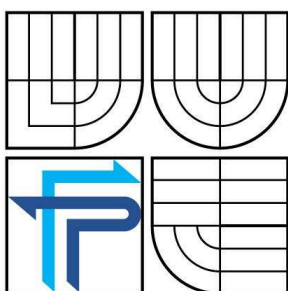


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA PODNIKATELSKÁ
ÚSTAV INFORMATIKY

FACULTY OF BUSINESS AND MANAGEMENT
INSTITUTE OF INFORMATICS

NÁVRH INTERNETOVÝCH STRÁNEK

WEB PAGES DESIGN

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

JAKUB ŠŤASTNÝ

VEDOUCÍ PRÁCE

SUPERVISOR

doc. Ing. MILOŠ KOCH, CSc.

BRNO 2008

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Šťastný Jakub

Manažerská informatika (6209R021)

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách, Studijním a zkušebním řádem VUT v Brně a Směrnicí děkana pro realizaci bakalářských a magisterských studijních programů zadává bakalářskou práci s názvem:

Návrh internetových stránek

v anglickém jazyce:

Web pages design

Pokyny pro vypracování:

Úvod

Teoretická východiska práce

Analýza problému a současné situace

Vlastní návrhy řešení

Závěr

Seznam použité literatury

Přílohy

Podle § 60 zákona č. 121/2000 Sb. (autorský zákon) v platném znění, je tato práce "Školním dílem". Využití této práce se řídí právním režimem autorského zákona. Citace povoluje Fakulta podnikatelská Vysokého učení technického v Brně. Podmínkou externího využití této práce je uzavření "Licenční smlouvy" dle autorského zákona.

Seznam odborné literatury:

KOSEK, J.: HTML. Tvorba dokonalých WWW stránek, Grada, 1998, ISBN 80-7169-608-0


ECKER, C.: Profesionální webdesign : techniky a vzorová řešení. 1. vyd. Brno: CP Books, 2005. 421 s. ISBN: 80-251-0547-4.


WELLING, L. a THOMSON, L.: PHP a MySQL - rozvoj webových aplikací. Praha: SoftPress, 2002. 720 s. ISBN 80-86497-20-8.

Vedoucí bakalářské práce: doc. Ing. Miloš Koch, CSc.

Termín odevzdání bakalářské práce je stanoven časovým plánem akademického roku 2007/08.




Ing. Jiří Kříž, Ph.D.
Ředitel ústavu


doc. Ing. Miloš Koch, CSc.
Děkan fakulty

V Brně, dne 15.2.2008

LICENČNÍ SMLOUVA

POSKYTOVANÁ K VÝKONU PRÁVA UŽÍT ŠKOLNÍ DÍLO

uzavřená mezi smluvními stranami:

1. Pan/paní

Jméno a příjmení: Jakub Šťastný
Bytem: Mikulčická 1072/16, 62700, Brno - Slatina
Narozen/a (datum a místo): 23.7.1984, Ivančice

(dále jen "autor")

a

2. Vysoké učení technické v Brně

Fakulta podnikatelská
se sídlem Kolejní 2906/4, 61200 Brno 12
jejímž jménem jedná na základě písemného pověření děkanem fakulty:
Ing. Jiří Kříž, Ph.D.

(dále jen "nabyvatel")

Článek 1

Specifikace školního díla

1. Předmětem této smlouvy je vysokoškolská kvalifikační práce (VŠKP):

- disertační práce
- diplomová práce
- bakalářská práce

jiná práce, jejíž druh je specifikován jako

(dále jen VŠKP nebo dílo)

Název VŠKP: Návrh internetových stránek

Vedoucí/školicel VŠKP: doc. Ing. Miloš Koch, CSc.

Ústav: Ústav informatiky

Datum obhajoby VŠKP:

VŠKP odevzdal autor nabyvateli v:

- tištěné formě - počet exemplářů 1
- elektronické formě - počet exemplářů 1

2. Autor prohlašuje, že vytvořil samostatnou vlastní tvůrčí činností dílo shora popsané a specifikované. Autor dále prohlašuje, že při zpracovávání díla se sám nedostal do rozporu s autorským zákonem a předpisy souvisejícími a že je dílo dílem původním.
3. Dílo je chráněno jako dílo dle autorského zákona v platném znění.
4. Autor potvrzuje, že listinná a elektronická verze díla je identická.

Článek 2
Udělení licenčního oprávnění

1. Autor touto smlouvou poskytuje nabyvateli oprávnění (licenci) k výkonu práva uvedené dílo nevýdělečně užít, archivovat a zpřístupnit ke studijním, výukovým a výzkumným účelům včetně pořizování výpisů, opisů a rozmnoženin.
2. Licence je poskytována celosvětově, pro celou dobu trvání autorských a majetkových práv k dílu.
3. Autor souhlasí se zveřejněním díla v databázi přístupné v mezinárodní síti
 - ihned po uzavření této smlouvy
 - 1 rok po uzavření této smlouvy
 - 3 roky po uzavření této smlouvy
 - 5 let po uzavření této smlouvy
 - 10 let po uzavření této smlouvy(z důvodu utajení v něm obsažených informací)
4. Nevýdělečné zveřejňování díla nabyvatelem v souladu s ustanovením § 47b zákona č. 111/1998 Sb., v platném znění, nevyžaduje licenci a nabyvatel je k němu povinen a oprávněn ze zákona.

Článek 3
Závěrečná ustanovení

1. Smlouva je sepsána ve třech vyhotoveních s platností originálu, přičemž po jednom vyhotovení obdrží autor a nabyvatel, další vyhotovení je vloženo do VŠKP.
2. Vztahy mezi smluvními stranami vzniklé a neupravené touto smlouvou se řídí autorským zákonem, občanským zákoníkem, vysokoškolským zákonem, zákonem o archivnictví, v platném znění a popř. dalšími právními předpisy.
3. Licenční smlouva byla uzavřena na základě svobodné a pravé vůle smluvních stran, s plným porozuměním jejímu textu i důsledkům, nikoliv v tísní a za nápadně nevýhodných podmínek.
4. Licenční smlouva nabývá platnosti a účinnosti dnem jejího podpisu oběma smluvními stranami.

V Brně dne:

.....

Nabyvatel

.....

Autor

Abstrakt

Hlavním cílem bakalářské práce je navrhnout a realizovat reprezentativní webové stránky pro stavební firmu APMB s.r.o. s nulovými vstupními náklady na použitý software. Součástí práce je také návrh firemního loga.

Klíčová slova

WWW, HTTP, HTML, XHTML, SEO, kaskádové styly, Internet, typografie na Internetu, web design

Abstract

The main goal of the bachelor's thesis is to design and create representative web pages for construction company APMB s.r.o. without any spending on software.

The work including concept of company's logo.

Key words

WWW, HTTP, HTML, XHTML, SEO, Cascading Style Sheets, Internet, typography on Internet, web design

Bibliografická citace práce:

ŠŤASTNÝ, J. *Návrh internetových stránek*. Brno: Vysoké učení technické v Brně, Fakulta podnikatelská, 2008. 49 s. Vedoucí bakalářské práce
doc. Ing. Miloš Koch, CSc.

Čestné prohlášení

Prohlašuji, že předložená bakalářská práce je původní a zpracoval jsem ji samostatně. Prohlašuji, že citace použitých pramenů je úplná, že jsem ve své práci neporušil autorská práva (ve smyslu Zákona č. 121/2000 Sb., o právu autorském a o právech souvisejících s právem autorským).

V Brně dne 17. května 2008

.....
Jakub Šťastný

Obsah:

Úvod	9
1. Vymezení problému a cíle práce	10
1.1. Vymezení problému	10
1.2. Hlavní cíl práce	10
1.3. Dílčí cíle práce	10
2. Teoretická východiska práce	11
2.1. Protokol HTTP	11
2.2. Značkovací jazyk HTML	12
2.2.1. Vývoj	13
2.2.2. Chronologický vývoj jazyka	13
2.2.3. HTML značky	15
2.2.4. Typy značkovacích elementů používaných HTML	16
2.3. CSS	17
2.3.1. Historie	18
2.3.2. CSS syntaxe	19
2.3.3. Jednoduché a kombinované selektory	22
2.3.4. Univerzální selektory	23
2.3.5. Selektor přiřazený konkrétnímu typu elementu (type selector)	23
2.3.6. Třídní selektor (class selector)	24
2.3.7. ID selektor	24
2.3.8. Zdroje stylového předpisu	25
2.4. Implementace	26
2.5. Použitelnost webové prezentace	28
2.6. SEO (Search Engine Optimization – optimalizace pro vyhledávače)	29
2.6.1. Faktory optimalizace	29
2.6.2. Klíčová slova	29
2.6.3. Pagerank	30
2.7. Typografie na webu	30
2.7.1. Výčet základních pravidel, která jsou často opomíjena	30
3. Analýza problému a současné situace	33
3.1. Volba nástrojů	33
3.1.1. Software využitý v projektu	33
4. Vlastní návrhy řešení	34
4.1. Logo firmy	34
4.1.1. Návrhy loga	34
4.2. Struktura webu	35
4.3. Technické provedení – rozvržení stránky – layout	36
4.4. Přístupnost stránek	39
4.4.1. Dekompozice grafického návrhu	40
4.4.2. Implementace grafických prvků	41
4.4.3. Obsah zdrojového kódu	43
5. Závěr	46

6.	<i>Seznam použité literatury</i>	47
6.1.	Knihy	47
6.2.	Internetové adresy	47
7.	<i>Seznamy</i>	48
7.1.	Seznam obrázků	48
7.2.	Seznam tabulek	48
8.	<i>Přílohy</i>	49

Úvod

Svět informačních technologií jde stále kupředu. Internet se za relativně krátkou dobu své existence stal pro většinu lidí nejrychlejším a nejvyhledávanějším nástrojem k získávání informací (podle Internet World Stats využívá v současnosti Internet asi 1,319 miliardy lidí¹). Jeho pole působnosti se postupem času stále rozšiřovalo z čistě akademické půdy i do komerčních oblastí.

