

Výuka HTML pro administrativní praxi

Teaching HTML for administrative practice

Marie Šebestová

Bakalářská práce
2011



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky
akademický rok: 2010/2011

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Marie ŠEBESTOVÁ**
Osobní číslo: **A08092**
Studijní program: **B 3902 Inženýrská informatika**
Studijní obor: **Informační a řídicí technologie**

Téma práce: **Výuka HTML pro administrativní praxi**

Zásady pro vypracování:

1. Seznamte se s HTML.
2. Zpracujte popis standardů HTML.
3. Realizujte sadu vzorových příkladů.
4. Navrhněte sadu testovacích úkolů včetně řešení.
5. Uveďte možný další rozvoj pomůcky.

Rozsah bakalářské práce:

Rozsah příloh:

Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam odborné literatury:

1. BROŽA, Petr. Tvorba WWW stránek pro úplné začátečníky. Praha : Computer Press, 2001. 149 s
2. STEJSKAL, Jan. Vytváříme WWW pomocí HTML, CSS a JavaScriptu. Praha : Computer Press, 2006. 250 s.
3. KUČERA, Miroslav. HTML - tipy a triky od profesionálů. Brno : UNIS publishing, 2001. 80 s.
4. HAUSER, Marianne; HAUSER, Tobias; WENZ, Christian. HTML a CSS - Velká kniha řešení. Praha : Computer Press, 2006. 912 s.
5. Jak psát web [online]. 2005, poslední aktualizace 20.12. 2010 [cit. 2011-01-31]. Dostupné z WWW: [http://www.jakpsatweb.cz/].

Vedoucí bakalářské práce:

Ing. Petr Šilhavý, Ph.D.

Ústav počítačových a komunikačních systémů

Datum zadání bakalářské práce:

25. února 2011

Termín odevzdání bakalářské práce:

7. června 2011

Ve Zlíně dne 25. února 2011

prof. Ing. Vladimír Vašek, CSc.
děkan



prof. Ing. Vladimír Vašek, CSc.
ředitel ústavu

ABSTRAKT

Cílem této bakalářské práce je vypracování základních podkladů pro seznámení a výuku jazyka HTML určených úplným začátečníkům, úvodu do CSS a dalších základních témat souvisejících s tvorbou webových stránek, v budoucnosti vyučovaných na Fakultě aplikované informatiky Univerzity Tomáše Bati ve Zlíně. Výstupem práce je CD obsahující vytvořené přednášky, vzorové příklady a zadání práce pro cvičení, související s jednotlivými probíranými kapitolami.

Klíčová slova:

HTML, CSS, webové stránky, webové prezentace, webdesign, publikování

ABSTRACT

An objective of this thesis is to create a base for teaching HTML language for beginners, which can be used in future at Faculty of Applied Informatics at Tomas Bata University in Zlín. The base will also contain of introduction to CSS and of other topics usefull for creating websites. An output of the thesis is a CD with lectures, examples and tasks for exercises/seminars, related to the chapters in the thesis.

Keywords:

HTML, CSS, web pages , web presentations, webdesign, web publishing

Dovoluji si upřímně poděkovat panu Ing. Petru Šilhavému, Ph.D. za cenné rady a pomoc vedoucí k realizaci mé bakalářské práce.

Dík patří i bývalé učitelce, paní Mgr. Anně Krušové za jazykovou korekturu této práce i psychickou podporu v době mých studií.

Prohlašuji, že

- beru na vědomí, že odevzdáním bakalářské práce souhlasím s jejím zveřejněním podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách) ve znění pozdějších právních předpisů bez ohledu na výsledek obhajoby;
- beru na vědomí, že bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému, dostupná k prezenčnímu nahlédnutí, jeden výtisk bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce;
- byla a jsem seznámena s faktem, že na mou bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – bakalářskou práci nebo poskytnout licenci k jejímu využití jen s předchozím písemným souhlasem Univerzity Tomáše Bati ve Zlíně, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše);
- beru na vědomí, že pokud bylo k vypracování bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji,

- že jsem bakalářskou práci vypracovala samostatně a použitou literaturu jsem přesně citovala. V případě publikace výsledků budu uvedena jako spoluautor.
- že odevzdaná verze bakalářské práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně 1.5.2011

.....
podpis diplomanta

OBSAH

ÚVOD.....	11
1 HTML.....	12
1.1 ÚVOD DO JAZYKA HTML.....	12
1.1.1 Co je HTML.....	12
1.1.2 Historie a vývoj HTML.....	12
1.1.3 Zdrojový text HTML.....	15
1.1.4 Základní značky pro tvorbu stránek	16
1.1.4.1 Hlava dokumentu	17
1.1.4.2 Tělo dokumentu	18
1.1.4.3 Poznámky.....	18
1.2 PROHLÍŽEČE	18
1.2.1 Internet Explorer	19
1.2.2 Firefox	19
1.2.3 Google Chrome	20
1.2.4 Opera	20
1.2.5 Safari	20
1.2.6 Další	21
1.3 MICROSOFT EXPRESSION WEB 4	21
1.4 TEXTY A BARVY	22
1.4.1 Formátování textu	22
1.4.1.1 Normální text	22
1.4.1.2 Odstavce a jejich zarovnání	23
1.4.1.3 Citace	24
1.4.1.4 Zobrazení předformátovaného textu	25
1.4.2 Nadpisy, písmo a velikost písma.....	25
1.4.2.1 Efekty a zvýrazňování písma	25
1.4.2.2 Fonty	26
1.4.2.3 Implicitní nastavení velikostí písma	27
1.4.2.4 Nadpisy	27
1.4.3 Barvy	27
1.4.3.1 Zadávání barev	27
1.4.3.2 Bezpečné barvy	28
1.4.3.3 Výchozí nastavení barev	29
1.5 SEZNAMY	29
1.5.1 Netříděný seznam	29
1.5.1.1 Víceúrovňové seznamy	30
1.5.2 Tříděný seznam	31
1.5.3 Seznam definic	32
1.6 TABULKY	33
1.6.1 Tvorba tabulek.....	33
1.6.2 Vlastnosti tabulky.....	33
1.6.2.1 Hlavička tabulky	33
1.6.2.2 Nadpis tabulky	34

1.6.3	Další nastavení tabulky	35
1.6.3.1	Okraj tabulky	35
1.6.3.2	Zarovnání tabulky	35
1.6.3.3	Pozadí a barvy	36
1.6.3.4	Vzdálenost buněk a textu	36
1.6.3.5	Ohraničení	37
1.6.4	Další parametry	38
1.6.4.1	Tvorba řádku tabulky	38
1.6.5	Buňky	39
1.6.5.1	Zarovnání textu v buňkách	39
1.6.5.2	Spojené buňky	40
1.6.5.3	Šířka a výška buňky	40
1.7	OBRÁZKY	41
1.7.1	Obrázky na webových stránkách	41
1.7.1.1	Používané formáty obrázků	42
1.7.1.2	Datový objem	42
1.7.2	Vkládání obrázku	42
1.7.3	Zarovnání obrázku	43
1.7.4	Velikost obrázků	44
1.7.5	Popiska	45
1.7.6	Rámeček	45
1.7.7	Volné okraje	46
1.7.8	Obrázky na pozadí stránky	46
1.8	ODKAZY	47
1.8.1	Tvorba odkazů	47
1.8.2	Odkaz na soubor	48
1.8.3	Odkaz na jiné místo dokumentu	49
1.8.4	Elektronická pošta	50
1.8.5	Další možnosti odkazů	51
1.8.5.1	Komentář odkazu	51
1.8.5.2	Otevírání nové stránky	51
1.9	FORMULÁŘE	51
1.9.1	Značka FORM	51
1.9.2	Prvky formuláře	52
1.9.3	Tlačítka	56
1.10	RÁMY	57
1.10.1	Definice ráků	57
1.10.1.1	Příklady použití ráků	58
1.10.2	Další parametry	59
1.10.2.1	Parametry FRAMESET	59
1.10.2.2	Parametry FRAME	60
1.10.3	Směrování stránek do ráků	61

2	ÚVOD DO CSS	62
2.1	HISTORIE	62
2.2	K ČEMU JSOU CSS DOBRÉ?	62
2.3	ÚVOD	62
2.3.1	Připojení stylů k dokumentu	62
2.3.1.1	Externí soubor	63
2.3.1.2	Definování stylu uvnitř dokumentu	63
2.3.2	Váha stylů	63
2.3.3	Přiřazení definic prvkům stránky	64
2.3.4	Dědičnost	64
2.3.5	Komentáře	64
2.4	FORMÁTOVÁNÍ TEXTU	65
2.4.1	Typ písma	65
2.4.2	Velikost písma	65
2.4.3	Řez písma	67
2.4.4	Varianta písma	68
2.4.5	Dekorace písma	68
2.4.6	Zarovnání textu	68
2.5	BARVY A OBRÁZKY	69
2.5.1	Barvení dokumentů	69
2.5.2	Pozadí	69
2.5.3	Opakování obrázku	69
2.5.4	Umístění pozadí (obrázku)	70
2.6	SEZNAMY	70
2.6.1	Odrážky a číslování	70
2.6.2	Umístění odrážky	71
2.6.3	Obrázek jako odrážka	71
2.7	ODKAZY	71
2.7.1	Barva odkazu	71
2.7.2	Ozdobení textu	72
2.7.3	Pseudotřídy	72
2.8	TŘÍDY A IDENTIFIKÁTORY V CSS	72
2.8.1	Třída CLASS	72
2.8.1.1	Dědičnost	73
2.8.1.2	Kontextová deklarace s třídou	73
2.8.2	Identifikátor ID	74
2.9	CSS VALIDÁTOR	74
3	TVORBA A PUBLIKOVÁNÍ WEBOVÝCH STRÁNEK	75
3.1	PROGRAMY POUŽÍVANÉ PRO TVORBU WEBOVÝCH STRÁNEK	75
3.1.1	PSPad	75
3.1.2	Elipse	76
3.1.3	NetBeans IDE	76
3.1.4	WYSIWYG editory	77
3.1.4.1	Nvu 1.0	77

3.1.4.2	WYSIWYG Web Builder	78
3.1.4.3	Adobe Dreamweaver.....	79
3.2	STRUČNÝ ÚVOD DO WEBDESIGNU	80
3.2.1	Grafický design	80
3.2.2	Rady do začátku	80
3.2.3	Barvy a pozadí.....	81
3.2.4	Výběr písma	81
3.2.5	Zvýrazňování textu.....	82
3.2.6	Základní typografická pravidla.....	82
3.2.6.1	Interpunkce	82
3.2.6.2	Uvozovky.....	82
3.2.6.3	Pomlčka	83
3.2.7	Grafické programy.....	83
3.3	UMÍSTĚNÍ STRÁNEK NA INTERNET	84
3.3.1	Jak správně připravit stránky ke zveřejnění	84
3.3.1.1	Pořádek v souborech	84
3.3.1.2	Relativní odkazy	84
3.3.1.3	Jména souborů	84
3.3.1.4	Web ≠ Windows + Microsoft + Internet Explorer.....	84
3.3.2	Webhosting	85
3.3.3	Možnosti přenosu stránek na webový server	85
3.3.3.1	FTP.....	85
3.3.3.2	FTPS	86
3.3.3.3	SCP	86
3.3.3.4	Další možnosti	86
ZÁVĚR		87
ZÁVĚR V ANGLIČTINĚ.....		88
SEZNAM POUŽITÉ LITERATURY.....		89
SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK		91
SEZNAM OBRÁZKŮ		92
SEZNAM TABULEK.....		93
SEZNAM PŘÍLOH.....		94

ÚVOD

Bakalářská práce je určena především jako základní učební materiál pro předmět, který se bude zabývat tvorbou webových prezentací za pomoci jazyka HTML.

Jazyk HTML je základním stavebním prvkem pro tvorbu jakýchkoliv webových prezentací. Projekt, který vznikl v CERNu, se za 20 let své existence rozrůstal a stále rozrůstá o další možnosti tvorby.

HTML jazyk není nikterak složitý a naučit se s ním pracovat může skutečně kdokoli. Mnoho programátorů jej ovládá do té míry, že pro svou tvorbu nepotřebují žádný editor a vystačí si s obyčejným Poznámkovým blokem, který je standardní součástí každé verze Microsoft Windows. Pro jednoduchost se zřejmě stal základním prvkem, na nějž se nabalovaly nové a složitější jazyky. Základ napsaný v HTML se dá dále upravovat například kaskádovými styly nebo-li CSS, či JavaScriptem. Využívá se i v kombinaci s PHP. K čemu by sloužila databáze bez rozhraní vytvořeného tak, aby v něm mohl pracovat člověk bez zkušeností a potřebných znalostí? Pomocí HTML jsou napsány všechny webové stránky. V současné době, kdy je Internet součástí téměř každé domácnosti, stal se nezbytným prvkem v administrativní činnosti či výuky na školách, je prakticky nemožné se vyhnout každodennímu setkání s výtvary napsané výše uvedeným jazykem.

1 HTML

1.1 Úvod do jazyka HTML

1.1.1 Co je HTML

Hypertext Markup Language (HTML) je jazyk určený k zobrazování dat předem zadaným způsobem; jedná se vlastně o podrobný návod, jak zobrazit přijatá data na obrazovce – kód HTML, je tedy přesným a stručným vyjádřením všeho, co chceme zobrazit v programu obecně zvaném prohlížeč. Jedná se o jazyk sloužící k popisu webové stránky. Aby prohlížeč tomuto vyjádření rozuměl, je potřeba jej přepsat právě do kódu HTML, který je podstatně stručnější a hlavně přesnější než slovní vyjádření. Člověku, jenž nikdy s HTML nepracoval, se může zdát způsob zápisu nesrozumitelný, ale postupným studiem si lze tento jazyk bez problémů osvojit. [1]

1.1.2 Historie a vývoj HTML

HTML jazyk a jeho první definice byla vytvořena v roce 1991 jako součást projektu, který vznikl v CERNu a na němž pracoval Tim Berners-Lee. Jednalo se o výsledek projektu, který měl vyřešit problémy se sdílením informací ve velké instituci. Tato verze HTML umožňovala rozčlenit text do několika logických úrovní, použít několik druhů zvýraznění textu včetně zařazení odkazů a obrázků.

Při návrhu HTML se nepředpokládalo, že by autoři webových stránek museli tento jazyk ovládat. První verze WWW softwaru byla napsána pro operační systém NextStep a obsahovala prohlížeč s integrovaným editorem webových stránek. Software, který byl vyvinut v CERNu včetně specifikace jazyka HTML, byl pro veřejnost uvolněn v roce 1991. Pracovníci NCSA vyvíjející prohlížeč Mosaic však považovali za příliš obtížné implementovat do programu rovnou i editor HTML. Z tohoto důvodu a rovněž kvůli faktu, že ne každý provozuje na svém počítači NextStep, bylo rozhodnuto o oddělení prohlížeče od editoru. Od tohoto okamžiku museli autoři jazyk HTML ovládat, neboť jinak by nemohli stránky vůbec tvořit.

Požadavky uživatelů na WWW vzrůstaly, a proto byl jazyk HTML rozšířen o nové prvky. Kvůli zachování kompatibility mezi jednotlivými modifikacemi HTML byl vytvořen nový návrh standardu HTML 2.0, který zahrnoval v té době všechny běžně používané prvky.

Verze HTML 2.0 má zároveň dvě úrovně. První z nich (Level 1) pouze málo rozšiřuje předchozí verzi HTML. Level 2 navíc definuje práci s formuláři.

Další rozšíření jazyka známé jako HTML+ zahrnují zejména navýšení o vytváření tabulek, matematických vzorců a dalších prvků pro kontrolu výsledného vzhledu. Dave Raggett, pracovník laboratoří Hewlett-Packard, HTML+ formalizoval a vytvořil jeho definici jako DTD v jazyce SGML. Na jaře roku 1995 vznikl návrh standardu HTML 3.0.

HTML 3.0 byl mohutným skokem vpřed. V té době se neobjevil nikdo, kdo by dokázal implementovat prohlížeč s jeho podporou. Tehdy vývoj standardů webu již koordinovalo konsorcium W3C. Jeho členové se shodli na vlastnostech, o něž rozšíří HTML 2.0 a vytvořili verzi HTML 3.2. Tato verze však zdaleka neobsahovala vše z HTML 3.0. Nové prvky HTML 3.2 jsou jen jakousi směskou, kterou v té době podporovaly nejnovější prohlížeče. Stejný postup jako při vzniku verze 2.0 se tedy opakoval – jazyk se sjednotil na průniku možností nejrozšířenějších prohlížečů. Kromě tabulek přibýly ve verzi 3.2 zejména realizace lepší kontroly formátování včetně mnohem volnějšího výběru použitých druhů písma. Další podstatné rozšíření se týkalo podpory Java appletů. HTML 3.2 neslo kódové označení Wilbur a od ledna 1997 bylo doporučením konsorcia W3C. Všichni by jej měli používat a tímto způsobem zajistit kompatibilitu na webu. Většina prohlížečů však již tehdy nabízela další rozšíření nad rámec verze HTML 3.2. Na jednu stranu nebyla nijak standardizována a některé prohlížeče jim nerozuměly, na stranu druhou byla mnohdy velmi užitečná a jejich používání a rozšíření urychlilo zařazení do standardu. Mezi tato rozšíření patří především rámy a klientské skriptovací jazyky (např. Javascript). Na jaře roku 1997 byly zveřejněny další plány na rozšíření HTML pod kódovým názvem Cougar. Cougar v sobě zahrnuje HTML 3.2 společně s běžně používanými konstrukcemi - rámy, skripty a obecné vkládání objektů. Dalšími novinkami se stala podpora vícejazyčných dokumentů.

Na začátku července 1997 by zveřejněn HTML 4.0, který vznikl drobnými úpravami Cougaru a vytvořením jediného komplexního dokumentu popisujícího návrh standardu. Od té doby se na dlouhou dobu samotný vývoj jazyka v podstatě zastavil. Pouze v prosinci 1999 byl vydán standard HTML 4.01, který pouze opravuje některé chyby v předchozí specifikaci.

Krátce po vydání HTML 4.0 zveřejnilo W3C ještě jeden standard – jazyk XML. Ten se od svého vydání v roce 1998 stal ve světě informačních technologií všudypřítomným formátem. XML ovlivnilo i další vývoj jazyka HTML.

Na začátku roku 2000 byla vydána specifikace jazyka XHTML 1.0, která měla svoji syntaxi odvozenou od XML. V zásadě se jednalo jen o kosmetickou změnu. Dostupné značky a jejich význam zůstal stejný, jen syntaxe prošla malými změnami. Právě pro odlišení od předchozích verzí HTML měla tato nová verze HTML založená na XML před svým názvem přidáno písmeno „X“. Na začátku tisíciletí se XHTML zdálo jako správná technologie nejen pro klasický web. Vznikly speciální zjednodušené verze XHTML. Kvůli snaze vytvářet jednodušší odnože jazyka XHTML, vzniklo XHTML 1.1, které do jazyka nepřidávalo v podstatě žádnou funkčnost, nýbrž popisovalo mechanismus modularizace celého jazyka a vytváření vlastních odvozenin. Historie však ukázala, že tato cesta nebyla pro další vývoj webu cestou nejschůdnější.

V W3C začala práce na vývoji další verze jazyka XHTML 2.0, která přinášela zajímavé vlastnosti, bohužel za cenu porušení kompatibility s předchozími verzemi. Pro výrobce prohlížečů se jednalo o příliš revoluční krok. Výrobci totiž preferovali konzervativnější přístup postupného vylepšování stávajících funkčních technologií. Všeobecná nespokojenost v řadách výrobců prohlížečů dospěla tak daleko, že v roce 2004 byla založena pracovní skupina WHATWG. Zde společně připravovala specifikaci platformy pro webové aplikace běžící v prohlížeči. Kromě rozšíření jazyka HTML specifikace zahrnovala i definice důležitých rozhraní pro využití ve skriptovacím jazyku Javascript. Rozštěpení dalšího vývoje HTML samozřejmě nebylo žádoucí. Proto v roce 2007 spojily síly W3C a WHATWG a pracovní skupina HTML vzala jako základ specifikace vytvořené ve WHATWG. Na jejich základě začalo vznikat HTML5.

HTML5 navazuje na HTML 4.01 a přidává pro vývojáře webových aplikací mnoho užitečných funkcí. Důležité je poznamenat, že v tuto dobu není HTML5 hotové. Na jeho specifikaci se neustále intenzivně pracuje. Některé její části jsou přitom již poměrně stabilní a podporuje je většina moderních prohlížečů. Naopak mnohé části HTML5 jsou zatím spíše experimentální a je otázkou, zda se časem prosadí a zůstanou ve finální verzi HTML5, jehož dokončení se neočekává před rokem 2020. [6]

1.1.3 Zdrojový text HTML

Všechny programovací jazyky mají svůj zdrojový text vytvářený programátorem, stejně jako i v HTML, který popisuje grafickou podobu stránky v prohlížeči. HTML má i svou přesnou syntaxi, zásady správného zapisování kódu. V případě, že se programátor dopustí ne zcela zásadních chyb, zobrazí se stránky v žádané podobě. Narazí-li však na zásadní chybu, element na stránce bude již špatně či nebude zobrazen vůbec. Prohlížeč totiž vše, co nezná, ignoruje a zobrazí pouze to, co umí. [1]

Veškeré příkazy v HTML jsou označovány jako značky nebo tagy. Jsou uzavřeny ve špičatých závorkách $\langle \rangle$ společně se svými parametry a hodnotami. Vše, co je mimo ně, se považuje za text, který prohlížeč zobrazí.

V HTML se používají tzv. *párové* a *nepárové* značky. Párové značky mají vliv na oblast, kterou uzavírají. První část se nachází před danými prvky, druhá se zobrazí za nimi.

Příklad:

```
<značka>  
    kód, na který mají mít značky vliv  
</značka>
```

Naopak nepárová značka má vliv pouze na sebe samu, definuje v HTML nějaký prvek, nebo má vztah k celému dokumentu.

Příklad:

```
<značka>
```

HTML je podstatně lehčí než jiné programovací jazyky a všechny principy tvorby jsou přesně dané. [2]

Programátor v jazyku HTML by měl též dbát na formátování kódu, aby bylo jasné, na jaké části bude mít část kódu vliv. Nedodržení této zásady může mít za následek nečitelnost nejen pro další osoby, které by mohly na kódu pracovat, ale časem i pro samotného tvůrce. Formátování kódu bohatě postačí odsazení textu od kraje, vynechání řádků mezi jednotlivými částmi HTML. [1] Dále se doporučuje všechno psát malými písmeny. Tato poučka se netýká pouze samotného kódu, ale i jmen souborů. Soubory budou mít vždy jméno napsané malými písmeny, nesmí obsahovat diakritiku a mezery. Tato skutečnost nám později zajistí správnou funkčnost celé práce.

Příklad:

```
<HTML>
  <HEAD>
    <TITLE>
      Jméno stránky
    </TITLE>
  </HEAD>
  <BODY>
    Část, která bude zobrazena v prohlížeči
  </BODY>
</HTML>
```



```
<HTML><HEAD><TITLE>Jméno stránky </TITLE></HEAD><BODY>Část,
která bude zobrazena v prohlížeči</BODY></HTML>
```



1.1.4 Základní značky pro tvorbu stránek

Kvůli existenci prvků (např. JavaScript, CSS, atd.), které lze implementovat do zdrojového kódu, je nutné rozlišovat část, která je HTML a část patřící ke zmiňovaným prvkům.

