

Česká zemědělská univerzita v Praze

Provozně ekonomická fakulta

Katedra informačních technologií



Bakalářská práce

Webová prezentace v HTML5 a CSS3

Lukáš Modrý

© 2014 ČZU v Praze

ČESKÁ ZEMĚDĚLSKÁ UNIVERZITA V PRAZE

Katedra informačních technologií

Provozně ekonomická fakulta

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Modrý Lukáš

Informatika

Název práce

Webová prezentace v HTML5 a CSS3

Anglický název

Website in HTML5 and CSS3

Cíle práce

Bakalářská práce je tematicky zaměřena na problematiku tvorby webových prezentací pomocí HTML5 a CSS3.

Hlavním cílem práce je vytvoření webové prezentace pomocí HTML5 a CSS3 a porovnání s možností vytvoření obdobné prezentace v XHTML 1.1, CSS2.1 a JavaScriptu.

Dílní cíle práce jsou:

- vypracování přehledu použitelných značek HTML5 a
- vypracování přehledu použitelných vlastností CSS3 pro současné webové prohlížeče.

Metodika

Metodika řešené problematiky bakalářské práce je založena na studiu a analýze odborných informačních zdrojů.

Vlastní práce spočívá ve vytvoření dvou podobných webových prezentací pomocí různých technologií. V prvním případě bude použit značkovací jazyk HTML5 a CSS3, ve druhém případě budou využity technologie XHTML 1.1, CSS2 a JavaScript. Na základě syntézy teoretických poznatků a výsledků praktické části budou formulovány závěry bakalářské práce.

Harmonogram zpracování

1. Studium odborných informačních zdrojů, stanovení dílčích cílů a postupu řešení bakalářské práce: 06/2013
2. Zpracování teoretických východisek práce: 07/2013 – 09/2013
3. Vypracování vlastního řešení, diskuze a zhodnocení výsledků: 10/2013 – 02/2014
4. Tvorba finálního dokumentu bakalářské práce: 03/2014
5. Odevzdání bakalářské práce a tezí: 03/2014

Rozsah textové části

30 - 40 stran textu.

Klíčová slova

HTML5, CSS3, JavaScript, Flash, webový prohlížeč

Doporučené zdroje informací

CASTRO, E. - HYSLOP, B. HTML5 a CSS3: Názorný průvodce tvorbou WWW stránek. Brno, Computer Press, 2012. 440 s. ISBN 978-80-251-3733-8.

MURPHY, C. - CLARK, R., STUDHOLME, O. - MANIAN, D. Beginning HTML5 and CSS3: The Web Evolved. Apress, 2012. 600 s. ISBN 978-1-4302-2874-5.

HOGAN, B. P. HTML5 a CSS3: Výukový kurz webového vývoje. Brno, Computer Press, 2012. 272 s. ISBN 978-80-251-3576-1.

LUBBERS, P. - ALBERS, B. - SALIM, F. HTML 5: Programujeme moderní webové aplikace. Brno, Computer Press, 2011. 304 s. ISBN 978-80-251-3539-6.

MARK PILGRIM, D. Into HTML5. [online]. MMIX–MMXI, 2013, URL: < <http://diveintohtml5.info> >.

Diskusní fórum Jak Psát Web. [online]. Dušan Janovský, 2013. URL: < <http://diskuse.jakpsatweb.cz> >.

W3C. Differences HTML4 and HTML5. [online]. Simon Pieters, Anne van Kesteren. Update: 16 May 2013. URL: < <https://raw.githubusercontent.com/whatwg/html-differences/master/Overview.html> >.

W3 School. 1999-2013, Refsnes Data. URL: < <http://www.w3schools.com> >.

Vedoucí práce

Šimek Pavel, Ing., Ph.D.

Termín odevzdání

březen 2014

doc. Ing. Zdeněk Havlíček, CSc.

Vedoucí katedry



prof. Ing. Jan Hron, DrSc., dr. h. c.

Děkan fakulty

V Praze dne 26.11.2013

Prohlašuji, že jsem bakalářskou práci na téma „*Webová prezentace v HTML5 a CSS3*“ vypracoval pod vedením vedoucího bakalářské práce samostatně za použití v práci uvedených pramenů a literatury.

Prohlašuji, že odevzdaná elektronická verze bakalářské práce je identická s její tištěnou podobou.

Praha, 17. 3. 2014

.....

vlastnoruční podpis

Touto cestou bych rád vyjádřil velké díky panu Ing. Pavlu Šimkovi, Ph.D. za jeho odborné rady, trpělivost a ochotu při vedení bakalářské práce. Dále bych chtěl poděkovat rodině a blízkým za podporu při psaní bakalářské práce.

.....

vlastnoruční podpis

Webová prezentace v HTML5 a CSS3

Website in HTML5 and CSS3

Resumé

Bakalářská práce pojednává o nových možnostech jazyků HTML5 a CSS3 a jejich porovnání se staršími verzemi.

Teoretická část se zabývá analýzou odborných zdrojů týkajících se tvorby webových prezentací. Dále jsou v této části popsány přínosy jazyků HTML5 a CSS3.

Součástí praktické části práce je tvorba dvou podobných webových prezentací, kde první je psána za pomoci jazyků xHTML1.1 a CSS2.1. Druhá je napsána v nových specifikacích HTML5 a CSS3. Následně jsou porovnány rozdíly mezi jednotlivými možnostmi tvorby webových stránek. Součástí praktické části je vytvoření seznamu HTML značek a CSS3 vlastností.

Klíčová slova

HTML5, CSS3, JavaScript, Flash, Webový prohlížeč

Abstract

The bachelor thesis deals with the new possibilities of both languages HTML5 and CSS3 and the comparison with their former versions.

The theoretical part covers the analysis of sources interested in the website design of presentations. Furthermore, both HTML5 and CSS3 are described in detail, including their assets and benefits.

As regards to the practical part of the thesis, there is a web design of two similar presentations. The former is created by means of the languages xHTML1.1. and CSS2.1. The latter is the result of the new specifications of HTML5 and CSS3. Creation the list of the HTML markups and the CSS3 properties are included in the practical part, too.

Keywords

HTML5, CSS3, JavaScript, Flash, Web browser

Obsah

1	Úvod.....	8
2	Cíl a metodika práce	9
2.1	Cíl práce.....	9
2.2	Metodika práce	9
3	Teoretická východiska	11
3.1	World Wide Web.....	11
3.2	Hyper Text Transfer Protokol.....	11
3.2.1	Chybové zprávy protokolu HTTP.....	11
3.3	Hyper Text Mark-Up Language	12
3.3.1	Historie HTML	12
3.3.2	Základní struktura HTML stránky.....	14
3.3.3	Problémy HTML	16
3.3.4	Přínosy HTML5	17
3.4	Kaskádové styly	30
3.4.1	Historie CSS	31
3.4.2	Struktura CSS.....	32
3.4.3	Přínosy CSS 3	32
4	Praktická část práce	36
4.1	Změna v zápisu DOCTYPE.....	37
4.2	Využití nových strukturálních značek.....	37
4.3	Doplnění sémantiky jazyka a přístupnost webu v HTML5	40
4.4	Přechody	41
4.5	Zakulacené rohy	43
4.6	Průhlednost.....	44