V současné době velké množství firem, a to i těch malých, nabízí své produkty a služby právě prostřednictvím Internetu. Základem toho, aby byla jejich prezentace úspěšná, tedy především oslovila návštěvníky stránek – potencionální zákazníky, musí splňovat mnoho parametrů, především: být vizuálně přívětivá, přehledná, na webu lehce dohledatelná a přístupná.

Ke splnění těchto vlastností správné prezentace je nezbytné dodržení mnoha pravidel pro jejich správnou tvorbu, kterými je třeba se při návrhu webu řídit.

A právě jejich analýzou se má práce zabývat. Důraz je kladen především na pravidla správného vzhledu stránek, tedy webdesignu.

Výstupní hodnotou této práce bude plně validní, webová prezentace pro stavební společnost APMB, s.r.o.

¹Internet World Stats [online]. 2008 [cit. 2008-04-22].
Dostupný z WWW: <<http://www.internetworldstats.com/stats.htm>>

1. Vymezení problému a cíle práce

1.1. Vymezení problému

Při návrhu a realizaci stránek je třeba dbát na přívětivý design, intuitivní ovládací prvky, validnost (správnost) zdrojových kódů dle směrnic a také správnou volbu klíčových slov (z důvodu snadné dohledatelnosti pomocí vyhledávacích algoritmů).

Jedním z problémů v samotných počátcích projektu je volba nástrojů, jež budou pro jeho tvorbu použity. Především jejich cena (zejména cena grafického softwaru se pohybuje v desítkách tisíc korun). Z tohoto důvodu byly zvoleny free software alternativy, které jsou nabízeny k užití zcela zdarma.

1.2. Hlavní cíl práce

Výstupem práce budou korektní webové stránky pro stavební společnost A.P.M.B, s.r.o. Ta v současné době žádnou prezentaci na Internetu nemá.

1.3. Dílčí cíle práce

Poukázat na důležitost dodržování standardů, časté chyby a zastaralé techniky, které, někteří při výstavbě stránek doposud používají a zdůraznit nutnost, často opomíjené, typografie.

2. Teoretická východiska práce

2.1. Protokol HTTP

Hypertext Transfer Protocol (HTTP) je komunikační protokol sloužící k přenosu informací na intranetu a WWW.

Jeho původní smysl byl poskytnout způsob k publikování a získávání hypertextových stránek pomocí Internetu.

Vývoj HTTP protokolu byl koordinován konsorciem W3C (World Wide Web Consortium) a IETF.

HTTP je standard typu **žádost/odpověď** mezi klientem a serverem. Klient je koncový uživatel, server webová stránka. Klient jež provede HTTP požadavek (request), užitím webového prohlížeče nebo end-user aplikace, je označován jako **user agent**.

Odpovídající server (který ukládá nebo vytváří zdroje jako HTML soubory a obrázky) je označován jako **origin server** (zdrojový server).

Mezi user agentem a origin serverem může existovat několik prostředníků, jako např. proxy, brány (gateway) a tunely.

HTTP není omezeno jen na používání aplikační vrstvy protokolu TCP/IP (na Internetu je jako nosná vlna pro HTTP používán nejvíce právě TCP/IP).

HTTP může být implementován na základě jakéhokoliv jiného protokolu používaného na Internetu nebo jiných sítích, avšak požaduje spolehlivý způsob transportu, tedy každý protokol, jež splňuje tuto podmínku, může být využit.

HTTP klient zahájí komunikaci odesláním požadavku (request). Je zahájeno TCP spojení (Transmission Control Protocol) na určitý port na vzdáleném serveru (standardně na port 80).

HTTP server naslouchá na tomto portu čekajíc na request od klientů. Po obdržení požadavku server odpoví – odešle klientovi status line – např. „HTTP/1.1 200 OK“ a samotnou zprávu (jejíž tělo je požadovaný soubor nebo nějaká jiná informace). (11)

Důvod, proč HTTP používá TCP místo UDP, je ten, že pro přenos webové stránky musí být odesláno a v pořádku přijato relativně velké množství dat. TCP protokol na rozdíl od UDP řídí datový tok, předkládá data ve správném pořadí a poskytuje korekci chyb.

Zdroj dat, který má být přístupný prostřednictvím HTTP je definován pomocí URL (Uniform Resource Identifier).

Pro zabezpečení HTTP v současné době existují dvě varianty:

- ⇒ *Schéma HTTPS (Hypertext Transfer Protocol over Secure Socket Layer) URI*: syntakticky identické prostému HTTP, které je používáno pro obvyklé spojení. Na rozdíl od HTTP však navíc signalizuje prohlížeči, aby použil šifrovanou vrstvu vloženou mezi HTTP a TCP, tedy SSL (Secure Sockets Layer) nebo TLS (Transport Layer Security) za účelem zabezpečení. SSL je pro HTTP obzvláště vhodné, protože poskytuje ochranu i v případě autentizace pouze jedné ze stran účastnících se komunikace. Toto je přesně případ HTTP zpráv prostřednictvím Internetu, kde se prokazuje pouze server (příčemž klient potvrdí serverem zasláný certifikát). Při spojení užitím HTTPS je používán standardně port 443.
- ⇒ *Vylepšenou HTTP 1.1 hlavičkou (HTTP 1.1 Upgrade header)*: Klient zahajuje komunikaci čistě textovým požadavkem (request), který je později povýšen na TLS. Klient nebo i server může požádat o povýšení komunikace na tuto vrstvu. Nejběžnější použití je vyslání čistě textové zprávy klientem, které je postoupeno požadavkem serveru o převod na zabezpečenou úroveň komunikace. (11)

2.2. Značkovací jazyk HTML

HTML je zkratka pro HyperText Markup Language, převládající značkovací jazyk pro webové stránky. Jeho pomocí lze definovat strukturu, v základu textových, informací v dokumentu – jednak označením určitého textu jako odkazu, nadpisu, seznamu apod. a včleněním interaktivních formulářů, obrázků a ostatních objektů do takového textu.

HTML je psáno ve formě značek (tagů), které jsou uzavřeny v hranatých závorkách.

Může, v určitých mezích, definovat vzhled a sémantiku dokumentu. Taktéž umožňuje zahrnutí kódů skriptovacích jazyků (takovým jazykem může být např. JavaScript), které ovlivní chování webových prohlížečů a dalších programů pro zpracování HTML. Dle úmluvy používají soubory typu html příponu .html nebo .htm. (10)

2.2.1. Vývoj

V roce 1980 Tim Berners-Lee, který byl nezávislý zaměstnanec CERN (European Organization for Nuclear Research) navrhl a vytvořil ENQUIRE, systém, který sloužil výzkumníkům CERNu ke sdílení dokumentů.

V roce 1989, Berners-Lee a systémový inženýr CERNu Robert Cailliau předložili samostatné návrhy na hypertext, který poskytoval podobnou funkcionalitu.

Následující roky spolupracovali na společném návrhu - WorldWideWeb (W3).

První veřejně dostupný popis HTML byl dokument nazývaný "HTML Tags", poprvé uveřejněný na internetu Bernersem-Lee v roce 1991. Tento dokument popisoval 22 elementů tvořících relativně jednoduchý, design HTML. Třináct z nich stále existuje v HTML verze 4. (10)

2.2.2. Chronologický vývoj jazyka

⇒ *Červenec, 1993*: Hypertext Markup Language bylo publikováno IETF jako pracovní návrh (v této době se zatím nejednalo o standard).

⇒ *Listopad, 1995*: Komunitou IETF bylo vydáno HTML 2.0 jako RFC (Request for Comments – série dokumentů zahrnující nový výzkum) – konkrétně:

⇒ RFC 1866

⇒ RFC 1867 – vyšlo tentýž měsíc co RFC 1866, přidává interaktivní formuláře

⇒ RFC 1942 – přidána schopnost vytvářet tabulky, květen 1996

⇒ RFC 2080 – podpora grafiky, srpen 1996

⇒ RFC 2070 – mezinárodní lokalizace, leden 1997

Nakonec byly v červnu 2000 podle RFC 2854 všechny deklarovány jako zastaralé.

V dubnu 1995 byl Davem Raggettem předložen IETF a nově vytvořené W3C standard HTML 3.0. Ten navrhoval nové schopnosti, jako např. zobrazení komplexních matematických elementů. I přesto, že byl designován jako zpětně kompatibilní s HTML 2.0, byl tehdy příliš komplikovaný na to, aby mohl být implementován. Vývojáři prohlížečů zvolili pouze částečnou podporu návrhu. Na druhou stranu integrovali ostatní konstrukce, které si přáli včlenit i do standardu.

Když platnost návrhu vyprchala v září 1995, práce v tomto směru byla přerušena kvůli nízké podpoře prohlížeči. (10)

HTML 3.1 nebylo nikdy oficiálně navrženo a další návrh standardu vyšel s označením HTML 3.2 (kódové označení "Wilbur"). Tato verze se vzdala většiny nových vlastností přinášených HTML 3.0 a místo toho přijala mnoho elementů a vlastností, které byly vytvořeny vývojáři pro prohlížeče Netscape a Mosaic.

⇒ *14. ledna, 1997*: bylo jako doporučení W3C publikováno HTML 3.2

HTML 3.2 nebylo nikdy předloženo IETF, jejíž HTML pracovní skupina byla ukončena v září 1996. Místo toho bylo publikováno, jako vůbec jedno z prvních, W3C "doporučení" v roce 1997. Podpora komplexních matematických elementů přišla o několik let později s odlišným standardem MathML (Mathematical Markup Language). (10)

⇒ *18. prosince, 1997*: vyšlo HTML 4.0, jako doporučení W3C, které nabízelo hned tři varianty:

Strict – ve kterém byly zastaralé elementy zakázány;

Transitional – ve kterém byly naopak zastaralé elementy povoleny;

Frameset – ve kterém jsou povoleny elementy mající vazbu na rámce (frames).

