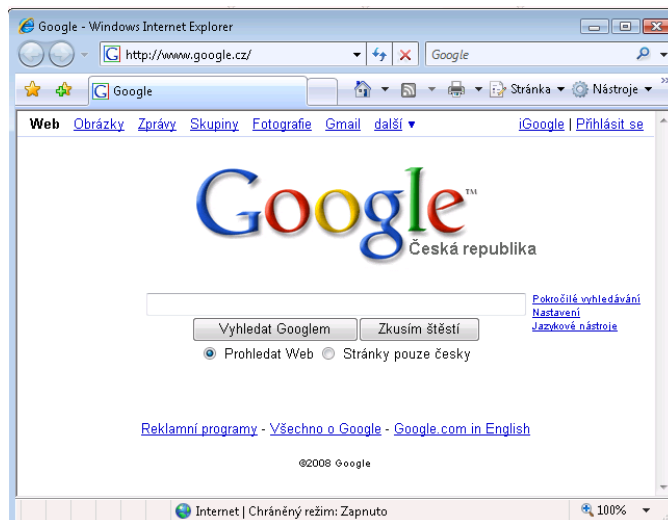
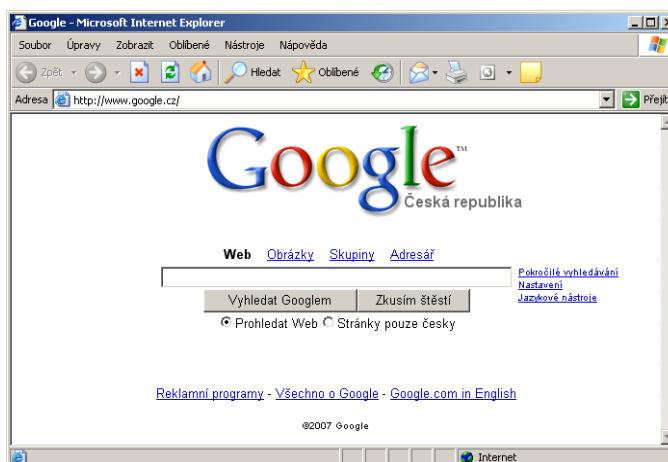
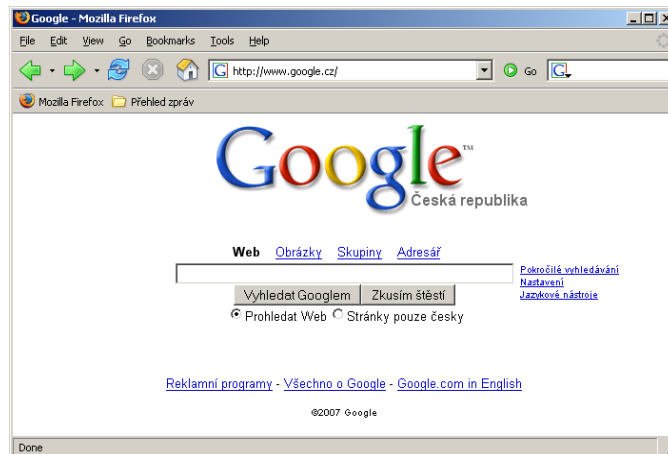


Střední odborná škola a střední odborné učiliště Kuřim, s.r.o.

UČEBNÍ TEXT „TVORBA WEBOVÝCH STRÁNEK“



OBSAH

OBSAH:	2
1. ÚVOD	4
1.1 Charakteristika učebního textu	4
1.2 Cílová skupina	4
1.3 Cíle učebnice – profil absolventa	4
1.4 Hodinová dotace	4
1.5 Vstupní požadavky (úroveň)	4
1.6 Úspěšné zvládnutí	5
1.7 Použité formátování a struktura	5
2. WEB ANEB JE TO NA INTERNETU	6
2.1 Jak vytvořit WWW stránku	6
2.2 Uspořádání klient–server	7
2.2.1 Klient (browser, prohlížeč)	7
2.2.2 Server	7
2.3 Standardy pro Web	7
2.3.1 HTTP (HyperText Transport Protocol)	7
2.3.2 HTML (HyperText Markup Language)	8
2.3.3 XHTML (eXtensible HypertextMarkup Language)	8
2.3.4 CSS (Cascading Style Sheets)	8
2.3.5 PHP (Hypertext Preprocessor)	9
2.3.6 JavaScript	9
3. ÚVOD DO XHTML	10
3.1 Elementy	10
3.1.1 Blokové elementy	10
3.1.2 Inline elementy	11
3.1.3 Nahrazované elementy	11
3.1.4 Atributy	11
3.1.5 Entity	11
3.2 Pravidla XHTML	12
3.3 Kostra stránky	13
4. ODKAZY - URL	14
4.1 Použití odkazů na stránce	14
4.2 Odkaz na jinou stránku	15
4.2.1 Absolutní odkaz	15
4.2.2 Relativní odkaz	15
4.3 Odkaz na konkrétní místo dokumentu	15
4.4 Otevření odkazu v novém okně	16
4.4.1 Způsob otevření nového okna:	16
4.5 Titulek odkazu	16
5. NADPISY	18
6. SEZNAMY	19
6.1 Neuspořádaný seznam	19
6.2 Uspořádaný seznam	19
7. TABULKY	20
7.1 Nadpis tabulky	21
7.2 Slučování buněk	21
8. OBRÁZKY	23
8.1.1 Grafika pro WWW stránky	23
8.1.2 Jaký zvolit grafický formát?	23
8.1.3 Grafický editor	24
8.2 Vložení obrázku v XHTML	24
8.3 Velikost obrázku	24
9. META TAGY	26
9.1 Meta charset – kódování češtiny	26

9.2	Meta description – popis stránky	26
9.3	Meta keywords – klíčová slova stránky	26
9.4	Meta author – autor stránky	27
9.5	Meta language – jazyk stránky.....	27
9.6	Další meta tagy	27
10.	ÚVOD DO CSS.....	28
10.1	Historie.....	28
10.2	K čemu je to dobré?	28
10.3	Začínáme.....	28
10.3.1	Dědičnost	29
10.3.2	Komentáře.....	29
10.3.3	Připojení stylu k dokumentu	29
10.3.4	Váha stylů	30
10.4	Třídy a identifikátory	30
10.4.1	Třída class v CSS	30
10.4.2	Pseudotřídy	31
10.4.3	Identifikátor <i>id</i> v CSS	31
10.5	Příklady zápisu (selektory).....	32
10.6	Jednotky	33
10.6.1	Délkové jednotky	33
10.6.2	Relativní jednotky:.....	33
10.6.3	Absolutní jednotky:.....	33
10.6.4	Procenta	33
10.7	Barvy.....	33
11.	PŘEHLED VLASTNOSTÍ CSS	34
11.1	Písmo (Font)	34
11.2	Text / odstavec	35
11.3	Barvy a pozadí	36
11.4	Velikost a obtékání	37
11.5	Okraje	38
11.6	Rámečky	38
11.7	Seznamy.....	40
11.8	Pozicování.....	40
11.9	Tabulky	41
12.	PRAVIDLA PRO TVORBU PŘÍSTUPNÉHO WEBU	43
12.1	obsah webových stránek je dostupný a čitelný	43
12.2	Práci s webovou stránkou řídí uživatel	44
12.3	Informace jsou srozumitelné a přehledné.....	44
12.4	Ovládání webu je jasné a pochopitelné	45
12.5	Odkazy jsou zřetelné a návodné.....	45
12.6	Kód je technicky způsobilý a strukturovaný	45
13.	VALIDNÍ KÓD	47
13.1	Co může způsobit nevalidní kód.....	47
13.2	Validátor W3C.....	47
14.	DOMÉNA A HOSTING	48
14.1	Doména.....	48
14.1.1	Doména prvního řádu	48
14.1.2	Doména druhého řádu.....	48
14.1.3	Doména třetího řádu	48
14.2	Hosting.....	48
14.3	Umístění stránek	48
	SLOVNÍČEK POJMŮ.....	50
	DOPORUČENÁ STUDIJNÍ LITERATURA.....	52

1. ÚVOD

1.1 CHARAKTERISTIKA UČEBNÍHO TEXTU

Absolvent by měl být schopen pomocí vhodného nástroje vytvářet internetové prezentace v souladu s webovými standardy a estetickými doporučeními.

Součástí učebnice je představení základních způsobů a technik publikování na WWW, pravidel a doporučení k tvorbě stránek a dostupného programového vybavení.

Učebnici je však možno použít i pro každého, kdo chce využít obrovský potenciál internetové služby World Wide Web. Pro každého studenta, který má zájem tvořit WWW stránky.

1.2 CÍLOVÁ SKUPINA

Učební text je určen studentům, středních odborných škol, středních odborných učilišť popřípadě gymnázií. Konkrétně těm studentům, kteří se rozhodnou maturovat z informačně technologického základu.

1.3 CÍLE UČEBNICE – PROFIL ABSOLVENTA

Cílem učebního textu není vychovat - vyučit profesionální webdesignéry a webmastery, ale spíše vybavit žáky takovými znalostmi a dovednostmi, které jim umožní vytvářet přístupné a esteticky hodnotné weby. Ukázat jim jaké jsou možnosti dnešního webdesignu. A v neposlední řadě je též připravit na příslušnou část maturity z informačně technologického základu.

1.4 HODINOVÁ DOTACE

Předpokládaná hodinová dotace je cca 34 hodin – prezenčně. To je při dvouhodinové dotaci týdně asi jedno pololetí. Dále se předpokládá samostatná domácí práce převážně na cvičeních, úkolech a na závěrečném projektu. Celkový rozsah domácí práce je cca 20 hodin. Záleží ovšem na schopnostech studenta.

1.5 VSTUPNÍ POŽADAVKY (ÚROVEŇ)

Student by měl mít základní dovednosti a znalosti v ovládnání počítače a základní dovednosti v používání Internetu. Ty by měl získat v předchozím studiu. Očekávané zařazení

je mezi 3. a 4. ročníkem. V této době by již student měl mít všechny potřebné předchozí znalosti.

Pokud by měly být naplněny bezesbytku i estetické požadavky je vhodné i umět pracovat s grafickým editorem.

1.6 ÚSPĚŠNÉ ZVLÁDNUTÍ

Na konci studia by měl být schopen student samostatně vytvořit prezentaci, která odpovídá webovým a estetickým standardům. Tuto prezentaci odladit od chyb. Zkontrolovat Validitu a nakonec umístit přes ftp. či webové rozhraní na server.

Student by měl také být schopen splnit úkoly a cvičení, která jsou průběžně uvedena v učebním textu.

1.7 POUŽITÉ FORMÁTOVÁNÍ A STRUKTURA

Tlustě – jsou uvedeny důležité pojmy

Kurzívou – jsou uvedeny odborné, technické názvy, zkratky a vysvětlení zkratk

Ve dvojitém rámečku se šedým podkladem jsou uvedeny různé úkoly a cvičení, na kterých by si studenti měli vyzkoušet získané znalosti, ověřit je v praxi nebo získat podněty pro zamyšlení nad problémem kapitoly. Zvládnutí těchto cvičení je předpokladem pro úspěšné pochopení a naučení dané látky.

⇒ Na začátku každé kapitoly je v modrém, plném rámečku se šipkou na začátku uveden cíl dané kapitoly – co by se měl student dozvědět a naučit.
Dále je zde předpokládána hodinová náročnost dané kapitoly.

V jednoduchém rámečku se šedým podkladem jsou neproporčním písmem uváděny zdrojové kódy a příklady technického zápisu kódu.

Na závěr je v učebním textu uveden seznam a vysvětlení použitých zkratk a cizích pojmů. Většinou se jedná o pojmy, které jsou v samotném textu průběžně formátovány kurzívou.

Dále je na konci uvedena doporučená studijní literatura a seznam WWW stránek zabývajících se tvorbou Webu. Tyto informační zdroje by měly sloužit pro získávání dalších znalostí o dané problematice.

2. WEB aneb JE TO NA INTERNETU

⇒ Cíl kapitoly:

Vysvětlit základní principy fungování WWW, způsoby jak vytvořit WWW stránku, uspořádání klient–server, standardy pro Web jazyky (X)HTML, definice stylů CSS.

⇒ Předpokládaná doba studia:

4 vyučovací hodiny

Asi nejpoužívanější internetová služba je dnes *World Wide Web* (toto se překládá jako celosvětová pavučina). Místo tohoto dlouhého názvu se běžně používá termín *web* a pokud Vám někdo řekne „je to na Internetu“, pak má s největší pravděpodobností na mysli právě službu WWW.

Tato služba byla vyvinuta již v roce 1989, primárně měla sloužit pro předávání informací mezi vědeckými pracovníky ve švýcarském *CERN - Centre Europeen pour Recherche Nucleaire* (Středisko jaderných výzkumů).

Podstata webu je sdělovat informace. Přestože je tato podstata od vzniku WWW stále stejná, web se ze své původní jednoduché podoby nevídaně rozšířil. A to nejen co se týče rozsahu, ale i struktury obsahu. Webové stránky dnes obsahují texty, obrázky, zvuky a video (dohromady tvoří multimediální obsah). K obsahu webu má dnes běžně přístup značná část populace obyvatel vyspělých států a web hraje důležitou úlohu nejen při komunikaci, ale také v obchodu, vzdělávání.

Vytvářet webové stránky je po zvládnutí základních znalostí poměrně jednoduchá záležitost. Pokud člověk umí na uživatelské úrovni ovládat počítač a procházet webové stránky, pak pro něj vytvoření jednoduché stránky nebude obtížné. Problém většinou nastává, když se uživatel pustí do rozsáhlejších nebo „na efekty bohatších“ materiálů, které chce na webu zveřejnit.

V publikování na WWW existují určité postupy, pravidla a techniky, které je dobré ovládat. Pokud to myslíme s publikováním na webu vážně, musíme se tato pravidla naučit a respektovat je.