Na úplném začátku kódu je deklarace typu *DOCTYPE*. Tímto zápisem prohlížeči oznámíte, v jaké verzi (X)HTML jste svou stránku napsali. Prohlížeč podle pokynu s kódem naloží. [11] HTML část je uzavřena párovou značkou `<HTML></HTML>`. [4] Ta v sobě uzavírá tzv. hlavu dokumentu a tělo dokumentu. [1] Podrobněji budou tyto části probrány v dalších podkapitolách.

Základní syntaxe zdrojového kódu vypadá takto

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<HTML>                                <!-- počátek zdrojového textu HTML-->
  <HEAD>                              <!-- počátek definice hlavy dokumentu-->
    ...
    ...
  </HEAD>                             <!-- konec definice hlavy dokumentu-->

  <BODY>                              <!-- počátek definice těla dokumentu-->

    Část, která bude zobrazena v prohlížeči

  </BODY>                             <!-- konec definice hlavy dokumentu-->
</HTML>                              <!-- konec zdrojového textu HTML-->
```

Tento kód nemá žádné grafické vyjádření, neboť hlava i tělo dokumentu jsou prázdné.

1.1.4.1 Hlava dokumentu

Pod označení `<HTML>` se ihned umísťuje párová značka `<HEAD>`. Ta v kódu označuje tzv. hlavičku stránky. V této části zdrojového kódu se definují dvě důležité značky. První definuje jméno stránky, které bude zobrazeno v záhlaví prohlížeče. Jméno by mělo být výstižné a stručné. Je definováno párovou značkou `<TITLE>`. V případě, že značka chybí, prohlížeč zobrazí cestu k danému souboru HTML, a proto by jméno nemělo v kódu chybět. Druhou důležitou značkou je `<BASE>`, kterou se definuje základní adresář, jenž je brán ve vztahu k relativní cestě jako kořenový. Využívá se v případech, kdy je adresářový systém různě rozvětven. Výhodou je usnadnění práce, například při přesunu na jiný počítač stačí pouze změnit cestu nadefinovanou v této značce. Není-li definován základní adresář, je jako základní adresář určen ten, k němuž se vztahují všechny relativní definice cesty k souborům obsahujícím aktuální dokument, který je právě zobrazován prohlížečem. [1] Dále jsou zde různé tagy `<META />`. Důležitý je hlavně meta tag týkající se kódování. [12] Tato část kódu obsahuje vše, co není na stránce vidět, ale danou stránku charakterizuje.

1.1.4.2 Tělo dokumentu

Tělo dokumentu je určeno párovou značkou `<BODY>`. V této části je obsaženo vše, co má být prohlížečem zobrazeno. Značek, které se mohou v `BODY` objevit, je mnohem více než v případě hlavičky. [12]

Při tvorbě dokumentu HTML v nějakém editoru jsou tyto základní značky již předpřipravené. Kód je většinou zapsán malými písmeny a doporučuje se udržovat formátování a tvar zápisu. V této práci jsou kódy zapsány velkými písmeny, aby se kód odlišil od teoretického textu.

1.1.4.3 Poznámky

Všechno, co se vyskytuje v HTML souboru mezi značkami `<!--` a `-->`, je považováno za poznámku a nezobrazuje se. `<!--` je tedy začátek poznámky, `-->` její konec. Jako synonymum se k poznámkám užívá slov "komentáře". Poznámky by si měl ve svém kódu vytvářet každý programátor webových stránek. Velké množství kódu se může stát pro další uživatele i pro samotného tvůrce nepřehledné. [11]

Příklad:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<HTML>
  <HEAD>
    <META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=UTF-8">
    <TITLE>Jméno</TITLE> <!--Jméno stránky-->
    <BASE HREF="www.stranky.cz/data/index.html"> <!--Zákl. adresář-->
  </HEAD>

  <BODY>
    Část, která bude zobrazena v prohlížeči
  </BODY>
</HTML>
```

1.2 Prohlížeče

Internetový prohlížeč je prostředníkem stojícím mezi samotným tvůrcem stránek a uživatelem. Úkolem každého programátora webových stránek je vytvářet práci, s níž si bude každý prohlížeč bude umět poradit.

1.2.1 Internet Explorer

Internet Explorer (zkracovaný na IE) je webový prohlížeč společnosti Microsoft. Je součástí operačních systémů rodiny Microsoft Windows. IE je nejrozšířenějším internetovým prohlížečem. [4] U odborné veřejnosti v oblasti informatiky však vyvolává nekonečné diskuse. Zatím poslední vyšlou verzí IE je verze 9 z března roku 2011. Hlavními prioritami této verze jsou výkon, HW akcelerace, kompatibilita a jednoduchost. Projevilo se i zlepšení podpory nových webových standardů. Částečně zahrnuje podporu HTML5 a CSS3. Další novinkou je vypnutí zobrazování reklam nebo počítadel návštěvnosti. Kvůli zpětné kompatibilitě vůči starším verzím IE9 obsahuje také renderovací jádro IE7 a IE8. Webový prohlížeč IE (především verze 6 či starší) je často kritizován pro nedokonalé zabezpečení ochrany umožňující šíření virů, spyware či adware. [8]

Konkrétní informace na:

- www.microsoft.com/windows/internet-explorer/ - oficiální stránky
- blogs.technet.com/b/iecz/ - české stránky o Internet Explorer

1.2.2 Firefox

Firefox je multiplatformní prohlížeč s otevřeným zdrojovým kódem nebo-li open-source. V současnosti zaujímá druhé místo ve využívání. Mezi vývojáři webových stránek je oblíbený díky nejlepší podpoře webových standardů mezi prohlížeči. Mozilla Corporation, která prohlížeč vyvíjí, klade velký důraz na bezpečnost produktu. V základní instalaci Firefoxu najdete různé funkce, například brouzdání v tabech, záložky, historie, blokáce nevyžádaných vyskakovacích oken, phishing ochranu, kontrolu pravopisu, integrované vyhledávání, náhled zdrojového kódu, Java Script error konzole, DOM inspektor a mnoho dalších. Firefox je nejrozšířenějším alternativním prohlížečem. Některými uživateli je však kritizován za rychlost a náročnost. [7]

Konkrétní informace :

- www.mozilla.com/firefox - oficiální stránky
- www.mozilla.cz - české stránky o Firefoxu a Mozille vůbec

1.2.3 Google Chrome

Google Chrome je nový prohlížeč od Google. Vyznačuje se velkou rychlostí zejména javascriptového enginu. Přináší rovněž nový koncept uživatelského rozhraní. Celkově uživatelské rozhraní působí jednodušeji. Google Chrome se samozřejmě neobejde od přímého napojení na služby Googlu. [7]

Konkrétní informace :

- www.google.com/chrome - oficiální stránky

1.2.4 Opera

Opera je praktický multiplatformní prohlížeč se spoustou funkcí. Bývá označována jako nejrychlejší prohlížeč na světě s malými nároky na paměť počítače. Tuto pověst prohlížeče dokazuje úspěšné splnění testu ACID2. Jediným stínem prohlížeče je horší podpora skriptování. I v otázce bezpečnosti zaujímá světovou špičku. Statistiky vykazují, že Opera má nejméně bezpečnostních nedostatků a pokud se projevily, došlo okamžitě k jejich odstranění. Opera disponuje kromě prohlížeče kvalitním emailovým klientem M2, RSS/Atom čtečkou, IRC chatem a integrovaným editorem zdrojového kódu. Uživatelské rozhraní je plně přizpůsobitelné a skinovatelné. Operu mohou užívat pomocí hlasového ovládání i handicapovaní uživatelé. Do uživatelského rozhraní je zapracován i integrovaný vyhledávač, v němž lze snadno měnit vyhledávače či přidávat vlastní. Mnoho IT profesionálů Operu velmi doporučuje. V oblasti mobilních telefonů a chytrých zařízení je dokonce Opera nejpoužívanějším browserem. Jedním z jejích problémů je zobrazování některých stránek optimalizovaných pouze pro Internet Explorer. [7]

Konkrétní informace :

- www.opera.com - oficiální stránky
- www.operacesky.net - české stránky o Opeře
- www.opera-prehliadac.com - slovenské stránky o Opeře, Další prohlížeče

1.2.5 Safari

Prohlížeč Safari se zaměřuje pouze na operační systém Macintosh. Podpora webových standardů je v uvedeném prohlížeči na vysoké úrovni. Rychlost zobrazování stránek v Safari patří ke špičce. V prohlížeči Safari nechybějí četné funkce, například brouzdání v

záložkách, správce oblíbených položek, RSS čtečka a emailový klient. Safari dokáže zobrazit i PDF soubory. Je moderní a rychlou alternativou pro systémy Mac OS. Jeho vzhled a uživatelské rozhraní přesně zapadá do stylu systému Mac OS. Prohlížeči se dostalo ocenění za dobrou podporu standardů a rychlé zobrazovací jádro. [7]

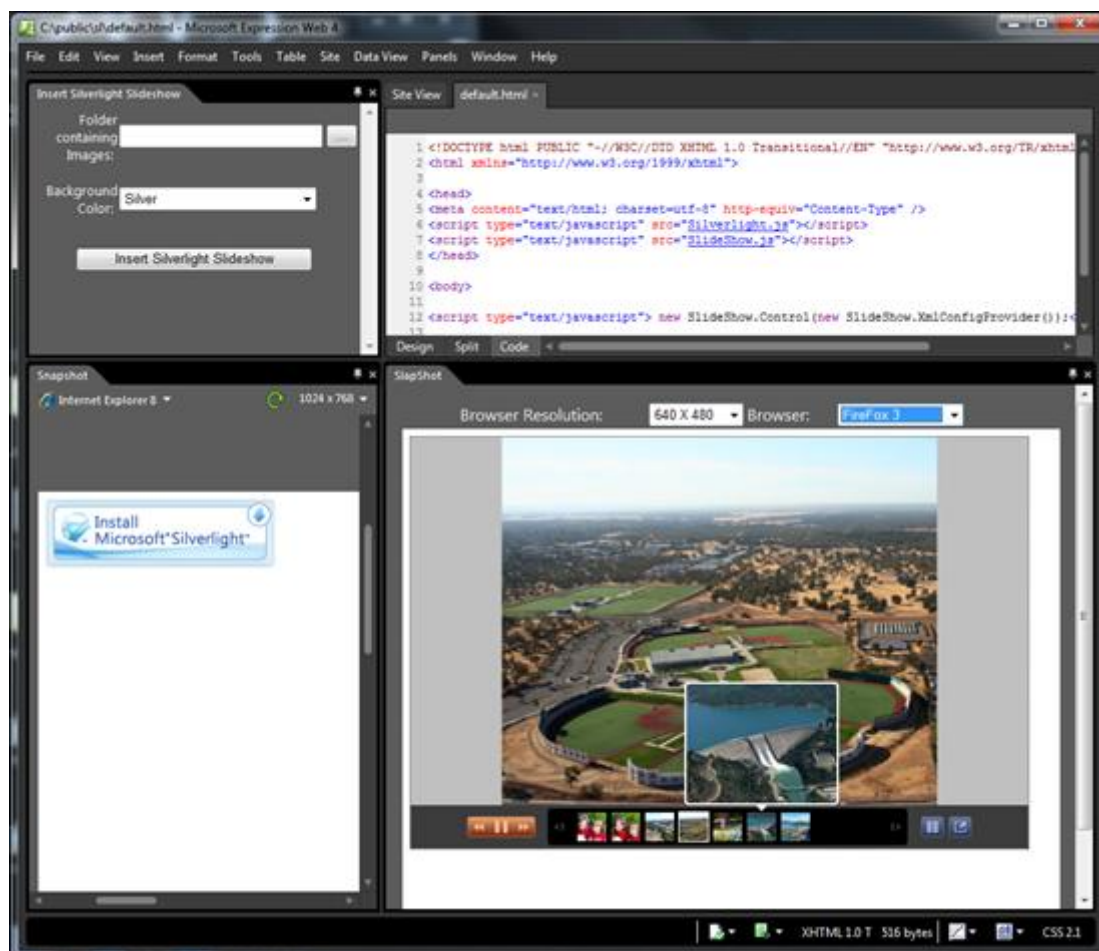
1.2.6 Další

Z dalších prohlížečů lze jmenovat například Amaya, Flock, Netscape, Lynx, Opera Mini, Opera Mobile, atd. [7]

1.3 Microsoft Expression Web 4

Microsoft Expression Web 4 je jedním z produktů řady Expression Studio 4, který se orientuje na tvorbu webových stránek. K dalším produktům této řady patří Expression Design, Expression Blend a Expression Encoder. Všechny tyto programy jsou určeny pro programátory, grafiky a 3D designéry. [9] Expression Web je nástroj pro tvorbu HTML, CSS a také trochu ASP.NET. Svým zaměřením je určen především web designerům, není tedy primárně určen pro programátory. Dalo by se říci, že se jedná o jakéhosi nástupce FrontPage z balíku MS Office, ale v mnoha věcech se zaměření obou produktů liší. Při prvním spuštění vidí uživatel produktů od Microsoft jistou podobnost s Visual Studiem. V zásadě je uživatelské rozhraní rozděleno na 3 hlavní sloupce. V pravé části programu je seznam souborů v projektu, seznam atributů HTML tagů a CSS atributů. Vlevo je umístěný "Toolbox", z něhož může uživatel přetahovat HTML tagy a ASP.NET controly. Prostor uprostřed je určen k editaci HTML. Uživatel si může vybírat mezi Design View a Code View. V Code View lze vidět surový HTML kód, Design View je ve stylu WYSIWYG. Navíc je zde hybridní SplitView, kdy uživatel vidí HTML kód i rendrovaný výsledek. V obou částech lze editovat a vidět, jaký kód WYSIWYG editor generuje, nebo opačně, jak bude vypadat kód, který byl vytvořen z předdefinovaných prvků. [10]

Pro studenty Univerzity Tomáše Bati ve Zlíně, Fakulty aplikované informatiky je po přihlášení do programu MSDN Academic Alliance k dispozici k volnému stažení Expression Studio ve verzi 3 a 4 Ultimate. Studenti mohou také získat plnou verzi tohoto programu na stránkách <https://www.dreamspark.com/default.aspx>.



Obr. 1. Prostředí programu MS Expression Web 4

1.4 Texty a barvy

1.4.1 Formátování textu

1.4.1.1 Normální text

Základním prvkem pro publikování na Internetu je text. K jeho publikování není zapotřebí používat žádné speciální značky. Za text se považuje vše, co není součástí nějaké značky uvnitř sekce `<BODY>` `</BODY>`. Základní text bez formátování může vypadat takto:

```
<BODY>
    Toto je text bez jakéhokoliv formátování.
</BODY>
```

Prohlížeč nerozlišuje, zda je použita klávesa ENTER, protože jej považuje za obyčejnou mezeru. Pro ukončení řádku má HTML speciální nepárovou značku `
`. Jejím úkolem je provést na místě, na které ji tvůrce kódu napsali, takzvané nucené odřádkování.

Prohlížeč v tomto místě odskočí na další řádek bez ohledu na velikost okna nebo rozlišení obrazovky.

Příklad:

```
<BODY>
  Toto<BR>
  je<BR>
  odřádkovaný<BR>
  text.<BR>
</BODY>
```

Nezáleží na tom, zda bude text napsaný stejným způsobem jak v předchozím příkladu nebo přímo za sebe na jeden řádek. V obou případech bude HTML kód zobrazen stejně. [1]

1.4.1.2 Odstavce a jejich zarovnání

HTML definuje dvojici užitečných značek, které se používají pro práci s odstavci. Jejich použití je podstatně jednodušší než příkladně použití značky `
`, zdrojový text je přehlednější a čitelnější.

Prvním z nich je `<DIV>`, který definuje odstavec textu. DIV je párová značka, k níž náleží ukončovaná značka `</DIV>`. Veškerý text uvnitř těchto značek je chápán jako odstavec. Do uvozující značky lze vložit parametr, určující zarovnání odstavce.

Tento parametr se jmenuje *ALIGN* a jeho syntaxe uvnitř značky je

`<DIV ALIGN = "zarovnání">`,

kde parametr "zarovnání" má hodnoty :

- LEFT – odstavec bude zarovnán doleva, toto zarovnání je nastaveno jako implicitní
- RIGHT – odstavec bude zarovnán doprava
- CENTER - odstavec bude zarovnán na střed

Veškeré hodnoty parametrů jsou zadávány v uvozovkách zejména kvůli přehlednosti, prohlížeče je však dokáže interpretovat bez uvozovek. Záleží jen na uživateli, který způsob zápisu si zvolí.

Příklad:

```
<DIV ALIGN="center">
...
</DIV>
```

Podobný význam jako <DIV> má značka <P>, která klade mezi odstavce prázdný řádek navíc. Funguje jako dvě značky
 za sebou. I v případě značky <P> se používá parametr ALIGN se stejným nastavením.

Příklad:

```
<P>První odstavec.</P> <P>Druhý odstavec.</P> <P>Třetí odstavec.</P>
```

Má-li být text umístěn na střed, nemusí být používané výše zmíněné značky s parametrem ALIGN = "CENTER". Stačí pouze daný text vložit do značky <CENTER>, která se o zarovnání sama postará. Její výhodou je vliv na veškeré prvky, které se uvnitř nacházejí. V případě, že má být nějaký text uvnitř <CENTER> </CENTER> zarovnaný na jiný okraj, nemusí se ukončovat. Stačí použít například <DIV>, přičemž po jeho ukončení se bude všechno opět centrovat. [1]

Příklad:

```
<CENTER>
  Tento text je za pomoci značky CENTER zarovnán na střed.
  <DIV ALIGN="left">
    Tato část text je však zarovnaná doleva
  </DIV>
  A zde je text opět vycentrován.
</CENTER>
```

1.4.1.3 Citace

Citace jsou zvláštním způsobem tvorby odstavce pomocí odsazeného textu. Vytváří se za pomoci párové značky <BLOCKQUOTE>. Značka se postará o odsazení odstavce, případné uvozovky však už musí doplnit sami. Tento tag má kromě obecných atributů ještě atribut cite, který udává URL dokumentu, z něž je citováno.[1]

Příklad:

```
<BLOCKQUOTE>
  <DIV>
    " Ani jeden mladý člověk není tak nešikovný, aby nedospěl k
    úspěchu aspoň v jedné věci, když je vytrvalý."
  </DIV>
  <DIV ALIGN="right">
    Leonardo Da Vinci
  </DIV>
</BLOCKQUOTE>
```

1.4.1.4 Zobrazení předformátovaného textu

Text, který je umístěn do zdrojového kódu, ztrácí při zobrazení v prohlížeči své formátování. Prohlížeč ignoruje vícenásobné mezery, odřádkování pomocí klávesy ENTER, odsazení od okraje. Má-li se text zobrazit v prohlížeči stejně jako ve zdrojovém kódu, lze využít párové značky `<PRE>`, do níž text uzavřeme. [11]

Příklad:

```
<PRE>
Předformátovaný text
    vložený do  značky P R E
                se v prohlížeči  zobrazí STEJNĚ
jako ve          zdrojovém kódu.
</PRE>
```

1.4.2 Nadpisy, písmo a velikost písma

K vytváření přehledného dokumentu nestačí pouze jedna velikost písma a žádné formátovací možnosti. Stránky by pak vypadaly jako změt' znaků a čísel. HTML nabízí několik prostředků, jak zpřehlednit text například pomocí nadpisů, různých fontů a jejich řezů, efektů a barev.

1.4.2.1 Efekty a zvýrazňování písma

Každé písmo, které bude v rámci HTML použito, může mít několik zvýrazňujících prvků. Tři základní a nejčastěji používané zvýrazňující prvky jsou :

- *B* – tučné písmo,
- *I* – kurzíva (nakloněné písmo),

- *U* – podržené písmo. [4]

Příklad:

```
<P>  
    <I>Toto je kurzíva.</I><BR> <B>Toto je tučně zvýrazněné.</B><BR>  
    <U>Toto je podržené.</U>  
</P>
```

HTML se však neomezuje pouze na tyto efekty, ale nabízí i další :

- *STRIKE* – přeškrtnutí písma,
- *BIG* – ohraničuje písmo, které bude o jeden bod větší než standardní,
- *SMALL* – ohraničuje písmo, které bude o jeden bod menší než standardní,
- *SUB* – dolní index,
- *SUP* – horní index. [1]

1.4.2.2 Fonty

V textu se není nutné omezovat a celý font se může zaměnit za jiný. Při výběru písma musí mít tvůrce kódu na paměti, že musí být nainstalované v seznamu fontů.

Font se definuje ** a jedná se o párovou značku. Značka má následující parametry :

- *SIZE* = "velikost" – určuje velikost písma, a to od 1 do 7, před hodnotu lze zadat znaménka + (plus) nebo – (mínus) a tím se změní velikost písma vzhledem k aktuálnímu tím způsobem, že se zadaná hodnota přičte nebo odečte.
- *FACE* = "písmo" – název písma, v případě použití se musí zadat přesně celý název fontu, v případě, že je zadání špatné, bude použito standardní písmo Times New Roman.
- *COLOR* = "barva" – definuje barvu písma, základní barvy lze zapsat slovním vyjádřením (red, green, blue apod.), v jiném případě se musí barva zadat hexadecimálním číslem. [4]

Příklad:

```
<FONT SIZE= 3 FACE="Arial, Helvetica" COLOR="red">  
    červené písmo s nastaveným fontem Arial o velikosti 3  
</FONT>
```

1.4.2.3 *Implicitní nastavení velikostí písma*

Značkou `<BASEFONT>` se nastavuje implicitní velikost všech písem. Pokud bude tato velikost změněna za pomoci ``, po jeho ukončení se všechna písma nastaví zpět do implicitní velikosti. Syntaxe je `<BASEFONT SIZE= "n">`, kde je velikost od 1 do 7, podobně jako u značky `FONT`. Musí se deklarovat bezprostředně za tagem `<BODY>`. [1]

1.4.2.4 *Nadpisy*

Jako nadpisy lze používat výše uvedené změny fontů, barev a velikostí, ale HTML má pro nadpisy vyhrazené speciální tagy. Celá tvorba nadpisů je pak mnohem jednodušší a praktičtější. K dispozici je šest úrovní nadpisů, v nichž je přímo nadefinovaná velikost. Tyto značky mají velikost `<H1>`, `<H2>`, ..., `<H6>`. Tag `H1` se zobrazí jako největší a plynule se zmenšuje až k `H6`. [4] Standardně jsou nadpisy zarovnané doleva, ale za pomoci parametru `ALIGN` je lze zarovnat na střed nebo doprava. Každý parametr `ALIGN` má vliv pouze na ten nadpis, v jehož tagu je umístěn. [1]

Příklad:

```
<H1>Nadpis velikosti H1</H1>  
<H3>Nadpis velikosti H3</H3>  
<H6>Nadpis velikosti H6</H6>
```

1.4.3 Barvy

1.4.3.1 *Zadávání barev*

Pro zobrazení barev na monitoru se používá systém barev RGB (Red Green Blue). Každá barva na monitoru se skládá ze tří základních barev (červené, zelené a modré) v různém poměru.