HTML 4.0 (původně pod pracovním názvem "Cougar" - puma) rovněž adoptoval mnoho elementů a atributů specifikovaných prohlížeči, ale ve stejné době také začal "čistit" standard pomocí označování některých elementů jako neschválených a vyzýváním k jejich nepoužívání. (10)

- ⇒ 24. dubna, 1998: Standard doznal mírné ediční revize bez zvýšení čísla verze.
- ⇒ 24. prosince, 1999: Další minoritní revize standardu. Vzniká HTML 4.1. Nabízí tytéž varianty jako HTML 4.0. Její poslední úprava byla publikována 12. května 2001.
- ⇒ 15. května 2000: je publikováno ISO/IEC 15445:2000 ("ISO HTML", založeno na HTML 4.01 Strict) jako ISO/IEC mezinárodní standard.
- ⇒ 22. ledna 2008: W3C vydává, jako pracovní návrh, HTML 5

2.2.3. HTML značky

⇒ *Elementy*

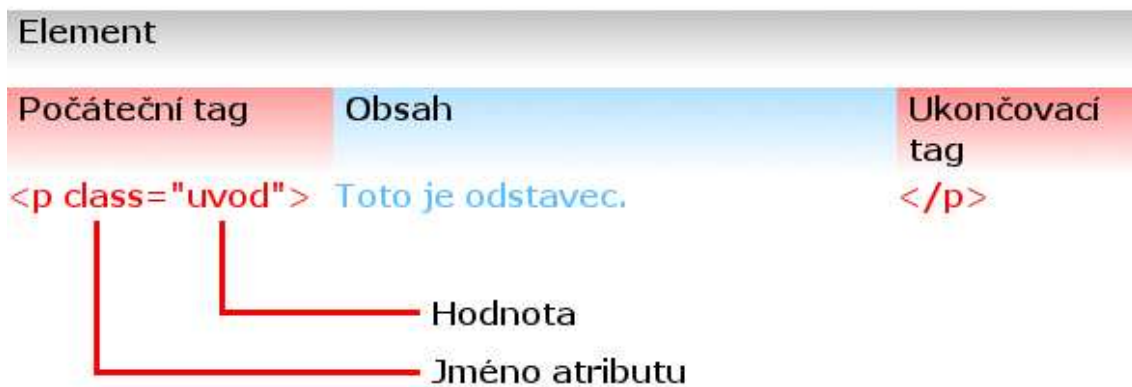
Představují základní strukturu pro značení HTML. Mají dvě hlavní vlastnosti: atribut a obsah. Každý atribut i každý obsah elementu má jistá omezení, která musí být splněna, aby mohl být HTML dokument považován za správně napsaný (validní).

Element má obvykle:

- ⇒ počáteční tag <jmeno-elementu>
- ⇒ ukončovací tag </jmeno-elementu>

Atributy elementů jsou obsaženy v počátečním tagu a obsah je umístěn mezi počáteční a ukončovací tag. Například:

```
<jmeno-elementu atribut="hodnota">Obsah</jmeno-elementu>
```



Obrázek 1: syntaxe HTML tagu

Některé elementy, jako třeba `
` (definující zalomení konce řádku), nemají žádný obsah a nesmějí mít ukončovací tag.

2.2.4. Typy značkovacích elementů používaných HTML

⇒ *Strukturální* – strukturální značky popisují účel textu. Například `<h3>Brno</h3>` definuje "Brno" jako nadpis třetí úrovně, který by měl být prohlížečem také takto interpretován. Strukturální značky neoznačují žádný specifický druh interpretace, avšak většina webových prohlížečů má standardy popisující, jak by měly být elementy formátovány. Text může být dále stylován za pomoci kaskádových stylů (CSC – Cascading Style Sheets). (10)

⇒ *Prezentační* – prezentační značky popisují vzhled textu, nehledě na jeho význam.

Např. `tučně` značí, že grafické výstupní zařízení (monitor, tiskárna...) by mělo vyjádřit "tučně" tučným textem, avšak nedává žádné náznaky toho, co by mělo provádět zařízení, které takového výstupu schopno není (např. pro zrakově postižené – tedy přístroje, které předčítají obsah dokumentu nahlas). Vedle `tučně`, `<i>kurzívou</i>` a jim podobných existují také elementy takové, které poskytnou ekvivalentní vizuální vzezření jimi definovaného textu (stejně jako ty, právě zmiňované), ale ve své podstatě jsou mnohem více sémantické, konkrétně např. `Na tuto větu je kladen důraz.` nebo

` důraz `. Většina prezentačních značkových elementů nebyla ve specifikaci HTML 4.0 schválena ve prospěch na CSS založeném designu. (10)

2.3. CSS

CSS (Cascading Style Sheets) je jazyk používaný pro popis dokumentu napsaném za pomoci značkovacího jazyka. Nejčastěji se užívá pro definování stylu webových stránek psaných v HTML či XHTML. Avšak tento jazyk může být aplikován na jakýkoliv druh XML dokumentu, zahrnujíc SVG (Scalable Vector Graphics) a XUL (XML User Interface Language). CSS specifikace jsou nadále vyvíjeny W3C konsorciem.

CSS je používáno k tomu, aby pomohlo web designérům definovat a čtenářům webových stránek předložit barvy, fonty textů, rozvržení stránek a další aspekty dokumentu.

Primárně je určeno k tomu, aby umožnilo separaci obsahu dokumentu, napsaného v HTML nebo jemu podobném značkovacím jazyku, od jeho vizuální prezentace, která je napsána za pomoci CSS.

Takováto separace může zvýšit přístupnost dokumentu, poskytnout větší flexibilitu a kontrolu v charakteru prezentace, redukovat složitost a redundantnost některých struktur uvnitř obsahu dokumentu.

CSS taktéž dovoluje prezentovat tutéž stránku za pomoci rozdílných stylů pro rozmanité způsoby interpretace informací na výstupních zařízeních jako jsou například: obrazovka počítače, tiskárna, hlasové zařízení (pro předčítání textu), doteková zařízení (předávající informace uživateli hmatem na principu Braillova písma).

CSS má dáno schéma priorit k určení toho, jaký styl aplikovat v případě, že pro jeden určitý element existuje více stylových pravidel. V této tzv. kaskádě jsou priority či váhy kalkulovány a přiřazeny k pravidlu, takže výsledky jsou předvídatelné. (7)

2.3.1. Historie

Stylový předpis existoval v nějaké své formě již od vytvoření jazyka SGML (Standard Generalized Markup Language) v sedmdesátých letech minulého století.

Jak se HTML vyvíjel, začal zahrnovat čím dál tím rozmanitější schopnosti, aby vyhověl požadavkům web designérů. Tato evoluce jim dala větší kontrolu nad vzhledem stránky, ale za cenu stále komplexnějšího HTML, jež byl čím dál tím složitější, jak pro zápis, tak pro údržbu.

Různorodé implementace v rozličných webových prohlížečích zapříčinily, že konsistentní stránky vypadaly v každém poněkud rozdílně a uživatel měl méně kontroly nad zobrazením webového obsahu.

Z důvodu nutnosti zlepšení této situace a schopností webových prezentací bylo předloženo devět návrhů na nový jazyk stylových předpisů. Dva z těchto návrhů byly vybrány pro základ budoucího CSS:

- ⇒ Cascading HTML Style Sheets (CHSS);
- ⇒ Stream-based Style Sheet Proposal (SSP).

První z nich navrhl Hakon Wium Lie (v současnosti pracuje jako CTO – Chief Technical Officer v Opera Software). Jednalo se o jazyk, který již měl jistou podobnost se současným CSS.

Bert Bos tou dobou pracoval na prohlížeči jménem Argo, jenž používal svůj vlastní jazyk stylových předpisů – Stream-based Style Sheet Proposal (SSP).

Tito dva následně spolupracovali ve společném úsilí vyvinout CSS standard (písmeno H bylo záhy ze zkratky vyňato, protože tento nový stylový předpis měl být použitelný i pro ostatní značkovací jazyky, nejen HTML).

Na rozdíl od tehdejších jazyků jako DSSSL (Dokument Style Semantics and Specification Language) nebo FOSI (Formatting Output Specification Instance), CSS umožňoval vzezření dokumentu ovlivnit mnoha stylovými předpisy.

Jeden stylový předpis mohl dědit vlastnosti od druhého, přičemž výsledný vzhled dokumentu byl rovnocenně kontrolovatelný jak designérem, tak uživatelem.

Hakonův návrh byl prezentován na konferenci „Mosaic and the Web“ v Chicagu, v roce 1994 a opět, společně s Bertem Bosem, v roce 1995.

V té době se právě formovalo konsorcium W3C, které projevilo zájem participovat na dalším vývoji CSS. Následně vytvořilo pracovní skupinu k tomu určenou. Hakon a Bert se stali hlavními techniky tohoto nově vzniklého týmu a spolu s ostatními členy, jako byl např. Thomas Reardon z Microsoftu, oficiálně v prosinci 1996 publikovali doporučení CSS 1 (CSS level 1 recommendation).

Vývoj HTML, CSS a DOM (Dokument Object Model) probíhal pod vedením jediné pracovní skupiny – The HTML Editorial Review Board (ERB).

Začátkem roku 1997 byla tato skupina rozdělena do tří nových pracovních skupin:

⇒ HTML Working group;

⇒ DOM Working group;

⇒ CSS Working group.

CSS skupina se začala zabývat rozšířením CSS 1, které 4. listopadu 1997 vyústilo ve vytvoření standardu CSS 2. To bylo vydáno 12. května 1998.

CSS 3, jehož vývoj byl započat roku 1998, je stále ve fázi vývoje.