4.7	Pseudotřídy	45
4.8	Mnohonásobné pozadí	48
4.9	Formátování textu.....	51
4.9.1	Font-face	51
4.9.2	Stínování textu	52
4.9.3	Transformování textu	53
4.9.4	Více sloupcový layout	53
4.10	Dynamické efekty.....	54
4.11	Využití formulářových prvků	55
4.12	Grafika za pomoci elementu Canvas.....	61
4.13	Použití multimédií	63
5	Výsledky a jejich zhodnocení.....	65
5.1	Analýza webů vytvořených v praktické části	65
5.2	Přínosy realizace stránek pomocí HTML5 a CSS3	66
5.3	Nevýhody realizace stránek v HTML5 a CSS3	67
5.4	Podpora v nových verzích prohlížečů.....	68
6	Závěr	69
	Seznam zkratk	70
	Seznam použitých zdrojů	71
	Seznam obrázků	74
	Seznam tabulek.....	74
	Přílohy	75
	Webové prezentace	75
	Přehled HTML značek.....	75
	Přehled CSS vlastností	82

1 Úvod

Webové prezentace jsou v dnešní přetechnizované době nejrozšířenějšími a nejpoužívanějšími zdroji informací a prostorem pro komunikaci mezi lidmi. Většina moderní společnosti se s webovými stránkami setkává každý den díky internetu, kde jsou k dispozici prostřednictvím služby WWW. Mnoho lidí si svět bez webu nedovede představit. Pro některé je to prostor zábavy, pro jiné způsob jejich obživy. Je to také nejrozsáhlejší zpravodajská platforma dnešní doby. Nespornou součástí internetu je i všudypřítomná reklama, která je mnohdy nežádoucí. V posledních letech se rovněž rozmáhá množství internetového obchodu prostřednictvím e-shopů.

Webové stránky jsou postaveny na značkovacím jazyce HTML, který slouží jako kostra celého webu a určuje základní strukturu a organizaci jednotlivých prvků v dokumentu. Pomocí tohoto jazyka se vytváří celý obsah každé webové stránky. Bez značkovacího jazyka HTML webovou prezentaci nelze vytvořit.

Další nepostradatelnou částí většiny prezentací jsou kaskádové styly, které doplňují prostou strukturu obsahu stránky uživatelsky přívětivým prostředím a dávají stránce přehlednost. Díky stylům nejsou dokumenty na internetu pouhým textem. Oddělují obsah dokumentu od jeho vzhledu.

Největším problémem při tvorbě webových prezentací je podpora jednotlivých prohlížečů a jejich způsob konkrétního zobrazení zápisu jednotlivých elementů a atributů. S tímto problémem se aktuálně weboví vývojáři nejvíce setkávají u specifikace HTML5 a CSS3. Tyto návrhy stále ještě nejsou ve své finální podobě a jejich podpora v jednotlivých webových prohlížečích stále není jednotná.

To má za následek komplikace se zobrazením jednotlivých možností těchto specifikací, které budou popsány v následujících kapitolách této práce.

Bakalářská práce je tematicky zaměřena na problematiku tvorby stránek pomocí HTML5 a CSS3 a jejich porovnání se starými specifikacemi.

2 Cíl a metodika práce

2.1 Cíl práce

Bakalářská práce je tematicky zaměřena na problematiku tvorby webových prezentací pomocí HTML5 a CSS3. Hlavním cílem práce je vytvoření webové prezentace pomocí HTML5 a CSS3 a porovnání s možností vytvoření obdobné prezentace v xHTML1.1, CSS2.1 a JavaScriptu.

Díličí cíle práce jsou:

- vypracování přehledu použitelných značek HTML5,
- vypracování přehledu použitelných vlastností CSS3 pro současné webové prohlížeče.

2.2 Metodika práce

Metodika řešené problematiky bakalářské práce je založena na studiu a analýze odborných informačních zdrojů.

Vlastní práce spočívá ve vytvoření dvou podobných webových prezentací pomocí různých technologií. V prvním případě bude použit značkovací jazyk HTML5 a CSS3, ve druhém případě budou využity technologie xHTML1.1, CSS2.1 a JavaScript. Na základě syntézy teoretických poznatků a výsledků praktické části budou formulovány závěry bakalářské práce.

Práce se dělí do 3 základních kapitol:

- 3. Teoretická východiska práce
- 4. Praktická část
- 5. Zhodnocení výsledků a doporučení

Kapitola „*Teoretická východiska práce*“ se zaměřuje na stručnou historii vývoje jazyka HTML a CSS. Dále na představení nových elementů a jejich možností použití, které přináší HTML5 a s ním i možnost grafického zpracování prostřednictvím kaskádových stylů.

„*Praktická část*“ práce je zaměřena na tvorbu dvou podobných webových prezentací, kde první bude vytvořena podle zcela podporovaných a zažitých standardů xHTML1.1 a CSS2.1. Následně podle ní bude vytvořen další web prostřednictvím

nových technologií a možností jazyků HTML5 a CSS3. Na základě těchto webů bude provedena analýza podpory jednotlivých nových možností nových specifikací a zhodnocení jejich využití v praxi. Samotné porovnání webů bude provedeno testováním zobrazení a funkce jednotlivých prvků v pěti různých a zároveň nejpoužívanějších prohlížečích internetu. Těmi jsou Internet Explorer 11, Opera 19, Mozilla Firefox 27, Apple Safari 5, Google Chrome 32. Dále bude provedena analýza zobrazování stránek jako celku.

Dílním cílem práce je vypracování seznamu použitelných značek HTML5 a dále přehledu použitelných vlastností CSS3 v současné době.

V kapitole „*Výsledky a jejich zhodnocení*“ bude vyhodnocena praktická část práce a zvážení použitelnosti specifikace HTML5 a CSS3 v praxi. Dále zhodnocení přínosů a nedostatků nových jazyků.

3 Teoretická východiska

3.1 World Wide Web

WWW je celosvětová síť mezi sebou propojených hypertextových dokumentů, které jsou k dispozici prostřednictvím internetové sítě. Fungování WWW zajišťuje protokol HTTP.

Autorem je Angličan Timothy Berners-Lee¹, který stojí rovněž za vývojem hypertextových dokumentů. Historie webu samotného se datuje do roku 1989. (Connolly, 2011)

Cílem jeho koncepce bylo umožnit uživatelům prohlížení dokumentů po částech, proto nebylo nutné načíst celý dokument, ale jen požadovanou část. Jednotlivé dokumenty na internetu jsou uloženy v mezi sebou propojených počítačích a uživatel mezi nimi může číst prostřednictvím hypertextových odkazů. (Longman, 1998)

3.2 Hyper Text Transfer Protokol

Protokol HTTP je základním kamenem pro komunikaci prostřednictvím WWW a obsahuje veškeré standardy pro správnou komunikaci mezi jednotlivými dokumenty.

