

Česká zemědělská univerzita v Praze

Provozně ekonomická fakulta

Katedra informačních technologií



Bakalářská práce

**Analýza možností nových technologií
HTML5 a CSS3**

Vojtěch Mikulka

© 2015 ČZU v Praze

ČESKÁ ZEMĚDĚLSKÁ UNIVERZITA V PRAZE

Katedra informačních technologií

Provozně ekonomická fakulta

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Vojtěch Mikulka

Informatika

Název práce

Analýza možností nových technologií HTML5 a CSS3

Název anglicky

Analysis of possibilities of new technologies HTML5 and CSS3

Cíle práce

Cílem práce je popis a analýza nových možností jazyka HTML 5 a CSS verze 3. Dále porovnání s možnostmi stávajících verzí těchto technologií, rozlišení postupů a metodik tvorby webu a analýza podpory těchto řešení v internetových prohlížečích. Analyzované postupy budou řešeny v rámci praktických příkladů případové studie. Na základě zjištěných výsledků bude formulováno ucelené doporučení pro práci s novými technologiemi.

Metodika

Použitá metodika bakalářské práce bude založena na studiu a analýze dostupných informačních zdrojů. Získané poznatky budou použity ke zpracování a porovnání jednotlivých částí práce. Na jejich základě budou formulovány závěry bakalářské práce.

Výsledky analýzy budou dále zhodnoceny v rámci provedení praktických příkladů případové studie. Na základě zjištěných výsledků bude formulováno ucelené doporučení pro práci s novými technologiemi.

Doporučený rozsah práce

30-40 stran

Klíčová slova

html, html5, css, css3, hypertext, web, internet, webové technologie, programování

Doporučené zdroje informací

Lamberta, Billy a Peters, Keith. Foundation HTML5 Animation with JavaScript. New York : friendsofED, 2011. 978-1430236658.

Lubbers, Peter, Albers, Brian a Salim, Frank. Pro HTML5 Programming. New York : Apress, 2011. 978-1-4302-3865-2.

Pilgrim, Mark. HTML5: Up and Running. O'Reilly, 2010. 978-0596806026.

Weyl, Estelle, Lazaris, Louis a Goldstein, Alexis. HTML5 & CSS3 For The Real World. 5 : 19, 2011. 978-0980846904.

Předběžný termín obhajoby

2015/02 (únor)

Vedoucí práce

Ing. Petr Benda, Ph.D.

Elektronicky schváleno dne 29. 1. 2014

Ing. Jiří Vaněk, Ph.D.

Vedoucí katedry

Elektronicky schváleno dne 4. 3. 2014

Ing. Martin Pelikán, Ph.D.

Děkan

V Praze dne 16. 03. 2015

Čestné prohlášení

Prohlašuji, že svou bakalářskou práci "Analýza možností nových technologií HTML5 a CSS3" jsem vypracoval samostatně pod vedením Ing. Petra Bendy, Ph.D. a s použitím odborné literatury a dalších informačních zdrojů, které jsou citovány v práci a uvedeny v seznamu literatury na konci práce. Jako autor uvedené bakalářské práce dále prohlašuji, že jsem v souvislosti s jejím vytvořením neporušil autorská práva třetích osob.

V Praze dne 16.3.2015

Poděkování

Rád bych touto cestou poděkoval vedoucímu, panu Ing. Petru Bendovi, Ph.D. za jeho čas, rady, vstřícný přístup a trpělivost při vedení bakalářské práce. Dále všem, kteří mě podporovali nejen při psaní této práce, ale i během celého studia.

Analýza možností nových technologií HTML5 a CSS3

Analysis of possibilities of new technologies HTML5 and CSS3

Souhrn

Bakalářská práce se zabývá novými možnostmi značkovacího jazyka HTML5 a kaskádových stylů CSS3 a jejich porovnáním se staršími specifikacemi.

Teoretická část je zaměřena na obecné informace ohledně historie HTML a CSS, jejich použití, technologické prvky, díky kterým je možné s webem pracovat a základy tvorby webových stránek a jazyka HTML.

Praktická část se zabývá vlastním porovnáváním nejprve HTML5 se staršími specifikacemi a následně CSS3 s verzí CSS 2.1. Porovnání je prováděno na základě zdrojových kódů a obecných principů pro řešení konkrétních situací.

Summary

Bachelor thesis deals with new possibilities of markup language HTML5 and Cascading Style Sheets (CSS3) and their comparison with older specifications.

The theoretical part of thesis is focused on general information about history of HyperText Markup Language and Cascading Style Sheets, use and technological components that makes possible to work with web, and basics of developing World Wide Web pages.

The practical part of thesis deals comparing new specification HTML5 with older one. Than compares CSS3 with older CSS 2.1. The comparsion is based on source codes and general principles to solving specific situations.

Klíčová slova: html, html5, css, css3, hypertext, web, internet, webové technologie, programování

Keywords: html, html5, css, css3, hypertext, web, internet, web technology, programming

Obsah

1	Úvod	10
2	Cíle práce a metodika	11
2.1	Cíle práce	11
2.2	Metodika	11
3	Teoretická východiska.....	12
3.1	Internet	12
3.2	World Wide Web	13
3.3	Hypertext Transfer Protocol.....	13
3.4	Jazyky pro tvorbu webu	14
3.5	Webový prohlížeč	14
3.6	Hypertext Markup Language	18
3.6.1	Historie HTML	20
3.7	Cascading Style Sheet	22
3.7.1	Historie CSS.....	22
3.7.2	Použití CSS	23
3.8	Struktura webové stránky.....	25
3.9	Pokročilejší základ	26
4	Praktická část.....	28
4.1	DOCTYPE	28
4.2	Znaková sada – charset	29
4.3	Struktura obsahu.....	29
4.4	Nové značky.....	32
4.4.1	Seskupovací značky.....	32
4.4.2	Stavové ukazatele.....	33

4.4.3 Další nové prvky	35
4.5 Původní značky s novým nebo upraveným významem	36
4.5.1 Formátování textu	36
4.5.2 Další prvky s rozšířenou funkcionalitou	37
4.6 Zastaralé a vyřazené značky.....	37
4.7 Audio, video – multimédia.....	39
4.7.1 Audio.....	39
4.7.2 Video.....	41
4.8 Kaskádové styly – CSS3	43
4.8.1 Zápis kaskádových stylů	43
4.8.2 Pseudotřídy	44
4.8.3 Zakulacené rohy.....	46
4.8.4 Stínování	47
4.8.5 Přejechání	48
5 Zhodnocení výsledků.....	50
6 Závěr.....	52
Seznam obrázků	54
Bibliografie	56

1 Úvod

Součástí a určitým symbolem dnešní doby se staly počítače, které se vyskytují v různých podobách téměř všude. Od jednoduchých a jednoúčelových zařízení, až po složitější a víceúčelové sestavy, se kterými přijde do styku téměř každý. A to například v podobě osobních počítačů, notebooků, chytrých telefonů nebo tabletů.

Prvek, který stále více neodmyslitelně patří k těmto zařízením, je internet. Dostupnost a možnosti připojení se za poslední roky změnily k lepšímu nejen dosahem, spolehlivostí, ale také cenou. Využití internetu je různé. Je možné díky němu například pracovat, komunikovat, využívat jej k zábavě, získávání informací, vzdělávání nebo nákupům. Většina těchto činností probíhá pomocí World Wide Web (WWW) - přes webové stránky neboli webové prezentace.

Uživatelé na ně přistupují pomocí webových prohlížečů, jako jsou Internet Explorer, Mozilla Firefox, Google Chrome, Opera nebo Safari. Aby prohlížeč věděl co, a hlavně jak, má uživateli zobrazit, musí mít zdrojová data, která jsou napsaná v jazyce HTML (Hyper Text Markup Language) spolu s CSS (Cascading Style Sheets). Tato data prohlížeč převede uživateli na čitelnou webovou prezentaci.

Stejně jako rostou nároky uživatelů, požadavky tvůrců a zvyšuje se výpočetní výkon zařízení, tak i tyto značkové jazyky se vyvíjí, rozšiřují a v některých případech zjednodušují. Dnes se HTML nachází u verze s označením 5 a CSS ve verzi 3.

Tématem práce je seznámení s možnostmi nových verzí a porovnání s verzemi předchozími, zhodnocení kladů a záporů a případných rizik.

2 Cíle práce a metodika

2.1 Cíle práce

Cílem práce je popis a analýza nových možností jazyka HTML 5 a CSS verze 3. Dále porovnání s možnostmi stávajících verzí těchto technologií, rozlišení postupů a metodik tvorby webu a analýza podpory těchto řešení v internetových prohlížečích. Analyzované postupy budou řešeny v rámci praktických příkladů případové studie. Na základě zjištěných výsledků bude formulováno ucelené doporučení pro práci s novými technologiemi.

2.2 Metodika

Použitá metodika bakalářské práce bude založena na studiu a analýze dostupných informačních zdrojů. Získané poznatky budou použity ke zpracování a porovnání jednotlivých částí práce. Na jejich základě budou formulovány závěry bakalářské práce.

Výsledky analýzy budou dále zhodnoceny v rámci provedení praktických příkladů případové studie. Na základě zjištěných výsledků bude formulováno ucelené doporučení pro práci s novými technologiemi.

3 Teoretická východiska

Teoretická část bude zaměřena na postupný a stručný popis technologií, díky kterým webové stránky mohou fungovat. Vysvětleny budou některé základní obecné pojmy, které budou dále sloužit k lepší orientaci a pochopení v odbornějších termínech a předmětné problematice. Následovat bude přehled o vývoji Hyper Text Markup Language a kaskádových stylů (CSS) od původních verzí k dnešnímu HTML5 a CSS3, vysvětlení souvislostí mezi HTML5 a CSS3, proč jsou často zmiňovány spolu.

3.1 Internet

Pojem „internet“ se používá velice často právě ve spojení s webovými stránkami, které jsou sice jeho součástí, ale zahrnuje daleko víc možností. Jedná se o celosvětovou síť navzájem propojených zařízení, která spolu komunikují, vyměňují si data, to vše podle určitých protokolů.

Kromě služby World Wide Web (WWW), kterou využíváme pro prohlížení webových stránek vytvořených například právě pomocí HTML5 a CSS3, fungují na internetu i další služby, mezi které patří textové komunikační - e-mail, nebo instant messaging (vzájemná přímá komunikace mezi dvěma a více lidmi, kdy zprávy chodí zpravidla okamžitě).

Dále se v poslední době více rozvíjí VoIP (Voice over Internet Protocol), tedy služba pro přenos hlasu pomocí internetového protokolu, zjednodušeně telefonování přes internet. Mezi její výhody v České republice patří především cena hovoru, která bývá nižší než u mobilních operátorů (pomineme-li relativně nová neomezená volání, u kterých nelze jednoduše určit minutovou sazbu), naopak nevýhodou je potřeba dostupné internetové připojení v odpovídající kvalitě.

3.2 World Wide Web

Služba nazvaná World Wide Web, zkráceně „Web“ nebo WWW. Slouží primárně právě pro internetové stránky, jak jsou známy z internetového prohlížeče. Jedná se o systém dokumentů a souborů, které jsou uloženy na webových serverech a navzájem propojeny pomocí hypertextových odkazů.

Za vznikem této služby stojí Angličan Timothy Berners-Lee, který pracoval pro Evropskou organizaci pro jaderný výzkum (CERN) ve Švýcarsku. V roce 1989 navrhl projekt pro globální sdílení dokumentů pro usnadnění spolupráce, předávání znalostí a výzkumů. O rok později dokončil práci i na prvním prohlížeči a zároveň editoru nazvaném WorldWideWeb a dalších částech potřebných pro správnou funkci webu. (Berners-Lee, 2001)

3.3 Hypertext Transfer Protocol

Hypertext Transfer Protocol (HTTP) slouží k přístupu k webovým stránkám. Jedná se o základní protokol služby World Wide Web, který funguje na principu request – respond, neboli požadavek – odpověď. Uživatel například kliknutím na odkaz odešle požadavek na server (požadavek = chci zobrazit stránku google.com), server tento požadavek zpracuje a odešle zpět uživateli odpověď. Pokud je vše v pořádku, server odpoví zobrazením požadované stránky. Pokud dojde k chybě, například stránka neexistuje, server vrátí chybovou hlášku (Stavový kód HTTP). (Raggett, a další, 1998)

Stavové kódy jsou součástí hlavičky získané v odpovědi od serveru, informují o stavu zpracovaného požadavku, ať už kladném nebo záporném, případně stručně specifikují vzniklý problém. Kódy obsahují tři číslice, první značí druh zprávy, další dvě konkretizují stav.