Pro zápis barvy se používá :

- Název barvy (v angličtině) – u předdefinovaných (pojmenovaných) barev. Při použití zápisu jménem barvy je validních pouze základních 16 VGA barev.

















- RGB zápis:
 - Procentuálně - intenzitu každé barvy určujeme procenty 0% (minimální intenzita) až 100% (maximální intenzita)
 - Desítkově - čísla 0 až 255
 - Šestnáctkově - 00 až FF (tj. 0-9, a-f), zápis začíná hash - markem # [4]

V kapitole Fonty bylo již řečeno, že barva textu se zadává značkou

``.

Pokud nejsou definované žádné barvy, budou použity defaultní barvy prohlížeče, což je obvykle transparentní pro pozadí, černá pro text, modrá pro odkazy, fialová pro navštívené odkazy a červená pro aktivní odkazy.

Základních 16 VGA barev [1]

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

Další barvy a jejich hexadecimální vyjádření se dá samozřejmě dohledat na Internetu.

1.4.3.2 Bezpečné barvy

Bezpečné barvy (též tzv. Netscape paleta) jsou takové barvy, které by měl podporovat jakýkoliv WWW klient na jakémkoliv operačním systému podporujícím alespoň 256 barev. Pokud jsou na stránkách použité tyto barvy, je zaručeno, že každý, kdo si stránky bude prohlížet, si je bude prohlížet v barvách, v jakých byly vytvořeny. V případě, že WWW klient nemá k dispozici definovanou barvu, snaží se ji „poskládat“ z jiných barev (tzv. dithering). Bezpečných barev je 216 a 16 odstínů šedi. Jsou to barvy, u nichž hodnota každé složky (červená, zelená a modrá) nabývá pouze hodnot 00, 33, 66, 99, CC nebo FF. [13]

1.4.3.3 Výchozí nastavení barev

Každý dokument má své výchozí nastavení zahrnující barvu pozadí, implicitní barvu textu, barvu odkazu na jinou stránku a podobně. Všechny tyto údaje se nastavují na začátku zdrojového textu ve značce BODY.

Jednotlivé parametry :

- BGCOLOR – barva použitá na pozadí
- TEXT – barva písma, standardně nastavená černá barva
- LINK – barva odkazu, standardně modrá
- VLINK – barva již navštíveného odkazu, implicitně nastavená fialová
- ALINK – barva odkazu, který jsme označili kliknutím, standardně červená [1]

Příklad:

```
<BODY BGCOLOR="blue" TEXT="white">  
    Část, která bude zobrazena v prohlížeči  
</BODY>
```

1.5 Seznamy

1.5.1 Netříděný seznam

Jedná se o výčet, kde položky nejsou závislé na pořadí, v jakém jsou uvedeny. Příkladem může být nákupní seznam či seznam úkolů.

Pro netříděný seznam má HTML k dispozici párovou značku ``. Jednotlivé položky seznamu jsou pak uvozeny tagem ``. Tato značka nemá ukončovací protějšek.

Jednoduchý seznam nákupu by v HTML jazyce vypadal asi následovně :

```
Musím nakoupit:<BR>  
<UL>  
    <LI>rajčata  
    <LI>mléko  
    <LI>jablka  
    <LI>chléb  
</UL>
```

Jakmile prohlížeč narazí na značku uvozující seznam, automaticky odskočí na další řádek a jeden řádek vynechá. Stejný případ nastane i při zakončení seznamu, kdy opět odskočí o jeden řádek. Tato funkce slouží pro celkovou přehlednost a úpravu seznamu. [1]

Pro netříděné seznamy existuje řada různých symbolů. Jako defaultní je u většiny prohlížečů nastaveno černě vyplněné kolečko, případně vyplněný kosočtverec. Má-li být použit jiný symbol, je k dipozici ke značce `` parametr *TYPE* nabývající hodnot :

- `disc` – plné kolečko, případně kosočtverec (podle prohlížeče);
- `circle` – prázdné kolečko
- `square` – čtvereček vyplněný [4]

Příklad:

```
<UL TYPE="disc">
  <LI>Seznam 1 - plné kolečko jako symbol
</UL>
<UL TYPE="circle">
  <LI>Seznam 2 - prázdné kolečko jako symbol
</UL>
<UL TYPE="square">
  <LI>Seznam 3 - čtverec jako symbol
</UL>
```

1.5.1.1 Víceúrovňové seznamy

V mnohých případech je jednoúrovňový seznam nedostačující, a proto lze v HTML vytvořit seznam víceúrovňový. Jednotlivé položky seznamu mají další podpoložky, ty pak mají další podpoložky a tak dále. Konkrétním případem může být výčet jednotlivých částí těla. Hlavními celky jsou hlava, tělo a končetiny. Každý celek je složen z dalších částí - hlava má oči, nos, ústa a uši. V ústech existují zuby a jazyk. V tomto seznamu se použilo zanoření až do třetí úrovně. Jednotlivé úrovně jsou odsazeny, aby byl výsledný seznam přehledný. [1]

Příkladem víceúrovňového seznamu mohou být jednotlivé části PC:

```
<UL> <LI>Monitor
      <LI>Myš
      <LI>Klávesnice
          <UL>
              <LI>Jednotlivé klávesy
          </UL>
      <LI>Skříň
          <UL>
              <LI>Základní deska
                  <UL>
                      <LI> Procesor
                      <LI> Paměť RAM
                      <LI> Rozšiřující karty
                  </UL>
              <LI>Zdroj
              <LI>Pevný disk
          </UL>
      </UL>
```

Prohlížeč odsazuje jednotlivé úrovně směrem doprava a přiřazuje různé odrážky v podobě puntíku, prázdného kolečka a čtverečku. Pro další úrovně se používají pouze jen čtverečky.

1.5.2 Tříděný seznam

Tříděný seznam je přesným opakem seznamu netříděného. Záleží na pořadí položek, které nelze libovolně zaměňovat. Příkladem tříděného seznamu může být návod k použití nebo kuchařský recept. Vždy je nutné dodržet postup práce. Uvedený seznam se může vytvořit poměrně jednoduše. Zamění se pouze značka používaná v netříděném seznamu za značku . Jednotlivé položky jsou číslovány jako v případě kuchařského receptu.

Příkladem může být návod přípravy řízku. Kód v HTML bude vypadat takto:

```
Jak připravit řízek:<BR>
<OL>
  <LI>Připravíme si rozklepané, osolené maso, hladkou mouku, vejce a
  strouhanku.
  <LI>Maso obalíme nejdříve v mouce, protáhneme v rozšlehaném vejci a
  nakonec obalíme v prosáté strouhance.
  <LI>Na plotně rozpálíme v pánvi stolní olej na potřebnou teplotu.
  <LI>Řízek vložíme do tuku a po obou stranách usmažíme dozlatova.
  <LI>Podáváme řízky ke konzumaci.
</OL>
```

V HTML však není omezení pouze na číslovaný seznam. Mohou se používat velká i malá písmena, arabské i římské číslice. K změně tohoto parametru se používá *TYPE*, jehož syntaxe je *TYPE=n*, kde *n* určuje typ odrážek.

Hodnotu parametru může nastavit takto :

- TYPE= a – malá písmena abecedy
- TYPE= A – velká písmena abecedy
- TYPE= i, ii, iii, iv,... – malé římské číslice
- TYPE= I, II, III, IV,... – velké římské číslice
- TYPE= 1 – klasické číslování 1, 2, 3,... [4]

1.5.3 Seznam definic

Posledním z druhů seznamů je definiční výčet, který se používá hlavně při vysvětlování jednotlivých pojmů. Pro vytvoření se uvádějí značky *<DL>*, *<DT>* a *<DD>*, jejichž použití je velmi podobné jako u tříděného a netříděného seznamu. [4]

```
<DL>
  <DT> Pojem 1
    <DD> vysvětlení, definice pojmu 1
  <DT> Pojem2
    <DD> vysvětlení, definice pojmu 2
</DL>
```

U seznamu se pro úpravu používají formátovací značky, kterými si můžeme text zpřehlednit. Mezi používané značky se řadí příkladně B, I, U nebo FONT.

1.6 Tabulky

1.6.1 Tvorba tabulek

V HTML existují tři základní tagy pro tvorbu tabulek :

- TABLE – základní značka ve zdrojovém kódu, určuje, odkud pokud bude definovaná tabulka
- TR – definuje jednotlivé řádky a jejich vlastnosti
- TD – nastavuje obsah jednotlivých políček tabulky

Z těchto značek lze vytvořit jednoduchou tabulku. [2]

Příklad:

```
<TABLE><!--Začátek tabulky-->
  <TR><!--První řádek-->
    <TD>Sloupec 1, řádek 1</TD>
    <TD>Sloupec 2, řádek 1</TD>
  </TR>
  <TR><!--Druhý řádek-->
    <TD>Sloupec 1, řádek 2</TD>
    <TD>Sloupec 2, řádek 2</TD>
  </TR>
</TABLE><!--Konec tabulky-->
```

Samotný kód pro vytvoření tabulky se může zdát komplikovaný, a proto se často doplňuje o komentáře vysvětlující jednotlivé kroky.

1.6.2 Vlastnosti tabulky

1.6.2.1 Hlavička tabulky

Hlavička tabulky se tvoří značkou `<TH>`, která umožňuje hlavičku tabulky definovat. Hlavička definuje úvodní řádek nebo sloupec určující, co dané položky znamenají. Výhodou této značky je skutečnost, že automaticky odliší hlavičky od ostatního textu tučností písma a zarovnáním na střed. [1]

Příklad:

```
<TABLE BORDER="1">
  <TR>
    <TH>Hlavička 1</TH>
    <TD>text</TD>
  </TR>
  <TR>
    <TH>Hlavička 2</TH>
    <TD>text</TD>
  </TR>
</TABLE>
```

Tento příklad má nadefinovány hlavičky v řádku.

1.6.2.2 *Nadpis tabulky*

Další atributem, který se může v tabulce definovat, je nadpis. Slouží zejména k upřesnění obsahu tabulky. Nadpis se v kódu zapisuje párovou značkou `<CAPTION>`, která umístí nad či pod tabulku text standardně zarovnaný na střed. Text není nijak ohraničen. Tag musí být uvnitř sekce `<TABLE></TABLE>`, ovšem vně části `<TR></TR>`.

Nadpis může být zarovnaný i jinak než na střed. K tomu slouží parametry :

- ALIGN – určuje horizontální zarovnání, konkrétně LEFT (vlevo), CENTER(střed) a RIGHT (vpravo),
- VALIGN – určuje vertikální zarovnání, konkrétně TOP (nad tabulkou), BOTTOM (pod tabulkou). [1]

Příklad:

```
<TABLE BORDER="1"><!--Začátek tabulky-->
  <CAPTION ALIGN="right">Nadpis tabulky(vpravo)</CAPTION>
  <TR>
    <TD>Sloupec 1, řádek 1</TD>
    <TD>Sloupec 2, řádek 1</TD>
  </TR>
  <TR>
    <TD>Sloupec 1, řádek 2</TD>
    <TD>Sloupec 2, řádek 2</TD>
  </TR>
</TABLE><!--Konec tabulky-->
```

```

<TABLE BORDER="1"><!--Začátek tabulky-->
  <CAPTION VALIGN="bottom">Nadpis tabulky (dole)</CAPTION>
  <TR>
    <TD>Sloupec 1, řádek 1</TD>
    <TD>Sloupec 2, řádek 1</TD>
  </TR>
  <TR>
    <TD>Sloupec 1, řádek 2</TD>
    <TD>Sloupec 2, řádek 2</TD>
  </TR>
</TABLE><!--Konec tabulky-->

```

1.6.3 Další nastavení tabulky

1.6.3.1 Okraj tabulky

Tabulka nemá bez nadefinování žádný rámeček či jakékoliv jiné ohraničení. Tabulka je proto nepřehledná, není jasné, kde která buňka končí a kde začíná. Proto se doporučuje nadefinovat ohraničení tabulky. Provádí se parametrem *BORDER* značky *TABLE*, *<TABLE BORDER="n">*. Hodnota *n* je šířka okraje v bodech. [2]

1.6.3.2 Zarovnání tabulky

Jako v případě obrázku či odstavce i tabulku lze zarovnat vlevo, na střed a vpravo. Parametr určující zarovnání je stejný - *ALIGN="zarovnání"*, kde zarovnání bude *RIGHT* (vpravo), *LEFT* (vlevo) a *CENTER* (střed). [2]

Příklad:

```

<TABLE BORDER="1"><!--Začátek tabulky-->
  ...
</TABLE><!--Konec tabulky-->

<TABLE BORDER="1" ALIGN="center"><!--Začátek tabulky-->
  ...
</TABLE><!--Konec tabulky-->

<TABLE BORDER="1" ALIGN="right"><!--Začátek tabulky-->
  ...
</TABLE><!--Konec tabulky-->

```

1.6.3.3 Pozadí a barvy

Barva pozadí tabulky se definuje parametrem *BGCOLOR*="barva". Hodnota barvy se zadává buď slovně nebo hexadecimálním vyjádřením, stejně jak bylo popsáno v kapitole Zadávání barev. V případě, že bude na pozadí umístěn obrázek, budou pro něj platit shodná pravidla jako pro obrázek umístěný na pozadí stránky. Obrázek umístíme na pozadí parametrem *BACKGROUND*="obrázek". Parametr *BACKGROUND* má přednost před *BGCOLOR*. Pokud by byly zadány oba parametry, objeví se obrázek. [2]

HTML dále umožňuje zadávat barvu rámečku a jednotlivým buňkám. Tato úprava navozuje dojem prostorovosti. Jednotlivé „stínovací“ barvy se definují parametry :

- *BORDERCOLORLIGHT* – určuje barvu strany, která je světlejší (implicitně světle šedá)
- *BORDERCOLORDARK* – určuje barvu strany, která je tmavší (implicitně tmavě šedá)

Oba parametry ovlivňují nejen rámeček kolem tabulky, ale i kolem jednotlivých buněk.

Příklad:

Okraj tabulky bude nastaven na velikost 4 body a vystínován modrou barvou

```
<TABLE BORDER="4" BORDERCOLORLIGHT="0000FF" BORDERCOLORDARK="000088">
```

Tabulku však lze vytvořit i bez efektu stínování, a to parametrem *BORDERCOLOR*="barva". [1]

1.6.3.4 Vzdálenost buněk a textu

Prohlížeč přizpůsobuje velikost buňky obsahovaného textu. Buňky jsou umístěny těsně vedle sebe, což nepůsobí estetickým dojmem. Aby byla tabulka přehlednější, lze vložit mezi jednotlivá políčka tabulky, text i okraj buňky mezery.

První možností je parametr *CELLSPACING*="mezera". Mezera je hodnota mezi políčky (vertikálně i horizontálně) v bodech.

Příklad:

```
<TABLE BORDER="4" CELLSPACING="15">
```

Příbuzným parametrem, který určuje odstup textu od vnitřního okraje, je *CELLPADDING*="mezera". Mezera je opět udávána v bodech.

Příklad:

```
<TABLE BORDER="4" CELLPADDING="15">
```

Oba parametry samozřejmě lze kombinovat a oddálit políčka tabulky i text od okrajů. [2]

Příklad:

```
<TABLE BORDER="4" CELLSPACING="10" CELLPADDING="15">
```

1.6.3.5 Ohraničení

Ohraničení bylo zmíněno již v předchozích kapitolách, ale zde se jedná o zobrazování jednotlivých částí tabulky a jednotlivých buněk. Parametry umožňují skrýt nebo ukazovat ohraničení. Kombinováním lze dosáhnout zobrazení tabulky mnoha způsoby.

Parametrem, který manipuluje se zobrazováním rámečku tabulky, je *FRAME* nabývající těchto hodnot :

- VOID – odstraní vnější rámeček,
- ABOVE – zobrazí jen rámeček nad tabulkou,
- BELOW – zobrazí jen rámeček pod tabulkou,
- LHS – zobrazí rámeček na levé straně tabulky,
- RHS – zobrazí rámeček na pravé straně tabulky,
- HSIDES – rámeček je vyobrazen nad a pod tabulkou,
- VSIDES – rámeček je vyobrazen na levé a pravé straně tabulky. [4]

Příklad:

```
<TABLE BORDER="1" FRAME="void">
  <CAPTION>Tabulka s nastaveným orámováním</CAPTION>
  <TR>
    <TH WIDTH="75">Řádek A</TH>
    <TD>A1</TD>
  </TR>
  <TR>
    <TH>Řádek B</TH>
    <TD>B1</TD>
  </TR>
</TABLE>
```

Dalším parametrem pro úpravu rámečku kolem buňky je *RULES*, lze mu přiřadit tyto hodnoty :

- NONE – všechny okraje budou odstraněny,
- ALL – budou zobrazeny všechny okraje,
- COLS – budou zobrazeny všechny okraje mezi sloupcečky tabulky,
- ROWS – budou zobrazeny všechny okraje mezi řádky tabulky. [4]

Příklad:

```
<TABLE BORDER="1" RULES="cols">
  <CAPTION>COLS</CAPTION>
  <TR>
    <TD>Sloupec A</TD>
    <TD>Sloupec B</TD>
  </TR>
  <TR>
    <TD>A1</TD>
    <TD>B1</TD>
  </TR>
</TABLE>
```

1.6.4 Další parametry

1.6.4.1 Tvorba řádku tabulky

Značka pro vytváření řádků <TR> má s <TABLE> shodné tyto parametry :

- BORDERCOLOR
- BORDERCOLORLIGHT
- BORDERCOLORDARK
- BGCOLOR

Pro všechny parametry platí stejná pravidla pro zadávání barev. [1]

Příklad:

```
<TABLE BORDER="2">
  <TR BGCOLOR="red">
    <TH>Sloupec A</TH>
    <TH>Sloupec B</TH>
    <TH>Sloupec C</TH>
  </TR>
  <TR BGCOLOR="green">
    <TD>A1</TD>
    <TD>B1</TD>
    <TD>C1</TD>
  </TR>
  <TR BGCOLOR="blue">
    <TD>A2</TD>
    <TD>B2</TD>
    <TD>C2</TD>
  </TR>
</TABLE>
```

1.6.5 Buňky

1.6.5.1 Zarovnání textu v buňkách

Zarovnání bylo již několikrát zmíněno, je mu tudíž věnována pouze část informativního prostoru. Stejně jako v předchozích případech je text zarovnáván ve vodorovném a svislém směru.

ALIGN – zarovnání ve vodorovném směru

- LEFT,
- RIGHT,
- CENTER.

VALIGN – zarovnání ve svislém směru

- TOP – zarovnání textu k levému hornímu okraji,
- BOTTOM – zarovnání textu k dolnímu okraji,
- MIDDLE – zarovnání na střed,
- BASELINE – zarovnání na stejnou základnu celého řádku. [2]

Značka `<TR>` má parametr ovlivňující formátování textu. Konkrétně se jedná o zákaz zalomení textu *NOWRAP*. [1]

1.6.5.2 Spojené buňky

HTML umožňuje spojit sousedící políčka v jednu buňku v řádku i ve sloupci. Tím vznikají tak zvané spojené buňky. Na spojování buněk jsou v HTML k dispozici 2 parametry.

Prvním parametrem je *ROWSPAN*=“*n*“. Hodnota *n* určuje počet řádků, přes které bude políčko „roztazeno“

Příklad:

```
<TABLE BORDER="2">
  <TR>
    <TD ROWSPAN="2">Spojené buňky</TD>
    <TD WIDTH="50">A</TD>
  </TR>
  <TR>
    <TD>B</TD>
  </TR>
</TABLE>
```

Na podobném principu funguje i druhý parametr pro spojování přes více řádků *COLSPAN*=“*n*“. [1]

Příklad:

```
<TABLE BORDER="2">
  <TR>
    <TD COLSPAN="2">Spojené buňky</TD>
  </TR>
  <TR>
    <TD>A</TD>
    <TD>B</TD>
  </TR>
</TABLE>
```

1.6.5.3 Šířka a výška buňky

Nastavení šířky buňky má stejný parametr jako nastavení šířky celé tabulky, tedy *WIDTH*=“*šířka*“. Šířku zadáváme v obrazových bodech nebo procentuálním vyjádření. Pokud však bude text delší než nastavená hodnota v parametru, bude parametr *WIDTH*

ignorován a velikost bude přizpůsobena textu. Je-li nastavena šířka tabulky, musí se tomuto faktu šířka buňky přizpůsobit. Šířka tabulky je pevně dána a nedá se změnit. Nebude-li šířka tabulky zadána, šířka buňky se nastaví na zadanou hodnotu. Rozšíření jedné buňky ovlivní všechny další buňky v sloupci na stejnou šířku. Pokud nastavíme několik hodnot šířky buňky, bude sloupec nastaven na hodnotu nejvyšší.

Příklad:

```
<TABLE BORDER="1">
  <TR>
    <TD WIDTH="75">Sloupec A</TD>
    <TD WIDTH="150">Sloupec B</TD>
  </TR>
  <TR>
    <TD>A1</TD>
    <TD>B1</TD>
  </TR>
</TABLE>
```

Podobné nastavení má i parametr výška *HEIGHT*="výška". Výška je opět zadána v obrazových bodech, ale neuvádí se již v procentech. Nejvyšší nastavená výška buňky ovlivní výšku celého řádku. [2]

Příklad:

```
<TABLE BORDER="1">
  <TR>
    <TH HEIGHT="50">Sloupec A</TH>
    <TH HEIGHT="75">Sloupec B</TH>
  </TR>
  <TR>
    <TD>A1</TD>
    <TD>B1</TD>
  </TR>
</TABLE>
```

1.7 Obrázky

1.7.1 Obrázky na webových stránkách

Obrázky jsou nedílnou součástí každé stránky. Díky nim jsou stránky barevnější, zajímavější a v neposlední řadě i přehlednější. Je potřeba však dávat pozor na jejich

velikost, protože žádný z uživatelů nebude ochoten dlouho čekat na načtení obrázku a raději se poohlédne po jiném zdroji. Pokud bude na stránkách umístěný obrázek, bude určitě komprimován na snesitelnou velikost.

1.7.1.1 Používané formáty obrázků

Obrázek je sám o sobě soubor. Na Internetu jsou nejčastěji vyžívané dva grafické formáty :

- JPG (JPEG) je grafický formát, který vždy používá ztrátovou kompresi znamenající, že část dat je vypuštěna. Je založena na skutečnosti, že ne všechno umí lidské oko zachytit. Jsou-li data z obrázku vypuštěna, člověk zpravidla nic nepozná. [1] Tento formát je vhodný pro ukládání fotografií a obrázků s velkou barevnou hloubkou. Neumožňuje tvořit průhledné nebo animované obrázky. Ukládá se do souborů s příponou .jpg nebo jpeg. [11]
- GIF také používá kompresi, ale neztrátovou, takže soubory jsou sice relativně větší než JPG, ale zato jsou přesné (zachovávají barvy a kontury). Barevná hloubka u tohoto formátu může mít pouze 256 barev. Jedna z barev se může označit jako průhledná (tu prohlížeč potom nezobrazí), též podporují animace. [11]

Další z formátů, který lze zmínit k této kapitole, je formát PNG (Portable Network Graphics). Přípona souboru je .png. Tento formát by měl obsahovat výhody gifu i jpegu. V současné době lze PNG bez obav používat.[11]

1.7.1.2 Datový objem

Obrázky se nesmí stahovat dlouho, měly by být co nejmenší. Proto jsou GIF i JPG formáty kompresní. Na web nedáváme obrázek větší než je 100 kB. Akceptovatelné limity se během let mění, ale stále platí, že čím jsou datově menší, tím lépe. [11]

1.7.2 Vkládání obrázku

HTML využívá pro vkládání obrázku na stránku nepárovou značku ``. Základní syntaxe vypadá takto ``. Definuje soubor obsahující obrázek, který se má na stránku umístit. V této značce musí být nadefinován parametr `SRC`. Odkazuje na místo, kde se soubor obrázku vyskytuje.

