • dĺžka • automaticky 	<ul style="list-style-type: none"> • blokové • nahradzované 	nie
výška				
float	<ul style="list-style-type: none"> • left • right • none 	<ul style="list-style-type: none"> • vľavo • vpravo • žiadny efekt 	všetky	nie
plávajúci objekt				
clear	<ul style="list-style-type: none"> • none • left • right • both 	<ul style="list-style-type: none"> • žiadne • zľava • zprava • z oboch strán 	všetky	nie
čakanie na skončenie plávajúcich objektov				

Klasifikačné vlastnosti

Vlastnosť	Hodnoty	Hodnoty	Elementy	Dedenie
display	<ul style="list-style-type: none"> • block • inline • list-item 	<ul style="list-style-type: none"> • blokový • inline 	všetky	nie
druh elementu	<ul style="list-style-type: none"> • none 	<ul style="list-style-type: none"> • položka zoznamu • nezobrazuje sa 		
white-space	<ul style="list-style-type: none"> • normal • pre • nowrap 	<ul style="list-style-type: none"> • normálne • predformátovaný text • nezalamuje medzery 	blokové	áno
medzery				
list-style-type	<ul style="list-style-type: none"> • disc • circle • square • decimal • decimal-leading-zero • lower-roman • upper-roman 	<ul style="list-style-type: none"> • plné kolečko • prázdne kolečko • štvorček • číslo • číslo doplnené nulou • malé rímske číslice • veľké rímske číslice 	display:list-item	áno
typ odrážky zoznamu	<ul style="list-style-type: none"> • lower-greek • lower-alpha • upper-alpha • armenian • georgian • none 	<ul style="list-style-type: none"> • malé grecké písmeno • malé písmeno • veľké písmeno • arménské • georgiánské • bez odrážky 		

Vlastnosť	Hodnoty	Hodnoty	Elementy	Dedenie
list-style-image	<ul style="list-style-type: none"> • <url> • none 	<ul style="list-style-type: none"> • URL obrázku odrážky • žiadny 	display:list-item	áno
obrázok odrážky zoznamu				
list-style-position	<ul style="list-style-type: none"> • inside • outside 	<ul style="list-style-type: none"> • odrážka vsadená do textu • odrážka predsadená 	display:list-item	áno
umiestnenie odrážky zoznamu				
list-style	<ul style="list-style-type: none"> • disc • circle • square • decimal • lower-roman • upper-roman • lower-alpha • upper-alpha • none • inside • outside • <url> 	<ul style="list-style-type: none"> • plné kolečko • prázdne kolečko • štvorček • číslo • malé písmeno • veľké písmeno • malé rímske číslice • veľké rímske číslice • malé grecké písmeno • veľké grecké písmeno • bez odrážky, bez obrázku • odrážka vsadená do textu • odrážka predsadená 	display:list-item	áno
odrážka zoznamu				

Vlastnosť	Hodnoty	Hodnoty	Elementy	Dedenie
		<ul style="list-style-type: none"> URL obrázku odrážky 		
<i>Vlastnosti polohovania</i>				
Vlastnosť	Hodnoty	Hodnoty	Elementy	Dedenie
position *	<ul style="list-style-type: none"> static relative absolute fixed 	<ul style="list-style-type: none"> normálne relatívne absolútne fixné 	všetky	nie
umiestnenie elementu				
top *	<ul style="list-style-type: none"> <length> <percentage> auto 	<ul style="list-style-type: none"> dĺžka percentuálne bez posunu 	polohované	nie
posun zhora o				
right *	<ul style="list-style-type: none"> <length> <percentage> auto 	<ul style="list-style-type: none"> dĺžka percentuálne bez posunu 	polohované	nie
posun zprava o				
bottom *	<ul style="list-style-type: none"> <length> <percentage> auto 	<ul style="list-style-type: none"> dĺžka percentuálne bez posunu 	polohované	nie
posun zdola o				
left *	<ul style="list-style-type: none"> <length> <percentage> 	<ul style="list-style-type: none"> dĺžka percentuálne 	polohované	nie
posun zľava o				