- 1xx – Informační
- 2xx – Úspěch (požadavek od klienta byl úspěšně přijat a zpracován)
- 3xx – Přesměrování (vyžaduje další akci od klienta)
- 4xx – Chyba na straně klienta (nenalezený obsah, nepovolený přístup)

- 5xx – Chyba na straně serveru (nedostupný, mimo provoz) (Berners-Lee, a další, 1996)

3.4 Jazyky pro tvorbu webu

V případě tvorby jednoduchých stránek je často zmiňován pojem programování, od toho i odvozená činnost programování webových stránek, a dále programovací jazyk HTML. Jedná se však o nepřesné označení.

Programovací jazyk

Jako programování se označuje proces algoritmizace, neboli vytváření postupu vedoucího k řešení dané úlohy. Mezi tyto jazyky patří například Basic, C, C++, C#, Java, PHP. Nikoliv však HTML.

Značkovací jazyk

Značkovací jazyk neslouží k vytváření postupů a algoritmů, ale k obohacení určitého obsahu dodatečnými informacemi, jako například význam určité části textu, strukturování a způsob jeho zobrazení. Do této kategorie HTML spadá, už podle svého názvu. Značkovací jazyk, v angličtině Markup Language. (Clevinger, 2006)

3.5 Webový prohlížeč

Web Browser, česky webový prohlížeč, je aplikace sloužící k zobrazení webových stránek uživateli a určité interakce s nimi. Přijímá od serveru požadované stránky ve zdrojovém kódu tvořeném převážně jazykem HTML a „překládá“ je do uživatelsky přívětivé podoby. Tedy strukturuje a formátuje text, zobrazuje obrázky a další multimediální prvky, označuje odkazy a provádí další úkony tak, jak určil tvůrce internetové stránky.

V úvodu služby World Wide Web byl zmíněn stejnojmenný prohlížeč z roku 1990, který byl prvním a v té době jediným prohlížečem pro zobrazování webových stránek. Později byl přejmenován na Nexus, aby nedocházelo k záměnám mezi softwarem a službou WWW. Díky své jedinečnosti, a tím, že byl součástí vznikajícího systému kolem webu, nemusel se řešit problém podpory ostatních prohlížečů.

V dnešní době existuje mnoho druhů zařízení a operačních systémů (Windows, Linux, MacOS, Android, iOS, Windows Mobile/Phone) a také spousta

prohlížečů. To přináší své výhody, kdy si uživatel vybere prohlížeč, který mu nejvíce vyhovuje po funkční, ergonomické nebo grafické stránce, funkčnosti na jeho systému nebo podle jiných kritérií. Zároveň ale jsou jisté nevýhody v podobě možné nepřesnosti v zobrazení napříč prohlížeči z důvodu nejednotnosti vývoje těchto aplikací.

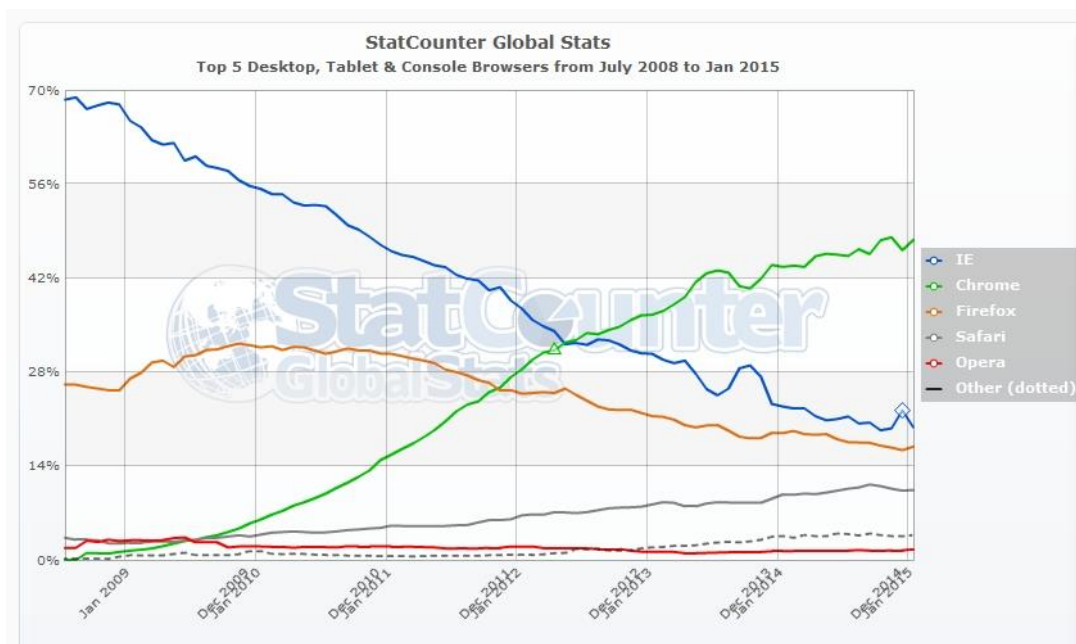
Mezi dnes nejpoužívanější prohlížeče patří:

- Google Chrome
 - Prohlížeč od společnosti Google, představený v roce 2008. Jedná se o multiplatformní prohlížeč, který není vázaný na jeden operační systém. V roce 2012 se stal nejpoužívanějším prohlížečem, kdy na přelomu dubna a května zaujal větší podíl na trhu než Internet Explorer od společnosti Microsoft. (Cullen, 2012)
- Internet Explorer
 - Internet Explorer představila firma Microsoft v roce 1995. Jedná se o prohlížeč pro operační systém Microsoft Windows. Byly dostupné i verze pro UNIX a MacOS, avšak po roce 2000 byl jejich vývoj zastaven, nejprve pro UNIX v roce 2001 a v roce 2003 také pro MacOS. (Microsoft, 2003) Přesto díky rozšíření operačního systému Windows, nemožnosti instalace alternativních prohlížečů na firemních počítačích, případně neochotě či neinformovanosti uživatelů, dominoval Internet Explorer na trhu prohlížečů, až do zmíněného roku 2012. (Cullen, 2012)
- Mozilla Firefox
 - V roce 2002 přibyl na poli internetových prohlížečů projekt s názvem Phoenix, později přejmenován na Firebird, až nakonec v roce 2004 s první stabilní verzí se ustálil na dnešním názvu Firefox. (USPT, 2004) Prohlížeč vyvíjí

Mozilla Corporation, dceřiná společnosti Mozilla Foundation.
(Mozilla.org, 2013)

- Safari
 - Apple Inc. uvolnil v roce 2003 první verzi svého prohlížeče Safari pro MacOS. V letech 2007 až 2012 probíhal vývoj verze i pro Windows, ale kvůli rozšíření Google Chrome fungujícím na stejném jádru se Apple rozhodl vývoj pro Windows ukončit. (Dilger, 2012)
- Opera
 - Norský prohlížeč Opera je vyvíjen od roku 1994 jako výzkumný projekt pod norským telekomunikačním operátorem Telenor. (Opera Software ASA, 2007) Od roku 1995 je Opera vyvíjena pod společností Opera Software, nejprve jako placená aplikace, rokem 2005 pak bez reklam a licenčních poplatků. Snahou vývojářů je co nejvíce dodržovat standardy W3C. (Opera Software ASA, 2005)
- Ostatní
 - Výčetem těchto pěti prohlížečů však trh nekončí, existuje jich více, některé multiplatformní, jiné fungují jen na určitých operačních systémech či zařízeních. Avšak jejich podíl na trhu je minimální, některé nejsou už ani dále vyvíjené. Za zmínku stojí z historického hlediska Netscape, dále lze narazit na prohlížeče Konqueror nebo třeba Maxthon, který byl v roce 2011 na šestém místě, tedy schovaný těsně za „velkou pětkou“. (Tomášek, 2011)

Na následujícím obrázku je rozdělení prohlížečů na trhu od července 2008 do ledna 2013 dle StatCounter. (StatCounter, 2015) Z vývoje je patrný pokles popularity prohlížeče Internet Explorer a naopak častější používání prohlížeče Chrome od Googlu. Vzhledem k určité stálosti používání prohlížečů Firefox, Safari, Opera a ostatních prohlížečů se dá usoudit, že velká část uživatelů Internet Exploreru přešla právě na Google Chrome.



Obrázek 1 - 5 nejpoužívanějších prohlížečů

Zdroj: StatCounter <http://gs.statcounter.com/>

3.6 Hypertext Markup Language

Hypertextový značkovací jazyk (HTML) se stal základním jazykem pro tvorbu internetových stránek. Jazyk se skládá z takzvaných tagů. Jedná se o značky kolem určité textové nebo grafické části dokumentu, které určují význam nebo styl zobrazení této konkrétní části. Značky se uvozují mezi dvojicí špičatých závorek `< >` a jejich funkce je zpravidla vyjádřena anglickým slovem, nebo zkratkou. Jako příklad slouží `<table>` pro tabulku, `` pro zvýraznění textu, `` pro obrázek (image) a `
` jako break, neboli ukončení řádku.

Ve specifikaci HTML jsou dva typy tagů. Prvním jsou párové tagy, které určují začátek a konec jejich funkce. Začátek je zapsán ve špičatých závorkách otevírací značkou `` a ukončovací má navíc lomítko za první závorkou ``. Říkají prohlížeči, odkud a kam má například zdůraznit text, změnit styl písma, která část textu je nadpis. Příkladem je, jak bude vypadat zdůraznění textu ve zdrojovém kódu:

```
Ukázkový text na použití <strong>párového</strong> tagu.
```

Výstup, jak bude zobrazen v prohlížeči:

*Ukázkový text na použití **párového** tagu.*

Pokud by se použil pouze uvozovací tag `` a nikde nebyl ukončen, byl by tučně zvýrazněn celý text až do konce stránky.

Kromě párových tagů jsou některé tagy nepárové, obvykle slouží ke vkládání jednoduchých prvků, které nevozují žádný text. Mezi ně patří tag `` pro vložení obrázku na stránku, nebo ukončení (zalomení) řádku `
`.

Dalším pojmem v HTML je „atribut“. Jedná se o rozšíření funkční povahy daného elementu. Zapisuje se do otevíracího tagu v následujícím tvaru:

```
<tag atribut1="hodnota1" atribut2="hodnota2">text</tag>
```

Pro jeden tag tedy můžeme být více atributů, které jsou odděleny mezerou, hodnoty jsou zapsány do uvozovek. Ukončovací tag atributy neobsahuje.

Atributy mohou v HTML určovat zdroj obrázku, alternativní popis, odkaz, titulky, určovat velikost elementů, ale je s nimi možné také měnit barvu, velikost a styl písma, ohraničení tabulek a další vzhledové úpravy. K těmto vzhledovým úpravám se ale doporučuje využívat kaskádové styly (CSS).

```

```

```
<a href="http://www.google.com">Odkaz na vyhledávač Google</a>
```

Na prvním příkladu je vidět využití atributů pro obrázek, kde je určen zdroj, kde má prohlížeč hledat obrázek k zobrazení, následně alternativní text, který zobrazí v případě nenačtení obrázku, nebo podržení ukazatele myši nad obrázkem a také určení velikosti obrázku.

Druhý příklad ukazuje použití určitého textu jako hypertextový odkaz na webovou stránku. Do uvozovek je ale možné vložit i název souboru pro stažení nebo zobrazení, případně odkaz s kotvou pro přeskočení na jinou část načtené stránky.

I když lze čistě pomocí HTML vytvořit celou stránku i s jednodušším formátováním a grafickým vzhledem, doporučuje se HTML kombinovat s již několikrát zmíněnými kaskádovými styly (CSS), které obohacují HTML po grafické stránce. HTML se stará o obsah, strukturu a význam. Tedy jaké texty budou na stránce a jaký další obsah v podobě obrázků nebo jiných multimediálních prvků bude zobrazen. Určuje, jak budou texty strukturovány a jaký mají význam. Co je nadpis, kde začíná a končí odstavec, kde je umístěn obrázek, který text je důležitý, který text je potřeba zvýraznit, co je odkaz, kde začíná a končí tabulka, kolik má sloupců, řádků a podobně.

Oproti tomu CSS má na starost převážně grafický vzhled, pomocí něj jsou definovány barvy všech prvků na stránce, od pozadí, přes text, rámečky. Také styl písma, velikost nadpisů, způsob zvýraznění, velikost mezer, odstavců, odsazení jednotlivých prvků od sebe a spoustu dalšího. (Castro, a další, 2012)

3.6.1 Historie HTML

Jako každý programovací jazyk, i HTML si prošlo určitým vývojem a ani ten není stále ještě u konce.

