

# Webové technológie a publikovanie na webe 1

## Identifikácia modulu

Aktivita projektu:	1.3 Ďalšie vzdelávanie kvalifikovaných učiteľov informatiky na 2. stupni ZŠ a na SŠ
Línia aktivity:	Digitálna gramotnosť učiteľa
Predmet:	Digitálna gramotnosť učiteľa

### Garant predmetu:

RNDr. Martin Homola  
homola@fmph.uniba.sk

### Autori:

RNDr. Martin Homola  
KAI FMFI UK, Bratislava  
FBK, Trento, Taliansko  
Mgr. Ján Guniš  
UPJŠ, Košice

## Zaradenie modulu



## Abstrakt modulu

Webové technológie a webové publikovanie nepochybne predstavujú oblasť informatiky s mimoriadne silným celospoločenským významom. Záujem o túto oblasť u študentov zaoberajúcich sa informatikou je značný, a to už na stredných školách. Zároveň ide o oblasť, ktorá sa rýchlo vyvíja, je plná všemožných technológií a prístupov, z ktorých väčšina je použiteľná na dosiahnutie stanoveného cieľa, nie všetky však zaručujú rovnakú úroveň kvality výsledku. Pedagóg, ktorý sa v problematike dobre orientuje a dokáže študentom poradiť a nasmerovať ich správnym smerom je tak prínosom nielen pre individuálny profesionálny rast svojich študentov, ale pre celú spoločnosť.

Cieľom modulu je vybudovať u učiteľov informatiky široký prehľad v oblasti webu, webových technológií, a tvorby webového obsahu, od textu až po multimédiá; a umožniť im tak zorientovať sa v tejto oblasti, v ktorej sa mnohí z ich študentov pohybujú s ľahkosťou a samozrejmosťou. Účastníci vzdelávania sa oboznámia s architektúrou webu, so základnými jazykmi webovej tvorby, s grafickými a multimedialnými formátmi používanými na webe, ale aj s postupmi a špecifikami webového publikovania, a so štandardami kvality webových dokumentov (použiteľnosť a prístupnosť). Celú problematiku predstavíme zoširoka, s cieľom vybudovať potrebný prehľad, povedomie o dôležitosti celej oblasti a motiváciu zaoberať sa ňou v individuálnom samovzdelávaní.



# Obsah

Webové technológie a publikovanie na webe 1 .....	1
Identifikácia modulu .....	1
Zaradenie modulu .....	1
Abstrakt modulu .....	1
Obsah .....	2
Kapitola 1: Úvod .....	3
Kapitola 2: Architektúra webu .....	4
Webový server a webový klient .....	4
Webové adresy .....	4
Protokol HTTP .....	5
Webový obsah a webové aplikácie.....	6
Ďalšie webové technológie.....	7
Čo sme sa naučili .....	7
Kapitola 3: HTML a XHTML .....	8
Štruktúra HTML dokumentov .....	8
Metadáta .....	9
Grafická reprezentácia a vykreslenie dokumentu .....	10
Členenie a formátovanie textu.....	10
Vkladanie obrázkov, objektov a hypertextových odkazov .....	11
Zoznamy a tabuľky .....	11
Formuláre .....	12
Čo sme sa naučili .....	12
Kapitola 4: Kaskádové štýly.....	13
Princípy CSS.....	13
Selektory .....	14
Fonty, formátovanie textu, farby a pozadia .....	15
Box model, dĺžka a dĺžkové jednotky .....	16
Ďalšie možnosti CSS .....	17
Čo sme sa naučili .....	17
Kapitola 5: Multimédiá .....	18
Grafika na webe .....	18
Rastrová grafika.....	19
Vektorová grafika .....	21
Animácie, video a interaktívne objekty .....	22
Zvuková informácia .....	23
Webové nástroje pre správu multimédií.....	23
Čo sme sa naučili .....	24
Kapitola 6: Publikovanie na webe .....	25
Žánre a typy webových stránok .....	25
Štruktúra a vzhľad webových stránok .....	26
Tvorba a úprava textu .....	27
Čo sme sa naučili .....	27
Kapitola 7: Použiteľnosť a prístupnosť na webe .....	28
Použiteľnosť (usability).....	28
Prístupnosť (accessibility).....	30
Čo sme sa naučili .....	32
Kapitola 8: Praktické cvičenia.....	33
Kapitola 9: Čo sme sa naučili v tomto module .....	35
Literatúra a použité zdroje .....	35

## Kapitola 1: Úvod

World Wide Web (WWW, ďalej len *web*) je internetová služba, ktorej počiatky siahajú do prvej polovice deväťdesiatych rokov. Napriek tomu, že ide o pomerne mladú službu, už dnes je možné hovoriť o webe ako o multimediálnom komunikačnom médiu so silným celospoločenským a ekonomickým významom, pričom jeho význam je neustále na vzostupe. Web dnes využívame denne za účelom vyhľadávania informácií všetkého typu, v práci, ale aj v súkromí. Pre spravodajské médiá a vydavateľov je web publikačným kanálom, cez ktorý informácie dorazia za čitateľmi najrýchlejšie. Pre obchodné spoločnosti po celom svete už dávno nie je prítomnosť na webe strategickým rozhodnutím, ale nevyhnutnosťou; web sa stal tým hlavným komunikačným kanálom, cez ktorý oslovujú klientelu, informujú o svojich produktoch a službách. Pre vedcov a odborníkov sa web stal knižnicou tretieho tisícročia, väčšina odborných publikácií z posledných rokov je dostupná prostredníctvom webu, či už zdarma, alebo formou platených prístupov do odborných časopisov.

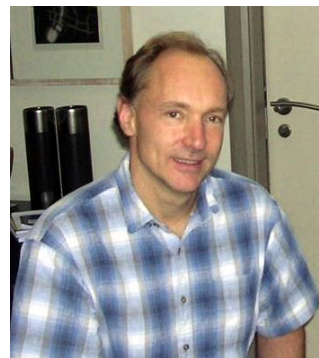
Možnosť publikovať na webe však nie je otvorená len internetovým médiám a vydavateľstvám, obchodným spoločnostiam, či univerzitám. Vďaka novým webovým službám ako sú blogy, wiki, sociálne siete, a iné, je možnosť publikovať otvorená prakticky každému. Jedinou nutnou investíciou je prístup k internetu. Vďaka tomuto novému prístupu, ktorý dostal názov web 2.0, sa web zmenil na dvojsmerné médium, ktorého obsah jeho používatelia nie len čítajú, ale v plnej miere sa podieľajú aj na jeho tvorbe (pozri stĺpec na nasledujúcej strane).

Následne, problematika webových technológií a webového publikovania dnes celkom prirodzene predstavuje oblasť informatiky s mimoriadne silným významom. Toto zd'aleka neplatí len v úzkych odborných kruhoch. Práve naopak, webové technológie, webdizajn a vývoj webových aplikácií sú obživou pre veľké množstvo programátorov a po webových špecialistoch je neustále silný dopyt. Študenti informatiky na vysokých školách, ale aj mnohí študenti stredných škôl, ktorí sa o informatiku a o programovanie zaujímajú, sú si toho vedomí, a prejavujú záujem o túto oblasť.

Keďže web sa neustále rozvíja a napreduje, platí to rovnako aj pre množinu technológií ktoré sú s ním späté. Mnohé z týchto technológií sú pomerne mladé, a aj keď ponúkajú množstvo užitočných vlastností a výhod, prinášajú aj úskalia a problémy. Rodina webových technológií sa stále rozrastá a mení, vývoj napreduje, prichádzajú nové technológie. Niektoré staršie naopak zastarávajú a sú nahradené novými. Navyše, web je značne heterogénnym prostredím, a žiadnu z webových technológií nemožno preto používať ako „univerzálny liek“; hoc väčšina technológií je použiteľná na dosiahnutie stanoveného cieľa, nie všetky zaručujú rovnakú úroveň kvality výsledku. Podstata kvalitných webových aplikácií často tkvie práve v správnej kombinácii rôznych webových technológií. Napriek tomu sa u webových programátorov, predovšetkým z radov začiatočníkov, môžeme stretnúť s prílišným nadšením pre niektorú konkrétnu technológiu, či platformu, ktorú potom používajú aj nevhodne.

A tu sa dostávame k cieľu tohto modulu a materiálu. Mnohí študenti sa o webové technológie zaujímajú. Niektorí z nich dokážu svojou aktivitou a zariadenosťou zdolať jednotlivé technológie s neuveriteľnou ľahkosťou. Nie vždy však sami dokážu problematiku uchopiť dostatočne zoširoka a uvedomiť si všetky súvislosti. Učiteľ informatiky, ktorý sa o web zaujíma, a o jednotlivých webových technológiách, a predovšetkým o princípoch, na ktorých je web vystavaný, má prehľad, dokáže študentov lepšie viesť, nasmerovať ich správne, upozorniť ich na začiatočnicke chyby, a pod. S prihliadnutím na celospoločenskú dôležitosť webu považujeme za potrebné oboznámiť učiteľov informatiky aspoň so základmi z tejto oblasti.

Cieľom modulu je oboznámiť učiteľov informatiky s heterogénnym prostredím webu, s jeho architektúrou a s princípmi jeho fungovania. Zoznámime sa jednak s vybranými technológiami ako protokol HTTP, jazyky (X)HTML a CSS, webová grafika a multimediálny webový obsah, ale tiež s princípmi a zaužívanými pravidlami webového publikovania a so štandardami kvality webových dokumentov, ako sú prístupnosť a použiteľnosť. K týmto témam pristupujeme zoširoka, snažíme sa budovať prehľad, oboznamujeme sa s ich silnými ale aj slabými stránkami. Modul sa snaží vzbudiť záujem o túto oblasť a motiváciu pre ďalšie samoštúdium. Poskytuje množstvo odkazov na literatúru. V Kapitole 8 pripájame aj sériu praktických cvičení, ktoré čitateľom odporúčame vypracovávať priebežne, počas čítania materiálu.



Za vynálezcu webu sa pokladá britský informatik **Tim Berners-Lee**, ktorý na prelome 80. a 90. rokov 20. storočia pracoval vo švajčiarskom inštitúte CERN, kde už v tom čase mali obrovské množstvo elektronických dokumentov a existoval aj problém, ako ich efektívne organizovať a ako v nich vyhľadávať. Web pôvodne vznikol práve za týmto účelom. Sir Tim, ktorý bol neskôr zásluhou svojho objavu povýšený do rytierskeho stavu sa preslávil aj ako ohromný vizionár, keď svoj vynález počas jednej z prvých prezentácií, pri ktorej bol použitý len jediný webový server, nazval „World Wide Web“, teda „pavučina pokrývajúca celý svet“.



V roku 1994 vzniká pod jeho vedením konzorcium W3C, ktoré je nezávislým „inštitútom“ webu a webových technológií. W3C sa stará o ich ďalší rozvoj a dbá nad ich správnym používaním.

Ani popri vedení W3C Sir Tim vizionárstvo nezaviesil na kliniec. V roku 2001 predstavil svetu víziu webu budúcnosti, na ktorom ľudia a počítače dokážu skutočne spolupracovať. Nazval ho **sémantický web**.



Tim O'Reilly sa výrazne podieľal na spopularizovaní pojmu **web 2.0**. Pre tento pojem existuje viacero „definícií“, v podstate ide o víziu „nového, lepšieho webu, ktorý má oveľa bližšie k používateľom ako web, ktorý sme poznali predtým. Web 2.0 ponúka jednoduchší spôsob, ako tvoriť a zdieľať obsah, nové služby, cez ktoré sa môžu používatelia ľahšie spojiť a komunikovať, čistejší a prehľadnejší dizajn, ako aj nové technológie pre rýchlejšie webové aplikácie a ešte rýchlejší prístup k informáciám. Kým pôvodný web bol o obsahu, web 2.0 je o ľuďoch.“

**Web 2.0 je v skutočnosti ten istý web**, ako predtým, Vďaka tejto vízii a vďaka obrovskému množstvu jej nasledovníkov, ho dnes však používame úplne inak, ako predtým. Pre web 2.0 sú typické predovšetkým rôzne užitočné nástroje a služby. Index týchto služieb ponúka napr. stránka: <http://www.go2web20.net/>

## Kapitola 2: Architektúra webu

Web je internetová služba, ktorá nám umožňuje publikovať na internete elektronicou formou rôzne dokumenty a rovnako aj k dokumentom publikovaným na internete prostredníctvom tejto služby pristupovať. Tieto dokumenty, ktorým hovoríme aj webové stránky, môžu byť rôznorodé - od jednoduchých textových dokumentov obohatených o obrázky a odkazy, cez multimediálne dokumenty, až po zložité programy, ktorým hovoríme webové aplikácie, a ktoré s používateľmi sofistikovaným spôsobom interagujú, a sprostredkujú im všemožné služby od odosielania elektronickej pošty po nákup tovaru v elektronických obchodných domoch.

Fungovanie webu je postavené na rodine technológií, medzi ktoré radíme predovšetkým: webový server a webový klient, webový adresný priestor (URI), doménový menný systém (DNS), komunikačný protokol HTTP, jazyky pre tvorbu webového obsahu a webových aplikácií; ale môžeme sem zaradiť aj mnohé ďalšie, ako napr. webové vyhľadávače, systémy správy obsahu (CMS), či syndikáciu obsahu (RSS). Všetky tieto technológie svojimi vlastnosťami a možnosťami determinujú vlastnosti a možnosti webu ako takého, a tak ich poznanie a pochopenie je kľúčom k pochopeniu fungovania, možností a obmedzení webového publikovania a webovej tvorby. Preto sa s nimi na úvod v krátkosti oboznámime.

### Webový server a webový klient

Architektúra webu je príkladom tzv. architektúry klient-server, kedy medzi sebou komunikujú dva programy, z ktorých jeden službu poskytuje na základe požiadavky druhého, ktorý službu využíva. Počítač, ako aj program, ktorý sprístupňuje webové stránky nazývame preto **webový server** (tiež webserver) a program, ktorý používame na to, aby sme dokumenty stiahli a prečítali, nazývame webový klient (oveľa častejšie však webový alebo internetový prehliadač, alebo jednoducho **prehliadač**). Webový server aj klient sú teda vlastne programy nainštalované na počítači, môžeme ich jednoducho nainštalovať na ľubovoľný počítač.

Pre webový server existuje niekoľko softvérov, mnohé sú voľne dostupné pod open-source licenciou. V súčasnosti najpoužívanejším webovým serverom je Apache, voľne dostupný pod licenciou GNU GPL. Inštalovať webserver na vlastnom počítači sa hodí hlavne na vývoj a testovanie webových stránok a webových aplikácií. Pokiaľ chceme nejakú webovú stránku umiestniť na internete, často je jednoduššie využiť **webhostingovú službu**, ktorá nám sprostredkuje všetko potrebné vrátane nákupu a konfigurácie domény, e-mailov, atď.

Existuje hneď niekoľko webových prehliadačov, všetky sú voľne dostupné. Najpoužívanejším z nich je Internet Explorer od Microsoftu, ktorého dominantné postavenie na trhu je dané tým, že je dodávaný spolu s operačným systémom Windows. Konkurenciu v súčasnosti predstavujú predovšetkým prehliadače Mozilla Firefox, Opera, Safari a Google Chrome. Keďže ide o veľmi používaný nástroj, pomocou ktorého často pracujeme s citlivými údajmi, oplatí sa vyskúšať viacero prehliadačov, a vybrať si z ponuky vyspelý a hlavne bezpečný produkt.

### Webové adresy

Adresy všemožných webových stránok nás dnes doslova obklopujú a valia sa na nás zo všetkých strán. Aj keď väčšinou stačí zadať do prehliadača adresu v podobe [www.matfyz.sk](http://www.matfyz.sk), či [matfyz.sk](http://matfyz.sk), a funguje to; vo všeobecnosti môžu mať webové adresy oveľa zložitejší tvar. Webové adresy podliehajú štandardom DNS a URI. Práve vďaka týmto dvom štandardom, konkrétne tým, že sú mimoriadne pružne navrhnuté, mohol vôbec web narásť do svojich dnešných rozmerov.

V prvom rade, adresy počítačov na internete sú v skutočnosti číselné. Hovoríme im tiež **IP adresy**. Reálna adresa, pomocou ktorej sa náš prehliadač spojí s webovým serverom, nie je [www.matfyz.sk](http://www.matfyz.sk), ale napr. 194.67.123.101. **Menné adresy** (tiež doménové mená alebo **domény**), ktoré zadávame do prehliadača my, vznikli predovšetkým preto, že tie číselné si ľudia veľmi zle pamätajú. Aby sme sa mohli so serverom pripojeným do internetu spojiť, potrebujeme mennú adresu servera preložiť na jeho skutočnú IP adresu. O preklad adres sa stará tzv. doménový menný

system (DNS, Domain Name System), ktorý má niekoľko zaujímavých vlastností:

- umožňuje nám doménu vlastniť a využívať, napr. pre potreby vytvorenia adresy pre webovú stránku;
- je hierarchický, pod každou doménou môžeme vytvoriť ľubovoľný počet subdomén, ktoré sa tvoria prefixom a bodkou, napr. pod doménou [matfyz.sk](http://matfyz.sk) vytvoríme subdomény [blog.matfyz.sk](http://blog.matfyz.sk) a [wiki.matfyz.sk](http://wiki.matfyz.sk);
- jednej IP adrese môže byť priradených hneď niekoľko domén; napr. webové stránky [blog.matfyz.sk](http://blog.matfyz.sk) a [wiki.matfyz.sk](http://wiki.matfyz.sk) môže v skutočnosti obsluhovať ten istý webový server. V realite napr. menší poskytovateľ webhostingu na niekoľko málo serveroch prevádzkuje tisíce webových stránok;
- naopak, jednej doméne môže byť priradených niekoľko IP adries; napr. veľmi vyťaženu webovú stránku môže obsluhovať hneď niekoľko serverov.

Doménové meno je v skutočnosti len časťou webovej adresy. Aj keď najčastejšie si vystačíme s doménovým menom a náš prehliadač si celú adresu domyslí, napr. [www.matfyz.sk](http://www.matfyz.sk) doplní na <http://www.matfyz.sk/>, zložité webové aplikácie používajú oveľa zložitejšie adresy. Prípustné formáty webových adries definuje štandard URI (Uniform Resource Identifier). Webové adresy teda nazývame tiež URI adresy a môžu mať vo všeobecnosti nasledovný tvar:

Príklad zložitej URI adresy:

<http://example.org:8080/path/to/document?par1=val&par2=val2&...#id333>

schéma      adresa      port      cesta      dotaz      fragment

#### Anatómia URI adries

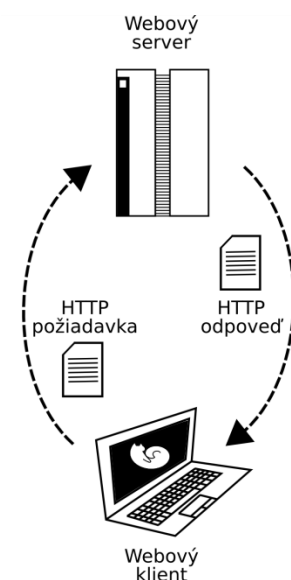
URI adresy sa skladajú z niekoľkých častí. Časť *schéma* určuje komunikačný protokol; časti *adresa* a *port* určujú adresu servera a port, na ktorom webserver beží; časť *cesta* určuje uloženie dokumentu na serveri; časť *dotaz* je odovzdaná webserveru na spracovanie; časť *fragment* určuje miesto vo vnútri dokumentu, na ktoré adresa ukazuje. URI adresy sa používajú na webe (schémy `http:` a `https:`), ale aj pri iných internetových službách (schémy `ftp:`, či `mailto:`). Adresa servera obhospodaruje štandard DNS. Väčšina webových serverov beží na dohodnutom porte 80, preto môžeme port z adresy vynechať. Kým schéma a adresa nie sú case-sensitive, zvyšok adresy vo väčšine prípadov áno. Časť *cesta* určuje, ktorý dokument má server poslať ako odpoveď, pričom tento dokument môže byť aj výstupom programu, ktorý sa na serveri vykoná - v takom prípade je časť *dotaz* odovzdaná tomuto programu ako vstup. No a nakoniec, časť *fragment* nám umožňuje adresovať nielen ucelený dokument, ale aj nejakú jeho časť - táto ale musí byť v dokumente vyznačená za pomoci tzv. fragmentového identifikátora.

## Protokol HTTP

Prehliadač komunikuje s webserverom pomocou **protokolu HTTP** (HyperText Transfer Protocol). Protokol špecifikuje formát správ, ktoré si prehliadač a webserver vymieňajú. HTTP protokol funguje na jednoduchom princípe požiadavka-odpoveď. Potom ako napíšeme do prehliadača URI adresu, prehliadač zistí IP adresu servera, vytvorí TCP spojenie, a odošle na adresu servera požiadavku (HTTP request). Server požiadavku spracuje, a pošle naspäť odpoveď (HTTP response). Keďže web slúži predovšetkým na získavanie dokumentov, HTTP odpoveď zvyčajne obsahuje dokument umiestnený na danej adrese. Správy HTTP protokolu sa skladajú z **hlavičky** a z **tela** správy. Požiadavka aj odpoveď sú textové správy, pričom v tele správy sa môžu prenášať aj binárne dáta (napr. vždy, keď sťahujeme obrázok). V tele správy sa teda prenáša samotný dokument, a v hlavičke sa prenášajú rôzne metadáta týkajúce sa komunikácie a dokumentu:

**stavový kód:** trojmiestny číselný kód identifikuje vyhodnotenie požiadavky webserverom (200 - dokument nájdený; 301, 302 - presmerovanie; 403 - prístup zamietnutý, 404 - dokument sa nenašiel, atď.);

**metóda** (HTTP request method): protokol HTTP rozlišuje niekoľko typov požiadaviek, ktorým hovoríme metódy. V praxi sa používajú predovšetkým metódy GET a POST. Metóda GET slúži na základný typ komunikácie: používame ju na



Princíp protokolu HTTP je veľmi jednoduchý. Klient pošle serveru požiadavku, ten ju spracuje a pošle odpoveď. Tento „cyklus“ sa opakuje po každom kliknutí.

Protokol HTTP patrí do **aplikačnej vrstvy** sieťovej komunikácie. Táto je postavená nad **transportnou vrstvou**, pomocou ktorej sa počítače vedú spojiť a komunikovať. Spojenie na úrovni tejto vrstvy je vytvorené prostredníctvom protokolu TCP, preto hovoríme aj o **TCP spojení**.