Vlastnosť	Hodnoty	Hodnoty	Elementy	Dedenie
	<ul style="list-style-type: none"> • auto 	<ul style="list-style-type: none"> • bez posunu 		
width * šírka elementu	<ul style="list-style-type: none"> • <length> • <percentage> • auto 	<ul style="list-style-type: none"> • dĺžka • percentuálne • automatická 	<ul style="list-style-type: none"> • blokové • nahradzované 	nie
height * výška elementu	<ul style="list-style-type: none"> • <length> • <percentage> • auto 	<ul style="list-style-type: none"> • dĺžka • percentuálne • automatická 	<ul style="list-style-type: none"> • blokové • nahradzované 	nie
overflow * spôsob zobrazenia elementov, ktoré sa nevojdú do svojich rozmerov	<ul style="list-style-type: none"> • visible • hidden • scroll • auto 	<ul style="list-style-type: none"> • pretečenie elementu • skrytie pretečeného • pridanie scroll-barov vždy • pridanie scroll-barov iba pokiaľ skutočne pretečie 	<ul style="list-style-type: none"> • blokové • nahradzované • bunky • inline bloky 	nie
clip * definícia viditeľnej časti	<ul style="list-style-type: none"> • <shape> • auto 	<ul style="list-style-type: none"> • zadanie vrcholov • normálne zobrazenie 	<ul style="list-style-type: none"> • position:absolute 	nie

Vlastnosť elementu	Hodnoty	Hodnoty	Elementy	Dedenie
visibility*	<ul style="list-style-type: none"> • visible • hidden • collapse 	<ul style="list-style-type: none"> • viditeľný • skrytý 	<ul style="list-style-type: none"> • všetky 	áno
viditeľnosť elementu				
z-index*	<ul style="list-style-type: none"> • auto • <integer> 	<ul style="list-style-type: none"> • normálne • číslo 	<ul style="list-style-type: none"> • polohované 	nie
umiestnenie elementu na osi z				

2.11. Úlohy

1. Formátujte stránku:

Formátovanie pre menu:

- Menu má šírku 300px;
- Položky menu sú odsadené 50px zhora a 10px zľava;
- Odkazy sú modrej farby, tučne a podčiarknuté;
- Ak je kurzor myši nad odkazom, zmení sa na červene písmo bez podčiarknutia;
- Farba pozadia je #E1E1FF;
- Písmo je Verdana, 10pt.

Formátovanie pre obsah:

- Písmo je Verdana
- Texty sú odsadené z každej strany o 10px od okraja
- Nadpis je úrovne h1 o veľkosti 12pt
- Texty odsekov majú veľkosť 10pt
- Okolo obrázka je 3px medzera vyplnená farbou #E1E1FF a celé je to ohraničené rámčekom hrúbky 1px šedej farby, okolitý text odsadený o 4px
- Slovo 'Obrázok:' je tučným písmom a celý text opisujúci obrázok je veľkosti 8pt
- Obrázok 180 x 150 px , alternatívny text = obrázok,

NADPIS

[menu1](#)

[menu2](#)

[menu3](#)

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nam convallis lorem et sapien bibendum sollicitudin. Suspendisse potenti. Duis tristique pellentesque justo vel auctor. Pellentesque nibh urna, semper sit amet pharetra sit amet, auctor vel nibh. Duis sit amet turpis quam, nec suscipit leo. Maecenas lacinia nisl eu lorem facilisis sodales. Nunc at velit ut est vehicula ullamcorper ut a massa. Suspendisse eget turpis nibh, quis placerat massa. Donec bibendum eros vel mauris fermentum fermentum et non risus. Praesent in ipsum vitae justo mattis euismod at id magna.



Obrázok

Nulla et pulvinar dui. Maecenas sed scelerisque ipsum. Curabitur eget lacus non leo vulputate sodales et vulputate ipsum. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Nulla dictum ligula in neque vestibulum adipiscing. Duis eu justo sapien, nec adipiscing dolor. Suspendisse vehicula aliquam eros at imperdiet. Proin nulla tellus, pellentesque et congue eu, interdum sed leo. Aliquam quis neque erat, eu semper orci. In sit amet pulvinar odio. Aenean hendrerit tristique lorem id ullamcorper. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Nunc purus sem, sodales non semper a, feugiat non turpis. Aliquam pretium odio vitae orci consectetur vehicula. Duis mattis orci sit amet felis ultricies iaculis. Proin sodales

tincidunt purus, non bibendum ligula tincidunt fringilla. Proin ante arcu, vehicula a commodo id, sollicitudin nec dolor. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Nullam augue nibh, mattis in tincidunt vel, ullamcorper eu purus. Vivamus a quam et felis consequat interdum et vel libero.

Základné požiadavky na web stránky:

- Formátovanie musí byť definované v externom súbore (.css).
 - Stránky musia spĺňať štandardy w3c (www.w3c.org - validácia HTML a CSS)
 -
2. Upravte stránky tak aby neobsahovali tabuľky, pričom vzhľad stránok sa nesmie zmeniť.