3.6.1.1 HTML (1)

První HTML souvisí se začátky World Wide Webu v roce 1990 a jeho zakladatelem Timothy Berners-Lee. Vzniklo za základě standardizovaného univerzálního značkovacího jazyka SGML (Standard Generalized Markup Language) pro potřeby tehdejšího prvního prohlížeče WorldWideWeb (později přejmenovaného na Nexus). V době vzniku nenesla žádné číselné označení a díky neexistenci alternativního prohlížeče byla zajištěná plná podpora zobrazování stránek. (Berners-Lee, 2001)

3.6.1.2 HTML 2.0

V průběhu následujících let došlo k uvolnění a rozvoji alternativních prohlížečů Mosaic od National Center for Supercomputing Applications (NCSA) a v roce 1994 Netscape od Netscape Communications Corporation, což s sebou neslo větší požadavky na rozšíření jazyka. Vývojáři doplňovali HTML o další prvky, proto byla zformována skupina IETF (Internet Engineering Task Force), aby spolupracovali, sjednotili požadavky, výsledky vývoje a v roce 1995 vydali specifikaci HTML 2.0. (Berners-Lee, a další, 1995)

3.6.1.3 HTML 3.0

Se stále rostoucí popularitou webu, kdy v roce 1995 bylo na světě téměř 100 000 serverů jazyka HTML, rostl i hlad po nových funkcích a schopnostech. Vývojáři implementovali velké množství nových tagů do svých prohlížečů, ale nastával problém s kompatibilitou mezi jednotlivými prohlížeči, neboť pokusy o vlastní implementaci efektů konkurence se nedařily. V té době byla utvořena skupina HTML Editorial Review Board, která měla pomoci se standardizací vývoje. Bohužel z důvodů velkých změn a pomalé implementace ze stran prohlížečů byl tento standard zamítnut. (Raggett, a další, 1998)

3.6.1.4 HTML 3.2

Po neúspěchu předchozí verze obsahující velké množství změn, vydalo v lednu 1997 World Wide Web Consortium (W3C) specifikaci HTML 3.2, která byla podobně jako HTML 2.0 souhrnem dosavadního vývoje. Nikoliv objemnější změnou jako v případě HTML 3.0. Velké kroky dopředu byly připraveny pro další verzi, která na sebe nenechala dlouho čekat. (Landofcode.com, 2014)

3.6.1.5 HTML 4.0

Ještě v prosinci roku 1997 vydalo W3 Consortium doporučení na HTML 4.0, které bylo větší evolucí. Na rozdíl od předchozích verzí bylo zapracováno na zjednodušení práce pro webové vývojáře, došlo k částečnému oddělení práce na vzhledu stránek, právě do kaskádových stylů. S tím souvisí i nové tagy pro práci se styly. Jako standard bylo HTML 4.0 přijato v dubnu 1998. (Consortium, 1999)

3.6.1.6 HTML 4.01

O Vánocích roku 1999 byla schválena revize stávajícího standardu HTML 4.0. Revize nesla označení HTML 4.01 a přinesla některé opravy chyb, vzniklých v předchozí verzi, ale také některá vylepšení a nové funkce. (Hlavenka, 2002) Zdálo se, že 4.01 byl posledním standardem pro HTML, neboť W3 Consortium se zaměřilo na vývoj jazyka XHTML (eXtensible Hypertext Markup Language) a k HTML se již nevracelo. Verze XHTML 1.0 a 1.1 byla zpětně kompatibilní a částečně podobná i HTML a v té době bylo vkládáno do XHTML hodně nadějí. (Consortium, 1999)

3.6.1.7 HTML5

V roce 2002 přišlo W3C s projektem XHTML 2.0, které nemělo být kompatibilní s předchozími verzemi, což vedlo k neshodám, následnému oddělení skupiny vývojářů a v roce 2004 založení vlastní pracovní skupiny s názvem WHATWG (The Web Hypertext Application Technology Working Group). Tato skupina pracovala na rozšíření jazyka HTML 4.01. Během té doby dospěli ve W3C k názoru, že XHTML 2.0 je slepá větev a spojili se s WHATWG za účelem pokračování vývojové řady HTML. V říjnu 2014 W3C oznámilo, že se HTML5 dostalo do stádia oficiálního doporučení. Přestože je používáno a podporováno delší dobu. (Consortium, 2014)

3.7 Cascading Style Sheet

Značkovací jazyk HTML původně sloužil ke strukturování dokumentů, které se vyznačovaly pouze textem a základním strukturováním. S rozšířením jeho využití a zapojováním mnohem širšího okruhu lidí kolem internetu, začaly být zvyšovány i nároky na vzhled dokumentů a možnosti jeho uzpůsobení. HTML sice obsahuje tagy a atributy pro úpravu zobrazení, jejich použití je však u rozsáhlejších stránek zbytečně složité na tvorbu i úpravy, znepřehledňuje zdrojový kód a v neposlední řadě nepodporuje myšlenku rozdělení HTML – struktura, obsah; CSS – vzhled a forma.

Z toho důvodu vznikly kaskádové styly, které mají za úkol specifikovat vzhled stránky. Pro tvorbu kaskádových stylů nám stačí, stejně jako u HTML, obyčejný textový editor, pomocí kterého styly zapisujeme.

3.7.1 Historie CSS

Postupným vývojem prošly i kaskádové styly, jejichž historie bude uvedena v této kapitole.

3.7.1.1 CSS1

V roce 1994 navrhl Håkon Wium Lie, pracující v CERNu pod zakladatelem World Wide Webu Timothy Berner-Lee, koncept kaskádových stylů pro web. Po dvou letech vývoje jej W3 Consortium schválilo a tak vzniklo doporučení CSS1. Jsou pod ním podepsaní Håkon Wium Lie a Bert Bos. Z počátku se CSS1 potýkalo s problémy s implementací a podpora prohlížečů byla jen částečná.

CSS1 přineslo podporu obarvení textů, pozadí, ohraničení a dalších elementů, možnosti určit typ písma, jeho proporce, mezery mezi slovy, řádky, odstavci, zarovnání elementů, odsazení, nebo unikátní identifikaci bloků. (Bos, a další, 1996)

3.7.1.2 CSS2

Další rozšíření bylo schváleno v květnu 1998 jako CSS2, obsahující více možností pozicování, grafických možností písma a zobrazování textu.

3.7.1.3 CSS2.1

Následující specifikace CSS2.1 měla poněkud složitější cestu, neboť první pokus o prosazení byl už v únoru 2004, ale byl navrácen k přepracování. Oficiálního vydání jako W3C doporučení se CSS2.1 dočkalo až v červnu 2011. (Meyer, 2006)

3.7.1.4 CSS3

Spolu s verzí HTML5 se vyvíjí i nové kaskádové styly verze CSS3. Přináší velké množství usnadnění a možností pro tvorbu layoutu (vzhledu rozvržení) stránky, ulehčuje nejenom tvorbu stránek, ale i jejich vlastní načítání a prohlížení, díky zjednodušení kódu, možností využít CSS3 na prvky, které dříve musely být vytvořeny v grafickém editoru a vloženy jako obrázek, případně nahrazuje některé funkční prvky, na které byla dřív použita jiná technologie. (Goldstein, a další, 2011)

3.7.2 Použití CSS

Existují tři způsoby, jak můžeme kaskádový styl zapsat:

1. Přimo do tagu (přímý zápis)
2. Do hlavičky webové stránky
3. Do externího souboru.

Přímý zápis do tagu se provádí zapsáním atributu `style` a jako hodnota se použije CSS zápis v uvozovkách. Tento způsob se dá považovat za nejvíc konkrétní, protože ovlivní pouze jeden výskyt daného elementu. Tím ale jeho výhody končí, neboť v případě více výskytů shodného formátování na jedné stránce se musí celý zápis pokaždé opakovat. Tím ztrácí význam zjednodušení a zpřehlednění zdrojového kódu a komplikuje případné úpravy.

```
<p style="color: #336699; font-family: Calibri;">Přímý zápis</p>
```

Obrázek 2 - Styly (přímý zápis)

Zápisem do hlavičky webové stránky je možné výrazně zjednodušit zdrojový kód, pokud se jedná o jednodušší a jednostránkovou prezentaci. Pokud by web obsahoval více oddělených stránek se stejným vzhledem, musela by se hlavička opakovat na každé stránce. V tomto případě se zapíše mezi tagy `<head></head>` (do hlavičky) další párový tag `<style></style>`. Uvnitř tagu `style` lze deklarovat jedno nebo více pravidel.

```

<head>
  <meta charset="utf-8">
  <title>Titulek stránky</title>
  <style>
    p {
      color: #336699;
      font-family: Calibri;
    }
  </style>
</head>

```

Obrázek 3 - Styly (zápis do hlavičky)

Třetí a zároveň nejrozšířenější metodou je zápis do externího souboru, na který se pouze odkazuje ve stránce. Tato metoda je nejuniverzálnější, může se použít pro jeden i více souborů, stejně tak pro jednoduché i složitější weby. Princip spočívá ve vytvoření jednoho (nebo více v případě potřeby) souboru, který bude obsahovat všechna pravidla pro daný web. Následně jedním řádkem v hlavičce stránky se odkáže na soubor, odkud prohlížeč bude čerpat potřebná pravidla. Na tento soubor lze odkazovat z libovolného množství stránek a na všech budou mít elementy stejné vlastnosti. Pokud se tvůrce stránek rozhodne, že změní například barvu písma na celém webu, stačí přepsat jedinou hodnotu v externím souboru stylů a změna se aplikuje na celý web. V případě zápisu stylů do hlavičky stránky by tato změna musela být provedena na každé stránce webu. V případě použití přímého zápisu by změna musela být provedena pro každý jednotlivý odstavec. Z toho jasně plyne výhoda externího souboru s kaskádovými styly.

Propojení se provádí v hlavičce stránky pomocí nepárového tagu `<link>` a atributů `rel`, `type` a `href`, s hodnotami „stylesheet“, „text/css“ resp. odkazem na externí soubor.

```

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

```

Obrázek 4 - Styly (externí soubor - propojení)

3.8 Struktura webové stránky

Aby prohlížeč zpracoval soubor jako webovou stránku, je potřeba vytvořit základní strukturu webové stránky, složenou z tagů. Základní struktura vypadá následovně:

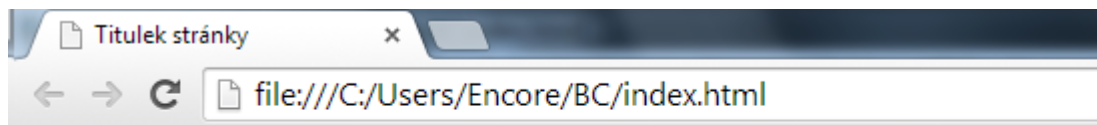
```
<!DOCTYPE html>
<html>
  <head>
    <title>Titulek stránky</title>
  </head>
  <body>
    <p>Libovolný obsah</p>
  </body>
</html>
```

Obrázek 5 - Základní struktura

Tag `<!DOCTYPE html>` určuje prohlížeči, podle jaké specifikace je stránka napsaná a jak ji má zpracovávat. Zde určuje, že se jedná o HTML5 dokument.

Následuje už vlastní struktura HTML stránky, uvozená tagy `<html></html>`. Uvnitř se nachází hlavička `<head></head>`, obsahující informace převážně pro prohlížeč. Zde se jedná o jednoduchou hlavičku, obsahující pouze titulek `<title></title>`, který prohlížeč zobrazí ve svém okně.

Dalším základním prvkem stránky je tělo `<body></body>`, jehož obsah již lze vidět v hlavním okně prohlížeče. Do těla se vkládá veškerý text rozšířený o tagy, který má být na stránce vidět. V tomto případě je zde text „Libovolný obsah“ vložený do tagů `<p></p>`, které ohraničují začátek a konec odstavce.



Libovolný obsah

Obrázek 6 - Základní stránka

Tato struktura je opravdu velmi základní a slouží přehlednému nastínění základů. V praxi je však nedostatečná z důvodů jednoduchosti a absence potřebných tagů.