2.1 JAK VYTVOŘIT WWW STRÁNKU

Způsobů či technik, jak WWW stránky vytvářet a publikovat existuje několik. Zde jsou nejpoužívanější způsoby:

⇒ a) **WYSIWYG editor** – "*What you see is what you get*", tedy "co vidíš, to dostaneš".

Práce s nimi je velmi pohodlná. Nevýhodou je, že neukazují úplně přesně to, co z toho vyleze a že občas dělají zmatek v kódu. Zástupci jsou všechny verze *Microsoft FrontPage*, *Dreamweaver*, *Adobe GoLive*, *NVU*.

⇒ b) **editor kódu (X)HTML**, *CSSx* někdy označované jako strukturní editory. Sice jsou poněkud těžkopádné, ten, kdo umí (X)HTML, s nimi dokáže mnohem více než ve WYSIWYG editoru. Program vypadá jako obyčejný textový editor, ale umožňuje trochu

sofistikovanější zadávání tagů. Používají je ale zejména profesionální tvůrci dynamických a jinak specifických stránek, kde je třeba mít naprostou kontrolu nad vytvářeným kódem. Zástupci takových editorů jsou například *HomeSite*, *UltraEdit*, *CoffeCup* nebo české *EasyPad* a *PSPad*.

⇒ c) **export** z textového editoru – kvalita výstupu bývá problematická. Není vhodné používat např. *Microsoft Word* a jeho HTML export jako prostředek pro vytváření stránek (HTML kód je nekvalitní a často se zobrazuje chybně).

⇒ d) **redakční a publikační systém** (*Content Management System – CMS*), *weblog* (nebo též *blog* či internetový deníček).

⇒ e) **přímý zápis** – psaní *XHTML* např. v *Poznámkovém bloku*. Poněkud těžkopádná varianta, která uživateli nijak neulehčuje práci. Je lepší zvolit *freeware* editor kódu.

Běžný uživatel asi nepotřebuje znát všechny výše popsané způsoby, ale pokud chceme, aby uživatel pracoval s při tvorbě webu efektivně, pak by měl být schopen sáhnout po nejvhodnějším způsobu, jak nějaký text na Internetu zveřejnit. Každý uživatel, který bude publikovat na Internetu častěji, si časem osvojí svůj oblíbený prostředek.

2.2 USPOŘÁDÁNÍ KLIENT–SERVER

2.2.1 Klient (browser, prohlížeč)

je **program**, který komunikuje s uživatelem a na základě jeho pokynů se obrací na jednotlivé servery, získává od nich data a zobrazuje je. Nejběžnější klienti: *Microsoft Internet Explorer*, *Netscape Navigator*, *Mozilla*.

2.2.2 Server

je bezobslužný **program**, který přijímá a obsluhuje požadavky klientů. Je potřeba zdůraznit, že WWW server je program, nikoli počítač. Na vlastnostech tohoto programu závisí, co server dovede a jaký má výkon. Nejběžnější servery: *Apache Web Server*, *Microsoft Internet Information Server*, *Zope*.

Protože klienti a servery pocházejí od různých producentů, je velmi důležité dodržování standardů, aby programy byly schopny se navzájem domluvit.

2.3 STANDARDY PRO WEB

2.3.1 HTTP (HyperText Transport Protocol)

definuje pravidla síťové komunikace mezi klientem a serverem. Je postaven na modelu dotaz–odpověď: klient pošle serveru dotaz a ten na něj odpoví, čímž je *HTTP* transakce ukončena.

2.3.2 HTML (HyperText Markup Language)

HTML je značkovací jazyk. Pomocí speciálních značek – *tagů* označuje části dokumentu, stránky. To znamená, že pomocí *HTML* určujeme, jaký **význam** bude mít ta která část *HTML* stránky. Tedy pomocí *HTML* řekneme, že tenhle kus textu bude nadpis, jiný kus textu odstavec a jiný tabulka. Interpretaci jazyka zajišťuje klient a na jeho vlastnostech záleží, jak bude výsledná stránka vypadat. Jestliže chcete udělat i ty úplně nejjednodušší webové stránky, *HTML (XHTML)* potřebujete. V roce 1999 byla zveřejněna poslední vývojová verze *HTML - HTML 4.0. HTML* již není nadále rozvíjeno.

2.3.3 XHTML (eXtensible HypertextMarkup Language)

vzniknul přepsáním *HTML* podle pravidel *XML*. *XML* je jazyk, který se prosazuje jako univerzální nástroj pro výměnu strukturovaných informací.

První verze *XHTML 1.0* nepřinesla žádné nové prvky v porovnání s *HTML 4*. Jednalo se jen o přepis existujících vlastností tak, aby vyhovovaly *XML*. Tímto však usnadňuje analýzu a umožňuje používat standardní *XML* nástroje pro zpracování *XHTML* textů.

Rozdíl mezi *HTML* a *XHTML* je zatím poměrně malý a je vhodné používat *XHTML* jako perspektivnější variantu.

XHTML je nástupce *HTML*. Rozlišujeme 3 druhy *XHTML*:

- ⇒ **XHTML 1.0 Strict (přísné)** – čistě strukturální značkování, **neobsahuje** žádné značky spojené s formátováním vzhledu
- ⇒ **XHTML 1.0 Transitional (přechodné)** – **povoluje** atributy pro formátování textu a odkazů v elementu *body* a některé další atributy
- ⇒ **XHTML 1.0 Frameset (s podporou rámců)** – povoluje použití **rámců** pro rozdělení okna prohlížeče na dvě nebo více částí. Jeho užití je však považováno za nevhodné. Pro rozdělení stránky na více částí existují efektivnější metody (např. tzv. *bez-tabulkový layout v CSS*).

2.3.4 CSS (Cascading Style Sheets)

CSS je technologie sloužící jako formátovací jazyk pro (*X*)*HTML*. To znamená, že určuje, jak bude která část web stránky vypadat. Pomocí *CSS* stylu tedy určujeme vzhled stránky. *CSS* říká, že nadpis (o kterém *HTML* určilo, že je nadpis) bude modrý, že odstavec se bude zarovnávat vlevo a že tabulka bude na zeleném pozadí. Ale už neříká, která část stránky bude tím nadpisem, odstavcem či tabulkou. *CSS* tedy určuje jen **vzhled** stránky. Tedy vše, co se týká barvy, písma, velikosti prvků a podobně. Problémem při nasazení je mnohdy rozdílná implementace v jednotlivých klientech.

Takže si shrneme, co určuje u stránky *HTML* a co *CSS*.

HTML Řekne toto bude **nadpis**

CSS Řekne tento nadpis bude **modrý**

CSS styly jsou k tvorbě stránek potřeba. Je pravda, že ne úplně nezbytně. Sice to jde i bez nich, ale takový způsob je považován za velký přežitek. Dělat stránky bez CSS je prostě špatné. Proto se už od začátku doporučuje učit CSS současně s *XHTML*.

2.3.5 PHP (Hypertext Preprocessor)

PHP je skriptovací jazyk. Dělat se pomocí něj různé skripty vytvářející webové **aplikace**. Pomocí PHP můžeme vytvořit knihu hostů, počítadlo návštěv, anketu, diskusní fórum, redakční systém, systém na blog, chat, atd. Pokud chcete používat *PHP*, musíte ovládat *HTML*. Proto nemá cenu se s PHP zabývat, pokud nerozumíte *HTML*.

2.3.6 JavaScript

JavaScript je také skriptovací jazyk. Hlavní rozdíl mezi *JavaScriptem* a *PHP* je v tom, že *PHP* funguje na serveru a *JavaScript* na klientovi. To znamená, že to, co dělá *JavaScript* se vždy odehrává na vašem počítači a nejsou odesílána žádná data na server. Pomocí *JavaScriptu* tedy nevytvoříte nic, co by muselo proběhnout Internetem. Výhodou *JavaScriptu* však je, že fungují, i když nejste k Internetu zrovna připojeni.

V *JavaScriptu* tedy nelze vytvořit chat nebo knihu hostů, protože by bylo potřeba někam posílat data. *JavaScript* slouží k tvorbě různých interaktivních menu, grafických prvků, pro ovládání prohlížeče (otvírání nových oken, výpisy do stavové lišty,...).

O rozvoj webových standardů se převážně stará *World Wide Web Konsorcium* (www.w3c.org). Je to nezisková organizace, která sdružuje akademické i komerční organizace zabývající se Internetem a souvisejícími technologiemi.

Cvičení:

Vyhledejte v českém případně světovém Internetu zajímavé stránky a servery, věnující se vytváření WWW stránek. Zkuste vyhodnotit, která se vám líbí nejvíce.

Cvičení:

Prohlédněte si zdrojový kód jednoduché stránky. (V IE nabídka Zobrazit – Zdrojový kód stránky). Porovnejte s výslednou podobou a pokuste se odhadnout, k čemu slouží jednotlivé konstrukce.

Cvičení:

Pokuste se vyhledat v Internetu programy sloužící k tvorbě WWW stránek. Zkombinujte své výsledky s výsledky ostatních studentů.

Cvičení:

Pokuste se vyjmenovat výhody a nevýhody jednotlivých technik pro tvorbu webu. Která technika se vám jeví jako nejlepší?

3. ÚVOD DO XHTML

⇒ Cíl kapitoly:

Vysvětlit syntaxi XHTML jazyka a jeho vlastnosti: elementy, atributy, pravidla XHTML, kostra stránky.

⇒ Předpokládaná doba studia:

4 vyučovací hodiny

XHTML lze svým způsobem chápat jako formát pro úpravu dokumentů. Od běžně používaných formátů (jako je např. *.doc* editor Microsoft Word) se liší v následujících vlastnostech:

- ⇒ je plně textový – text i příkazy pro jeho úpravu jsou v textové formě
- ⇒ je otevřený – formát je zdokumentován a jeho specifikace je volně dostupná
- ⇒ neobsahuje vše – WWW stránka je typicky složena z několika souborů (základní *XHTML* kód, jednotlivé obrázky, případné definice stylu a další)

V *HTML* se používají speciální značky – **tagy**. Tagy jsou tvořeny znaky „<” a „>” mezi nimiž je název tagu (takto: <tag>). Vše ostatní, co není mezi těmito znaky, se zobrazuje jako výsledný text na stránce. Tagy pak určují, jaký má text význam (např. jestli se jedná o nadpis, tabulku, či hypertextový odkaz).

V *XHTML* jsou všechny tagy párové, to znamená, že ke každému počátečnímu tagu musí existovat tag ukončovací, ten se liší od počátečního tím, že před názvem tagu obsahuje lomítko („/”). Celé to pak vypadá takto:

```
<tag>Text zobrazovaný na stránce.</tag>
```

Všechny tagy píšeme malým písmenem, *XHTML* je *case-sensitive* (rozlišuje velká a malá písmena), tzn. že <TAG> vlastně vůbec nezná.

3.1 ELEMENTY

Elementem nazýváme celou sekvenci počínaje počátečním tagem a konče tagem ukončovacím. Existují tři základní druhy elementů, *blokové*, *inline* a *nahrazované*. Podle významu, který textu přiřazují, je můžeme rozdělit ještě na elementy pro strukturování dokumentu, textové elementy, elementy pro tvorbu odkazů, elementy pro tvorbu tabulek, elementy pro tvorbu seznamů a podobně.

3.1.1 Blokované elementy

Jsou to elementy, které tvoří nějaký blok. Zjednodušeně to znamená, že po takovém elementu je text dokumentu zalomen, odřádkován. Blokovými elementy jsou například `h1` pro nadpis nebo `p` pro odstavec.

3.1.2 Inline elementy

Inline elementy jsou ty, které se nachází uvnitř textu, nedochází po nich k zalomení. Obvykle plní funkci zvýraznění nějaké části textu. Je to například `a` pro hypertextový odkaz.

3.1.3 Nahrazované elementy

Ty jsou nahrazeny nějakým obsahem, pro začlenění dokumentu jsou důležité jejich rozměry. Například `img` pro obrázek.

3.1.4 Atributy

Elementy mohou mít své atributy. Ty jsou přiřazením nějaké vlastnosti danému elementu. Atributy se píšou do počátečního tagu, může jich být více (oddělují se mezerou), nemusí být žádný. Každý atribut má svou hodnotu. Hodnota atributu musí být v *XHTML* zapsána v uvozovkách.

```
<element atribut="hodnota _atributu">Text zobrazovaný na stránce.</element>
```

Pozn.: Hodnotu atributu můžeme zapsat i do apostrofů.

3.1.5 Entity

Jelikož se znaky „<“ a „>“ používají k vymezení značek, stávají se v *XHTML* speciálními. Chcete-li například zapsat: $x > y$, nelze to udělat přímo, protože by „<“ bylo interpretováno jako zahájení značky. Pro zápis speciálních znaků slouží speciální konstrukce, tzv. entity. Zapisují se pomocí ampersandu (&), názvu příslušné HTML entity a středníku (;). Tedy např.: `&jméno;`

znak	entita	ascii	popis česky	popis anglicky
"	quot	34	rovné uvozovky (palce)	quotation mark (APL quote)
&	amp	38	znak <i>and</i>	ampersand
<	lt	60	znak <i>menší než</i>	less-than sign
>	gt	62	znak <i>větší než</i>	greater-than sign
nbsp	nbsp	160	nedělitelná mezera	non-breaking space
©	copy	169	copyright	copyright sign
°	deg	176	stupeň	degree sign
±	plusmn	177	znak <i>plus minus</i>	plus-or-minus sign
¶	para	182	znak odstavce	pilcrow sign (paragraph sign)
×	times	215	krát	multiplication sign
–	ndash	8211	pomlčka šířky <i>n</i>	en dash
—	mdash	8212	pomlčka šířky <i>m</i>	em dash
”	rdquo	8221	pravé horní uvozovky	right double quotation mark
„	bdquo	8222	levé dolní uvozovky	double low-9 quotation mark
/	frasl	8260	šikmé lomítko	fraction slash
€	euro	8364	euro	euro sign

Pozn.: Při použití *WYSIWYG* editorů se uživatel nemusí o používání symbolů starat. Zapiše-li některý ze speciálních znaků, editor se sám postará o vložení odpovídajícího symbolu (entity).