Vytvorenie WWW stránky s použitím XHTML a CSS

Ako by mohla vyzeráť jednoduchá web stránka si ukážeme na príklade. Stránka je zámerne veľmi jednoduchá - nie je v nej použitá žiadna grafika. Všetky farby a spôsoby zobrazenia sú dosiahnuté za pomoci použitia jednoduchých CSS štýlov.

Zdrojový tvar XHTML a CSS pre jeho dĺžku neuvádzame, nájdete ho na web stránke k projektu Škola mladého podnikateľa.

Pri praktických cvičeniach si ukážeme, ako sa dá využiť takýto základný návrh pri tvorbe viacerých podstránok. Rozoberieme si štruktúru HTML dokumentu, ukážeme si ako sa zobrazujú jednotlivé prvky na stránke bez štýlov a so štýlmi. Vyskúšame si prepísať štýly, zmeniť texty a upraviť vzhľad stránky.

Tiež si preberieme štýlový predpis CSS a na názornom príklade si ukážeme, aké vlastnosti bolo potrebné zmeniť, aby sa stránka zobrazila v podobe, ako vidíte na priloženom obrázku a v prehliadačoch. Stránka a jej zdroj:

Moja firma

Predaj najlepších výrobkov široko ďaleko.

O firme Ponuka Odkazy Kontakt

O firme

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat.

História firmy

Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt explicabo. Quis autem vel eum iure reprehenderit qui in ea voluptate velit esse quam nihil molestiae consequatur, vel illum qui dolorem eum fugiat quo voluptas nulla pariatur?

ROK	UDALOSŤ
1996	Založenie firmy
1999	Firma sa transformovala na s.r.o.
2001	Firma sa rozšírila do ďalších miest
2004	Firma založila prvú pobočku v zahraničí

Tabuľka z históriou firmy.

Súčasnoscť firmy

At vero eos et accusamus et iusto odio dignissimos ducimus qui blanditiis praesentium voluptatum deleniti atque corrupti quos dolores et quas molestias excepturi sint occaecati cupiditate non provident, similique sunt in culpa qui officia deserunt mollitia animi, id est laborum et dolorum fuga.

Budúcnosť firmy

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.

At vero eos et accusamus et iusto odio dignissimos ducimus qui blanditiis praesentium voluptatum deleniti atque corrupti quos dolores et quas molestias excepturi sint occaecati cupiditate non provident, similique sunt in culpa qui officia deserunt mollitia animi, id est laborum et dolorum fuga.

```
1 PSPad [c:\projekt\moja_firma\index.html]
2 Sukor Projekt Upravy Vyhľadavanie Jazyk Formát Nástroje Skripty HTML Nastavenie Otvor Pomocník
3
4 .. index.html
5
6 12 <body>
7 13 <div id="hlavicka">
8 14 <h1>Moja firma</h1>
9 15 <p>Predaj najlepších výrobkov široko ďaleko.</p>
10 16 </div>
11
12 <div id="menu">
13 <ul>
14 <li><a href="index.html">O firme</a></li>
15 <li><a href="ponuka.html">Ponuka</a></li>
16 <li><a href="odkazy.html">Odkazy</a></li>
17 <li><a href="kontakt.html">Kontakt</a></li>
18 </ul>
19 </div>
20
21 <div id="obsah">
22 <h2>O firme</h2>
23 <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat.
24
25 <h3>História firmy</h3>
26 <p>Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt explicabo. Quis autem vel eum iure reprehenderit qui in ea voluptate velit esse quam nihil molestiae consequatur, vel illum qui dolorem eum fugiat quo voluptas nulla pariatur?
27
28 <table>
29 <tr class="pcp">
30 <td width="44">ROK</td><td>UDALOSŤ</td>
31 </tr>
32 <tr class="neparny">
33 <td>1996</td><td>Založenie firmy</td>
34 </tr>
35 <tr class="parny">
36 <td>1999</td><td>Firma sa transformovala na s.r.o.</td>
37 </tr>
38 <tr class="neparny">
39 <td>2001</td><td>Firma sa rozšírila do ďalších miest</td>
40 </tr>
41 <tr class="parny">
42 <td>2004</td><td>Firma založila prvú pobočku v zahraničí</td>
43 </tr>
44 </table>
45 <p>Tabuľka z históriou firmy.</p>
46
47 <h3>Súčasnoscť firmy</h3>
48 <p>At vero eos et accusamus et iusto odio dignissimos ducimus qui blanditiis praesentium voluptatum deleniti atque corrupti quos dolores et quas molestias excepturi sint occaecati cupiditate non provident, similique sunt in culpa qui officia deserunt mollitia animi, id est laborum et dolorum fuga.
49
50 <h3>Budúcnosť firmy</h3>
51 <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.
52
53 <p>At vero eos et accusamus et iusto odio dignissimos ducimus qui blanditiis praesentium voluptatum deleniti atque corrupti quos dolores et quas molestias excepturi sint occaecati cupiditate non provident, similique sunt in culpa qui officia deserunt mollitia animi, id est laborum et dolorum fuga.
54 </div>
55
56 </body>
57
58 57: 304 (65) [3.115] 46 128 HTML multihighlighter UNIX Kódovanie: Windows (CP 1250)
```