**Metadáta** sú „dáta o dátach“. Metadáta majú v informatike široké spektrum využitia. Napr. keď vytvoríme digitálnu fotografiu, fotoaparát uloží do súboru s fotografiou aj údaje o expozícii, čas vytvorenia, formát v ktorom je zaznamenaná, či jej orientáciu. Pri sieťovej komunikácii sú metadáta potrebné hlavne na to, aby prijímajúca strana vedela, ako má s nimi naložiť (formát, kódovanie), či bol súbor doručený celý (veľkosť) a dokedy je platný (doba expirácie).

vyžiadanie si dokumentu z webservera na základe URI adresy. Metóda POST slúži naopak na odoslanie dát na server za účelom ich uloženia, či spracovania; dáta sa posielajú v tele správy. Aj výsledkom metódy POST je nakoniec nejaký dokument, väčšinou informácia o spracovaní odoslaných dát. Komuničnú metódu volí vždy iniciátor komunikácie, teda webový klient;

**cookies:** webserver môže do odpovede „pribaliť“ špeciálne dáta, ktorým hovoríme cookies. Podstatou cookies je, že klient pri každej ďalšej komunikácii so serverom pošle vždy všetky cookies naspäť. Toto sa najčastejšie využíva na identifikáciu opakovanej komunikácie s tým istým klientom. Napr. aby sme nemuseli opakovane zadávať meno a heslo vždy s každou novou požiadavkou.

**platnosť dokumentu:** dátum dokedy je dokument aktuálny a informácia o tom, či sa má ukladať v cache-pamäti alebo nie;

**ďalšie metadáta** potrebné na spracovanie dokumentu: typ odoslaných dát (MIME), veľkosť, kódovanie, a pod.

## Webový obsah a webové aplikácie

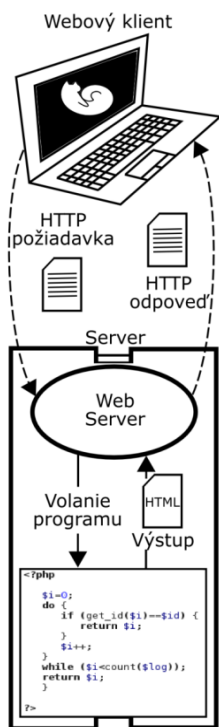
Protokol HTTP umožňuje prenášať dáta v ľubovoľnom formáte, a tak publikovať na webe môžeme vlastne akýkoľvek súbor. Podstatou webu je však jeho hypertextový obsah. **Hypertext** je text obohatený o tzv. **hypertextové odkazy**, ktoré sú linkami z jedného dokumentu do druhého. Podstatou hypermedií je práve možnosť ľubovoľne sa odkazovať v dokumente na iné dokumenty. Základným typom dát na webe sú teda hypertextové dokumenty v jazyku **HTML**. Tento jazyk definuje predovšetkým štruktúru dokumentov. Pre ich typografickú úpravu a grafické spracovania sa používa jazyk **kaskádových štýlov (CSS)**. S týmito jazykmi sa podrobnejšie oboznámime v nasledovných kapitolách tohto modulu. Je tiež nutné poznamenať, že dnes už väčšinou nepíšeme celé HTML súbory v textovom editore; za účelom úpravy webovej stránky používame program, ktorý sa volá **system správy obsahu (CMS)**, vid' nižšie).

HTML dokumenty môžu byť obohatené o rôzny multimediálny obsah. Následne hovoríme o **hypermediálnych** dokumentoch, ktoré môžu obsahovať vložené obrázky a zvuky v rôznych formátoch. Aj keď natívne (t.j. samy od seba) prehliadače podporujú iba niekoľko základných formátov, ďalšie formáty je možné doinštalovať vo forme rôznych **pluginov**. Takto môžeme prehliadač obohatiť o schopnosť prehrávať video ako aj zobrazovať pomerne pokročilé multimediálne formáty, ako je napr. veľmi populárny formát Flash. Multimédiám v kontexte webu je venovaná posledná kapitola modulu.

HTML dokumenty môžu byť tiež obohatené o tzv. skripty. Skript je programový kód vložený do dokumentu, ktorý sa v prehliadači inicializuje a vykoná po načítaní stránky, prípadne potom ako ho používateľ aktivuje kliknutím na nejaký ovládací prvok. Skript môže vypísať hlásenie, môže spracovať a prepísať časť dokumentu, a s používateľom interagovať. Keďže skript sa vykoná v prehliadači, hovoríme o tzv. skriptovaní alebo tiež **programovaní na strane klienta (client-side)**. Najpoužívanejším skriptovacím jazykom je jazyk **JavaScript** (niekedy tiež **ECMAScript**), ktorý je veľmi dobre podporovaný prakticky v každom modernom prehliadači. Na skriptovaní stavia aj v súčasnosti veľmi populárna technológia **AJAX**, ktorá nám umožňuje pomocou skriptu načítať doplnujúce dáta z webservera a následne ich do stránky vložiť bez nutnosti načítať a pregenerovať celú stránku.

Ešte častejšie sa na webe stretávame s **programovaním na strane servera (server-side)**. V takomto prípade HTML dokumenty nie sú dopredu pripravené a uložené na webovom serveri, ale sú vytvárané na serveri špeciálnym programom. Aj keď technicky je možné napísať takýto program v úplne ľubovoľnom programovacom jazyku, v praxi sa používajú predovšetkým jazyky PHP, Java, ASP, a niektoré ďalšie. Princíp fungovania je vždy jednotný a je založený na tzv. rozhraní **CGI (Common Gateway Interface)**:

1. webserver dostane požiadavku od klienta a pri jej spracovaní zistí, že nejde o požiadavku na statický HTML dokument, ale na server-side skript (najčastejšie podľa koncovky súboru, napr. „.php“);
2. webserver inicializuje a vykoná príslušný skript, pričom na vstupe mu odovzdá *dotazovú* časť URI adresy, prípadne dáta zaslané metódou POST;



**Rozhranie CGI** je princípom fungovania webových aplikácií.

3. skript po vykonaní vráti výstup webserveru - v typickom prípade vygenerovaný HTML dokument;
4. webserver odošle klientu HTTP odpoveď s vygenerovaným dokumentom.

Zložitejšie programy vykonávané touto technikou na webovom serveri nazývame **webové aplikácie**. Aj keď je tento pojem spájaný predovšetkým so server-side programovaním, moderné webové aplikácie ho často kombinujú aj so skriptovaním na strane klienta, pričom sa hojne využíva technológia AJAX. Väčšina stránok, ktoré navštevujeme, sú v skutočnosti webové aplikácie. Celá táto oblasť v súčasnosti prudko napreduje a intenzívne sa vyvíja vo všetkých svojich aspektoch. Už dnes napr. existujú ucelené integrované vývojové prostredia (IDE), v ktorých je možné webovú aplikáciu vyvíjať podobne ako vyvíjame projekt v Delphi, či Javovskú aplikáciu v prostredí Eclipse. Takéto IDE sa postará o vygenerovanie potrebného kódu, ktorý pobeží na serveri, ako aj na strane klienta, umožní ladenie, testovanie, atď. Ozývajú sa aj hlasy, ktoré predpovedajú, že budúcnosť bežného, napr. kancelárskeho softvéru je výlučne vo webových aplikáciách. Tvrdí to napr. spoločnosť Google, ktorá za týmto účelom v súčasnosti vyvíja platformu Google Chrome.

## Ďalšie webové technológie

V tejto časti sme vymenovali hlavné technológie, ktoré vytvárajú architektúru webu. Tento zoznam však nie je úplný, web je veľmi dynamické médium a rôzne nové technológie sa objavujú veľmi často, pritom niektoré iné po čase upadnú do zabudnutia, sú nahradené inými alebo sa z nejakého dôvodu prestanú používať. V širšom zmysle musíme medzi aktuálne webové technológie zaradiť aj:

**webové vyhľadávače:** webových stránok je obrovské množstvo a sú veľmi premenlivej kvality, nájsť na webe to čo hľadáme si bez vyhľadávačov už neodkážeme predstaviť. Význam vyhľadávačov stále rastie: kým pred niekoľkými rokmi bolo najcennejšou komoditou na webe ľahko zapamätateľné a výstižné doménové meno, dnes jeho význam klesá - oveľa dôležitejšie je byť dostatočne vysoko vo výsledkoch vyhľadávania. Vznikol dokonca celý nový biznis, ktorý sa tým zaoberá: **SEO** (Search Engine Optimization), t.j. zabezpečenie lepšieho umiestnenia webovej stránky vo výsledkoch vyhľadávania;

**systémy správy obsahu** (Content Management System, **CMS**): je akákoľvek webová aplikácia, ktorá nám umožňuje jednoduchou formou spravovať štruktúru a obsah webovej stránky. Jej úlohou je, aby sme obsah mohli tvoriť jednoducho, často bez podrobných znalostí webových jazykov ako HTML, CSS, PHP a pod. Aj typický blog a wiki sú vlastne CMS systémy;

**syndikácia webového obsahu** (Really Simple Syndication, **RSS**): RSS súbor veľmi stručne opisuje obsah stránky. RSS mávajú hlavne stránky, ktoré sa rýchlo a pravidelne menia (spravodajstvo, blogy). RSS nám umožňuje mať prehľad o tom, či sa na webe deje niečo nové a zaujímavé bez toho, aby sme stále prechádzali a kontrolovali množstvo stránok. Sledujeme len svoju RSS čítačku...

## Čo sme sa naučili

Stručne sme sa oboznámili so základmi prvkami architektúry webu: s doménovými menami a s ich prekladom na IP adresy (DNS), s formátom webových adries (URI), s protokolom na prístup k webovým dokumentom (HTTP), s hypertextovým a hypermediálnym webovým obsahom, ako aj s princípom webového programovania a webových aplikácií, ale aj s ďalšími príbuznými technológiami. Všetky tieto prvky dohromady web tvoria, podmieňujú jeho fungovanie a prispievajú k jeho rôznorodosti. Ich uchopenie je kľúčom k pochopeniu podstaty, možností a obmedzení, ktoré nám web a publikovanie na webe prinášajú.

**Bezpečnosť webových aplikácií** je závažný problém. Je to predovšetkým vďaka značnej rozmanitosti webových technológií. Bezpečnostné problémy môžu byť spojené s kódom bežiacim na strane servera, ale aj na strane klienta. Často sa stretávame s nedostatočným zabezpečením pri interakcii s databázou. Prítom webové aplikácie často uchovávajú a spracovávajú dáta, ktoré im s dôverou zverili používatelia. Bezpečnosť je preto o to citlivejší problém.



Užitočné informácie o bezpečnosti webových aplikácií a nástroje na testovanie bezpečnosti nájdeme napr. na webe združenia OWASP: <http://www.owasp.org/>

XML (eXtensible Markup Language) bolo navrhnuté ako syntaktická platforma pre rôzne formáty v kontexte webu s cieľom nahradiť staršiu technológiu SGML (Standard Generalized Markup Language), na ktorej bolo HTML pôvodne založené. A tak sa z HTML 4.01, postaveného nad SGML, stalo XHTML 1.0, postavené nad XML. XML je oveľa jednoduchšie a „prísnejšie“ než SGML. Výsledkom je, že všetky XML-formáty majú jednotnú syntax. Výhodou XML je napr. to, že na všetky XML formáty nám stačí jeden XML parser. Rôzne vývojové prostredia dnes XML parser bežne obsahujú ako súčasť štandardných knižníc. XML sa postupne stalo veľmi populárnym; v súčasnosti predstavuje *de facto štandard* pre ľubovoľné dáta, ktoré nie sú inherentne binárne. Okrem XHTML medzi XML formáty patria SVG (pozri Kapitolu 5), MathML, ale napr. aj dátové formáty kancelárskeho balíku OpenOffice.org, a iné.

**Na rozdiel od HTML v XHTML vždy musíme:**

- všetky elementy vždy uzatvoriť;
- elementy nesmieme prekryvať;
- hodnoty atribútov musia byť vždy v úvodzovkách (buď „...“ alebo '...');
- „<“, „>“ a „&“ vždy nahradiť špeciálnym znakom (pozri nasl. stranu);
- mená elementov a atribútov vždy písať malými písmenami.



Správnosť (validitu) našich (X)HTML dokumentov si môžeme overiť pomocou **on-line validačnej služby**, ktorú poskytuje W3C: <http://validator.w3.org/>

## Kapitola 3: HTML a XHTML

Hypertextový a hypermediálny obsah je najeseniciálnejšou podstatou webu. Základný typ webového obsahu predstavujú hypertextové dokumenty v jazyku **HTML** (HyperText Markup Language). Ide o pomerne jednoduchý jazyk, umožňujúci prakticky každému vytvoriť vlastný dokument s odkazmi na ľubovoľný iné webové dokumenty. Jediné, čo k tomu potrebujeme je textový editor. HTML radíme medzi tzv. značkovacie jazyky. Tieto fungujú na jednoduchom princípe, v textovom dokumente nám umožňujú pomocou špeciálnych syntaktických značiek vyznačiť časti s odlišným formátovaním. V súčasnej dobe najpoužívanejšia verzia je XHTML 1.0, založená na jazyku XML. XHTML sa od HTML líši len veľmi málo. V ďalšom používame výraz (X)HTML, keď chceme povedať, že niečo platí pre obe verzie.

### Štruktúra HTML dokumentov

Každý HTML dokument je textový súbor. Začína sa **deklaráciou typu dokumentu** (DTD), ktorá určuje, v ktorej verzii HTML je dokument zapísaný. Špecifikáciu príslušnej verzie je následne nutné striktné dodržať. XHTML dokumenty obsahujú ešte aj **XML hlavičku**, v ktorej je vyznačené kódovanie dokumentu, a musí povinne obsahovať deklaráciu tzv. **namespace** priradeného k XHTML. Tieto časti (uvádzame ich v nasledovnom výpise) nie sú v tejto chvíli príliš podstatné, dôležité je ich **vždy v dokumente uviesť**.

#### HTML 4.01:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN">
<HTML>
  <!-- Obsah dokumentu -->
</HTML>
```

#### XHTML 1.0:

```
<?xml version="1.0" encoding="UTF-8"?>
<!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">
  <!-- Obsah dokumentu -->
</html>
```

*Inicializácia HTML a XHTML dokumentu. Znak „<!--“ a „-->“ označujú komentár*

(X)HTML dokument ďalej obsahuje text, a keďže ide o značkovací jazyk, môžu sa v ňom nachádzať značky. Značkám v (X)HTML hovoríme **tagy** a sú vždy uzavreté v lomených zátvorkách „<“ a „>“, napr. tag <html> v príklade vyššie. Tagy slúžia na vyznačenie **elementov**. Element sa začína otváracím tagom, napr. <html> a ukončený je uzatváracím tagom, ktorý má rovnaké meno ako otvárací tag, ale začína sa navyše znakom „/“, teda v tomto prípade </html>. Medzi otváracím a uzatváracím tagom je **obsah elementu**, ktorým môže byť buď text, alebo text obohatený o ďalšie elementy. Kým v HTML sa môžu niekedy vyskytovať aj neuzatvorené elementy, v XHTML musíme element vždy uzavrieť. Pokiaľ je XHTML element prázdny, môžeme miesto uzatváracieho tagu vložiť znak „/“ nakoniec tagu otváracieho, napr. <br/>. Elementy môžu mať parametre, ktorým hovoríme **atribúty**, a sú vždy súčasťou otváracieho tagu. Element s dvoma atribútmi vyzerá napr. takto: .

Keďže elementy môžu byť do seba vnorené, celý dokument má **stromovitú štruktúru**. Vrcholmi v tomto strome sú jednotlivé elementy, pričom koreňom stromu je vždy element html. Tento element obsahuje vždy dva elementy head a body, ktoré rozdeľujú dokument na dve časti - **hlavičku** a **telo**. Hlavička (X)HTML dokumentu obsahuje povinne názov dokumentu (element title). A môže tiež obsahovať metadáta rôzneho typu (viď nižšie).

Telo (X)HTML dokumentu obsahuje samotný viditeľný obsah dokumentu. Formátovanie možno robiť za pomoci širokého spektra elementov, ktoré majú predovšetkým **štruktúrálnu** funkciu - vyznačujú, akú má dokument štruktúru: kde je nadpis, kde



odstavce, kde odkaz, kde tabuľka, ale neslúžia na to, aby sme opísali ako presne majú tieto prvky vyzerat' (túto časť má na starosti samostatný jazyk CSS, o ktorom pojednáva Kapitola 4). Príklad veľmi jednoduchého XHTML dokumentu:

#### hello-world.html

```
<?xml version="1.0" encoding="UTF-8">
<!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>
    <title>Náš prvý XHTML dokument</title>
  </head>
  <body>
    <h1>Ahoj Svet!</h1>
    <p>Toto je prvý odstavce nášho prvého dokumentu</p>
  </body>
</html>
```

#### Náš prvý XHTML dokument

## Metadáta

Aj (X)HTML dokument môže obsahovať metadáta, a mal by vždy obsahovať aspoň nutné minimum metadát. Metadáta sa v okne prehliadača nezobrazujú. Predstavujú napr. informácie nutné na správne spracovanie dokumentu, ako kódovanie a jazyk, v ktorom je napísaný text dokumentu, či informácia o dobe platnosti dokumentu. Pre zápis metadát existujú nasledovné štyri možnosti:

**názov dokumentu:** obsahuje ho povinný element `title` v hlavičke dokumentu. Názov je veľmi dôležitý. Zobrazuje sa napr. v záhlaví okna prehliadača, ale aj vo výsledkoch vyhľadávania webovými vyhľadávačmi;

**elementy meta:** nachádzajú sa v hlavičke a možno pomocou nich zachytiť väčšinu bežne používaných metadát: kódovanie a jazyk dokumentu, dátum expirácie, kľúčové slová a krátku anotáciu, ale aj meno autora, a ďalšie;

**jazyk a kódovanie:** okrem elementov meta (nepovinné), je nutné jazyk povinne vyznačiť aj atribútom `lang` elementu `html` a v XHTML aj atribútom `xml:lang`. V XHTML musíme kódovú sadu dokumentu vyznačiť v XML hlavičke pomocou atribútu `encoding`;

**previazanosť dokumentu s ďalšími súbormi:** pomocou elementu `link` v hlavičke dokumentu môžeme prilinkovať napr. kaskádový štýl, ktorý má byť aplikovaný na dokument, či RSS. Skripty, ktoré sa majú v dokumente použiť, môžeme prilinkovať pomocou elementu `script`, opäť v hlavičke dokumentu.

V nasledujúcom výpise z XHTML dokumentu sú vyznačené základné metadáta:

#### metadata.html

```
<?xml version="1.0" encoding="UTF-8">
<!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>
    <title>Náš prvý XHTML dokument</title>
    <meta name="expires" content="fri, 21 Dec 2012"/>
    <meta name="description"
      content="Dôležitý príklad ako pracovať s metadátami"/>
    <meta name="keywords"
      content="XHTML, príklad, metadáta"/>
  </head>
  <!-- Obsah dokumentu -->
</html>
```

#### XHTML dokument s metadátami



Screenshot dokumentu.

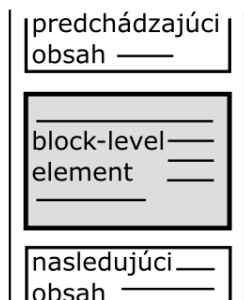
(X)HTML definuje sadu špeciálnych znakov, ktorým hovoríme aj entity. Entity začínajú znakom „&“ a končia znakom „;“.

Niektoré najpoužívanejšie entity sú:

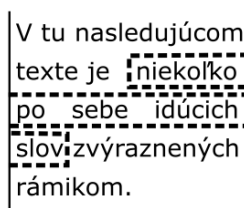
- `&lt;`; - znak „<“
- `&gt;`; - znak „>“
- `&amp;`; - znak „&“
- `&nbsp;`; - nedeliteľná medzera
- `&shy;` - delenie slov na konci riadku (nepodporujú všetky prehliadače)
- `&copy;` - znak „©“
- `&ndash;` a `&mdash;`; pomlčky „–“ a „—“

## Grafická reprezentácia a vykreslenie dokumentu

HTML bolo špeciálne navrhnuté za účelom poskytnúť dostatočne flexibilný formát pre hypertextové dokumenty publikované na internete. Internet je ale veľmi heterogénnym prostredím a HTML je tomu plne prispôsobené. Okrem možnosti obohatiť dokument o metadáta, ktoré pomôžu napr. pri indexovaní dokumentu vo vyhľadávачoch, to vplýva aj na režim vykresľovania dokumentu na obrazovke. Používatelia pristupujú k webu za pomoci rozličných počítačov a zariadení, ktoré môžu mať rôzne rozlíšenie a farebnú hĺbku obrazovky, nainštalovaný rôzny operačný systém, prehliadač, či sadu predinštalovaných fontov. Nedá sa teda s istotou povedať, aký veľký priestor bude na obrazovke webovým stránkam vyhradený - aká má byť šírka, či výška strany. Autori HTML preto prišli so šalamúnskym riešením: webová stránka nemá strany, ale je taká dlhá, ako treba (koľko obsahu obsahuje), a posúvame sa po nej pomocou rolovania v okne prehliadača. Horizontálnemu rolovaniu sa ale autori HTML snažili vyhnúť, aby sme pri čítaní nemuseli nikdy opätovne rolovať na začiatok a potom na koniec každého riadku. HTML dokumenty sú preto presne také široké, koľko miesta je v okne prehliadača, a keď zmeníme šírku okna, stránka sa prispôsobí a prekreslí. Je nutné poznamenať, že pomocou CSS je možné takéto správanie vypnúť, čo sa v praxi aj často využíva. Môžeme ho vidieť napr. na stránkach Wikipédie (<http://sk.wikipedia.org/>).



Blokový element.



Riadkový element.

Elementy, ktorých obsah je viditeľný (teda takmer všetky elementy, ktoré sa vyskytujú v tele dokumentu) a využívame ich za účelom formátovania dokumentu, nazývame tiež **vizuálne** elementy. Takýmito elementmi sú napr. h1 a p v príklade vyššie zodpovedajúce nadpisu a odstavcu. Vizualne elementy rozdeľujeme na základe spôsobu ich vykresľovania do dvoch skupín:

**blokové (block-level) elementy:** zaberajú na obrazovke miesto obdĺžnikového tvaru, pričom nasledujúci element sa zaradi vždy pod element predchádzajúci. Patria sem napr. elementy pre nadpisy a odstavce (h1 a p, pozri nižšie);

**riadkové (inline) elementy:** ukladajú sa za sebou v riadku. Pokiaľ element už nemôže v riadku pokračovať, dôjde k zalomeniu a element pokračuje na nasledujúcom riadku. Nasledujúci riadkový element sa v riadku radí vždy vpravo od predchádzajúceho riadkového elementu. Riadkovými elementmi sú napr. elementy em, strong, či a, s ktorými sa zoznámime nižšie.

V nasledujúcej časti tejto kapitoly sa podrobnejšie oboznámime s najdôležitejšími možnosťami a vlastnosťami (X)HTML ako aj s jednotlivými elementmi, ktoré (X)HTML za účelom zabezpečenia tej ktorej funkcionality obsahuje.

## Členenie a formátovanie textu

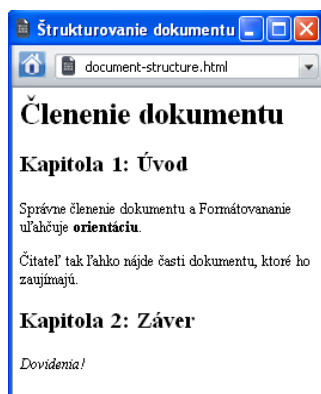
Základné členenie textu vykonáme vyznačením nadpisov a odstavcov v dokumente. (X)HTML ponúka **šesť úrovní nadpisov**, elementy h1, h2, ..., h6. Každý ucelený text by mal byť členený do **odstavcov** (element p), element br (nový riadok) by sa mali používať len výnimočne. Všetky tieto elementy vidíme vo výpise:

document-structure.html (výňatok)