Umístění souboru lze zadat pomocí :

- absolutní cesty – jedná se o úplné zadání cesty k objektu, příkladem může být třeba URL adresa, kterou někdo někomu posílá `http://www.stranky.cz/index.html`,
- relativní cesty – záleží na místě, ze kterého je cílový soubor hledán, určuje cestu mezi adresáři, která vede z daného místa k požadovanému souboru, k jednomu souboru může vést pokaždé jiná relativní cesta, podle toho, kde se nachází stránka, z níž je na soubor odkazováno.

Při psaní webu se používají relativní cesty, protože se nejčastěji bude odkazovat na soubory v jeho rámci. Pokud je obrázek ve stejné složce jako soubor HTML, stačí zadat pouze název souboru. [12]

Příklad:

```
<BODY>
    Ukázka vkládání obrázků - ve stejné složce, relativní a absolutní
    adresování.<BR>
    <IMG SRC="obrazek.gif">
    <IMG SRC="../data/obrazky/obrazek.gif">
    <IMG SRC="c:/moje_stranky/data/obrazky/obrazek.gif">
</BODY>
```

1.7.3 Zarovnání obrázku

Obrázky na stránce lze zarovnávat vzhledem k okolnímu textu, bohužel pouze s omezením a nemá takové možnosti, jako například textový editor. Základní zarovnání se provádí za pomoci parametru *ALIGN*. Syntaxe vypadá následně :

```
<IMG SRC="obrazek.gif" ALIGN="zarovnání">
```

Jednotlivé hodnoty parametru *ALIGN* jsou následující :

- *RIGHT* – umístění vlevo, text obtéká obrázek z pravé strany
- *LEFT* – umístění vpravo, text obtéká obrázek z levé strany
- *TEXTTOP* – horní okraj obrázku je zarovnán s nejvyšším písmenem v řádku
- *TOP* – horní kraj je zarovnán s nejvyšším prvkem v řádku
- *MIDDLE* – linka, na níž je usazen text, je zarovnaná na střed obrázku
- *ABSMIDDLE* – prostředek řádku je zarovnán na střed obrázku
- *BOTTOM* – spodní okraj obrázku je zarovnaný s linkou, na níž je text

- ABSBOTTOM – zarovná spodní okraj obrázku s nejspodnější písmenem v textu, například j, p ,y a podobně.

Standardní nastavení je zarovnání BOTTOM. Většinou se používá LEFT nebo RIGHT, aby text obrázky hezky obtékal. [1]

Příklad:

```
<BODY>

<P>
<IMG SRC="obrazek.gif" ALIGN="LEFT">
Obrázek, který je zarovnaný doleva a text jej obtéká z pravé strany
po celé jeho výšce. Obrázek umíme umístit tam, kde bude začínat
obtékání, ve většině případů na začátek odstavce. Jakmile text
dosáhne spodního okraje obrázku, pokračuje od levého okraje.

</P>

<P>
<IMG SRC="obrazek.gif" ALIGN="RIGHT">
Obrázek, který je zarovnaný doprava a text jej obtéká z levé strany
po celé jeho výšce. Opět platí totéž jak v předchozím případě
zarovnání doleva. Dosáhne-li text spodní části okraje obrázku,
pokračuje pod obrázkem až k pravé části okna prohlížeče.

</P>

</BODY>
```

Všechna ostatní zarovnání fungují na stejném principu .

Hodnoty parametru *ALIGN* se nedají nastavit, aby byl obrázek vycentrován na střed stránky. Tento problém se dá řešit za pomoci tagu <CENTER>, který byl již probrán v kapitole Odstavce a jejich zarovnání.

```
<CENTER>

  Na stránce je umístěn obrázek. Obrázek i text jsou zarovnány na
  střed.

  <IMG SRC="obrazek.gif">

</CENTER>
```

1.7.4 Velikost obrázků

Další možností, jak upravovat obrázky, je nastavení jejich velikosti. Konkrétně se jedná o šířku a výšku obrázku. Pokud je na stránku obrázek vložen bez jakéhokoliv nastavení velikosti, zobrazí se přesně ve svých reálných rozměrech. Někdy je však zapotřebí zobrazit obrázek větší nebo menší než je opravdová velikost.

Pro nastavení obrázku se využívají parametry značky

WIDTH = "šířka"

HEIGHT = "výška"

Jsou povoleny jednotky délky a procenta. Při použití procent se velikost obrázků mění s velikostí prohlížeče. [4]

Příklad:

```
<IMG SRC="obrazek.gif" WIDTH="50" HEIGHT="150">
```

Výše uvedený kód zobrazí obrázek o velikosti 50x150 (šířka x výška) bodů bez ohledu, jaké skutečné rozlišení má. Při změně velikosti stačí zadat pouze jeden parametr a zbývající rozměr obrázku se již dopočítá.

Příklad:

```
<IMG SRC="obrazek.gif" WIDTH="50">
```

Pokud byla původní velikost změněna ze 75 na 50 bodů, což je 2/3, i druhý parametr, tedy výška, se musí zmenšit v poměru 2/3. [1]

1.7.5 Popiska

HTML umožňuje přiřadit každému z obrázků jeho vlastní popisku. Většinou si lze popisku představit jako bublinu, která se objeví, když najedeme kurzorem myši na obrázek. Nejedná se o jedinou funkci popisky. Popiska se objeví při načítání obrázku ve vyhrazeném rámečku nebo se ukáže na místě obrázku, pokud je vypnuto zobrazování obrázku.

Parametr popisky má syntaxi *ALT*="popiska". [1] Celý kód pro vložení obrázku s popiskou vypadá potom takto :

```
<IMG SRC="obrazek.gif" ALT="Popiska">
```

1.7.6 Rámeček

U obrázku se může nastavit rámeček různé velikosti. Rámeček se definuje parametrem *BORDER*="n". Hodnota *n* parametru reprezentuje šířku rámečku v bodech. Barva rámečku je implicitně nastavená na černou. [2] Výjimkou je případ, kdy je obrázek zároveň odkazem. Barva rámečku je pak modrá, velikost dva body. Není-li parametr *BORDER*

uveden, rámeček se nezobrazí. Stejný případ nastane, když do hodnoty parametru zadáme nulu. [1]

Příklad:

```
<IMG SRC="obrazek.gif" ALT="Bez okraje" BORDER="0">  
<IMG SRC="obrazek.gif" ALT="S okrajem" BORDER="5">
```

1.7.7 Volné okraje

Z části věnované zarovnání obrázku je patrné, že text, který obtéká obrázek, je v jeho těsné blízkosti. Tento efekt nepůsobí velmi esteticky, a proto HTML nabízí dva parametry, kterými tomu nedostatku lze čelit. Konkrétně nastavují volný prostor mezi obrázkem a dalšími prvky v jeho okolí.

HSPACE="n" – volný prostor z pravé a levé strany obrázku v bodech

VSPACE="n" - volný prostor nad a pod obrázkem v bodech

Příklad:

```
<BODY>  
  <P>  
    <IMG SRC="obrazek.gif" ALIGN="left" HSPACE="15">  
    Tento obrázek je zarovnaný doleva a má nastavený volný prostor  
    zprava i zleva o velikosti 15 bodů.  
  </P>  
  <P>  
    <IMG SRC="obrazek.gif" ALIGN="left" HSPACE="25" VSPACE="25">  
    Tento obrázek je zarovnaný doleva a má nastavený volný prostor ze  
    všech stran o velikosti 25 bodů.  
  </P>  
</BODY>
```

Definování volného prostoru odsadilo i samotný obrázek od okraje. [1]

1.7.8 Obrázky na pozadí stránky

Obrázky se mohou na stránku umísťovat přímo jako její pozadí. Použití obrázku nadefinujeme přímo ve značce <BODY> parametrem *BACKGROUND*="obrazek".

Obrázek na pozadí je soubor včetně cesty, která může být absolutní i relativní, dokonce obrázek z jiného serveru. Zápis je totožný jak u značky .

S pozadím stránky souvisí i efekt, kdy obrázek stojí, ale obsah se roluje podle provádění záměru uživatele. Standardně se obrázek pohybuje s obsahem. Kýženého efektu dosáhneme za pomoci parametru *BGPROPERTIES*=*“FIXED”*, který umístíme do tagu `<BODY>`. [3]

Před použitím obrázku jako pozadí je vhodné využití několika dobře míněných rad:

- Obrázky se nevybírají velké, protože by zbytečně protahovaly dobu stahování a zatěžovaly paměť.
- Obrázky by se neměly „tlouci“ s textem. Obsah stránky by pak mohl být nečitelný a stránka velmi špatně použitelná. Za ideální se považují obrázky v šedé paletě.
- Je dobré používat bezešvé obrázky, které na sebe navazují, aby pozadí vypadalo jako jedna plocha. Prohlížeč skládá obrázky vedle sebe a jakmile dojde na konec řádku, pokračuje opět zleva.
- Vždy je správné stránku prohlédnout a vyzkoušet s jiným rozlišením. [1]

1.8 Odkazy

1.8.1 Tvorba odkazů

K vytvoření odkazů se používají párové značky `<A HREF>`, kde *A* je vlastní tag a *HREF* parametr. Tento parametr odkazuje na místo, ve kterém se soubor vyskytuje. Umístění se zadává pomocí absolutní nebo relativní cesty (viz kapitola Vkládání obrázku). [4] Na úvod se bude téma věnovat pouze odkazu na HTML souboru, který je uložen ve stejné složce.

Příkaz bude vypadat takto :

```
<A HREF="stranka.html">Při kliknutí, budete zobrazená další stránka</A>
```

Kdyby se mezi značkami nevyskytoval žádný text, jednalo by se o tak zvaný slepý odkaz. Odkaz existuje, ale v prohlížeči by se nic nezobrazilo a nebylo by nač kliknout.

Odkaz bude mít v prohlížeči modrou barvu a bude podtržený. Barvu odkazu však můžeme definovat v sekci BODY. Odkazy mohou mít různé tvary. Ve výše uvedeném příkladu je odkazem celá věta. Odkazem může být i jedno či více slov v odstavci textu.

Příklad:

```
Toto je má <A HREF="oblibena_stranka.html"> oblíbená stránka</A>.
Navštěvovala jsem <A HREF="stredni.html">střední školu</A> v Olomouci a
nyní studuji na <A HREF="vysoka.html">vysoké škole</A> ve Zlíně. Mám
svou vlastní stránku. <A HREF="vlastni.html">Zde</A> je odkaz.
```

Odkaz může být tvořen i obrázkem nebo kombinací obrázku a textu. Pokud má být obrázek použitý jako odkaz, bude zdrojový kód vypadat takto:

```
<A HREF="stranka.html">
    <IMG SRC="obrazek.gif">
</A>
```

Obrázek se v prohlížeči zobrazí s modrým rámečkem. V kapitole Obrázky byly již zmíněny tagy a parametry, jimiž se dá obrázek vkládat a upravovat. [1]

1.8.2 Odkaz na soubor

Odkaz na soubor může být definován několika způsoby. Soubor, většinou HTML, na který má odkaz ukazovat, je definován stejným způsobem jako v úvodu kapitoly.

Jako hodnota parametru HREF může být použita i stránka z úplně jiného serveru umístěného kdekoli na světě. V tomto případě je nutné zadat internetovou adresu (URL)

```
<A HREF="http://www.stranka.cz/data/stranky/stranka.html">
```

Má-li se v prohlížeči zobrazit pouze úvodní stránka, postačí zadat jméno internetového serveru :

```
<A HREF="http://www.stranka.cz">
```

Posledním typem odkazu je vše, co není HTML stránka, tedy tzv. binární soubor. Jedná se hlavně o archivní soubor, video, hudbu, textový dokument a podobně.

```
<A HREF="dokument.doc">
```

Prohlížeč nabídne výběr ze dvou možností. Buď se soubor stáhne na disk, nebo se přímo spustí. K tomu je zapotřebí vlastnit aplikaci, jež se souborem umí pracovat. [1]

1.8.3 Odkaz na jiné místo dokumentu

V některých případech není vhodné text kvůli jeho vlastnostem (sled informací, návaznost) rozdělovat na více stránek. Text však může být takového rozsahu, že se stane pro čtenáře nepřehledným a lehce se v něm ztratí. V tomto případě je výhodné využít odkaz na jiné místo v dokumentu. Na jeho začátku lze vytvořit obsah složený z odkazů, který v prohlížeči na dané místo odskočí.

Odkaz na jiné místo stránky se definuje podobně jako normální odkaz s tím rozdílem, že se před něj přidá znak # (zahrádka) :

```
<A HREF="#kapitola">Název kapitoly</A>
```

Ve zdrojovém kódu se pak musí objevit zarážka ve tvaru :

```
<A NAME="#kapitola"></A>
```

Prohlížeč ji ve zdrojovém kódu najde a o od ní stránku zobrazí. V případě, že bude mezi značkami <A NAME> nějaký text, bude považován za normální text. Většinou se však mezi značky žádný text nevsunuje, slouží pouze jako zarážka.

Konkrétním příkladem může být třeba zahradnická příručka:

1. Jak zasadit stromek
2. Jak jej zalévat
3. Jak se o něj starat a ošetřovat
4. Jak sklízet plody

Zdrojový kód bude vypadat takto:

```
<BODY>

  <A HREF="#kapitola1">Jak zasadit stromek</A><BR>
  <A HREF="#kapitola2">Jak jej zalévat</A><BR>
  <A HREF="#kapitola3">Jak se o něj starat a ošetřovat</A><BR>
  <A HREF="#kapitola4">Jak sklízet plody</A><BR>

  <A NAME="#kapitola1"></A>
  <P> Jak zasadit stromek </P>
  ...
  <A NAME="#kapitola2"></A>
  <P>Jak jej zalévat</P>
  ...
  <A NAME="#kapitola3"></A>
  <P>Jak se o něj starat a ošetřovat</P>
  ...
  <A NAME="#kapitola4"></A>
  <P>Jak sklízet plody</P>
  ...

</BODY>
```

Existuje tzv. kombinovaný odkaz, který odkazuje na externí stránku, třeba od poloviny.

Příklad:

```
<A HREF="http://www.stranka.cz/index.html#kapitola1">Kombinovaný  
odkaz</A>
```

Samozřejmě, že index.html musí mít někde ve svém kódu definovanou zarážku. [1]

1.8.4 Elektronická pošta

Většina firem, které mají svou prezentaci v podobě WWW stránek na Internetu, nabízí možnost kontaktu. Nejčastějším kontaktem je samozřejmě e-mail. Za pomoci HTML si vytvoří odkaz, který již ve svém kódu obsahuje e-mailovou adresu předanou výchozímu klientu elektronické pošty. [1] Obecná syntaxe odkazu vypadá potom takto :

```
<A HREF=mailto:e-mail">Text</A>
```

Konkrétním příkladem pak je tento odkaz

```
<A HREF="mailto:jmeno@server.cz">Kontakt - napište nám</A>
```

V tomto případě je elektronická adresa jmeno@server.cz

1.8.5 Další možnosti odkazů

1.8.5.1 Komentář odkazu

Stejně jako u obrázku se i v případě odkazů může vytvořit popiska, která se objeví při posunutí kurzoru myši na odkaz. Komentář se zadává pomocí parametru *TITLE*="komentář". Odkaz bude v prohlížeči vypadat stále stejně, jakoby tento parametr chyběl. [4]

Příklad:

```
<A HREF="stranka.html">Odkaz</A> bez komentáře
```

```
<A HREF="stranka.html" TITLE="Komentář k odkazu">Odkaz</A> s komentářem
```

1.8.5.2 Otevírání nové stránky

Při procházení stránek na Internetu se po kliknutí odkaz častokrát objeví v původním okně. Uživatel však může mít otevřeny obě strany najednou. To znamená, že se v prohlížeči vytvoří nové okno. Tento efekt nám umožní parametr *TARGET*=" _blank" [4]

1.9 Formuláře

Formuláře se staly nedílnou součástí stránek a jedním z jejich nejpoužívanějších prvků. Formuláře slouží k odesílání informací na server nebo e-mailem. Klasický formulář se skládá ze tří základních částí – značky FORM, prvků formuláře a tlačítek.

1.9.1 Značka FORM

Tento párový tag uvozuje a ukončuje formulář. Uzavírá skupinu prvků do jednoho formuláře, který pak bude odeslán. K nastavení funkce formuláře existují tyto parametry:

- ACTION – použití k zpracování dat, která se většinou zasílají skriptem nebo e-mailem, není-li nijak uvedeno, odešlou se data stránce
- METHOD – způsob předávání dat, standardně se nastavují hodnoty GET a POST
 - GET – výchozí metoda, kde se data předávají jakou součástí URL
 - POST – zabalí odesílaná data, nezávisle je odesílá, nejsou viditelná, tato metoda se doporučuje u delších formulářů

- ENCTYPE – parametr, který slouží pro zakódování dat, není nutné jej zadávat, pokud nejde o zasílání souborů, pro zasílání jednoduché pošty se standardně nastavuje text/plain [2]

Příklad:

```
<FORM ACTION="script.php" METHOD="get" ENCTYPE="text/plain">
  <INPUT TYPE="text" VALUE="Jednoduchý formulář - skript" SIZE="35">
  <INPUT TYPE="submit" VALUE="Odeslat">
</FORM>

<FORM ACTION="mailto=adresa@server.cz" METHOD="post"
ENCTYPE="text/plain">
  <INPUT TYPE="text" VALUE="Jednoduchý formulář - e-mail" SIZE="35">
  <INPUT TYPE="submit" VALUE="Odeslat">
</FORM>
```

1.9.2 Prvky formuláře

Základem mezi prvky formuláře je vstupní pole. Jedná se o nepárový tag `<INPUT>`, který má k dispozici parametry konkretizující funkce vstupního pole.

Parametry:

- TYPE – určuje jakého typu bude vstupní pole, existuje několik možností
 - TEXT – vstupní pole bude obsahovat obyčejný text, s TEXT souvisí další parametry, kterými se nastavuje pole
SIZE – nastavuje šířku pole podle počtu nastavených znaků,
MAXLENGTH – číslem se nastaví maximální počet znaků,
AUTOCOMPLETE – doplnění známých dat
 - PASSWORD – pole pro heslo, psané znaky budou skryté (hvězdičky, puntíky), další parametry
SIZE – nastavuje šířku pole podle počtu nastavených znaků,
MAXLENGTH – číslem se nastaví maximální počet znaků,
 - RADIO – přepínač, většinou formou puntíků, v případě, že máme více vstupních polí různých hodnot toho typu nastavujeme všem stejné jméno, tím vytvoříme skupinu možností
CHECKED – přednastaví zaškrtnutí v přepínači

- CHECKBOX – zaškrťavající pole
CHECKED – přednastaví zaškrtnutí pole
- HIDDEN – pole je skryto a má přednastavenou hodnotu
- FILE – umožňuje zadávat soubory, v poli se objeví absolutní cesta
- SUBMIT – tlačítko, které potvrdí odeslání formuláře
- RESET – tlačítko, vymaže obsah formulář (stejnou funkci má i klávesa Esc)[11]
- NAME – udává jméno vstupního pole
- VALUE – hodnota pole
- READONLY – obsah se nedá měnit, je určený pouze pro čtení
- ALIGN – zarovnání (stejně jako např. u obrázků)[2]

Příklad:

```
<FORM ACTION=" " METHOD="get" ENCTYPE="text/plain">
  text<br>
  <INPUT TYPE="text" NAME="text" VALUE="Text v poli" SIZE="30">
  <INPUT TYPE="text" NAME="text" VALUE="Pouze pro čtení"
  READONLY><br><br>
  password<br>
  <INPUT TYPE="password" NAME="heslo" MAXLENGTH="8"><br><br>
  radio<br>
  <INPUT TYPE="radio" NAME="radio" checked="checked">
  <INPUT TYPE="radio" NAME="radio">
  <INPUT TYPE="radio" NAME="radio"><br><br>
  checkbox<br>
  <INPUT TYPE="checkbox" NAME="checkbox" checked="checked"><br><br>
  hidden<br>
  <INPUT TYPE="hidden" NAME="skryte">existuje, ale není zobrazen
  <br><br>
  file<br>
  <INPUT TYPE="file" NAME="soubor"><br><br>
  submit<br>
  <INPUT TYPE="submit" NAME="odesli" VALUE="Odeslat"><br><br>
  reset<br>
  <input TYPE="reset" NAME="vymaz" VALUE="Vymaž"><br><br>
</FORM>
```

Další z řady základních prvků pro tvorbu formuláře je výběr, nebo-li párová značka `<SELECT>`. Vytvoří pole s roletkovou nabídkou. Stejně jako u INPUT jsou k dispozici další parametry pro definování funkce a vzhledu :

- NAME – jméno,
- MULTIPLE – definuje hromadný výběr za pomoci klávesy CTRL,
- SIZE – určuje počet řádků,
- DISABLED – nastavuje blokaci výběru, pole je zašedlé a nelze měnit,

položky výběrového pole se pak nadefinují párovou značkou `<OPTION>` s parametry,

- VALUE – udává řetězec či hodnotu pole,
- SELECTED – předdefinovaná položka,
- DISABLED. [11]

Příklad:

```
<SELECT NAME="staty" SIZE="5">
    <OPTION VALUE="sk">Slovensko</OPTION>
    <OPTION VALUE="cz" SELECTED>Česká republika</OPTION>
    <OPTION VALUE="ger">Německo</OPTION>
    <OPTION VALUE="gb">Anglie</OPTION>
    <OPTION VALUE="gb" DISABLED>Mezopotámie</OPTION>
</SELECT>
<SELECT NAME="staty">
    <OPTION VALUE="sk">Slovensko</OPTION>
    <OPTION VALUE="cz" SELECTED>Česká republika</OPTION>
    <OPTION VALUE="ger">Německo</OPTION>
    <OPTION VALUE="gb">Anglie</OPTION>
    <OPTION VALUE="gb" DISABLED>Mezopotámie</OPTION>
</SELECT>
```

Klasické pole může být pro delší texty nepřehledné, a proto existuje rozsáhlé vstupní pole. Pro vytvoření se používá párový tag `<TEXTAREA>`, který je zobrazen v rámečku s lištou. Parametry pro nastavení jsou :

- NAME,
- COLS – nastavuje šířku pole počtem znaků,
- ROWS – nastavuje výšku pole počtem znaků,
- DISABLED,

- REDONLY – pouze pro čtení, nelze upravovat,
- WRAP – jedná se o nastavení zalomení řádku
 - SOFT – řádky mají povoleno se zalomit pouze v místě mezery, zalamují se ve slově, přesahuje-li celý řádek, odešle, co je zapsané,
 - HARD – řádky jsou zalomeny v místě mezery nebo libovolně v dlouhém slovu, v případě, že se zalomí, je zlom považován za konec řádku a bude tak odeslán,
 - OFF – řádek se nezalamuje, dle potřeby se objevuje vodorovný posuvník, odešle, co je zapsané. [11]

Příklad:

```
<FORM ACTION="" METHOD="get" ENCTYPE="text/plain">
  <TEXTAREA NAME="textpole" COLS="25" ROWS="7" WRAP="soft">
    Toto je textové pole vytvořené tagem TEXTAREA s nastavením 25
    znaků na řádek, 7 znaků sloupec a zalomením textu SOFT.
  </TEXTAREA>
</FORM>
```

K poli samozřejmě patří i jejich popisek či štítek, který se většinou vyskytuje před nimi (prohlížeč nemusí popisky podporovat a mohl by uživatele zmást). V případě kliknutí se aktivuje pole formuláře. Vlastní text se zadává mezi párové značky *<LABEL>*

Také tato značka má svůj atribut :

- FOR – slouží k propojení s polem stejného identifikátoru. [11]

Příklad:

```
<LABEL FOR="vyucujici">Zde napište jméno vyučujícího:</LABEL>
<INPUT TYPE="text" ID="vyucujici">
```

Jednotlivá pole se mohou sdružovat v celek a pro viditelnost tohoto spojení se kolem vykreslí ohraničující rámeček. Jedná se o skupinu polí, která se vytvoří za pomoci *<FIELDSET>*. Tato značka má obecné atributy.