Protokol funguje na principu dotaz / odpověď. Uživatel zadává serveru dotaz prostřednictvím klienta po nějakém dokumentu, k čemuž využívá jeho URL adresu. Server následně dotazu buď vyhoví, nebo vrátí chybovou zprávu. Protokol určuje komunikaci mezi klientem a serverem. (Janovský, nedatováno)

3.2.1 Chybové zprávy protokolu HTTP

Protokol HTTP obsahuje seznam stavových zpráv, díky kterým uživatelé informuje o nastalém problému a stavu zpracování jejich dotazu. Tento seznam byl představen na konsorciu W3 v roce 1992 a je součástí protokolu už od verze 1.0. (Cooper, 2010)

¹Timothy Berners-Lee (* 8. června 1955, Londýn), zdroj www.w3.org

Stavové kódy se dělí do 5 skupin:

- 1xx – Informační, zprávy zasílané konkrétní aplikací
- 2xx – Informace o úspěšném zpracování požadavku
- 3xx – Přesměrování vyžadující interakci uživatele
- 4xx – Chyba na straně klienta, která vzniká například špatně zadanou URL adresou nebo snahou o zobrazení už neexistujícího dokumentu
- 5xx – Chyba na straně serveru. Tyto chyby uživatel nemůže nijak ovlivnit a musí počkat na napravení chyby na straně serveru (Google, 2014)

3.3 Hyper Text Mark-Up Language

HTML je značkovací jazyk určený pro přesné a jednoznačné strukturování dokumentů na internetu, které jsou publikovány prostřednictvím WWW a protokolu HTTP. Jazyk je základním stavebním kamenem všech internetových prezentací.

3.3.1 Historie HTML

Poprvé jazyk ve verzi HTML 1.0 definoval v roce 1991 Timothy Barners-Lee v laboratoři v CERNu, která se nachází na švýcarsko-francouzských hranicích. (Foundation, 2014)

Jeho snahou bylo umožnit sdílení dat mezi jednotlivými vědci z celého světa a poskytnout možnost odkazování se z jedné odborné publikace na jinou. Klíčové bylo umožnit jednoduché sdílení informací mezi velkým množstvím uživatelů po celém světě.

První verze specifikace byla popsána v dokumentu *HTML Tags*. Umožňovala základní členění dokumentů do jednotlivých logických celků, možnosti pro zvýrazňování textů a propojení jednotlivých dokumentů pomocí hypertextových odkazů.

V roce 1993 se objevily první internetové prohlížeče podporující grafické rozhraní a zároveň byla vydána i nová specifikace jazyka HTML ve verzi 2.0. Oficiální uvedení se konalo až v roce 1995, kdy byl současně představen její nástupce, návrh HTML 3.0.

V roce 1997 vznikl projekt *Cougar*, jehož předmětem bylo doplnění stávající specifikace o některé prvky (například rámy), které do té doby nebyly součástí

standardu, ale mezi uživateli a vývojáři se velmi rozšířily. To mělo v polovině roku 1997 za následek uveřejnění verze HTML 4.0, která se koncem roku stala standardem. (Longman, 1998)

V roce 1999 se objevuje nový standard s označením 4.01, který opravuje některé zásadní chyby v předchozí verzi.

Na přelomu tisíciletí se objevuje nová specifikace XHTML1.0, která je postavena na značkovacím jazyce XML. Hlavní rozdíl od původní specifikace je ve striktnosti validace kódu a nutnosti uzavírání nepárových značek. Následoval vývoj XHTML2.0, který však byl pozastaven z důvodu nezájmu distributorů webových prohlížečů o tuto specifikaci. (Kosek, 2013)

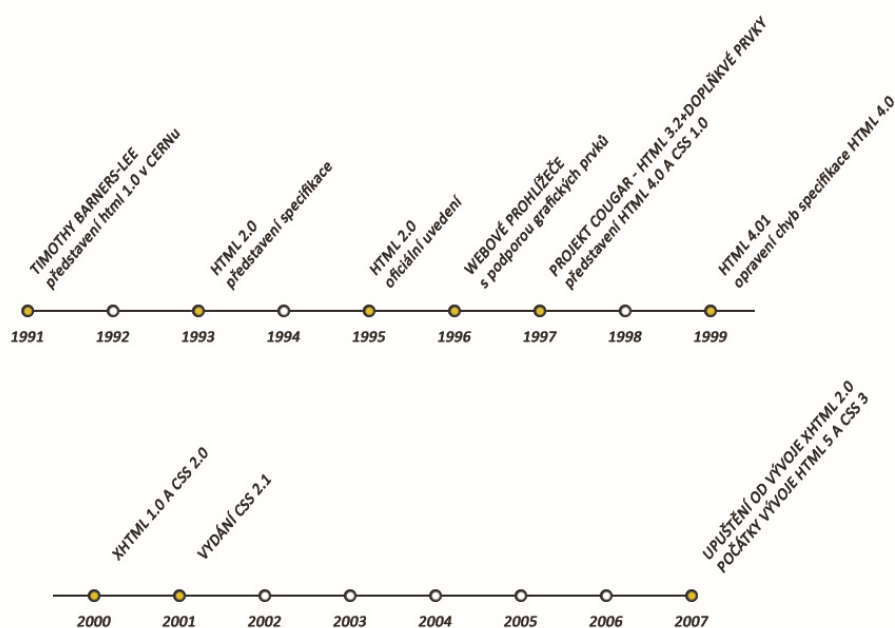
V roce 2007 začíná nezávislá skupina WHATWG² s vývojem nové specifikace HTML5. Konsorcium W3 tuto specifikaci následně přebírá a zapojuje výsledky do oficiální verze. (Castro, a další, 2012)

HTML5 vychází ze svého předchůdce, verze 4.01. Cílem nové verze je doplnění jazyka o nové funkce, které do té doby byly nahrazovány jinými, složitějšími způsoby. Dalším cílem je sémantika jazyka, která v předchozí verzi nebyla zcela jednoznačná. (Hickson, 2014)

Jazyk HTML5 obohacuje původní specifikaci o 32 nových značek, jiné naopak přestává podporovat. Dále doplňuje kooperaci s programovacím jazykem JavaScript a umožňuje vývojářům vytvářet množství nových aplikací. (Schools, 2014)

Mojmír Greguš o nové specifikaci na webu H1.cz napsal: „HTML5 je velkým technologickým krokem dopředu a dává vývojářům webů nové možnosti, jak naplnit cíle webu.“ (Greguš, 2013)

² www.whatwg.org



Obrázek 1 - Časová osa vývoje HTML a CSS

3.3.2 Základní struktura HTML stránky

Každá HTML stránka má svou pevnou strukturu, kterou je nutné dodržet. Základním prvkem pro psaní zdrojového kódu stránky jsou značky, které se uzavírají do špičatých závorek <značka>.

Ty mohou dále obsahovat jednotlivé atributy, kterým se za pomoci konkrétních hodnot přidávají další vlastnosti. Například identifikátory elementů `id` a `class`, cesta hypertextového odkazu, popisky obrázků, cesta k obrázku, případně přežití in-line styly stránky.

Značky se v HTML dělí do dvou základních skupin:

- Párové, které musí mít svou výchozí a koncovou značku, mezi něž se vkládá obsah daného elementu.

`<p>` Obsah odstavce, který se reálně zobrazí na stránce `</p>`

- Nepárové značky není potřeba ukončovat párovou značkou, což však neplatí v přísnější specifikaci xHTML, kde se ukončují lomítkem „/“.

``

``

Snahou HTML5 je doplnění sémantiky jazyka, která dbá na čitelnost zdrojového kódu a správnou strukturu jednotlivých prvků na stránce. Každá stránka by měla dodržovat základní logickou strukturu stěžejních elementů.