Umiestnenie hotovej WWW stránky na Internete - výber poskytovateľa WWW priestoru a použitie služby FTP pre umiestnenie a úpravy hotovej WWW stránky

Umiestnenie hotovej web stránky na Internete je záverečným krokom pre jej sprístupnenie verejnosti. Pre umiestnenie web stránky potrebujeme web priestor - miesto, kam môžeme vytvorené stránky umiestniť (nahrať) a v prípade budúcich zmien (ktoré určite prídu) ich aj zmeniť. Pre umiestnenie hotovej web stránky existuje viacero možností: poskytovatelia web priestoru zdarma, platení poskytovatelia web priestoru vlastný web server.

Poskytovatelia web priestoru zdarma (free webhosting)

Tento spôsob je z hľadiska ceny za službu (zdarma) najvýhodnejší, ale z hľadiska spoľahlivosti poskytovaných služieb ten najhorší možný. Pokiaľ chcete mať vašu prezentáciu umiestnenú na takomto serveri, musíte spravidla súhlasiť s určitými podmienkami, ktoré napríklad dovoľujú poskytovateľovi služby umiestniť na vašu web stránku reklamu, zmazať vašu web stránku bez uvedenia dôvodu a podobne. Takáto služba je ale vhodná na testovanie funkčnosti web stránok (aj keď najlepšou alternatívou pre vývojára web stránok je skúšať funkčnosť stránky priamo u seba, na počítači, na ktorom sa web stránka vytvára).

Príklady poskytovateľov web priestoru zdarma:

<http://www.ic.cz/>

<http://mail.szm.com/free-e-mail/>

V prípade využitia takejto služby väčšinou dostanete (môžete si vybrať) tzv. doménu tretej úrovne (podľa služby zabezpečujúcej pomenovanie počítačov pripojených do Internetu - DNS - Domain Name System), čiže adresu, na ktorej bude vaša web stránka dostupná. Doména tretej úrovne vyzerá takto:

moj astranka.ic.cz

Slovo **mojastranka** pritom môže byť hocikaké (mimo tých, ktoré už existujú a sú použité).

Platení poskytovatelia web priestoru (webhosting)

Platených poskytovateľov web priestoru je veľmi veľa. Výber je široký, a preto je potrebné výber toho správneho nezanedbať. V prvom rade je dobré získať na poskytovateľa web priestoru dobré referencie. Ak je veľa ľudí so službami, poskytovanými naším vybraným poskytovateľom web priestoru spokojných, budeme pravdepodobne spokojní aj my.

Príklad ponuky platených poskytovateľov web priestoru:

MINIMUM 300 MB	OPTIMUM 1500 MB	MAXIMUM 5000 MB
<ul style="list-style-type: none">♦ Linux, PHP 5♦ neobmedzený počet schránok♦ POP3, SMTP, IMAP, Webmail♦ FTP, WebFTP, štatistiky, ...	<ul style="list-style-type: none">♦ Linux alebo Windows♦ PHP 5, ASP, ASP.NET♦ MySQL, PostgreSQL, Access♦ neobmedzený počet schránok♦ POP3, SMTP, IMAP, Webmail♦ FTP, WebFTP, štatistiky, ...	<ul style="list-style-type: none">♦ Linux alebo Windows♦ PHP 5, ASP, ASP.NET, CGI♦ MySQL, PostgreSQL, Access♦ neobmedzený počet schránok♦ POP3, SMTP, IMAP, Webmail♦ FTP, WebFTP, štatistiky, ...
100,- AKCIA: 65 Sk/mes. NOVINKA!!!	200,- AKCIA: 111 Sk/mes. a zľava 50 % z ceny domény!	300,- AKCIA: 169 Sk/mes. a zľava 100 % z ceny domény!