```
<h1>Členenie dokumentu</h1>
<h2>Kapitola 1: Úvod</h2>
<p>Správne členenie dokumentu a Formátovanie uľahčuje
  <strong>orientáciu</strong>.
</p>
<p>Čitateľ tak ľahko nájde časti dokumentu, ktoré ho zaujímajú.</p>
<h2>Kapitola 2: Záver</h2>
<p><em>Dovidenia!</em></p>
```

*Ukážka správneho štruktúrovania (X)HTML dokumentu*

Pre **zvýrazňovanie v texte** používame predovšetkým elementy strong (silný dôraz, zobrazí sa tučným) a em (dôraz, zobrazí sa kurzívou). Existuje aj element b (tučné písmo) a element i (kurzíva), ich významom je však zmena rezu písma, *nie zvýraznenie*, preto je lepšie sa im vyhnúť. Tiež sa neodporúča zvýrazňovať elementom u (podčiarknuté), pretože by to čitateľ mohol vnímať ako hypertextový odkaz, čo by



Screenshot dokumentu.

naňho pôsobilo zbytočne mäťúco. Existuje aj prečiarknuté písmo (element strike). Neproporcionálny font zapneme elementom code, ktorý použijeme v prípade, že ide o fragment zdrojového kódu, ak nejde o zdrojový kód, použijeme element tt.

Už sme sa naučili, že šírka webovej stránky sa prispôsobuje šírke okna. (X)HTML pri vykresľovaní dokumentu ignoruje všetky znaky konca riadku v zdrojovom kóde a riadky automaticky zalamuje podľa potreby danej šírky okna prehliadača alebo šírky elementu, ak je nejakým spôsobom určená. Rovnako sú ignorované viaceré znaky medzier nasledujúce za sebou, a vykreslená je vždy len jedna medzera. K **zalomeniu riadku** dôjde na konci odstavca vyznačeného elementom p, prípadne môžeme riadok v prípade potreby zalomiť vložením prázdneho elementu br. Niekedy je však takéto správanie nežiaduce a kontraproduktívne, napr. pri sadzbe výňatkov zo zdrojových kódov programov, či pri sadzbe poézie vo veršoch. V takomto prípade môžeme využiť element pre - **preformátovaný text**. Pokiaľ text zabalíme do tohto elementu, (X)HTML bude rešpektovať všetky medzery a a konce riadkov.

## Vkladanie obrázkov, objektov a hypertextových odkazov

**Vkladanie obrázkov** do dokumentu umožňuje element img. Má dva povinné atribúty: atribút src udáva URI adresu súboru s obrázkom, a atribút alt udáva textovú alternatívu, t.j. krátky textový popis obrázku. Za pomoci elementu object možno vkladat' aj iné **multimediálne objekty**, ich zobrazenie v prehliadači je však závislé od nainštalovania príslušného pluginu.

**Hypertextové odkazy** vytvárame pomocou elementu a. Atribút href obsahuje URI adresu dokumentu, kam odkaz smeruje, v atribúte title môžeme voliteľne uviesť vysvetľujúci text, ktorý sa zobrazí „v bubline“, keď ponad odkaz prejdeme kurzorom myši. Element a tiež slúži na vyznačenie miesta v dokumente, na ktoré sa možno odkazovať cez tzv. fragmentový identifikátor. Toto miesto sa volá kotva (angl. anchor). Pokiaľ ho chceme vytvoriť použijeme atribút name (starší spôsob) alebo id (novší spôsob) tak, ako to vidíme vo výpise kódu nižšie. Môžeme v ňom vidieť niekoľko druhov odkazov.

links-and-images.html (výňatok)

```
<p>Skok na <a href="#obr"> obrázok nižšie</a> a odkaz na  
  <a href="http://www.sme.sk/"> úplne inú stránku</a>  
</p>  
<p><a name="obr" id="obr"/>  
  <a href="http://hlavicka.sk/" title="Jurkova stránka">  
      
    <br/>Moje foto  
  </a>  
</p>
```

Vkladanie obrázkov a odkazov

## Zoznamy a tabuľky

(X)HTML pozná tri druhy zoznamov: číslované, nečíslované a definičné zoznamy. **Číslovaný zoznam** predstavuje element ol, **nečíslovaný zoznam** element ul, pričom v oboch prípadoch položky zoznamu predstavuje séria elementov li. Definičné zoznamy zapuzdruje element dl, pričom zoznam tvorí striedavá séria elementov dt (pojmem, ktorý sa definuje) a dd (definícia pojmu). Zoznamy možno do seba vnárať.

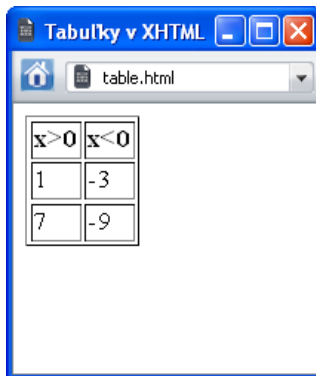
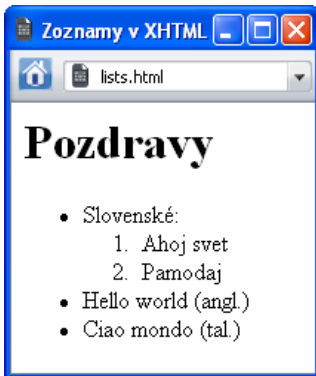
Existuje tiež pomerne komplexná sada elementov pre tvorbu **tabuliek**. Tabuľku zapuzdruje element table. Základné členenie tabuľky vykonáme elementmi caption (titulok), thead (záhlavie), tbody (telo tabuľky), tfoot (päta). Tri posledné z týchto elementov obsahujú riadky tabuľky (element tr), ktoré obsahujú bunky tabuľky (element td, prípadne th, ak ide o bunky záhlavia). Atribúty rowspan a colspan umožňujú vytvoriť bunky rozprestierajúce sa cez niekoľko riadkov, či stĺpcov. Tvorbe tabuliek je nutné venovať zvýšenú pozornosť. Tabuľky by sa napr. nikdy nemali používať na tvorbu dizajnu, a pod. (viac v Kapitole 5).

**Nadpisy musíme korektne vnárať:** hlavný nadpis dokumentu je h1 a mal by byť práve jeden, pri ďalšom štruktúrovaní dokumentu sa postupne vnárame do hlbších úrovní nadpisov, ale nikdy by sme nemali žiadny preskočiť (napr. zaradiť h4 ako podnadpis nadpisu h2). Úroveň nadpisu by sme nikdy nemali voliť podľa toho, ktorý nadpis sa nám zdá „primerane veľký“ - za týmto účelom môžeme veľkosť písma v nadpisoch zmeniť pomocou CSS. Je to potrebné kvôli *prístupnosti* (pozri Kapitolu 7).

**Tvorba kotvy:** v budúcnosti sa prejde na atribút id, ale niektoré prehliadače ho stále nepodporujú, preto sa v súčasnosti odporúča použiť obidva atribúty súčasne s rovnakou hodnotou (pozri príklad).



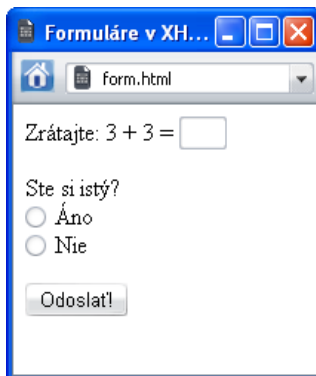
Screenshot dokumentu.



Screenshots oboch dokumentov.

Tlačidlo **Reset** sa neodporúča používať. Ak naň používateľ klikne omylom, príde o všetku prácu...

Existujú aj ďalšie formulárové prvky, napr. `select` a `textarea`, ich použitie je však nad rámec tohto textu.



Screenshot dokumentu. Vyplnením odpovede „7“ a označením políčka „Áno“ bude vygenerovaná požiadavka na: [http://moja.sk/anketa.php?answer=7&sure=yes&correct\\_answer=6](http://moja.sk/anketa.php?answer=7&sure=yes&correct_answer=6)

lists.html (výňatok)	table.html (výňatok)
<pre>&lt;h1&gt;Pozdravy&lt;/h1&gt; &lt;ul&gt;   &lt;li&gt;Slovenské:     &lt;ol&gt;       &lt;li&gt;Ahoj svet&lt;/li&gt;       &lt;li&gt;Pamodaj&lt;/li&gt;     &lt;/ol&gt;   &lt;/li&gt;   &lt;li&gt;Hello world (angl.)&lt;/li&gt;   &lt;li&gt;Ciao mondo (tal.)&lt;/li&gt; &lt;/ul&gt;</pre>	<pre>&lt;table border="1" summary="Čísla"&gt;   &lt;thead&gt;     &lt;tr&gt;       &lt;td&gt;x&amp;gt;0&lt;/td&gt;&lt;td&gt;x&amp;lt;0&lt;/td&gt;     &lt;/tr&gt;   &lt;/thead&gt;   &lt;tbody&gt;     &lt;tr&gt;&lt;td&gt;1&lt;/td&gt;&lt;td&gt;-3&lt;/td&gt;&lt;/tr&gt;     &lt;tr&gt;&lt;td&gt;7&lt;/td&gt;&lt;td&gt;-9&lt;/td&gt;&lt;/tr&gt;   &lt;/tbody&gt; &lt;/table&gt;</pre>

Zoznamy a tabuľky. Všimnime si tiež entity „&gt;“ a „&lt;“

## Formuláre

Už skôr sme sa naučili, že dotazová časť URI adresy tvoriacej HTTP požiadavku, ako aj dáta odoslané metódou POST sú vstupom pre webové aplikácie. V praxi je však potrebné získať tento vstup nejakým spôsobom od používateľa, aby sme nemuseli do prehliadača zadávať komplikované URI adresy. (X)HTML nám preto ponúka sadu formulárových elementov, ktoré predstavujú **rozhranie** medzi používateľom a webovou aplikáciou. Z hodnôt, ktoré používateľ zadá do formulára, prehliadač vyskladá URI adresu v zodpovedajúcom tvare a požiadavku odošle webovému serveru. Formulár je možné odoslať aj metódou POST, kedy sa dáta odošlú v tele správy.

**Formulár** vytvoríme za pomoci elementu `form`, do ktorého vložíme jednotlivé formulárové prvky. Tieto môžeme naformátovať, podobne ako text. Pomocou atribútov `action` a `method` určíme URI adresu server-side skriptu, kam majú byť dáta odoslané a metódu, ktorá sa má pritom použiť. Väčšinu **formulárových prvkov** vytvoríme pomocou elementu `input`, pričom atribútom `type` vyznačíme, o aký prvok sa jedná. Základné hodnoty sú: `text`, `checkbox`, `radio`, `hidden` (viď príklad nižšie). Hodnoty `submit` a `reset` predstavujú tlačidlá na odoslanie formuláru, či návrat k prednastaveným hodnotám. Atribút `name` je meno parametra v URI adrese, ktorému sa pomocou daného prvku priraduje hodnota, atribút `value` je voliteľná prednastavená hodnota. Pozrime si príklad jednoduchého formulára:

form.html (výňatok)
<pre>&lt;form action="http://moja.sk/anketa.php" method="get"&gt;   &lt;p&gt;Zrátajte: 3 + 3 =     &lt;input type="text" name="answer" size="2"/&gt;   &lt;/p&gt;&lt;p&gt;     Ste si istý?&lt;br/&gt;     &lt;input type="radio" name="sure" value="yes"/&gt; Áno&lt;br/&gt;     &lt;input type="radio" name="sure" value="no"/&gt; Nie   &lt;/p&gt;   &lt;input type="hidden" name="correct_answer" value="6"/&gt;   &lt;input type="submit" value="Odoslať!"/&gt; &lt;/form&gt;</pre>

Jednoduchý formulár

## Čo sme sa naučili

Zoznámili sme sa s jazykom HTML a s jeho modernejšou verziou XHTML, ktoré slúžia na zápis hypertextových dokumentov na webe. Naučili sme sa, akú majú (X)HTML dokumenty syntax a štruktúru. Rozlišujeme pojmy *tag*, *element*, a *atribút*. Vieme, že vykresľovanie (X)HTML v prehliadači je flexibilné a prispôsobuje sa možnostiam obrazovky a šírke okna. Podrobnejšie sme sa zoznámili so základnou funkcionalitou: vieme vytvárať odstavce, nadpisy aj podnadpisy, vkladať do dokumentov obrázky aj hypertextové odkazy, vieme vytvárať zoznamy a tabuľky. Stručne sme sa oboznámili aj s formulármi.

## Kapitola 4: Kaskádové štýly

Štandard CSS (Cascading Style Sheets, tiež *kaskádové štýly*), priniesol do webdizajnu možnosť oddeliť **obsah** webových dokumentov od **prezentácie**. Pod prezentáciou tu rozumieme predovšetkým vizuálne grafické spracovanie dokumentu a jeho typografiu. Oddelením obsahu od prezentácie sa stala tvorba (X)HTML dokumentov jednoduchšou a prehľadnejšou. CSS zároveň umožňuje ďaleko väčšiu kontrolu nad prezentáciou dokumentu, a tým aj ďaleko väčšiu slobodu pri návrhu prezentácie, než bolo kedy možné prostredníctvom HTML. (Po zavedení CSS bola časť funkcionality súvisiaca výhradne s prezentáciou z HTML postupne odobraná.) Vďaka týmto skutočnostiam spôsobilo CSS doslova revolúciu vo webdizajne a je dnes základným nástrojom webdizajnu.

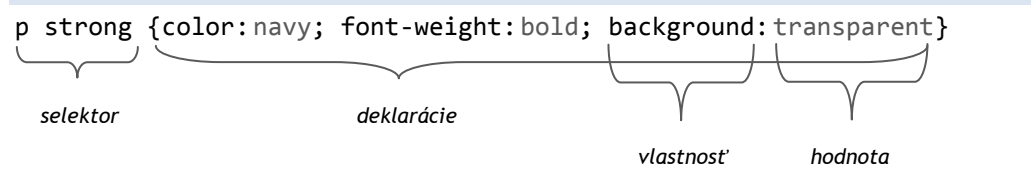
V tejto kapitole sa zoznámime s princípom fungovania CSS a s jeho základmi, a naučíme sa upraviť vzhľad a typografiu jednoduchej webovej stránky.

### Princípy CSS

(X)HTML je vybavené preddefinovaným vizuálnym a typografickým štýlom, ktorý je úmyselne veľmi strohý a konzervatívny. Toto preddefinované zobrazovanie našej webovej stránky v prehliadači budeme pomocou CSS meniť. Základnú funkcionality (X)HTML môžeme dokonca vo výraznej miere rozšíriť. Tú, ktorá sa nám hodí, ponecháme, inú zmeníme.

CSS je **deklaratívny jazyk**, s jednoduchou a ľahko čitateľnou syntaxou. CSS súboru hovoríme **šablóna** alebo tiež **štýl** (style sheet). Podobne ako (X)HTML dokumenty, aj CSS šablóny sú vlastne textové súbory. Šablóna pozostáva z **pravidiel**. Pravidlo sa skladá z dvoch častí. Prvá je podmienka, ktorej hovoríme **selektor**, a ktorá určuje, na ktoré elementy v dokumente bude pravidlo aplikované. Druhá časť je séria **deklarácií**, uzavretá v zložených zátvorkách, z ktorých každá mení nejakú vlastnosť cieľového elementu. Príklad CSS pravidla vidíme vo výpise:

```
pravidlo.css
p strong {color:navy; font-weight:bold; background:transparent}
```



#### Anatómia CSS pravidla

Selektor pravidla hovorí, že pravidlo bude aplikované na každý element `strong`, ktorý je v strome dokumentu nasledovníkom nejakého elementu `p`, t.j. nachádza sa vo vnútri niektorého odstavca. Na každý takýto element budú aplikované tri deklarácie: jeho farba sa zmení na tmavomodrú, font bude tučný, a pozadie bude priesvitné. Ako vidíme, každá deklarácia sa skladá z dvoch častí, z názvu **vlastnosti**, ktorú mení, a z **hodnoty**, ktorú danej vlastnosti priraduje. Vlastnosti môžu mať rôzne hodnoty, napr. kľúčové slová, dĺžkové údaje, čísla, farebné kódy, textové reťazce, alebo URI adresy. Pre každú vlastnosť je v špecifikácii presne povedané, aké hodnoty sú prípustné, a aká hodnota je preddefinovaná.

CSS je veľmi pružný jazyk. Umožňuje nám vytvárať pravidlá, ktoré budú aplikované na jediný element, ale zároveň aj pravidlá, ktoré formátujú naraz veľký počet elementov v dokumente. Všetko závisí do selektora. Podobne môže byť na jeden element aplikovaných viacero pravidiel - všetky, ktorých selektor je splnený na danom elemente. Napr. uvažujme, že okrem predchádzajúceho pravidla máme ešte pravidlo `strong {color: red}`. Pokiaľ sa v dokumente nachádza element `p` a v ňom element `strong`, aplikujú sa obe pravidlá. Aká však bude farba textu nášho elementu `strong`? Pokiaľ si pravidlá odporujú, víťazí to **špecifickejšie**. To, ktoré pravidlo je špecifickejšie, závisí od selektora, a v prípade niekoľkých zložitejších selektorov môže byť pomerne komplikované to určiť. Zjednodušene možno povedať, že čím zložitejší selektor, tým špecifickejšie pravidlo. Napr. aj v našom prípade „víťazí“ prvé pravidlo, lebo jeho selektor obsahuje hneď dva názvy elementov, teda je zložitejší.



Autorom CSS je **Håkon Wium Lie**, pričom CSS vzniklo ako jeho dizertačná práca. Pochopiteľne, aj tá je celá naformátovaná výlučne s použitím CSS. V súčasnosti pôsobí ako technologický riaditeľ spoločnosti Opera, ktorá vyvíja prehliadač s rovnakým menom.

Zaujímavá osobná stránka H. W. Lie: <http://people.opera.com/howcome/>

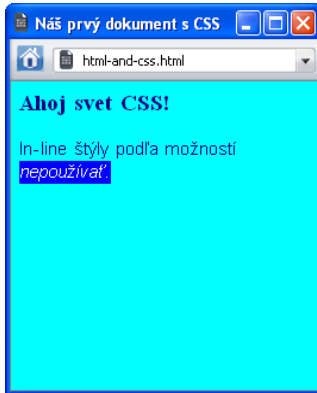
Momentálne je aktuálnou verziou **CSS 2.1**, a aj v tomto materiáli sa venujeme výlučne tejto verzii. Vývoj CSS bol v posledných rokoch zdĺhavý a závisel od toho, ktorým smerom sa boli ochotní uberať dodávateľia prehliadačov. Zdá sa však, že situácia sa pomaly mení a v horizonte niekoľkých rokov privítame nový štandard **CSS 3**, v ktorom sa pripravuje toľko zásadných zlepšení a zmien, že by mohol spôsobiť ďalšiu revolúciu vo webdizajne. Zaujímavý blog o CSS 3: <http://css3.info/>



Správnosť CSS kódu môžeme skontrolovať napr. online CSS validátorom, ktorý prevádzkuje W3C: <http://jigsaw.w3.org/css-validator/>

Samozrejme, špecifikácia to určuje presne a jednoznačne.

Aby sa formátovanie spustilo, musíme **pripojiť CSS šablónu k dokumentu**. Máme dve možnosti. Prvou je vložiť CSS kód priamo do (X)HTML dokumentu, v hlavičke. Kód vložíme do elementu `style`. Oveľa lepšou možnosťou je uložiť CSS šablónu do samostatného súboru a ten **prilinkovať** s využitím elementu `link`, opäť v hlavičke. Takto môžeme *tú istú šablónu opätovne využiť* na formátovanie viacerých dokumentov. Okrem týchto dvoch možností existuje v (X)HTML ešte atribút `style`, ktorý môže mať každý vizuálny element. Hodnotou atribútu nie sú pravidlá ale séria deklarácií oddelených bodkočiarkou. Takéto použitie, nazývané tiež **inline štýl**, je zo všetkých *najmenej odporúčané*; znovupoužiteľnosť kódu je tu fakticky nulová.



Screenshot dokumentu.

#### html-and-css.html (výňatok)

```
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="sk" lang="sk">
  <head>
    <title>Náš prvý dokument s CSS</title>
    <link rel="stylesheet" type="text/css" href="style.css"/>
    <style type="text/css">
      body { color: navy; background: aqua }
      h1 { font-size: 128% }
      p { font-family: Arial, Helvetica, sans-serif }
    </style>
  </head>
  <body>
    <h1>Ahoj svet CSS!</h1>
    <p>Inline štýly podľa možnosti
      <em style="color: white; background: blue">nepoužívať.</em>
    </p>
  </body>
</html>
```

#### HTML dokument s CSS

## Selektory

Selektor je, ako už vieme, časť pravidla, ktorá určuje, na ktoré elementy bude pravidlo aplikované. Základné druhy selektorov sú:

- selektor typu:** vyberá element podľa mena elementu; už sme sa s ním stretli v príkladoch vyššie. Občas sa hodí aj **univerzálny selektor**, znak „\*“, ktorý vyberá každý element;
- ID selektor:** vyberie element, ktorý má hodnotu atribútu `id` nastavenú na nejakú konkrétnu hodnotu. Takýto element môže byť v celom dokumente iba jeden. Selektor má tvar `#<hodnota>`;
- selektor triedy:** vyberá podľa hodnoty atribútu `class`, hodnotou ktorého môže byť medzerami oddelený zoznam *tried*. Má tvar `.<trieda>`. Je to jeden z najpoužívanejších a najpraktickejších selektorov;
- selektor pseudo-triedy:** vyberá element, ktorý je v určitom *stave*, v závislosti od akcie, ktorú vykonal používateľ. Tento selektor začína znakom „:“. Najdôležitejšie pseudo-triedy sú `:link` (nenavštívený odkaz), `:visited` (navštívený odkaz), `:hover` (element, nad ktorým je práve kurzor), a `:active` (element, nad ktorý používateľ drží stlačené tlačidlo myši).

V súvislosti so selektormi podľa ID a podľa triedy sa často hodia dva (X)HTML elementy `div` a `span`, z ktorých prvý je blokovým a druhý riadkovým elementom a inak už nemajú žiadne iné naviazané vizuálne formátovanie. Pomocou priradeného ID, ale predovšetkým pomocou tried možno z týchto elementov „vyrobiť“ nový element so špeciálnym formátovaním, aký v (X)HTML nie je.

Selektory je možné kombinovať a vytvárať tak **zložené selektory**. Zretazením viacerých selektorov *bez medzier* vzniká ich **konjunkcia**, t.j. vybrané budú len elementy, pre ktoré sú všetky selektory splnené. Pokiaľ oddelíme dva selektory medzerou, vznikne zložený selektor **potomka**, v takomto prípade sa musia v strome dokumentu nachádzať dva elementy, z ktorých druhý (splňajúci druhý selektor) je potomkom

prvého (splňajúceho prvý selektor), pričom výsledkom selekcie je druhý element. Existujú aj ďalšie kombinácie: **priamy potomok (>)** **nasledovník (+)**, tieto sú však slabšie podporované (nepodporuje IE6). Znak čiarka predstavuje **disjunkciu** selektorov.

Nasledujúci príklad predpokladá, že formátujeme XHTML dokument, ktorý obsahuje text a navigačné menu. Toto menu je zapuzdrené v elemente div, s nastaveným atribútom id="menu". Prvé tri pravidlá sú všeobecné a platia pre všetky odkazy v dokumente; definujú rôzne módy zobrazenia, pre navštívené a nenavštívené odkazy. Definujú tiež hover-efekt (pri prejdení kurzorom myši). Elementy v menu sú ale formátované odlišne, pričom rovnaký majú iba hover-efekt. Zmena je vďaka poslednému pravidlu so zloženým selektorom, ktoré má precedenciu pred prvými dvoma pravidlami. Zložený selektor využíva fakt, že položky v menu sú v strome potomkami elementu div s nastaveným atribútom id="menu".

#### selectors.css (výňatok)