Roku 2005 se pracovní skupina CSS rozhodla zpřísnit požadavky standardu, což znamená, že již publikované standardy jako CSS 2.1, CSS 3 Selektory a CSS 3 Text byly vráceny zpět z kandidátních doporučení (Candidate Recommendation) na pouhé pracovní návrhy (do předchozího stavu bylo CSS 2.1 vráceno 19. července 2007). (7)

2.3.2. CSS syntaxe

CSS má jednoduchou syntaxi, přičemž používá mnoho anglických slov ke specifikaci rozličných vlastností stylů.

Stylový dokument (soubor s příponou .css) je složen ze seznamu pravidel. Každé pravidlo, nebo sada pravidel, sestává z jednoho či více selektorů a bloku deklarácí.

Blok deklarácí je složen ze seznamu středníkem oddělených dílčích deklarácí uzavřených v kulatých závorkách. Každá deklarace sama o sobě je složena z vlastnosti, dvojtečky (:), hodnoty a středníku (;)

V CSS se selektory používají k deklaraci toho, na které elementy má být daný styl aplikován, tedy jistý výraz spojení. Selektory mohou platit pro všechny elementy

určitého typu nebo pouze na ty, které se shodují v určitém atributu. Spojení s elementy může být taktéž uskutečněno v závislosti na jejich vzájemném relativním umístění ve značkovacím kódu stránky nebo na jejich pozici v modelu dokumentu (DOM – dokument object model).

CSS selektor je tvořen ze vzoru, který je porovnáván se všemi elementy ve stromové struktuře dokumentu. Pokud platí všechny podmínky ve vzoru, veškeré deklarace uvnitř pravidla jsou aplikovány na shodný element nebo elementy. Příklad:

```
p { color: #fff; }
```

Selektor je ta část CSS pravidla, která je definována před složenou závorkou „{“. Selektor je v tomto případě `p`, jež je spojeno se všemi `p` elementy v dokumentu. Těm přiřazuje bílou barvu (`#fff`).

<i>Typ selektoru</i>	<i>Vzor</i>	<i>Popis</i>
Univerzální (Universal)	*	Odpovídá jakémukoliv elementu.
Type (Typ)	Q	Odpovídá každému Q elementu.
Vlase (Třída)	.info	Odpovídá každému elementu jehož třída obsahuje hodnotu info.
ID	#pata	Odpovídá každému elementu s ID „pata“
Descendent (Následovník)	Q W	Odpovídá jakémukoliv elementu W jenž je následovníkem elementu Q.
Child (Potomek)	Q > W	Odpovídá jakémukoliv elementu W, který je potomkem elementu Q.
Adjacent (Přilehlý selektor)	Q + W	Odpovídá každý element W bezprostředně předcházený sourozeneckým elementem Q
Attribute (atribut)	Q[att]	Odpovídá jakýkoliv element Q, jež obsahuje att atribut, nehledě na jeho hodnotu.

Tabulka 1: Přehled selektorů – první část (8)

<i>Typ selektoru</i>	<i>Vzor</i>	<i>Popis</i>
Atribute (atribut)	Q[att=val]	Odpovídá libovolný element Q, jehož atribut att je roven hodnotě val.
Atribute (atribut)	Q[att~=val]	Odpovídá každý element Q, jehož hodnota val atributu att je rovna jedné ze seznamu hodnot oddělených mezerou.
Atribute (atribut)	Q[att =val]	Odpovídá jakýkoliv element Q, jehož atribut att obsahuje seznam hodnot začínajících hodnotou val.
:first-child pseudo třída	Q:first-child	Element Q odpovídá pokud je Q první z „dětí“ svého rodičovského elementu.
Pseudo třídy pro odkaz (link)	a:link a:visited	Odpovídá zatím nenavštívenému (:link) nebo naopak už navštívenému (:visited) odkazu.
Dynamické pseudo třídy	Q:active Q:hover Q:focus	Odpovídá elementu Q během určitých akcí uživatele.
Jazyková pseudo třída (the :language pseudo-class)	Q:lang(c)	Odpovídají elementy typu Q, jež jsou v jazyku c.
Pseudo element prvního řádku (The :first-line pseudo-element)	Q:first-line	Odpovídá obsahu prvního formátovaného řádku elementu Q
Pseudo element prvního znaku (The :first-letter pseudo-element)	Q:first-letter	Odpovídá prvnímu znaku elementu Q

Tabulka 1: Přehled selektorů – druhá část (8)

⇒ *Descendent (potomek, následovník)*

Element, který je potomek (child), prapotomek (grandchild) atd. ve vztahu k jinému elementu ve stromové struktuře dokumentu.

⇒ *Ancestor (předek)*

Element, který zaujímá rodičovské (parent), prarodičovské (grandparent) atd. postavení ve vztahu k druhému elementu.

⇒ *Child (přímý následovník)*

Přímý potomek elementu. Žádné jiné elementy nemohou být vloženy mezi tyto dva.

⇒ *Parent (rodičovský element)*

Přímý předchůdce elementu. Mezi tyto dva není možno vložit kterýkoliv jiný element.

⇒ *Sibling (sourozenec)*

Element, který má stejné rodiče jako aktuální element. (8)

2.3.3. Jednoduché a kombinované selektory

Existují dvě základní kategorie selektorů: *jednoduché* a *kombinované*.

⇒ *Jednoduchý selektor* je složen buď ze selektoru přiřazeného konkrétnímu typu elementu (type selector), nebo z univerzálního selektoru, jež je následován žádným nebo naopak více atributy ID selektorů či pseudo tříd.

```
p.info { background: #000; }
```

⇒ *Kombinovaný selektor* (někdy též nazývaný kontextový) je tvořen ze dvou nebo více oddělených jednoduchých selektorů.

```
div p { line-height: 1.8; }
```

Toto pravidlo bude aplikováno na všechny p elementy, které jsou potomky elementu div. (8)

2.3.4. Univerzální selektory

Univerzální selektor je ve stylovém předpisu reprezentován znakem „*“ (asterisk) a odpovídá všem elementům v dokumentu. K vidění je však velmi zřídka. Na druhou stranu je velice často používán se selektory typu třída a ID.

Pokud není univerzální selektor jedinou komponentou jednoduchého selektoru, může být znak „*“ vynechán:

⇒ .myclass je ekvivalentní zápisu *.myclass;

⇒ #myid je ekvivalentní zápisu *#myid.

Jedno z použití univerzálního selektoru, které se stalo poměrně populární, je jeho užití k nastavení vnějších a vnitřních okrajů (margins, paddings) u všech elementů na nulu.

```
* { margin: 0; padding: 0; }
```

Tento způsob je občas označován jako „Global White Space Reset“².

2.3.5. Selektor přiřazený konkrétnímu typu elementu (type selector)

Tento druh selektoru odpovídá každé instanci určitého typu elementu. Například následující pravidlo se vztahuje na všechny elementy typu odstavec, které jsou obsaženy v dokumentu a nastavuje pro ně velikost použitého fontu na 3em (8):

```
p { font-size: 3em; }
```

² *Global white space reset* [online]. 2008 [cit. 2008-04-23].

Dostupný z WWW: <<http://leftjustified.net/journal/2004/10/19/global-ws-reset/>>

2.3.6. Třídní selektor (class selector)

Je reprezentován tečkou „.“ a hodnotou atributu třídy, na jejímž základě je spjat s cílovými elementy. Například následující pravidlo bude aplikováno na všechny elementy `p` se jménem třídy „anotace“

```
p.anotace { background-color: #a2a9a6; }
```

Existuje i možnost jak přiřadit elementu více tříd – atribut `class` může obsahovat mezerou oddělovaný jmenný seznam tříd. Třídní selektory potom mohou být užity k tomu, aby byly aplikovány pouze na elementy, které obsahují několik jmen tříd (class names) zároveň.

Níže uvedené pravidlo bude použito pro elementy typu `p`, jež mají zároveň uvedeno „anotace“ a „duraz“ v jejich jmenném seznamu tříd.

```
p.anotace.duraz { background-color: #a2a9a6; font-weight: bold; }
```

Tyto vícečetné třídní selektory nefungují v IE6, avšak v IE7 jsou již podporovány. Není třeba definovat typ elementu. Vynechání je stejné jako při užití univerzálního selektoru místo typu selektoru.

Ukázkové pravidlo bude aplikováno na všechny elementy s hodnotou jména třídy „anotace“, nehledě na jejich typ. (8)

```
.anotace { background-color: #a2a9a6; }
```

2.3.7. ID selektor

ID selektor je reprezentován „hash“ znakem „#“. S cílovým elementem je spjat jejich vzájemnou hodnotou jména tohoto selektoru. Je důležité zmínit, že ID selektory mají

větší specifičnost než class selektory a že hodnota ID musí být jedinečná v celém dokumentu, a proto tedy může být ID selektor aplikován pouze na jediný element v dokumentu.

Pravidlo bude užito na element, jehož id je „obsah“, nehledě na typ elementu. (8)

```
p.obsah { background-color: #fff; }
```

2.3.8. Zdroje stylového předpisu

CSS informace může být poskytnuta mnoha zdroji. Stylový předpis je buď připojen v samostatném dokumentu, nebo může být vnořen v HTML dokumentu. Je možné importovat mnoho různých stylů. Lze definovat alternativní styly, mezi kterými si může uživatel zvolit.

Různé styly mohou být použity podle používaného výstupního zařízení. Například verze určená pro zobrazení na monitoru počítače může být úplně jiná než verze pro tiskárnu, což znamená, že autor může přizpůsobit prezentaci vhodně pro každé médium.

⇒ *Styly definované autorem* mohou být použity ve formě:

- *Externích stylových předpisů*, např. referencí v těle dokumentu na soubor CSS;
- *Vnořených stylových předpisů*, tzn. blok CSS informací přímo v těle HTML dokumentu;
- *Inline styly*, přímo v těle HTML dokumentu, stylová informace pro jeden samostatný element, definovaná užitím atributu „style“ v těle elementu.

⇒ *Uživatelské styly*

- Určený lokálním CSS souborem, který je definován volbou uživatele ve webovém prohlížeči. Má vyšší prioritu než autorský stylopis, jež nahradí.