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

Nutnou součástí každé stránky je specifikování typu dokumentu (DTD) umístěné na první řádce zdrojového kódu. Element `<!DOCTYPE>` není součástí standardu HTML. Je to instrukce webovému prohlížeči o tom, jaká specifikace HTML byla pro napsání stránky použita a jak jí má prohlížeč přeložit do grafického výstupu. (Peters, a další, 2014)

Webové stránky jsou rozděleny do dvou základních částí:

- Hlavička stránky
- Tělo stránky

V hlavičce, která je ohraničena značkami `<head></head>` je prostor pro titulek stránky, připojení externích stylů a souborů se skripty, nastavení meta značek pro internetové vyhledávače nebo určení cesty k ikoně stránky. Zde se také deklaruje použití znakové sady, což je nezbytné pro správné čtení dokumentů v jednotlivých prohlížečích. Je možné použít jedné ze tří základních sad znaků – *ISO*, *UTF* a *Windows*. K tomuto účelu slouží zmíněné meta značky.

```
<meta http-equiv="Content-Type" content="text/html;
charset=iso-8859-2 nebo windows-1250 nebo UTF-8">
```

Tělo stránky `<body></body>` je už samotná obsahová část webu, která se zobrazí návštěvníkovi stránky po zadání správné URL adresy.

Každý webový tvůrce by měl dbát na dodržení těchto základních logických prvků stránky pro její správné fungování a vedle toho psát svůj zdrojový kód validní podle specifikace W3C.

Validní zdrojový kód napomáhá jednotnému zobrazení stránky na všech platformách ve veškerých webových prohlížečích. Bohužel 100% shody v praxi nejde téměř dosáhnout, což je zapříčiněno nejednotností vývoje jednotlivých prohlížečů. Rovněž jsou validní stránky přitažlivější pro internetové vyhledávače a díky tomu získávají lepší výsledky v rámci SEO optimalizace. (Kubíček, 2010)

3.3.3 Problémy HTML

Webový vývojáři se setkávají s řadou problémů. Tím největším je při tvorbě webových stránek jejich optimalizace a jednotnost pro širokou škálu webových prohlížečů.

Konkurenční boj mezi jednotlivými vývojáři softwaru pro prohlížení internetu zaviňuje nejednotnost podpory různých vykreslovacích algoritmů, podporu různých elementů na stránce a způsob načítání stránek jako celku.

Dalším problémem při psaní webových stránek je nedodržení validního zdrojového kódu, ať už narušením logické struktury stránek, či sémantiky, popř. pouhými překlepy při psaní.

`<div><p> obsah odstavce </div></p>` - prohozené koncové značky

`<h1> Nadpis úrovně H1 </h>` - špatně napsaná koncová značka

`` - hodnota atributu `src` není v uvozovkách

S modernizací internetového připojení do domácností, které se v dnešní době pohybuje v řádech megabajtů za vteřinu, se rozvíjí i pestrý grafický design jednotlivých stránek s množstvím barev, obrázků, animovaných efektů a jiných grafických prvků, které mnohdy s funkčností webu nemají nic společného. Jejich hlavním úkolem je upoutávat cílovou skupinu návštěvníků webu. Zde vzniká problém s datovou velikostí stránek, která se mnohdy pohybuje i v řádech desítek megabajtů, což vylučuje uživatele s pomalejším připojením k internetu, uživatele mobilních zařízení nebo se slabším hardwarovým zařízením.

3.3.4 Přínosy HTML5

Nová specifikace jazyka přináší mnohé novinky v oblasti vývoje webových prezentací. Součástí nové verze je množství nových značek a atributů. Jiné značky již naopak nejsou podporovány. (Pilgrim, 2014)

3.3.4.1 Zjednodušený zápis DOCTYPE

S novou verzí HTML5 se zjednodušuje zápis deklarace typu dokument *doctype*. V dřívějších verzích se používala složitá varianta zápisu s množstvím nastavení.

HTML 4.01 Strict:

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

HTML 4.01 Transitional:

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

HTML 4.01 Frameset:

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

Výhodou nového zpracování deklarace *doctype* je její jednoduchost a přehlednost, vývojáři si nemusí pamatovat, případně dohledávat a kopírovat, složitou syntaxi s URL adresou na *.dtd* dokument.

Nová podoba deklarace vypadá následovně:

```
<!DOCTYPE html>
```

3.3.4.2 Strukturální značky

Jednou ze základních novinek jsou nové strukturální značky, které by měly nahradit zastaralý blokový element `<div>`, ty musí pro konkrétní identifikaci využívat jedinečné identifikátory. Na stránkách se velmi často objevuje nespočet bloků, které obalují obsah stránky a jejich následná editace a čitelnost zdrojového kódu je velmi náročná.

```

<div id="hlavni">
  <div id="hlavicka">
    <div id="menu">
      <ul>
        <li> Odkaz 1 </li>
        <li> Odkaz 2 </li>
      </ul>
    </div>
  </div>
</div>

```

V názorné ukázce je vidět, jak je seznam s odkazy umístěn do 3 blokových elementů `<div>`, kde každý z nich může mít jedinečné nastavení stylů pomocí identifikátoru `id`. Tento efekt se neoficiálně nazývá „*předivování*“ webové stránky.

Problémem je nepřehlednost a složitost zdrojového kódu, který často obsahuje velké množství velmi podobně nastavených elementů `<div>` a jejich uspořádání a logické použití je na hranici chaosu.

Nové strukturální značky se snaží rozdělit obsah stránky do logických celků podle jejich významu, což omezí množství používaných identifikátorů pro jejich selekci v kaskádových stylech. Tím se zlepšuje přehlednost zdrojového kódu a čitelnost jak webovým prohlížečům, tak správcům stránek při následné editaci obsahu či struktury webu.

Mezi tyto značky patří:

`<header>` identifikující záhlaví stránky, což je ideální prostor pro umístění loga a názvu společnosti. Množství použití elementu záhlaví není omezeno počtem, vývojáři ho mohou používat jako záhlaví každého článku na stránce.

`<footer>` identifikující zápatí stránky pro umístění informací o autorovi webu a autorských právech o kopírování obsahu.

`<nav>` identifikující oblast s navigací stránky. Oblast s navigací je jednou z nejdůležitějších částí každého webu zajišťující orientaci po stránce.

`<section>` identifikující logickou oblast stránky, tento blokový element je prakticky totožný se starým prvkem `<div>`. Při používání elementu `<section>`

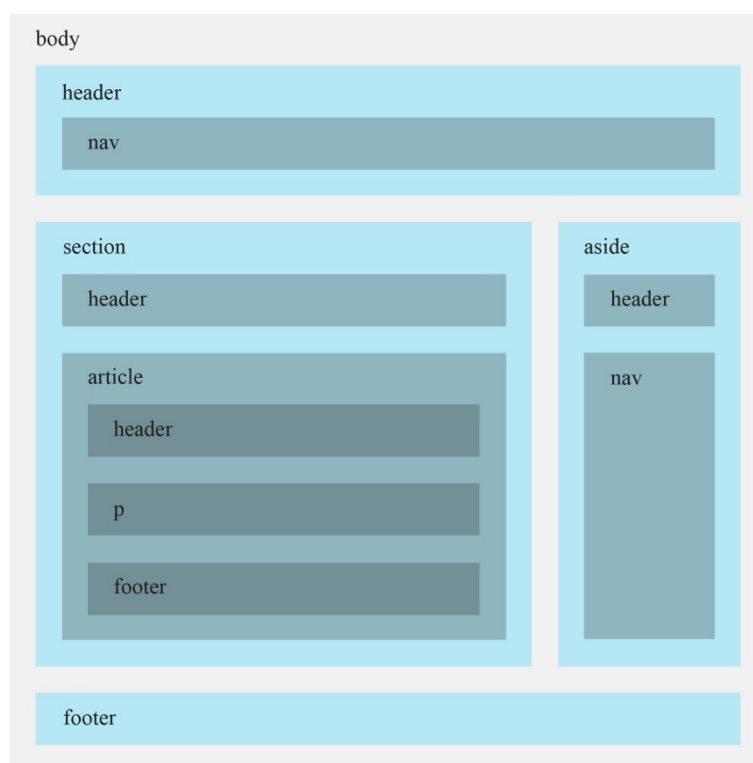
je nutno dbát zvýšené pozornosti, aby nevznikl stejný problém, jako při používání `<div>`, a to tzv. „*předivování*“ stránky.