Pozrite si [porovnanie a detailnú špecifikáciu](#) našich webhostingových služieb!

OKREM CIEN UVEDENÝCH NA NAŠICH STRÁNKACH NEPLATÍTE ŽIADNE ĎALŠIE SKRYTÉ POPLATKY.
V prípade objednania služieb v akcii ručíme za nezvýšenie ceny za webhosting aj v budúcnosti!

Všetky služby s non-stop dohľadom nad servermi a garantovanou 100 % dostupnosťou!
Ceny sú uvedené bez 19 % DPH.

Pri výbere toho správneho poskytovateľa musíme postupovať aj podľa toho, aké podmienky pre svoj beh bude naša web stránka potrebovať - napríklad či potrebuje pre svoj beh nejaký určitý skriptovací jazyk, databázu a podobne. Pokiaľ by sme chceli robiť web stránku, na ktorej budú fotogalérie s veľkým množstvom fotografií a je na nej predpokladaná vysoká návštevnosť, budeme potrebovať rýchle pripojenie a neobmedzenú alebo aspoň dostatočne veľkú kapacitu pre uloženie a prenos údajov (tieto veci má väčšina poskytovateľov webhostingových služieb limitované).

Príklady platených poskytovateľov web priestoru:

<http://www.domains.sk/web/main.php>

<http://www.yegon.sk/>

Využitie platených poskytovateľov web priestoru je väčšinou spojené s registráciou domény druhej úrovne.

Doména druhej úrovne vyzerá takto:

mojastranka.sk

Slovo **mojastranka** pritom môže byť hocijaké (mimo tých, ktoré už existujú a sú použité). Sk na konci samozrejme znamená Slovensko, vy si však môžete (samozrejme po zaplatení poplatku za službu) zaregistrovať doménu druhej úrovne s koncovkou **.cz**, **.info**, **.com** atď. - podľa vhodnosti - pre koho má byť prevádzkovaná stránka určená a aký obsah má mať.

Dobrý poskytovateľ webhostingových služieb vám zároveň s doménou druhej úrovne poskytne možnosť vytvárať si neobmedzené množstvo domén tretej úrovne. Napríklad:

www.mojajstranka.sk obchod.mojajstranka.sk hocico.mojajstranka.sk

Zároveň sú bežne poskytované služby ako možnosť založiť si vlastné e-mail schránky pre firmu a samozrejme FTP prístup pre zmenu a úpravu web stránok. Web stránku je bežne možné spravovať za pomoci špecializovaného web rozhrania, za pomoci ktorého si môžete jednoducho kontrolovať a spravovať všetky parametre web stránky.

Využiť tento spôsob je pre bežnú web stránku najrozumnejším a najefektívnejším riešením. Ceny sú veľmi nízke a možnosti bohaté - pri správnom výbere budete určite s poskytovanými službami spokojní.

Vlastný web server

Najlepšia možnosť, ktorá ale nie je pre každého, je použitie vlastného web servera.

Vlastný web server je možné prevádzkovať aj na vlastnom internetovom pripojení, ale je to spojené s nevýhodami. Vysokorýchlostné internetové pripojenie, aké je nevyhnutné pre prevádzku niektorých špecializovaných web stránok, je veľmi drahé, takisto priestory, elektrická energia či nutnosť dohľadu, môže prevádzkovanie vlastného web servera predražiť.

V súčasnosti sa poskytujú služby umožňujúce buď prenájom a umiestnenie hardvéru (servera), alebo umiestnenie vlastného hardvéru (servera) v priestoroch spravovaných poskytovateľom služby - tzv. server housing. Poskytovateľ tejto služby zabezpečí priestory s dostatočne dimenzovaním napájaním elektrickou energiou, chladením, pripojením na internet a ďalšími službami spojenými s prevádzkou podobnej služby. Web server sa dá efektívne spravovať aj na diaľku, bez potreby byť priamo pri hardvéri. Preto je využitie server housingu asi najlepším riešením pre tých, čo chcú mať plnú a neobmedzenú kontrolu nad tým čo beží na ich serveri, nad jeho konfiguráciou a nad ďalšími, na ňom bežiacimi službami.