```
a:link { color: blue; text-decoration: underline}
a:visited { color: gray }
a:hover { color: navy; background: aqua }
div#menu a:link, div#menu a:visited {
    color: blue;
    text-decoration: none
}
```

#### Príklad zložených selektorov

## Fonty, formátovanie textu, farby a pozadia

S **farbou textu a pozadia** sme sa už čiastočne oboznámili; vieme, že ich môžeme nastavovať vlastnosťami color a background. Hodnotou môžu byť kľúčové slová pre farby, napr. navy, lime, red, magenta, a ďalšie. Tých je však len obmedzený počet, pre ďalšie farby použijeme farebné kódy podľa farebného modelu RGB s 24-bitovou hĺbkou (pozri Kapitolu 5). Možné sú nasledovné dva tvary:

**#RRGGBB**: kde RR, GG a BB sú čísla v šestnástkovej sústave v rozsahu 00-ff;

**rgb(R,G,B)**: kde R, G a B sú desiatkové hodnoty s rozsahom 0-255.

Pomocou vlastnosti background je možné vložiť aj **obrázok na pozadí**, a to do ľubovoľného elementu. Možný je napr. nasledovný tvar:

#### Syntax vlastnosti background

```
background: [farba] [url("<URI_obrázka>")] [opakovať];
```

#### Jeden z možných tvarov vlastnosti background

kde [farba] je farebná hodnota, [url("<URI\_obrázka>")] je URI adresa zadaná pomocou „funkcie“ url(), a [opakovať] je jedno z kľúčových slov repeat (opakovať), repeat-x, repeat-y (opakovať len v osi X, resp. Y) a no-repeat (neopakovať).

Jednou z najdôležitejších vlastností písma je **voľba fontu** alebo tiež rodiny písma. V CSS na jej nastavenie slúži vlastnosť font-family, ktorej hodnota je čiarkami oddelený zoznam názvov fontov. Viac fontov je potrebné uviesť preto, lebo možnosť použitia každého z nich závisí od toho, či ich používateľ má nainštalované na svojom počítači alebo nie. Z toho istého dôvodu by poslednou položkou zoznamu malo **vždy** byť kľúčové slovo označujúce tzv. generickú rodinu. Hlavné **generické rodiny** sú tri: serif (pätkové písmo), sans-serif (bezpätkové písmo) a monospace (neproporcionálne písmo). Prehliadač by pre každú z nich mal mať k dispozícii nejaký font daného typu, pre prípad, že by žiadny z predchádzajúcich fontov v zozname nebol k dispozícii. Na webe sa najčastejšie využíva písmo bezpätkové (pozri tiež Kapitolu 6).

**Veľkosť písma** určuje vlastnosť font-size. Jej hodnotou je buď dĺžka, alebo lepšie percentuálna hodnota, teda napr. „120%“. Táto je relatívna vzhľadom na rodičovský element. Veľkosť písma by sme nikdy nemali nastavovať v absolútnych jednotkách

V CSS3 pribudne niekoľko ďalších praktických selektorov, ktoré nám uľahčia prácu. Už dnes si môžeme vyskúšať napr. selektory :nth-child a :target.

#### Príklady farebných kódov:

green	#008000
yellow	#ffff00
red	#ff0000

V CSS 3 bude pridaný aj farebný model HSL, ako aj verzie oboch modelov umožňujúce polopriehľadnosť: RGBA a HSLA. (Pozri nasledujúcu kapitolu).

Požiadavka, že font musí byť nainštalovaný v operačnom systéme používateľa je pre webových tvorcov značne obmedzujúca. Dôsledkom je, že na celom webe sa stretáme len s hŕstkou fontov. V CSS3 bude doplnená možnosť použiť vlastný font, ktorý tiež umiestnime na web, prehliadač si ho stiahne a použije (tzv. **web font**). Už dnes túto funkciu niektoré prehliadače podporujú.

(pozri nižšie), a ani by sme nemali nastavovať príliš malé písmo, napr. menej než 80% z prednastavenej hodnoty (pozri Kapitolu 6).

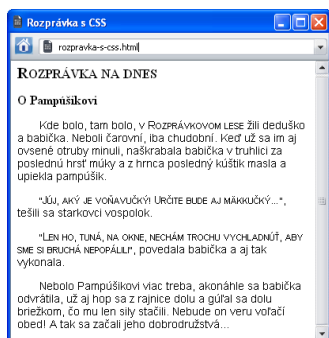
Ďalšie typografické vlastnosti písma a textu, ktoré máme prostredníctvom CSS pod kontrolou sú napr.:

**rez písma:** pomocou vlastností `font-style`, `font-weight`, `font-variant`, a `text-transform` môžeme prepínať kurzívu, tučné, kapitálky, ako aj rôzne kombinácie týchto rezov;

**podčiarknutie, nadčiarknutie a prečiarknutie:** vlastnosť `text-decoration` umožňuje zapnúť podčiarknutý, nadčiarknutý a prečiarknutý text;

**odsadenie odstavca:** odsadenie prvého riadku odstavca nám umožňuje zapnúť vlastnosť `text-indent`;

**zarovnanie textu:** pomocou vlastnosti `text-align` môžeme prepínať medzi zarovnaním na ľavý, pravý alebo na oba okraje súčasne. Možné je tiež text centrovat'. Na webe sa najviac využíva zarovnanie na ľavý okraj.



Screenshot vzorového dokumentu. Všimnime si zmenu písma, a to, že odstavce sú vertikálne odsadené, ako štandardne v HTML.

#### font-and-text.css (výňatok)

```
h1, h2 { font-family: "Times New Roman", "Nimbus Roman", serif }
h1 { font-size: 150% }
h2 { font-size: 120% }
h1, em { font-variant: small-caps; font-style: normal }
p { font-family: Arial, Helvetica, sans-serif; text-indent: 2em }
```

*Príklad typografického štýlu s dvoma fontmi a s kapitálkami*

## Box model, dĺžka a dĺžkové jednotky

Box modelom nazývame skupinu vlastností CSS, ktoré nám umožňujú **meniť rozmery elementov**. Z predchádzajúcej kapitoly vieme, že v (X)HTML sa (takmer) všetky elementy delia na *blokové* a *riadkové* elementy a podľa toho sa zobrazujú. Pomocou vlastnosti `display` môžeme medzi týmito dvoma módmi (hodnoty `block` a `inline`) prepínať: z riadkových elementov môžeme spraviť blokové a naopak. Pomocou hodnoty `none`, môžeme dokonca zobrazenie niektorého elementu úplne vypnúť.

Každý blokový element je na obrazovke reprezentovaný obdĺžnikovým tvarom, ktorému hovoríme **box**. Stadiaľ pomenovanie „box model“. Riadkové elementy generujú hneď niekoľko boxov, a síce na každom riadku jeden. Vlastnosti `width`, `min-width`, `max-width`, `height`, `min-height` a `max-height` nastavujú **šírku a výšku** blokových elementov. Dvojica vlastností `min-width` a `max-width` umožňuje „pružnú“ šírku, s ktorou sa stretávame aj v základnom formátovaní (X)HTML, ale s **vymedzenými hranicami**. Takýto dizajn je veľmi praktický. Naopak, stanovenie fixných širok cez vlastnosť `width` je problematické s ohľadom na malé displeje (na ktorých sa potom nevyhne horizontálnemu rollovaniu). Vlastnosti šírky a výšky môžu mať ako hodnotu buď dĺžku, alebo percentuálny údaj. Pokiaľ vlastnosť nenastavíme, element správa totožne ako pri základnom (X)HTML formátovaní.

U riadkových elementov sa vlastnosti šírky a výšky neaplikujú. Môžeme ich síce nastaviť, ale nemá to žiadny vplyv na zobrazenie dokumentu. Vlastnosťou `line-height` môžeme ale nastaviť **výšku riadku**.

Ďalšie vlastnosti patriace pod box model sú `border` (rámik), `margin` (okraj medzi rámkom a okolím) a `padding` (odstup medzi rámkom a obsahom elementu). Hodnotami pre `margin` a `padding` sú dĺžky. Vlastnosť `border` je o čosi zložitejšia:

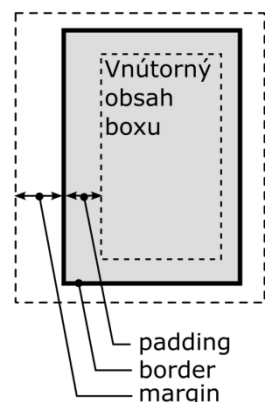
#### Syntax vlastnosti border

```
border: <šírka> <štýl> <farba>;
```

*Jeden z možných tvarov vlastnosti border*

Vlastnosti `border`, `margin` i `padding` možno nastavovať pre celý box jednotne, ale aj pre každú stranu boxu zvlášť. Ukážeme príklad s vlastnosťou `margin` (zvyšné dve fungujú analogicky):

- máme k dispozícii štyri samostatné vlastnosti: `margin-top`, `margin-left`,



Box model v CSS.



margin-bottom, margin-right;

- môžeme však tiež uviesť viacero hodnôt pri vlastnosti margin (oddelené medzerou), pokiaľ uvedieme len jednu, platí pre všetky štyri strany. Pokiaľ uvedieme dve, prvá platí pre vrchnú a spodnú stranu, druhá pre ľavú a pravú, pokiaľ uvedieme štyri, platia v poradí pre vrchná, pravá, spodná, ľavo.

Pri práci s box modelom často nastavujeme dĺžky. Dĺžkové údaje majú vždy tvar <N><u>, kde <N> je racionálne číslo a <u> dĺžková jednotka. Jediná dĺžka, ktorá môže byť bez jednotky je nulová. Podporované jednotky delíme na tri skupiny:

**absolútne:** jednotky cm, mm, in (palec, angl. *inch* = 2,54 cm), a typografické jednotky pt (bod, angl. *point* =  $\frac{1}{72}$  in) a pc (pika, angl. *pica* = 12 pt);

**relatívne vzhľadom k veľkosti písma:** jednotky em (šírka malého písmena „m“) a ex (výška malého písmena „x“);

**relatívne vzhľadom k zariadeniu:** jednotka px (pixel).

#### boxes.css (výňatok)