`<article>` identifikující oblast pro článek, popř. obsahovou část webu. Tento element je svým použitím velmi podobný již zmíněnému `<section>`. Zároveň je třeba si uvědomit, že slouží k definování konkrétního obsahu stránky a ne jen jako logická oblast pro rozmístění elementů na webu.

`<aside>` identifikující prostor pro vedlejší obsah stránky, například komentář autora, postranní panel nebo například plovoucí blok s reklamou.

`<figure>` identifikující prostor pro obrázky, grafy a tabulky.

Všechny tyto nové elementy jsou v základu nastaveny jako blokové. Jsou tedy totožné s elementem `<div>`, a jejich hlavní rozdíl je především v sémantice, kde jsou jednoznačně pojmenovány podle logických oblastí webu. (Hogan, 2012)



Obrázek 2 - Logická struktura stránky pomocí nových sémantických značek

3.3.4.3 Uživatelsky přívětivé formuláře

Dalším velmi užitečným rozšířením HTML5 jsou nové formulářové prvky. V původní specifikaci je možnost použití polí pro vkládání uživatelských dat mnohdy

omezeno na pouhé pole typu *text*. Problémem je velmi skromné nastavení, které se týká efektů při vkládání dat a dále jejich následným zpracováním.

O rozšířené možnosti formulářových prvků se mnohdy webový vývojáři uchylují k použití dalších programovacích jazyků a jejich knihoven, jako je například JavaScriptová knihovna jQuery. Tato varianta nabízí široké spektrum možností, které je podporované na všech platformách bez nutnosti instalace nezbytného plug-inu, jako je tomu například u flash aplikací.

Nevýhodou těchto aplikací je však jejich náročnost na realizaci, která je mnohdy velmi nákladná. Jejich následná editace se často stává časově velmi náročnou a rovněž je nutno dbát velké pozornosti na ověřování dat vložených do jednotlivých polí.

Proto specifikace HTML5 přináší množství standardně používaných typů elementu `<input>`, které se doposud musely modifikovat speciálními identifikátory a programátorskými úpravami, popř. je bylo nutné kompletně modelovat prostřednictvím zásuvných modulů.

HTML5 nabízí následující možnosti:

`<input type="email">` je alternativa elementu `<input>` pro vkládání emailové adresy. Na první pohled vypadá totožně jako klasický `<input>` typu „*text*“. Při odeslání však automaticky kontroluje, zdali zadaná emailová adresa je ve správném formátu.

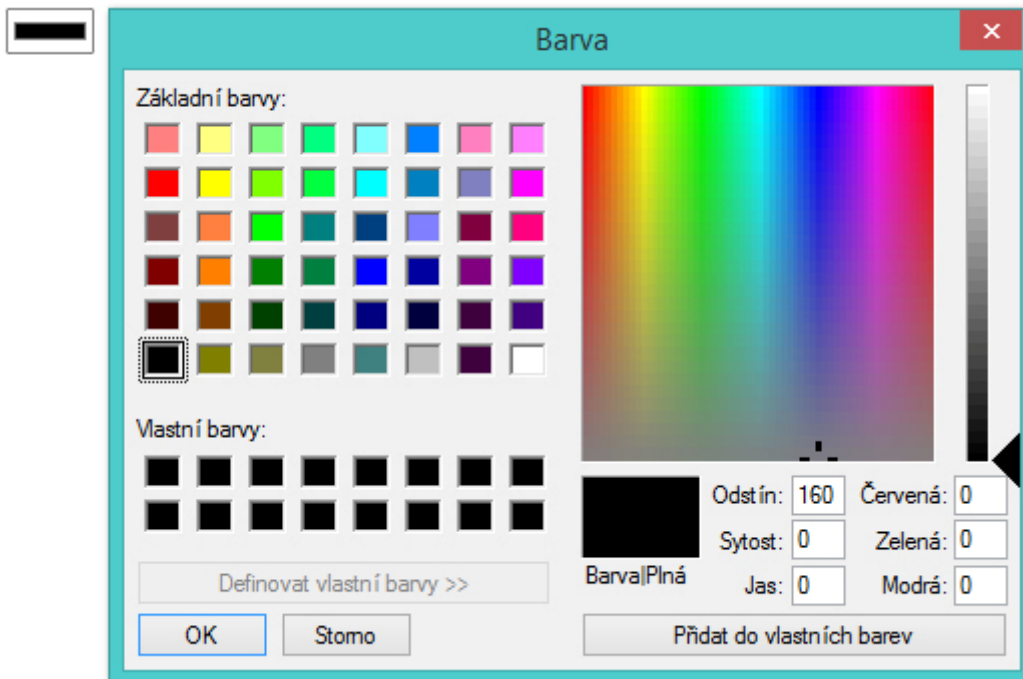
`<input type="tel">` je alternativa elementu `<input>` pro vkládání telefonního čísla, rovněž je na první pohled totožná s klasickým `<input>` typu „*text*“. Toto pole je předurčeno pro vkládání telefonního čísla. Žádné ověření platnosti však žádný z dnešních prohlížečů nepodporuje.

`<input type="search">` je alternativa elementu `<input>` pro vkládání frází a slovních spojení pro fulltextové vyhledávání.

`<input type="url">` je alternativa elementu `<input>` pro vkládání URL adresy. Po odeslání je zkontrolován správný formát adresy.

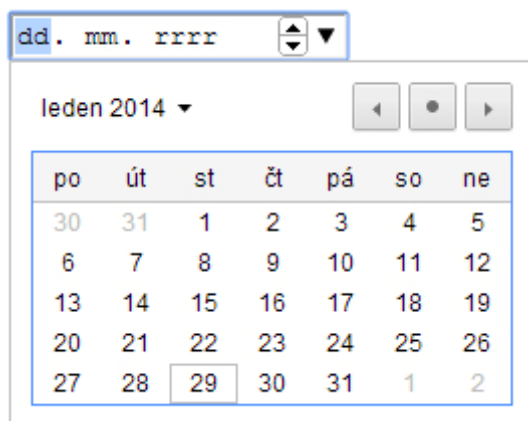
`<input type="number">` je alternativa elementu `<input>` pro vkládání celých čísel z předepsaného rozsahu.

`<input type="color">` je alternativa elementu `<input>` pro výběr barvy z předdefinované palety.



Obrázek 3 - Pole pro výběr barvy [Google Chrome 33.0]

`<input type="date">` je alternativa elementu `<input>` pro vkládání data. Po kliknutí na formulářový prvek vyjede automaticky generovaná konzole pro vybrání data z kalendáře. Uživatel datum může napsat i ručně.