Použitie služby FTP pre správu súborov web stránky

Váš poskytovateľ priestoru pre web stránku vám musí poskytnúť aj spôsob, ktorým môžete spravovať súbory vašej web stránky. Bežne sa používa služba FTP (File Transfer Protocol), ktorá umožňuje vzdialený prístup, prenos a manipuláciu zo súbormi v priestore, kde je uložená vaša web stránka.

Ako využiť pri vytváraní obsahu WWW stránky rôzne programy - oboznámenie

Pre tvorbu web stránok existujú aj špecializované nástroje - editory web stránok. V podstate je možné vytvoriť web stránku bez akejkoľvek znalosti HTML. To je však veľmi zjednodušený pohľad na vec. Aby sme web stránku dokázali naozaj efektívne vytvoriť, potrebujeme poznať množstvo vecí, ktoré s tvorbou web stránok súvisia. Príkladom môže byť napríklad využitie formulárov. Áno, v editore web stránok dokážete naklikat' pekný formulár, ale aby sa dal naozaj používať, musíte poznať spôsob, akým treba jednotlivé formulárové prvky pomenovať, poslať, nehovoriac o ich spracovaní.

každom prípade aj pri použití niektorého z editorov web stránok je nutné, aby sme poznali základné princípy a pravidlá tvorby web stránok - základné HTML značky, spôsob ich zápisu, ich parametre a podobne.

Keď sme oboznámení s týmito nevyhnutnými základmi, dokáže nám dobrý editor web stránok veľmi urýchliť a zefektívniť prácu. Bez týchto znalostí však získate ďalší nástroj, s ktorým sa budete pasovať a bude vám pripadať veľmi zložitý a nepochopiteľný.

Typy editorov web stránok

Dá sa povedať, že web stránky sa dajú tvoriť v akomkoľvek jednoduchom textovom editore. Klasickým príkladom je Windows Notepad. Samozrejme je však ďaleko efektívnejšie použiť nástroj, ktorý je určený priamo ako pomôcka pre tvorbu web stránok. Poznáme dva základné typy editorov:

editory zdrojového kódu web stránok,

WYSIWYG (What You See Is What You Get - to, čo vidíš aj dostaneš) editory.

Editory zdrojového kódu

podstate jestvuje niekoľko desiatok nástrojov určených ako pomôcka k editovaniu zdrojového kódu web stránok (HTML a CSS). Niektoré sú univerzálne editory, umožňujúce editovať nielen web stránky, ale aj množstvo ďalších typov súborov. V čom nám dokáže taký editor vlastne pomôcť? Tieto editory používajú na zvýšenie efektívnosti práce niekoľko nástrojov:

zvýraznenie farebnej syntaxe zdrojového kódu (***syntax color highlighting***),

automatické dopĺňanie značiek a ponuka parametrov značiek,

klávesové skratky (napríklad na hrubé, šikmé písmo či vloženie tzv. tvrdej medzery ** **, ktorá slúži na typografické spojenie predložky so slovom),

otvorenie a editovanie súboru priamo zo vzdialeného web serveru,

a množstvo ďalších nástrojov.

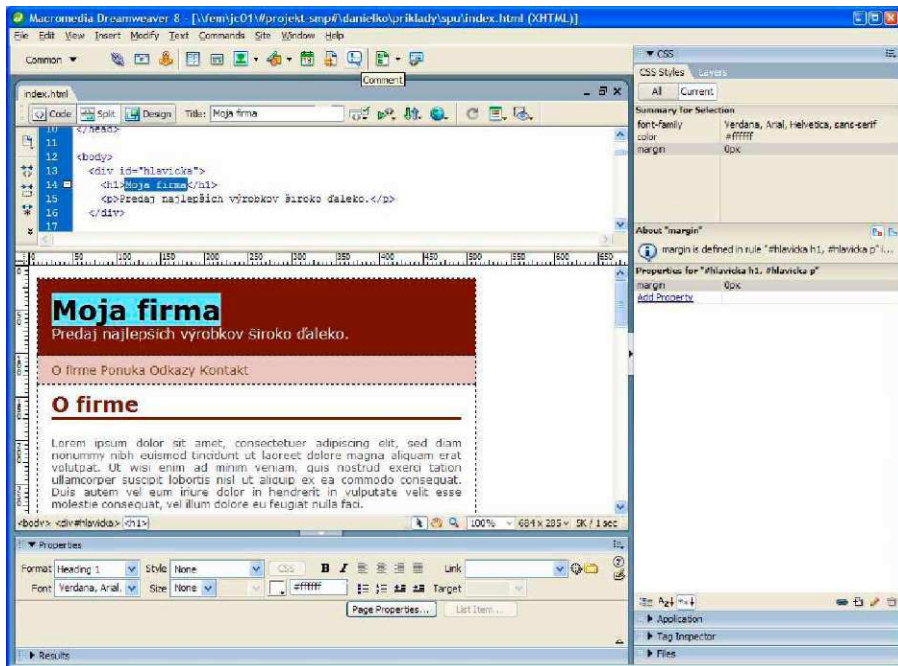
Príklady na v praxi používané editory:

Homesite - starší, ale stále veľmi obľúbený editor špecializovaný na tvorbu web stránok,
PSPad - univerzálny textový editor s množstvom nástrojov a pomôcok pre tvorbu web stránok
(<http://www.pspad.com/sk/>) - je ho možné používať zdarma,
a množstvo ďalších... (množstvo príkladov nájdete na serveri <http://www.slunecnice.cz/> v sekcii
Grafika-a-webdesign/Nastroje- webdesign/Editory-HTML/).

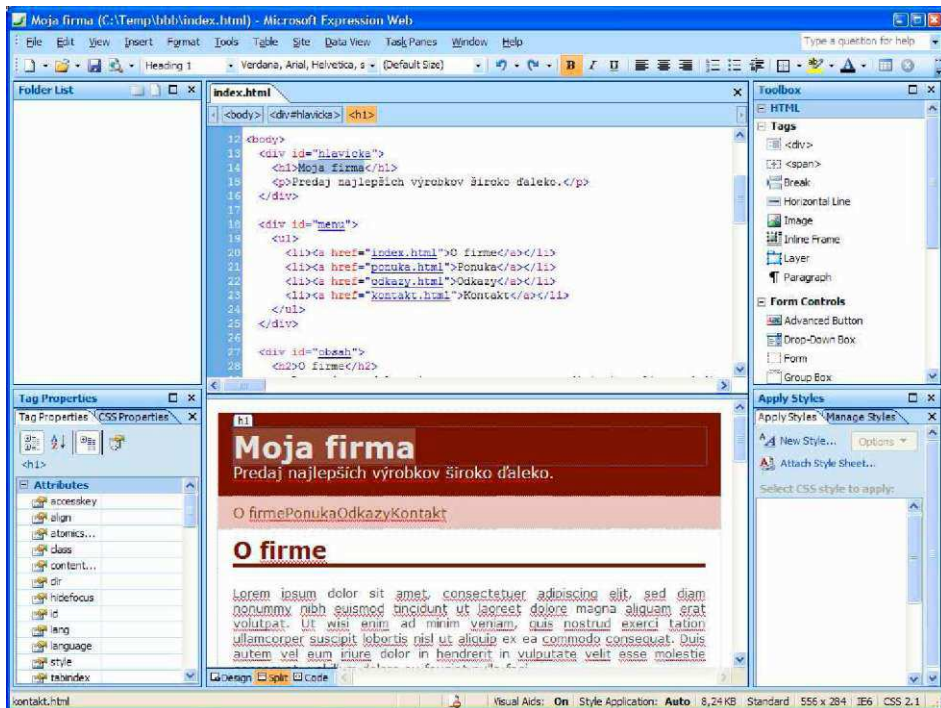
WYSIWYG editory

Výhodou WYSIWYG editorov je spôsob práce rovnaký ako napríklad v MS Word. Jednoducho píšete, zvýrazňujete text, nastavujete veľkosť písma, farby, vytvárate tabuľky - to všetko za pomoci výberov z menu. Každá zmena, ktorú urobíte, sa hneď zobrazí tak, ako bude vo výsledku zobrazená v prehliadačoch web stránok. Všetky zápisy do HTML kódu sa odohrávajú na pozadí - editor sám určí, ktorú značku má použiť pre požadované zobrazenie.

Ako sme však už písali predtým, stále je dôležité poznať princípy, na ktorých je tvorba web stránky založená. Aj v takomto editore je veľakrát nutné zasiahnuť do zdrojového tvaru web stránky, priamo do HTML kódu. Pre profesionála je v oblasti WYSIWYG editorov výber jasný. Asi najdokonalejším nástrojom, spájajúcim všetky výhody textových editorov zdrojového kódu a WYSIWYG editorov s dokonalým zobrazením editovanej stránky je Adobe Dreamweaver (v aktuálnej verzii 10).



Microsoft ponúka ako konkurenciu pre Dreamweaver program Microsoft Expression Web Designer (predtým MS FrontPage). Tento program je celkom dôstojným súperom pre Dreamweaver a môže vám tiež celkom efektívne pri tvorbe web stránok pomôcť.