Se skupinou polí je spjat i jeho popisek, je definován párovými značkami *<LEGEND>* s parametrem :

- ALIGN – zarovnání (stejně jako v případě obrázků). [11]

Příklad:

```
<FIELDSET>
  <LEGEND>Registrace</LEGEND>
  Jméno:<INPUT TYPE="text" NAME="jmeno"><BR>
  Příjmení:<INPUT TYPE="text" NAME="prijmeni"><BR>
  E-mail:<INPUT TYPE="text" NAME="mail" VALUE="@"><BR>
  Heslo:<INPUT TYPE="password" NAME="heslo">
</FIELDSET>
```

Tag `<OPTGROUP>` souvisí s výše uvedeným výběrem SELECT. Jedná se o párový tag, v jehož prostoru je uzavřená skupina tagů `<OPTION>`. Jedná se o nadpis, který bude zobrazen tučnou kurzívou. Parametrem k značce je :

- LABEL – definuje nadpis

Některé prohlížeče tuto značku ignorují. [11]

Příklad:

```
<SELECT NAME="staty" SIZE="4">
  <OPTGROUP LABEL="Státy">
    <OPTION VALUE="sk">Slovensko</OPTION>
    <OPTION VALUE="cz" SELECTED>Česká republika</OPTION>
    <OPTION VALUE="ger">Německo</OPTION>
  </OPTGROUP>
</SELECT>
```

1.9.3 Tlačítka

Tlačítko nemusí být vytvořeno jen za pomoci INPUT, ale má i svoji vlastní značku. Výhodou je, že mezi uvozující a zakončující značku lze cokoliv vložit, například obrázky, nadpisy. Jedná se o párovou značku `<BUTTON>` s parametry :

- TYPE – určuje, jakého typu bude vstupní pole, existuje několik možností nastavení tlačítka
 - SUBMIT – tlačítko odešle formulář
 - RESET – tlačítko vymaže formulář
 - BUTTON – váže se na skript, který určuje jeho činnost
- NAME – jméno pole
- VALUE – hodnota určená k odeslání

V praxi se však doporučuje zůstat u značky INPUT kvůli chybám v prohlížeči, konkrétně v IE. [2]

Příklad:

```
<BUTTON NAME="tlacitko1" VALUE="tlacitko1">tlačítko vytvoření tagem  
<I>BUTTON</I></BUTTON><BR>  
<BUTTON NAME="tlacitko1" VALUE="tlacitko2"><IMG SRC="obrazek.gif"  
HEIGHT="50" WIDTH="50"></BUTTON>
```

1.10 Rámy

I když se rámy v dnešní době nepoužívají, je dobré o jejich existenci a použití vědět. Když uživatel prochází Internetem, vidí rozložení stránek. Stránky mívají nějakou hlavičku, menu s odkazy a centrální část, kde se zobrazuje vybrané téma. Jediná stránka se jeví jako nedostačující. Rámy jsou dobrý způsobem, jak do jedné velké stránky seskládat několik stránek malých.

1.10.1 Definice rámu

Na začátku je důležité určit, z kterých stránek se bude stránka skládat. Úvodním příkladem mohou být *stranka_a.html* a *stranka_b.html* s libovolným obsahem. K nim musí být vytvořená stránka *index.html*, v níž bude kód pro výslednou stránku s rámy. Rámy budou nadefinované jako sloupce.

Příklad:

```
<HTML>  
<HEAD>  
  <TITLE></TITLE>  
  <FRAMESET COLS="30%, *">  
    <FRAME NAME="RamA" SRC="stranka_a.html">  
    <FRAME NAME="RamB" SRC="stranka_b.html">  
  </FRAMESET>  
</HEAD>  
</HTML>
```

Úvodní značkou při tvorbě rámu je `<FRAMESET>`, která definuje, jak bude rozdělení stránky vypadat. Vzhled stránky, konkrétně rozdělení na rámy se definuje za pomoci dvou parametrů :

- ROWS="a,b" – velikost ráků ve vodorovném směru (řádky),
- COLS="c, d" – velikost rámu ve svislém směru (sloupce).

Pro zadávání hodnot parametrů (a, b, c, d) existuje několik způsobů :

- Přesný počet pixelů – zadáváme celými čísly v rozmezí velikosti stránky, např. COLS="40, 50"
- Procentuální zadání – velikost je zadávána v procentech, např. 25% bude čtvrtina z celkové velikosti okna prohlížeče.
- Hvězdička – tento znak představuje hodnotu, která bude dopočítaná na základě první hodnoty, např. COLS="25%, *"

Další značkou je `<FRAME>`, která definuje, jaký soubor bude zobrazen v určených rámech. Jeho parametry jsou :

- NAME – jméno rámu,
- SRC – zdroj (cesta k souboru). [1]

Další parametry budou vysvětleny a zmíněny v následujících kapitolách.

1.10.1.1 Příklady použití ráků

- 3 sloupcové ráky

K vytvoření jsou zapotřebí 3 stránky, kterými se vytvoří jedna stránka s ráky.

Příklad:

```
<HTML>
<HEAD>
  <TITLE>Sloupcové ráky</TITLE>
  <FRAMESET COLS="25%, 35%, *">
    <FRAME NAME="RamA" SRC="stranka_a.html">
    <FRAME NAME="RamB" SRC="stranka_b.html">
    <FRAME NAME="RamC" SRC="stranka_c.html">
  </FRAMESET>
</HEAD>
</HTML>
```

- 3 řádkové rámy

Postup je stejný s tím rozdílem, že nahradíme COLS za ROWS.

Příklad:

```
<FRAMESET ROWS="25%, 35%, *">
    <FRAME NAME="RamA" SRC="stranka_a.html">
    <FRAME NAME="RamB" SRC="stranka_b.html">
    <FRAME NAME="RamC" SRC="stranka_c.html">
</FRAMESET>
```

- Zanořené rámy

```
<FRAMESET ROWS="50%, *">
    <FRAMESET COLS="50%, *">
        <FRAME NAME="RamA" SRC="stranka_a.html">
        <FRAME NAME="RamB" SRC="stranka_b.html">
    </FRAMESET>
    <FRAME NAME="RamC" SRC="stranka_c.html">
</FRAMESET>
```

1.10.2 Další parametry

1.10.2.1 Parametry FRAMESET

FRAMEBORDER – nastavuje ohraničení rámu, standardně je nastavena 1, tedy rám je vidět. V případě, že rám nemá být vidět, bude zadána do parametru 0.

Příklad:

```
<FRAMESET ROWS="50%, *">
    <FRAMESET COLS="50%, *" FRAMEBORDER="0">
        <FRAME NAME="RamA" SRC="stranka_a.html">
        <FRAME NAME="RamB" SRC="stranka_b.html">
    </FRAMESET>
    <FRAMESET COLS="50%, *" FRAMEBORDER="1">
        <FRAME NAME="RamC" SRC="stranka_c.html">
        <FRAME NAME="RamD" SRC="stranka_d.html">
    </FRAMESET>
</FRAMESET>
```

FRAMESPACING – definuje počtem bodů prostor mezi rámy, který nebude nijak vyplněn.[4]

Příklad:

```
<FRAMESET ROWS="50%,*" FRAMESPACING="15">
  <FRAMESET COLS="50%,*">
    <FRAME NAME="RamA" SRC="stranka_a.html">
    <FRAME NAME="RamB" SRC="stranka_b.html">
  </FRAMESET>
  <FRAMESET COLS="50%,*">
    <FRAME NAME="RamC" SRC="stranka_c.html">
    <FRAME NAME="RamD" SRC="stranka_d.html">
  </FRAMESET>
</FRAMESET>
```

1.10.2.2 Parametry FRAME

NAME – jméno je důležité pro směřování stránek do ráků, nejčastější použití se týká odkazů.

FRAMEBORDER – nezávisle na značce FRAMESET nastavuje viditelnost rámu.

NORESIZE – zákaz změny velikosti rámu práci s prohlížečem (zmenšení velikosti tažením).

SCROLLING – definuje, zda bude možnost rolovat dokumenty.

- YES – budou zobrazeny rolovací šipky
- NO – nebudou zobrazeny
- AUTO – objeví se v případě, že stránka nebude dostačující pro zobrazení celého dokumentu

MARGINHEIGHT – definuje výšku volného okraje nad a pod rámem.

MARGINWEIGHT – definuje šířku volného okraje vlevo a vpravo od rámu [4].

Příklad:

```
<FRAMESET ROWS="50%,*">
  <FRAMESET COLS="50%,*">
    <FRAME NAME="RamA" SRC="stranka_a.html" FRAMEBORDER="0"
      SCROLLING="yes">
    <FRAME NAME="RamB" SRC="stranka_b.html" MARGINHEIGHT="5"
      MARGINWEIGHT="5">
  </FRAMESET>
  <FRAMESET COLS="50%,*">
    <FRAME NAME="RamC" SRC="stranka_c.html" FRAMEBORDER="0">
    <FRAME NAME="RamD" SRC="stranka_d.html" FRAMEBORDER="0"
      SCROLLING="yes">
  </FRAMESET>
</FRAMESET>
```

1.10.3 Směrování stránek do rámu

Pozornost bude opět věnována problematice naznačené v minulé kapitole - směrování stránek do rámu.

Každý vytvořený rám by měl mít své jméno určené parametrem *NAME*.

Příklad:

```
<FRAME SRC="stranka_a.html" NAME="Název rámu">
```

Do kódu odkazu pak bude nadefinován parametr, který bude určovat, kam se má stránka načíst.

Příklad:

```
<A HREF="stranka.html" TARGET="Název rámu">
```

Parametr TARGET má další hodnoty v případě, kdy není určen konkrétní rám, v němž bude stránka načtená, neboť závisí na aktuálním stavu.

- *_BLANK* – vytvoří se nové okno v prohlížeči
- *_SELF* – načtení stránky proběhne do stejného okna, z kterého byl volán
- *_TOP* – stránka se načte do celého okna prohlížeče
- *_PARENT* – stránka se načte z aktuálního do nadřazeného okna, z kterého byl odkaz zavolán [1]

2 ÚVOD DO CSS

2.1 Historie

Jazyk Cascading Style Sheets (tabulky kaskádových stylů) nebo-li CSS vznikl 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, nyní se pracuje na verzi CSS 3. [13]

2.2 K čemu jsou CSS dobré?

CSS se používá k formátování obsahu dokumentů typu HTML, XHTML a XML. V porovnání s HTML má formátovací schopnosti, které práci rozšiřují a usnadňují. Styly umožňují přesně určit, jak bude který element vypadat. Jednotný vzhled elementu lze definovat pro celý dokument za pomoci zápisu pro příslušný element. Stejně lze pomocí stylu určit odlišné formátování i pro jediný výskyt určitého elementu. Ušetří se velké množství kódu a samotný kód se stane mnohem přehlednějším. Nespornou výhodou je i možnost editace kódu. V případě, že bude zapotřebí provést změny, například barvu písma všech odstavců, bude tato práce otázkou chvíle proti změnám každého atribut u každého elementu v HTML. Jeden styl lze snadno použít pro libovolné množství stránek. [13]

2.3 Úvod

Chce-li tvůrce stránek začít formátovat za pomoci kaskádových stylů od začátku, vytvoří nejdříve strukturovaný dokument, který nezohledňuje budoucí prezentaci stránky. Tento dokument bude popisovat pouze uspořádání obsahu, jeho rozdělení do celků, posloupnost a hierarchii informací, které má finální stránka sdělit. Dokument v tomto tvaru je připravený k formátování kaskádovými styly.

2.3.1 Připojení stylů k dokumentu

Do HTML stránek lze připojit kaskádové styly několika způsoby. Lze je definovat přímo v dokumentu nebo v externím souboru, způsoby se mohou i kombinovat.

2.3.1.1 Externí soubor

Má-li být styl uložený v externím souboru, což je výhodný způsob při používání jednoho stylu pro více dokumentů, bude definovaný styl zapsán do souboru s příponou *css*. Ten se pak připojí k dokumentu zápisem v hlavičce, například v tagu *LINK* [11]

Příklad:

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

2.3.1.2 Definování stylu uvnitř dokumentu

Definování CSS uvnitř dokumentu lze provést v tagu *STYLE*. Říká se mu "stylopis". Stylopis je seznam stylů. Do stránky se запиše mezi značky *<style>* a *</style>*. Rozdíl mezi tímto a předchozím způsobem znamená, že na místo odkazu na externí styl přijde přímo definice stylu.

Příklad:

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

Styl lze definovat přímo samotnému elementu, což se jeví výhodou v případě, kdy je nadefinovaný jednotný styl, ale zde se má použít jiné pravidlo. Pak se запиše v příslušném tagu atribut *STYLE* [2]

Příklad:

```
<h1 style="color: green; text-align:left">Tento nadpis bude zobrazen
zeleně a zarovnan doleva</h1>
```

2.3.2 Váha stylů

Definuje-li tvůrce stránek pro stejný element stejnou vlastnost dvakrát, vyšší váhu bude vždy mít ta deklarace, která byla definovaná později (na nižším řádku), ta také bude realizována. Pokud by měla některá deklarace mít vyšší prioritu, lze ji nastavit za pomoci *!important*. [13]

Příklad:

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

2.3.3 Přiřazení definic prvkům stránky

Styl se skládá z pravidel pro jednotlivé elementy, které mají být formátovány. Každé pravidlo se skládá ze dvou částí :

- selektor - název elementu, pro který má toto pravidlo platit
- deklarace - určujeme vlastnost a její hodnotu, deklarace je uzavřena do složených závorek.

Celý zápis - *selektor {vlastnost: hodnota_vlastnosti}*

Příklad:

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

V případě, že má být elementu nedeklarováno více vlastností, budou vlastnosti od sebe oddělené středníkem. Takto lze definovat libovolné množství vlastností pro jeden element.

selektor {vlastnost1: hodnota_vlastnosti1; vlastnost2: hodnota_vlastnosti2;}

Pokud mají mít dva elementy společnou vlastnost, budou od sebe jednotlivé selektory odděleny čárkou.

selektor1, selektor2 {vlastnost: hodnota_vlastnosti;} [13]

2.3.4 Dědičnost

Převážná většina vlastností se dědí. To znamená, že element, který nemá definovanou vlastnost, ji dědí po nadřazeném elementu. Dědičnost se týká především vlastností písma. Pokud má být nadefinovaná vlastnost společná pro všechny elementy, bude definovaná pro element BODY. [13]

2.3.5 Komentáře

Všeobecně se doporučuje zapisovat si do jakéhokoliv kódu poznámky pro lepší orientaci. Poznámky se zapisují do komentářů. Ty se v CSS tvoří pomocí /* a */. Mezi hvězdičky lze umístit i několikařádkový komentář. Ten se samozřejmě ve výsledném zobrazení neobjeví. [11]

Příklad:

```
body {color: blue} /* toto je příklad komentáře*/
```


2.4 Formátování textu

2.4.1 Typ písma

V CSS se typ písma určuje vlastností *font-family*. Definuje konkrétní název písma, které bude použito - *<family-name>*, jméno rodiny písma. Dále by mělo být zapsáno písmo, které bude použito v případě, že první definované písmo nemá prohlížeč k dispozici. Dá se určit i několik alternativních písem. V případě, že by prohlížeč neměl k dispozici žádné z písem, používá se takzvaná obecná rodina, nebo-li *<generic-family>*. Měla být použita vždy na konci výpisu písem.

Obečné rodiny:

- serif - patkové písmo (např. Times New Roman)
- sans-serif - bezpatkové písmo (např. Arial, Arial CE, Arial Narrow, Verdana, Helvetica)
- monospace - neproporciální písmo (např. Courier, Courier New)
- cursive - ozdobná kurzíva (např. Comic Sans)
- fantasy - ozdobné písmo (např. Western) [13]

Příklad:

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

2.4.2 Velikost písma

Velikost písma se určuje vlastností *font-size*. Jako hodnota se používají klíčová slova nebo číselné hodnoty s jednotkou, například 12px, případně procenta. Texty na stránce by neměly mít stejnou velikost. Dědičnost z *BODY* to ani nemůže předeepsat. Mnoho prvků má totiž nadefinovanou hodnotu velikosti ve výchozí tabulce prohlížeče, zejména nadpisy.

Jedna z možností, jak určit velikost písma, je zapsaná níže v tabulce *Tab. 1*. Lze využít 7 různých velikostí [11]

<i>hodnota, klíčové slovo</i>	<i>odpovídá </i>	<i>velikost odpovídá v IE v normálním nastavení a v Opeře bez doctype (quirk mode)</i>	<i>odpovídá ve standardním nastavení Mozilly a ve standard modu Opery a Exploreru 6 (standard mode)</i>
xx-small	1	10px	9px
x-small	2	13px	10px
small	3	16px	13px
medium	4	18px	16px
large	5	24px	18px
x-large	6	30px	24px
xx-large	7		30px
Bez zadání	3 nebo 4	16px	16px

Tab. 1. Velikost písma

Příklad:

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

Velikost písma se dá určit v pixelech, milimetrech a jiných jednotkách..

- px – pixel, obrazový bod,
- mm – milimetr,
- in – palec,
- em – výška písmena M. [13]

Příklad:

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

Velikost písma lze určit relativně vzhledem k aktuální velikosti písma :

- larger - větší než aktuální písmo,
- smaller - menší než aktuální písmo. [13]

Příklad:

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

Relativně vzhledem k aktuální velikosti písma můžeme velikost určit v procentech. [13]

Příklad:

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

Font-size je v prohlížečích velmi dobře podporovaná vlastnost. Stručné porovnání je uvedeno v tabulce Tab. 2. [11]

<i>Prohlížeč</i>	<i>Podpora ve verzích</i>	<i>Poznámka</i>
Internet Explorer	všechny verze	IE 5.0 chybí u larger a smaller špatně interpretuje slovně zadané hodnoty (v quirk módu je kreslí o stupeň větší)
Mozilla (Firefox, Netscape 6+ atd.)	všechny verze	
Chrome	všechny verze	
Opera	ano	

Tab. 2. Podpora velikosti písma v prohlížečích

2.4.3 Řez písma

Pro styl (řez) písma jsou k dispozici v CSS vlastnosti *font-style* a *font-weight*. Vlastnost *font-style* určí, zda se bude jednat o písmo normální, kurzívu, nebo skloněné písmo. Rozdíl mezi kurzívou a skloněným písmem spočívá ve skutečnosti, že kurzíva je jedinečným řezem písma, zatímco skloněné písmo vzniká transformací normálního řezu.

- normal – normální,
- italic – kurzíva,
- oblique – skloněné písmo.

Příklad:

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

Vlastnost *font-weight* definuje sílu písma, konkrétněji poměr tloušťky písmových tahů k výšce písma.

- normal – základní, netučné písmo,
- bold – tučné písmo. [11]

Příklad:

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

2.4.4 Varianta písma

Existují dvě varianty písma - normální písmo a kapitálky. Určují se za pomoci vlastnosti *font-variant*.

- normal – normální,
- small-caps – kapitálky. [11]

Příklad:

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

2.4.5 Dekorace písma

Text lze opatřit dalšími efekty za pomoci vlastnosti *text-decoration*. V CSS jsou k dispozici:

- underline – podtržení textu,
- overline – linka nad textem,
- linethrough – přeškrtnutí,
- blink – blikající text,
- none – žádný efekt.

Tyto dekorace se většinou používají pro zvýraznění odkazů. Pro běžný text jsou zanedbatelné. [13]

Příklad:

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

2.4.6 Zarovnání textu

Za pomoci vlastnosti *text-align* lze horizontálně zarovnat obsah blokových prvků. Tato vlastnost může nabývat hodnot:

- left – zarovnání doleva,
- right – doprava,
- center – na osu,
- justify – zarovnání do bloku. [11]

Výchozí hodnotou bývá LEFT (v jazycích píšících zprava doleva je to RIGHT). Není-li určeno jinak, text bude zarovnán doleva. [13]

2.5 Barvy a obrázky

V této části půjde pouze o doplnění informací v oblasti CSS. Zápis barev v kaskádových stylech je stejný jako v HTML (název barvy v angličtině, RGB zápis).

2.5.1 Barvení dokumentů

Většině elementů v dokumentu lze zadat barvu pozadí pomocí vlastnosti *background-color*.

Příklad:

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

Barva textu se zadává vlastností *color*.

Příklad:

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

Pokud nebudou definovány žádné barvy, budou použity výchozí barvy prohlížeče. [13]

2.5.2 Pozadí

Na pozadí stránky může být jakýkoliv obrázek. Text však musí být čitelný ve všech částech obrázku. Samotný obrázek bude určen vlastností *background-image*, její hodnotou je cesta k obrázku, zapsaná v závorce za parametrem *url*. [13]

Příklad:

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

2.5.3 Opakování obrázku

Obrázek na pozadí se může opakovat nebo může být umístěn jen jednou. Standardně se obrázek opakuje ve vertikálním i horizontálním směru až do zaplnění celé určené plochy. Pro určení způsobu opakování se používá vlastnost *background-repeat*. Pokud se má obrázek opakovat jen ve směru x, tedy horizontálně, bude nadefinována hodnota *repeat-x*, ve směru y, tedy vertikálně, půjde o hodnotu *repeat-y*. Nemá-li se obrázek opakovat, využije se hodnota *no-repeat*. [13]

Příklad:

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

2.5.4 Umístění pozadí (obrázku)

Jestliže má být obrázek umístěn jinak než ve výchozí pozici, lze tuto skutečnost změnit vlastností *background-position*. Umístění určují dvě hodnoty - horizontální a vertikální. Hodnoty horizontálního umístění mohou být left (vlevo), center (na střed) a right (vpravo). Hodnoty vertikálního umístění jsou top (nahoru), center (na střed) a bottom (dolů). [11]

Příklad:

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

2.6 Seznamy

Seznamy se v CSS dají formátovat stejně jako jakékoliv jiné elementy. Navíc lze určit typ odrážky u číslovaného i nečíslovaného seznamu. Za pomoci obrázku lze nadefinovat i vlastní odrážku.