3.2 PRAVIDLA XHTML

⇒ Před samotným dokumentem se nachází deklarace XML.

```
<?xml version="1.0" encoding="iso-8859-2"?>
```

Pozn.: Kódování lze použít samozřejmě dle libosti.

⇒ Povinná je deklarace typu dokumentu (*DTD*)

XHTML 1.0 Strict

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

XHTML 1.0 Transitional

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

XHTML 1.0 Frameset

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

⇒ Kořenový element *html* obsahuje atribut *xmlns*, který určuje jmenný prostor dokumentu (*namespace*) a jazyk, který je v dokumentu použit.

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

⇒ Element *html* vždy obsahuje **dva elementy**, *head* (hlavičku) a *body* (tělo dokumentu). Hlavička musí obsahovat element *title* a měla by obsahovat i metatag pro kódování (kvůli starším prohlížečům).

⇒ Všechny tagy i atributy musí být **malými písmeny**, *XHTML* je case-sensitive.

⇒ Všechny hodnoty atributů musí být v *XHTML* v **uvozovkách**.

⇒ Všechny *XHTML* **tagy** musí být **párové**. Při použití prázdného tagu se musí tag ukončit lomítkem, např. ``

⇒ Tagy se nesmí nikdy **křížit**.

Špatně:

```
<tag1>Text <tag2>zobrazovaný</tag1> na stránce.</tag2>
```

Dobře:

```
<tag1>Text <tag2>zobrazovaný</tag2> na stránce.</tag1>
```

⇒ Striktní *XHTML* neobsahuje **žádné** atributy sloužící k **formátování**. Oproti *HTML* jsou z *XHTML* vypuštěny formátovací tagy (jako např. `font`, `b`, `i`).

3.3 KOSTRA STRÁNKY

V předchozí kapitole – pravidlech XHTML už jsme si vysvětlili, že dokument začíná deklarací XML, následuje *DTD-Doctype*, potom tag `HTML` obsahující hlavičku a tělo dokumentu. Můžeme tedy sestavit celý základ stránky.

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

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

<head>
  <meta http-equiv="Content-Type" content="text/html;
charset=windows-1250" />
  <title>Titulek stránky</title>
</head>

<body>
  <p>Odstavec textu.</p>
</body>
</html>
```

Tak takto by měla vypadat kostra stránky. Soubor uložíme s příponou **htm** nebo **html**. Teď něco k použitým tagům.

- ⇒ **html** označuje, že se bude jednat o HTML dokument, obsahuje `head` a `body`
- ⇒ **head** hlavička, obsahuje `title` (titulek stránky), metatagy (např. kódování), odkazy na externí dokumenty definice stylu, skripty, apod.
- ⇒ **title** titulek stránky, to co se objeví jako název okna
- ⇒ **body** tělo dokumentu, tady se bude nacházet celý obsah stránky
- ⇒ **p** označení odstavce

Cvičení:
Pokuste se najít HTML a XHTML stránku, srovnajte kód stránky vytvořené v HTML a XHTML.

4. ODKAZY - URL

⇒ Cíl kapitoly:

Vysvětlit práci s odkazy: použití odkazů. Rozdíl mezi absolutním odkazem a relativním odkazem.

⇒ Předpokládaná doba studia:

3 vyučovací hodiny

URL - Uniform Resource Locator (URL) vznikl společně s WWW jako univerzální formát adresy, která umožňuje jednoznačně identifikovat valnou většinu informací v Internetu. Není omezen na WWW, pomocí URL lze popsat také adresy pro elektronickou poštu, data dostupná pomocí FTP aj.

⇒ Nejčastější tvar URL je: protokol://server/cesta

⇒ Například: `http://www.seznam.cz/instituce/stredni_skoly/`

V adresách se také rozlišují malá písmena od velkých. Je vhodnou konvencí zvyknout si používat v názvech souborů a adresářů zásadně malá písmena a toto pravidlo důsledně dodržovat. Je také třeba důrazně nedoporučit používání znaků s diakritikou, mezer a dalších potenciálně problémových znaků v názvech souborů a adresářů.

4.1 POUŽITÍ ODKAZŮ NA STRÁNCE

Odkazy jsou základem hypertextovosti Internetu. Bez odkazů by se uživatel nikam nedostal. V textu jsou odkazy vizuálně odlišeny, standardně je to barevným odlišením a podtržením. Při přejetí myši přes odkaz se mění kurzor (obvykle ze šipky na tlapičku), kliknutím je odkaz aktivován a dochází k přesměrování.

Za obecný nešvar je považováno nadepisovat odkazy „klikněte zde“. Tento text uživateli neříká vůbec nic. Důležité je zvýraznit co se čtenář dozví, když klikne.

Špatně:

Vlastnosti našeho produktu najdete **zde**. Zajímají-li Vás možnosti jeho použití, **klikněte zde**. Pro kompletní ceník **klikněte zde**.

Dobře:

Skvělé vlastnosti našeho produktu jej předurčují pro řadu možných **použití**. Pokud vás zaujal a přemýšlíte o koupi, podívejte se na **kompletní ceník**.

Cvičení:

Projděte si několik (cca 5) svých oblíbených serverů. Na každém z nich si prohlédněte několik běžných stránek a sledujte jejich URL, která klient zobrazuje ve stavovém řádku.

4.2 ODKAZ NA JINOU STRÁNKU

Pro odkaz se používá element `a`. Jeho atributem je `href`. Ten udává, kde je umístěn cílový dokument (tedy na kterou stránku odkaz směřuje). Samotné umístění pak zadáváme absolutní nebo relativní cestou.

4.2.1 Absolutní odkaz

Absolutní odkaz se používá pro odkázání na dokument umístěný na jiném serveru, např. jako odkaz na jinou stránku. Odkaz na jinou stránku musí vždy obsahovat `http://`.

```
<a href="http://www.nekde.cz">Někde.cz</a>
```

Hodnotou atributu `href` je tedy adresa cílového dokumentu (v tomto případě `http://www.nekde.cz`). Zobrazovaný a klikatelný text bude Někde.cz.

4.2.2 Relativní odkaz

Relativní odkaz použijeme, pokud odkazujeme v rámci stránky. Při použití relativního odkazu prohlížeč sám doplňuje URL stránky, z které je odkazováno. Pokud tedy odkazujeme na dokument nacházející se ve stejném adresáři, stačí napsat jen jméno souboru.

```
<a href="uvod.htm">úvod</a>
```

Pokud se dokument nachází v jiném adresáři, musíme do URL zahrnout celou cestu od místa, kde se cesty k odkazovanému a odkazujícímu dokumentu začaly odlišovat.

```
<a href="fotky/zvirata/zirafy.htm">fotky žiraf</a>
```

Pokud se budeme chtít dostat v adresářové struktuře o úroveň výš, použijeme k tomu dvou teček, tj. `..`. Například takto:

```
<a href="../rostliny/fialky.htm">fotky fialek</a>
```

Cvičení:

Prohlédněte si zdrojové kódy několika stránek z Internetu. Studujte, jak jejich autoři používají absolutní a relativní odkazy.

4.3 ODKAZ NA KONKRÉTNÍ MÍSTO DOKUMENTU

Někdy je dobré odkazovat i na části stránky, to se hodí jednak, pokud je stránka delší — pak v úvodu použijeme odkazy na její jednotlivé části a dále pokud z jiné stránky potřebujeme použít odkaz na přesně určenou část stránky (aby uživatel nemusel hledat). Pro označení místa, na které chceme odkázat, použijeme atribut `id`, v odkazu potom použijeme jako vždy atribut `href`, který ale bude tentokrát začínat znakem `#`. Často se toto používá také při odkazování na začátek stránky.

```
<a href="#zacatek">zpět na začátek</a>
```

Stejným způsobem pak odkazujeme i z jiné stránky, jen musíme přidat klasickým způsobem odkaz na celý dokument a za něj odkaz na konkrétní místo.

```
<a href="svetadily.htm#evropa">Evropa</a>
```

Takový odkaz můžeme dát někam nakonec stránky, po kliknutí prohlížeč sám „odroluje – odskočí” tak, že se element označený příslušným názvem dostane do horní části okna prohlížeče.

Část dokumentu, na kterou takto odkazujeme, označíme v HTML stránce takto (v našem případě budeme odkazovat na hlavní nadpis stránky):

```
<h1 id="zacatek">Hlavní nadpis stránky</h1>
```

4.4 OTEVŘENÍ ODKAZU V NOVÉM OKNĚ

V novém okně se obvykle otvírají odkazy, které směřují na „cizí stránky”. Otvírat v novém okně odkazy na vlastní stránky (tím myslím například otevírání položek menu v novém okně) je většinou nesmysl. Uživatel je snad natolik inteligentní, aby se sám rozhodl, jestli si otevře odkaz v novém nebo ve stejném okně. Jakmile určíme otevření v novém okně, uživatel již tuto možnost volby nemá, což není dobré.

4.4.1 Způsob otevření nového okna:

⇒ pomocí atributu `target`

otevřeme klasické okno se všemi panely a lištami, s výchozím nastavením velikosti (pozn.: Atribut `target` není povolen v XHTML 1.0 Strict, v Transitional ano)

```
<a href="http://www.okna.com" target="_blank">odkaz v novém okně</a>
```

⇒ pomocí *JavaScriptu* a funkce `window.open`

otevřeme nové okno s nastavitelnými vlastnostmi

```
window.open("http://www.okna.com","_blank", width=100);
```

4.5 TITULEK ODKAZU

Každý odkaz by měl mít svůj titulek. Tím je text, který uživateli přiblíží, kam vlastně odkaz směřuje. Titulek se zobrazí při najetí kursoru myši na odkaz. Titulky odkazů jsou důležité také např. pro čtecí zařízení, umožňují lepší orientaci na webu.

```
<a href="http://www.zamekkurim.cz" title="Úvodní stránka tohoto webu (zamekkurim.cz)">Domů</a>
```

Cvičení:

Najděte cca pět zajímavých stránek relevantních pro předmět Výpočetní technika. Vytvořte WWW stránku, která bude obsahovat těchto pět odkazů – ke každému odkazu uveďte jeho jméno (název stránky na kterou odkazuje) a stručný popis.

Cvičení:

Do některé ze stránek, kterou jste vytvořili v rámci kursu, přidejte pod text své jméno, jakožto jméno autora a udělejte z něj odkaz vedoucí na vaši e-mailovou adresu (atribut „mailto:“), aby vám čtenář snadno mohl poslat e-mail.

Cvičení:

Vytvořte seznam studentů třídy s jejich elektronickými adresami. Adresy by měly být odkazy se schématem „mailto:“, aby se pouhým kliknutím dal zaslat e-mail libovolnému studentu.

Cvičení:

Udělejte abecední seznam žáků třídy (můžete též použít výsledek vaší práce v některém z předchozích cvičení). Seznam rozdělte na části podle počátečních písmen jednotlivých jmen a na jeho začátek přidejte „rozsok“ podle počátečního písmene. Jedná se o seznam písmen, kde každé písmeno představuje odkaz, který uživatele přivede na první jméno, které začíná tímto písmenem.

5. NADPISY

⇒ Cíl kapitoly:
Vysvětlit proč nadpisy používat. Vysvětlit vytváření nadpisů.
⇒ Předpokládaná doba studia:
2 vyučovací hodiny

Nadpisům je třeba věnovat značnou pozornost.

- ⇒ usnadňují čtenáři orientaci na stránce
- ⇒ vyhledávací roboti je zvyhodňují – obsahuje-li stránka uživatelem hledané slovo v nadpisu, dostane vyšší skóre než stránka, která je obsahuje jen v běžném textu.
- ⇒ Stránka by měla mít **právě jeden** nadpis velikosti 1 (nadpis celé stránky).
- ⇒ V hierarchii velikostí nepřeskakujte. Základní části stránky by měly být nadepsány velikostí 2, jejich části velikostí 3 atd.
- ⇒ I prohlížeč, který nepodporuje (nebo nenačte) CSS styly, nadpisy zformátuje alespoň podle velikosti.

Existuje 6 úrovní nadpisů, označují se tagy h1, h2, h3, h4, h5 a h6, kde 1 je úroveň nejvyšší. Úrovně se liší velikostí písma, všechny nadpisy jsou implicitně zarovnávané vlevo.

```
<h1>Nadpis 1. úrovně</h1>
<h2>Nadpis 2. úrovně</h2>
<h3>Nadpis 3. úrovně</h3>
```

A takový je výsledek:

Nadpis 1. úrovně

Nadpis 2. úrovně

Nadpis 3. úrovně

Cvičení:

Vytvořte stránku, která bude obsahovat nadpis velikost 1, za ním krátký odstavec textu, nadpis velikosti 2, opět odstavec textu (klidně stejný), nadpis velikosti 3, odstavec atd. až po nadpis velikosti 6, následovaný odstavcem textu. Porovnejte, jakým písmem klient zobrazí jednotlivé velikosti nadpisů a jaké mezery vynechá před nimi a za nimi. Zkuste v klientovi změnit základní velikost písma a pozorujte, jak se změna na stránce projeví.

6. SEZNAMY

⇒ Cíl kapitoly:
Vysvětlit práci se seznamy
⇒ Předpokládaná doba studia:
1 vyučovací hodina

6.1 NEUSPOŘÁDANÝ SEZNAM

Neuspořádaný neboli nečíslovaný seznam se značí tagem `ul` (*unordered list*). Položka seznamu je `li` (*list item*). Položka může obsahovat i více odstavců. Před každou položkou se standardně vytváří odrážka.

```
<ul>
  <li>žirafa</li>
  <li>slon</li>
  <li>velbloud</li>
</ul>
```

Výsledek:

- žirafa
- slon
- velbloud

6.2 USPOŘÁDANÝ SEZNAM

Uspořádaný neboli číslovaný seznam se značí tagem `ol` (*ordered list*). Položka seznamu je opět `li`. Před položku se automaticky vypisuje její pořadové číslo.