```
body { margin: 0; padding: 0 }
div#main { min-width: 10em; max-width: 30em;
            border: double aqua 6px; margin: 1ex; padding: 1ex }
h1 { padding-top: 3em; padding-bottom: 1em; padding-left: 1em;
      margin: 0; background: aqua }
h2 { padding: 0; margin: 1.5em 0.5em 0.5em 2em }
p { margin: 0.5ex 0 }
```

#### Upravené okraje a rámiky elementov pomocou box modelu

Výber správnych jednotiek je veľmi dôležitý. Vo všeobecnosti sa najviac odporúča používať jednotky em a ex. Problém s absolútnymi jednotkami, ale aj s pixelmi, je, že niektoré prehliadače neumožňujú dĺžky zadané cez tieto jednotky nijako zväčšiť (funkcia zoom). Predstavme si situáciu, že stránku pri jej tvorbe testujeme na typickom 17" LCD displeji s rozlíšením 1024×768 px. Písmo sa nám zdá príliš veľké, a tak jeho veľkosť neobozretne nastavíme na 9px. Podobne aj rozmery boxov na stránke nastavíme fixne v pixeloch. Keď stránku následne otvorí používateľ na modernom biznis-class laptope s 13" displejom s rozlíšením 1440×900px, celá stránka bude zmenšená a písmo bude až nečitateľne malé. Ak všade použijeme jednotky em a ex, prípadne percentá, veľkosť stránky i fontu sa bude odvíjať od prednastavenej veľkosti fontu, ktorú si v prehliadači nastavil sám používateľ. Vyjdeme tak najviac v ústrety používateľom našich webových stránok.

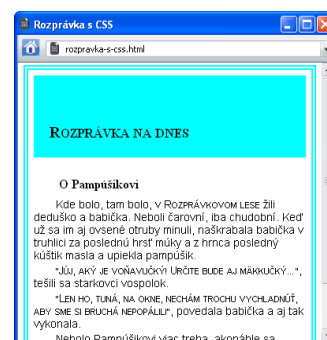
## Ďalšie možnosti CSS

Okrem vlastností CSS, ktoré sme si ukázali v tejto kapitole, má CSS ešte mnohé iné. Pomocou *vizuálneho formátovacieho modelu*, je možné vytvárať obtekané elementy, ba či dokonca elementy rozmiestňovať po celej stránke podľa ľubovôle. CSS tiež poskytuje bohatú sadu vlastností na úpravu vzhľadu zoznamov a tabuliek. Je možné dogenerovať obsah, ktorý v pôvodnom dokumente nie je, napr. číslovanie nadpisov. Dá sa špecifikovať iné formátovanie pre zobrazenie stránky v okne prehliadača a iné do tlače. O niektorých z týchto možností si povieme viac v module *Webové technológie a publikovanie na webe 2*.

## Čo sme sa naučili

Zoznámili sme sa s princípmi jazyka CSS, ktorý slúži na špecifikáciu prezentačnej vrstvy nad (X)HTML dokumentom. Výhoda CSS tkvie v tom, že prezentáciu dokumentu umožňuje zadefinovať oddelene od obsahu, jeho správnym používaním sa teda tvorba webových dokumentov stáva jednoduchšou a prehľadnejšou, a do jej procesu sa vnáša modularita a znovupoužitelnosť kódu starajúceho sa o prezentáciu dokumentu. Oboznámili sme sa tiež so základmi CSS: vieme vytvoriť CSS súbor a prilinkovať ho k HTML dokumentu, vieme písať jednoduché CSS pravidlá, vieme pracovať s farbami a s pozadím, nastavovať typografické vlastnosti písma a textu, a vieme pomocou box modelu meniť vzhľad jednotlivých elementov.

Ďalšou novinkou v CSS 3 bude možnosť robiť rámiky s oblými rohmi priamo pomocou vlastností rámika. Aj keď sú takéto rámiky veľmi populárne, v skutočnosti musia byť prevedené napr. vyskladaním z rastrových obrázkov, čo je veľmi nepraktické.



Do prechádzajúceho príkladu sme *doplnili* aj štýl boxes.css. Okrem orámovania si všimnime, že nastavením vlastnosti margin na menšiu ako je preddefinovaná sme docielili formátovanie odstavcov bez vertikálneho odsadenia.

## Kapitola 5: Multimédia

Kvalita a dostupnosť webových služieb, penetrácia rýchleho pripojenia do internetu umožňujú čoraz väčšiemu množstvu ľudí vytvárať a publikovať multimediálny obsah. Pôvodne bol internet nástrojom, pomocou ktorého veľmi úzka skupina autorov publikovala na internete informácie pre široké spektrum používateľov. Tento spôsob využívania sa však postupne mení. Nové webové technológie umožňujú, aby sa autorom stal každý z nás. Webové stránky poskytujú priestor pre vzájomné zdieľanie informácií a spoločnú tvorbu internetového obsahu. Internet už dávno neposkytuje len textové informácie, čoraz častejšie v ňom vyhľadávame a získavame informácie v grafickej, zvukovej či video podobe.

Táto, na prvý pohľad úžasná zmena využívania internetu však má aj svoje negatívne stránky. Internet je zahltený množstvom neidentifikovateľného a ťažko opísateľného obsahu, množstvom informačného balastu. Na rozdiel od textových informácií, ktoré sú pre počítače najľahšie spracovateľné, s informáciou v grafickej podobe je to už horšie. Informačná hodnota obrázku je pre počítače len ťažko identifikovateľná. Zvuková informácia je na tom ešte horšie. Aj keď počítače vedia porovnať dva zvuky, identifikovať hlas človeka vo zvukovom zázname, jeho obsahu nerozumejú. Publikovanie multimediálnej informácie by preto malo mať svoje opodstatnenie. Pokiaľ takáto forma nepriňa vyššiu informačnú hodnotu ako text, jej použitie je otáznе. Ak sa napokon rozhodneme informáciu zverejniť v multimediálnom formáte, nemali by sme zabudnúť k nej pripojiť dostatočne zrozumiteľný textový ekvivalent.

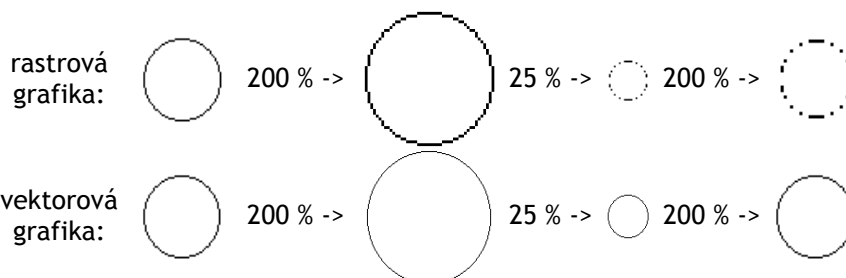
V tejto kapitole sa dozvieme základné pravidlá a postupy pre publikovanie multimediálneho obsahu na internete. Spoznáme niektoré nástroje pre správu a publikovanie multimédií, povieme si o rôznych formátoch informácií a naučíme sa ich používať.

### Grafika na webe

Informácia v grafickej podobe je zrejme najčastejšie sa vyskytujúcou formou informácie po textovej forme. Výkon počítačov a prenosové kapacity počítačových sietí nám umožňujú vybrať si z množstva formátov. Napriek tomu pre publikovanie na webe nie sú vhodné všetky formáty. Vo všeobecnosti môžeme počítačovú grafiku, podľa spôsobu jej reprezentácie, rozdeliť do dvoch skupín, na rastrovú a vektorovú. Pri **rastrovej grafike** je obraz reprezentovaný postupnosťou farebných obrazových bodov (pixelov). Každý bod má určenú svoju pozíciu a farbu. Pri zobrazení obrazu príslušná aplikácia vyskladá výsledný obraz postupným zobrazením všetkých bodov.

**Vektorová grafika** používa principiálne iný spôsob reprezentácie. Výsledný obraz je zložený z grafických objektov (čiary, krivky, obdĺžniky, a pod.). Každý objekt je reprezentovaný matematickým popisom. Tento popis možno chápať ako návodom na zostrojenie daného objektu. Napr. pre zobrazenie kruhu je potrebné zaznamenať súradnice stredu, polomer, farbu čiary, farbu výplne, a pod. Na základe týchto údajov dokáže aplikácia kruh vykresliť. Pri každom vykreslení vektorového obrázka, príslušný softvér vykreslí jednotlivé objekty na základe „návodu“.

Podľa toho, ako je grafika zobrazená na obrazovke počítača nemožno povedať, či ide o rastrovú alebo vektorovú grafiku. Rozdiel sa prejaví až pri manipulácii s danou grafikou. Pri rastrovej grafike dochádza k zmene na úrovni farebných bodov (farba, počet, umiestnenie). Pri vektorovej grafike sa upraví popis pre zostrojenie konkrétnych objektov. Jeden z dôsledkov demonštruje nasledujúci príklad:



Vidíme, že vektorovo reprezentovaný kruh možno na rozdiel od rastrovej zmenšovať a opätovne zväčšovať bez straty kvality. Z uvedeného príkladu by sme mohli unáhle-

Na webových stránkach <http://similar-images.googlelabs.com/> a <http://www.bing.com/> môžete vyhľadávať obrázky podľa ich vzájomnej podobnosti.

Rozumejú tieto vyhľadávacie stroje obsah obrázkov alebo ide o podobu čisto vizuálnu?

Spresniť výsledky vyhľadávania obrázkov pomáhajú vyhľadávaču Google tisíce dobrovoľníkov, tak že sa hrajú „hru“ *Google image labeler*: <http://images.google.com/imagelabeler/>



Rozdiel medzi rastrovou (hore) a vektorovou grafikou (dolu) je markantný po zväčšení.

ne usúdiť, že vektorová grafika je ideálnym spôsobom reprezentácie grafiky. V skutočnosti tomu tak ale nie je. Problém je v tom, že nie všetky obrázky vieme dostatočne efektívne popísať matematicky pomocou rôznych geometrických tvarov. Predstavme si napr. líuku plnú rozkvitnutých, pestrofarebných kvetov. Je takmer nemožné tento obraz popísať a už vôbec nie vyjadriť ho matematicky. *Kým vektorová reprezentácia sa hodí viac pre rôzne nákresy a kresby, napr. pri fotografiách víťazí jasne rastrová reprezentácia.*

Pri definícii farieb (grafických bodov, objektov) používame niektorý z farebných modelov. Pod pojmom farebný model rozumieme spôsob číselnej reprezentácie farieb. Napr. **farebný model RGB** definuje úroveň zastúpenia troch základných farieb, červenej (red), zelenej (green) a modrej (blue) vo výslednej farbe. Zastúpenie každej z farieb je vyjadrené hodnotou od 0 (minimálna úroveň) po 255 (maximálna úroveň). Napr. farba (0,0,255) je jasne modrá, farba (0,0,0) čierna a najjasnejšia farba (255,255,255) je biela. Rozšírením modelu RGB je **farebný model RGBA**. Okrem úrovne troch základných farieb definuje aj úroveň priehľadnosti, tzv. **alfa-kanál**.

Ludskému vnímaniu farieb je o niečo bližšie **farebný model HSL**. Ten definuje výslednú farbu pomocou troch parametrov: *odtieň* (hue), *sýtosť* (saturation) a *svetlosť* (luminosity). Tieto tri parametre vystihujú farbu tak, ako ju prirodzene vnímame. Podobne ako u RGB aj tu pridaním alfa-kanálu vzniká **farebný model HSLA**.

Aj keď existuje veľké množstvo rôznych grafických formátov, pre použitie na webe je vhodných len niekoľko málo spomedzi nich. Je to preto, že internetové prehliadače ostatné formáty nepodporujú (prípadne nie dostatočne). Toto má dobré opodstatnenie, keby mali podporovať veľké množstvo formátov, veľkosť týchto aplikácií by neúmerne narástla, ich vývoj by sa spomalil, a len veľmi ťažko by mohli fungovať aj na všetkých prenosných počítačoch a rôznych mobilných zariadeniach, s čím sa dnes bežne stretávame. Práve preto W3C požaduje, aby všetky prehliadače natívne podporovali iba nasledovné formáty: GIF, JPG, PNG a SVG (o všetkých sa podrobne zmienujeme nižšie), z ktorých sú dostatočne podporované zatiaľ len prvé tri.

**Natívna podpora** znamená, že prehliadač sám obsahuje všetky algoritmy a knižnice, potrebné na zobrazenie (prehratie, interpretáciu) formátu. Okrem natívnej podpory väčšina prehliadačov podporuje systém pluginov (tiež *zásuvných modulov*), ktorý umožňuje do prehliadača potrebnú funkcionálnosť pre spracovanie nových multimediálnych formátov.

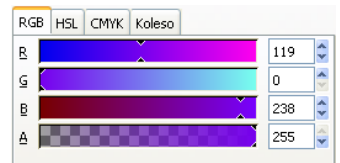
**Podpora pluginom** však nie je to isté ako natívna podpora. Plugin je programový kód, ktorý prehliadač spustí a odovzdá mu spracovanie dát v danom formáte. Plugin si používateľ musí nainštalovať, s čím nemôžeme automaticky počítať. Mnohí používatelia sú k pluginom od tretích strán nedôverčiví. Navyše príslušný plugin musí byť dostupný pre náš prehliadač a pre náš operačný systém (hlavne ak používame iný operačný systém ako Windows). Preto je vhodné vždy uprednostniť niektorý z natívne podporovaných formátov, pokiaľ je to čo i len trochu možné. Ak sa aj rozhodneme pre publikáciu v pluginom podporovanom formáte, mali by sme si vybrať čo najširšie podporovaný plugin. Zároveň, HTML nám umožňuje poskytnúť aj alternatívny obsah, ak plugin nie je dostupný, čo by sme podľa možností mali vždy využiť.

## Rastrová grafika

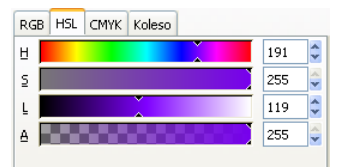
Pri voľbe grafického formátu rastrovej grafiky by sme mali zohľadniť vizuálnu stránku obrázku a vlastnosti daného formátu. Nesprávna voľba formátu môže značne znížiť výslednú kvalitu obrázku, prípadne nepriaznivo ovplyvniť jeho veľkosť.

V tabuľke na nasledujúcej strane si všimneme rozdiely medzi tromi najpoužívanejšími grafickými formátmi. Na obrázku zapadajúceho slnka vo formáte **GIF** môžeme vidieť nepekne vykreslenú oblohu. Je to spôsobené tým, GIF vie zobraziť len 256 rôznych farieb. Redukcia počtu farieb spôsobila, že podobné farby na oblohe boli nahradené len jednou farbou. Priemet hranola je naopak zobrazený bez rušivých elementov.

Formát **JPG** je na tom presne naopak. Zapadajúce slnko je zobrazené veľmi verne. Na hranole si môžeme všimnúť rozmazanie okolo čiar priemetu. Toto rozmazanie je spôsobené stratovou kompresiou ktorú formát JPG používa. Obrázok vo formáte **PNG** je na tom zrejme najlepšie. V oboch prípadoch je obrázok zobrazený kvalitne.



Mixovanie farieb pomocou farebného modelu **RGBA**.



Mixovanie farieb pomocou farebného modelu **HSLA**.

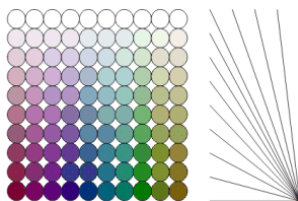
„Nútiť“ používateľa kvôli korektnému zobrazeniu nainštalovať na svoj počítač akýkoľvek nový softvér vrátane pluginov sa považuje za prehrášok voči použiteľnosti (pozri aj Kapitulu 7).



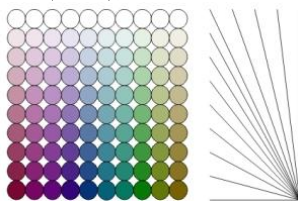
Pre úpravu rastrových obrázkov, predovšetkým fotografií, môžeme použiť open-source program **GIMP**.

Problematika grafických formátov je omnoho širšia než nám dovoľuje rozsah tohto materiálu. Výbornú sériu článkov na túto tému nájdeme na adrese: <http://www.root.cz/serialy/graficke-formaty/>

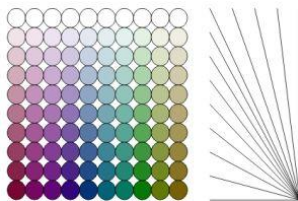
GIF, 18 kB



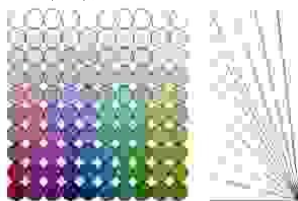
JPG (100%), 59 kB



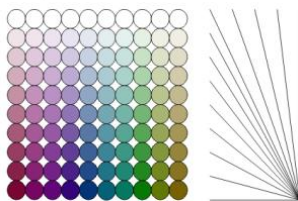
JPG (80%), 19 kB




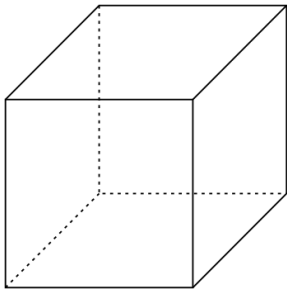

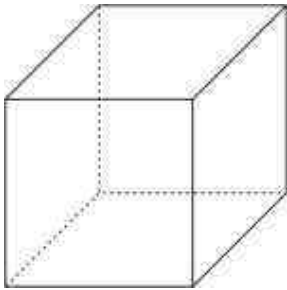

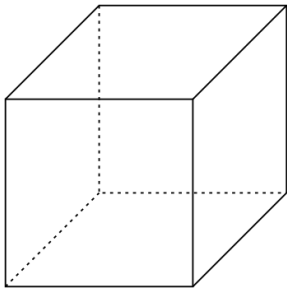
JPG (0%), 2 kB



PNG, 49 kB



Spoločnosť Unisis, ktorá vlastní patent na formát GIF svojho času požadovala licenčné poplatky od výrobcov softvérov, ktoré s formátom pracujú. Táto skutočnosť bola jedným z hlavných impulzov pre vznik formátu PNG. Tento patent medzitým expiroval.

formát PNG		
formát JPG		
formát PNG		

Predstavme si teraz jednotlivé formáty bližšie:

**GIF** (Graphic Interchange Format): obrázok môže mať maximálne 256 farieb (8-bitová paleta) a používa jednoduchú bezstratovú kompresiu (pri ukladaní nedochádza k strate kvality). Formát GIF sa používa pre kresby, diagramy, čiary, logá, jednoduché skice a pod. Umožňuje definovať jednu farbu ako presvitnú farbu, čo sa často využíva najmä na webových stránkach. Okrem toho umožňuje vytvárať animácie. V jednom súbore dát je uložených viac obrázkov, ktoré pri zobrazení vytvárajú animáciu. Štandardnou príponou je .gif.

**JPEG** (Joint Photographics Experts Group): obrázok má vždy farby TrueColor (24-bit). Pomocou zložitej stratovej kompresie (pri ukladaní dochádza k strate kvality) je možné dosiahnuť kompresný pomer až 1:20 bez viditeľnej straty kvality. Úroveň kompresie je možné nastaviť pri ukladaní súboru. Tento formát je vhodný najmä pre digitálne fotografie, kde sa aj najviac používa. Neodporúča sa používať ho pri kontrastnej grafike, kde aj malé straty kvality sú už pomerne dobre viditeľné. Štandardnou príponou je .jpg.

**PNG** (Portable Network Graphics, tiež „PNG is Not GIF“): bol navrhnutý ako prenositeľný formát v počítačových sieťach, ktorý mal nahradiť formát GIF. PNG podporuje 24-bitovú farebnú paletu. Ako jeden z mála grafických formátov ho nezaťažujú žiadne patenty (pozri stĺpec). Používa bezstratovú kompresiu, zväčša o niečo efektívnejšiu ako GIF. Okrem priehľadnej farby podporuje aj tzv. polopriehľadnosť (alfa-kanál). Do nedávna jeho väčšiemu rozšíreniu bránila slabšia podpora v niektorých prehliadačoch. Ak však odhliadneme od animácií, už dnes môžeme nazvať bezstratovým webovým grafickým formátom číslo jedna. Štandardnou príponou je .png.

Pri voľbe rastrového formátu teda musíme zvážiť predovšetkým, akú farebnú hĺbku potrebujeme, vhodnosť stratovej, či bezstratovej kompresie, ako aj požadovanú výslednú veľkosť súboru. Veľkosť súboru je v súvislosti s webom mimoriadne dôležitá, keďže na webovej stránke typicky býva viacero obrázkov, a ak by boli všetky príliš veľké, stránka by sa sťahovala veľmi dlho.

Výber formátu grafiky je len jedným z problémov, ktoré je potrebné pri publikovaní vyriešiť. Pri zverejnení informácie v grafickej podobe by mal byť jej súčasťou aj jej textový ekvivalent. Textovú alternatívu obrázku uvádzame pomocou povinného atribútu `alt` elementu `img`. Tento text sa zobrazí, napr. ak obrázok nie je k dispozícii. Ďalej, atribút `title` môžeme použiť pre vytvorenie titulku, ktorý sa zobrazí ak nad element posunieme kurzor myši. Ak je obrázok veľmi zložitý (napr. grafy, schémy a pod.) a jeho popis by bol príliš dlhý, využijeme atribút `longdesc`. Jeho hodnotou je URI adresa textového súboru s opisom obrázku. Nie všetky prehliadače však tento atribút podporujú. Pre, ak sme použili `longdesc`, umiestnime v blízkosti obrázka aj tzv. *description link* (d-link). Je to samostatný odkaz s textom „[D]“, smerujúci na ten istý súbor ako `longdesc`.

## Vektorová grafika

V úvode kapitoly sme sa dozvedeli, že vektorová grafika má mnohé výhody v porovnaní s rastrovou, teda, za predpokladu, že pracujeme s dátami, na ktoré sa vektorová reprezentácia hodí. Vektorová grafika môže v mnohých prípadoch priniesť úsporu veľkosti prenášaných súborov, ako aj kvalitu výsledného zobrazenia, hlavne ak pri vykresľovaní stránok dochádza k ich škálovaniu, čo sa na webe často hodí. Situácia s vektorovou grafikou na webe je však značne komplikovanejšia, než je to v prípade rastrovej grafiky. Máme k dispozícii dve možnosti: vektorový grafický formát SVG, ktorého podpora dnes však stále nie je dostatočná, a multimediálny formát Flash, ktorý je ale podporovaný iba ako plugin. V tejto časti sa zoznámime s SVG. O formáte Flash si viac povieme v nasledujúcej časti, do ktorej patrí, keďže ide o multimediálny a interaktívny formát.

O formát **SVG (Scalable Vector Graphics)** sa stará W3C. Štandardizovaný bol v roku 2004, a odvtedy sa W3C snaží presadiť jeho používanie na webe a požaduje od výrobcov prehliadačov, aby ho podporovali natívne. Napriek snahe W3C sa to však zatiaľ nepodarilo. SVG zatiaľ vôbec nepodporuje IE, ktorý má najväčší podiel na trhu prehliadačov. Ostatné prehliadače ho už dnes podporujú pomerne dosť dobre.

SVG umožňuje reprezentovať základné geometrické tvary, polygóny a krivky, farebné výplne a farebné prechody (gradienty). Každý objekt má svoj vlastný alfa-kanál. Okrem toho SVG umožňuje vložiť rastrové obrázky a text. SVG je XML formátom, jeho súbory teda môžeme otvoriť a upraviť aj v textovom editore. Okrem toho SVG podporuje aj kompresiu. Koncovka súboru je `.svg` a `.svgz` (skomprimované súbory). V praxi pochopiteľne nepracujeme s SVG v textovom editore, použijeme softvér na prácu s vektorovou grafikou. Exportovať do SVG možno z väčšiny z nich. Existuje aj voľne dostupný nad SVG postavený open-source program Inkscape.

Nasledujúci kód popisuje obrázok elipsy. Jej vzhľad si môžeme pozrieť priamo v okne webového prehliadača. Všimnime si aj elementy `title` a `desc`, ktoré poskytujú textový ekvivalent informácie v grafickej podobe.

```

ellipse.svg
<?xml version="1.0" encoding="UTF-8"?>
<svg xmlns="http://www.w3.org/2000/svg" version="1.0"
  width="600" height="800">
  <title>Elipsa</title>
  <desc>Obrázok elipsy v svg formáte</desc>
  <ellipse transform="translate(200 150) rotate(-30)"
    rx="100" ry="50"
    fill="red" stroke="blue"
    stroke-width="20" />
</svg>

```

Obrázok elipsy vo formáte SVG

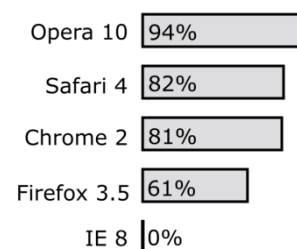
W3C má pre presadzovanie SVG dobré dôvody. Práca s rastrovou grafikou je pre webdizajnérov značne obmedzujúca. Väčšina z nich v skutočnosti aj tak pracuje s vektorovými formátmi, a všetky grafiky musia pre použitie na webe vyexportovať do rastrových formátov. Často sa stáva, že stránka je optimalizovaná pre fixné rozlíšenie (napr. 1024×768), čo v konečnom dôsledku vedie k zníženiu prístupnosti (pozri

**Zamyslime sa:** aké faktory majú vplyv na veľkosť súboru pri jednotlivých formátoch?

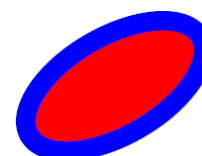


## INKSCAPE

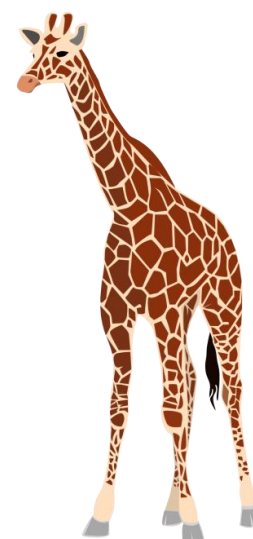
Na prácu s vektorovou grafikou môžeme použiť open-source vektorový editor Inkscape.



**Podpora SVG v prehliadačoch.** Zdroj Jeff Schiller: <http://www.codedread.com/svg-support.php>



Obrázok elipsy z príkladu.



Príklad zložitejšieho obrázku vo formáte SVG.

Plugin Adobe SVG Viewer:  
<http://www.adobe.com/svg/viewer/install/>

HTML 5 bude pravdepodobne obsahovať za účelom vkladania videí špeciálny element video.

Pre spracovanie videa môžeme použiť open-source softvér **Virtualdub**:  
<http://www.virtualdub.org>

Skúšobnú verziu programu **Adobe Flash CS4 Professional** môžeme získať z webovej stránky :  
<http://www.adobe.com/>

Jednoduchý on-line generátor Flash-ových efektov nájdete na adrese  
<http://www.effectgenerator.com/>.

Flashovú animáciu pohybujúceho sa auta nájdeme na URL adrese:  
<http://virl.com/flash/>.

Kapitolu 7). Ľubovoľný natívne podporovaný vektorový formátby tak mohol vyvolať ďalšiu revolúciu vo webdizajne. Formát SVG má zatiaľ najbližšie k tomuto cieľu.

## Animácie, video a interaktívne objekty

Na webových stránkach sa bežne stretávame aj s animáciou a s videom. **Jednoduché animácie** môžeme reprezentovať napr. formátom GIF. S týmto sa na webe stretávame pomerne často, keďže je v súčasnosti stále najlepšie podporovaná možnosť. Okrem toho animáciu podporuje aj formát SVG.

Najjednoduchšia možnosť ako vložiť do webovej stránky video je pomocou (X)HTML elementu object. Jeho prehratie však vyžaduje plugin pre použitý video formát. (napr. avi, alebo mpeg). Pluginy pre tieto formáty však nie sú veľmi rozšírené. V súčasnosti sa preto takmer výlučne za týmto účelom používa formát Flash, ktorý okrem iných možností umožňuje aj prehrávanie videa.

Podpora pluginov v prehliadačoch otvorila cestu rozšíriť ich v podstate o akúkoľvek funkcionálnu. Jedným z prvých **interaktívnych** „webových“ formátov sa stali javové applety. Tieto boli pomerne populárne koncom deväťdesiatych rokov minulého storočia. Dnes sa s nimi na webe stretáme už len výnimočne, túto technológiu takmer úplne vytlačil formát **Flash**, umožňujúci tvorbu interaktívnych multimedialných prezentácií (či dokonca programov), ktoré kombinujú rastrovú a vektorovú grafiku, animáciu, video a audio, a ktorý dnes predstavuje *de facto štandard* v tejto oblasti. Typické flashové aplikácie siahajú od jednoduchých animácií a prezentácií, cez interaktívne grafy, interaktívne ovládacie prvky (aj keď tie sú diskutabilné z hľadiska prístupnosti, pozri Kapitolu 7), po vkladanie videa do webových stránok, či flashové hry bežiacie v okne prehliadača.

Výhodou Flashu pred inými podobnými formátmi (pozri stĺpec) je dobrá podpora viacerých operačných systémov, ale predovšetkým jeho popularita a s ňou spojená vysoká penetrácia Flash pluginu. Kým plugin je zadarmo, pre vytváranie prezentácií v tomto formáte potrebujeme vývojové prostredie od spoločnosti Adobe, to je ale už komerčný produkt. Voľne dostupné softvéry sú zväčša len veľmi primitívne.

Problémom Flashu, a iných podobných formátov, je **znižovaná prístupnosť** interaktívnych prvkov, s ktorou sa často stretávame. Bez vizuálneho vnemu a bez myšky je väčšina týchto prvkov len veľmi ťažko ovládateľná. Vývoj v tejto oblasti síce napreduje, a prostredie od Adobe obsahuje aj nástroje pre zvýšenie prístupnosti, hlavným problémom je však to, že tieto formáty umožňujú tvorbu ľubovoľných, aj veľmi komplikovaných ovládacích rozhraní - ak s nimi pracuje človek s nižším povedomím prístupnosti, prístupnosť výsledku je vážne ohrozená.

Pozrieme teraz ako vytvoriť (X)HTML kód pre správne vloženie flashového objektu na webovú stránku. Za týmto účelom použijeme element object. Budeme vkladat animáciu nachádzajúcu sa v súbore auto.swf.

```
animation.html
```

```
<object type="application/x-shockwave-flash"
      data="auto.swf" width="320" height="280"
      title="Jazda terénnym autom do prudkého kopca">
  <param name="movie" value="auto.swf" />
  