2.6.1 Odrážky a číslování

Typ odrážky a číslování se určuje v CSS vlastností *list-style-type*. Tato vlastnost může být přiřazena k selektoru seznamu (UL, OL) i položce seznamu (LI).

Příklad:

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

Nečíslované seznamy

- disc – vyplněné kolečko v IE a Opeře; vyplněný kosočtverec v Mozille a Netscape,
- circle – prázdné kolečko v IE a Opeře; prázdný kosočtverec v Mozille a Netscape,
- square – čtvereček vyplněný.

Číslované seznamy

- decimal – arabské číslice (1.),
- lower-roman – malé římské číslice (i.),
- upper-roman – velké římské číslice (I.),
- lower-alpha – malá písmena (a.),
- upper-alpha – velká písmena (A.),
- none – bez odrážky. [11]

Existují i další možnosti číslování seznamů, ale ty nemusí fungovat ve všech prohlížečích. Například decimal - leading-zero (arabské číslice), lower-greek – malá písmena řecké abecedy, lower-latin – malá písmena, upper-latin – velká písmena, armenian – arménština. [13]

2.6.2 Umístění odrážky

Odrážka je v textu umístěna pomocí vlastnosti *list-style-position*. V praxi mohou nastat dva případy :

- inside - uvnitř textu,
- outside - vně textu. [13]

2.6.3 Obrázek jako odrážka

Jako odrážka v seznamu může být použit i vlastní obrázek. Proto existuje vlastnost *list-style-image*, jejíž hodnotou je URL obrázku. [11]

Příklad:

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

2.7 Odkazy

Návštěvník stránek se na webu orientuje především podle odkazů. V mnohých případech text ani nečte, sleduje, kam směřují odkazy, a proto je vhodné, aby odkazy byly od okolního textu zřetelně odlišeny.

2.7.1 Barva odkazu

Výchozí nastavení v prohlížeči zobrazují odkazy modré, podtržené, navštívené odkazy jsou fialové, aktivní odkazy červené. Barva odkazu se nastavuje stejně jako barva kteréhokoliv elementu pomocí vlastnosti *color*. [11]

Příklad:

```
a {color: green}
```

2.7.2 Ozdobení textu

V HTML se odkazy zobrazují jako podtržené, neboť pro všechny je barevná odlišnost nedostačující. CSS nabízí další úpravu odkazů k odlišení od normálního textu.

K definování stylu ozdobení slouží vlastnost *text-decoration*. Tato vlastnost byla již zmíněná v kapitole Formátování textu. Jedná o *none* (bez ozdobení), *underline* (podtržení), *overline* (nadtržení) a *line-through* (přeškrtnutí). Všechny vlastnosti lze použít najednou.

2.7.3 Pseudotřídy

Pseudotřídy slouží k nastavení odlišnosti vlastností odkazů (link), již navštívených odkazů (visited) a aktivních odkazů (active). Zapisují se za tag s dvojtečkou.

Příklad:

```
a:visited{color:purple}
```

CSS nabízí pro úpravu odkazů další možnost - změnu vzhledu při přejetí kurzoru myši. Tato vlastnost se nazývá *hover* a opět se zapisuje za tag s dvojtečkou. [3]

Příklad:

```
a:hover{color:white; text-decoration:underline overline}
```

2.8 Třídy a identifikátory v CSS

Třídy a identifikátory v CSS slouží k tomu, aby tvůrce stránek mohl jednotlivé elementy formátovat různými způsoby. Jako konkrétní příklad mohou sloužit odkazy. Obvykle jsou odkazy v menu odlišné od odkazů umístěných v textu.

2.8.1 Třída CLASS

Třída *class* v CSS se vytvoří připojením k elementu v HTML. Hodnotou bude nějaký řetězec zapsaný v CSS stylu jako selektor. Element, který má být formátován, bude formátován podle pravidel deklarované třídy, na ostatních elementech se tato pravidla neprojeví. Tato pravidla je nutné určit v CSS stylu. [5]

Příklad:

```
<html>
  <head>
    ...
    <style type="text/css">
      ...
      .trida {color: blue;}           /*styl se aplikuje na všechny
                                     elementy s třídou trida*/
      h2.trida{color:blue;}         /*styl se aplikuje na všechny
                                     elementy h2 s třídou trida*/
    ...
  </style>
</head>

<body>
  ...
  <td class="trida" >
    ...
    <h2 class="trida">Text</h2>
    ...
  </body>
</html>
```

2.8.1.1 Dědičnost

Třída elementu bude dědit veškeré vlastnosti daného elementu a zároveň bude mít své vlastní vlastnosti.

Příklad:

```
p {text-align: right; color: green}
.trida {font-size:x-small; color: black}
```

Výše uvedený zápis potvrzuje, že všechny odstavce budou zarovnané doprava a text bude mít zelenou barvu (týká se první deklarace). Toto platí pro všechny odstavce na stránce. Odstavec, který bude ve svém kódu obsahovat třídu *trida*, bude mít navíc další vlastnosti - menší písmo. Barva je v tomto případě určena v obou deklaracích. Nastane-li tento stav, vyšší prioritu bude mít odstavec s třídou. Text bude černý a ostatní odstavce budou mít text barvy zelené. [13]

2.8.1.2 Kontextová deklarace s třídou

Možnost kontextové deklarace platí i pro třídu. Následující zápis určuje, že odkazy v odstavci třídy *trida* budou červené.

Příklad:

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

2.8.2 Identifikátor ID

Rozdíl mezi třídou a identifikátorem spočívá v jednoznačnosti identifikátoru. To znamená, že identifikátor může být na každé stránce, přesněji v souboru s příponou .html, použit pouze jednou. Ovšem v rámci celého webu může být stejný identifikátor použit mnohokrát. Identifikátory se používají v případě, kdy je jisté, že se element na stránce objeví jen jednou. Jsou vhodné pro box celé stránky, menu, záhlaví či zápatí. Identifikátory se označují znakem #, tak zvanou zahrádkou. Jejich zápis je stejný jako v případě tříd. [5]

Příklad:

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

2.9 CSS validátor

CSS validátor je určen k validaci CSS stylu. Jeho úkolem je najít chyby, které CSS styl obsahuje. Validátory se dají dohledat na Internetu. Příkladem může být <http://validator.w3.org/> či jeho česká verze <http://jigsaw.w3.org/css-validator/>.

Validovat lze:

- CSS styl z Internetu zápisem URL adresy stylu do políčka Validate by URI a stisknutím tlačítka check
- CSS styl uložený na disku vašeho počítače přímým zápisem cesty k souboru na lokálním disku do políčka Validate by File Upload nebo výběrem s procházením disku pomocí tlačítka browse a stisknutím tlačítka check

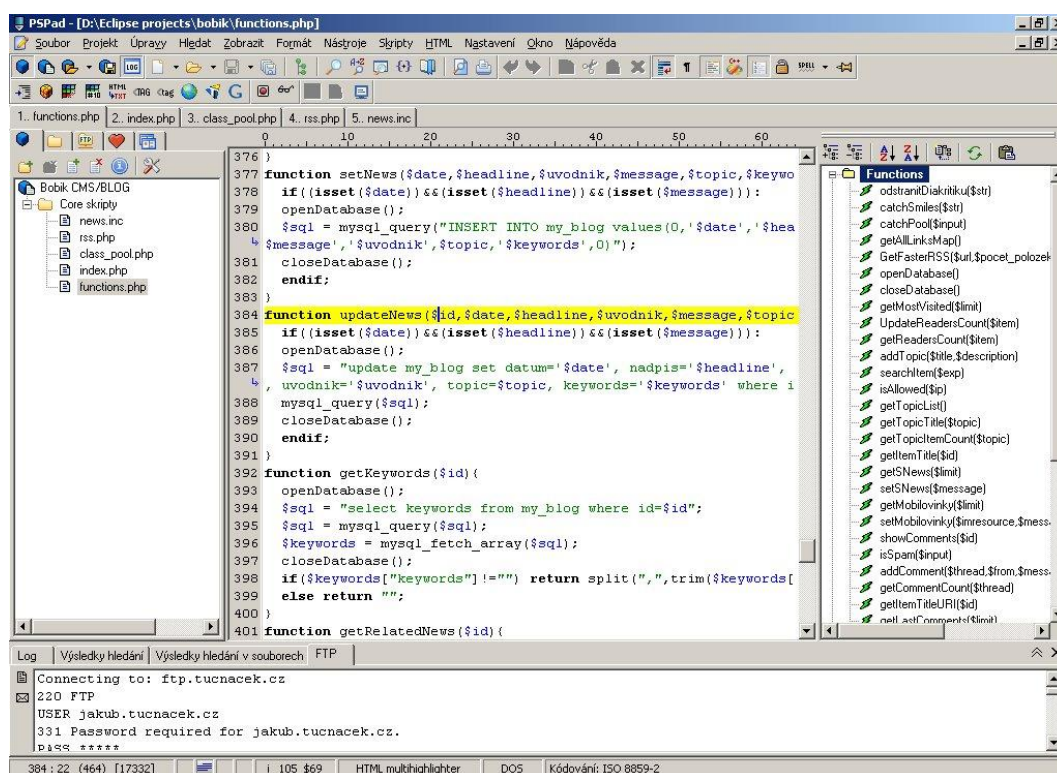
V případě, že je kód bezchybný, validátor nám tuto skutečnost oznámí. Je-li však v kódu chyba či varování, vypíše chybovou hlášku. [13]

3 TVORBA A PUBLIKOVÁNÍ WEBOVÝCH STRÁNEK

3.1 Programy používané pro tvorbu webových stránek

3.1.1 PSPad

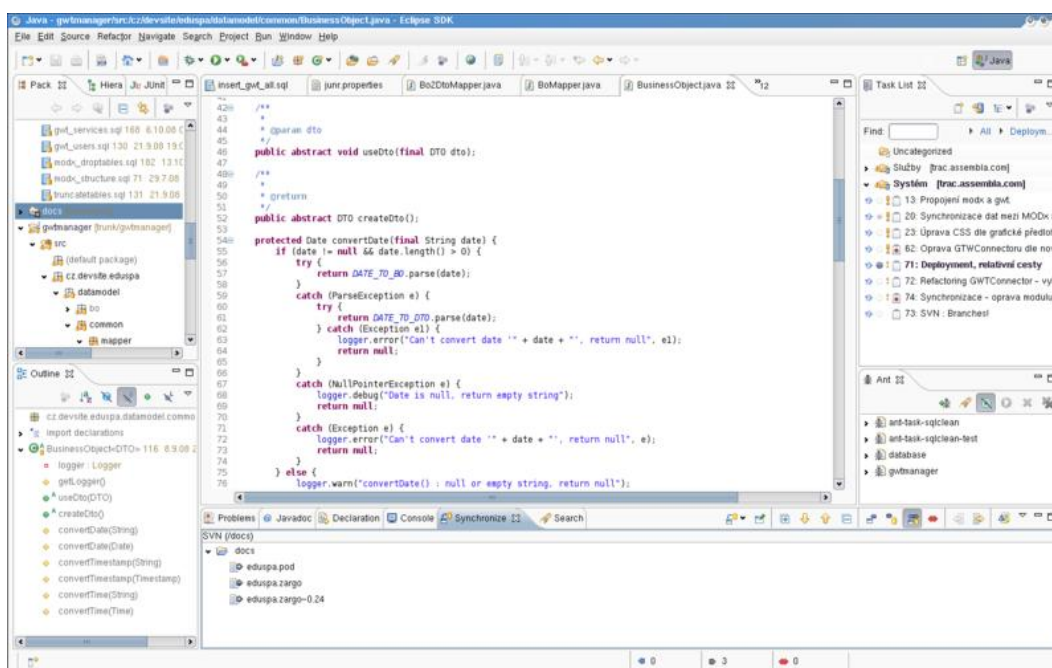
PSPad je český freewarový editor, určený programátorům, ale jeho možnosti lze využít i na běžné úpravy textových souborů. Nabízí jednoduché prostředí s možností snadné integrace do Windows. Kromě obvyklých základních funkcí textových editorů PSPad umí zvýraznit syntaxi jednotlivých souborů, což velmi usnadní jejich úpravy. Kromě tradičního HTML podporuje editor zvýraznění syntaxe: C, C++, Java, Javascript, Perl, PHP, SQL, Visual Basic atd. Uživatel může využít funkce náhledu stránky v prohlížeči s volbou "rozlišení" obrazovky. PSPad lze zdarma stáhnout z domovské stránky programu <http://www.pspad.com/> [14]



Obr. 2. Prostředí programu PSPad

3.1.2 Elipse

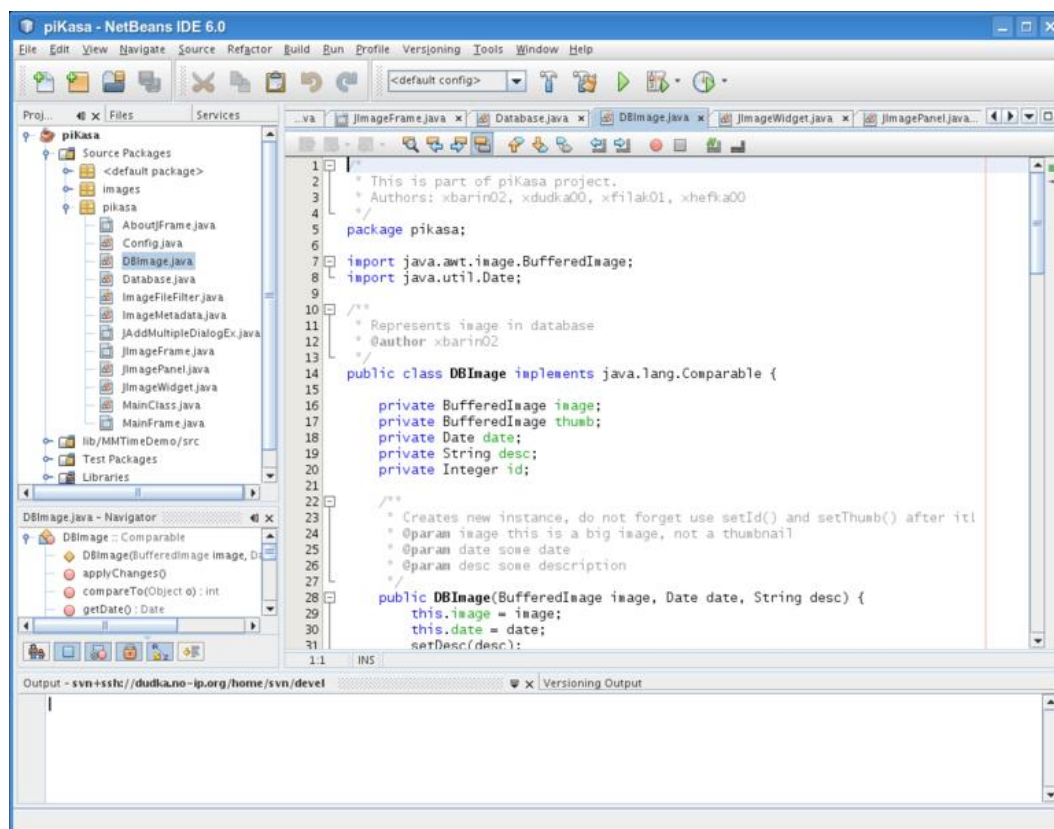
Eclipse je open source vývojová platforma, která je především známá jako vývojové prostředí určené pro programování v jazyce Java. Návrh této platformy dovoluje rozšířit seznam podporovaných programovacích jazyků za pomoci pluginů. Právě pluginy umožňují toto vývojové prostředí rozšířit například o návrh UML, zápis HTML nebo XML. Elipse je dostupný ke stažení na oficiálních stránkách <http://www.eclipse.org/>. [15]



Obr. 3. Prostředí programu Eclipse

3.1.3 NetBeans IDE

NetBeans je open source projekt s rozsáhlou uživatelskou základnou. Vývojové prostředí NetBeans IDE je nástroj, s jehož pomocí mohou programátoři psát, překládat, ladit a šířit programy. NetBeans IDE je napsáno v jazyce Java a postaveno na stejnojmenné platformě. Je určen pro vývoj aplikací v jazyce Java, ale může podporovat i další programovací jazyky například C++, PHP, Ruby. Lze jej použít na operačních systémech Windows, Linux, Mac OS X a Solaris. Stáhnutí se může realizovat na stránkách <http://netbeans.org/>. [16]



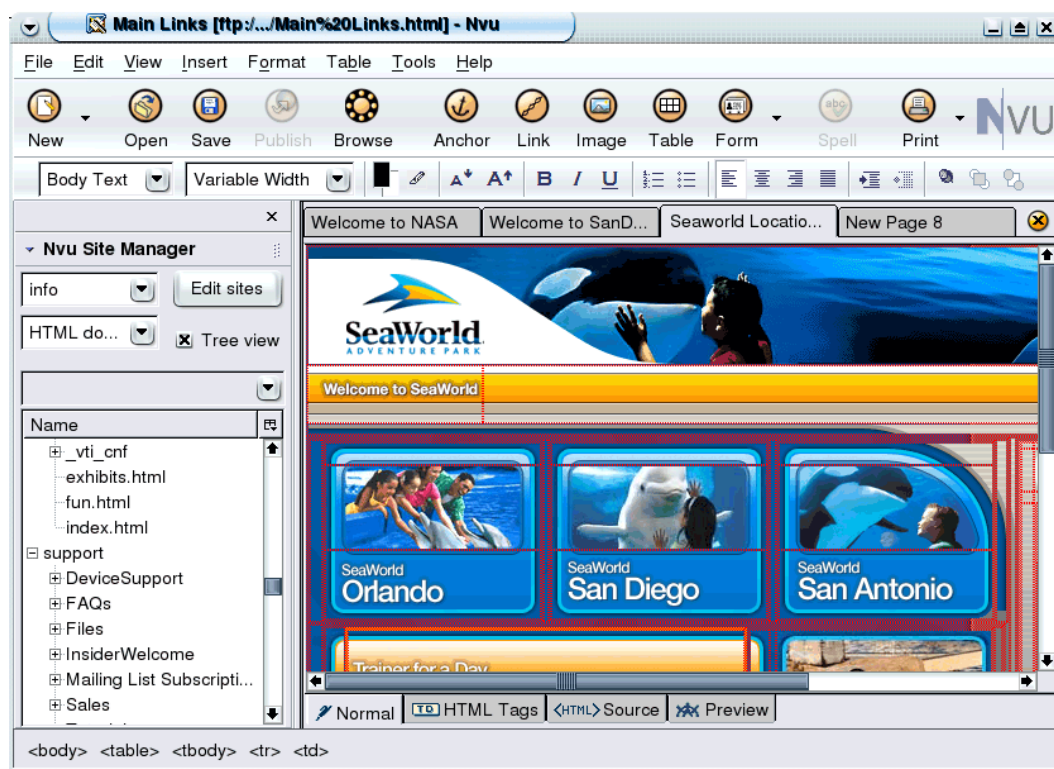
Obr. 4. Prostředí programu Netbeans IDE

3.1.4 WYSIWYG editory

WYSIWYG je akronym anglické věty „What you see is what you get”, tedy „Co vidíš, to dostaneš”. Tyto editory značně zjednodušují tvorbu internetových stránek. Přípravu webu lze přirovnat k vytváření dokumentu ve Wordu a webová stránka lze vytvářet bez znalostí HTML.