```
<ol>
  <li>žirafa</li>
  <li>slon</li>
  <li>velbloud</li>
</ol>
```

Výsledek:

1. žirafa
2. slon
3. velbloud

Cvičení:
Vytvořte stránku s pořadovým seznamem žáků ve vaší třídě.

Cvičení:
Vytvořte stránku s jednoduchým popisem pracovního postupu – např. uvaření vajíčka natvrdo, uvaření konvice čaje, pověšení obrazu apod. Postup запиšte ve formě krátkých bodů opatřených pořadovými čísly.

7. TABULKY

⇒ Cíl kapitoly:

Vysvětlit způsoby vytváření tabulek, použití a práci s tabulkami

⇒ Předpokládaná doba studia:

3 vyučovací hodiny

Původně měly tabulky v *XHTML* umožnit zařazení dat tabulkovitého charakteru na WWW stránky. Postupně se jich však začalo velmi intenzívně využívat k formátování celých stránek. Byla to vlastně jediná možnost, jak přímo v *HTML* rozmístit materiál netriviálním způsobem po stránce.

V současnosti však podpora *CSS* dozrála natolik, že lze tyto metody opustit a formátovat stránky bez použití tabulek (tzv. *bez-tabulkový design*). Formátování pomocí tabulek se považuje za přežitek!

Tabulka se vytváří pomocí elementu `table`. Pro řádek tabulky slouží tag `tr` (*table row*), pro buňky tabulky tagy `th` (*table head*) pro buňky v záhlaví tabulky a `td` (*table data*). Buňky se vkládají uvnitř řádku tabulky, kolik buněk, tolik bude mít tabulka sloupců. Buňky záhlaví jsou standardně formátovány tučným písmem, zarovnány na střed.

V tabulce se používá atribut `summary`, který shrnuje obsah tabulky. Slouží pro lepší přístupnost webu.

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

Výsledek:

1	2
aaa	bbb
ccc	ddd

7.1 NADPIS TABULKY

Pro nadpis tabulky se používá element `caption`. Ten by měl být uveden uvnitř tagu `table` ještě před prvním řádkem.

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

Nadpis
tabulky

1	2
aaa	bbb
ccc	ddd

Výsledek:

7.2 SLUČOVÁNÍ BUNĚK

Atribut `colspan` sloučí dohromady několik buněk v jednom řádku. Atribut `rowspan` sloučí buňky ve sloupci. Jako hodnotu těchto atributů zapisujeme počet buněk, které chceme takto sloučit.

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

Nadpis
tabulky

1	
aaa	bbb
	ccc

Výsledek:

Cvičení:

Spočítejte počet dívek a chlapců ve třídách prvního ročníku. Výsledky uspořádejte do tabulky. Každé třídě bude odpovídat jeden řádek. Jednotlivé sloupce budou obsahovat: jméno třídy, počet dívek, počet chlapců a celkový počet studentů ve třídě.

Cvičení:

Udělejte tabulku malé násobilky. Záhloví řádků i sloupců budou tvořit čísla od jedné do deseti. V průsečíku sloupce a řádku bude vždy hodnota odpovídající součinu těchto dvou čísel.

Cvičení:

Najděte aktuální žebříčky nejlepších tenistů a tenistek nebo třeba golfistů a golfistek (např. www.idnes.cz - sportovní sekce). Vytvořte WWW stránku, která bude obsahovat dvě tabulky – nejlepších pět mužů a žen. Uveďte vždy pořadí, jméno a počet bodů.

Cvičení:

Vyberte z novin dva krátké články a umístěte je na stránku vedle sebe jako dvousloupcový text. V každém sloupci bude jeden článek. Poté zkuste rozvržení, které se více podobá novinové sazbě: Články budou pod sebou, jejich nadpisy budou na celou šířku stránky, ale text každého článku bude rozdělen do dvou sloupců.

Cvičení:

Vytvořte stránku se zasedacím pořádkem vaší třídy. Jména žáků rozmístěte pomocí tabulky tak, jak sedí ve třídě. Doporučuji oddělit jednotlivé řady lavic prázdným sloupcem, aby se zvýraznilo jejich uspořádání. Lavice můžete zvýraznit atributem `bgcolor`.

Cvičení:

Vytvořte WWW stránku s takovouto křížovkou:

Školní budova

Iva

Automobilový závod u nás

Krůpěj

Květina

				A
				A
				A
				A
				A

8. OBRÁZKY

⇒ Cíl kapitoly:

Práce s grafikou v XHTML, obecná pravidla při vkládání grafiky do WWW stránky. Grafika pro WWW stránky, jaký zvolit grafický formát? Vložení obrázku v XHTML.

⇒ Předpokládaná doba studia:

4 vyučovací hodiny

Skoro vždy platí „méně je více“. Pokud se autor krotí a používá minimum zdobných prvků, pravděpodobně vytvoří stránku, která bude sice konzervativní a nenadchne, ale na druhé straně neodpuzeje.

Zdaleka ne každý má dostatečně vyvinuté estetické cítění a řada laických autorů prostě nedovede posoudit úroveň svých výsledků. Stránky pak často vyrábějí stejným způsobem, jako když pejsek s kočičkou vařili dort. Výsledek bývá podobně chutný.

Především je vhodné vyhnout se různým grafickým oddělovačům, animovaným obrázkům, obrázkovému pozadí stránky a ikonám posbíraným různě v Internetu. Jejich používání je něco jako stavět si na zahrádce trpaslíky.

Je velmi důležité, pokud to myslíte s webdesignem vážně, naučit se s grafikou správně zacházet.

8.1.1 Grafika pro WWW stránky

Obrázky a další grafické prvky stránek hrají ve WWW velmi významnou roli. Na jedné straně při vhodném použití výrazně zvyšují atraktivitu stránek. Na straně druhé citelně zvětšují objem přenášených dat a zpomalují odezvy. Příprava grafiky pro Web představuje do značné míry hledání vhodného kompromisu mezi kvalitou obrázku a velikostí dat.

Obrázky, které nemají jinou funkci než „zlepšení designu“ stránky, by měly být vkládány **pouze pomocí CSS** stylu a to jako obrázek na pozadí.

8.1.2 Jaký zvolit grafický formát?

Použitý grafický formát má zásadní vliv na kvalitu a velikost obrázku. Každý případ by měl autor stránky posuzovat individuálně a experimentovat s parametry při ukládání, nicméně existují obecně platná pravidla, která ve většině případů nezklamou.

⇒ Podle nich je na obrázky charakteru fotografie či malby (velký počet barev a plynulé přechody mezi nimi) nejvhodnější *JPEG*. Jeho kompresní algoritmus je pro takovéto případy optimalizován a přináší jednoznačně nejlepší poměr mezi velikostí a kvalitou.

⇒ Pro obrázky charakteru kresby či nápisy (jsou charakterizovány menším počtem barev a ostrými hranami) bývají naopak lepší formáty s omezeným počtem barev – *PNG* nebo *GIF*.

8.1.3 Grafický editor

Chcete-li vytvořit obrázek zcela „na zelené louce“, lze použít grafický editor. Tato cesta se používá například pro přípravu grafického menu či orientačních prvků stránky, které se prostě nedají nafotit.

Další oblastí využití je kreslení různých schémat či vytváření zcela volné grafiky (kresby, malby). Grafický editor představuje povinnou výbavu autora stránek, protože určité úpravy grafických souborů je třeba dělat vždy.

8.2 VLOŽENÍ OBRÁZKU V XHTML

Pro vložení obrázku slouží nepárový element `img`. **Povinnými** atributy jsou `src`, jehož hodnotou je název obrázku, případně i s cestou a `alt` obsahující alternativní text, který se zobrazí, pokud je obrázek prohlížeči nedostupný, buď proto, že se na zadané adrese nenachází nebo proto, že prohlížeč obrázku nezobrazuje. Alternativní text by měl tedy nějakým způsobem přiblížit co je na obrázku těm, kteří jej nevidí.

```

```

Pro porovnání ještě uvedu, jak by vypadal tento zápis v klasickém *HTML*.

```
<img src= obrazky/obrazek.jpg width=100 height=91 alt=Obálka HTML příručky >
```

8.3 VELIKOST OBRÁZKU

Velikost obrázku není povinné zadávat, přesto je **velkou chybou** toto opomenout. Pokud totiž velikost obrázku nezadáte, prohlížeč si na jeho zobrazení nevyhraní dostatečný prostor a jelikož okolní text se samozřejmě načte dřív než obrázek (protože je co se týče velikosti dat menší), způsobí to jakési „poskakování“ stránky ve chvíli, kdy se začínají načítat obrázky, které potřebují více prostoru, než mají a okolní text jim vlastně musí uhybat.

Stejně tak je vhodné zadávat obrázku jeho skutečnou velikost, nezmenšovat ani nezvětšovat pomocí prohlížeče, ale pomocí grafického editoru. Pokud totiž obrázek o velikosti např. 100×100 pixelů zmenšíme v prohlížeči na 50×50, docílíme tak sice požadované velikosti, ale uživatel bude zbytečně stahovat více dat, protože bude stahovat samozřejmě obrázek v původní velikosti.

Pro určení velikosti obrázku se používají atributy *width* (šířka) a *height* (výška) nebo stejnojmenné vlastnosti v CSS.

Cvičení:

Najděte v Internetu použitelný (tedy dostatečně kvalitní a dostatečně velký – alespoň 200 x 200 pixelů) obrázek jablka, stolu a míče. Můžete použít například vyhledávací servery www.ditto.com či images.google.com. Nalezené obrázky vložte do stránky.

Cvičení:

Vyzkoušejte na stránku vložit grafický nadpis. Vyberte některou ze svých předchozích stránek a pomocí grafického editoru vytvořte nadpis (bezpatkové písmo větší velikosti) odpovídající textu nadpisu této stránky. Pohrajte si s použitým grafickým formátem a nastavením parametrů. Obsah značky <h1> pak nahraďte tímto obrázkem (zachovejte jeho uzavření do značky <h1> kvůli vertikálním mezerám). Porovnejte původní verzi s výslednou. Jaká je nyní celková velikost stránky? Je nová verze hezčí? Pokud ano, stojí to za nárůst velikosti?

Cvičení:

Napište krátký text (cca půl stránky) o škole. Doplňte jej jednou až dvěma ilustračními fotografiemi (současná či historická podoba školy, fotografie interiéru apod.).

Cvičení:

Vytvořte reklamní „plakát“ na nějaký výrobek či službu. Měl by obsahovat dvě až tři fotografie, výčet vlastností, informace o ceně atd. Fotografie buď získejte sami, nebo z Internetu.

9. META TAGY

⇒ Cíl kapitoly:

Práce s Meta tagy: kódování češtiny, popis stránky, klíčová slova stránky, jazyk stránky.

⇒ Předpokládaná doba studia:

1 vyučovací hodina

Meta tagy obsahují *meta data* – informace o stránce, nachází se v hlavičce (`head`) HTML dokumentu.

9.1 META CHARSET – KÓDOVÁNÍ ČEŠTINY

Nejdůležitějším a dá se říct, že jediným opravdu nezbytným meta tagem je meta tag pro kódování češtiny. Pokud ho nepoužijete, text stránek, resp. české znaky se budou zobrazovat jako všelijaké paznaky, čtverečky a podobně.

Kódování češtiny je nutné uvést ještě **před tagem `title`**.

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

Část `charset=windows-1250` označuje použitou znakovou sadu. Pro češtinu se často používají tyto znakové sady:

`windows-1250`

`iso-8859-2` – doporučuji

`utf-8`

9.2 META DESCRIPTION – POPIS STRÁNKY

Pro popis obsahu stránky slouží meta tag `description`. Jeho obsah se může objevit jako popisek odkazu ve vyhledávačích, takže je určitě dobré dbát na to, aby neobsahoval nesmyslné údaje. Slova obsažená v tomto meta tagu mívají také pro vyhledávače vyšší váhu.

```
<meta name="description" content="Ne těchto stránkách najdete všechny důležité informace o mně, na které jste se báli zeptat" />
```

9.3 META KEYWORDS – KLÍČOVÁ SLOVA STRÁNKY

Meta tag `keywords` obsahuje seznam klíčových slov stránky. To jsou nejdůležitější slova, kterými se stránka zabývá.

```
<meta name="keywords" content="osobní stránky, blog, fotky" />
```

Meta tag `keywords` nemá příliš velký význam. Určen je pro vyhledávače, ale většina vyhledávačů jeho obsah ignoruje zcela, ostatní alespoň do té míry, že nepřisuzují hodnotu slovům obsaženým pouze v `keywords` a nikde jinde ve stránce.

9.4 META AUTHOR – AUTOR STRÁNKY

Meta tag `author` obsahuje informace o autorovi stránky.

```
<meta name="author" content="Antonín Ňouma" />
```

9.5 META LANGUAGE – JAZYK STRÁNKY

Meta tag `language` obsahuje jazyk použitý ve stránce.