</object>
```

### *Vloženie flashového prvku pomocou elementu object*

Keďže element object slúži na vkladanie prvkov interpretovaných rôznymi pluginmi, umožňuje nám vložiť aj alternatívny obsah, ktorý sa zobrazí v prípade, že plugin nie je dostupný. Tento obsah vložíme jednoducho do vnútra elementu object. My sme si ako alternatívu pripravili GIF animáciu. Keďže ide o grafický prvok, súčasťou kódu musí byť ešte aj textový ekvivalent. Ten sme vložili štandardne pomocou atribútu alt. Keďže nám na prístupnosti veľmi záleží, vložili sme tento text ešte aj pomocou atribútu title do elementu object.

V nasledujúcej tabuľke uvádzame prehľad technológií a formátov pre zobrazenie grafiky na webe aj s ich využitím.

	Raster		Vektor	
	nákres, skica	reálna	nákres, skica	reálna
statická	GIF, PNG	JPG	SVG, Flash	-
animovaná	GIF	avi, mpeg, Flash, SilverLight	SVG, Flash, SilverLight	-
interaktívna	Java Applet, Flash, SilverLight			-

## Zvuková informácia

(X)HTML v súčasnosti priamo nepodporuje vkladanie zvuku vo webových stránok. Napriek tomu, pri publikovaní informácie v zvukovej podobe máme na výber z niekoľkých možností. Záznam zvuku v niektorom zo zvolených formátov nahráme na server a na webovej stránke vytvoríme odkaz na príslušný dokument. Ďalšou možnosťou je využiť niektorý multimedialny plugin (najčastejšie Flash) a zvuk publikovať pomocou nich.

Najlepšie podporované **audio formáty** sú MP3 (MPEG-1 Layer III) alebo OGG. Obidva formáty pracujú so stratovou kompresiou. Oproti pôvodnému záznamu (napr. wave) majú menšiu dátovú kapacitu, ale aj ich kvalita je o niečo nižšia. To ako veľmi majú byť dáta skomprimované na úkor veľkosti súboru môžeme pri konvertovaní nastaviť. Pri publikovaní napr. záznamu prednášky na webe použijeme silnejšiu kompresiu, naopak, pri archivovaní hudby pre reprodukciu na kvalitnej audio sústave slabšiu.

**MP3** je známy a všeobecne podporovaný formát. Jeho nevýhodou je množstvo patentov s ním spojených. S jeho prehrávaním si však poradí väčšina počítačov ako aj a prenosných prehrávačov. **OGG** je patentovo neviazaný otvorený formát pre kódovanie multimedialneho obsahu. Navyše, pri rovnakej veľkosti výsledného súboru dosahuje vyššiu kvalitu ako záznam vo formáte MP3. Jeho podpora je však o niečo slabšia, a to hlavne u prenosných prehrávačov, počítače si s ním poradia lepšie.

Zvukovou analógiou vektorových grafických formátov je **syntetizovaný zvuk**. Formát **MIDI** (Musical Instrument Digital Interface) popisuje činnosť hudobných nástrojov. Na jeho prehratie je potrebný syntetizátor, ktorý „vygeneruje“ každý tón. Ako syntetizátor môžeme použiť aj počítačový program. Takto generovaný zvuk je však „umelý“ a aj keď je vytvorený s počítačovou presnosťou, nemôže konkurovať dobrému živému hudobníkovi. Formát MIDI sa v digitálnej dobe používa hlavne na komunikáciu medzi hudobnými nástrojmi a zariadeniami na spracovanie zvuku. Napr. k počítaču pripojené klávesy posielajú do počítača signály o tom kedy, ako dlho a ako silno bol ktorý kláves stlačený. Týmto spôsobom sa samozrejme nedá vygenerovať ľudský hlas.

## Webové nástroje pre správu multimédií

V predchádzajúcich častiach sme si ukázali, že publikovanie multimedialneho obsahu (obrázky, zvuky, videá) nie je jednoduchou záležitosťou. Na prácu s takýmto obsahom často potrebujeme špeciálny softvér. Našťastie existuje celá rada voľne širiteľných softvérov na prácu s multimédiami, ako aj mnohé on-line nástroje. Výhoda on-line nástrojov je, že ich nemusíme inštalovať, a môžeme ich použiť na každom počítači. Práve s vybranými on-line nástrojmi sa zoznámime v tejto časti.

Medzi najznámejšie on-line systémy pre **správu obrázkov** (a videí) patrí **Picasa**. Po bezplatnej registrácii nám systém umožní nahrávať a v galérii verejne publikovať grafické súbory. Súbory môžeme triediť do kategórií (albumov), pridávať k nim popisy, nastaviť pravidlá pre ich zdieľanie. Picasa podporuje aj niektoré prvky sociálnych sietí: albumy môžeme vzájomne prepájať a zdieľať, používatelia si môžu

Flash nie je jediným nadstavbovým formátom, ktorý môžeme pomocou pluginu vložiť do webových stránok. Za účelom vkladania interaktívnych multimedialnych objektov môžeme použiť aj konkurenčnú technológiu Microsoft **Silverlight**, ďalej môžeme takéto objekty naprogramovať v jazyku **Java**, či dokonca v nám známom prostredí **Imagine Logo**. Pre vkladanie **3D objektov** existujú rozšírenia **VRML** a **X3D**.

*Rozšírenie príslušných pluginov je však obmedzené, preto by sme mali uprednostniť formát Flash, ktorý je oveľa rozšírenejší.*

Zaujímavou možnosťou pre vloženie zvuku do webovej stránky je Yahoo! media player <http://developer.yahoo.com/mediaplayer/>

HTML 5 bude pre vkladanie zvuku obsahovať element audio.

**Audacity** je voľne dostupný open-source softvér na prácu so zvukom: <http://audacity.sourceforge.net/>

Na webovej stránke <http://www.freemidi.org/> nájdeme množstvo voľne stiahnuteľných skladieb vo formáte MIDI.



Súbory do galérie **Picasa** môžeme prenášať aj pomocou lokálne inštalovanej rovnomennej aplikácie. Picasa poslúži aj ako jednoduchý editor obrázkov, nástroj pre vytváranie plagátov, koláží či videí zo súborov v galérii: <http://picasa.google.sk/>



Viac o jeho možnostiach sa dozvieme na adrese:  
<http://www.flickr.com/tour/>

Kód, ktorý generuje server YouTube pre vloženie videa na vlastnú stránku nie je správny. „Opraviť“ nám ho pomôže nástroj na adrese:  
[http://www.tools4noobs.com/online\\_tools/youtube\\_xhtml](http://www.tools4noobs.com/online_tools/youtube_xhtml)



Okrem konverzie medzi formátmi súborov, dokáže Zamzar skontrolovať video z video galérií (YouTube, Google Video, Myspace, ...) do vybraného formátu:  
<http://www.zamzar.com/>

Bohatý zoznam on-line televízií nájdeme na:  
<http://www.online-television.tv/>

Podcasty niektorých slovenských rádii:  
**Rádio FM:**  
<http://www.radiofm.sk/podcast/list/>  
**Rádio Expres:**  
<http://www.expres.sk/vysielanie/audiopodcasty/1>



Juice Receiver je open-source program na preberanie audio podcastov z webu:  
<http://juicereceiver.sourceforge.net/>

fotografie vzájomne komentovať. Zaujímavosťou je funkcia, pre rozpoznávanie tváří na fotografiách a automaticky generovaný obsah RSS kanálu.

Ďalšou nemenej známou aplikáciou pre zdieľanie fotografií je Flickr. Flickr bol jedným z prvých nástrojov webu 2.0, ktorý umožňoval pridávať k fotografiám komentáre a tagy. Používatelia môžu fotografie zoskupovať podľa obsahu, čo neskôr uľahčí ich prehľadávanie. Pri nahrávaní fotografií si môžeme zvoliť, pod akou licenciou obrázkov chceme zverejniť. Keďže licencia väčšiny obrázkov umožňuje ich ďalšie šírenie, môžeme Flickr chápať ako obrovskú fotobanku.

Pre publikovanie a zdieľanie video súborov sa „kráľom“ stal dnes zrejme najväčší internetový server YouTube. YouTube poskytuje príležitosť ako sa zviditeľniť. Stojí za zrodom mnohých internetových hviezd, ktorých video si pozrelo niekoľko miliónov používateľov. Jedným z problémov ktorým YouTube čelí je jeho obsah, ktorý často porušuje autorské práva. Sloboda prejavu, ktorú umožňuje je dôvodom, že v niektorých krajinách je jeho obsah cenzurovaný alebo blokovaný úplne.

Nahrané video súbory sú skonvertované do formátu Flash. Ich zobrazenie je tak vďaka rozšírenosti Flashu bezproblémové na väčšine počítačov. Videá je možné vyhľadávať a komentovať. Zaujímavosťou je, že reagovať na video môžeme vlastnou video odpoveďou. Pri zobrazení konkrétneho videa server automaticky vygeneruje (X)HTML kód pomocou ktorého môžeme video vložiť aj do ďalších webových stránok, pričom video je stále uložené na serveri YouTube. Táto funkcia nám môže značne uľahčiť publikáciu videa na internete.

Pri práci s multimédiami sa nevyhneme vzájomnej konverzii medzi formátmi. Dôvodom môže byť nevhodný zdrojový formát alebo CMS systém, ktorý daný formát nepodporuje. Webová aplikácia Zamzar nám pomôže vzájomne konvertovať niektoré z najpoužívanejších formátov.

## Audio a video z internetu

Špeciálnym formátom multimédií (najmä audiovizuálnych) je streaming (tiež kontinuálny/prúdový prenos). Pri takomto prenose sa dáta zobrazujú postupne ako sa cez sieť prenášajú. Nie je tak potrebné čakať na prenesenie celého súboru. Pokiaľ sú streamy dostupné cez web, hovoríme aj o webcastingu. Môže sa uskutočniť v reálnom čase, vtedy stream beží stále a my sa môžeme kedykoľvek pripojiť/odpojiť (internetová televízia alebo rádio), alebo systémom na vyžiadanie (on demand), kedy sa stream spustí až po kliknutí na príslušný link. Streamový príjem videa on demand ponúka napr. YouTube. Pre prúdový príjem viacerými používateľmi súčasne, musí mať prevádzkovateľ tejto služby okrem obsahu aj streamovací server, ktorý zaistí komunikáciu s používateľmi a plynulé vysielanie dát.

## Podcasting

Zaujímavým spôsobom zverejnenia informácie je tzv. podcast. Podcast je audio alebo video súbor, ktorý jeho autor umiestnil na internete. Zoznam podcast súborov býva často distribuovaný vo forme RSS. Obsahom týchto súborov môže byť záznam z vysielania rádia, obsah knihy nahovorený jej autorom, záznam z prednášky, kurzy cudzieho jazyka, a pod. Špeciálny program **podcatcher** zabezpečí pravidelné sťahovanie multimediálnych súborov z vybraného RSS kanálu. Oblúbené rozhlasové alebo televízne relácie si tak môžeme vypočuť aj napriek tomu, že sme to nestihli v čase ich vysielania.

## Čo sme sa naučili

Oboznámili sme sa základnými možnosťami publikácie multimediálneho obsahu na internete. Poznáme výhody a nevýhody jednotlivých formátov a podľa situácie sa vieme rozhodnúť, kedy ktorý z nich použiť. Vieme, že z dôvodu prístupnosti, by mal mať každý netextový objekt aj svoj textový ekvivalent. Spoznali sme aj formáty pre publikovanie audiovizuálneho a multimediálneho obsahu. Oboznámili sme sa s on-line službami pre publikovanie a správu multimédií sa s podcastingom.



## Kapitola 6: Publikovanie na webe

Web je špecifické médium. Ak chceme vytvoriť dobrú webovú stránku, musíme jeho špecifikám porozumieť. Publikovanie na webe je odlišné od publikovania v tlači, a odvíja sa od špecifických vlastností webu a predovšetkým od zvyklostí a očakávaní používateľov:

- používatelia trávajú 99% času na iných webových stránkach, preto je potrebné našu stránku prispôsobiť tomu, čo je zaužívané;
- používatelia majú na výber množstvo iných webových stránok s podobným obsahom, ak nie sú s našou stránkou spokojní, informácie nájdu inde;
- vďaka vyhľadávacím službám môže čitateľ „pristáť“ na ľubovoľnej podstránke nášho webu. Každá podstránka musí preto celé naše webové sídlo dostatočne reprezentovať, a mala by byť prepojená s hlavnou stránkou;
- práca s webom je rýchla a nárazová;
- web je charakterizovaný absenciou kontextu: kým pri čítaní knihy máme dost dobrú predstavu, čo asi môžeme očakávať, keď otočíme stranu, na webe často nevieme, čo nás čaká, keď klikneme na niektorý hypertextový odkaz.

V tejto kapitole sa dozvieme, aké sú základné druhy webových stránok, a čo je pre ne typické. Povieme si základné zásady týkajúce sa vzhľadu a usporiadania webových stránok, a naučíme sa písať a upraviť text pre publikovanie na webe.

### Žánre a typy webových stránok

Webové stránky sú rôznych druhov. Podľa určenia rozlišujeme prezentačné stránky, náučné stránky, vyhľadávače a katalógy, webové obchody, webové médiá, blogy, sociálne siete, a iné. Zaužívané štandardy a očakávania používateľov sú pre každý typ iné. **Prezentačné stránky** tvoria veľkú skupinu webových stránok. Informujú o osobách, organizáciách, atď. Podľa toho môžu mať rôzny obsah a rôznu podobu:

**osobná stránka:** je jedným z najvoľnejších žánrov na webe. Typicky na nej nájdeme informácie vzťahujúce sa k osobe jej autora (majiteľa). Aké informácie kto chce zverejniť môže však byť veľmi rôzne. Osobné stránky boli populárne hlavne v deväťdesiatych rokoch minulého storočia, dnes ich vytláčajú blogy;

**stránka inštitúcie:** slúži pre zamestnancov, či členov, a pre styk s verejnosťou (napr. stránka školy). Informuje o činnosti inštitúcie, a o jej službách, či produktoch. Môže mať verejnú a neverejnú časť (za heslom). Veľké organizácie môžu mať objemné a zložité webové stránky (napr. veľká univerzita). Takéto stránky často majú viac navigačných menu, usporiadaných pre rôzne skupiny používateľov (napr. iné pre zamestnancov, pre študentov, a pre verejnosť);

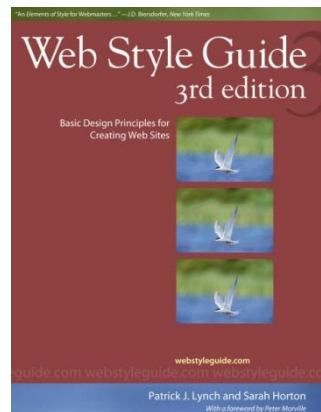
**firemná stránka:** v popredí je prezentácia produktov a služieb. Je dôležitejšie, čo sa predáva, ako to, kto to predáva. Seriózna obchodná spoločnosť zverejní aj kontaktné a ďalšie informácie, prípadne stručnú históriu firmy, a pod., tieto informácie sa však spravidla neobjavujú podrobne hneď na titulnej stránke;

**stránka produktu, kampane:** samostatná stránka pre konkrétny produkt (nový produkt, podujatie, ...), informačnú kampaň (za čisté mesto, ...) a pod. V takomto prípade sú typicky všetky ostatné informácie zminimalizované, a miesto nich je uvedený len odkaz na stránku výrobcu, či realizátora kampane.

Pozrime sa podrobnejšie aj na niektoré iné žánre webových stránok:

**vyhľadávače a katalógy:** zvyčajne žiadny obsah priamo neprodujú, umožňujú nám nájsť obsah na iných webových stránkach. Kým vyhľadávač typicky používa index webových stránok získaný automatizovaným prehľadávaním webu, katalóg ponúka zoznam stránok, ktoré doň zaradili platení editori alebo bežní používatelia. Pridanie do katalógu môže byť zadarmo alebo platené.

**webové médiá:** často fungujú ako webová verzia tlačeného denníka alebo časopisu, ale sú aj čisto internetové médiá. Opäť, mali by umožňovať rýchlo nájsť to, čo nás zaujíma, v záplave iných správ. Správy sú často prinášané „hlavne rýchlo“, toleruje sa menšie množstvo drobných chýb a preklepov, v tlačených médiách nepredstaviteľné. Správy môžu byť po vydaní viackrát aktualizované. Čitatelia majú často možnosť na správu zareagovať. Internetové spravodajstvo je veľmi populárne, lebo je rýchle a je zadarmo. Dôsledkom je, že tieto



*Web Style Guide*, výborná publikácia o webovej tvorbe a publikovaní, dostupná on-line na:

<http://webstyleguide.com/>



Autormi WSG sú Patrick J. Lynch a Sarah Horton.

**Osobná stránka učiteľa** môže byť príkladom zmysluplnej a zaujímavej profesionálnej osobnej stránky. Na takejto stránke môže učiteľ zverejniť napr. informácie o predmetoch, ktoré učí, študijné materiály, či ďalšie informácie pre študentov. Stránka môže byť skombinovaná s blogom.

Príkladom môže byť **webová stránka kampane**, ktorú škola vytvorí za účelom oslovenia nových študentov.

Platených katalógov je dnes len veľmi málo. Skôr sa stretne s tým, že služba ponúka za poplatok zobrazenie na zvýhodnenom mieste.

Slovo *portál* má v tomto kontexte znamenať „bránu“ do sveta internetu.

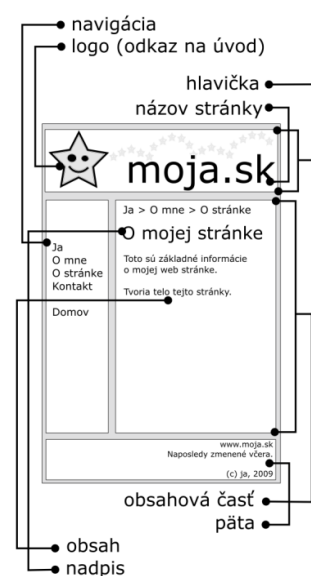
Dnes **klasické portály** ustupujú, trend ide smerom k jednoduchším stránkam. Na slovenskom internete sa s nimi ale stále môžeme pomerne často stretnúť.

Sociálne siete možno využiť na udržiavanie kontaktu so starými známymi, ako aj na hľadanie nových. Existujú však aj služby orientované na udržiavanie **profesionálnych kontaktov** a vyhľadávanie odborníkov v rôznych oblastiach. Takouto stránkou je napr.:

<http://www.linkedin.com/>

**Breadcrumbs bar** (z angl. *breadcrumbs* = omrvinky) je štruktúrna navigačná lišta s výpisom cesty od hlavnej stránky po aktuálnu podstránku, napr.:

[home](#) > [produkty](#) > čaj



Anatómia webovej stránky.

stránky často udávajú krok vo webovom dizajne, keďže používatelia sú na ne zvyknutí;

**internetové portály:** spravodajské weby, katalógy, či vyhľadávače sa niekedy štylizujú do formy portálu: stránky prinášajúcej veľké množstvo informácií, rôznych nástrojov a služieb, ale predovšetkým odkazov na zaujímavý obsah;

**náučné stránky:** tvoria širokú a rôznorodú skupinu. Vlastne sem môžeme zaradiť čokoľvek od technickej dokumentácie, cez on-line jazykové kurzy, po Wikipédiu, ale trebárs aj stránky s receptami. Používatelia od nich očakávajú, že rýchlo nájdu to, čo potrebujú. Zriedkakedy ich čítajú celé. Na stránku sa často dostanú cez vyhľadávač. Preto je dôležité prelinkovanie na hlavnú stránku, ale aj medzi témami, členenie textu a optické záchytné body (viď nižšie);

**blogy:** blog je webová stránka, ktorá umožňuje jej autorovi jednoducho publikovať kratšie príspevky, ktoré bývajú chronologicky usporiadané, najnovší navrchu. Blog by mal byť tematicky vyhranený, publikovanie by malo byť pravidelné, a mal by reprezentovať názory a osobnosť autora. Takmer vždy je možné na články zareagovať. Blogy sú veľmi populárne a autori na nich publikujú na veľmi rôzne témy, od úplne súkromných po vysoko odborné. Možno ich využiť pri vzdelávaní, ale napr. aj na marketingové účely;

**sociálne siete:** stránky ktoré umožňujú spojiť sa s inými používateľmi, kontaktovať ich a komunikovať. Typickou súčasťou je osobný profil, ktorý si vyplní každý používateľ, a funkcia, ktorá nám umožní zostaviť, udržiavať a sledovať zoznam našich známych. Tieto stránky často integrujú blogy, diskusné fóra, ale aj rôznu inú funkcionálnu, vrátane rôznych nástrojov a on-line hier. Naopak, rôzne črty sociálnych sietí často obsahujú aj rôzne iné stránky.

## Štruktúra a vzhľad webových stránok

Webové sídlo by malo mať **jednotný vzhľad**. Každá podstránka nášho webu potrebuje mať jednotnú **hlavičku a päť**. Tieto by svojím grafickým prevedením mali vystihovať tému našej stránky (napr. logo organizácie/webu v hlavičke). Pri prezeraní jednotlivých podstránok tak používateľ stále vie, že sa nachádza na tom istom webe. Tiež je výborné v hlavičke naznačiť polohu podstránky na vašom webe. Za týmto účelom môžeme použiť napr. tzv. *breadcrumbs bar*. Každá podstránka by mala obsahovať opisný nadpis, odkaz na hlavnú stránku, dátum poslednej zmeny a informácie o autorských právach, či podmienkach používania.

K jednotlivým podstránkam sa dostaneme prostredníctvom navigačného menu. To by sa malo tiež nachádzať na každej podstránke (aspoň najvrchnejšia úroveň), aby sme sa nemuseli zakaždým vracieť na hlavnú stránku. Navigačné menu môže mať rôznu štruktúru, hovoríme o **navigačnej štruktúre webovej stránky**. Tá býva buď **lineárna** (jednoduchšie stránky), alebo častejšie **stromovitá** (o čosi zložitejšie stránky). U veľmi zložitých stránok (napr. Wikipédia) stromovitá štruktúra nie je možná, lebo usporiadať všetky stránky zmysluplne do stromu jednoducho nie je možné.

Pri väčšine webových stránok si vystačíme so **stromovou štruktúrou**. Je najprehľadnejšia a najpoužívanejšia. Strom by mal zodpovedať tematickému členeniu nášho webu a mal by sa zhora-dolu otvárať *od všeobecného k špecifickému*. Nemal by byť príliš hlboký, ani by na žiadnej úrovni nemalo byť príliš veľa možností. Snažme sa, aby každá podstránka bola „vzdialená“ od hlavnej stránky najviac tri kliknutia. Toto je možné aj u pomerne zložitých webových stránok: do stromu s hĺbkou tri a s desiatimi podstránkami na každej úrovni vieme umiestniť  $10^3 = 1000$  stránok.

Veľkú pozornosť by sme mali venovať aj výberu farebnej schémy a **grafickému dizajnu** jednotlivých častí stránky. Tieto časti môžu do veľkej miery vtisnúť našej webovej stránke originálny „punc“. Používatelia si tak našu stránku ľahšie zapamätajú a skôr sa v budúcnosti vrátia. Grafický dizajn by však nemal byť prehnaný a príliš samoučelný. Mal by slúžiť predovšetkým na optické oddelenie jednotlivých obsahových prvkov stránky a jej prehľadnosť.