⇒ *User agent style*

- Standardní stylový předpis aplikovaný uživatelským agentem (kterým může být např. standardní prezentace webového prohlížeče).

Jedním z hlavních cílů CSS je také poskytnout uživateli větší volnost nad vzezřením prezentace. Například ti, kterým se zdá žluté písmo, použité pro prezentaci autorem stránky, špatně čitelné, mají možnost použít jiný stylový předpis pro danou stránku.

V závislosti na použitém prohlížeči a webové stránce si uživatel může vybírat z různých variant stylových předpisů pro něj předpřipravených web designérem.

Dokonce je mu umožněno, aby vyňal všechny předdefinované styly a zobrazil stránku jen s použitím standardního zobrazení prohlížeče nebo může potlačit pouze pro něj nepříjemnou žlutou barvu textu, které bylo použito pro text stránky, aniž by změnil vzezření nějakého jiného objektu v těle stránky. (8)

2.4. Implementace

Ačkoliv specifikace CSS1 byla kompletní už v roce 1996 a Microsoft Internet Explorer 3, který zahrnoval jistou limitovanou podporu CSS, byl vydán téhož roku, trvalo nakonec více než další tři roky, než některý z webových prohlížečů dosáhl téměř kompletní implementace specifikací.

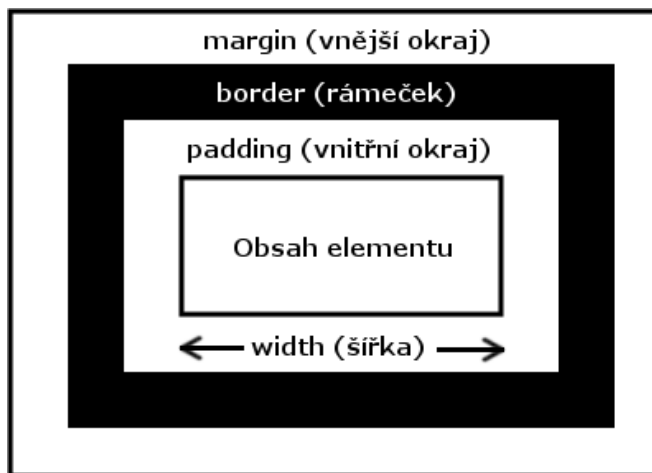
Internet Explorer 5.0 pro platformu Macintosh, vydaný v březnu 2000, byl první prohlížeč, který měl plnou (větší než 99%) podporu CSS1, porážejíc prohlížeč Opera, jenž měl vůdčí postavení v rozsahu implementace CSS1 od chvíle, kdy její tým ohlásil podporu tohoto standardu, tedy o patnáct měsíců dříve. Ostatní prohlížeče dohnaly náskok brzy poté.

Mnoho z nich dodatečně implementovalo části CSS2, avšak až do roku 2006 žádný prohlížeč plně tuto verzi nezahrnoval.

Rané prohlížeče jako Internet Explorer 3 a 4, nebo Netscape 4.x sice měly podporu CSS, avšak ta byla značně neúplná, sužována mnoha chybami, což byla výrazná překážka na cestě k implementaci.

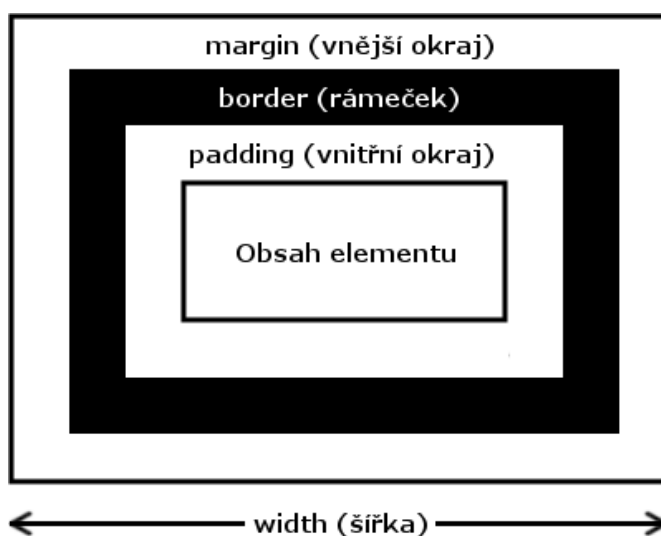
Když pozdější prohlížeče začaly nabízet relativně slušnou podporu kaskádových stylů, byly přesto stále nekorektní v interpretaci některých elementů a obtěžkány nekonzistentnostmi, chybami a drobnějšími defekty. (7)

Následující obrázek ilustruje správné chápání okrajů ve většině prohlížečů – dle normy:



Obrázek 2: Správná interpretace okrajů (12)

A jejich špatnou interpretaci v některých verzích programu Internet Explorer:



Obrázek 3: Špatná interpretace okrajů podle IE (12)

S každou následující generací prohlížečů se objevovalo mnoho dalších a dalších chyb a nesrovnalostí, které výrazně ztížili designérům dosažení konzistentního vzhledu jejich prezentací různými druhy prohlížečů.

Někteří autoři se proto běžně uchýlili k využití ne příliš korektních klíčků (jako jsou CSS hacky a filtry) sloužících k dosažení konzistentního vzezření prezentace napříč prohlížeči.

Problémy se špatným přijetím CSS browsery, spolu s množstvím úprav vedl konsorcium W3C k revizi standardu CSS2.1, na niž lze nahlížet jako na praktickou, pracovní verzi současné podpory CSS v prohlížečích.

Některé vlastnosti CSS2, které žádný prohlížeč nebyl schopen dostatečně dobře integrovat, byly vypuštěny a v několika případech již definovaná pravidla byla následně přetvořena tak, aby souhlasila s jejich existující majoritní implementací v prohlížečích. (7)

2.5. Použitelnost webové prezentace

Jde o množinu vlastností, které by měla splňovat každá správná webová prezentace. Touto použitelností je míněna jednoduchost a srozumitelnost ovládání.

To by mělo být takové, aby jí každý návštěvník stránek porozuměl a neztratil při pročítání webu orientaci.

Podle *Jakoba Nielsena*³ (jeden z předních consultantů na téma webové přístupnosti, jenž je držitelem titulu Ph.D. v oboru „interakce mezi počítačem a člověkem“ z Dánské technické univerzity v Copenhagenu – Technical University of Denmark,) jsou klíčovými vlastnostmi pro přístupnost webu:

- ⇒ *Pochopitelnost* – udává, jak lehce je návštěvník schopen vykonat základní úkony při prvním navštívení daného webu;
- ⇒ *Efektivita* – definuje rychlost, jakou je návštěvník schopen provést požadované úkony poté, co se na webu již zorientoval;
- ⇒ *Zapamatovatelnost* – vyznačuje lehkost s jakou se návštěvník opětovně zorientuje při další návštěvě;
- ⇒ *Chybování* – znamená, jak moc návštěvník při svém prozkoumávání webu chybuje. Jak moc jsou chyby vážné a jak rychle je schopen zotavení;
- ⇒ *Subjektivní uspokojení* – definuje míru pohodlí při používání webu a radost s jakou s ním pracuje.

³ *Measuring the Usability of Reading on the Web* [online]. 2005 [cit. 2008-02-26]. Dostupný na WWW: < <http://www.useit.com/alertbox/readingmetrics.html> >

2.6. SEO (Search Engine Optimization – optimalizace pro vyhledávače)

SEO je metoda, díky níž lze navést cílené návštěvníky na naši webovou prezentaci nebo internetový obchod.

Její pomocí lze docílit vysoké návštěvnosti tak, že naše stránky budou zaujímat první pozice ve výsledcích vyhledávačů⁴.

2.6.1. Faktory optimalizace

⇒ *On-Page faktory* – tyto faktory mohou být ovlivněny přímo na stránkách:

- klíčová slova
- délku zdrojového kódu
- vzájemné provázání odkazy
- sémantika dokumentu

⇒ *Off-Page faktory* – nemohou být ovlivněny v rámci našeho webu:

- zpětné odkazy
- efekt domény

2.6.2. Klíčová slova

Klíčová slova jsou taková, která návštěvníci Internetu používají nejvíce při vyhledávání, tedy slovo či slovní spojení, které charakterizuje hledaný subjekt.

⁴ *Seomaster: optimalizace webových stránek* [online]. 2008 [cit. 2008-02-26]. Dostupný z WWW: <<http://www.seomaster.cz/co-je-SEO-optimalizace>>

2.6.3. Pagerank

Těž se někdy označuje termínem „link popularity“. Je veličinou přímo úměrnou počtu stránek odkazujících se na danou stránku, přičemž je brán v potaz i jejich pagerank.

Zároveň je nepřímo úměrná počtu odkazů na těchto stránkách. Pagerank byl původně vynálezem společnosti Google. Následně však s touto veličinou začali operovat i algoritmy ostatních vyhledávačů (6).

2.7. Typografie na webu

Obdobně, jako u tištěných dokumentů, nemělo by se zapomínat přinejmenším na základní, typografická pravidla ani při tvorbě webových prezentací.

2.7.1. Výčet základních pravidel, která jsou často opomíjena

⇒ *Nedělitelná mezera*

- Žádný řádek nesmí končit jednopísmennou předložkou, jedinou výjimkou je malé písmeno a.
- Číslo a zkratka jednotky (např. 100 kg), titul a jméno (Ing. Starý) nebo iniciála s příjmením (J. Šťastný) nesmí být odděleny.
- V vytvoření nedělitelné mezery využívá HTML entitu ** **;

⇒ *Zúžená mezera*

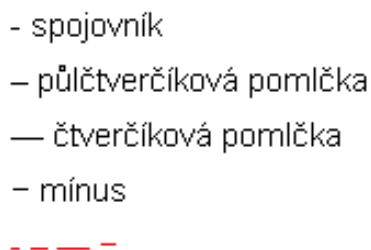
- Je také nedělitelná, o kousek kratší než standardní mezera. Používá se pro rozdělování čísla po třech cifrách (např. při oddělování tisíců).
- Dále nachází využití jako oddělovač mezi číslicí a znakem procenta, ale pouze v případech, kdy je procentuelně vyznačena jednotka (např. „5 procent“). Pokud se jedná o část slova, mezera se před znakem procenta nepoužívá (např. „pěti procentní“).
- Zúžená mezera se zapisuje entitou ** **.