```
<meta name="Content-language" content="cs" />
```

9.6 DALŠÍ META TAGY

Jinak je metatagů ještě hodně, jejich využití je však mizivé.

Cvičení:

Vytvořte XHTML soubor a zapište do něj větu obsahující znaky s háčky a čárkami. Obvyklá testovací věta je „Žluťoučký kůň úpěl šílené ódy“, která obsahuje spoustu různých znaků s diakritickými znaménky. Zkontrolujte (pokud možno klienty ze dvou operačních systémů), zda se zobrazuje správně. Dávejte pozor především na písmena „š“ a „ž“, ve kterých se odlišuje kód Microsoftu od standardu ISO 8859–2 používaného v Unixu.

A to je v podstatě to nejdůležitější, co je k vytvoření HTML stránky potřeba, bez těch pár dalších věcí se dá bez problému obejít.

10. ÚVOD DO CSS

⇒ Cíl kapitoly:

Proč vznikly a jak používat CSS styly. Vysvětlit k čemu je to dobré, dědičnost, připojení stylu k dokumentu, třídy a identifikátory, jednotky, barvy.

⇒ Předpokládaná doba studia:

6 vyučovacích hodin

10.1 HISTORIE

CSS (Cascading Style Sheets) neboli kaskádové styly vznikly jako souhrn metod pro úpravu vzhledu stránek. První návrh normy byl zveřejněn v roce 1994, v roce 1996 byla pak vydána specifikace *CSS 1*, v roce 1998 *CSS 2*, v roce 2006 *CSS 3*.

10.2 K ČEMU JE TO DOBRÉ?

CSS se využívá k formátování obsahu *HTML*, *XHTML* a *XML* dokumentů. Ve srovnání s formátováním pomocí atributů v *HTML* formátovací schopnosti rozšiřuje. Styly umožňují přesně určit, jak bude který element vypadat. Na rozdíl od atributů stylem můžeme definovat jednotný vzhled elementu pro celý dokument (např. že všechny nadpisy úrovně 1 budou červené) a to jediným zápisem pro příslušný element (nikoli v každém tagu příslušného elementu). Stejně tak můžeme pomocí stylu určit odlišné formátování pro třeba jen jediný výskyt určitého elementu. Tím se jednak zbavíme velkého množství kódu, jednak se tento kód stane mnohem přehlednější. Navíc pokud se jednou rozhodneme změnit například barvu písma všech odstavců, bude to pro nás otázka několika málo vteřin, měnit každý atribut u každého elementu v *HTML* by byla katastrofa. Jeden styl můžeme snadno použít pro libovolné množství stránek.

10.3 ZAČÍNÁME

Styl se skládá z pravidel pro jednotlivé elementy, které mají být formátovány. Každé takové pravidlo má dvě části, **selektor** (název elementu, pro který má toto pravidlo platit) a **deklaraci** (co pro něj má platit). V deklaraci určujeme vlastnost a její hodnotu, deklarace je uzavřena do složených závorek. Celé to zapisujeme takto:

```
selektor {vlastnost: hodnota_vlastnosti}
```

A konkrétně:

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

Selektorem, tedy elementem, který formátujeme je zde `h1` (nadpis 1. úrovně). Deklarací je `{color: blue}`. Ta určuje, že vlastnost `color` bude mít hodnotu `blue`.

Celé dohromady to tedy znamená, že všechny nadpisy 1. úrovně v dokumentu budou mít modrou barvu.

Pokud budeme chtít určit elementu více než jednu vlastnost, jednotlivé vlastnosti od sebe oddělíme středníkem. Takto můžeme definovat libovolné množství vlastností.

```
selektor {vlastnost1: hodnota_vlastnosti1; vlastnost2:
hodnota_vlastnosti2;}
```

Pozn.: Samozřejmě je možný i zápis každé vlastnosti zvlášť, ale to je zbytečné.

Pokud budeme chtít určit dvěma elementům jejich společnou vlastnost, oddělíme od sebe jednotlivé selektory čárkou.

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

10.3.1 Dědičnost

Většina vlastností se dědí. To znamená, že element, který nemá vlastnost definovanou jí dědí po nadřazeném elementu. Týká se to především vlastností písma — barvy, velikosti, stylu atd. Pokud tedy chceme definovat nějakou vlastnost, kterou budou mít všechny elementy společnou (a později případně jen vytvářet výjimky), definujeme ji pro element body.

10.3.2 Komentáře

Pokud si chceme ke stylu psát nějaké poznámky pro lepší orientaci, zapíšeme ji do komentářů. Ty se v CSS tvoří pomocí `/*` a `*/`. Mezi hvězdičky pak můžeme umístit i několikařádkový komentář, ten se samozřejmě ve výsledném zobrazení neobjeví.

```
body {color: blue} /* tady si piši komentář, že mám všechny texty
modré */
```

10.3.3 Připojení stylu k dokumentu

Styl můžeme k dokumentu připojit několika způsoby, můžeme definovat přímo v dokumentu nebo v externím souboru, způsoby můžeme i kombinovat.

10.3.3.1 Externí soubor

Pokud chceme mít styl uložený v externím souboru (což je velmi výhodné při používání jednoho stylu pro více dokumentů), v nějakém textovém editoru uložíme námi definovaný styl do souboru s příponou `css`. Ten pak připojíme k dokumentu zápisem v hlavičce (tj. mezi tagy `<head>` a `</head>`) buď v tagu `link`

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

nebo v tagu `style`

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

Pokud je styl umístěn na jiném serveru, tak můžeme použít zápis:

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

Zápisem `@import "styl.css"`; můžeme také vložit jeden styl do druhého stylu.

10.3.3.2 Definování stylu uvnitř dokumentu

To můžeme provést opět v tagu `style`, kam tentokrát místo odkazu na externí styl umístíme přímo definici stylu.

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

Nebo můžeme definovat styl přímo nějakému elementu, což se hodí zvláště v případě, kdy máme definovaný jednotný styl, ale pro například jedno konkrétní slovo chceme použít jiné pravidlo. Potom použijeme v příslušném tagu atribut `style`.

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

10.3.4 Váha stylů

Pokud ve stylu definujeme pro stejný element stejnou vlastnost dvakrát, vyšší váhu má ta deklarace, která byla definovaná později (myšleno na pozdějším řádku) a ta se také provede. Pokud bychom chtěli některé deklaraci přiřadit větší důležitost, použijeme `!important`.

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

Pozn.: Starší (ale opravdu hodně staré) prohlížeče styly vůbec nepodporují.

10.4 TŘÍDY A IDENTIFIKÁTORY

Třídy a identifikátory v CSS slouží k tomu, abychom mohli různé elementy formátovat různě. Například odkazy na stránce. Každý z nás asi chce mít na stránce různé druhy odkazů, ne jen jeden. Jinak se obvykle dělají odkazy v menu, jinak odkazy v textu.

10.4.1 Třída `class` v CSS

Třídy vytvoříme snadno tak, že k elementu v HTML přidáme atribut `class`. Jeho hodnotou bude nějaký řetězec písmen, stejný pak budeme používat v CSS stylu jako selektor.

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

Tímto říkáme, že tento odstavec bude formátován podle pravidel třídy *poznamka*, na formátování ostatních odstavců se tato pravidla neprojeví. Teď musíme ještě ta pravidla určit v CSS stylu.

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

Teď tedy budeme mít všechny odstavce stejné, jen odstavec s třídou *poznamka* bude vypadat jinak (malým černým písmem). Resp. jinak budou vypadat všechny odstavce s třídou

poznamka, protože stejnou třídu můžeme použít u libovolného množství elementů. Dokonce i u různých elementů.

```
<p class="poznamka">Nějaký odstavec</p>
<li class="poznamka">Položka seznamu</li>
```

```
.poznamka {color: black} styl se aplikuje na všechny elementy s
třídou poznamka
li.poznamka {color: blue} styl se aplikuje jen na elementy li s
třídou poznamka
```

Chcete-li zařadit do třídy jen slovo či několik slov, použijte k tomuto účelu párovou značku ``. Například v následující ukázce je slovo „Praha“ zařazeno do třídy *mesto*:

```
<span class="mesto">Praha</span> je hlavním
městem <span class="zeme">České republiky</span>.
```

10.4.2 Pseudotřídy

Speciálním případem jsou takzvané pseudotřídy, které byly zavedeny pro odkazy (značka `<a>`) a umožňují předepsat vzhled pro různé stavy odkazu. Oddělují se dvojtečkou:

```
a:link { color: #0000ff } /*nenavštívený odkaz*/
a:visited { color: #000099 } /*navštívený odkaz*/
a:hover { color: #00ffff } /*odkaz pod kurzorem myši*/
a:active { color: #ff0000 } /*odeslaný odkaz*/
```

10.4.3 Identifikátor *id* v CSS

Identifikátor se od třídy liší tím, že se jedná vždy o **jednoznačný identifikátor**. To znamená, že ho na každé stránce můžeme použít jen jednou. Třidu lze použít libovolněkrát na každé stránce webu.

Identifikátory se tedy používají právě tam, kde je jisté, že se daný element objeví ve stránce jen jednou. Ideálně se tedy hodí pro věci jako je box celé stránky, menu, záhlaví nebo zápatí. Identifikátory se označují dvojkřížkem (`#`). Jinak je jejich zápis stejný jako zápis třídy.

```
<div id="menu"> ... </div>
```

a v CSS definici potom:

```
#menu {width:14em; background-color:black}
#menu a {color: white}
```

Pozn.: Párová značka `<div> ... </div>` se velmi často používá pokud se odlišné formátování má týkat rozsáhlejší části stránky.

10.5 PŘÍKLADY ZÁPISU (SELEKTORY)

druh selektoru	zápis	příklady	význam příkladu	poznámky
tag	jméno tagu	p { color: red }	Normální text <p>červený text</p>	
identifikátor	#identifikátor	#cervene { color: red }	Normální text ovlivněný text	
	tag#identifikátor	b#cerverne { color: red }	Jenom tučný <b id="cervene">tučný a červený	
třída	.třída	.cervena { color: red }	Normální text červený text	Vztahuje se na každý tag, který má uvedeno správné class.
	tag.třída	i.cervena { color: red }	<i>Jenom kurzíva</i> <i class="cervena">červená kurzíva</i>	Vztahuje se jen na konkrétní tag, který má uvedeno správné class.
hromadná deklarace	selektor, selektor	H1, H2, H3 { color: red }	<h1>Červený nadpis</h1> <h3>Také červený</h3>	Seznam libovolných platných selektorů (tagů, tříd apod.) oddělený čárkou.
kontextová deklarace	nadřazený Selektor selektor (oddělené mezerou)	li a { font-weight: bold }	normální text seznamu tučný odkaz	Příklad: ztučňuje odkazy (<a>) uvnitř seznamu ().
		i b { color: red }	<i>Červená tučná kurzíva</i> <i>Normální tučná kurzíva</i>	Záleží na pořadí.
pseudotřída	tag:pseudotřída	a:hover { color: red }	Zčervená při přejetí myši	Pseudotřídy a:link, a:visited, a:active, jsou pouze u odkazů. :hover funguje v Exploreru pouze jako a:hover.
		p:first-line { color: red }	<p>První řádka odstavce bude červená</p>	Funguje pouze v Mozille a v IE 5.5 Existuje i :first-letter.
přímá deklarace v HTML	<tag style="zápis stylu">	<p style="color: red">Červený odstavec</p>	Nezapisuje se do stylopisu.	

10.6 JEDNOTKY

10.6.1 Délkové jednotky

Délkové údaje se zapisují jako celá nebo desetinná čísla s nebo bez znaménka. Dvojice písmen identifikující jednotky musí být připojena ihned za číslem.

10.6.2 Relativní jednotky:

em	Výška aktuálního písma. Odpovídá šířce písmene 'M'
ex	Výška písmene 'x'
px	Pixely – 1 pixel odpovídá jednomu bodu obrazovky

10.6.3 Absolutní jednotky:

in	Palce. 1 in = 2,54 cm = 72 pt
cm	Centimetry
mm	Milimetry. 10 mm = 1 cm
pt	Body. 1 pt = 1/72 in = 1/12 pc
pc	Pica. 1 pc = 12 pt

10.6.4 Procenta

Procenta se zapisují jako celá nebo desetinná čísla s nebo bez znaménka, za kterými následuje znak "%". Hodnoty zadané jako procento se relativně vztahují k nějaké jiné hodnotě, od které se odvodí absolutní velikost. Pokud používáme procenta, musíme si vždy uvědomit, od které hodnoty se bude absolutní velikost odvíjet.

10.7 BARVY

Barva se v HTML a CSS zapisuje nejčastěji:

⇒ **Jménem** v angličtině – např.: ``. Existuje několik desítek pojmenovaných barev.

⇒ **Šestnáctkovým** RGB zápisem – např.: ``. (tvar #RRGGBB)

Tento způsob je nejjistější, nejpoužívanější a nejlepší.

Pro zvolení vhodných barevných kombinací doporučuji užitečnou pomůcku vytvořenou Petrem Pixy Staníčkem, naleznete ji na: <http://wellstyled.com/tools/>.

Úkol

Uložte si v Internet Exploreru několik stránek (Soubor – Uložit jako – Úplná webová stránka). Projděte si adresáře s uloženými daty. Vyhledejte soubory s příponou css a prohlédněte si jejich zdrojový kód. Porovnáním s výslednou podobou stránky se pokuste odhadnout, k čemu slouží jednotlivé konstrukce.

11. PŘEHLED VLASTNOSTÍ CSS

⇒ Cíl kapitoly:

Základní přehled toho, k čemu je možné CSS používat. Vysvětlit užití formátování písma, textu, odstavce, barvy, velikosti, obtékání, pozicování, tabulky. Obsahem této kapitoly je spíše přehled použitelných vlastností. Nemělo by být cílem učit se všechny zde uvedené vlastnosti. Důležité je uvědomit si možnosti co CSS nabízí, umět vyhledat potřebnou vlastnost a tu aplikovat.

⇒ Předpokládaná doba studia:

5 vyučovacích hodin

Stručný přehled vlastností a hodnot kaskádových stylů. CSS se rychle vyvíjí, vlastností fungujících v prohlížečích přibývá.

V prvním sloupci jsou vlastnosti použitelné při deklaraci stylu, v dalším sloupci použitelné hodnoty, v třetím výklad významu těchto hodnot. Nezáleží na velikosti písem. Zápis stylu do hlavičky dokumentu je potom symbolicky. Uvedené příklady se mohou lišit od skutečné interpretace v Internetovém prohlížeči!

11.1 PÍSMO (FONT)

Vlastnost	Hodnoty	Význam	Příklady	Poznámky
font-family	<i>seznam písma</i>	druh písma, font	font-family: Arial CE, sans-serif	Může se zadávat více písem za sebou, odděluje se čárkami. Pokud klient nemá v systému první font, bere další atd. Vyvarujte se používání „exotických“ písem!
font-style	normal italic oblique	normální kurzíva skloněné	font-style: normal <i>font-style: italic</i> <i>font-style: oblique</i>	Skloněné písmo (oblique) má být prostá geometrická transformace, kurzíva je jiný řez. Prohlížeče většinou užívají kurzívu i při oblique.
font-variant	normal small-caps	normální kapitálky	FONT-VARIANT: SMALL-CAPS	Kapitálky jsou velká písmena velikosti malých. Velká písmena by měla být trochu větší. IE 5 udělá sice kapitálky, ale zmenší i velká písmena, což by neměl.
font-size	xx-small x-small small medium large x-large xx-large <i>výška</i> <i>procento</i>	mrňavé maličké malé střední velké obří extra velké výška zvětšení	font-size: xx-small font-size: x-small font-size: small font-size: medium font-size: large font-size: 14pt font-size: 16px font-size:	Netscape se na procenta tváří divně. MS IE 3.x zase neumí správně zobrazovat jednotky em a ex. V IE 6 je vykreslována velikost závislá na !doctype.

			125%	
font-weight	normal bold bolder lighter 100, 200, 300,...900	normální tučné trochu tučnější trochu světlejší duktus vyjádřený číslem	font-weight: normal font-weight: bold font-weight: lighter font-weight: 100 font-weight: 400 font-weight: 600	U většiny fontů mají smysl jenom základní tloušťky: záleží to na výrobci fontu. Bolder a lighter se doporučuje nepoužívat.
font	všechny možné předchozí hodnoty nebo systémové písmo		font: italic bold 20px Arial	Tato deklarace je citlivá na pořadí jednotlivých údajů. Musí se použít v pořadí: kurzíva tučnost velikost jméno. Použije-li se v deklaraci např. font: 12pt/14pt, údaj za lomítkem se vztahuje k vlastnosti line-height.

11.2 TEXT / Odstavec

Vlastnost	Hodnoty	Význam	Příklady	Poznámky
text-decoration	none underline overline line-through blink	bez dekorace podtržení "nadtržení" přeškrtnutí blikání	text-decoration: none <u>text-decoration: underline</u> text-decoration: overline text-decoration: line-through text-decoration: blink	Teoreticky se dá zadávat více vlastností najednou. MS IE neumí blink.
text-transform	none capitalize uppercase lowercase	bez transform. Začátky Slo Velké VELKÁ PÍSMENA malá písmena	Text-Transform: none Text-Transform: capitalize TEXT-TRANSFORM: UPPERCASE text-transform: lowercase	
word-spacing	normal <i>délka</i>	zvětšená mezislovní mezera	word-spacing: normal word - spacing : 100px	Prohlížeče podporují od šestých verzí.
letter-spacing	normal <i>délka</i>	prostrkání znaků zvětšené o délku	letter-spacing: normal l e t t e r - s p a c i n g : 5 p t	
line-height	normal <i>výška</i> <i>násobek</i> <i>procento</i>	výška řádku absolutní výška násobek zvětšení	line-height: 3 line-height: 8px line-height: 80%	
text-	<i>délka</i>	odsazení	text-indent: 50px;	

indent	<i>procento</i>	prvního řádku	druhý řádek odstavce	
text-align	left right center justify	zarovnání vlevo vpravo na střed do bloku	text-align: left text-align: right text-align: center text-align: justify blablaba blablaba blablaba bla bla	Dá se použít jen u blokových elementů, tj. u věcí, které má smysl zarovnávat, například u odstavců.
vertical-align	baseline sub super top text-top middle bottom text-bottom <i>procento</i>	na řádek dolní index horní index co nejvýše vršek k vršku střed na střed co nejnižší spodek na spodek procento výšky	baseline řádek sub řádek super řádek top řádek text-top řádek middle řádek bottom řádek text-bottom řádek 30% řádek	Vertikální zarovnání nízkého prvku na vyšším řádku. Vlastnosti top, middle a bottom se dají použít u buněk tabulky a u obrázků na řádku.
display	block inline list-item none	blokový element řádkový element seznam nezobrazí se	display: block display: inline display: list-item 	Jde o to říct prohlížeči, že některý element je druhu odstavec (blok), nebo že má být zarovnán do řádku, nebo že je to seznam. Nejzajímavější je možnost nezobrazení.

11.3 BARVY A POZADÍ

Vlastnost	Hodnoty	Význam	Příklady	Poznámky
color	<i>barva</i>	barva písma	color: blue color: #00FF00	Barva písma a základních rámečků nebo barva toho, k čemu se to vztahuje.
background-color	<i>barva</i> transparent	barva pozadí průhledné pozadí	background-color: yellow background-color: transparent	Barva pozadí. Dá se zadávat libovolná barva.
background-image	none url(<i>cesta</i>)	obrázek na pozadí	background-image: url('pozadi.gif')	
background-repeat	repeat no-repeat repeat-x repeat-y	pozadí se opakuje neopakuje opakuje v ose X nebo v ose Y	background-image: url('pozadi.gif'); background-repeat: repeat background-repeat: no-repeat background-repeat: repeat-x	

background-attachment	scroll fixed	pozadí se posouvá pozadí je jako přibité		Fixed se používá v souvislosti s rámy.
background-position	top, center, bottom left, center, right, <i>délka,</i> <i>procento</i>	Poloha obrázku na pozadí (nejčastěji pokud se neopakuje)	background-image: url('pozadi.gif'); background-repeat: no-repeat background-position: right 50%	2 hodnoty se oddělují mezerou. První patří k horizontální, druhá hodnota k vertikální poloze.
background		<i>všechny výše uvedené hodnoty</i>	background: url('pozadi.gif') no-repeat scroll silver center bottom	

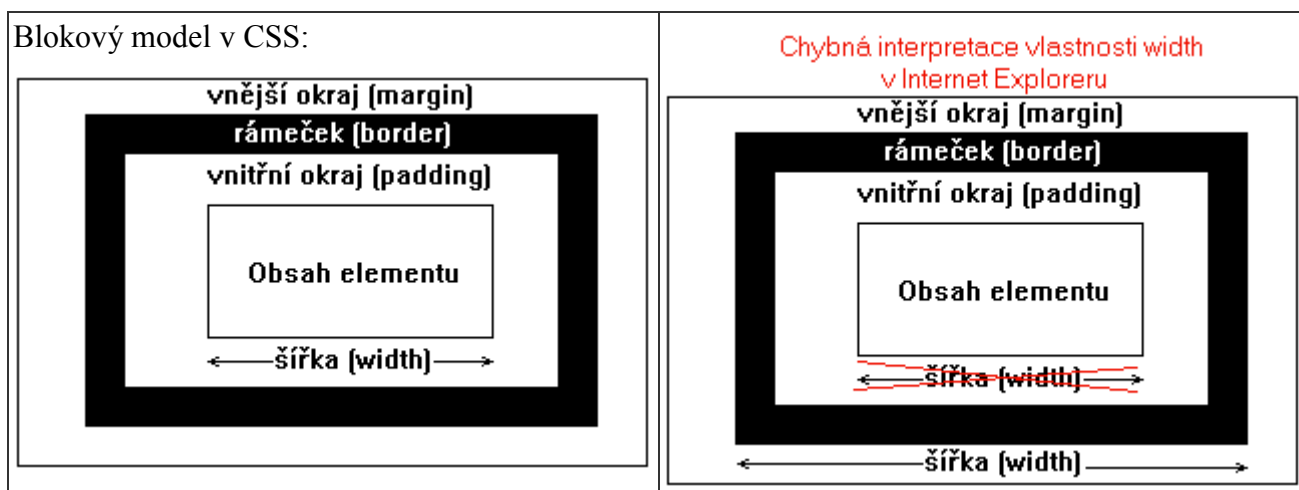
URL se zadává do závorek a apostrofů, např.: background-image: url('pozadi.gif'). Jsou ale možné i uvozovky nebo jenom závorky. URL může být absolutní i relativní, je však citlivé na velikost písmen.

11.4 VELIKOST A OBTÉKÁNÍ

Vlastnost	Hodnoty	Význam	Poznámky
width	auto <i>šířka</i> <i>procento</i>	automatická šířka nastavená šířka procento *	V IE nelze nastavit width pro body. Prohlížeče se velmi liší v tom, u kterých objektů jsou ochotny šířku akceptovat. V Internet Exploreru 4 a vyšších je do šířky nesprávně započítávána šířka paddingu a borderu – viz. kapitolu Okraje.
height	auto <i>výška</i> <i>procento</i>	automatická výška nastavená výška procento *	Dá se nastavit jenom blokovým tagům. Nejlépe funguje u <div>.
float	left right none	umístění plovoucího (obtékaného) objektu, či zda je neplavec	
clear	left right both none	čekání na skončení plovoucích objektů zleva, zprava, obou, nebo žádných	Používá se namísto atributu "clear" u tagu BR. Většinou u nadpisů pod obrázky.