Obrázek 4 - Formulářový prvek pro vkládání data [Google Chrome 33.0]

`<input type="datetime">` je alternativa elementu `<input>` pro vkládání data a času. Je velmi podobný typu „date“, jen umožňuje připojit navíc ještě čas. Další možností pro nastavení tohoto elementu je `type="time"`, který nabízí vkládání pouze času, případně `type="datetime-local"` pro vkládání místního času a data.

`<input type="range">` je alternativa elementu `<input>` pro výběr z předdefinovaného rozsahu prostřednictvím jezdecky.



Obrázek 5 - Formulářový element typu range[Google Chrome 32]

Vedle nových atributů elementu `<input>` se objevují i zcela nové prvky `<meter>` a `<progress>`. Ty umožňují sledovat stav a průběh určité akce.

Element `<meter>` umožňuje zobrazovat polohu mezi dvěma pevně nastavenými body, což je výhodou například při potřebě zobrazit kolik zbývá do konce vyplnění celého formuláře, nebo například kolik je potřeba peněz do naplnění celkové částky. Funkce tohoto prvku je pasivní a zobrazuje jen fixní body na ose.

Konkrétní zapsání elementu `meter` vypadá následovně:

```
<meter title="penize" value="500000" min="0" max="1000000">
```

Momentálně je zapláceno 500 000,- Kč z celkové částky `</meter>`, kde atribut `value` určuje množství zpracovaných dat a atributy `min` a `max` definují počátek a konec osy.



Obrázek 6 - Element `<meter>`

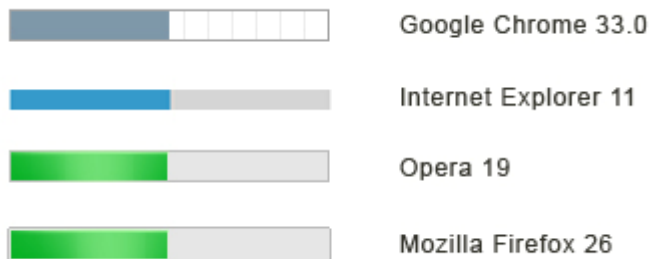
Element `<progress>` naopak funguje aktivně a zobrazuje živý průběh zvolené operace, což může například být stav nahrávání souboru na FTP server.

Zápis elementu je velmi podobný jako u předchozího příkladu, lze však rozlišit, zda-li je průběh zpracování konkrétní hodnota:

```
<progress id="stav" max="100" value="50">
```

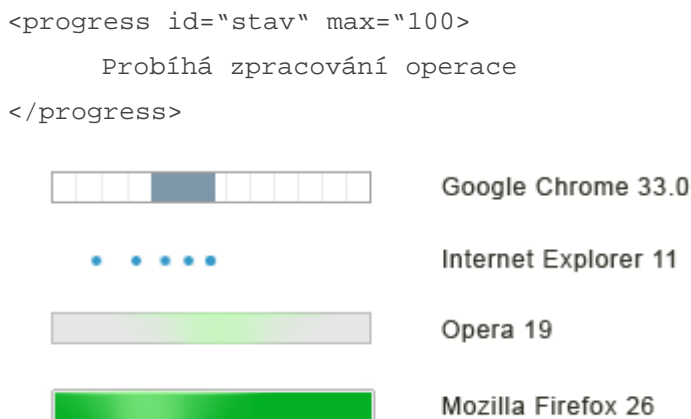
Zpracováno 50% operace.

```
</progress>
```



Obrázek 7 - Progress bar s nastavenou hodnotou na 50 / 100

Může se stát, že není nastaveno, jaká část již proběhla a kolik zbývá. Potom se zobrazí nekonkrétní jezdec indikující právě probíhající akci:



Obrázek 8 - Progress bar s nenastavenou aktuální hodnotou

Další novinkou v oblasti tvorby webových formulářů je atribut `autofocus`, který vybraný prvek nastaví jako aktivní, čímž se uživatel snadno navede na stránce k vyplnění například přihlášení. Na jedné stránce může být použito i více prvků s atributem `autofocus`. Nevýhodou je však potlačení samotné funkce. Aktivním polem se stává jen poslední na stránce, který má hodnotu `autofocus` nastavenou.

Většina těchto nových elementů se potýká s nejednotnou podporou v prohlížečích, které nemají jednotný standard zobrazení ani funkce. Výhodou však je jejich zpětná podpora u starších prohlížečů, které sice nedovedou mnohdy rozlišit konkrétní typ elementu `<input>`, dovedou ho však převést na klasický typ „`text`“. To znamená, že je tvůrci webových stránek mohou bez větších obav používat na svých stránkách. (Hogan, 2012)

3.3.4.4 Element Canvas

Velkou novinkou je element `<canvas>`, který funguje jako plátno pro kreslení vektorové grafiky. Je to také ideální nástroj pro jednoduché animace, grafy, či diagramy, který nahrazuje zásuvný modul flash.

Element `canvas` funguje jako kontejner (podobně jako `script`) a umožňuje webovým vývojářům kreslit objekty na plátno prostřednictvím objektově orientovaného programovacího jazyku JavaScript. Je nutné nastavit atribut `id` pro jeho jednoznačnou identifikaci prostřednictvím funkce `getElementById`. Dále je potřeba si předem rozmyslet, jak velké plátno bude nutné pro kreslení, protože následná editace velikosti prostřednictvím kaskádových stylů není podporována.

Základní zápis elementu vypadá následovně:

```
<canvas id="moje_platno" width="500" height="500"></canvas>
```

Element pracuje na souřadnicovém principu s osami X a Y a počátečním bodem (0,0), který se nachází v levém horním rohu plátna.

Na plátno lze snadno kreslit jednoduché čáry, geometrické obrazce nebo pomocí pokročilých funkcí vytvářet složité tvary a objekty.

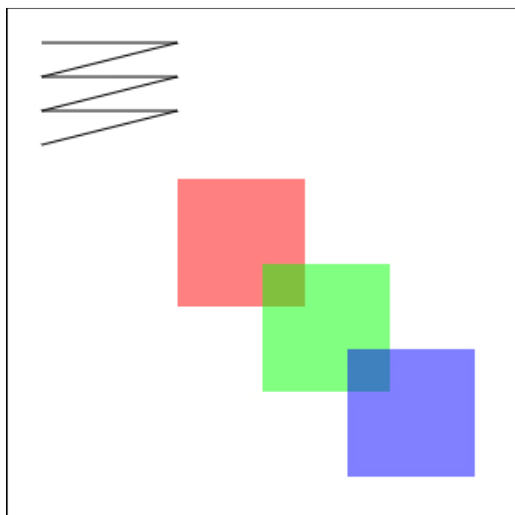
Pro použití plátna je potřeba identifikovat ho pomocí funkce `getElementById` a vytáhnout 2D kontext plátna:

```
var platno=document.getElementById("moje_platno");  
var kontext=platno.getContext("2d");
```

Pro kreslení obrazců složených z více objektů existují funkce `beginPath()` a `closePath()`, které definují začátek a konec složené cesty. Dále funkce pro vykreslování čar a objektů (v následující ukázce pro vykreslení obdélníku):

```
kontext.beginPath();  
    kontext.moveTo(20,20);  
    kontext.lineTo(100,20);  
    kontext.lineTo(20,40);  
    kontext.lineTo(100,40);  
    kontext.lineTo(20,60);  
    kontext.lineTo(100,60);  
    kontext.lineTo(20,80);  
    kontext.stroke();  
  
    kontext.fillStyle="rgba(255,0,0,0.5)";  
    kontext.fillRect(100,100,75,75);  
    kontext.fillStyle="rgba(0,255,0,0.5)";  
    kontext.fillRect(150,150,75,75);  
    kontext.fillStyle="rgba(0,0,255,0.5)";  
    kontext.fillRect(200,200,75,75);  
kontext.closePath();
```

Výstupem výše zmíněného zdrojového kódu jsou čáry nakreslené ve tvaru zubů a 3 různě barevné průhledné čtverce – červený, zelený a modrý. (Hogan, 2012)



Obrázek 9 - Základní použití elementu canvas

3.3.4.5 Přístupnost webu

V dnešní době se rozmáhá i množství uživatelů, kteří prohlíží webové stránky prostřednictvím čteček obrazovky popř. speciálních čtecích zařízení pro nevidomé.