⇒ *Spojovník a pomlčka*

- Rozdíl mezi spojovníkem a pomlčkou je ten, že pomlčka je delší.
- Spojovník slouží ke spojování slov (např. česko-anglický slovník).
- Pomlčka se používá k oddělování větných celků. Před i za pomlčkou píšeme běžnou mezeru.
- Ve významu „až“ (a, versus, až do) je pomlčka zapisována bez mezer po obou stranách
- Existují dva druhy pomlček – půlčtverčiková a čtverčiková. První z nich se vyjadřuje entitou **–**; čtverčiková potom entitou **—**.

⇒ *Matematické znaménko mínus*

- Mínus je o něco delší než spojovník a přitom kratší než půlčtverčiková pomlčka. Hlavní rozdíl je ve vertikálním umístění – mínus je totiž o něco výš než pomlčka či spojovník. Píše na úrovni horizontální čáry znaménka plus.
- Pro zápis slouží entita **−**



Obrázek 3: Různá interpretace znamének (15)

⇒ *Výpustka*

- Všeobecně se místo znaku výpustky používají tři po sobě jdoucí tečky. Avšak správně by na jejich místě měl stát znak jediný, a to právě výpustka. Rozdíl mezi třemi za sebou jdoucími tečkami a výpustkou je jednak v délce mezer mezi tečkami, přičemž u výpustky je mezera mezi tečkami o něco větší a kratší mezera je též před první tečkou.
- Výpustka se zapisuje pomocí html entity **…**

- ■ ■ tři tečky
- ■ ■ výpustka

Obrázek 5: Rozdíl mezi výpustkou a třemi tečkami (15)

⇒ *Znak „krát“*

- I pro vyjádření „krát“ existuje v html jazyce speciální znak. Mnoho lidí jej však zaměňuje s malým „x“. To vypadá obzvláště nevhodně při použití patkového písma.
- „Krát“ je stejné na šířku jak na délku, není položeno na spodní lince řádku, ale je vertikálně zarovnáno na jeho střed.
- Pro jeho zápis slouží entita `×` (15)

⇒ *Sloveso Viz*

- Jedná se o sloveso ve významu „podívejte se na“. Proto se za ním nepíše tečka. Není to totiž zkratka.

⇒ *Úmrtní křížek*

- Pro narozen a zemřel se často používají zástupné znaky hvězdička a křížek. Zatímco se zápisem hvězdičky není problém, často se stává, že úmrtní křížek je nahrazen znakem plus.
- Úmrtní křížek se zapisuje entitou **`†`** (14)

3. Analýza problému a současné situace

Webová prezentace bude vytvořena pro stavební společnost APMB s.r.o. Adresa firmy: Masarykova 808, 664 61 Rajhrad. Firma se zabývá především:

- ⇒ stavbami rodinných domů na klíč
- ⇒ rekonstrukcemi a přípravou pro stavby
- ⇒ zemními pracemi
- ⇒ vnějším i vnitřním zateplováním
- ⇒ výstavbou průmyslových staveb

I přesto, že stavební firma APMB s.r.o. existuje na trhu již řadu let, jako taková neměla nikdy žádnou webovou prezentaci a dokonce ani firemní logo.

Jednatel firmou panem Milošem Pelouškem mi bylo zadáno vytvořit několik návrhů firemního loga, z nichž si vybere takové, které se mu nejvíce zamlouvá.

To bude následně použito jak na všech oficiálních tiskovinách firmy, tak i na webových stránkách.

3.1. Volba nástrojů

Vzhledem k tomu, že vývojáři majoritně používaný software pro tvorbu a úpravu rastrové grafiky Photoshop, či jiný komerční program, je pro mne finančně nedostupný, využiji free software alternativu – grafický editor *Gimp* (GNU Image Manipulation Program), který, i navzdory méně sofistikovanému uživatelskému rozhraní – oproti komerčnímu Photoshopu, nabízí veškeré funkce, které by grafik mohl potřebovat. A především je zcela zdarma.

3.1.1. Software využitý v projektu

Mezi další využitý software patří textový editor VIM (free software) jež je rovněž nabízen zdarma a XHTML/CSS validátor vyvíjený konsorciem W3C (k ověření správné syntaxe). K nalezení na oficiálních stránkách W3C.

4. Vlastní návrhy řešení

4.1. Logo firmy

Z několika mnou vytvořených návrhů si jednatel firmy pan Peloušek vybral právě jedno. Pro veškeré zde uvedené návrhy bylo použito zmiňovaného rasterového editoru Gimp.

4.1.1. Návrhy loga



Obrázek 6: Návrh loga #1



Obrázek 7: Návrh loga #2



Obrázek 8: Návrh loga #3



Obrázek 9 : Návrh loga #4

Vítězným se stal poslední ze zde prezentovaných návrhů. Ten byl také použit pro hlavičku webové prezentace. Grafika použitá pro vytvoření prvních dvou návrhů loga bude užitečně využita v jiné části prezentace.

4.2. Struktura webu

Prezentace firmy je poměrně stručná, proto bude rozdělena do pouze několika sekcí:

- ⇒ *Úvod* – uvítání návštěvníka na webových stránkách
- ⇒ *O společnosti* – historie společnosti, představení řídicích pracovníků
- ⇒ *Služby* – seznam a rozpis nabízených služeb
- ⇒ *Reference* – stručná ukázka již hotových projektů společnosti
- ⇒ *Kontakt* – kontaktní informace – adresa, mail, telefon

4.3. Technické provedení – rozvržení stránky – layout

Při vytváření prezentace je rozvržení stránky většinou nejsložitějším, avšak zároveň klíčovým elementem k dostupnosti prezentace i příjemnému designu.

Dříve, obzvláště před vznikem CSS, dominoval na poli layoutů tzv. tabulkový design, kdy jednotlivé dílčí části prezentace byly de facto buňkami všezahrnující tabulky. Tento zastaralý, do sémantiky dokumentu nezapadající, avšak velice snadno pochopitelný a prohlížeči dobře interpretovaný, koncept byl během několika let nahrazen sofistikovanějším rozvržením za pomoci kaskádových stylů.

I přes to se tabulkového designu mnoho webdesignérů dlouhou dobu nechtělo vzdát. Na internetu takových prezentací stále existuje velké množství, především jako pozůstatek z dřívějších dob.

Co se týče rozvržení stránky za pomoci CSS, jeho správnou výstavbou se zabývá na mnoha webech a diskusních fórech napříč Internetem velký počet designérů. Ti vytvořili spoustu návrhů k dosažení různorodých typů layoutů, které jsou jak korektní dle platných norem W3C, tak konzistentní při použití různorodých prohlížečů.

Na druhou stranu jsou většinou podstatně obtížnější na pochopení i realizaci než jednoduchý tabulkový design.

Při tvorbě rozvržení jsem použil návrh italského webdesignéra Alessandra Fulcinitiho(13).

Ten vytvořil přímočarý návrh, který využívá techniky „negativních“ či jinak řečeno „záporných“ vnějších okrajů (negative margins). Tento princip spočívá v tom, že při zadání záporné hodnoty velikosti vnějšího okraje se prvek posune na opačnou stranu.

Toto využití vnějšího okraje je poněkud neobvyklé. Avšak implementace je velice jednoduchá. Fungující napříč prakticky všemi webovými prohlížeči (testováno na různých verzích prohlížečů Internet Explorer, Opera, Firefox, Galon, Epiphany, Konqueror a Safari). A na rozdíl od mnoha jiných konceptů nevyužívá žádné hacky či filtry.

Tento návrh jsem vhodně modifikoval a aplikoval na něj mnou vytvořený design.

Stránky mé prezentace jsou statické (neexistuje žádný důvod k vytvoření dynamického webu – prezentace je, na základě majitelem poskytnutých materiálů,

relativně stručná a její obsah se nebude v čase měnit), s pevně stanovenou šířkou základních částí (dynamická šířka obsahu dle velikosti okna prohlížeče není podle mého názoru příliš vhodná, neboť velký počet znaků na jednom řádku výrazně zneprůjemňuje četbu).