Pri voľbe **farebnej schémy** musíme zohľadniť predovšetkým existujúcu farebnú signatúru organizácie, či produktu, ak už existuje. Ďalej tému a účel stránky (napr. lesníctvo: zelená, oblečenie pre bábätká: ružová, a pod). Vyhnime sa farbám, ktoré sú zaužívané ako výstražné (červená, čierna so žltou). Stránka by mala byť vyfarbená účelne. Pozadia a podklady by mali byť v pastelových farbách, naopak satureované

farby použijeme tam, kde chceme pritiahnúť pozornosť čitateľa - hlavička a päta, navigačné prvky, a pod. Mali by sme dbať aj na to, aby kontrast medzi farbou textu a pozadia bol vždy dostatočný, a aby sme nepoužili farebné kombinácie, ktoré môžu splyvať používateľom s obmedzeným vnímaním farieb (pozri aj Kapitulu 7).

## Tvorba a úprava textu

Webové stránky málokedy čítame lineárne odhora až dolu. Častejšie hľadáme nejakú konkrétnu informáciu, otvárame pritom rôzne stránky, ktoré najprv rýchlo „preskenujeme“ očami, a hľadáme v texte *záchytné body*. Ak nás nejaká časť zaujme, začítame sa. Aby sme takéto čítanie textu v čo najväčšej miere podporili je potrebné text členiť a štruktúrovať, väčšie bloky textu rozčleniť do odstavcov, používať často podnadvisy a zoznamy, zvýrazňovať.

**Texty nadpisov a odkazov** sú mimoriadne dôležitým a hodnotným obsahom. Týmto častiam venujú zvýšenú pozornosť aj vyhľadávače. Mali by byť stručné a opisno-výstižné. Nadpis by mal napovedať o obsahu stránky. Prílišná snaha o zábavnosť, či senzáciu môže pôsobiť odpudzujúco. Pri tvorbe odkazov by sme sa mali do maximálnej možnej miery pokúsiť vykompenzovať absenciu kontextu na webe, pre upresnenie môžeme využiť aj atribút `title` elementu a.

Práca s webom je rýchla a nárazová. Informácie treba podávať stručne, jasne a výstižne, pokiaľ možno ucelenou formou. Na web sa hodí tzv. **štýl obrátenej pyramídy**:

- sumár textu sa presúva na začiatok, oproti klasickému štýlu, kde sa nachádza zväčša na konci. V prvom odstavci sumárne a stručne podáme najdôležitejšie poznatky ako aj závery, ktoré text komunikuje;
- až v ďalšom texte postupne jednotlivé témy rozvedieme do hlbších detailov;
- postup aplikujeme rekurzívne. Aj pre každú podtému začneme sumárom a postupne rozmieňame na drobné. Rovnako pre ďalšie podtémy;
- podobne aj v každom odstavci začneme vetou, ktorá ho zhrňa. Ak je to možné, pokúsime sa vety začínať vždy najdôležitejším nosným slovom, okolo ktorého sa veta točí, predovšetkým pokiaľ ide o prvú vetu odstavca, a o nadpisy.

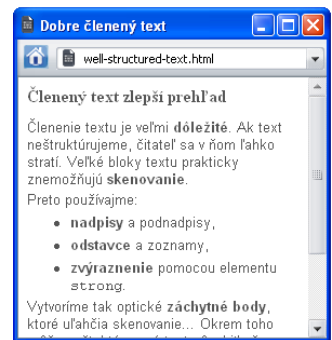
Výsledkom je text, ktorý je možné čítať odhora smerom dolu až po moment, keď sa už oboznámime s potrebnou mierou detailov. V texte sa dá dobre orientovať: *strategicky umiestnené kľúčové slová uľahčujú nelineárne čítanie textu*. Typický webový používateľ bude text čítať presne takto. S vysokou pravdepodobnosťou v istom momente klikne na odkaz, ktorý ho zaujme a prejde na inú stránku. Práve štýl obrátenej pyramídy zaručí, že v tomto momente bude už oboznámený so všetkým podstatným, a to, k čomu sa už nedostal, sú len ďalšie detaily.

**Webová typografia** je špecifická, a kvalitná typografia je na webe nutnou ingredienciou. Základnú typografiu textu je nutné dodržať, inak výsledok pôsobí neseříozne. Najdôležitejšie typografické pravidlá:

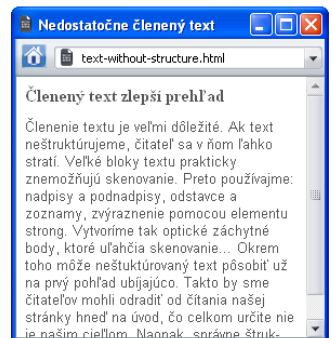
- väčšinou sa preferuje **bezpätkový font**, umožňujúci rýchle čítanie;
- delenie slov nefunguje, preto zarovnáme text iba na ľavý okraj, aby v ňom nevznikli priveľké, rušivé medzery;
- **odkazy** musia byť odlišené zaužívaným spôsobom;
- rozlišujeme medzi pomlčkou (elipsa) a rozdeľovníkom (divis).
- píšeme správne slovenské úvodzovky („“), pomlčky, a pod., aj keď nemáme tieto znaky na klávesnici.

## Čo sme sa naučili

Spoznali sme najdôležitejšie druhy webových stránok. Oboznámili sme sa so zvyklosťami používateľov a s tým, čo asi môžu od našej stránky očakávať. Je to dôležité, lebo používatelia webu majú na výber. Keď nie sú s niektorou stránkou spokojní, nájdu hľadané informácie inde. Pre celé webové sídlo by sme mali vytvoriť jednotný vizuálny štýl. Na každú podstránku by sme mali dať nadpis, linku na hlavnú stránku, dátum poslednej zmeny a informácie o autorstve. S grafickým prevedením a farbami musíme pracovať citlivo. Rovnako dôležité je aj štruktúrovať text a použiť pisateľský štýl, vhodný na web.



Dobre členený text.



Nedostatočne členený text.

AaBbCc

**Pätkový font** (angl. *serif*) používame hlavne na dlhšie texty, o ktorých predpokladáme, že budú čítané lineárne. Pätkový font pomáha odbúrať únavu pri dlhšom čítaní. Je charakterizovaný tzv. pätkami (na obrázku vyznačené červenou).

AaBbCc

**Bezpätkový font** (angl. *sans-serif*) využijeme pri textoch pre rýchle a nelineárne čítanie. Číta sa rýchlejšie a uľahčuje orientáciu v texte. Pri dlhšom čítaní viac unavuje.



Jakob Nielsen je uznávaným expertom na použiteľnosť. Preslávil sa ako veľký popularizátor a „guru“ v tejto oblasti. Je autorom viacerých výborných kníh o použiteľnosti na webe. Na svojom webe publikuje voľne dostupný a veľmi poučný internetový magazín o použiteľnosti s názvom **Alertbox**:  
<http://www.useit.com/alertbox/>

Alternatívou ku zdĺhavému štúdiu je zveriť kontrolu našej webovej stránky odborníkom na použiteľnosť. Ich služby však môžu byť pomerne nákladné...

## Kapitola 7: Použiteľnosť a prístupnosť na webe

Použiteľnosť a prístupnosť informačných zdrojov a používateľských rozhraní sú dve zásadné miery kvality vo webdizajne a pri publikovaní nielen na webe. Zaoberajú sa teda problémom, ako zistiť, či je daná webová stránka alebo aplikácia dobrá a kvalitná. Kontrola kvality je bezpochyby nutnosťou pri akejkoľvek produkčnej činnosti, publikovanie na webe nevynímajúc.

Použiteľnosť sleduje *lahkosť použitia a spokojnosť* používateľa s produktom. Prístupnosť zakladá na jednoduchom a *efektívnom spracovaní informácií aj používateľom so zdravotným, či iným obmedzením*. Osvojenie si oboch mierok webovými tvorcami prináša úžitok všetkým používateľom webových stránok a aplikácií, ktoré vytvoríme, a môže v podstatnej miere prispieť k celkovému úspechu webových projektov. Napriek tomu je táto oblasť podceňovaná a predovšetkým v slovenskom kontexte je celkové povedomie o nej nedostatočné a to dokonca aj u profesionálnych webových špecialistov.

V tejto kapitole sa s použiteľnosťou a s prístupnosťou zoznámime, a osvojíme si ich princípy. Následne sa oboznámime aj s praktickými dôsledkami, ktoré pre webovú tvorbu vyplývajú: ako pri práci s webovými technológiami postupovať, a na čo si dať pozor, aby výsledkom boli použiteľné a prístupné webové stránky.

### Použiteľnosť (usability)

**Použiteľnosť** je miera, ktorá vyhodnocuje, či je používanie daného systému, produktu alebo akejkoľvek veci (v ďalšom *nástroj*) ľahké (vysoká použiteľnosť) alebo naopak ťažké (nízka použiteľnosť). Použiteľnosť je dôležitá v celej oblasti priemyselnej dizajnu, nielen v informatike, či v publikácii. Jakob Nielsen zdefinoval päť faktorov, z ktorých sa použiteľnosť skladá:

**naučiteľnosť** (lernability): ako ľahko sa používateľ s nástrojom zoznámiť a použiť ho v prípade, že sa s ním pracuje po prvýkrát?

**efektivita** (efficiency): keď už sa používateľ naučil nástroj používať, ako rýchlo a efektívne s ním dokáže pracovať?

**zapamätateľnosť** (memorability): ak používateľ nástroj istý čas nepoužíval, ako dlho trvá, kým pri opätovnom používaní dosiahne predchádzajúcu úroveň efektivity?

**chybovosť** (errors): ako často robí pri používaní nástroja používateľ chyby? Akých závažných chýb sa dopúšťa? Ako ľahko sa dokáže s chybou vysporiadať?

**spokojnosť** (satisfaction): ako veľmi je používateľ s nástrojom spokojný?

Nedostatočná použiteľnosť je charakterizovaná porušením niektorého z jej piatich faktorov. Pozrime sa najprv bližšie na prvé tri z nich. Ak je používateľské rozhranie navrhnuté nezvyčajným spôsobom a je ťažké mu intuitívne porozumieť, ide o defekt použiteľnosti. Ak je napriek intuitívnosti ovládacích prvkov používanie nástroja zorganizované do náročných a zdĺhavých akcií, ide o defekt použiteľnosti. Ak je spôsob používania náročný na zapamätanie a po istej dobe nepoužívania sa nástroj musíme učiť používať nanovo, ide o defekt použiteľnosti.

Vo webdizajne najčastejšie dochádza k porušeniu týchto faktorov, pokiaľ je grafické prevedenie rozhrania webových aplikácií príliš inovatívne, a neodpovedá zaužívaným štandardom. Ako už vieme, používatelia trávajú drvivú väčšinu času na iných stránkach, nie na tej našej, preto práve my musíme prispôbiť náš web ich zvyklostiam a očakávaniam. Štúdiom literatúry a sledovaním iných webových stránok by sme si mali vytvoriť prehľad o tom, či je zaužívané, ako jednotlivé prvky webových stránok typicky vyzerajú, a kde bývajú najčastejšie umiestnené. Špeciálnu pozornosť pri tom treba venovať **ovládacím a navigačným prvkom**, ktoré spájajú našu webovú stránku dohromady a umožňujú prístup k jednotlivým jej častiam. Ťažkú prácu potrebnú pre úspešnú navigáciu na webovej stránke by mal odvieť pri jej tvorbe webdizajner, nie používateľ stránky.

Ďalším dôležitým faktorom je spôsob, akým používatelia web používajú, ich **návyky a očakávania**, ktoré majú na webové stránky. Ale tiež aj to ako fungujú vyhľadávače a rôzne iné služby cez ktoré používatelia na náš web pristupujú. Na tieto otázky nájdeme opäť odpoveď v literatúre týkajúcej sa použiteľnosti. Existujú napr. štúdie

sledujúce pohyb očí používateľa, ktoré nás utvrdzujú v tom, že text treba dobre štruktúrovať, aby sme uľahčili nelineárne čítanie, ako sme sa už dozvedeli. Mnohé o tom, ako používatelia pracujú s našou stránkou sa môžeme dozvedieť aj s použitím analytických nástrojov pre sledovanie návštevnosti (napr. Google Analytics). Môžeme tu napr. sledovať aké navigačné cesty sú často využívané (ale aj mnohé iné užitočné údaje). Ak je niektorá z často používaných ciest prídlhá (napr. tretia úroveň v navigačnom menu), môžeme zareagovať a preradiť položku v menu, čím uľahčíme používanie stránky.

Samostatnou kapitolou je **chybovosť**. Každý sa občas pomýli a ak sa tak stane, je dôležité aby webová aplikácia vhodne „zareagovala“ a pomohla nám problém vyriešiť. Pokiaľ sa používateľ pomýli počas zadávania vstupu, nemal by systém už zadané dáta vymazať a tým prinútiť používateľa zadať ich odznova. Pri práci s webom sa napr. môže stať, že vyplníme sériu formulárov, ale pred finálnym odoslaním sme vyrušení a po chvíli neaktivity prihlásenie expiruje. Ani v takomto prípade by po opätovnom prihlásení nemal byť používateľ prinútený vyplniť všetko odznova. Pokiaľ je webová (alebo akákoľvek iná) aplikácia neodladená a poruchová, na používateľa pôsobí odrádzajúco.

Pokiaľ ku chybe dôjde, systém by mal vždy zobrazit' chybové hlásenie, a na chybu používateľa upozorniť. Hlásenie by malo byť jasné a zrozumiteľné. Ak napr. aplikácia „vráti“ používateľovi na opravu nesprávne vyplnený formulár, je potrebné na toto navrchu dokumentu výrazným spôsobom upozorniť, uviesť dôvod, prečo bol formulár vrátený, a prípadne poradiť ako chybu napraviť („heslo musí obsahovať čísla, veľké aj malé písmená“ je oveľa lepšie ako „nesprávne heslo“). Chybové hlásenie by teda malo vždy informovať o tom, čo robiť za účelom odstránenia chyby. Rozlišujeme pritom dva druhy chýb:

**chyba vstupu:** túto vyvolal používateľ nesprávnym zadaním vstupu, alebo nesprávnou interakciou so aplikáciou. V tomto prípade by chybové hlásenie malo obsahovať postupnosť krokov, pomocou ktorej používateľ sám chybu napraviť;

**chyba aplikácie:** za chybu nemôže používateľ, ale chyba nastala pri vykonávaní programu, napr. nie je možné sa pripojiť do databázy, a pod. Ak používateľ nemôže sám odstrániť chybu, hlásenie by malo používateľa inštruovať, že má kontaktovať správcu systému (prípadne ho ubezpečiť, že na odstránení chyby sa už pracuje, lepšie aj nejaký odhad času potrebného na jej odstránenie).

**Spokojnosť** je posledným a najhlavnejším faktorom použiteľnosti. Ostatné štyri faktory práve k dosiahnutiu spokojnosti smerujú. Samy osebe však spokojnosť nezaručujú. Mali by sme sa preto spokojnosťou zaoberať primárne a sústrediť sa na ňu. Predovšetkým v kontexte webu často platí, že ak používateľ nie je spokojný, odíde. Zvyčajne má totiž na výber. K spokojnosti používateľov môže pridať všeličo, od pekného dizajnu, po zaujímavé služby a nástroje. Špeciálnu pozornosť musíme tiež venovať obsahu, ktorý môže pôsobiť odpudzujúco, napr. príliš veľa reklamy, prípadne reklama na nevhodnom mieste (miesto navigácie, a pod.)

Použiteľnosť je inžinierska disciplína. Vysokú mieru použiteľnosti v praxi dosahujeme aplikovaním metodológie, ktorá sa nazýva **dizajn zameraný na používateľa** (user centered design) pri ktorej sa ako tvorcovia nikdy neriadime tým, či sa náš produkt páči a dobre používa nám, ale naopak snažíme sa vyhodnocovať návrh vzhľadom na bežného používateľa a „vcítiť“ sa do jeho pozície. Pri návrhu každého ovládacieho prvku preto musíme zvážiť, či sa nám zdá intuitívny a jednoduchý na použitie, lebo taký naozaj je, alebo je to naopak tým, že dopredu vieme ako funguje lebo sme ho sami navrhli. Pritom sa snažíme docieľiť naplnenie jednotlivých faktorov použiteľnosti, a v konečnom dôsledku spokojnosť s našim produktom.

Aby sme sa o úspechu uistili, musíme často a opakovane **testovať použiteľnosť** nášho produktu. Testovanie je pritom potrebné previesť s niekoľkými rôznymi používateľmi, ktorí nikdy nemali byť dopredu s testovaným produktom oboznámení (okrem testovania naučiteľnosti). Predovšetkým by sme nikdy nemali testovať sami na sebe, prípadne na členoch vývojového tímu a kolegoch, ktorí sú už so systémom čiastočne oboznámení. Je to preto, že otestovať potrebujeme práve pochopenie návrhu **nezainteresovaným používateľom**.

Samotné testovanie prevedieme napr. tak, že testera inštruujeme aby sa pokúsil s našim nástrojom vykonať určitú úlohu. Sledujeme, ako si pritom počína, či sa mu



Eye tracker (zariadenie na sledovanie pohybu očí) ukazuje ako čitatelia doslova „skáču“ očami z miesta na miesto (červenou je označené, kam sa používateľ pozeral najdlhšie). Zdroj: Johny Holland Magazine: <http://johnnyholland.org/>

Problémom z hľadiska použiteľnosti je formulárové tlačidlo **Reset**. Ak naň používateľ klikne omylom, príde o všetky dáta, ktoré do formuláru vyplnil, a môže vyplňať odznova. Mali by sme ho preto použiť len výnimočne, ak je naozaj opodstatnené. Zároveň pomocou JavaScriptu je vhodné implementovať kontrolnú otázku „Naozaj chcete vymazať?“

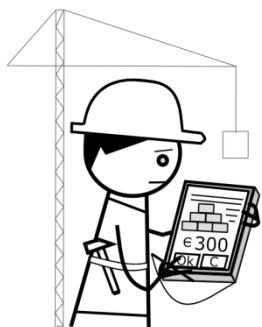
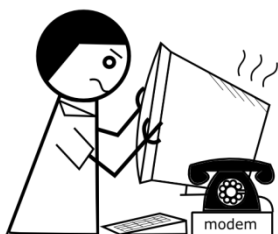


Dizajn zameraný na používateľa.



Testovanie použiteľnosti.

Desať najväčších chýb podľa Jakoba Nielsena:  
<http://www.useit.com/alertbox/9605.html>



Používatelia, ktorí sú v danej situácii obmedzovaní.

darí, alebo nie. Sústreďíme sa na chyby, ktoré robí a komplikácie, s ktorými si musí poradiť. Tieto si poznačíme a neskôr sa ich pokúsime odstrániť a používanie zjednodušiť. Dôležité je navrhnuť testovacie úlohy tak, aby sme pokryli celú funkčnosť nástroja. Testovať je potrebné opakovane, aj počas vývoja systému, akonáhle máme k dispozícii prvé prototypy rozhraní, s ktorými prídu používatelia do styku. Takto si už počas vývoja môžeme overiť, či návrh a vývoj postupuje správnym smerom, vykonať potrebné zmeny a úpravy včas, a ušetriť tak v budúcnosti čas a prostriedky.

Zoznámili sme sa s potrebnou teóriou a metodológiou z oblasti prístupnosti. Táto má veľké množstvo praktických dôsledkov, na ktoré sa v praxi pri vývoji webových stránok a webových aplikácií musíme zamerať. Tieto sú svojím rozsahom výrazne nad rámec tohto materiálu, spomeňme aspoň zoznam desiatich najväčších chýb, ktoré zostavil Jakob Nielsen:

1. zlé vyhľadávanie. Vyhľadávanie na stránkach, ktoré si nedokáže poradiť s preklepmi, so skloňovaním, a inými variantmi hľadaných slov. Rovnako, keď vyhľadávanie vracia na popredných pozíciách aj dokumenty, ktoré nemajú veľký súvis s hľadanými slovami;
2. PDF, či iné formáty súborov miesto HTML. Tieto je možné použiť ako verziu pre tlač, ale nie ako primárny formát pre webovú stránku. Používatelia neradi spúšťajú či dokonca inštalujú nový softvér aby si mohli dokumenty prečítať;
3. navštívené odkazy nezmenia farbu. Používateľ po chvíli čítania a navigácie po dokumente už nevie, ktoré odkazy navštívil, a ktoré ešte nie;
4. zle štruktúrovaný text, v ktorom sa ťažko orientuje;
5. fixná veľkosť písma, ktorá sa nedá zmeniť. Preto by sme veľkosť písma nemali nikdy nastavovať v absolútnych jednotkách, ani v pixeloch;
6. nevhodne volený titulok stránky. Titulok stránky (element `title`), ktorý zvolíme, preberú webové vyhľadávače a použijú ho ako text odkazu vo výsledkoch vyhľadávania. Nevhodne volený titulok (predovšetkým nedostatočne opisný) spôsobí, že k nám cez vyhľadávače používatelia vôbec neprídu, alebo prídu s očakávaním iného obsahu, a odídu sklamaní. Pritom používatelia skutočne sa zaujímajú o náš obsah stránku minú;
7. čokoľvek, čo vyzerá ako reklama. Reklama odrádza, prípadne býva často len „trpená“ a ignorovaná. Pokiaľ dáme na stránku príliš veľa reklamy, je možné, že používateľ odíde. Vyhnúť sa treba tiež obsahu, ktorý vyzerá ako webová reklama, aj keď ňou v skutočnosti nie je (bude ignorovaný);
8. nedodržanie zaužívaných dizajnových konvencií. Pretože používatelia sa len veľmi neradi učia nové veci;
9. otváranie odkazov nových okien prehliadača. Ak sa odkaz otvorí v novom okne, stratí sa história z pôvodného okna, kde sa nachádzal odkaz, na ktorý sme klikli. Navyše, ak sa okno otvorí maximalizované, čo sa často stáva, používateľ si nemusí uvedomiť, že je v novom okne, čo môže byť mätúce. Navyše, okná, ktoré sa otvárajú, treba zatvárať a túto prácu už musí vykonať používateľ sám.
10. Nezodpovedanie používateľových otázok. Ak používateľ nenájde informáciu, ktorú hľadá, je sklamaný. Typickým príkladom je, ak na stránke s popisom nejakých produktov, či tovarov nevedieme cenu. Práve tá používateľa väčšinou najviac zaujíma.

## Prístupnosť (accessibility)

**Prístupnosť** je miera kvality, ktorá vyhodnocuje do akej miery sú publikované informácie prístupné aj používateľom, ktorých prístup k týmto informáciám je nejakým spôsobom obmedzený. Táto miera je teda čiastočne príbuzná použiteľnosti, rozdiel je v tom, že sa sústreďuje predovšetkým na zásadu, že všetci používatelia majú mať zabezpečený rovnocenný prístup k informáciám. Táto zásada, a právo na univerzálny prístup k informáciám je dokonca v niektorých krajinách zakotvená v legislatíve. Prístupnosť je často mylne chápaná výlučne ako ústretovosť k zdravotne postihnutým používateľom webu. Aj keď táto skupina pod prístupnosť spadá, v skutočnosti je celá skupina používateľov, ktorí zo zvýšenej prístupnosti dokumentov profitujú oveľa širšia. Rozlišujeme totiž hneď niekoľko druhov obmedzení, ktoré rôznych používateľov môžu postihnúť:

- zdravotné postihnutie** rôzneho druhu: používateľ majúci sťažené videnie, sluch, či nemôže obvyklým spôsobom pracovať s klávesnicou alebo s myšou;
- sťažené pochopenie textu** a orientácia: opäť môže ísť o zdravotné obmedzenie, ale napr. aj o slabšiu znalosť jazyka, v ktorom je text napísaný, prípadne ide o menej skúseného používateľa;
- obmedzenie zariadením:** starší, alebo menej výkonný počítač, staršia verzia prehliadača, textový prehliadač, pomalé internetové pripojenie, mobilný prehliadač alebo iné zariadenie s malým displejom (bez myši a klávesnice), a pod.;
- obmedzenie situáciou:** používateľ je v situácii, kedy sú jeho oči, uši, či ruky zaneprázdnené. Napr. zamestnanec na rušnom, či rizikovom pracovisku, ktorý nemôže celú svoju pozornosť venovať práci s počítačom, používateľ v hlučnom prostredí, a pod.

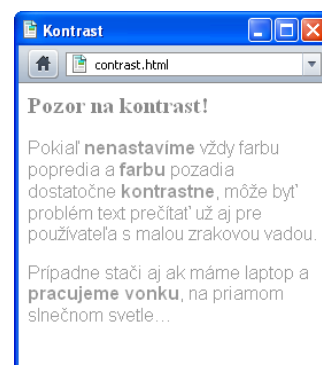