Samozrejme Dreamweaver a Microsoft Expression Web Designer nie sú jedinými predstaviteľmi WYSIWYG editorov. Dokonca aj MS Word umožňuje uložiť editovaný dokument ako HTML - čím sa dá povedať, že dokáže podobné veci ako špecializované editory. Je to však naozaj núdzové riešenie. MS Word nedokáže pracovať so štýlmi a ním vytvorené stránky obsahujú veľmi neefektívny HTML kód - pre ďalšie úpravy v inom editore ako je MS Word nepoužiteľný.

Záver

Aby sme boli schopní vytvoriť web stránku, ktorá bude spĺňať dnešné požiadavky a štandardy na kvalitnú webovú prezentáciu, musíme byť naozaj všestranní. Musíme dokázať vytvoriť grafický návrh na úrovni, vedieť pracovať s grafickými editormi, poznať XHTML, CSS a (pokiaľ chceme vytvoriť plne interaktívnu a dynamickú web stránku) ďalšie technológie ako je FLASH, JavaScript, PHP, ASP.NET a podobne. Je toho veľmi veľa a nájde sa len málo profesionálnych webdizajnérov, ktorí dokážu obsiahnuť všetky tieto požiadavky v jednej osobe.

Preto aj v odbore tvorby web stránok dnes vládne špecializácia. Návrh grafiky spracúva grafik, samotnú kostru web stránky - jej prepracovanie z návrhu do XHTML a CSS zabezpečuje webdizajnéer, ďalšie spracovanie do podoby dynamickej web stránky zabezpečí web programátor.

Pokiaľ chceme, aby webová prezentácia našej firmy bola naozaj profesionálne vytvorená, je lepšie obrátiť sa na špecialistov - firmy, ktoré ponúkajú tvorbu web stránok a majú dobré referencie. Určite nebudete sklamaní. Pokiaľ to však myslíte s tvorbou web stránok vážne, nájdite si čas na štúdium literatúry a zdrojov na Internete. Počítajte s tým, že veľa najlepších zdrojov je dostupných len v anglickom jazyku.

No a na záver sa pripravte na to, že najlepší postup, ako sa naučiť tvoriť web stránky, je experiment. Treba skúšať, pozorovať a meniť zdrojové kódy web stránok. Je v tom veľa z tvorivej činnosti podobnej umeniu.

Pokiaľ nezastanete v polovici cesty, môžu sa z vás stať profesionáli aj v odbore tvorby web stránok.

[URL1] WIKIPEDIA - Hypertext markup language

<http://sk.wikipedia.org/wiki/HTML> [URL2] WIKIPEDIA - Extensible Hypertext Markup Language

<http://sk.wikipedia.org/wiki/XHTML> [URL3] WIKIPEDIA - Cascading Style Sheets

<http://sk.wikipedia.org/wiki/Kaskádové> štýly

[URL4] Quirksmode.ORG - The position declaration

<http://www.quirksmode.org/css/position.html> [URL5] Jak psát web

<http://www.jakpsatweb.cz/> [URL6] Kurz Internet a tvorba www stránek

<http://www.hmw.sk/kurz-ke/index.html> [URL7] Interval.CZ - XHTML - kompletní průvodce

<http://interval.cz/serialy/xhtml-kompletni-pruvodce/> [URL8] Interval.CZ - Kaskádové styly - kompletní průvodce
začátečníka <http://interval.cz/serialy/kaskadove-styly-kompletni-pruvodce-> zacatecnika/

[URL9] Interval.CZ - CSS2 - kaskádové styly druhé generace

<http://interval.cz/serialy/css2-kaskadove-styly-druhe-generace/> [URL10] World Wide Web Consortium (W3C)

<http://www.w3.org/> (v angličtině)

PÍSEK, S: HTML a XHTML, začínáme programovat. Praha : GRADA Publishing, 2003. 256 s. ISBN
80-247-0571-0

SNÍŽEK, M.: CSS pro zelenáče. Praha : Neocortex, 2004. 295 s. ISBN 8086330-14-1

MIKLE, P.: XDHTML - HTML, XHTML, DHTML - úplná přesná referenční příručka. Brno : Zoner Press, 2004.
208 s. ISBN 80-86815-01-3

MIKLE, P.: XCSS - CSS1, CSS2, CSS3 - úplná přesná referenční příručka. Brno : Zoner Press, 2004. 232 s.
ISBN 80-86815-13-7