3.1.4.1 Nvu 1.0

Nvu je WYSIWYG editor určený k tvorbě a správě webových stránek, který je založen na Editoru z balíku Mozilla Suite. Jeho distribuce je zdarma. Při tvorbě stránek umožňuje přepínání mezi WYSIWYG editorem a editorem zdrojového kódu. Výsledek práce si lze prohlédnout přímo v editoru bez potřeby spouštět jakýkoliv prohlížeč. S editorem je možné v rámci WYSIWYG editoru psát PHP kód. Nvu má integrovaný CSS editor. Nedílnou součástí Nvu je Správce stránek, který umožňuje snadnou úpravu vytvářené webové stránky následně je nahrávat a zveřejňovat prostřednictvím protokolu FTP. Ke stažení je připraven na stránkách <http://net2.com/nvu/>. [17]



Obr. 5. Prostředí programu Nvu 1.0

3.1.4.2 WYSIWYG Web Builder

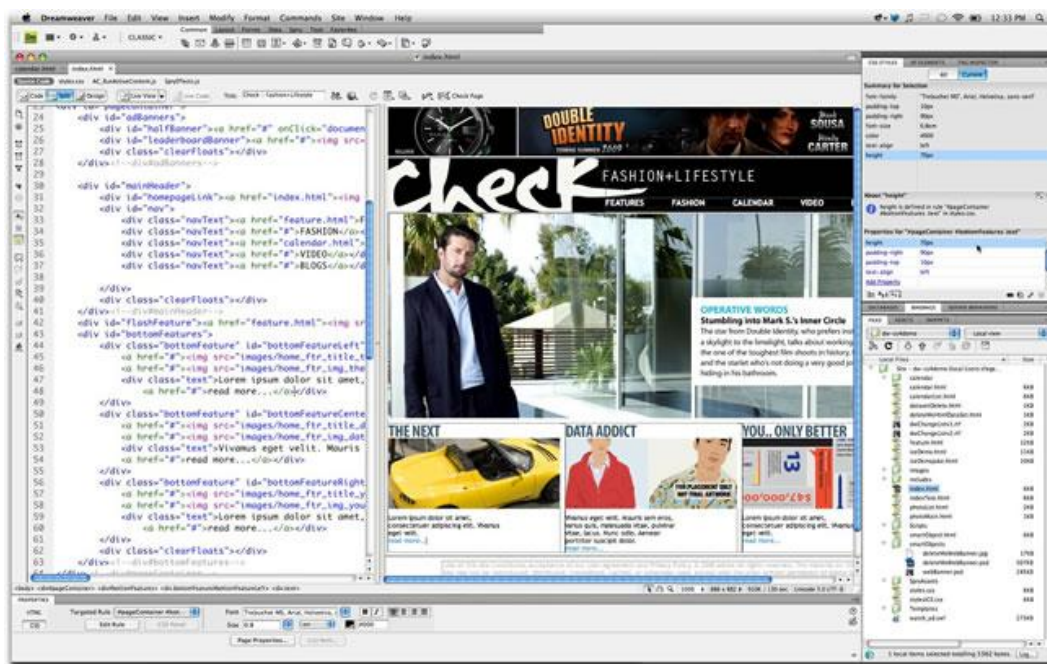
WYSIWYG Web Builder se orientuje v převážné míře na domácí uživatele, kterým nabízí snadné skládání a následné úpravy webových stránek. Tvorba probíhá jednoduchým umisťováním prvků na požadované místo za pomoci metody drag&drop, neboli přetahování objektů a komponent myši do náhledu stránky. Uživatel programu má možnost využít předdefinované šablony, do nichž je možno dále umisťovat vlastní obrázky, fotografie, texty. [18] Uživatel usazuje jednotlivé prvky přímo do zvoleného místa a editor sám zajistí vytvoření odpovídajícího zdrojového HTML kódu. Tento editor podporuje vkládání bloků textu, obrázků, tabulek, prvků Java, ActiveX, Flash, Windows Media a podobně přímo do internetových stránek. Uživatel si může svou práci ihned prohlédnout v náhledu. Součástí programu je také FTP klient, který hotové stránky publikuje na web. Česká verze WYSIWYG Web Builderu není k dispozici jako trial, nýbrž pouze jako placená licence pro 3 počítače. [19]



Obr. 6. Prostředí programu WYSIWYG Web Builder

3.1.4.3 Adobe Dreamweaver

Adobe Dreamweaver je software sloužící k vytváření webových stránek a různých dynamických aplikací. Podporuje HTML, XHTML, CSS, jQuery, JavaScript a jiné. Dreamweaver byl původně vyvíjený společností Macromedia. Následně jej převzala společnost Adobe Systems, která jej vyvíjí od verze 9.0 do aktuální verze CS5. Editor umožňuje tvorbu web grafickou cestou nebo přímo psaním kódu. K dispozici existují celkem tři módy. V módu pro čistě vizuální návrh jako v WYSIWYG editorech, pro úpravy kódu nebo v tak zvaném Split módu, který zároveň umožňuje upravovat kód a pozorovat změnu vzhledu ve vedlejším okně. Z nejzajímavějších nástrojů Dreamweaveru lze jmenovat zajištění kompatibility s různými prohlížeči za pomoci služby Adobe BrowserLab, podporu nápovědy při psaní čistého kódu, kontrolu CSS stylů či validace kódu přímo v aplikaci. [20]



Obr. 7. Prostředí programu Adobe Dreamweaver

3.2 Stručný úvod do webdesignu

3.2.1 Grafický design

Každý tvůrce stránek si dává samozřejmě záležet i na celkovém vzhledu jeho produktu. Bohužel však ne každému z nás bylo dáno do vínku grafické cítění a dovednosti. Tento problém se však dá vyřešit. U stránek, na jejichž vzhledu záleží, je vhodná úzká spolupráce s profesionálním grafikem. Druhou alternativou, jak dosáhnout pěkného designu stránek, je využití cizí práce stáhnutím hotového řešení. Příkladem mohou být šablony pro WordPress, které jsou sice původně určeny pro redakční systém, ale dají se úspěšně aplikovat na libovolný web. Mezi další zdroje volných designů patří Open Source Web Designs. Velký počet šablon a rozsáhlost samotného Internetu je zárukou originality. Stejně jako existují galerie volně dostupných šablon, existují i weby nabízející šablony za poplatek (např. 4Templates). [21]

3.2.2 Rady do začátku

Pro design většiny běžných webů platí pravidlo „v jednoduchosti je krása“. Design stránky by proto měl být jednoduchý. Výsledkem může být nevkusná stránka, která se navíc dlouho stahuje. Podíváte-li se na nejnavštěvovanější stránky na Internetu, zjistíte, že mají většinou

velmi jednoduchou grafiku, střízlivé barvy, jednobarevné světlé pozadí a někde jen malé logo stránek. [6] Další rada začátečníkům se může jevit nedůstojně, ale může víc pomoci než ublížit: „Opisujte, kde se dá“. Prohlížení stránek není závadou, mohou sloužit jako inspirace či prvotní myšlenka v případě, že se tvůrce rozhodl k vlastnímu výtvaru. [21]

3.2.3 Barvy a pozadí

Existují barvy, které spolu v kombinaci ladí a ty, které se k sobě absolutně nehodí. S citem pro barvy si je může tvůrce grafiky vybrat sám nebo přímo využít předdefinovaných barevných sad. Při výběru barev je důležité si uvědomit, že jisté procento populace trpí alespoň částečnou barvoslepostí, takže nemusí mít schopnost odlišit barevnou informaci, která má na něco důležitého upozornit. V každém případě je důležité volit dostatečně kontrastní barvy, aby text byl bez problému čitelný. Opatrnost je nezbytná i při výběru pozadí. Je vhodné používat jen jednu souvislou barvu. Pomocí prostředků HTML i CSS lze jednoduše vložit na pozadí libovolný obrázek, ale ten může velmi snadno snížit či úplně znemožnit čitelnost textu. Obrázek na pozadí je lépe nepoužívat, pokud ano, je třeba použít nenápadný. Ideální kombinací je nakonec klasická bílá barva nebo jemný světlý odstín nějaké barvy. Pro text je nejvhodnější černá, případně tmavě modrá barva. [6]

3.2.4 Výběr písma

Na výsledku práce má svůj nemalý podíl i volba písma. Web bohužel není spolehlivý co se týče přesného určení písma. Každý operační systém, prohlížeč a uživatel mohou mít k dispozici různá písma. Nelze se tedy spoléhat na jejich dostupnost. Tento problém se řeší určením několika variant písma, například Verdana, Helvetica, sans-serif. Prohlížeč zvolí první písmo ze seznamu, které je k dispozici. Problém může nastat i s autorskými právy. U většiny písem, které jsou nainstalovány v počítači, jejich licence zakazuje další šíření. Jejich použití může být považováno za porušení licenčního ujednání.

Není vhodné využívat na stránce příliš druhů písma. Většinou postačí dva různé druhy písma – pro běžný text a pro nadpisy. Pro web jsou vhodnější bezpatková písma, protože monitor má nízké rozlišení na precizní vykreslení patek. Pokud je použito písmo v malé velikosti, je vhodné pro lepší čitelnost použít písmo širšího typu. Speciálně pro web bylo navrženo písmo Verdana.

Pravidla existují i pro výběr správné velikosti písma. Web není určen jen teenagerům, ale i dalším generacím. Písmo, které je menší než 10 pixelů, je pro některé věkové kategorie málo čitelné. Dalším osvědčeným pravidlem, které určuje čitelnost textu, je počet znaků na řádek. Řádka textu by neměla obsahovat více než 40 až 60 znaků. V opačném případě je již řádek dlouhý a lidské oko může mít problém s čtením při přechodu na další řádek, rychlost čtení se podstatně snižuje. [6]

3.2.5 Zvýrazňování textu

V textu většinou není zapotřebí nic zvlášť zvýrazňovat, v odstavci stačí zvýraznit jedno či dvě slova nebo sousloví. Může převládat pocit, že zvýrazněním textu se lépe poukáže na důležitost informací (například vlastnosti nějakého produktu). Tento problém lze lehce obejít za pomoci seznamu. Důležitost informace se dává zvýraznit i změnou barvy textu, nikdy však změnou velikosti písma. Tvůrce by se měl vyvarovat podtrhávání částí textu. Tento zvyk je zakořeněný ještě z dob užívání psacích strojů. Podtržení snižuje čitelnost, protože zasahuje do spodních tahů písmen „gjpqy“. Podtrhávání textu může návštěvníky stránek zmást. Většina uživatelů si uvědomuje, že podtržením jsou v textu označeny odkazy. [6]

3.2.6 Základní typografická pravidla

3.2.6.1 Interpunkce

Pro interpunkci platí jednoduché pravidlo – mezera se píše za interpunkčními znaménky (tečka, čárka, vykřičník, otazník), nikdy před nimi. Kdyby byla mezera napsaná před interpunkčním znaménkem, mohlo by přejít na začátek řádku. Tento jev nevyvolá při prohlížení stránek příznivý dojem. [6]

3.2.6.2 Uvozovky

Spolehlivým způsobem, jak správně zapsat uvozovky, je využití znakových entit. Zápis entit do kódu vypadá takto: `“text v uvozovkách”`. [6] HTML jazyk skýtá další možnost vkládání uvozovek. K dispozici je element `q`, který se dá využít pro označování citací. Prohlížeč obsah elementu uzavře do uvozovek. Pokud se několik elementů `q` vloží do sebe, bude se znak pro uvozovky měnit. [11]

Příklad:

```
<p><q>
    názorná ukázka textu v uvozovkách, do kterého byl
    <q>vnořený další</q> text
</q></p>
```

3.2.6.3 Pomlčka

Dalším problematickým znakem je pomlčka. Existuje ve dvou variantách.

Krátká pomlčka ‘-’ se používá hlavně pro zápis intervalů, například 100–150 cm. Tento znak lze zapsat pomocí znakové entity *–*. Správně by se měl spojovník používat jen k rozdělování slov nebo k oddělování slovních spojení (budeme-li). Většina lidí ji používá jako spojovník (k rozdělování slov) i jako pomlčku (při zápisu intervalů a oddělování vět). Tento znak je přímo dostupný na klávesnici.

Dlouhá pomlčka ‘—’ naznačuje v mluvené řeči přestávky nebo se využívá pro výrazné oddělení částí projevu, například „Lidé celého světa – žádný národ nevyjímajíc – touží po míru“. Pro zápis dlouhé pomlčky existuje entita *—*. [6]

3.2.7 Grafické programy

Existuje mnoho programů, jejichž pomyslným vrcholem je Adobe Photoshop. Nicméně Photoshop svými možnostmi značně přesahuje rozsah schopnosti většiny normálních uživatelů. Schopnou náhradou může být Gimp. Tomuto programu se říká “Photoshop zadarmo“. Jeho možnosti jsou trochu omezené proti Photoshopu, ale mnohým uživatelům se zdá přátelštější. Grafik se nemusí omezovat na rastrovou grafiku, i vektorová splní svůj účel, ačkoliv její využití je v této oblasti neobvyklé. Pro tvorbu ve vektorové grafice lze využít program Inkscap. Jedná se o opensource, tedy je zcela zdarma. Pro Gimp i pro Inkscap existuje na Internetu velké množství tutoriálů. [21]

3.3 Umístění stránek na Internet

3.3.1 Jak správně připravit stránky ke zveřejnění

3.3.1.1 Pořádek v souborech

Vytvořené stránky by měly mít svůj vlastní samotný adresář, který bude určen výhradně pro ně. V tomto adresáři budou uloženy jen webové stránky, případně obrázky, které mají být na stránkách. Nic jiného! Je-li vytvořeno hodně stránek, je vhodné rozčlenit je do podadresářů. Dobrým zvykem bývá ukládat obrázky do samostatného podadresáře, nepletou se pak s HTML stránkami.

3.3.1.2 Relativní odkazy

Mezi stránkami by měly být pouze relativní odkazy. V případě, že budou stránky obsahovat absolutní odkazy, nemusí na serveru pracovat správně, jejich přesun na jiné místo bude obtížný. Častou chybou je uvádění úplných cest k souboru na disku místo relativní cesty.

3.3.1.3 Jména souborů

Základem dobré stránky není pouze kvalitní obsah a HTML kód, nýbrž i název. Pro zvolení dobrého názvu stránky (případně adresáře ve struktuře webových stránek) je vhodné :

- V názvech souborů musí být použita výhradně malá písmena. Snadněji se píše, velikost písmen je třeba sjednotit kvůli jednoznačnosti. URL adresy jsou na velikost písmen citlivé, stejně i webové servery pracující pod unixovými systémy, kterých je použito nejvíce.
- Ze stejných důvodů se v názvech souborů nepoužívá diakritika a mezery. Pro oddělení víceslovných spojení v názvu souboru se používá místo mezery podtržítko.
- Jména stránek by měla být co nejkratší a přitom výstižná.

3.3.1.4 Web ≠ Windows + Microsoft + Internet Explorer

Microsoft v dobré víře usnadnit uživatelům práci se stránkami zařadil do svého prohlížeče mnoho funkcí, které na stránkách opravují chyby. Pracuje-li tvůrce hodně ve Windows,

měl by si dávat pozor. Doporučuje se stránky cvičně otevřít a prohlédnout i v jiném prohlížeči než je IE. Většina podobných chyb se odhalí snadněji. [6]

3.3.2 Webhosting

Má-li být webová stránka veřejně dostupná, musí se umístit na Internet. Za tímto účelem existují webhostingy, které umožňují zaregistrovat část místa na svých serverech. Nutností je i solidní webová adresa, která se dá zaregistrovat nebo je dodávána přímo s hostingem.

K dispozici jsou dva typy hostingů - placené a neplacené. Hlavním rozdílem mezi těmito typy je velikost poskytovaného místa. U freehostingů se kapacita pohybuje od 200 do 500 MB, u placeného hostingu většina tarifů začíná nejméně na 1GB. Placený hosting má však navíc další výhody, například podporu nových technologií, dostupnost, lepší technickou podporu ze strany poskytovatele či vyšší povolený trafik. V případě freehostingu však neexistuje žádná záruka dostupnosti a majitel stránek musí počítat s nějakou formou reklamy (reklamní lišty, bannery).

Další důležitou částí celého procesu je webová adresa. U placeného hostingu je adresa přímo v ceně balíčku nebo je nutné si ji zaregistrovat zvlášť. Každopádně se nejdříve vybírá nejvyšší doména, tedy 1. úrovně, mezi které patří generické domény (com, net, org, info nebo národní cz, sk, uk atd). Doménu si lze vybrat jakoukoliv, samozřejmě neobsazenou. V případě neplaceného hostingu se obvykle dostane doména rovnou přibalená, přičemž je nabízen výběr z omezeného počtu domén 2. úrovně. K tomu lze doplnit libovolnou neobsazenou doménu 3. úrovně.

Z placeného hostingu lze jmenovat například PIPNI.cz, Forpsi, Superhosting, Webhosting C4 . Z freehostingu to může být Webzdarma nebo Internet Centrum (IC). Existuje samozřejmě mnoho dalších, zaleží jen na jednotlivci, který si je na Internetu vyhledá, případně vybere. [22]

3.3.3 Možnosti přenosu stránek na webový server

3.3.3.1 FTP

Nejčastěji využívanou službou pro přenos souborů je File Transfer Protocol (FTP). Umožňuje přenášet soubory na vzdálený počítač. V tomto případě se jedná o přenos souborů se stránkami z počítače na webový server poskytovatele. Pro použití FTP je nutné,

aby nám poskytovatel přidělil uživatelské jméno a heslo, které chrání přístup na webový server. Pro uskutečnění přenosu stránek se vyžaduje spojení s webovým serverem poskytovatele za pomoci protokolu FTP. Lze využít specializovaného FTP klienta nebo webový prohlížeč. Mezi nejznámější samostatné FTP klienty patří WS_FTP, ale přirozeně existují i další. FTP bývá součástí správců souborů jako je Midnight Commander nebo Windows Commander. Po připojení za pomoci FTP je k dispozici vzdálený adresář, s nímž pracujeme podobně jako s lokálním diskem.

3.3.3.2 FTPS

FTPS je stejné jako FTP, s tím rozdílem, že komunikace se šifruje. Je umožněno šifrovat jakýkoliv aplikační protokol postavený na TCP – nejběžněji se používá ve spojení s HTTP protokolem. Všechny web-hostingové firmy však tento přístup nenabízejí. Důvodem je méně serverů podporujících FTPS než v případě klasického FTP. FTPS podporují nejrozšířenější prohlížeče (IE a Netscape).

3.3.3.3 SCP

Dalším z používaných protokolů je Secure Copy (SCP). Využívá protokol Secure Shell (SSH), který umožňuje zabezpečený přístup na vzdálený počítač. Po připojení SCP klient posílá příkazy pro průchod adresářovou strukturou a pro přenos souborů. SCP je běžně používán na unixových serverech. Existují i implementace pro Windows.

3.3.3.4 Další možnosti

Některé firmy dnes nabízejí bezpečný přístup přes webové rozhraní. Stránky přístupné přes HTTPS obsahují formuláře, které umožňují upload souborů na server. Tento způsob je sice bezpečný, ale může být značně nepohodlný zejména při větším množství souborů. Existuje tudíž možnost přenést celý zazipovaný web a skript, který je na serveru si jej pak automaticky rozbalí do adresářové struktury. [6]

ZÁVĚR

Cílem této bakalářské práce bylo zpracování studijního materiálu pro předmět sloužící k výuce jazyka HTML, úvodu do CSS a základních informací, které se týkají tvorby webových stránek.

V této práci se teoretická a praktická část z pedagogického hlediska vzájemně pojí. Text práce je uzpůsobený uživatelům začátečníkům, kteří ještě nikdy nenabyli žádné zkušenosti s jazykem HTML. Náročnost kapitol se postupně zvyšuje a na konci procesu by měl student prokázat schopnost vytvoření vlastní jednoduché stránky za pomoci jazyka HTML s využitím jednoduchých prvků z oblasti CSS a uplatněním základních sounáležitostí týkajících se web tvorby. Ke každé kapitole jsou vytvořeny prezentace, které doplní hodiny přednášek, vzorové příklady pro pochopení probíraných informací a testovací část se zadáním pro uplatnění v hodinách praktických cvičení. Všechny kódy zapsané v dokumentech jsou zároveň realizovány a k dokumentům přiloženy. Některé z kapitol jsou doplněny o videosekvence z www.youtube.com sloužící k názorné ukázce dané problematiky – viz. přiložené CD. V příloze bakalářské práce je ukázka výukové práce z jedné z probíraných kapitol.

Materiál je časově zpracován pro třináctitýdenní výuku. Dvanáctá hodina výuky v praktických cvičeních je věnována kontrolnímu testu, který prověří praktické i teoretické znalosti studenta získané během výuky. Poslední hodiny cvičení jsou věnovány práci při plnění závěrečného úkolu. Každý student využije možnost vlastní volby a práci, kterou bude prezentovat jako zápočtovou, vypracuje bez cizí pomoci zcela sám.

ZÁVĚR V ANGLIČTINĚ

The objective of the thesis was to create study material for a course teaching HTML, introduction to CSS and basic information concerning web production.

The theoretical and the practical part are mutually connected from pedagogical perspective. Text of the thesis is adjusted to students - beginners, who have not reached any experiences with HTML yet. Difficulty is rising with each chapter and at the end of the process a student should show an ability to create own simple web presentation using HTML, simple parts of CSS and a knowledge about web development. For each chapter there are presentations for lectures, examples for enhancing understanding of the topics and testing part for exercises created. All the codes written in the documents are also realized and attached to the documents. Some of the chapters are supplemented with videos from www.youtube.com to show examples for the topics – see attached CD. An example of teaching work from one of the chapters is in the appendix.

The teaching material is made for 13-weeks teaching period. The 12th lesson of exercise lessons is dedicated to the test, which will examine practical and theoretical knowledge of a student, acquired during the course. Last lessons of exercises are aimed on work completing a final project. Each student will use own choice of project which will be presented as a final project, which will be worked on autonomously.

SEZNAM POUŽITÉ LITERATURY

Monografie:

- [1] BROŽA, Petr. Tvorba WWW stránek pro úplné začátečníky. Praha : Computer Press, 2001. 149 s
- [2] STEJSKAL, Jan. Vytváříme WWW pomocí HTML, CSS a JavaScriptu. Praha : Computer Press, 2006. 250 s.
- [3] KUČERA, Miroslav. HTML - tipy a triky od profesionálů. Brno : UNIS publishing, 2001. 80 s.
- [4] HAUSER, Marianne; HAUSER, Tobias; WENZ, Christian. HTML a CSS - Velká kniha řešení. Praha : Computer Press, 2006. 912 s.
- [5] STANÍČEK, Petr. CSS : Kaskádové styly. Brno : Computer Pres, 2003. 178 s.

Internetové zdroje:

- [6] KOSEK, Jiří. HTML5 [online]. 1997 [cit. 2011-03-13]. Dostupné z WWW: <<http://htmlguru.cz/>>.
- [7] MIKULA, Jan. Prohlížeče.info : internetové prohlížeče a moderní tvorba stránek [online]. 2005 [cit. 2011-04-]. Dostupné z WWW: <<http://prohlizece.info/>>.
- [8] Wikipedia, the free encyclopedia [online]. 2011 [cit. 2011-05-14]. Internet Explorer. Dostupné z WWW: <http://en.wikipedia.org/wiki/Internet_Explorer#Internet_Explorer_10>.
- [9] Oficiální portál pro vývojáře a softwarové inženýry : MSDN Česká republika [online]. 2011 [cit. 2011-04-21]. Microsoft Expression. Dostupné z WWW: <<http://www.microsoft.com/cze/msdn/expression/default.aspx>>.
- [10] MALÝ, Jakub. Jakub Malý : blog [online]. 2007 [cit. 2011-04-21]. Jak se pracuje s Microsoft Expression Web. Dostupné z WWW: <<http://blog.jakubmaly.cz/programovani/webdesign/jak-se-pracuje-s-microsoft-expression-web.aspx>>.
- [11] JANKOVSKÝ, Dušan. Jak psát web [online]. 2005, poslední aktualizace 20.12. 2010 [cit. 2011-02-26]. Dostupné z WWW: [<http://www.jakpsatweb.cz/>].

- [12] Polopatě : naučte se tvořit web správně [online]. 2006, poslední aktualizace 25. 7. 2009 [cit. 2011-03-14]. Dostupné z WWW: <<http://polopate.jakpsatweb.cz/>>.
- [13] WebTvorba [online]. 2004 [cit. 2011-04-03]. CSS kaskádové styly. Dostupné z WWW: <<http://www.webtvorba.cz/css/>>.
- [14] MRÁZEK, Oldřich. HW.cz [online]. 2007 [cit. 2011-05-03]. Textový editor PSPad. Dostupné z WWW: <http://hw.cz/software/pspad/editor_pspad.html>.
- [15] Eclipse : The Eclipse Foundation open source community website [online]. 2011 [cit. 2011-05-03]. Dostupné z WWW: <<http://www.eclipse.org/>>.
- [16] Netbeans [online]. 2011 [cit. 2011-05-03]. Dostupné z WWW: <<http://netbeans.org/>>.
- [17] Projekt CZilla [online]. 2003 [cit. 2011-05-03]. Nvu. Dostupné z WWW: <<http://www.czilla.cz/produkty/nvu/>>.
- [18] NĚMEC, Luboš. Grafika On-line : denní zpravodajství ze světa grafiky, polygrafie a digitální fotografie [online]. 2010-07-09 [cit. 2011-05-04]. WYSIWYG Web Builder: levný vizuální HTML editor. Dostupné z WWW: <<http://www.grafika.cz/art/webdesign/wysiwygwebbuilder657.html>>.
- [19] WYSIWYG Web Builder : webdesign, tvorba webu jednoduše [online]. 2009 [cit. 2011-05-04]. WYSIWYG Web Builder tvorba webových stránek během několika minut. Dostupné z WWW: <<http://www.wysiwygwebbuilder.cz/>>.
- [20] MAIER, Zbyněk. Programy a internet : SWMag.cz [online]. 2010-07-31 [cit. 2011-05-04]. Adobe Dreamweaver CS5 - Návrh, vývoj a správa webů. Dostupné z WWW: <<http://www.swmag.cz/686/adobe-dreamweaver-cs5-navrh-vyvoj-a-sprava-webu/>>.
- [21] POTOČEK , Tobiáš. Vše o HTML, CSS, PHP a JavaScriptu : Webtvorba [online]. 2010-02-08 [cit. 2011-04-30]. Jak na webový design – návod pro líné amatéry. Dostupné z WWW: <<http://webtvorba.howto.cz/jak-na-webovy-design-navod-pro-line-amatery/>>.
- [22] POTOČEK , Tobiáš. Vše o HTML, CSS, PHP a JavaScriptu : Webtvorba [online]. 2009-01-06 [cit. 2011-05-01]. Kam umístit web. Dostupné z WWW: <<http://webtvorba.howto.cz/kam-umistit-web/>>.