Venovať zvýšenú pozornosť prístupnosti je preto prínosom nielen pre používateľov so zdravotným postihnutím, ale uľahčí používanie webových stránok a prístup k informáciám všetkým používateľom. Zvláštnou skupinou s mimoriadne veľkým úžitkom z prístupnosti sú **začiatocníci**, ale rovnako aj **používatelia pracujúci s webom veľa**, ktorí s ním potrebujú pracovať rýchlo a efektívne.

V niektorých krajinách môže byť prístupnosť vyžadovaná zákonom. Na súkromných weboch je to, pravda, vyžadované len zriedka, ale napr. v EU aj v USA existuje legislatíva vyžadujúca splnenie istého štandardu prístupnosti na weboch štátnych a verejných inštitúcií.

Špeciálnym nástrojom pre kontrolu miery použiteľnosti webových dokumentov sú tzv. **Web Content Accessibility Guidelines (WCAG)** z dielne W3C konzorcia. Ide o pomerne rozsiahly dokument obsahujúci sériu podmienok, ktoré musí každá webová stránka spĺňať, ak má byť prístupná. WCAG bolo štandardizované, a W3C sa snaží docieľiť jeho používanie v praxi, tj. tvorbu prístupných dokumentov, ktoré mu vyhovujú, nedarí sa to však presadiť v dostatočnej miere.

V najnovšej verzii WCAG 2.0, boli preto podmienky prístupnosti, nachádzajúce sa v tomto dokumente, upravené a spresnené, tak aby bolo možné vo vyššej miere kontrolovať ich aj automatizovane. Zároveň bol dokument výrazne sprehľadnený. Podmienky sú rozdelené podľa štyroch základných princípov:

- rozpoznateľnosť:** každý používateľ musí mať rovnakú možnosť pre vnem obsahu. Popredie a pozadie (u textu, obrázkov, ale aj audia) musí byť dostatočne kontrastné. Okrem presne stanovených výnimiek (CAPTCHA, dekoratívny obsah, a pod.), je nutné vždy poskytnúť textový ekvivalent netextového obsahu, a to vrátane informácií zachytených vo farbe, či rozmiestnení a usporiadaní jednotlivých častí obsahu na stránke. Pre médiá prebiehajúce v čase (audio, video) je nutné alternatívny obsah poskytnúť synchronizovane (podobne ako titulky pre nepočujúcich vo filme);
- ovládateľnosť:** prvky používateľského rozhrania musia byť ľahko ovládateľné, viacerými spôsobmi. Základné ovládacie prvky musia byť použiteľné klávesnicou vo forme „klávesových skratiek“. Používateľ musí mať dostatok času na prečítanie a použitie obsahu, ako aj na jednotlivé akcie ovládania. Webová stránka musí mať intuitívne navigačné prvky, a používateľ vždy vie, v ktorej je časti sa práve nachádza. Význam hypertextových odkazov musí byť zreteľný z ich textu. Do tejto časti patrí a požiadavka, že webová stránka nesmie obsahovať blikajúci obsah, ktorý by mohol spôsobiť epileptický záchvat.
- pochopiteľnosť:** obsah, ale aj ovládanie webovej stránky musia byť ľahko pochopiteľné. Jazyk textu musí byť vyznačený (atribút lang), a text musí byť písaný čo možno najjednoduchšie, bez nezvyčajných slov a zbytočných odborných termínov. Skratky musia byť vysvetlené. Navigácia musí byť konzistentná a ľahko rozlíšiteľná od obsahu. Ovládanie musí byť dostatočne predvídateľné. Webová stránka by mala napomáhať pri chybách používateľského vstupu;
- kompatibilita:** maximalizovať kompatibilitu so súčasnými, ako aj s budúcimi prehliadačmi, vrátane špeciálnych nástrojov slúžiacich k prístupu na web telesne postihnutým. Tvorit' validné dokumenty podľa príslušných špecifikácií, vyhnuť sa funkcionalite, ktorá sa považuje za zastaralú (zastaralosť, z angl. *deprecated*, niektorých elementov, atribútov, či iných funkcií je vždy označená v špecifikáciách od W3C).



Takýto kontrast je silne nedostatočný.



Informácia kódovaná iba farbou.

Pod každým princípom sa nachádza niekoľko presne formulovaných podmienok, ktoré je nutné dodržať. Tieto podmienky majú priradenú jednu z troch úrovní dôležitosti „A“ (veľmi dôležité), „AA“ (stredne dôležité) a „AAA“ (menej dôležité). Splňať WCAG je teda možné v troch rôznych úrovniach:

- A, základná úroveň prístupnosti:** ak sú splnené všetky podmienky označené „A“, túto úroveň by mala splňať každá webová stránka;
- AA, stredná úroveň prístupnosti:** ak sú splnené všetky podmienky označené „A“ a navyše aj všetky označené „AA“;
- AAA, vysoká úroveň prístupnosti:** ak sú splnené úplne všetky podmienky. Táto úroveň predstavuje vysoko kvalitné stránky, veľmi dobre usporiadané aj pre potreby obmedzovaných používateľov.

Ideálne by každý webový dokument mal splňať najvyššiu úroveň AAA, a mali by sme sa o to vždy snažiť. Keď to však v praxi nie je možné docieľiť, napr. z dôvodu navýšenia produkčných nákladov spojených s testovaním a zvyšovaním prístupnosti, uspokojíme sa aj s nižšou úrovňou. Zamyslime sa však aj nad tým, že zvýšená prístupnosť môže mať za následok zvýšenú návštevnosť a popularitu stránky, a tým pádom sa investícia vráti.

Príkladom veľmi konkrétnej, a dobre automaticky kontrolovateľnej podmienky je požiadavka, že kontrast popredia a pozadia u textu musí byť aspoň 4,5:1, pre dosiahnutie úrovne „AA“, prípadne aspoň 7:1, pre úroveň „AAA“. Žiaľ, nie všetky podmienky vieme jednoducho strojovo spracovať a vyhodnotiť. Napr. implementovať plne automatizovaný nástroj, ktorý vyhodnotí, či poskytnutý textový ekvivalent naozaj v dostatočnej miere vystihuje obrázok, či audiovizuálny obsah je prakticky nemožné. Preto je potrebné úroveň prístupnosti vždy otestovať manuálne, vyhodnotením všetkých podmienok.

**Testovanie prístupnosti** je pomerne náročná úloha. Bežne sa stretávame s tým, že výrobca alebo prevádzkovateľ deklaruje, že stránka podmienky splňa, pričom to nie je úplná pravda. Napriek tomu, že existujú rôzne nástroje na kontrolu týchto podmienok, už vieme, že úplná automatická kontrola nie je možná. Až po uistení sa, že stránka splňa všetky kontrolné body, máme istotu, že stránka vyhovuje WCAG, a je teda skutočne prístupná. Našťastie, W3C pre uľahčenie testovania ponúka hneď niekoľko doplnkových dokumentov, v ktorých je každá podmienka detailne vysvetlená a ku každej je priložený aj návod, ako ju splniť (pozri zoznam literatúry).

Prístupnosť je citlivá a dôležitá oblasť, s ktorou by mal byť každý, kto sa podieľa na webovej tvorbe oboznámený a stotožnený. Po prečítaní tejto kapitoly sme ale pochopili aj to, že tvorba prístupných stránok a testovanie prístupnosti sú pomerne náročné činnosti. Štandardy prístupnosti preto môžu niekedy zostať dizajnérmi a vývojármi webových aplikácií nepochopené. Toto je do istej miery spôsobené aj charakterom WCAG: mnohé požiadavky sa nedajú automaticky kontrolovať, a je potrebné dôkladné, pre začiatočníka zdĺhavé štúdium špecifikácie. Druhý problém bol čiastočne odbúraný novou verziou špecifikácie WCAG 2.0, ktorá bola výrazne sprehľadnená a niektoré diskutabilné požiadavky boli vypustené.

## Čo sme sa naučili

Zoznámili sme sa s princípmi používateľnosti a prístupnosti - dvoch dôležitých štandardov používaných na vyhodnotenie kvality webových stránok a webovej tvorby. Používateľnosť znamená ľahkosť použitia, a skladá sa z naučiteľnosti, efektivity použitia, zapamätateľnosti, chybovosti a celkovej spokojnosti používateľa. Používateľnosť dosahujeme pomocou *dizajnu zameraného na používateľa* kombinovaného s častým a opakovaným testovaním miery používateľnosti. Prístupnosť sa sústreďuje na prístup používateľa k informáciám, presnejšie, na to, kedy je nejakým spôsobom sťažená. Aj keď prístupnosť je primárne cieľná na zdravotne obmedzovaných účastníkov webu, z jej zvyšovania máme v konečnom dôsledku úžitok všetci. Štandardom prístupnosti sú pravidlá WCAG z dielne konzorcia W3C. Aj keď existujú pomocné nástroje na testovanie prístupnosti, v skutočnosti sa dá poriadne otestovať jedine manuálne. Pri testovaní sa riadime súborom kontrolných bodov WCAG *checklist of checkpoints*, ktorú tie ž vydalo W3C.



Voľným on-line nástrojom na testovanie používateľnosti, ktorý nám môže čiastočne napomôcť je Cynthia Says: <http://www.cynthiasays.com/>



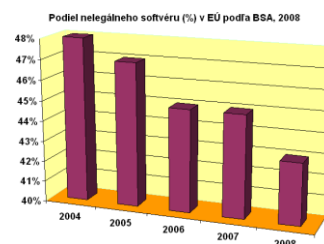
## Kapitola 8: Praktické cvičenia

Tu uvádzame praktické cvičenia, ktoré čitateľovi napomôžu lepšie si osvojiť učivo, s ktorým sme sa v module zoznámili. Tieto zadania je možné robiť až nakoniec, ale určite je lepšie vyskúšať si vždy niekoľko z nich s každou jednou kapitolou.

<b>Zadanie 1</b>	<b>Prvé XHTML.</b> Inšpirujte sa príkladmi v Kapitole 2 a vytvorte na svojom počítači XHTML súbor. Nazvite ho „Cvičenia z web-dizajnu“. Rovnaký názov použite v hlavičke, aj v nadpise v tele stránky. Na stránku vložte odstavec s textom „Ahoj svet!“
<b>Zadanie 2</b>	<b>Validátor.</b> Vložte do stránky úvodné XHTML deklarácie, a nastavte správne kódovanie a jazyk dokumentu. Správnosť stránky overte W3C on-line validátorom. Stránku overte aj po každom ďalšom zadaní.
<b>Zadanie 3</b>	<b>Štruktúra.</b> Pre každé zo zadaní 1-3, ako aj pre každé ďalšie, vytvorte nadpis druhej úrovne a pod ním 3-4 vetami opíšte, ako ste postupovali. Pokiaľ Vás zadanie žiada do stránky niečo vložiť (a nehovorí kam) vložte to do priestoru pod príslušným podnadpisom.
<b>Zadanie 4</b>	<b>Odkazy.</b> Do stránky vložte odkazy na dve svoje obľúbené stránky. Snažte sa text odkazu formulovať podľa zásad z Kapitoly 6 a 7.
<b>Zadanie 5</b>	<b>Obrázky a odkazy.</b> Z Flickr, Wikipédie, alebo z OpenClipart stiahnite tri obrázky v troch rôznych formátoch. Z každého z nich urobte zmenšeninu a vytvorte v dokumente tri „obrázkové“ odkazy. Použite správny textový ekvivalent.
<b>Zadanie 6</b>	<b>Menu.</b> Pre každý nadpis vytvorte fragmentový identifikátor a pomocou elementu a vložte predeň kotvu. V hornej časti stránky, pod úvodným nadpisom vytvorte zoznam s odkazmi na jednotlivé podnadvisy.
<b>Zadanie 7</b>	<b>Farebná schéma, fonty.</b> Vyberte si farbu popredia a farbu pozadia. Nastavte tieto farby prostredníctvom prilinkovaného CSS. Špecifikujte tiež fonty a upravte podľa potreby veľkosť nadpisov. Správnosť CSS kódu overte CSS validátorom.
<b>Zadanie 8</b>	<b>CSS Menu.</b> Zmenou vlastností box modelu preformátujte zoznam s menu tak, aby elementy boli v riadku. Vaše menu by mohlo vyzeráť napr. takto:  <u><a href="#">Prvé XHTML</a></u>   <u><a href="#">Validátor</a></u>   <u><a href="#">Štruktúra</a></u>   <u><a href="#">Odkazy</a></u>   ...  Odkazy v menu a odkazy v texte naformátujte rozlične tak, ako sme to popísali v príklade v Kapitole 4. Zdefinujte pre každý druh odkazov aj hover-efekt. Nezabudnite odvalidovať!
<b>Zadanie 9</b>	<b>Zložitý obrázok.</b> Vložte do dokumentu obrázok s grafom, ktorý vidíme v pravo. Snažte sa dodržať pravidlá prístupnosti, a doplňte textový ekvivalent, ktorý dostatočne vystihuje informáciu, ktorá je zachytená na obrázku. Textový ekvivalent vložte pomocou atribútu longdesc, ale tiež ako d-link.
<b>Zadanie 10</b>	<b>SVG.</b> V katalógu OpenClipart nájdite obrázok vo formáte SVG a vložte do dokumentu. Prekonvertujte obrázok do rastrového formátu PNG. SVG obrázok vložte tak, aby sa v prípade nedostupnosti zobrazilo PNG. Nezabudnite na textový ekvivalent.




Knižnica obrázkov Open-Clipart: <http://www.openclipart.org/>



Zdroj: <http://global.bsa.org/globalpiracy2008/studies/globaldiracv2008.pdf>

Video súbor rekurzia.vmw  
nájdete na adrese:  
<http://virl.com/rekurzia/>

Picasa:  
<http://picasaweb.google.sk/>

<b>Zadanie 11</b>	<b>Video.</b> Umiestnite do svojej webovej stránky video rekurzie. Súbor s videom má názov rekurzia.vmw. Snažte sa pritom dodržať pravidlá prístupnosti. 
<b>Zadanie 12</b>	<b>Flashový banner.</b> Pomocou on-line aplikácie <a href="http://www.effectgenerator.com/">www.effectgenerator.com/</a> vytvorte reklamný banner pre svoju (školskú) webovú stránku vo formáte Flash. Výsledný súbor si uložte do počítača a vložte ho Vášho dokumentu. Poskytnite opisný textový ekvivalent. Aký ekvivalent okrem textového by sme mali ešte poskytnúť?
<b>Zadanie 13</b>	<b>Audio.</b> Na webovej adrese <a href="http://virl.com/mp3/">http://virl.com/mp3/</a> je uvedených niekoľko spôsobov ako vložiť mp3 súbor do webovej stránky. Vyskúšajte niektorú z možností. Ani po tomto kroku nezabudnite overiť správnosť výsledného XHTML kódu.
<b>Zadanie 14</b>	<b>Skúšame Picasu.</b> Zaregistrujte sa v aplikácii Picasa. Nahrajte do albumu niekoľko fotografií obsahujúcich ľudské tváre. Priradte rozpoznaným tváram mená. Vyskúšajte aj ďalšie možnosti: zdieľanie albumov, prepájanie galérií, komentovanie obrázkov, a pod.
<b>Zadanie 15</b>	<b>Hľadáme multimédiá.</b> V tomto zadaní budeme porovnávať obtiažnosť vyhľadávania rôznych drohov obsahu: <ul style="list-style-type: none"><li>• nájdite na internete stránky, z ktorých sa môžete dozvedieť informácie o rôznych nástrojoch na prácu s multimédiami;</li><li>• štátny pedagogický ústav je riešiteľom projektu DVUi. Nájdite na internete jeho logo;</li><li>• žiakom by ste radi pustili ukážku štatnej hymny Slovenskej republiky. Najdite jej zvukovú podobu na internete. Pre porovnanie skúste nájsť aj štátnu hymnu Etiópie.</li></ul> Porovnajme navzájom výsledky vyhľadávania. Vyhľadanie ktorej informácie bolo najjednoduchšie? Pri ktorom vyhľadávaní vám vyhľadávací stroj vrátil najviac irelevantných informácií? Pokúste sa to zdôvodniť.
<b>Zadanie 16</b>	<b>Štruktúra a štýl textu.</b> Ak ste postupovali striktné podľa zadania 3, má Váš text už celkom dobrú štruktúru. Pomocou elementu <code>strong</code> zvýraznite v texte najdôležitejšie kľúčové slová. Dajte pozor, a neprežeňte to. Upravme teraz text do štýlu obrátenej pyramídy: <ul style="list-style-type: none"><li>• napíšte jednoduchstavecový sumár (5-7 viet) a umiestnite ho pod nadpis Vašej webovej stránky;</li><li>• ak obsahuje niektorá podsekcia dlhší text, napíšte krátky sumár aj pre ňu;</li><li>• Pred každý odstavec, vrátane sumárov, ktoré ste práve napísali, dopíšte ešte jednu vetu, ktorá ho zhrňa. Pokúste sa každú z týchto viet začať nosným slovom (pojmom), o ktorom daný odstavec hovorí.</li></ul>
<b>Zadanie 13</b>	<b>Kontrola prístupnosti.</b> Na záver prekontrolujte prístupnosť dokumentu podľa požiadaviek WCAG. Dosiahli ste skutočne prístupný dokument? Ak nie, viete ako prístupnosť zlepšiť?

## Kapitola 9: Čo sme sa naučili v tomto module

Absolvovaním modulu sme získali základný prehľad o architektúre a fungovaní webu, o najdôležitejších webových technológiách a publikovaní na webe. Zoznámili sme sa s možnosťami webového obsahu, a so základnými pravidlami pre jeho správne publikovanie, od textu, cez obrázky a grafiku, po multimediálny obsah. Veľkú pozornosť sme venovali tiež problematike kvality vo webdizajne a vo webovej tvorbe; osvojili sme si pojmy použiteľnosť a prístupnosť webových dokumentov a aplikácií. Oboznámili sme sa jednotlivými technológiami, ktoré dohromady tvoria web ako internetovú službu. Týchto technológií je pomerne veľký počet, v dôsledku čoho je web značne heterogénnym prostredím. Do webovej tvorby a webového programovania to prináša mnohé špecifiká.

Web je hypermediálny priestor, postavený na možnosti ľubovoľne prepájať webové stránky za pomoci hypertextových odkazov. Základným jazykom pre webový obsah je (X)HTML, určené hlavne na reprezentáciu štruktúry webových dokumentov, na čo ponúka dostatočné možnosti. Prezentačnú vrstvu má na starosti jazyk kaskádových štýlov (CSS), umožňujúci nezávisle a oddelene upraviť dizajn a typografiu dokumentov. CSS dnes ponúka dostatočne bohaté možnosti na to, aby sme nikdy nemuseli siahať napríklad po realizácii viacstĺpcových dizajnov prostredníctvom (X)HTML tabuliek.

Súčasťou webu sú aj obrázky, ale je možné vložiť aj audiovizuálne a interaktívne objekty. Pri vkladaní grafiky musíme správne vybrať formát, podľa typu obrázka. Žiaduca je malá veľkosť a často sa siahajú po stratovej kompresii (JPG), u niektorých typov grafík dosahujú ale lepšie výsledky bezstratové formáty (PNG, GIF). Novinkou je možnosť vkladať vektorové obrázky vo formáte SVG, jeho podpora je však stále slabšia. U multimediálnych objektov musíme zvážiť, že ich podpora nie je natívna, ale vyžaduje plugin, ktorý napr. nemusí byť vždy dostupný. Mali by sme vždy doplniť aj alternatívny obsah, vrátane textových ekvivalentov.

Pri tvorbe webu musíme brať v úvahu zvyklosti a očakávania používateľov. Pre každý druh webových stránok platia iné očakávania. Text je potrebné správne členiť, mal by byť stručný a výstižný. Hodí sa štýl obrátenej pyramídy (sumarizovať na začiatku, písať od všeobecného k špecifickému). Hlavnými mierami kvality pre web sú použiteľnosť a prístupnosť. Tieto nás učia brať ohľad na používateľov, a snažiť sa informácie zachytiť formou, ktorá je uchopiteľná aj v prípade, že je práca s webom nejakým spôsobom obmedzovaná. Tieto štandardy je potrebné si osvojiť, a pri vývoji opakovane mieru prístupnosti a použiteľnosti testovať.

Špecifikom webu je napr. fakt, že komunikácia cez HTTP je bezstavová a je rozdelená do požiadaviek a odpovedí. Toto prináša niektoré obmedzenia, a vytvára priestor pre nové technológie, ktoré ich pomáhajú prekonávať (napr. cookies, RSS, AJAX).

Multimediálny formát Flash je v súčasnosti široko podporovaný a je de facto štandardom pre multimediálny obsah.

Viac než 99% času tvoria používatelia na iných stránkach, preto sa musíme prispôsobiť tomu, čo je zaužívané.

## Literatúra a použité zdroje

- [1] Lynch, P. J., Horton, S. (2009) *Web Style Guide 3<sup>rd</sup> Edition: Basic Design Principles for Creating Web Sites*. Yale University Press. On-line verzia: <http://webstyleguide.com/wsg3/>
- [2] Pamberton, S., a kol. (2002) *XHTML™ 1.0 The Extensible HyperText Markup Language (Second Edition): A Reformulation of HTML 4 in XML 1.0.*, W3C Recommendation. On-line verzia: <http://www.w3.org/TR/xhtml1/>
- [3] Raggett, D., Le Hors, A., Jacobs, I. (1999) *HTML 4.01 Specification*. W3C Recommendation. On-line verzia: <http://www.w3.org/TR/HTML401/>
- [4] Bos, B., Čelik, T., Hickson, I., Lie, H. W., (2009) *Cascading Style Sheets Level 2 Revision 1 (CSS 2.1) Specification*. W3C Candidate Recommendation. On-line verzia: <http://www.w3.org/TR/CSS2/>
- [5] Caldwell, B., Cooper, M., Reid, L. G., Vandeheiden, G. (1999) *Web Content Accessibility Guidelines 2.0*. W3C Recommendation. On-line verzia: <http://www.w3.org/TR/WCAG20/>
- [6] Vandeheiden, G., Reid, L. G., Caldwell, B., Henry, S. L., (1999) *How to Meet WCAG 2.0*. On-line verzia: <http://www.w3.org/WAI/WCAG20/quickref/>
- [7] Nielsen, J. *Alertbox: Current Issues in Web Usability*. ISSN 1548-5552. On-line verzia: <http://www.useit.com/alertbox/>
- [8] Krug, S. (2003) *Weg design: Nenut'te uživatele přemýšlet*. Computer Press.
- [9] Murray, J. D., van Ryper, W. (2004) *Encyklopédie grafických formátů II*. Computer Press.
- [10] Špinar, D. (2004) *Tvoříme přístupné webové stránky*. Zoner Press.

Tento študijný materiál vznikol ako súčasť národného projektu Ďalšie vzdelávanie učiteľov základných škôl a stredných škôl v predmete informatika v rámci Aktivity „Ďalšie vzdelávanie kvalifikovaných učiteľov informatiky na 2. stupni ZŠ a na SŠ“.

Autori © RNDr. Martin Homola  
Mgr. Ján Guniš

Ilustrácie © Michaela Danišová

Fotografie Wikimedia Commons (Uldis Bojārs, Robert Scoble, Kmosman)  
WebStyleGuide.com  
Uselt.com

Názov Ďalšie vzdelávanie učiteľov základných škôl a stredných škôl v predmete informatika

Podnázov Webové technológie a publikovanie na webe 1

Študijný materiál prešiel recenzným pokračovaním.

Recenzenti Doc. Ing. Ľudovít Trajtel', PhD.  
PaedDr. Roman Hrušecký

Počet strán 36

Náklad 400 ks

**Prvé vydanie, Bratislava 2009**

Všetky práva vyhradené.

Toto dielo ani žiadnu jeho časť nemožno reprodukovat' bez súhlasu majiteľa práv.

Vydal Štátny pedagogický ústav, Pluhová 8, 830 00 Bratislava, v súčinnosti s Univerzitou Pavla Jozefa Šafárika v Košiciach, Univerzitou Komenského v Bratislave, Univerzitou Konštantína Filozofa v Nitre, Univerzitou Mateja Bela v Banskej Bystrici a Žilinskou univerzitou v Žiline

Vytlačil BRATIA SABOVCI, s r.o., Zvolen

**ISBN 978-80-8118-016-3**