Ty převádějí textový obsah stránky do mluveného slova, k čemuž využívají jak samotný obsah webu, tak HTML elementy použité ve zdrojovém kódu, pomocí čehož se snaží na stránce orientovat. (Friendly, 2014)

Takto optimalizované weby by měly dodržovat metodiku Blind Friendly Web 2.3³.

Ta ve stručnosti vypovídá o základní sémantice webové stránky a principu použití jednotlivých elementů na stránce:

- všechny grafické objekty na stránce musí mít svojí textovou alternativu,
- prvky zobrazené prostřednictvím zásuvných modulů nebo jiných aplikací musí být dostupné i při vypnutí těchto modulů,
- tabulky jsou čitelné po řádcích, protože většina čteček obrazovky postupuje z levého horního rohu po řádcích až do pravého dolního rohu stránky,
- obsah stránek se mění jen po interakci uživatele se stránkou, je třeba aktivovat nějaký prvek – například kliknout na hypertextový odkaz,
- validita a správná sémantika zdrojového kódu,

³<http://www.blindfriendly.cz>

- odkazy jsou atributem `title` a obsahem samotného odkazu popsány konkrétně a pochopitelně i bez nutnosti čtení kontextového obsahu,
- odkazy, které směřují na externí adresu, jsou řádně popsány poznámkou o jejich cestě.

A je mnoho dalších podobných standardů pro ulehčení pochopitelnosti webové stránky nevidomým uživatelům. (Pavlíček, 2005)

HTML5 přináší několik novinek, které se snaží rozšířit možnosti přístupného webu, které převzala z aplikace RIA (WAI-ARIA)⁴.

Výhodou nových atributů je, že nijak neovlivňují funkčnost elementů v zastaralých verzích prohlížečů, takže je možné je bez obav využívat.

Atribut `role` identifikuje logickou část webu nebo samotného obsahu stránky a zařazuje ji do tematického bloku podobně jako strukturální značky, což čtečkám usnadňuje orientaci v obsahu.

Zápis atributu `role` vypadá následovně:

```
<div role="typ role"></div>
```

Role se dělí na *role orientačních bodů* a na *role dokumentové struktury*.

3.3.4.5.1 Role orientačních bodů

Role rozdělují stránku na logické části podle jejich obsahového sdělení. Většina stránek vždy začíná hlavičkou, která je ve všech jednotlivých podstránkách totožná. Role umožňují čtečkám tyto části při opakování přeskakovat a chytat se kotvy, které jsou umístěny až na začátku obsahové části.

Mezi předdefinované role orientačních bodů patří:

`banner`, která identifikuje oblast s hlavičkou webu;

`search`, která identifikuje oblast s vyhledávacím formulářem, případně mapu stránky;

`navigation`, která identifikuje oblast s ovládacími prvky stránky;

`main`, která identifikuje oblast s hlavním obsahovým sdělením webu;

⁴<http://www.w3.org/WAI/intro/aria.php>

`contentinfo`, která identifikuje oblast s právy o kopírování obsahu, sdělení o datu vytvoření stránky, o autorovi apod.;

`complementary`, která identifikuje oblast s vedlejším obsahem;

`application`, která identifikuje oblast s webovou aplikací.

3.3.4.5.2 Role dokumentové struktury

Rozdělují obsahovou část dokumentu na základní části, které usnadňují pochopení obsahu.

`document`, která identifikuje obsah samotného dokumentu, vhodná i pro definování sloučení statického a dynamického obsahu stránky,

`article`, která identifikuje jednotlivé články umístěné v dokumentu,

`definition`, která identifikuje definici použitého pojmu,

`direktory`, která identifikuje seznam odkazů,

`heading`, která identifikuje záhlaví dokumentu,

`img`, která identifikuje oblast s vloženým obrázkem,

`list`, která identifikuje seznam,

`listitem`, která identifikuje jednotlivé položky v seznamu,

`math`, která identifikuje matematickou funkci nebo výraz,

`note`, která identifikuje poznámku,

`presentation`, která identifikuje obsah dokumentu, který je určen k prezentaci,

`row`, která identifikuje řádky v tabulce,

`rowheader`, která identifikuje názvy sloupců v tabulce.

Velké množství těchto rolí jednoznačně identifikuje jednotlivé HTML elementy. I tak je vhodné těmto prvkům přesto tyto role přidělovat, protože samotné použití rolí může přesněji specifikovat jednotlivé části stránky. (Hogan, 2012)

3.3.4.6 Práce s multimédií

Mezi nové elementy HTML5 rovněž patří `<audio>` a `<video>`, které nabízejí možnost použití audio a video souborů bez nutnosti zásuvných modulů.

Multimediální soubory jsou prakticky jen kontejnery dalších dat, podobně jako archiv typu ZIP. Skládají se z video stopy, audio stopy a meta dat.

Nejpopulárnějšími formáty pro vytváření video souborů jsou: .avi (Audio Video Interface), .flv (Flash Video, .mp4 (MPEG4), .mkv (Matroska) a .ogg (formát ogg).

3.3.4.6.1 Kodeky

Kodeky (***Kodér a Dekodér***) jsou předpisy pro kódování a dekodování audio nebo video stopy, což komprimuje původní typ nahrávky a snižuje její datovou velikost. (Lubbers, a další, 2011)

Nejčastěji používané audio kodeky jsou ACC, MPEG-3 a OggVorbis.

Nejčastěji používané video kodeky jsou H.264, VP8, OggTheora.

3.3.4.7 Možnosti vytváření aplikací

Mezi jiná využití jazyka HTML5 patří některé další aplikace založené na využívání JavaScriptu. Ty však nejsou vzhledem k rozsahu předmětem této bakalářské práce.

3.3.4.7.1 WebStorage

Ve spojení s pojmem webového úložiště je nutné popsat i jeho předchůdce, soubory Cookies.

Ty poskytují předávání textových hodnot mezi klientem a serverem, pomocí kterých lze sledovat aktivitu uživatele a získávat o něm informace. Nejčastější použití souborů Cookies je v oblasti permanentního přihlašování uživatele na stránkách.

Soubory Cookies se však využívají i k analýze návštěvnosti jednotlivých stránek, což vede k cílení reklamy na uživatele. To je mnohdy nežádoucí a mnoho uživatelů si z toho důvodu používání Cookies zablokuje.