* Procenta v této tabulce se vztahují k šířce (výšce) rodičovského elementu. Šířka rodiče je nejčastěji šířka dokumentu (nezávislá na okně), kdežto procentuální výška nevnořených elementů se počítá z výšky **okna!**

11.5 OKRAJE



Vlastnosti uvedené v této tabulce lze spolehlivě aplikovat pouze na tzv. blokové elementy, což jsou většinou buňky tabulky nebo odstavce. Obrázek ilustruje významy vlastností.

Vlastnost	Hodnoty	Význam	Poznámky
margin	délka procento auto	šířka vnějšího okraje procento * automatický okraj	Možno zadávat všechny čtyři okraje dohromady nebo zvlášť **. IE 5 asi nepodporuje záporné hodnoty, IE 4 a NN 4 ano.
margin-top margin-left margin-bottom margin-right	<i>jako u margin</i>	vnější okraj horní levý spodní pravý	Zápis vlastnost-strana Netsape 4 v některých verzích neumí.
padding	délka procento	šířka vnitřního okraje procento *	
padding-top padding-left padding-bottom padding-right	<i>jako u padding</i>	horní vnitřní okraj levý spodní pravý	Zápis vlastnost-strana Netsape 4 neumí.

** Při zadávání čtyř hodnot najednou se vztahují ke stranám elementu v pořadí: horní, pravá, dolní, levá. Např.: padding: 12px 3px 6px 9px. Prostě hodinové ručičky.

11.6 RÁMEČKY

Vlastnost	Hodnoty	Význam	Příklady	Poznámky
border-width	thin medium	šířka rámečku slabá		Možno zadávat všechny čtyři okraje dohromady nebo zvlášť **.

	thick <i>délka</i>	normální tlustá nastavená		
border-top-width border-left-width border-bottom-width border-right-width	<i>jako u border-width</i>	horní šířka rámečku levá spodní pravá		
border-color	<i>barva</i>	barva rámečku	<u>border-color: red;</u> <u>border-style: solid</u>	Možno zadávat všechny čtyři okraje dohromady nebo zvlášť **. Pro NN 4 nelze rozlišovat jednotlivé čtyři okraje.
border-style	none, dotted, dashed, solid, double, groove, ridge, inset, outset	Druh rámečku: žádný tečkovaný čárkovaný plný dvojitý příkop val d'olík návrší	border-style: none border-style: dotted border-style: dashed border-style: solid border-style: double border-style: groove border-style: inset	IE 4 a 5 zobrazuje dotted a dashed jako solid a stínuje vše černou barvou. Ostatní prohlížeče včetně IE 5.5 zobrazují správně a stínují šedou. IE 6 zobrazuje úzkou dotted jako dashed. Netscape styl rámečku podporuje pouze v zápisu border.
border-top border-left border-bottom border-right	<i>barva, tloušťka a styl</i>	celkové vlastnosti strany rámečku		
border	<i>barva, tloušťka a styl</i>	všechny vlastnosti rámečku	<u>border: solid blue</u> <u>2px</u>	

Slovníček okrajů a rámečků

border	margin	padding	width	top	bottom	left	right
rámeček	vnější okraj	vnitřní okraj	šířka (rámečku)	horní	spodní	levý	pravý

Prohlížeče se velmi liší v tom, na jaký tag dovolí okraje a velikost aplikovat. U některých tagů styl prostě ignorují.

11.7 SEZNAMY

Všechny vlastnosti seznamů lze aplikovat na tagy , <dir>, <menu> a .

Vlastnost	Hodnoty	Význam	Příklady	Poznámky
list-style-type	disc circle square decimal lower-roman lower-alpha upper-alpha none	puntík kolečko čtvereček číslování římské číslice ábecedkování ABCD bez odrážek	<ul style="list-style-type: none"> • disc ○ circle ▪ square 4. decimal v. lower-roman f. lower-alpha G. upper-alpha H. none	
list-style-image	none URL(<i>cesta</i>)	normální nebo obrázková odrážka	<ul style="list-style-type: none"> • none • list-style-image: URL('pozadi.gif') 	
list-style-position	inside outside	odrážky v úrovni textu odrážka mimo text	<ul style="list-style-type: none"> • list-style-position: inside • list-style-position: outside 	Při inside je puntík jakoby součástí dalšího textu.

11.8 POZICOVÁNÍ

Následující vlastnosti nefungují v IE 3, NN 3 a v Opeře 3.

Vlastnost	Hodnoty	Význam	Poznámky
position	absolute relative static	absolutní umístění relativní umístění normální umístění	Vizte např. http://www.jakpsatweb.cz/css/css-pozicovani.html .
left	auto <i>délka</i> <i>procento</i>	bez posunutí posunutí vpravo o délku nebo o procento	Pro prvky s position: absolute nebo position: relative. Vlevo se posouvá zápornou hodnotou.
top	auto <i>délka</i> <i>procento</i>	bez posunutí posunutí dolů o délku nebo o procento	Pro prvky s position: absolute nebo position: relative. Nahoru se posouvá zápornou hodnotou.
right	auto <i>délka</i> <i>procento</i>	pozicování od pravého okraje okna nebo elementu	Variace na vlastnosti left a top (top a left ale vždy vyhrají). Pouze pro objekty s position: absolute. Podpora od IE 5, v Opeře a v Mozille, ve starších prohlížečích většinou vůbec nebo špatně.
bottom	auto <i>délka</i> <i>procento</i>	pozicování od spodního okraje okna nebo elementu	

overflow	visible hidden scroll auto	nechat přetékat oříznout vždy rolovat rolování, je-li třeba	Pro elementy, které mají nastavené rozměry a nevejdou se do nich. V IE fungují i overflow-x a overflow-y.
z-index	auto číslo	definice překrývání elementů jakoby v ose z	Nefunguje pro tagy iframe, select (v IE) pro a flashové animace.
visibility	visible hidden	viditelný element skrytý (neviditelný)	U skrytých objektů se vyhradí místo, jako by tam byly (narozdíl od display: none).

11.9 TABULKY

Vlastnost	hodnoty	význam	poznámky
table-layout	auto fixed	nerozměrovaná tabulka se přizpůsobuje oknu; fixed = tabulka se nezužuje do okna	Podpora v IE 5+, v Mozille a v Opeře. Použití pouze u tagu <table>.
border-collapse	separate collapse	buňky v tabulce mají rámečky oddělené sousední buňky mají vykreslený rámeček společně jednou čarou	Podpora v IE 5+, v Mozille a v Opeře. Použití pouze u tagu <table>.

Úkol

Vytvořte soubor s příponou htm a soubor s příponou css. Do html dokumentu vložte kód, kterým zajistíte připojení externího css souboru.

Cvičení:

Vyhledej v předcházejících tabulkách formátování, které se ti líbí a pokus se ho aplikovat na libovolné konkrétní stránce.

Cvičení:

Navrhněte definici stylu, která zvýrazní text (obsah značky) nebude zvýrazňovat kurzívou, ale barevným pozadím (např. barvou #ffff99).

Cvičení:

Vytvořte definici stylu s následujícími vlastnostmi

- všechna písmena budou bezpatková (definice v selektoru body). Vyzkoušejte různé fonty
- text formátovaný elementem <h1> bude zobrazen kapitálkami
- hypertextové odkazy se po najetí myši změní na červenou barvu
- pozadí celého okna prohlížeče bude #dfdfa7. Elementy <h1> a <h2> budou mít barvu pozadí #ffe680
- stáhněte si libovolný obrázek, který bude na pozadí celého dokumentu. Odkoušejte různé varianty nastavení vlastností background.

Vytvořte stránku, která bude demonstrovat výsledný vzhled.

Cvičení:

Vytvořte definici stylu s následujícími vlastnostmi:

- šířka textu bude 60 % šířky okna
- text bude umístěn uprostřed stránky (levý i pravý okraj stejně velké)
- barva pozadí stránky #000066, barva textu #ffffff, barva odkazů #ffcc00 a navštívených odkazů #999966.

Vytvořte stránku, která bude demonstrovat výsledný vzhled.

Cvičení:

Vytvořte definici stylu s následujícími vlastnostmi:

- text formátovaný elementy `<h1>` a `<h2>` bude mít definovanou výplň o velikosti `1ex` nahoře a dole a `1em` vpravo a vlevo
 - text formátovaný elementy `<h2>` bude orámován čárkovanou čarou o tloušťce `1px` a barvě `#666633`
 - text formátovaný elementem `<p>`, bude zarovnán do bloku.
 - text formátovaný elementem `<a>`, bude po najetí myši nepodtržen
 - že text formátovaný elementem `<h2>` bude převeden na velká písmena (verzálky)
 - seznam se třídou `seznamprojektu` bude mít jako odrážku prázdný kroužek.
- Vytvořte stránku, která bude demonstrovat výsledný vzhled.

Cvičení:

Vytvořte stránku s tabulkou 4 x 6 buněk.

Vytvořte definici stylu s následujícími vlastnostmi:

- tabulka bude orámována jednoduchou černou čarou o síle `1px`
- záhlaví tabulky bude mít tučné písmo
- záhlaví a zápatí tabulky (formátované elementy `<thead>` a `<tfoot>`) budou mít barvu pozadí `#999966`
- uvnitř tabulky bude mřížka vykreslená jednoduchou čarou o síle `1px` s barvou `#333300`
- jednotlivé buňky budou mít výplň o velikost `0.5ex`.

Cvičení:

Vyhledej na Internetu stránky zabývající se CSS. Zkus odpovědět na otázku, jestli se CSS dále vyvíjí. Pokud ano, tak zkus nalézt nějaké nové prvky, které CSS umí. Pokus se je aplikovat. Jde to? Pokud ne, tak proč by tomu tak molo být?

12. PRAVIDLA PRO TVORBU PŘÍSTUPNÉHO WEBU

⇒ Cíl kapitoly:

Vysvětlit technologická a estetická pravidla při tvorbě webu. Navigaci na stránkách. Zásady psaní webu např.: práci s webovou stránkou řídí uživatel, informace jsou srozumitelné a přehledné, ovládání webu je jasné a pochopitelné, kód je technicky způsobilý a strukturovaný.

⇒ Předpokládaná doba studia:

3 vyučovací hodiny

Přístupný web je takový, který **neklade** uživateli žádné překážky v jeho používání a to bez ohledu na uživatelské technické vybavení, jeho schopnosti, znalosti či aktuální zdravotní stav. Přístupnost webu si tedy klade za cíl poskytnout každému návštěvníkovi stránek stejné (tj. všechny) informace, umožnit web používat všem.

Existuje **nepovinný** předpis, který říká, jak by měl vypadat správný web a čeho by se měli autoři na svých stránkách vyvarovat.

12.1 OBSAH WEBOVÝCH STRÁNEK JE DOSTUPNÝ A ČITELNÝ

⇒ Každý netextový prvek nesoucí významové sdělení má svou **textovou alternativu**.

Obrázky s textem

Obrázky, které v sobě mají obsažený text (logo webu, obrázková tlačítka, obrázkové nadpisy...) mají jako alternativní hodnotu text, který je v obrázku.

Obrázky s informační hodnotou, ale bez textu

U obrázků, které nesou obrazovou informaci (fotky) mají jako alternativní text krátký popis toho, co je na obrázku. U fotografie člověka je tím popisem jeho jméno.

⇒ Informace sdělované prostřednictvím skriptů, objektů, appletů, kaskádových stylů, obrázků a jiných doplňků na straně uživatele **jsou dostupné** i bez kteréhokoli z těchto doplňků.

⇒ Informace sdělované barvou jsou dostupné i bez barevného rozlišení.

⇒ Barvy popředí a pozadí jsou dostatečně **kontrastní**. Na pozadí není vzorek, který snižuje čitelnost.

⇒ Předpisy určující velikost písma **nepoužívají absolutní jednotky**.

Velikost písma v prohlížeči musí být za všech okolností zvětšitelná. Není to jen kvůli uživatelům s horším zrakem, je to i kvůli všem ostatním, kteří si třeba nechtějí kazit oči i pro všechny ostatní kdo se dostali k webům s módou miniaturního písma.

Jak zvětšit písmo

Ve většině prohlížečů umožňuje zvětšení písma sám prohlížeč ať už zadáte jakékoliv jednotky. Internet Explorer toto neumožňuje, pokud je písmo zadáno v jednotkách pt, px, mm, cm. Proto je nutné používat vždy relativní jednotky, tj. např. jednotky em či procentuální vyjádření (např. 80%).

⇒ Předpisy určující typ písma obsahují **obecnou rodinu** písem.

V definici rodiny písma – CSS vlastnost font-family – musí být ve výpisu písem vždy definována obecná rodina a to vždy **jako poslední alternativa**, např.:

```
font-family: 'Arial CE', 'Helvetica CE', Arial, sans-serif
```

12.2 PRÁCI S WEBOVOU STRÁNKOU ŘÍDÍ UŽIVATEL

⇒ Obsah WWW stránky se mění, jen když **uživatel** aktivuje nějaký prvek.

⇒ Webová stránka bez přímého příkazu uživatele **nemanipuluje** uživatelským prostředím.

⇒ Nová okna se otevírají jen v odůvodněných případech a uživatel je na to předem **upozorněn**.

⇒ Na webové stránce **nic nebliká** rychleji než jednou za sekundu.

Blikání na stránce, resp. jakýkoliv rychlý pohyb je pro uživatele nepříjemný a odvádí pozornost od obsahu stránky, zkrátka ruší, někdy nesnesitelně. Pravidlo se samozřejmě týká neustálého blikání, které bliká samo, nejde tu o žádné efekty při přejetí myši apod.

⇒ Webová stránka **nebrání uživateli** posouvat obsahem rámu.

⇒ Obsah ani kód webové stránky nepředpokládá ani nevyžaduje konkrétní způsob použití ani konkrétní výstupní či ovládací zařízení.

12.3 INFORMACE JSOU SROZUMITELNÉ A PŘEHLEDNÉ

⇒ Webové stránky sdělují informace **jednoduchým jazykem** a srozumitelnou formou.

⇒ **Úvodní webová stránka** jasně popisuje smysl a účel webu. Název webu či jeho provozovatele je zřetelný.

⇒ Webová stránka i jednotlivé prvky textového obsahu uvádějí své hlavní sdělení na svém **začátku**.

⇒ Rozsáhlé obsahové bloky jsou rozděleny do menších, výstižně nadepsaných celků.

⇒ Informace zveřejňované na základě zákona jsou dostupné jako textový obsah stránky.

⇒ Na samostatné webové stránce je uveden kontakt na technického správce a prohlášení jasně vymezující míru přístupnosti webu a jeho částí. Na tuto webovou stránku odkazuje každá stránka webu.

12.4 OVLÁDÁNÍ WEBU JE JASNÉ A POCHOPITELNÉ

- ⇒ Každá webová stránka má smysluplný **název**, vystihující její obsah.
- ⇒ Navigační a obsahové informace jsou na webové stránce zřetelně odděleny.
- ⇒ **Navigace je srozumitelná** a je konzistentní na všech webových stránkách.
- ⇒ Každá webová stránka (kromě úvodní webové stránky) obsahuje odkaz na vyšší úroveň v hierarchii webu a odkaz na úvodní WWW stránku.
- ⇒ Všechny webové stránky rozsáhlejšího webu obsahují odkaz na přehlednou mapu webu.
- ⇒ Obsah ani kód webové stránky nepředpokládá, že uživatel již navštívil jinou stránku.
- ⇒ Každý formulářový prvek má přiřazen výstižný nadpis.
- ⇒ Každý rám má vhodné jméno či popis vyjadřující jeho smysl a funkčnost.

12.5 ODKAZY JSOU ZŘETELNÉ A NÁVODNÉ

- ⇒ Označení každého odkazu výstižně popisuje jeho cíl i bez okolního kontextu.
- ⇒ Stejně označené odkazy mají stejný cíl.
- ⇒ **Odkazy jsou odlišeny** od ostatního textu, a to nikoli pouze barvou.

Pravidlo podtrhávání odkazů je velmi důležité, hlavně kvůli použitelnosti webu. Netýká se žádné menšiny uživatelů, týká se úplně všech a jeho nedodržování pohyb uživatele po webu ovlivňuje skutečně hodně. Aby mělo podtrhávání odkazů vůbec nějaký význam, je zároveň důležité nepodtrhávat žádný text, který odkazem není.

- ⇒ Uživatel je předem jasně upozorněn, když odkaz vede na **obsah jiného typu**, než je webová stránka. Takový odkaz je doplněn sdělením o typu a velikosti cílového souboru.

12.6 KÓD JE TECHNICKY ZPŮSOBILÝ A STRUKTUROVANÝ

- ⇒ Kód webových stránek odpovídá nějaké zveřejněné finální **specifikaci** jazyka *HTML* či *XHTML*. **Neobsahuje** syntaktické **chyby**, které je správce webových stránek schopen odstranit. Viz kapitolu Validní kód.
- ⇒ V metaznačkách je uvedena použitá **znaková sada** dokumentu.
- ⇒ Prvky tvořící nadpisy a seznamy jsou korektně vyznačeny ve zdrojovém kódu. Prvky, které netvoří nadpisy či seznamy, naopak ve zdrojovém kódu takto vyznačeny nejsou.

- ⇒ Pro popis vzhledu webové stránky jsou upřednostněny **stylové předpisy (CSS)**.
- ⇒ Je-li tabulka použita pro rozvržení obsahu webové stránky, neobsahuje záhlaví řádků ani sloupců. Všechny tabulky zobrazující tabulková data naopak záhlaví řádků a/nebo sloupců obsahují.
- ⇒ Všechny tabulky dávají smysl čtené po řádcích zleva doprava.

13. VALIDNÍ KÓD

⇒ Cíl kapitoly:

Vysvětlit co je to validita, proč psát validně, možnosti ověření validity - Validátor W3C.

Co může způsobit nevalidní kód.

⇒ Předpokládaná doba studia:

1 vyučovací hodina

Validní kód znamená, že výsledný kód webových stránek odpovídá nějaké zveřejněné finální specifikaci jazyka *HTML* či *XHTML*. **Neobsahuje syntaktické chyby**, které je správce webových stránek schopen odstranit.

Validní kód je takový, který neobsahuje žádné chyby v syntaxi podle zvolené specifikace jazyka. To znamená, že stránka musí mít určený `DOCTYPE` a být oproti němu validní. Validita webu se pak dá snadno zjistit pomocí validátoru.

13.1 CO MŮŽE ZPŮSOBIT NEVALIDNÍ KÓD

Pokud kód není validní, v nejjednodušším případě to může znamenat **chybné zobrazení** stránky, kdy některá část stránky může překrýt jinou. Takový problém se pak týká všech uživatelů. Většinou je ale toto pravidlo důležité spíše pro interpretaci stránky např. hlasovou čtečkou, která se v nevalidním kódu může snadno ztratit nebo chybně interpretovat význam. Stejně se pak může ztratit i vyhledávací robot a stránku chybně zaindexovat nebo nezaindexovat vůbec.

13.2 VALIDÁTOR W3C

Online validátor W3C nalezneme na adrese <http://validator.w3.org>. Práce s ním je jednoduchá. Na úvodní stránce je třeba sdělit validátoru, který soubor se bude validovat. Zvolit můžete buď **Validate by URL**, kde se do políčka **Address** zadá URL adresa stránky. Nebo můžete zvolit **Validate by File Upload** a pomocí tlačítka **browse** projít svůj disk a vybrat (a potvrdit tlačítkem **check**) soubor k validaci.

Do políčka **Address (URL)** stačí správně zadat URL adresu stránky, jejíž validitu kontrolujeme.

Cvičení:

Zkontrolujte některou jednodušší stránku on-line validátorem. Analyzujte výsledky.

Cvičení:

Najdi na Internetu nástroj na kontrolu validity CSS souborů. Zkontrolujte několik jednodušších CSS souborů on-line validátorem. Analyzujte výsledky.

14. DOMÉNA A HOSTING

⇒ Cíl kapitoly:

Možnosti a způsoby umístění stránek na Internet. Vysvětlení pojmů doména, hosting.

⇒ Předpokládaná doba studia:

2 vyučovací hodiny

Pokud již považujeme své webové stránky za hotové, nic nebrání jejich zveřejnění na Internetu. V tuto chvíli stojíme před rozhodnutím kam a za kolik peněz je umístíme.

14.1 DOMÉNA

Prvním krokem bude výběr domény. Název domény je v podstatě jméno pro stránky. Nejdříve se musíme rozhodnout, zda budeme chtít doménu druhého nebo třetího řádu.

14.1.1 Doména prvního řádu

Pro Českou republiku je (*top-level doména*) .cz a spravuje ji zvolený registrátor. Tuto doménu nelze vlastnit.

14.1.2 Doména druhého řádu

Doména druhého řádu je obvykle placená. Má tvar *www.nazevdomeny.cz*.

14.1.3 Doména třetího řádu

Doména třetího řádu bývá obvykle zdarma umístěna na některém ze serverů poskytujících tyto služby (např. *webzdarma.cz*, *pipni.cz*, *pes.cz*). Obvykle má tvar *nazevdomeny.jmenoposkytovatele.cz* popř. *www.jmenoposkytovatele.cz/nazevdomeny*. Někdy je URL ještě složitější, což je hlavní nevýhodou těchto domén. Obvykle také musíte počítat s povinným umístěním reklamy serveru na vaše stránky.

14.2 HOSTING

Hostingem se rozumí místo, kde jsou stránky fyzicky umístěné. Pokud si vybereme doménu 3. řádu, bude umístěna na server, který jsme zvolili. U domény 2. řádu můžeme zaregistrovat **zvlášť doménu** a **zvlášť hosting**, ten musíme vybrat. Obvykle jde o hosting placený, ale existují i různé verze freehostingů. Placený hosting nabízí daleko lepší služby včetně různých garancí, obvykle nabízí větší prostor na disku, lepší administraci, statistiky, podporu více databází, také více e-mailů a podobně.

14.3 UMÍSTĚNÍ STRÁNEK

Pokud již máme prostor pro své stránky zaregistrovaný, dostaneme *login* (uživatelské jméno) a heslo pro přístup. V závislosti na charakteru námi vybrané služby budeme moci ke

svým stránkám přistupovat buď jen přes webové rozhraní, nebo i přes FTP klienta. Ať už tak či onak, jediné, co teď zbývá je zkopírovat stránky z disku na server. K přesunu se nejčastěji používají běžně dostupní FTP klienti (např. volně šiřitelný *CoffeeCup*, *Free FTP*, *FTP Commander* či *Total Commander*).

Významným rizikem FTP je, že přenáší uživatelské heslo a jméno v otevřeném tvaru – při odposlechu lze zneužít. Vhodnější variantou je tedy použití *scp*, které veškerou komunikaci šifruje. Volně šiřitelnou implementací pro operační systém MS Windows je např. *Putty*, lepší je však použít grafický program *WinSCP* (<http://winscp.vse.cz/eng/>).

Na webovém rozhraní obvykle procházíte disk, vyberete soubory a zvolíte přidat soubory. Přes FTP klienta jde o klasické kopírování souborů.

Vstupní stránku (stránka, která se má první zobrazit po zadání adresy webu do adresního řádku) musíme obvykle pojmenovat *index.html*, *index.htm*, *index.php* apod. Je možné, že se setkáte s jiným požadovaným názvem vstupní stránky (např. *default.htm*). Záleží na poskytovateli hostingu.

15. SLOVNÍČEK POJMŮ

- browser** – prohlížeč, program, který zobrazuje dokumenty, zveřejněné na Internetu
- CSS** – *Cascading Style Sheet* – Kaskádové styly – metoda pro pokročilé formátování HTML stránek
- DHTML** – *Dynamic HTML*, společné označení pro HTML, CSS a skriptování, které umožňuje lepší interakci s uživatelem
- DIV** – *division* – "oddíl, oddělení" HTML tag, který obklopuje určitou logickou oblast stránky. Je to tzv. blokový element, tzn. dělá za sebou zalomení řádku
- download** – "stažení" souboru
- font** – druh písma
- FTP** – *File Transfer Protocol* – Jeden z protokolů založených na *TCP/IP*, sloužící k přenosu souborů mezi dvěma uzly
- fulltext** – Způsob vyhledávání nebo také způsob organizace databáze textů umožňující porovnávání každého slova dokumentu se zadaným vzorem
- GIF** – *Graphic Interchange Format* – Výhodou, na rozdíl od formátu JPEG, je bezztrátová komprese dat. Může obsahovat i více obrázků, které se ve stanovených intervalech střídají - tzv. animovaný gif. Může obsahovat ale maximálně jen 256 barev
- HTML** – *hypertext markup language* – hypertextový značkovací jazyk
- HTTP** – *Hypertext Transfer Protocol* – Jeden z protokolů založených na *TCP/IP*, sloužící k přenosu hypertextového obsahu, nejčastěji pak stránek systému WWW
- hypertextový odkaz** – jedná se o určitou část dokumentu, například text nebo obrázek, který odkazuje na jiný dokument nebo jeho část. V internetovém prohlížeči se uživatel po kliknutí na odkaz přenesse na příslušný odkazovaný dokument
- JavaScript** – Programovací jazyk použitelný zejména v HTML stránkách. Vykonává se na počítači u klienta (tj. čtenáře)
- JPEG/JPG** – *Joint Photographic Experts Group* – Grafický formát, na rozdíl od GIF je použití až 16 milionů barev a malý objem obrázku, dosažený ale ztrátovou kompresí
- Meta tagy** – umísťují se do hlavičky HTML souboru a nesou nějakou další informaci, která není součástí dokumentu, např. informaci o autorovi nebo klíčová slova
- PHP** – původně *Personal Home Page*, nyní *hypertext preprocessor PHP* – Skriptovací jazyk fungující na straně serveru
- PNG** – *Portable Network Graphics* – Moderní grafický formát - možný nástupce formátu GIF. Formát barvy může být až 48-bitový, podporuje možnost zobrazení v různém rozlišení na obrazovce a při tisku
- protokol** – předpis, který určuje způsob komunikace mezi počítači, zajišťuje uskutečnění přenosu: navázání, udržení a ukončení spojení

SCP – náhrada FTP využívající SSH (*Secure SHell*) zabezpečení

server – bezobslužný program (nikoli tedy stroj) běžící na počítači permanentně zapojeném v počítačové síti, který nabízí služby dalším počítačům a serverům

skript – většinou je skriptem myšlen menší program napsaný nějakým skriptovacím jazykem, souslednost příkazů k provedení

software – programové vybavení, jakýkoliv program (např. editor textů, internetový prohlížeč, elektronický slovník)

Tag – "značka" – HTML jazyk sestává z tagů, které vymezují vzhled a smysl dokumentu

TCP/IP – *transmission control protocol / Internet protocol* – protokol, dle kterého funguje komunikace v síti Internet

upload – termín používaný pro přenos dokumentu (souboru) z uživatelského počítače na internetový server (nebo jiný počítač)

URL – *Uniform Resource Locator* – adresa umístění webové stránky (též internetová adresa; webová adresa)

W3C – *World Wide Web Consortium* – mezinárodní organizace založená v roce 1994 stanovující standardy pro WWW

webhosting – pronájem prostoru na WWW serveru. Každá webová prezentace využívá nějaký způsob webhostingu

weblog, blog – internetový deníček, jedná se zpravidla o jednoduchý redakční systém, který umožňuje publikovat na webu aktuální novinky autora blogu – blogera

webmaster, webdesigner – osoba, která vytváří, udržuje a stará se o webovou prezentaci

WWW – *World Wide Web* – zkratka pocházející z anglického WorldWide Web (v překladu celosvětová pavučina). Jedná se o jednu z nejužívanějších služeb Internetu, která nabízí k prohlížení „webové“ stránky. Charakteristickým rysem webu jsou tzv. hypertextové odkazy (odtud pavučina). Často se používá pouze zkrácený termín web

WYSIWYG editor – zkratka *What You See Is What You Get* znamená, že stránka psaná v takovémto typu editoru se zobrazuje už při psaní tak, jak bude (měla by) vypadat později v prohlížeči. HTML (popř. XHTML) kód je tak před uživatelem ukryt a není tedy potřeba jeho znalost

XHTML – *eXtensible HTML*, neboli značkovací jazyk, který nahrazuje dnes již zastaralý jazyk HTML

16. DOPORUČENÁ STUDIJNÍ LITERATURA

- ⇒ Musciano Chuck, Kennedy Bill: HTML a XHTML, kompletní průvodce, Computer Press, 2000, ISBN 80-7226-407-9
- ⇒ Kosek Jiří: HTML, tvorba dokonalých WWW stránek, Grada Publishing, 1998, ISBN 80-7169-608-0
- ⇒ Broža Petr: Tvorba WWW stránek pro úplné začátečníky, Computer Press, 2001, ISBN 80-7226-423-0
- ⇒ Staníček Petr: CSS kaskádové styly, Kompletní průvodce. ComputerPress, 2003, ISBN 80-7226-872-4
- ⇒ Hlavenka Jiří a kol.: Vytváříme WWW stránky a spravujeme moderní web site, Computer Press, 2002, ISBN 80-7226-748-5
- ⇒ King Andrew B.: Zrychlete své WWW stránky!, Zoner Press, 2003, ISBN 80-86815-02-1
- ⇒ Krug Steve: Web design - nenuťte uživatele přemýšlet, Computer Press, 2003, ISBN 80-7226-892-9
- ⇒ Prokop Marek: CSS pro webdesignery, Mobil Media, 2003, ISBN 80-86593-35-5
- ⇒ Satrapa Pavel: Web design, Neokortex, 1997, ISBN 80-902230-1-X
- ⇒ Siegel David: Creating Killer Web Sites, 2. vydání, New Riders Publishing, 1998, ISBN 1568304331
- ⇒ Simpson Alan: Jak si udělat svou první Webovou stránku, Svojtka & Co., 2000, ISBN 80-7237-302-2
- ⇒ Spainhour Stephen, Eckstein Robert: Webmaster v kostce. Computer Press, 1999, ISBN 80-7226-450-8
- ⇒ Špinar David: Tvoříme přístupné webové stránky. Zoner Press, 2004, ISBN 80-86815-11-0

Doporučené WWW stránky zabývající se tématy tvorby webu

- ⇒ Jak psát web, návod na HTML stránky – velice obsáhlý web o tvorbě internetových stránek <http://www.jakpsatweb.cz/>
- ⇒ Web tvorba: tvorba WWW stránek – aktualizované stránky o XHTML a jiných webových technologiích <http://www.webtvorba.cz/>.
- ⇒ Pixylophone - weblog Petra Staníčka s informacemi o webdesignu a CSS <http://www.pixy.cz/> (již neaktualizováno).

- ⇒ Sova v síti – Články, rady a novinky pro webmastery, webdesignéry a informační architekt od Marka Prokopa <http://www.sovavsiti.cz/>
- ⇒ Přístupnost – Postupy jak vytvářet weby bez bariér <http://pristupnost.nawebu.cz/>
- ⇒ Vše o WWW – stránky věnující se WWW od Jiřího Koska <http://www.kosek.cz/>
- ⇒ Validátor kódu – nástroj pro kontrolu validity stránek od W3C <http://validator.w3.org/>
- ⇒ Typo – Průvodce typografií <http://www.typo.cz/>
- ⇒ OpenSource CMS (aglicky) – mnoho zajímavých informací o CMS, zejm. a výběru vhodného produktu. Na stránkách si lze jednotlivé CMS i vyzkoušet.
- ⇒ Jak na web – vše pro web - stránky věnující se všemožným webovým technologiím <http://www.jaknaweb.com/>
- ⇒ Interval – webdesign a e-komerce <http://www.interval.cz/>

Seznam použitých zdrojů

➤ Internet:

- [1] VĚTROVSKÁ, Petra. *Web tvorba: tvorba WWW stránek* [online]. 2006, [cit. 2006-07-29]. <<http://www.webtvorba.cz/>>.
- [2] JANOVSKEÝ – YUHŮ, Dušan. *Jak psát web, návod na html stránky* [online]. Poslední revize 5. července 2006. [cit. 2006-07-29]. <<http://www.jakpsatweb.cz/>>.
- [3] GÁLA, Libor. *4IT121: Web Design* [online]. 2005. [cit. 2006-07-29]. <<http://nb.vse.cz/~GALA/it120/index.htm>>.
- [4] JEDNOTA ŠKOLSKÝCH INFORMATIKŮ, *Publikování na WWW: Slovníček pojmů* [online]. 2006. [cit. 2006-07-29]. <<http://www.jsi.cz/stranky/slovnicek.htm>>.
- [5] KOSEK, Jiří. *Domovská stránka Jirky Koska -- "VŠE O WWW"* [online]. Poslední revize 4. května 2006. [cit. 2006-07-29]. <<http://www.kosek.cz/>>.

➤ Knihy:

- [6] SATRAPA, Pavel. *Podklady pro vzdělávací kursy pedagogů - Vytváření WWW stránek*. 2005.