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

CERN	Centre Européenne pour la Recherche Nucléaire
CSS	Cascading Style Sheets
DTD	Document Type Definition
FTP	File Transfer Protocol
FTPS	File Transfer Protocol Secure
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
IE	Internet Explorer
NCSA	National Center for Supercomputing Applications
PHP	Hypertext Preprocessor
SCP	Secure Copy
SGML	Standard Generalized Markup Language
SQL	Structured Query Language
TCP	Transmission Control Protocol
W3C	World Wide Web Consortium
WHATWG	The Web Hypertext Application Technology Working Group
WWW	World Wide Web
XHTML	eXtensible HyperText Markup Language
XML	eXtensible Markup Language

SEZNAM OBRÁZKŮ

Obr. 1. Prostředí programu MS Expression Web 4	22
Obr. 2. Prostředí programu PSPad	75
Obr. 3. Prostředí programu Eclipse	76
Obr. 4. Prostředí programu Netbeans IDE	77
Obr. 5. Prostředí programu Nvu 1.0	78
Obr. 6. Prostředí programu WYSIWYG Web Builder	79
Obr. 7. Prostředí programu Adobe Dreamweaver	80

SEZNAM TABULEK

Tab. 1. Velikost písma 66

Tab. 2. Podpora velikosti písma v prohlížečích..... 67


SEZNAM PŘÍLOH


Příloha PI. Ukázka vytvořené výukové práce

Příloha PII. CD disk obsahující:

- Podklady k přednáškám ve formě prezentací PowerPoint a ve formátu PDF
- Vzorové příklady k jednotlivým kapitolám
- Zadání jednotlivých cvičení

PŘÍLOHA P I: UKÁZKA VYTVOŘENÉ VÝUKOVÉ PRÁCE




 Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Seznamy


Přednáška č. x

Tvorba HTML pro administrativní praxi



Přehled tématu

- 1. Netříděný seznam**
 - 1.1 Netříděné seznamy
 - 1.2 Víceúrovňové seznamy
- 2. Tříděný seznam**
- 3. Seznam definic**



2

1. Netříděný seznam

1.1 Netříděný seznam

Jedná se o výčet, kde položky nejsou závislé na pořadí, v jakém jsou uvedeny. Příkladem může být nákupní seznam či seznam úkolů. Nezáleží na pořadí v jakém jsou položky uvedeny.

Musím nakoupit:

- rajčata
- mléko
- jablka
- chléb

3

1. Netříděný seznam

Pro netříděný seznam má HTML k dispozici párovou značku ``.
Jednotlivé položky seznamu jsou pak uvezeny tagem ``.
Tato značka nemá ukončovací protějšek.

Parametry:

PARAMETR	HODNOTA	DALŠÍ NASTAVENÍ
TYPE	disk circle square	

Syntaxe:

```
<UL parametry>  
  <LI>  
  ...  
</UL>
```

4

1. Netříděný seznam

Příklad:

Musím nakoupit:

rajčata

mléko

jablka

chléb

5

1. Netříděný seznam

Příklad:

<UL TYPE="disc">

Seznam 1 - plné kolečko jako symbol

<UL TYPE="circle">

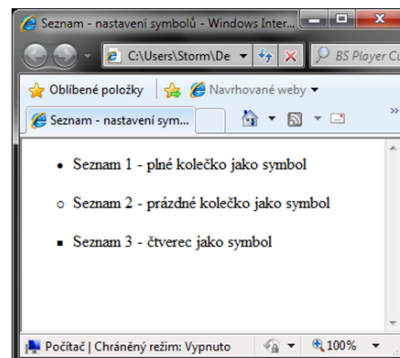
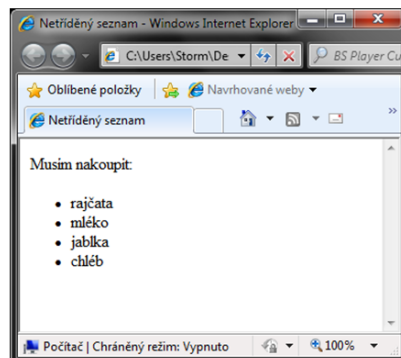
Seznam 2 - prázdné kolečko jako symbol

<UL TYPE="square">

Seznam 3 - čtverec jako symbol

6

1. Netříděný seznam



7

1. Netříděný seznam

1.2 Víceúrovňové seznamy

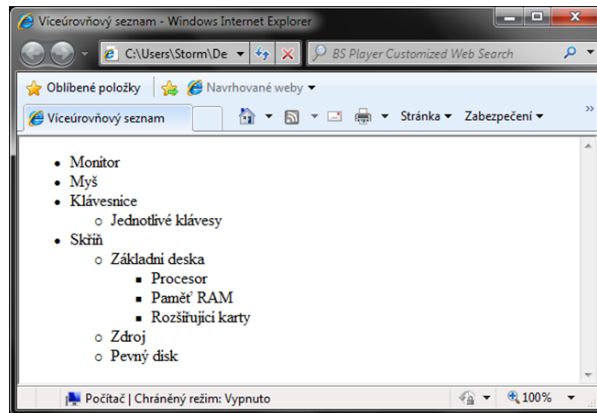
Jednotlivé položky seznamu mají další podpoložky, ty pak mají další podpoložky atd. Víceúrovňový seznam se vytvoří logickou kombinací značek pro tvorbu netříděného seznamu (viz Vzorové příklady).

Například:

- výčet jednotlivých částí těla
- části PC

8

1. Netříděný seznam



Prohlížeč odsazuje jednotlivé úrovně směrem doprava a přiřazuje různé odrážky v podobě puntíku, prázdného kolečka a čtverečku.

9

2. Tříděný seznam

Záleží na pořadí položek seznamu a nelze je libovolně zaměňovat.

Například:

- kuchařský recept

Tříděný seznam se vytváří stejným způsobem jak netříděný.

S tím **ROZDÍLEM**, že se značka

``

zamění za značku

``.

Jednotlivé položky jsou v seznamu číslovány .

10

2. Tříděný seznam

Neexistuje omezení pouze na číslovaný seznam.

K změně tohoto parametru se používá *TYPE*

Parametry:

PARAMETR	HODNOTA	DALŠÍ NASTAVENÍ
TYPE	a	
	A	
	i	
	I	
	1	

Syntaxe

<OL parametr = hodnota>

11

2. Tříděný seznam

Příklad:

Jak udělat řízek:

Nachystáme si maso, které má být podáváno, mouku, vejce a strouhanku.

Maso obalíme nejdříve v mouce, pak ve vejci a nakonec ve strouhance.

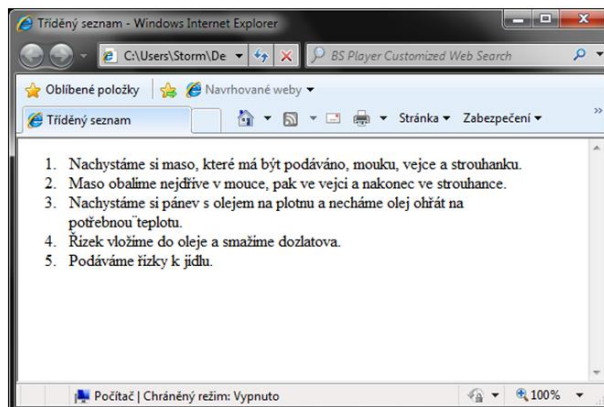
Nachystáme si pánev s olejem na plotnu a necháme olej ohřát na potřebnou teplotu.

Řízek vložíme do oleje a smažíme dozlatova.

Podáváme řízky k jídlu.

12

2. Tříděný seznam



13

3. Seznam definic

Definiční výčet se používá při vysvětlování jednotlivých pojmů.

Pro vytvoření se používají značky

`<DL>`, `<DT>` a `<DD>`

jejíž použití je velmi podobné jako v případě tříděného a netříděného seznamu.

14

3. Seznam definic

Syntaxe:

<DL>

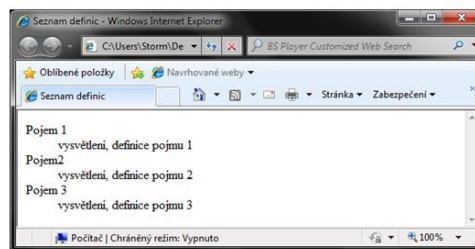
<DT> Pojem 1

<DD> vysvětlení, definice pojmu 1

<DT> Pojem2

<DD> vysvětlení, definice pojmu 2

</DL>



Vzorové příklady

Příklad č. 1

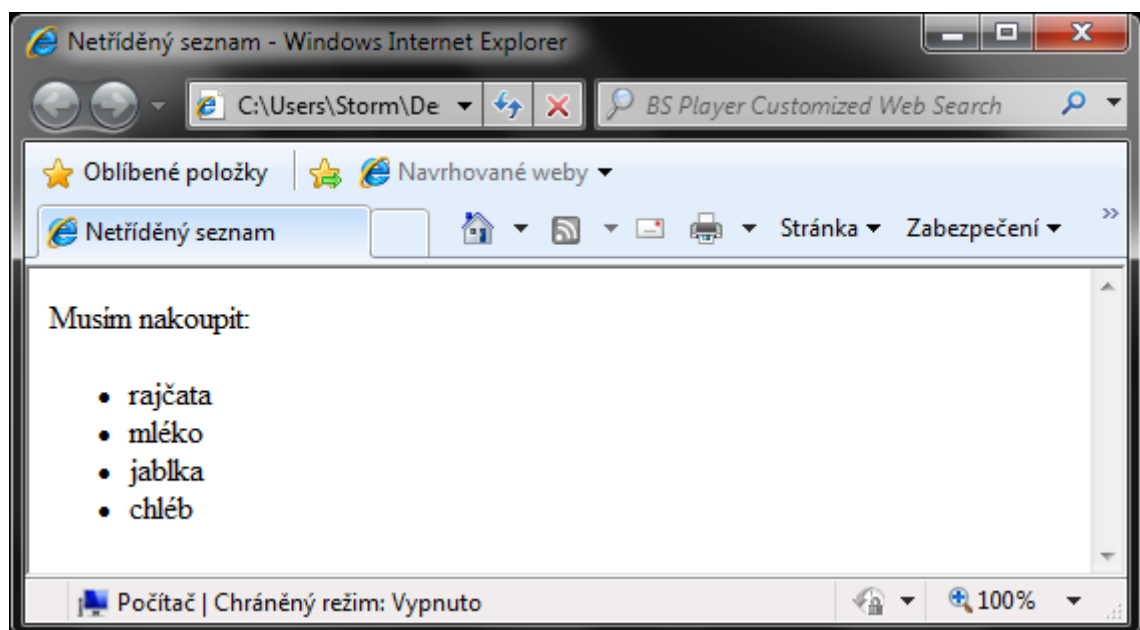
Musím nakoupit:

rajčata

mléko

jablka

chléb



Příklad č. 2

```
<UL TYPE="disc">
```

```
    <LI>Seznam 1 - plné kolečko jako symbol
```

```
</UL>
```

```
<UL TYPE="circle">
```

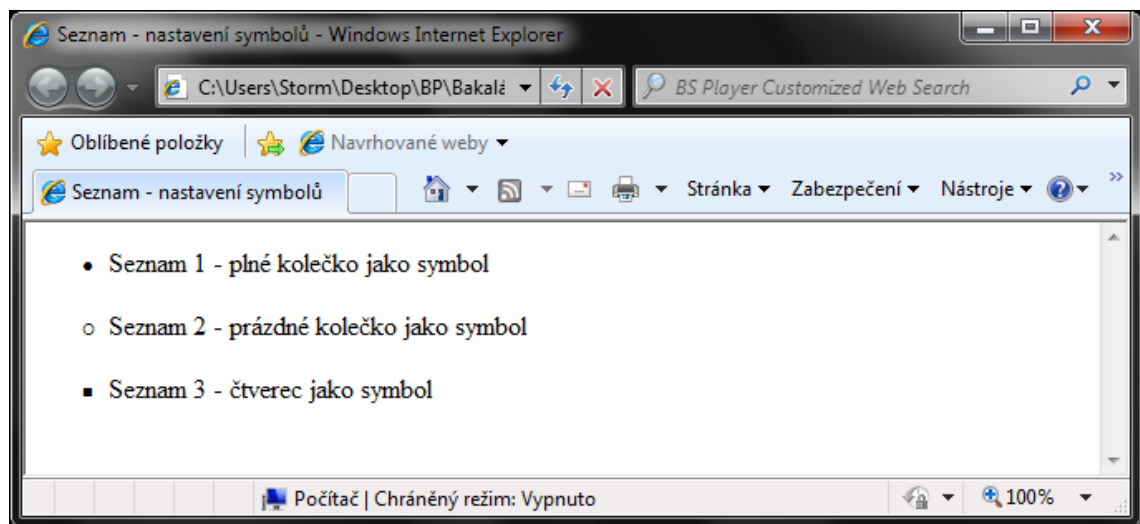
```
    <LI>Seznam 2 - prázdné kolečko jako symbol
```

```
</UL>
```

```
<UL TYPE="square">
```

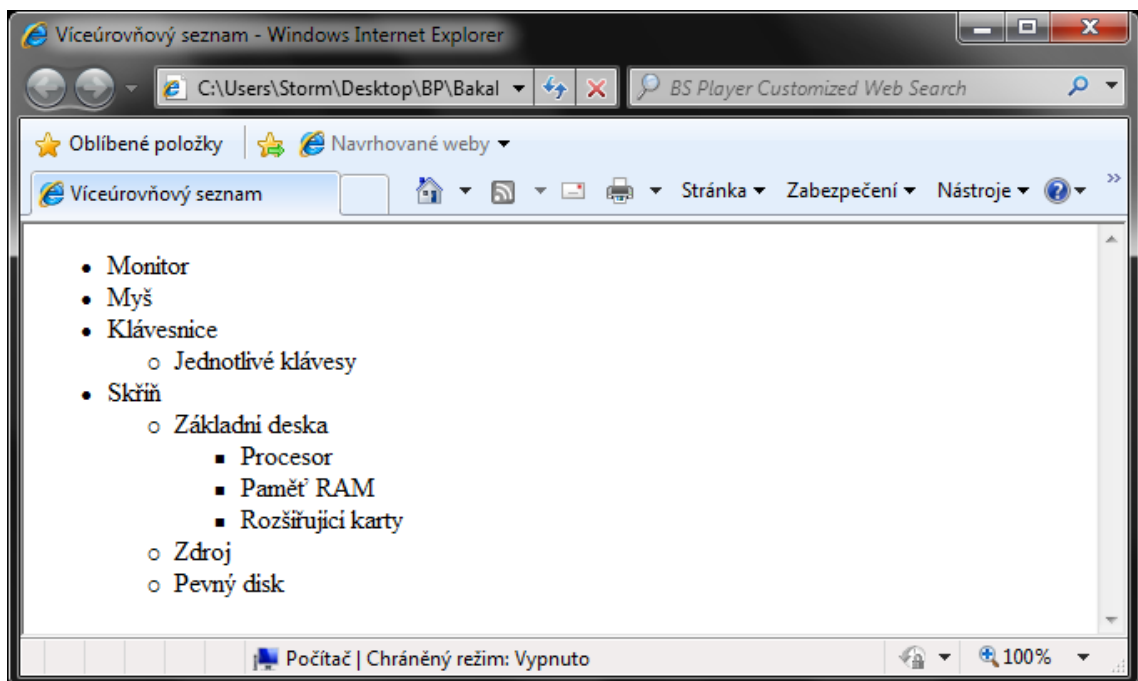
```
    <LI>Seznam 3 - čtverec jako symbol
```

```
</UL>
```



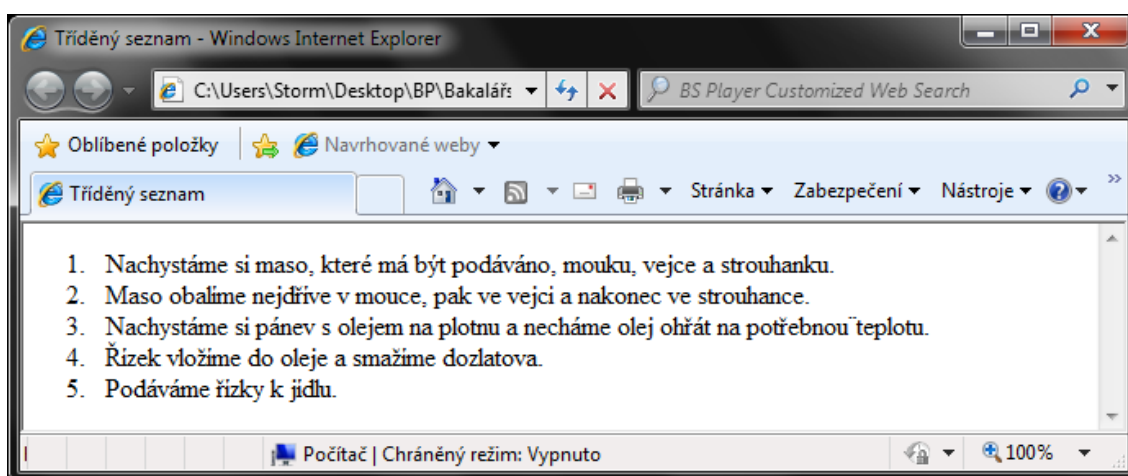
Příklad č. 3

```
<UL>
  <LI>Monitor
  <LI>Myš
  <LI>Klávesnice
  <UL>
    <LI>Jednotlivé klávesy
  </UL>
  <LI>Skříň
  <UL>
    <LI>Základní deska
    <UL>
      <LI> Procesor
      <LI> Paměť RAM
      <LI> Rozšiřující karty
    </UL>
    <LI>Zdroj
    <LI>Pevný disk
  </UL>
</UL>
```



Příklad č. 4

- Nachystáme si maso, které má být podáváno, mouku, vejce a strouhanku.
- Maso obalíme nejdříve v mouce, pak ve vejci a nakonec ve strouhance.
- Nachystáme si pánev s olejem na plotnu a necháme olej ohřát na potřebnou teplotu.
- Řízek vložíme do oleje a smažíme dozlatova.
- Podáváme řízky k jídlu.



Příklad č. 5

<DL>

<DT> Pojem 1

<DD> vysvětlení, definice pojmu 1

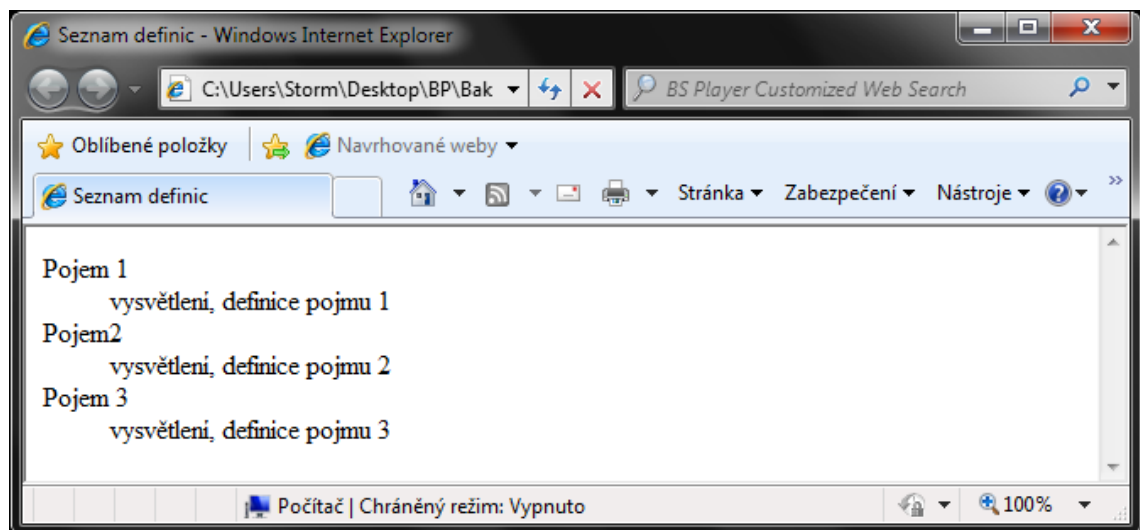
<DT> Pojem2

<DD> vysvětlení, definice pojmu 2

<DT> Pojem 3

<DD> vysvětlení, definice pojmu 3

</DL>



Seznamy – CVIČENÍ

V kapitole popsané tagy, parametry a hodnoty

TAG	PARAMETR	HODNOTA	SYNTAXE
UL	TYPE	disk	<ul type="disk">
		circle	<ul type="circle">
		square	<ul type="square">
OL	TYPE	A	<ol type="A">
		a	<ol type="a">
		I	<ol type="I">
		i	<ol type="i">
		1	<ol type="1">
LI	TYPE	A	<li type="A">
		a	<li type="a">
		I	<li type="I">
		i	<li type="i">
		1	<li type="1">
DL			<dl>
DT			<dt>pojem
DD			<dd>vysvětlení pojmu
			</dl>

Zadání:

Vytvořte HTML stránku **seznamy.html** :

1. Nadpisem H1 vycentrovaným na střed strany libovolné barvy napište: TVORBA SEZNAMŮ
2. Barvu pozadí a font písma libovolně navolte. V titulku bude napsáno Seznamy.
3. Nadpisem H2 napište **Netříděný seznam** a vytvořte tyto seznamy:

Moje oblíbené předměty:

a uveďte min. 4 předměty – použijte odrážku plné kolečko (DISC)

Moje neoblíbené předměty:

a uveďte min. 4 předměty – použijte odrážku čtvereček (SQUARE)

Moje oblíbené prohlížeče:

a uveďte min. 4 prohlížeče – použijte odrážku kolečko (CIRCLE)

4. Nadpisem H2 napište **Tříděný seznam** vytvořte tyto seznamy:

Pořadí ovoce podle prodejnosti:

1. jablka
2. pomeranče
3. banány
4. mandarinky

V dalších seznamech použijte číslování:

a., b., c., d.
i., ii., iii., iv.

5. Nadpisem H2 napište **Seznam definicí** vytvořte tento definiční seznam:

HTTP

Transportní protokol využívaný k přenosu souborů obsahujících popis WWW-stránek v jazyce HTML

FTP

Transportní protokol používaný k přenosu souborů

NNTP

Transportní protokol používaný k přenosu news

6. Nadpisem H2 napište **Vnořené seznamy** a bude obsahovat tento seznam:
Pro tvorbu stránek v jazyce HTML můžeme použít:

1. obyčejné textové editory
 - Notepad
 - Programmer's File Editor
 - Emacs

2. Textové editory podporující vytváření struktury HTML dokumentu
 - HotMetalPro
 - AsWedit
 3. WYSIWYG editory HTML stránek
 - FrontPage
 - Netscape Editor
 - WebMagic Pro
7. Vaši hotovou práci uložte na www.vyuka.fai.utb.cz.