⇒ *Návrh sestává z jednoduché HTML konstrukce:*

```
<div id="container">
  <div id="header">Hlavička</div>
  <div id="wrapper">
    <div id="content">Obsah</div>
  </div>
  <div id="navigation">Navigace</div>
  <div id="extra">Pravý boční sloupek</div>
  <div id="footer">Pata</div>
</div>
```

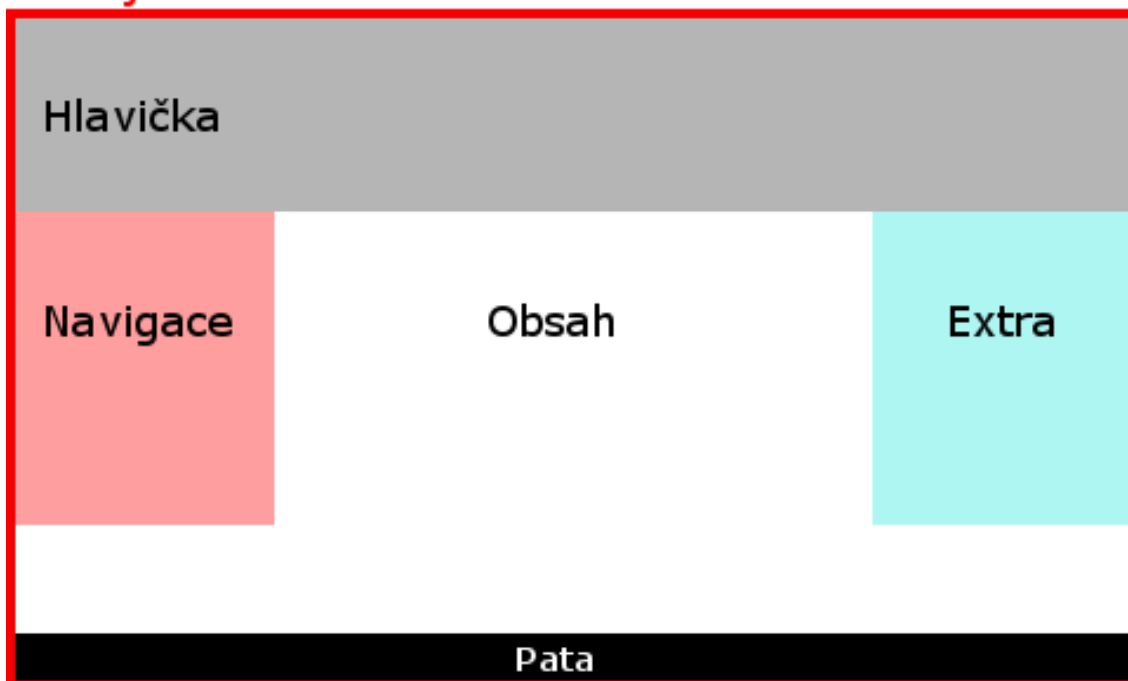
⇒ *Na ni je aplikován následující stylový předpis:*

```
div#container{width:700px;margin:0 auto}
div#wrapper{float:left;width:100%}
div#content{margin: 0 150px}
div#navigation{float:left;width:150px;margin-left:-700px}
div#extra{float:left;width:150px;margin-left:-150px}
div#footer{clear:left;width:100%}
```

<i>Název elementu</i>	<i>Popis</i>	<i>šířka</i>
Container	Kontejner – celková šířka stránky	700px
Navigation	Navigace – šířka levého sloupečku s menu	150px
Extra	Extra – sloupeček v pravé části prezentace	150px
Content	Obsah – (na ten připadá zbytek šíře, tedy Content = Container - Navigation – Extra)	400px
Footer	Pata (přes celou šíři)	700px

Tabulka 2: Definice elementů a jejich šířek na stránce dle CSS

Kontejner



Obrázek 10: Schéma rozvržení

Pro mnou navrženou webovou prezentaci není třeba pravého sloupce, proto byl z rozvržení stránky vypuštěn a obsah rozšířen.

```

div#container {width: 749px; margin: 0 auto;}
div#wrapper {float: left; width: 100%; background: #fff
url(../img/background.png) top left repeat-y;}
div#content {margin: 0 0px 100px 175px;}
div#global-navigation {float: left; width: 175px; margin-
left: -749px;}
div#footer {clear: left; width: 100%}

```

4.4. Přístupnost stránek

K dosažení co možná největší přístupnosti prezentace je třeba zohlednit i uživatele s pomalejším přístupem k Internetu (modem, připojení za pomoci mobilního telefonu...), to znamená aplikovat určitá pravidla:

⇒ *Správná sekvence zobrazení* (tzn. pořadí elementů v těle XHTML dokumentu). Nejdříve se, ač nevýhodně – protože obsahuje málo relevantních informací, načte hlavička dokumentu, která je tvořena obrázkem (zabírá nejvíce kB ze všech elementů na úvodní stránce, proto je také zobrazení relativně pomalé). Její umístění na začátku stránky technicky zabraňuje zmínit ji v těle dokumentu až později. Následuje obsah a až poté navigační menu a pata stránky. Prakticky to znamená, že návštěvník s pomalým připojením k Internetu, může okamžitě po načtení hlavičky a obsahu získávat informace – číst obsah, zatímco ostatní prvky stránky jsou stále načítány (např. reklamní bannery, grafické prvky).

⇒ *Minimalizace grafických elementů*. Z důvodu rychlého načtení stránky (čili transferu ze strany serveru ke klientovi) byly na veškerou grafiku použity ztrátové kompresní algoritmy (obrázky se na stránce vyskytují ve formátech .jpeg a .png) – což však v tomto případě není na škodu, neboť s dalšími úpravami grafiky se nadále nepočítá.

⇒ *Správné zobrazení stránky i bez použití kaskádových stylů.* Musíme počítat i s možností, že návštěvník webové prezentace si nebude stránky prohlížet na obrazovce počítače, ale např.: prostřednictvím mobilního telefonu, či zařízení PDA. Z tohoto důvodu by informativní obsah stránek měl být korektně zobrazen i při úplném ignorování stylového předpisu zobrazovacím zařízením.

4.4.1. Dekompozice grafického návrhu

Původní grafický návrh stránek byl dekomponován na dílčí prvky, které byly za pomoci kaskádových stylů opětovně zkompletovány, přičemž rastrové grafiky bylo použito co nejméně.



Obrázek 11: Hlavička webové prezentace

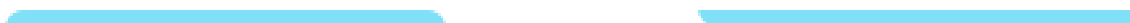


Obrázek 12: Pata prezentace

Pro patičku prezentace byl z estetických důvodů použit efekt odrazu na vodní hladině.



Obrázek 13: Horní a spodní část tlačítka menu



Obrázek 14: Horní a spodní část tlačítka menu, na němž stojí kurzor myši

4.4.2. Implementace grafických prvků

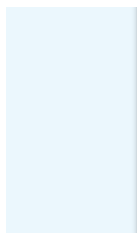
⇒ *Realizace tlačítka v XHTML*

```
<li><a href="#"><span>0 společnosti</span></a></li>
```

⇒ *Realizace tlačítka v CSS*

```
#menu a {background: #cdf8ff url(..img/menu-button-  
bottom.png) bottom left no-repeat;}
```

```
#menu a span {display:block; background:url(..img/menu-  
button-top.png) top left no-repeat; padding: 0.6em 0;}
```



Obrázek 15: Ukázka levého vertikálního okraje

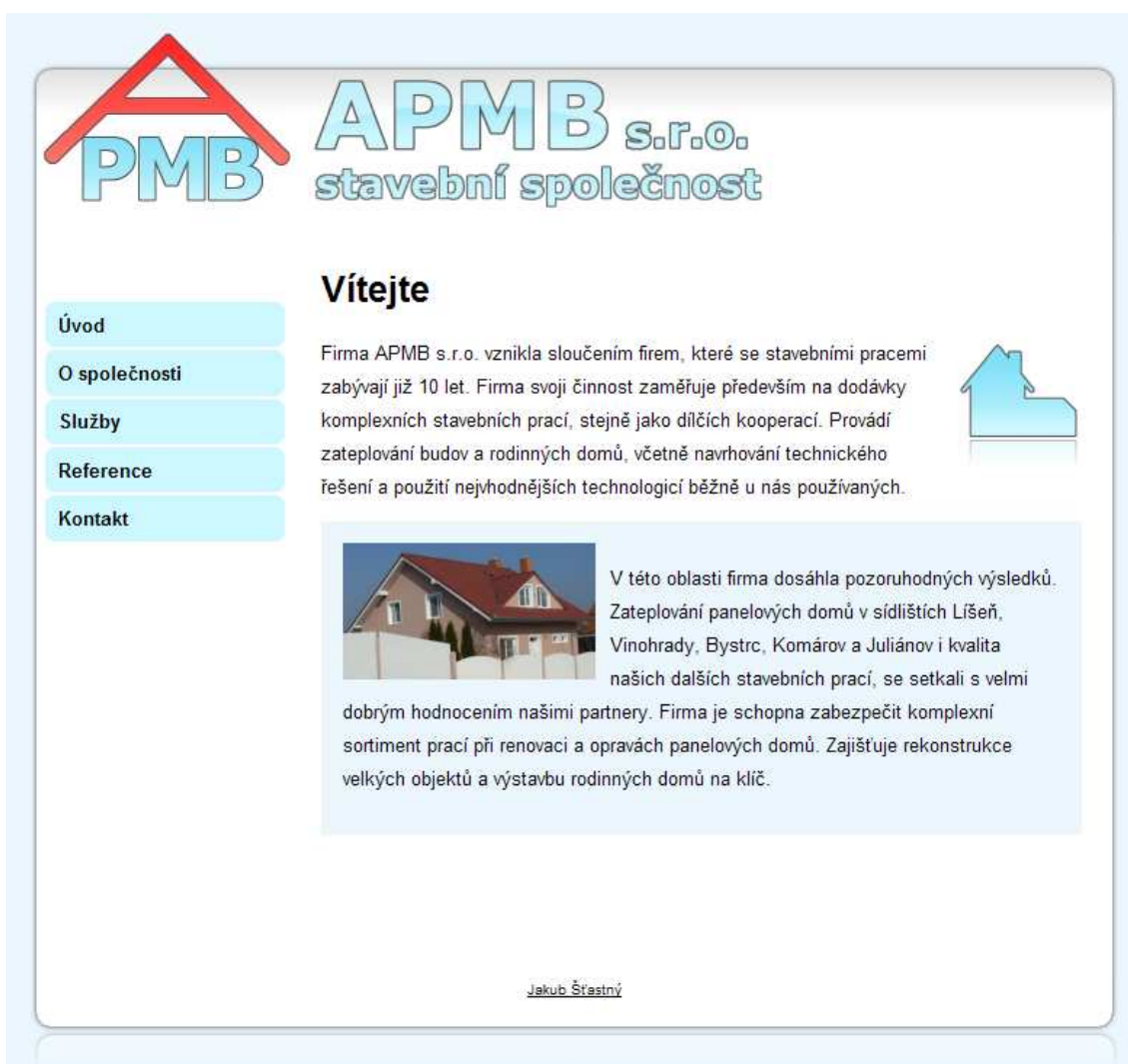
Stínovaného okraje po obvodu prezentační části stránky bylo dosaženo za pomoci filtru „Vržený stín“ v editoru Gimp, přičemž parametry definující posun stínu v ose X a Y byly nastaveny na hodnotu 0.