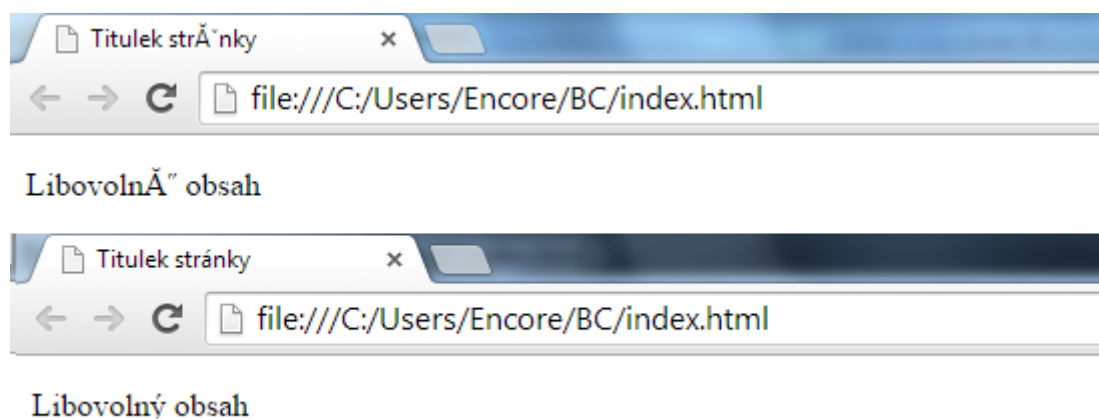
3.9 Pokročilejší základ

Pro spolehlivé a funkční vystavení stránky na web je potřeba doplnit navíc některé další tagy do hlavičky. Pro stránky v češtině je více než důležité upozornit prohlížeč, v jaké znakové sadě je soubor uložen. K tomu slouží atribut `charset`, který se zapisuje do tagu `<meta>`. I když je na výběr z několika možností, dnes se nejčastěji doporučuje využívat UTF-8. Výhodou je jeho univerzálnost a široké rozšíření.

```
<meta charset="utf-8">
```

Obrázek 7 - Charset

Zápis potom vypadá jako na obrázku výše. V případě použití nesprávné znakové sady, nebo její úplné vynechání, nemusely by se správně zobrazit některé české nebo speciální znaky. Pro porovnání následující obrázek: V horní části není uvedena znaková sada, v dolní části ano.



Obrázek 8 - Charset porovnání

Aby web nějak vypadal a nebyl to jen strohý text, používají se kaskádové styly. V úvodu kaskádových stylů byly uvedeny tři možnosti, které se používají pro jejich zápis, z nichž byla vybrána varianta s externí šablonou stylů jako obecně nejlepší. Tato možnost vyžaduje propojení v hlavičce stránky, které se provede dalším nepárovým tagem nazvaným `<link>`. Jako atribut se uvádí `rel`, tedy vztah k souboru. Hodnotou je „stylesheet“. Následně je potřeba určit, kde se soubor nachází. K tomu slouží atribut `href` s hodnotou umístění požadovaného souboru.

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

Obrázek 9 - Link stylesheet

Další často podstatnou součástí hlavičky jsou metadata pro vyhledávače, která usnadní vyhledávačům nalézt a zařadit naše stránky a následně je zpřístupnit ve vyhledávání. K tomu slouží atribut `name` s hodnotami „keywords“ a „description“. Následně do atributu `content` patří obsah, tedy klíčová slova a stručný popis.

Jednou další hodnotou je i `author`, kde se může uvést autor stránek.

```
<meta name="description" content="Popis: Ukázková stránka">
<meta name="keywords" content="bachelors, web, HTML5">
<meta name="author" content="Vojtěch Mikulka">
```

Obrázek 10 - Meta name

Výše uvedený obsah hlavičky je dostatečný pro běžné stránky, není však konečný, je možné přidat další prvky, kombinovat je, nebo měnit pořadí některých z nich, neboť jejich pořadí má vliv na fungování stránek. Jedná se ale o pokročilejší možnosti. Výsledný náhled na zdrojový kód hlavičky stránky:

```
<head>
  <meta charset="utf-8">
  <title>Titulek stránky</title>
  <link rel="stylesheet" type="text/css" href="main.css">
  <meta name="description" content="Popis: Ukázková stránka">
  <meta name="keywords" content="bachelors, web, HTML5">
  <meta name="author" content="Vojtěch Mikulka">
</head>
```

Obrázek 11 - Kompletní hlavička

4 Praktická část

V praktické části bakalářské práce budou popsány a na ukázkových kódech předvedeny základní rozdíly mezi HTML 4.01 a HTML5 a novinky, které HTML5 přináší. Stejně tak v případě kaskádových stylů.

4.1 DOCTYPE

V té nejzákladnější struktuře stránky se oproti předchozí verzi mnoho nezměnilo, pořadí a význam tagů je stále stejný. Nicméně jedním prvkem, který nesmí chybět na žádné stránce je DOCTYPE, document type declaration. Určuje v jakém značkovacím jazyce a jakém režimu je stránka napsaná. Tuto informaci přebírá prohlížeč a podle ní stránku vykresluje.

V případě starší verze HTML 4.01, která byla založená na SGML, se používaly tři definice, podle kterých mohla být stránka napsána:

- HTML 4.01 Strict
- HTML 4.01 Transitional
- HTML 4.01 Frameset

Definice se lišily přísností a s tím souvisejícími povolenými postupy a tagy, využívanými při tvorbě daného webu. Od nepřísnějšího HTML 4.01 Strict, který prezentační (vzhledové) tagy odděluje do stylů, přes HTML 4.01 Transitional, jenž je benevolentnější k použití prezentačních tagů v HTML, až k HTML 4.01 Frameset, povolujícímu navíc rámy (frameset).

Zápis starého DOCTYPE uváděl jeho název a odkaz na příslušný DTD, neboli document type definition, soubor s konkrétními definicemi. Pro jednotlivé verze vypadal takto:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">  
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">  
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN" "http://www.w3.org/TR/html4/frameset.dtd">
```

Obrázek 12 - DOCTYPE 4.01

Syntaxe nového HTML5 již není vytvořena na základě SGML (Standard Generalized Markup Language), přestože se na první pohled zdá velmi podobná. Z toho důvodu není potřeba odkazovat na DTD, avšak DOCTYPE zůstalo zachováno pro informaci, že dokument je vytvořen právě v HTML5. Nový zápis je velmi zjednodušen.

```
<!DOCTYPE html>
```

Obrázek 13 - DOCTYPE 5

Výhodou tohoto nového zápisu je snadnější zapamatování pro kodéry, kteří doposud tento zápis převážně kopírovali z již existujících projektů.

4.2 Znaková sada – charset

Další zjednodušení se odehrává hned v hlavičce stránky, mezi tagy `<head></head>`. Jedná se o definování znakové sady, ve které je dokument (webová stránka) uložen. Důsledky špatného nebo chybějícího definování znakové sady dokumentu byly ukázány v kapitole Pokročilejší základ.

Ve starší verzi bylo potřeba pro definování znakové sady provést složitější zápis:

```
<meta http-equiv="content-type" content="text/html; charset=utf-8">
```

Obrázek 14 - Charset HTML 4.01

Nové HTML5 přináší snazší zápis na zapamatování:

```
<meta charset="utf-8">
```

Obrázek 15 - Charset HTML 5

4.3 Struktura obsahu

Většina webových stránek jsou složitější prezentace, skládající se z více stránek a různého druhu obsahu. Mezi jednodušší typy patří dokumenty složené z hlavičky obsahující logo firmy, následně vertikálního menu, pomocí kterého se lze pohybovat po zbytku webu, následuje vlastní obsah stránky s informacemi pro

návštěvníky a obvykle na konci bývá patička, ve které je více či méně podstatných informací, které jsou pro celý web neměnné. V patičce se často objevují i prvky podporující SEO. (Kubíček, 2008) Pro takovéto logické rozdělení stránky se používají blokové elementy `<div> </div>`, které jako atribut mají jedinečné ID. Jako hodnota ID bývá nejčastěji uváděna hlavní funkce nebo účel daného bloku. V HTML 4.01 bylo tedy možné použít pro vytvoření úvodní hlavičky obsahu stránky ID „header“, pro menu „menu“, případně „nav“, pro hlavní článek/obsah „main“ a pro patičku se často využívá hodnota „footer“. Na ukázkou jednoduchý kód:

```
<body>
  <div id="header">
    
  </div>
  <div id="menu">
    <ul>
      <li>Domů</li>
      <li>Co děláme</li>
      <li>Profil společnosti</li>
      <li>Kontakt</li>
    </ul>
  </div>
  <div id="main">
    <h1>Nadpis hlavní části obsahu stránky</h1>
    <p>Obsah právě načtené stránky.</p>
  </div>
  <div id="footer">
    <p>Mapa stránek, smluvní podmínky a další drobné info.</p>
  </div>
</body>
```

Obrázek 16 - Sekce HTML 4.01

Nevýhodou použití identifikátorů je jejich jedinečná použitelnost v dokumentu, nesmí se opakovat. Především z důvodu jejich rozšířené funkcionality jako kotva pro adresování hypertextovým odkazem.

Ve výše uvedené ukázce je stránka rozdělena na čtyři hlavní části. Bloky `<div>` je možné do sebe vnořovat a tím upravovat například různé odsazení, pozicování, jinak barevný vzhled, nebo tím obejít nekompatibilitu či nedokonalost

některých prohlížečů. V případě složitějšího webu pak dochází k horší přehlednosti kódu a předivování, kdy se mohou objevit nadbytečné bloky.

HTML5 přichází v této problematice s novými sémantickými tagy, tedy takovými, které mají sami o sobě význam. Patří mezi ně `<header>`, `<nav>`, `<section>`, `<article>`, `<aside>` a `<footer>`. Jedná se o párové blokové tagy, tedy každý má svůj ukončovací protějšek `</navez_tagu>`. Už na první pohled lze z tagů jednoduše vyčíst, k čemu se používají, tedy na hlavičku, navigaci, hlavní sekci stránky, jednotlivé články, poznámky na straně a patička. Na rozdíl od tagů `<div>`, nebo podobnému ``, které nejsou sémantické a význam pro pochopení je jim dán až hodnotou atributu ID. A to ne vždy, ID může být i náhodný řetězec, číslo.

Přepracuje-li se ukázkový kód dle nové specifikace, vznikne nejenom na první pohled přehlednější zdrojový kód, ale také lepší funkčnost stránky například pro čtecí zařízení v rámci Blind Friendly Web, které budou mít jasně definováno, kde začíná hlavní obsah a co je součástí menu.

```
<body>
  <header>
    
  </header>
  <nav>
    <ul>
      <li>Domů</li>
      <li>Co děláme</li>
      <li>Profil společnosti</li>
      <li>Kontakt</li>
    </ul>
  </nav>
  <section>
    <h1>Nadpis hlavní části obsahu stránky</h1>
    <p>Obsah právě načtené stránky.</p>
  </section>
  <footer>
    <p>Mapa stránek, smluvní podmínky a další drobné info.</p>
  </footer>
</body>
```

Obrázek 17 - Sekce HTML 5

Že se jedná opravdu o nové tagy, ukazuje i červené zvýraznění těchto tagů v programu PSPad 4.5.2. Zvýraznění je způsobeno neaktuální sadou HTML

zvýrazňovačů v této verzi programu, které nerozeznají nové tagy jako součást HTML5, ale považují je za neznámá slova (Unknown Word).

Jedním ze záměrů těchto značek byly přesnější výsledky vyhledávání nebo lepší pozice webových stránek ve vyhledávačích z pohledu SEO (Search Engine Optimization). Dnešní vyhledávací algoritmy však zatím neberou tyto značky v potaz z důvodů znevýhodnění starších webů používajících tagy `<div>`. Naopak velkou výhodou je značná podpora u nových čtecích zařízení pro zrakově postižené uživatele. (TyfloCentrum Brno, 2015)

Výhoda a zlepšení orientace je znát hlavně u složitějších kódů, v případě takto jednoduchých není přínos z pohledu uživatelské orientace v kódu nějak významný.

4.4 Nové značky

Kromě zmíněných základních strukturálních značek přináší HTML5 několik dalších nových elementů přinášejících nové funkce a prvky do stránky.

4.4.1 Seskupovací značky

V nové specifikaci přibyly značky pro skupiny k sobě patřících elementů v kódu. Slouží především pro jeho přehlednost a strojové zpracování. Na zobrazeném výsledku stránky se tyto značky neprojevují.

figure

Párový tag `<figure> </figure>` slouží jako blokový element, především k seskupení významově spolu souvisejících prvků na stránce (článek, textu). Nejjednodušší příklad je obrázek a jeho popis, kdy do jednoho kontejneru `figure` lze vložit obrázek `` a k němu přidružit vysvětlující popis, pomocí také nového tagu `<figcaption>`. Použití kontejneru `<figure>` zdůrazňuje strukturu dokumentu určením, kde začíná a končí jiný prvek, přerušující souvislost prvku jiného (obvykle ilustrace v souvislém textu). Usnadňuje také grafické formátování stránky pomocí CSS. Není potřeba pozicovat například zvlášť obrázek a zvlášť popis, případně další prvky v kontejneru, ale lze tuto činnost zvládnout pozicováním pouze kontejneru `figure`.

Použití `<figure>` a `<figcaption>` může být následující:


```
<figure>
  
  <figcaption>Komentář, popis výše uvedeného obrázku.</figcaption>
</figure>
```

Obrázek 18 – Figure

hgroup a ukázka vývoje jazyka

Další tag, který by patřil na toto místo, by byl `<hgroup> </hgroup>`. Opět se jedná o párový tag a také by se dal nazvat kontejnerem, obsahujícím několik dalších elementů. V tomto případě nadpisů. Hgroup totiž sdružuje nadpisy a podnadpisy v textu, tedy pokud je potřeba umístit více nadpisů různých úrovní bezprostředně za sebou.

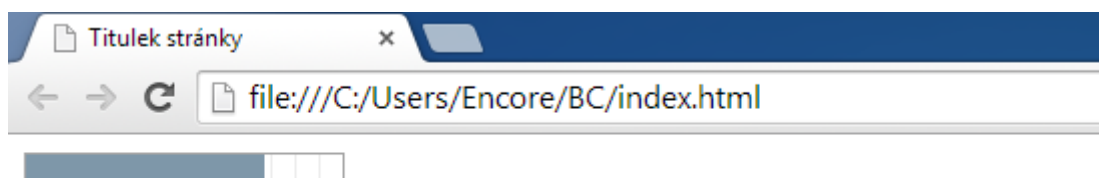
Tento tag je však ukázkou postupného vývoje jazyka HTML5, kdy byl nasazen, podporován prohlížeči, byla sepsána dokumentace, vznikly návody a popisy, jak hgroup používat. V dubnu 2013 ale došlo k rozhodnutí vyjmout hgroup ze specifikace a dále jej nepoužívat.

4.4.2 Stavové ukazatele

Pro zjednodušení tvorby stavových ukazatelů byly do specifikace zařazeny nové elementy, které minimalizují nutnost využívání JavaScriptových prvků, případně jejich alternativ v podobě obrázků a dalších.

progress

Další novinkou v HTML5 je párový `<progress> </progress>`, vytvářející známý progress bar, neboli ukazatel průběhu, postupu. Jeho využití je velice široké, od zobrazování průběhu stahování nebo nahrávání souborů, přes ukazatel postupu prací a další záležitosti ohledně práce, úkolů a jejich průběhu.



Obrázek 19 - Progressbar – ukázka

Jak už bylo zmíněno, `progress` je párový, dovnitř tagu se zapisuje text, případně jiný element, který se zobrazí v prohlížečích nepodporujících tuto funkci. Jde o obdobu alternativního textu. Má dva atributy, první je `value`, který nastavuje hodnotu pokroku (příp. počet hlasů). Druhý atribut je `max`, značící nejvyšší dosažitelnou hodnotu. Na W3C School je uvedeno vysvětlení pro atributy následovně: `Max` specifikuje, kolik práce je potřeba pro dokončení úkolu. `Value` specifikuje, kolik práce již bylo provedeno.

Není-li uvedena hodnota `max`, bere se výchozí rozsah ukazatele od 0 do 1 a hodnoty je třeba zadávat na intervalu nula až jedna. Pokud není uvedena hodnota `value`, ukazatel se přepne do stavu neznámé hodnoty.

Grafické provedení ukazatelů lze upravit pomocí kaskádových stylů, ve výchozím nastavení se mění v závislosti na použitém operačním systému a webovém prohlížeči.

```
<progress value="75" max="100">Získáno 75% hlasů.</progress>
```

Obrázek 20 - Progressbar – kód

meter

Velmi podobným ukazatelem je `<meter> </meter>`. Po funkční stránce je také párový, mezi tagy je umístěn alternativní text pro případ, že by ukazatel nebyl prohlížečem podporován. Významově jde o ukazatel určité části z celku, kdy je definovaná maximální hodnota a zároveň se nejedná o pokrok na práci nebo úkolu. Tedy nelze jej využít pro zobrazení například váhy, výšky (nelze určit maximální hodnotu), ani průběhu práce na úkolu (pro to slouží `<progress>`).

Pro `meter` lze definovat atributy `low`, `high`, `min`, `max`, `optimum` a `value`. `Low` a `high` určují, kdy je hodnota považována na nízkou a vysokou, `min` a `max` určují nejnižší a nejvyšší hodnotu, `optimum` určuje optimální hodnotu a `value` aktuální, skutečnou.

4.4.3 Další nové prvky

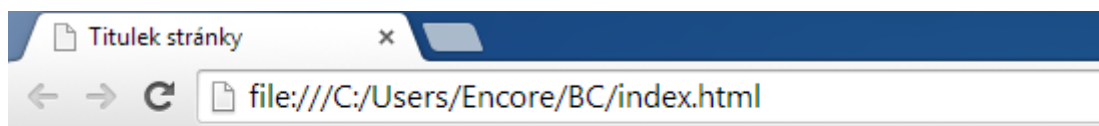
mark

Novým párovým tagem je také `<mark>` `</mark>`. Slouží ke zvýraznění části textu, která je nějakým způsobem důležitá. Může jít o výsledky vyhledávání, citace, klíčová slova nebo jiné části textu, které autor uzná za vhodné. Pro `<mark>` je možné definovat různé styly zobrazení pomocí kaskádových stylů, výchozí je žlutě podbarvený text.

Následuje ukázka výchozího zvýraznění.

```
<p>Areál se nachází na adrese <mark>Kamýcká 129</mark>.</p>
```

Obrázek 21 - Mark - kód



Areál se nachází na adrese **Kamýcká 129**.

Obrázek 22 - Mark – zobrazení

time

Datum a čas je pro člověka snadno pochopitelný v různých formátech, horší je to ale pro stroje, vyhledávače. Na některých webech je uvedeno datum pouze čísly (9. 7. 2015), jinde je rozepsaný měsíc (9. července 2015). Další problém nastává na anglických a amerických webech, kdy jsou někde uvedeny první v pořadí dny, jinde měsíce, neboť britská angličtina začíná datum dnem 9/7/2015, zatímco americká angličtina měsícem 7/9/2015. Pro zjednodušení vyhledávačům proto vznikl párový tag `<time>` `</time>` s atributem `datetime`, obsahujícím strojově čitelné hodnoty podle přesně definovaného schématu. Uvnitř párového tagu se může nacházet datum v libovolném formátu, jeho hodnota by ale měla korespondovat s hodnotou atributu `datetime`.

```
<p>Právě je <time datetime="2015-07-09 18:25:45">9. 7. 2015, 18:25</time>.</p>
```

Obrázek 23 – Time

4.5 Původní značky s novým nebo upraveným významem

Další změnou prošly v nové verzi některé stávající značky, které již existovaly dříve, ale v HTML5 dostaly nový význam. Většina značek vypadá a chová se stejně, rozdíl je v jejich sémantickém významu.

4.5.1 Formátování textu

Pro potřeby formátování textu existují následující značky.

b – bold (tučný text)

Párový tag `` po grafické stránce dělá tučný text. Protože po sémantické stránce neměl `` význam, byl postupně vytlačován sémantickým `` pro zdůraznění důležitého textu. S HTML5 se vrací do specifikace a je s ním počítáno pro zvýraznění textu, který je třeba zdůraznit a odlišit, ale nemá žádný zvláštní význam.

i – italic (kurzíva)

Stejně jako tag ``, vrací se i tag `<i>` pro mírné sklonění textu doprava. Platí pro něj podobné změny, jako pro tučný text a používá se tam, kde je třeba odlišit slovo nebo část textu od zbytku, ale nemá v něm žádný zvláštní význam oproti zbytku, aby se dalo použít ``, plnicí po grafické stránce stejný účel.

s - strike

Do specifikace HTML5 se vrací i párový tag `<s>`, sloužící pro přeškrtnutí textu. Starší verzi HTML 4.01 byl označen jako zastaralý a nahrazen sémantickým tagem `` pro smazaný text. HTML5 přidává význam i pro `<s>` jako část textu, která není nadále platná. Praktické využití může mít ve větě nebo článku, kdy by smazáním neplatné části textu ztratil článek význam. Případně pro zdůraznění změny, novinky.

small

Dalším párovým tagem je `<small> </small>`, slouží stejně jako dříve pro menší text, v HTML5 ale získává sémantický význam. Definuje postranní komentáře a texty drobným písmem.

strong

Párový tag ` ` po grafické stránce určuje tučné písmo. Technické ani grafické provedení se nezměnilo, na rozdíl od sémantiky, kdy `` dostal nový význam. Dříve určoval zdůrazněný text, v HTML5 definuje důležitý text.

4.5.2 Další prvky s rozšířenou funkcionalitou

hr – horizontal rule

Nepárový tag `<hr>` slouží ke grafickému horizontálnímu oddělení dvou odstavců nebo prvků na stránce. Ve starších verzích šlo pouze o vykreslení čáry bez zvláštního udaného významu. HTML5 toto upřesňuje a definuje `<hr>` jako tematickou pauzu pro rozdělení textu na více částí. Grafické pojetí zůstává nezměněno. Oproti starší verzi nejsou dovoleny atributy (`align`, `noshade`, `size`, `width`) přímo v tagu, ale je nutné provést případné úpravy pomocí CSS.

script

Párový tag `<script> </script>` definuje skript na straně klienta (například JavaScript). Významově se nic nemění oproti starším verzím, mění se ale některé jeho atributy. Novinkou je atribut `async`, určující asynchronní spouštění skriptů. Naopak není dále podporován atribut `xml:space` pro zachování mezer ve skriptu.

4.6 Zastaralé a vyřazené značky

S HTML5 přišlo i vyřazení některých značek ze specifikace. Jsou tři hlavní skupiny:

1. Značky, jejichž význam je čistě prezentační
2. Značky, které zhoršují použitelnost a přístupnost
3. Značky, které nebyly často používány, nebo byly nahrazeny jinou.

Vyřazením ze specifikace ale neznamená, že jsou značky nefunkční. Jen nejsou validní pro HTML5 a neměly by se používat.

basefont

Nepárový tag `<basefont>` sloužil k výchozímu nastavení stylu, velikosti a barvy písma na stránce. Ukládal se do hlavičky stránky. Jeho funkci přebraly kaskádové styly.

big

Pro zvětšený text sloužil párový tag `<big> </big>`, pro který ale nebyl použit sémantický význam. Nahrazen je opět funkcemi kaskádových stylů a v HTML5 není specifikován.

center

Pozicovací párový tag `<center> </center>` se využíval pro umístění prvků a textů na střed stránky. Jedná se čistě o prezentační tag, nahraditelný kaskádovými styly, proto byl z nové specifikace odebrán.

font

Stejně jako v případě `<basefont>` sloužil párový tag ` ` k nastavení stylu (typu) písma, jeho velikosti a barvy. `` se používal pro změnu oproti výchozímu nastavení stránky. Lze jej nahradit pomocí kaskádových stylů, a proto není v nové specifikaci definován.

strike

Párový `<strike> </strike>` obaluje text, který má být přeškrtnutý. Na rozdíl od `<s> </s>` a ` ` nemá ani sémantický význam, proto byl ze specifikace HTML5 vyřazen.

tt

Pod tagem `<tt> </tt>` se skrývá Teletype Text. Jedná se o text, který je napsán písmem s pevnou šířkou znaků. Tedy znak ‚m‘ zabere stejně místa ve slově, jako znak ‚i‘. Používá se například pro přehlednost při psaní kódů. Z důvodů chybějícího sémantického významu a existenci jiných značek jako třeba `<code>`, byl vyřazen.

4.7 Audio, video – multimédia

Webové stránky nejsou jen o textu, jednoduché grafice a statických obrázcích. Některé stránky obsahují hudbu, hlasové nahrávky nebo zvuky, na jiných nalezneme videa.

Před příchodem HTML5 se implementace multimédií do webových stránek řešila například pomocí animovaných obrázků (.gif), nebo některých zásuvných modulů, jejichž typickými zástupci jsou Adobe Flash, Apple QuickTime, nebo Windows Media.

Díky HTML5 je možné multimédia vkládat přímo do stránek, bez nutnosti využívat externí přehrávače, skripty a aplikace třetích stran, u nichž není zaručena kompatibilita, dostupnost na uživatelsky zařízením ani funkčnost nebo podpora do budoucna. Základem je multimediální obsah v podporovaném formátu, jako například zvuk, hudba v mp3, nebo video ve formátu mp4. Tento soubor je potřeba mít uložen na webu, buď na vlastním serveru (hostingu), nebo je možné se na soubor odkazovat jinam. Nevýhoda uložení na vlastním hostingu je zabírání kapacity úložiště, odkazování jinam zase nezaručí funkčnost do budoucna, pokud nemá autor stránek externí úložiště pod kontrolou.

4.7.1 Audio

Základem umístění zvukového souboru na stránku je soubor, obsahující zvukovou stopu v podporovaném formátu. Protože existuje velké množství zvukových formátů a kodeků, je náročné zajistit podporu všech. Pro HTML5 byly vybrány tři z nejrozšířenějších, mezi které patří MP3 (MPEG Audio Layer III), WAV (Waveform Audio File Format) a OGG. Tento soubor je možné uložit na vlastní sever nebo na vzdálené úložiště, kde bude soubor pro přehrávač volně dostupný.

Další částí je vložení přehrávače na stránku. K tomu slouží nový párový tag `<audio> </audio>`. Dovnitř tagu se zadává další tag `<source>`, který je nepárový a udává cestu k audio souborům a jejich typ, pokud je k dispozici více formátů. Přehrává se pak první podporovaný. Dalším prvkem, který je mezi značkami `<audio> </audio>` je text, který se zobrazí v případě, že návštěvníkův prohlížeč nebo nastavení nepodporuje přehrávání zvukových souborů.

K `<audio>` patří několik atributů sloužících k nastavení a upřesnění chování přehrávače na stránce. První čtyři atributy jsou typu boolean, tedy nezadává se k nim žádná hodnota. Pokud jsou uvedeny, jejich funkce se provádí.

- `autoplay`: Určuje, zda bude zvuk přehrán automaticky, jakmile bude připraven.
- `controls`: Určuje, budou-li zobrazena ovládací tlačítka přehrávače.
- `loop`: Určuje smyčku, tedy přehrávání stále dokola.
- `muted`: Přehrávání začne, ale hlasitost je nulová.
- `preload`: Určuje, jakým způsobem proběhne přednačítání. Hodnotami jsou `auto` (načtení audiosouboru proběhne zároveň s načítáním stránky), `metadata` (během načítání stránky se načtou pouze metadata audiosouboru) a `none` (audiosoubor není načten během načítání stránky, ale až po aktivaci uživatelem).
- `src`: Pokud existuje soubor jen v jednom formátu, lze zadat odkaz na zdroj jako hodnotu zde. Je možné zadat relativní i absolutní cestu.

Zjednodušený zápis s jedním souborem, zapnutým automatickým spuštěním přehrávání a zobrazeným přehrávačem:

```
<audio autoplay controls src="audio1.mp3">  
  Prohlížeč nepodporuje audio.  
</audio>
```

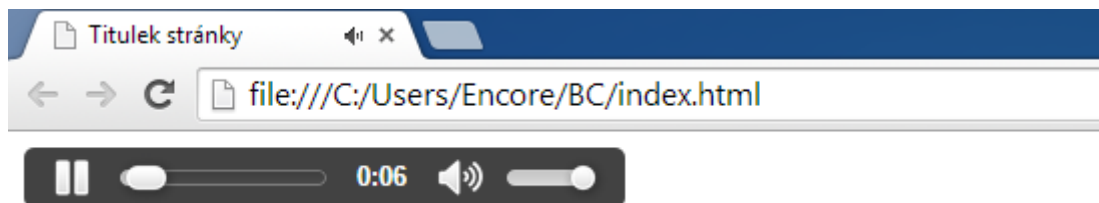
Obrázek 24 - Audio zjednodušené

Zápis pro případ, že je audio k dispozici ve více formátech, opět se zapnutým automatickým spuštěním a zobrazeným přehrávačem:

```
<audio autoplay controls>  
  <source src="audio1.mp3" type="audio/mpeg">  
  <source src="audio1.ogg" type="audio/ogg">  
  Prohlížeč nepodporuje audio.  
</audio>
```

Obrázek 25 - Audio více formátů

Na stránce pak přehrávač vypadá například jako na následujícím obrázku. Opět se může lišit podle použitého prohlížeče, zde je na ukázkou přehrávat v Google Chrome pro Windows.



Obrázek 26 - Audio přehrávač

4.7.2 Video

Pro video jsou podmínky a pravidla téměř shodná s audiem. Základem je videosoubor v podporovaném formátu se správným video a audio kodekem. V případě HTML5 jsou podporovány soubory mp4 (MPEG4, obsahující kodek H264 pro video a AAC audio), dále pak WebM (s kodekem VP8 pro video a Vorbis pro audio) a třetím podporovaným typem je Ogg, obsahující Theora video kodek a opět Vorbis audio.

Obdobné je to i se zápisem v HTML5, pro videopřehrávač se používají párové tagy `<video>` `</video>`. Uvnitř tohoto tagu může být jeden nebo více tagů `<source>` pro odkaz na kompatibilní videosoubory. V případě více tagů `<source>` bere prohlížeč nejprve první soubor, pokud není kompatibilní, zkouší přehrát druhý, třetí. Dále uvnitř `<video>` nalezneme alternativní text pro případ, že video tag není podporován prohlížečem.

Pro `<video>` jsou stejné i atributy, sloužící pro nastavení chování přehrávače a přehrávaného souboru, první čtyři atributy jsou typu boolean:

- `autoplay`: Určuje, zda bude video přehráno automaticky, jakmile bude připraveno.
- `controls`: Určuje, budou-li zobrazena ovládací tlačítka přehrávače.
- `loop`: Určuje smyčku, tedy přehrávání stále dokola.
- `muted`: Přehrávání začne, ale hlasitost je nulová (ztlumeny).

- `preload`: Určuje, jakým způsobem proběhne přednačítání. Hodnotami jsou `auto` (načtení videosouboru proběhne zároveň s načítáním stránky), `metadata` (během načítání stránky se načtou pouze metadata videosouboru) a `none` (videosoubor není načten během načítání stránky, ale až po aktivaci uživatelem).
- `src`: Pokud existuje soubor jen v jednom formátu, lze zadat odkaz na zdroj jako hodnotu zde. Je možné zadat relativní i absolutní cestu.

Video přidává tři nové atributy pro přehrávač:

- `height`: Nastavuje výšku videopřehrávače.
- `width`: Nastavuje šířku videopřehrávače.
- `poster`: Specifikuje absolutní nebo relativní cestu (odkaz) na obrázek, který se zobrazí během načítání videa. Pokud není atribut `poster` použit, vezme se první snímek z videa.

Ukázkový kód obou případů, jak se dá video vložit do stránky. Opět je první možnost zjednodušená a uvádí odkaz pouze na jeden formát videa, v druhé ukázce je vidět, že zdrojový soubor je dostupný ve dvou formátech (MP4 a WebM). Pokud nepůjde přehrát soubor MP4, prohlížeč jej přeskočí a zkusí přehrát WebM. Není nastaveno autoplay, jako v předchozím případě audia, jsou však zadány rozměry video přehrávače a samozřejmě zobrazeno ovládání atributem `controls`.

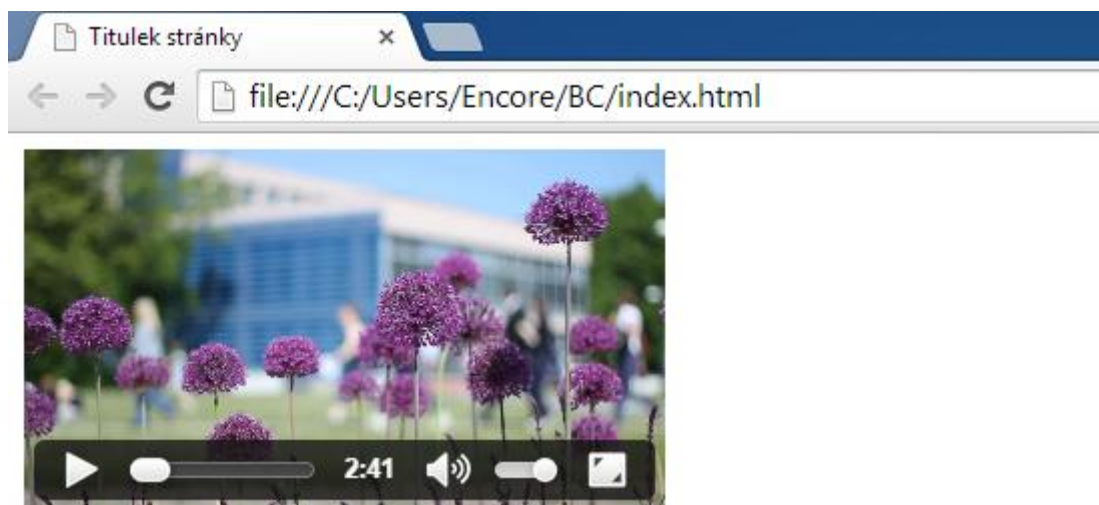
```
<video src="video.mp4" width="360" height="240" controls>
Prohlížeč nepodporuje HTML5 video.
</video>
```

Obrázek 27 - Video zjednodušené

```
<video width="360" height="240" controls>
  <source src="video.mp4" type="video/mp4">
  <source src="video.webm" type="video/webm">
  Prohlížeč nepodporuje HTML5 video.
</video>
```

Obrázek 28 - Video více formátů

Videopřehrávač vypadá na stránce přibližně takto, opět záleží na zvoleném prohlížeči a nastavení, proto se může lišit:



4.8 Kaskádové styly – CSS3

Nová verze kaskádových stylů CSS3 přichází převážně s usnadněním tvorby webové grafiky. Nabízí usnadnění při tvorbě grafických prezentací, kdy ve starších verzích bylo potřeba složitějšího kódu, používání obrázků, které navyšovaly velikost webové stránky, případně použití dokonce jiných jazyků jako například JavaScript nebo náročnějších Flashových animací.

Styl a propojení s HTML5 bylo nastíněno v teoretických východiscích této práce, následovat proto bude ukázka zápisu kaskádových stylů a představení nových možností CSS ve verzi 3.

4.8.1 Zápis kaskádových stylů

V kapitole Použití CSS byl zmíněn jako obecně nejlepší způsob zápisu použití externího souboru a jeho propojení s webovou stránkou. Následující část práce bude proto zaměřena na konkrétní zápisy do externího stylopisu.

Soubor stylopisu se nijak nevozuje, obsahuje pouze vlastnosti prvků a jejich hodnoty, případně komentáře. Vlastnosti jsou mezi sebou odděleny středníkem. Je možné provádět zápis v řádku, což ale není moc přehledné. Proto se používá blokový zápis, jako je na následujícím obrázku. Vždy je potřeba určit, jaký element bude upravován, vlastnosti a hodnoty jsou ve složených závorkách. Mezi vlastnost a hodnotu patří dvojtečka. Na rozdíl od HTML není potřeba hodnoty dávat

do uvozovek. Komentáře mohou být kdekoliv, začátek komentáře se provádí zápisem znaků `/*` a ukončuje `*/`.

```
elementHTML {
  vlastnost1: hodnota1; /* komentář k vlastnosti */
  vlastnost2: hodnota2;
  vlastnost3: hodnota3;
}
```

Obrázek 29 - CSS struktura

4.8.2 Pseudotřídy

Elementy je možné rozšiřovat pomocí pseudotříd, kdy dochází k doplnění funkčnosti daného prvku. Ve starší specifikaci jsou často využívány pseudotřídy ve spojení s hypertextovými odkazy na stránce. Dle nastavení je možné měnit jejich chování v neaktivním stavu, při najetí kurzorem, jinak zvýraznit kliknutí nebo podobu již navštíveného odkazu. K tomu slouží pseudotřídy `:link`, `:hover`, `:active`, `:focus`, `:visited`.

Na ukázce je vidět zápis pseudotříd pro změnu barvy odkazů.