Interpretace okraje v těle dokumentu byla vytvořena následovně:

Nejprve byl z celkového návrhu stránky vyňat úzký proužek obsahující po stranách pravý a levý okraj o rozměrech 749 (šířka prezentační části stránky) × 1 px, který byl uložen ve formátu png.

Ten byl následně pomocí CSS direktivy rozšířen ve vertikálním směru (stránka vertikálně automaticky přizpůsobuje svoji velikost dle obsahu).

```
div#wrapper {float: left; width: 100%; background: #fff url(..img/background.png) top left repeat-y;}
```



Obrázek 16: Výsledný vzhled stránky

4.4.3. Obsah zdrojového kódu

⇒ *Hlavička*

Hlavička xhtml dokumentu obsahuje kromě definování jazyka, autora stránek a jejich popisu také informace určené pro programy ke shromažďování metadat nebo dokumentů jako takových – roboty. Jedná se o tyto metaelementy⁵:

Metaelement robots

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

Index – znamená pro robota to, že si smí uložit URL stránky společně s dalšími užitečnými informacemi.

Follow – značí, že robot může pokračovat v prohledávání webu pomocí hypertextových odkazů uvedených v těle dokumentu.

Metaelement googlebot – je uveden výhradně pro potřeby vyhledávače Google (řídí se podle něj i některé další roboty).

```
<meta name="googlebot" content="index, follow, snippet, archive" />
```

Archive – značí to, že si Google smí uložit snímek stránky do své databáze a ten potom poskytovat k nahlédnutí. Toto chování často nevyhovuje poskytovatelům placeného obsahu – např. různým webovým zpravodajům.

Snippet – definuje zobrazení výsledků hledání, jehož součástí je hledaná stránka. Při použití tohoto kódového slova se zobrazí i část textu kolem hledaného slova. Pokud by

⁵ Interval.cz : Metadata a hlavička pro XHTML dokument [online]. 2008 [cit. 2008-02-29]. Dostupný z WWW<<http://interval.cz/clanky/metadata-a-hlavicka-pro-xhtml-dokument/>>

toto kódové slovo použito nebylo, zobrazil by se jen popis dokumentu, který je uveden v metaelementu „description“

Keywords (klíčová slova) – Tento metaelement by měl obsahovat klíčová slova, jež mají pro stránku mimořádný význam. V praxi se nedoporučuje používat více než čtyři až pět slov.

```
<meta name="keywords" content="stavební společnost,  
stavby, tepelné izolace, fasády, zateplení domů" />
```

Metaelement description

Definuje výstižný popis dokumentu. Ten by neměl přesáhnout délku 256 znaků

```
<meta name="description" content="APMB s.r.o. - stavební  
společnost" lang="cs" />
```

⇒ *Navigace*

Navigační menu, které zabírá levou část stránky, je tvořeno za pomoci xhtml značek definujících seznam. Na tuto konstrukci je následně aplikován stylový předpis, který dá menu jeho grafickou podobu.

Z důvodu přístupnosti je možno aktivovat prvky menu nejen za pomoci myši, ale též pomocí klávesových zkratk. Konkrétně ALT + SHIFT + „znak“,

```
<li><a href="./index.html" accesskey="1" title="Úvodní  
strana, shortcut key=1"><span>Úvod</span></a></li>
```

Pokud nebude při zobrazení stránek použito stylového předpisu, budou na stránce dodatečně zobrazeny dílčí navigační prvky pro snadnější orientaci – lokální odkazy. Na

úplném začátku „Přeskočit na navigaci“ a na konci obsahu „nahoru“. Oběma je taktéž přiřazena klávesová zkratka.

```
<a class="hidden" href="#global-navigation" accesskey="p"
title="Přeskočit na globalní navigaci">Přeskočit na
navigaci</a>
<a class="hidden" href="#container" accesskey="n"
title="Přeskočit na začátek stránky">nahoru</a>
```

V souboru stylových předpisů je třída `hidden` definována následovně (parametr `display` definuje viditelnost, proto jsou tyto elementy, na které je vázán, zobrazeny pouze v situaci, kdy není stylový předpis aplikován).

```
.hidden {display: none;}
```

5. Závěr

V bakalářské práci se zabývám návrhem a realizací webových stránek pro stavební firmu APMB, s.r.o.

Firma neměla před mým návrhem žádnou prezentaci tohoto typu, proto pro ni je tato práce velice přínosná. Jejím úkolem je především seznámit potencionální zákazníky s jak firmou samotnou, tak nabídkou jejich služeb.

Vzhledem k malé rozlehlosti webu jsem pro jeho tvorbu zvolil pouze základní statický koncept za pomoci jazyků XHTML a CSS.

Součástí práce je také návrh oficiálního loga společnosti, které bude mít pro firmu širší využití i mimo účel webové prezentace. Pro vytvoření návrhu i dílčích grafických částí webové stránky je použito programu Gimp, který je poskytován zdarma a umožnil mi tak společně s editorem Vim (rovněž zdarma) vytvořit produkt zcela bez finančních nákladů na softwarové vybavení.

Ke zkompletování teoretické části, jenž sloužila jako základ realizace praktické části projektu, jsem využil poznatků, které jsem získal studiem převážně odborných textů dostupných na Internetu.

V praktické části práce se zabývám konkrétním grafickým návrhem webu a postupy aplikovanými během výstavby až k jeho finální podobě.

6. Seznam použité literatury

6.1. Knihy

- (1) ECKER, C.: Profesionální webdesign: techniky a vzorová řešení. 1. vyd. Brno: CP Books, 2005. 421 s. ISBN: 80-251-0547-4.
- (2) KOSEK, J.: HTML. Tvorba dokonalých WWW stránek, Grada, 1998, ISBN 80-7169-608-0
- (3) STANÍČEK, P.: CSS Kaskádové styly Kompletní průvodce. 2003. 192 s. ISBN 80-7226-872-4.
- (4) STEJSKAL, Jan. Vytváříme WWW stránky pomocí HTML, CSS a JavaScriptu. 2004. 256 s. ISBN 80-2510-167-3.
- (5) WELLING, L. a THOMSON, L.: PHP a MySQL – rozvoj webových aplikací. Praha: SoftPress, 2002. 720 s. ISBN 80-86497-20-8.

6.2. Internetové adresy

- (6) *Co je to PageRank* [online]. 2008 [cit. 2008-03-13].
Dostupný z WWW: <<http://vyhledavace.info/seo-faq/23/pagerank-jyxorank/>>
- (7) *CSS: Cascading Style Sheets* [online]. 2008 [cit. 2008-02-13].
Dostupný z WWW: <http://en.wikipedia.org/wiki/Cascading_Style_Sheets>
- (8) *CSS 2.1 selectors* [online]. 2005 [cit. 2008-03-10].
Dostupný z WWW: <<http://www.456bereastreet.com/archive/200509/>>
- (9) *Česká a světová typografie na Internetu* [online]. 2006 [cit. 2008-04-10].
Dostupný z WWW: <http://www.typo.cz/_typo/typo-pravidla-hladka.html>
- (10) *HTML: Hypertext Markup Language* [online]. 2008 [cit. 2008-02-12].
Dostupný z WWW: <<http://en.wikipedia.org/wiki/Html>>
- (11) *HTTP: Hypertext Transfer Protocol* [online]. 2008 [cit. 2008-02-14].
Dostupný z WWW: <<http://en.wikipedia.org/wiki/Http>>
- (12) *Jak psát web: Okaje objektů – margin a padding* [online]. 2008 [cit. 2008-04-01]. Dostupný z WWW: <<http://www.jakpsatweb.cz/okraje.html>>
- (13) *Layout Gala by Alessandro Fulciniti* [online]. 2006 [cit. 2008-01-25].
Dostupný z WWW: <<http://blog.html.it/layoutgala/>>
- (14) *Typografie na počítači: Nejčastější chyby a jak se jim vyhnout* [online]. 2005 [cit. 2008-03-10]. Dostupný z WWW: <<http://typografie.wz.cz/chyby.html>>
- (15) *Typografie na webu* [online]. 2008 [cit. 2008-03-11].
Dostupný z WWW: <<http://www.webtvorba.cz/web/typografie-na-webu.html>>

7. Seznamy

7.1. Seznam obrázků

<i>Obrázek 1: syntaxe HTML tagu</i>	16
<i>Obrázek 2: Správná interpretace okrajů (12)</i>	27
<i>Obrázek 3: Špatná interpretace okrajů podle IE (12)</i>	27
<i>Obrázek 3: Různá interpretace znamének (15)</i>	31
<i>Obrázek 5: Rozdíl mezi výpustkou a třemi tečkami (15)</i>	32
<i>Obrázek 6: Návrh loga #1</i>	34
<i>Obrázek 7: Návrh loga #2</i>	34
<i>Obrázek 8: Návrh loga #3</i>	35
<i>Obrázek 9 : Návrh loga #4</i>	35
<i>Obrázek 10: Schéma rozvržení</i>	38
<i>Obrázek 11: Hlavička webové prezentace</i>	40
<i>Obrázek 12: Pata prezentace</i>	40
<i>Obrázek 13: Horní a spodní část tlačítka menu</i>	40
<i>Obrázek 14: Horní a spodní část tlačítka menu, na němž stojí kurzor myši</i>	40
<i>Obrázek 15: Ukázka levého vertikálního okraje</i>	41
<i>Obrázek 16: Výsledný vzhled stránky</i>	42

7.2. Seznam tabulek

<i>Tabulka 1: Přehled selektorů</i>	20
<i>Tabulka 2: Definice elementů a jejich šířek na stránce dle CSS</i>	38

8. Přílohy

Příloha 1: Zdrojový kód úvodní stránky

Příloha 2: Soubor se stylovým předpisem