```
selektor:pseudotřída { vlastnost:hodnota;}
a:link {color: blue;} /* nenavštívený odkaz bude modrý */
a:hover {color: yellow;} /* při najetí kurzorem odkaz zežloutne */
a:active {color: red;} /* při nakliknutí bude odkaz červený */
a:visited {color: blue;} /* navštívený odkaz bude modrý */
```

Obrázek 30 - CSS pseudotřídy

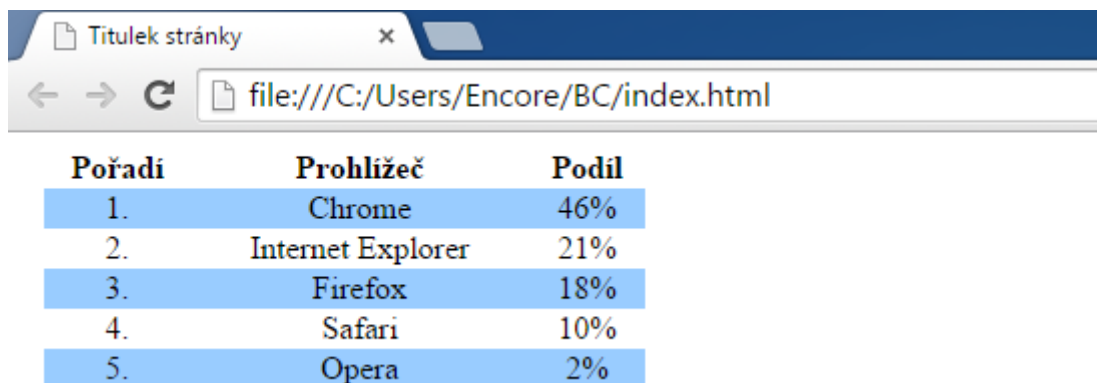
Specifikace CSS3 přináší nové pseudotřídy, se kterými je možné pracovat a usnadnit kodérům a webdesignerům práci při tvorbě webu. Jedním z příkladů je práce s tabulkami a jejich grafické odlišení řádků. V případě CSS2.1 bylo potřeba v HTML kódu určit každému řádku atribut `class` a jako hodnotu třeba sudý/lichý, což značně komplikovalo kód a vzrůstala pracnost. Následně byl kaskádovými styly definován vzhled řádků pro třídu sudý a lichý. V CSS3 přichází pseudotřída `:nth-child(n)`. Její hlavní výhodou je značné zjednodušení při práci s opakujícími se prvky na stránce a jejich grafickému oddělení. Díky ní je možné definovat vzhled n -tého potomku nadřazeného elementu. Za (n) lze dosadit číslo, vzorec nebo slovo.

Například pokud $n=2$ a pseudotřída bude definována u tagu `<p>`, bude úprava aplikována na druhý element `<p>` na stránce. Pro $n=\text{odd}$, aplikuje se pro každý sudý element `<p>`.

V ukázce je vytvořena tabulka s daty a pouze pomocí pseudotřídy je dosaženo obarvení všech sudých řádků v tabulce.

```
table tr:nth-child(even) {  
  background-color: #99CCFF;  
}
```

Obrázek 31 - CSS pseudotřída kód



Pořadí	Prohlížeč	Podíl
1.	Chrome	46%
2.	Internet Explorer	21%
3.	Firefox	18%
4.	Safari	10%
5.	Opera	2%

Obrázek 32 - CSS pseudotřída tabulka

Využitím této nové pseudotřídy ve specifikaci CSS3 dochází ke zjednodušení a zrychlení tvorby webu, při výpisu dat z databází, kdy není předem znám rozměr tabulky a znatelné úspore kódu, která se projeví především v obsáhlejších tabulkách a složitějších prezentacích.

S obdobným přínosem existují od nové specifikace další pseudotřídy, fungující na podobném principu a lišící se způsobem vyhodnocování ovlivnitelných prvků. `:nth-last-child(n)` funguje stejně, jako dříve zmíněný `:nth-child(n)`, jen počítá výskyty od konce. Dále třeba `:last-child` a `:first-child` upravující poslední a první potomek elementu.

4.8.3 Zakulacené rohy

Struktura webové stránky je složena z jednotlivých bloků obdélníkového tvaru. Tyto obdélníky nebo čtverce jsou na sobě poskládány dle určitého návrhu. Pokud by se zobrazily hrany, vykreslily by se ostré rohy. Pro přívětivější design se webdesignéři snaží tyto ostré hrany zjemnit určitým přechodem, například zakulacením hran. Starší specifikace CSS neznají nic jako zakulacené hrany, proto se využívají obrázky na pozadí, vytvořené v grafickém editoru, jejichž správným poskládáním na webové stránce vznikne vzhled právě se zakulacenými rohy. Nevýhodou tohoto postupu je určitá časová náročnost na vytvoření obrázku, následně správné pozicování a v neposlední řadě také větší množství přenesených dat, potřebných k zobrazení stránky.

Výhodou CSS3 je implementovaná podpora zakulacených okrajů, která eliminuje výše zmíněné nevýhody. Odpadá práce v externím grafickém editoru, web se snáze přizpůsobí různému obsahu a není potřeba přenášet objemnější obrázkové soubory pro vykreslení.

Pro tuto funkci se využívá vlastnost `border-radius`, jako hodnotu lze standardně použít poloměr zakulacení pro všechny čtyři rohy. Lze také určit pro každý roh zvlášť, kdy zapíšeme hodnoty v pořadí levý horní, pravý horní, pravý spodní, levý spodní.

Ukázka, jak jednoduše lze vytvořit díky CSS3 zakulacené okraje. Je zde opět vidět nová vlastnost `border-radius` díky zastaralému zvýrazňovači programu PSPad 4.5.2.

```
div {
  border: 2px solid;
  border-radius: 10px;
  background-color: #99CCFF;
  width: 400px;
  /* ukázka border-radius pro každý roh zvlášť: */
  /* border-radius: 10px 10px 20px 20px */
}
```

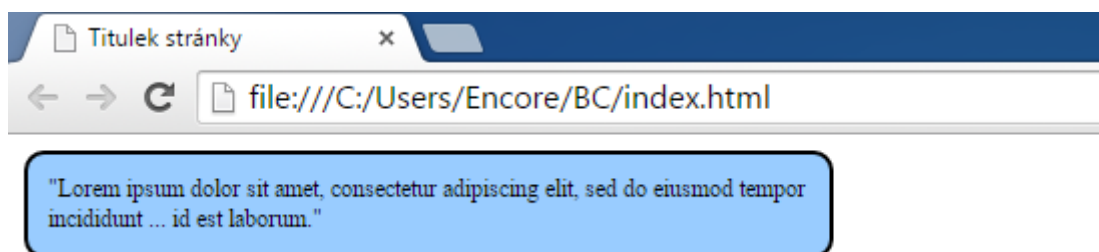
Obrázek 33 - CSS zakulacené rohy

V HTML5 není potřeba nic definovat, pouze základní strukturu a obsah, který je třeba zobrazit.

```
<div>
  <p>"Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed
  ↪ eiusmod tempor incididunt ... id est laborum."</p>
</div>
```

Obrázek 34 - CSS zakulacené rohy HTML

Výsledek se zakulacenými rohy může vypadat například jako na následujícím obrázku.



Obrázek 35 - CSS ukázka - zakulacené rohy

4.8.4 Stínování

Jednou z novinek CSS3 je stínování, které lze aplikovat na text nebo na některé blokové prvky. Stejně jako v minulém případě, i zde bylo dříve potřeba používat grafické editory a vkládání obrázků za účelem stínování na stránce.

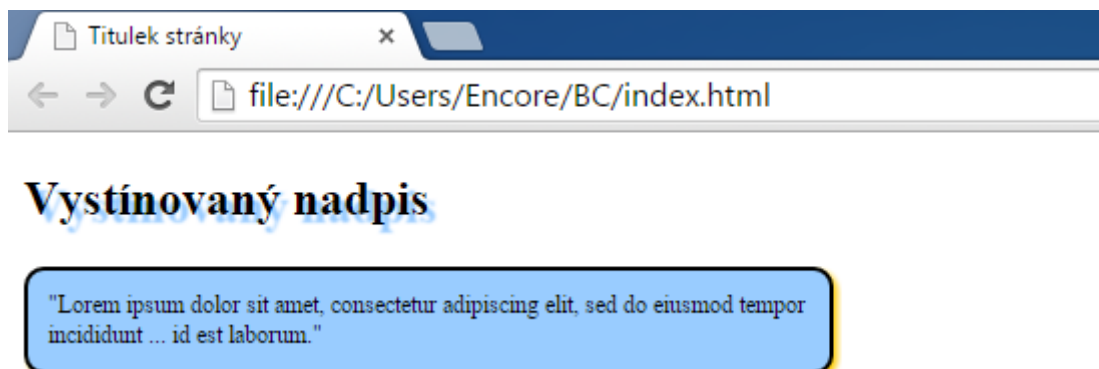
Součástí CSS3 jsou vlastnosti `text-shadow` a `box-shadow`, které odstraňují tento handicap starší specifikace. Syntaxe je v obou případech stejná, zadávají se následující hodnoty: posun stínu na ose X, posun stínu na ose Y, mohutnost stínu a jeho barva.

```
div {
  box-shadow: 2px 2px 4px #FFCC00;
}

h2 {
  text-shadow: 4px 4px 1px #99CCFF;
}
```

Obrázek 36 - CSS stínování kód

Ve stínování se odehrál velký posun oproti CSS2.1, kde byla nulová podpora. Protože se jedná o efekt dodávající určitou prostorovost stránky a lepší čitelnost při správném nastavení, usnadní mnoho práce při tvorbě designu. Následující ukázka shrnuje vystínování nadpisu i blokového prvku.



Obrázek 37 - CSS ukázka stínování

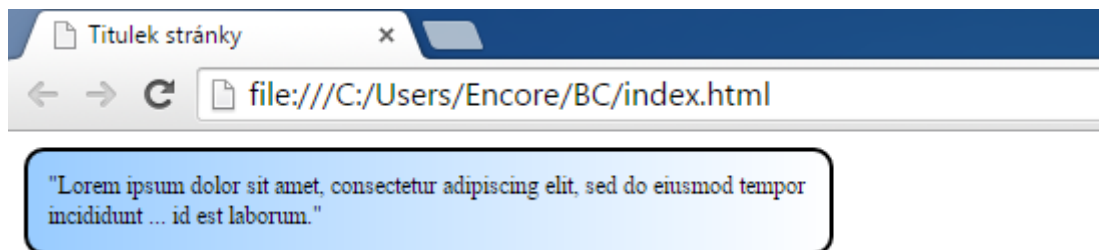
4.8.5 Přechody

Nejen zakulacené rohy a stínování vylepšuje vzhled stránky. Mezi tyto prvky patří také barevné přechody, které oživují pozadí jednotlivých částí nebo celých stránek. Ve starší specifikaci kaskádových stylů neexistovala podpora barevných přechodů, proto byly do CSS3 implementovány za účelem zjednodušení práce a odlehčení datovému provozu. Dříve totiž byla tato funkce nahrazována obrázky vytvořenými v jiných programech, což mělo za následek stahování větších objemů dat a delší načítání stránek.

Zápis přechodů se provádí pomocí hodnoty vlastnosti `background`. Hodnota má následující formu: `linear-gradient(směr přechodu, začínající barva, končící barva);`. Na obrázcích je zobrazen konkrétní příklad na prvním řádku a ukázka výsledku:

```
div {
  background: linear-gradient(to right, #99CCFF, #FFFFFF);
  border: 2px solid;
  border-radius: 10px;
  width: 400px;
}
```

Obrázek 38 - CSS přechody



Obrázek 39 - CSS ukázka přechodů

Přechod na pozadí lze nastavit nejen blokovým prvkům, jak je vidět v ukázce, ale je možné jej nastavit i pro celou stránku a dosáhnout tak žádaného výsledku.

Nevýhodou byla dříve horší podpora a nejednotnost v prohlížečích, tyto nedostatky však byly postupně odstraňovány a dnes jsou přechody podporovány všemi majoritními jádry.

5 Zhodnocení výsledků

Vývoj webových stránek jde neustále kupředu s tím, jak stoupají nároky uživatelů, kodérů i webdesignerů. Ve chvíli, kdy přestávají stačit základní schopnosti značkovacího jazyka, začnou uživatelé i vývojáři hledat jiné způsoby, jak se dostat k požadovaným funkcím. To je určité znamení pro skupinu lidí, starající se o standardy a specifikace, aby tyto funkce zvážili a případně se postarali o jejich nativní podporu v příštích verzích jazyka. Díky tomuto principu jsou k dispozici aktuálně používané jazyky HTML5 a CSS3 pro tvorbu webu, značně zjednodušující některé složité postupy ze starších specifikací HTML 4.01 a CSS 2.1. Zároveň přináší nové funkce nejenom pro koncové návštěvníky stránek, ale i pro snazší tvorbu webu a ulehčení práce.

Určitou nevýhodou a do jisté míry zábranou v nasazení nových specifikací včetně HTML5 a CSS3 bývá horší podpora ze strany výrobců webových prohlížečů. Tento problémem se nevyhnul předchozím verzím, ani té současné. V tuto chvíli však nový standard podporují všechny současně vyvíjené prohlížeče.

Provedením praktického porovnání nového kódu se starší verzí HTML bylo dosaženo výsledku, ukazujícím na významné zjednodušení mnohých značek, které pomáhají přehlednějšímu kódu a snazšímu zapamatování jejich zápisů. To do jisté míry může vést k rychlejší a finančně méně náročné tvorbě webu.

V HTML5 se více klade důraz na významový smysl značek, bylo přidáno několik sémantických elementů, které v dřívějších verzích neexistovaly. Zároveň došlo k dosazení sémantického významu pro dříve existující značky, které jej do této doby neměly. Ostatně oživení původních značek, které jsou v podvědomí kodérů a mají existující podporu napříč staršími i novými prohlížeči, je lepším krokem, než zakládat nové.

Přínosem bylo také přidání podpory multimédií v podobě zvukového a videopřehrávače, díky které není nutné spoléhat se na externí programy a zásuvné moduly do prohlížečů, které občas obsahovaly bezpečnostní chyby, kvůli kterým je bylo nutné zvlášť aktualizovat. Nativní podpora HTML5 přehrávače v prohlížeči tato omezení a nepohodlí nemá.

Značné usnadnění práce přinesly také kaskádové styly ve verzi 3, jejichž nové vlastnosti nahrazují složitější postupy, používané ve starším CSS 2.1. Především z větší části eliminují nutnost používat datově náročné obrázky pro tvorbu vzhledu webové stránky, kdy lze některé prvky nechat vykreslit prohlížeč.

6 Závěr

Cílem teoretické části práce bylo uvedení pojmů spojených s fungováním internetových stránek a technologiemi, díky kterým může web fungovat. Dále byl součástí stručný přehled používaných webových prohlížečů a následně historie předmětného značkovacího jazyka. Součástí teoretické části byl také základ jazyka HTML, jeho struktura a princip.

V úvodní části byl definován pojem internet jako celosvětová síť navzájem propojených zařízení, a jejíž součástí je i služba World Wide Web, která se využívá pro webové stránky. Následoval princip fungování protokolu HTTP a popis jeho stavových kódů, se kterými se uživatelé mohou setkat při procházení internetových stránek. Dále byl vysvětlen rozdíl mezi programovacím a značkovacím jazykem, neboť často nejsou tyto dva rozdílné pojmy rozlišovány. Další část byla věnována webovým prohlížečům od nejstaršího prvního prohlížeče až po dnešní moderní a rozšířené aplikace. Následně došlo na popis jazyka HTML a jeho historii. Stejně tak v případě kaskádových stylů, které byly popsány spolu s historií a způsobem použití ve spolupráci s HTML. Na konci teoretické části byl uveden základ k tvorbě webových stránek.

Cílem praktické části práce byla analýza nových funkcí a možností jazyků HTML5, CSS3 a následné porovnání s jejich staršími specifikacemi HTML 4.01 a CSS 2.1, vyzdvižení výhod nové verze a přínos pro tvůrce a uživatele webových stránek.

Začátek praktické části obsahoval změny v základní struktuře webu, kde došlo k značnému zjednodušení a zpřehlednění kódu oproti starším verzím. Následně byly popsány nové značky, které v HTML 4.01 neexistovaly, shrnuty jejich funkce, způsoby zápisu a porovnání s dřívějším řešením. Další kapitolou byly původní značky, které existovaly v dřívějších specifikacích, ale v HTML5 dostaly nový sémantický význam, nebo jim byl význam přidělen poprvé. Protože si tyto značky své funkce zachovaly, popis byl zaměřen především na sémantický význam. Dalším bodem byly vyřazené značky ze specifikace, které by se nadále neměly používat, neboť byly nahrazeny jiným způsobem provedení. Na závěr HTML5 byly ponechány

značky pro multimediální obsah na stránce, protože se jednalo o nové a důležité prvky na stránce. Poslední částí praktické práce byl popis kaskádových stylů a některých jejich nových funkcí, zaměřených především na úsporu datového provozu a zjednodušení tvorby grafického designu stránky. Kromě představení novinek bylo i popsáno, jak se funkce řešily dříve v rámci původních specifikací.

Praktické ukázky byly prováděny formou porovnání zdrojového kódu, ukázkou výsledku v prohlížeči, případně slovním popisem. Na základě zmíněných porovnání elementů a postupů bylo dosaženo závěru, že nová specifikace přináší velké množství změn k lepšímu v podobě přehlednějšího a snadnějšího vypracování, znatelnou úsporu kódu, má prostor pro snížení náročnosti na přenesená data a díky implementaci multimediálního přehrávače i lepší podporu u koncových návštěvníků.

Jistou nevýhodou mohou být problémy se zobrazením ve starších a neaktualizovaných prohlížečích, nicméně neaktualizované aplikace přinášejí větší rizika v podobě bezpečnostních problémů, než nesprávném zobrazení stránky.

Seznam obrázků

Obrázek 1 - 5 nejpoužívanějších prohlížečů	17
Obrázek 2 - Styly (přímý zápis)	23
Obrázek 3 - Styly (zápis do hlavičky).....	24
Obrázek 4 - Styly (externí soubor - propojení)	24
Obrázek 5 - Základní struktura	25
Obrázek 6 - Základní stránka	25
Obrázek 7 - Charset.....	26
Obrázek 8 - Charset porovnání	26
Obrázek 9 - Link stylesheet	27
Obrázek 10 - Meta name	27
Obrázek 11 - Kompletní hlavička	27
Obrázek 12 - DOCTYPE 4.01	28
Obrázek 13 - DOCTYPE 5	29
Obrázek 14 - Charset HTML 4.01	29
Obrázek 15 - Charset HTML 5	29
Obrázek 16 - Sekce HTML 4.01	30
Obrázek 17 - Sekce HTML 5	31
Obrázek 18 - Figure	33
Obrázek 19 - Progressbar – ukázka.....	33
Obrázek 20 - Progressbar – kód.....	34
Obrázek 21 - Mark - kód.....	35
Obrázek 22 - Mark – zobrazení.....	35
Obrázek 23 - Time	35
Obrázek 24 - Audio zjednodušené.....	40

Obrázek 25 - Audio více formátů	40
Obrázek 26 - Audio přehrávač	41
Obrázek 27 - Video zjednodušené	42
Obrázek 28 - Video více formátů.....	42
Obrázek 29 - CSS struktura	44
Obrázek 30 - CSS pseudotřídy.....	44
Obrázek 31 - CSS pseudotřída kód.....	45
Obrázek 32 - CSS pseudotřída tabulka	45
Obrázek 33 - CSS zakulacené rohy.....	46
Obrázek 34 - CSS zakulacené rohy HTML	47
Obrázek 35 - CSS zakulacené rohy ukázka	47
Obrázek 36 - CSS stínování kód	47
Obrázek 37 - CSS stínování ukázka.....	48
Obrázek 38 - CSS přechody.....	48
Obrázek 39 - CSS přechody ukázka	49

Bibliografie

Berners-Lee, Sir Tim. 2001. Tim Berners-Lee: The WorldWideWeb browser. *World Wide Web Consortium (W3C)*. [Online] 2001. [Citace: 6. leden 2015.] <http://www.w3.org/People/Berners-Lee/WorldWideWeb.html>.

Berners-Lee, Tim a Daniel, Connolly. 1995. Hypertext Markup Language 2.0. *World Wide Web Consortium*. [Online] 22. 9 1995. [Citace: 6. leden 2015.] http://www.w3.org/MarkUp/html-spec/html-spec_toc.html.

Berners-Lee, Tim, Fielding, Roy a Nielsen, Henrik Frystyk. 1996. Hypertext Transfer Protocol -- HTTP/1.0. *World Wide Web Consortium (W3C)*. [Online] 1996. [Citace: 6. leden 2015.] <http://www.w3.org/Protocols/HTTP/1.0/spec.html>.

Bos, Gijsbert a Wium, Lie Håkon. 1996. Cascading Style Sheets, level 1. *World Wide Web Consortium (W3C)*. [Online] 17. 12 1996. [Citace: 6. leden 2015.] <http://www.w3.org/TR/CSS1/>.

Castro, Elizabeth a Hyslop, Bruce. 2012. *HTML5 a CSS3 - Národní průvodce tvorbou WWW stránek*. Brno: Computer Press, 2012. ISBN 9788025137338.

Clevinger. 2006. *Programování*. [Online] 2006. [Citace: 13. leden 2015.] <http://www.k-prog.wz.cz/progjaz/index.php>.

Consortium, W3. 1999. HTML 4 Changes. *World Wide Web Consortium*. [Online] 1999. [Citace: 13. leden 2015.] <http://www.w3.org/TR/REC-html40/appendix/changes.html>.

Consortium, World Wide Web. 2014. HTML5 is a W3C Recommendation. *World Wide Web Consortium (W3C)*. [Online] 2014. [Citace: 14. leden 2015.] <http://www.w3.org/blog/news/archives/4167>.

Cullen, Aodhan. 2012. Independent web analytics firm StatCounter confirms milestone as Chrome overtakes IE globally for first calendar month. *StatCounter Global Stats*. [Online] 1. červen 2012. [Citace: 14. leden 2015.] <http://gs.statcounter.com/press/chrome-overtakes-ie-globally-monthly>.

Dilger, Daniel Eran. 2012. Apple apparently kills Windows PC support in Safari 6.0. *Apple Insider*. [Online] 25. červenec 2012. [Citace: 14. leden 2015.] http://appleinsider.com/articles/12/07/25/apple_kills_windows_pc_support_in_safari_60.

Goldstein, Alexis , Lazaris, Louis a Weyl, Estelle. 2011. *HTML5 & CSS3 For The Real World*. místo neznámé : SitePoint, 2011. ISBN 978-0980846904.

Hlavenka, Jiří. 2002. *Vytváříme WWW stránky*. Praha : Computer Press, 2002. ISBN 80-7226-748-5.

Hogan, Brian. 2012. *HTML5 a CSS3 - Výukový kurz webového vývoje*. Brno : Computer Press, 2012. ISBN 9788025135761.

Kubíček, Michal. 2008. *Velký průvodce SEO*. Brno : Computer Press, 2008. ISBN 978-80-251-2195-5.

Landofcode.com. 2014. History of HTML. *Landofcode.com*. [Online] 2014. [Citace: 14. leden 2015.] <http://www.landofcode.com/html-tutorials/html-history.php>.

Lubbers, Peter, Albers, Brian a Salim, Frank. 2011. *HTML5 - Programujeme moderní webové aplikace*. Brno : Computer Press, 2011. ISBN 9788025135396.

Meyer, Eric. 2006. *CSS: The Definitive Guide, 3rd Edition*. místo neznámé : O'Reilly Media, 2006. ISBN 978-0-596-52733-4.

Microsoft. 2003. Information about Internet Explorer versions. *Microsoft Knowledge Base*. [Online] 2003. [Citace: 13. leden 2015.] <http://support.microsoft.com/kb/969393>.

Mozilla.org. 2013. History of the Mozilla Project. *Mozilla*. [Online] 2013. [Citace: 14. leden 2015.] <https://www.mozilla.org/en-US/about/history/details/>.

Opera Software ASA. 2007. About Opera. *Opera Software*. [Online] 2007. [Citace: 12. leden 2015.] <http://web.archive.org/web/20071203140601/http://www.opera.com/company/about>.

Opera Software ASA. 2005. Feel Free: Opera Eliminates Ad Banner and Licensing Fee. *Opera Software*. [Online] 20. září 2005. [Citace: 12. leden 2015.] <http://www.operasoftware.com/press/releases/desktop/feel-free-opera-eliminates-ad-banner-and-licensing-fee>.

Raggett, Dave, a další. 1998. *Raggett on HTML 4*. Harlow : Addison Wesley Longman, 1998. ISBN 0-201-17805-2.

StatCounter. 2015. StatCounter Global Stats - Browser, OS, Search Engine including Mobile Usage Share. *StatCounter Global Stats* . [Online] 2015 [Citace: 12. leden 2015.]. <http://gs.statcounter.com>

Tomášek, Jan. 2011. Živě.cz. *Jak si vedou alternativní prohlížeče?* [Online] 7. červenec 2011. [Citace: 12. leden 2015.] <http://blog.blog.zive.cz/2011/01/jak-si-vedou-alternativni-prohlizece/>.

TyfloCentrum Brno, o. p. s. 2015. Blind Friendly Web - přístupnost webových stránek. *Blind Friendly Web*. [Online] 2015. [Citace: 12. leden 2015.] <http://blindfriendly.cz/>.

USPT. 2004. Trademark Status and Document Retrieval. *United States Patent and Trademark Office*. [Online] 9. listopad 2004. [Citace: 14. leden 2015.] http://tsdr.uspto.gov/#caseNumber=78344043&caseType=SERIAL_NO&searchType=statusSearch.

W3C Consortium. About W3C. *World Wide Web Consortium (W3C)*. [Online] [Citace: 12. leden 2015.] <http://www.w3.org/Consortium/>.