Rozhraní WebStorage nabízí možnost využívání podobných služeb i bez potřeby komunikace se serverem, což je jeho největší výhodou. Dalším plusem je, že rozhraní

umožňuje lokálně ukládat soubory o velikosti až několik megabajtů. To je velmi vhodné pro ukládání obsáhlých dokumentů. (Lubbers, a další, 2011)

3.3.4.7.2 Rozhraní WebSocket

Rozhraní WebSocket je novinkou specifikace HTML5 pro obousměrnou komunikaci prostřednictvím jednoho socketu. Vedoucí projektu HTML5 a zaměstnanec společnosti Google, Ian Hickson, o rozhraní řekl:

„Zmenšení objemu dat z kilobajtů na 2 bajty a snížením odezvy ze 150 ms na 50 ms rozhodně není nevýznamné. Tyto 2 faktory sami o sobě činí rozhraní WebSocket velmi zajímavým pro Google.“ (Hickson, 2009)

Jak už bylo řečeno výše, protokol HTTP pracuje na principu dotaz / odpověď. To v praxi znamená, že uživatel zašle serveru požadavek na získání dat, ten server vyhodnotí a odešle odpověď zpět uživateli. Pro získávání dat v reálném čase je potřeba zasílat dotaz neustále dokola serveru, což zatěžuje linku a zvyšuje odezvu serveru. Tato funkce je obvykle řešena prostřednictvím technologie AJAX.

Rozhraní WebSocket umožňuje navázání spojení mezi uživatelem a serverem prostřednictvím stejnojmenného protokolu WebSocket, který nahrazuje dotazovací HTTP protokol, a toto spojení udržuje po celou dobu přenosu. Není nutné zasílání opakovaných dotazů o aktualizace na straně serveru. (Lubbers, a další, 2011)

Funkce WebSocket nabízí obousměrnou komunikaci mezi 2 zařízeními. Ta umožňuje zasílat data pomocí jednoho vyčleněného socketu a nabízí možnosti, které AJAX a jiné aplikace nemohly. (Malý, 2009)

3.3.4.7.3 Geolokace

Dalším rozšířením funkcí HTML5 je geolokace, která využívá možností zařízení na zjištění geografických souřadnic, kde se uživatel právě nachází. HTML5 však neobsahuje samotnou funkci pro vygenerování těchto dat, ale pouze funkci pro jejich získání. O samotné zjištění hodnot se stará zařízení, na kterém je web prohlížen. (Albers, a další, 2010)

3.4 Kaskádové styly

CSS tvoří jakýsi obal webové stránky a udává její vzhled, který je uživatelsky přívětivější a čitelnější, než pouhý čistý text. Pomocí kaskádových stylů je možné určit vzhled téměř všech elementů HTML, jejich rozložení a umístění. Je to jazyk popisující způsob zobrazení elementů HTML. Jeho hlavním smyslem je oddělení grafické podoby stránek od jejich struktury.

Správné použití stylů umožňuje zvýšení přehlednosti stránky. Vývojáři mohou prostřednictvím množství pseudotříd vytvářet efekty, které uživatele navádějí po stránce a umožňují mu tak snazší orientaci v jinak velkém množství informací.

Mezi tyto pseudotřídy patří například:

`:hover` efekt po najetí na prvek, nejčastěji používaný u odkazů,

`:link` identifikuje doposud nenavštívený odkaz na stránce,

`:visited` oproti tomu identifikuje již navštívený odkaz, tedy odkaz, na který uživatel již kliknul,

`:active` nebo `:focus` je ten prvek, na který se naposledy kliklo, popř. ten, na který uživatel najede pomocí tabulátoru.

Další velkou výhodou kaskádových stylů je možnost používání selektorů k výběru konkrétních typů elementů k určení jejich unikátního vzhledu.

Selektorů je více druhů:

- Univerzální selektor (*) umožňuje stylovat všechny elementy pomocí jednoho stylu. Možno je k němu též připojit identifikátor `class` (.) nebo `id` (#) pro specifikování jen konkrétní elementů.
- Selektory `class` a `id`, které identifikují konkrétní elementy na stránce.
- Selektor dítěte, který využívá struktury rozmístění elementů na stránce pro jejich stylování.

`div.a p { color: red; }` všechny odstavce v divu se třídou „a“ budou mít červené písmo.

- Selektor skrze atributy umožňuje identifikovat elementy prostřednictvím některého z definovaných atributů uvnitř tohoto elementu.

`a[href=url]` umožňuje vybrat všechny odkazy, kde se atribut *href* rovná zadané URL adrese.

Je nutno si uvědomit, že podpora jednotlivých selektorů není jednotná ve všech prohlížečích a hlavně, že se selektory skrze atributy je velký problém. (Dudek, 2002)

3.4.1 Historie CSS

V roce 1996 se objevily první webové prohlížeče podporující grafické prvky, což dalo podnět k vytvoření galerie stylů, která by se dala aplikovat na jednotlivé elementy HTML.

Proto o rok později vychází první verze kaskádových stylů – CSS 1.0.

3.4.1.1 CSS 1.0

První verze CSS měla největší podporu v prohlížeči Internet Explorer, který se v té době objevoval ve verzi 3. Podpora CSS se zde omezovala hlavně na definování rodiny fontů a určování barev jednotlivých elementů. Zde již byla myšlenka oddělení vzhledu webu od jeho obsahu.

O něco lepší podpory se první verzi stylů dostává s novou verzí IE, která se snažila poskytnout širší podporu. Ale ani ta nebyla 100%, proto se vývojáři webu raději vyhýbali jejímu používání. Hlavním přínosem první verze stylů byla podpora pseudotřídy `:hover`. (Janovský, nedatováno)

3.4.1.2 CSS 2.0

Druhá verze kaskádových stylů byla představena v roce 2000. S jejím příchodem však nastává velký problém s její podporou a jednotností v prohlížečích, což se v průběhu času změnilo.

Cílem kaskádových stylů druhé generace je snaha stát se hlavním formátovacím jazykem napříč různými platformami. V dnešní době je jeho aktualizace (verze 2.1) nejrozšířenější a nejvíce podporovanou verzí jazyka vůbec. (Janovský, nedatováno)

3.4.1.3 CSS 3

S příchodem HTML5 se objevuje i nová verze stylů. Ta přináší mnohé nové možnosti zobrazování jednotlivých elementů na stránce, možnost vícenásobného

pozadí, nové pseudotřídy a spoustu jiných prvků, které se do této doby musely nahrazovat složitými JavaScriptovými aplikacemi.

Třetí verze stylů se stále ještě vyvíjí a její dokončení je předpokládáno na rok 2015. (Janovský, nedatováno)

3.4.2 Struktura CSS

Formátovací jazyk CSS má velmi jednoduchou strukturu zapisování jednotlivých prvků:

```
Element_HTML.identifikátor:pseudotřída {  
    CSS_vlastnost: hodnota_vlastnosti;  
}
```

3.4.3 Přínosy CSS 3

3.4.3.1 Nové grafické možnosti elementů

Kaskádové styly třetí generace přináší pár vylepšení v oblasti nastavení jednotlivých elementů HTML, což v praxi velmi usnadňuje tvorbu graficky přívětivých stránek a rovněž snižuje datovou velikost samotného webu.

3.4.3.1.1 Zakulacené rohy

Často je při tvorbě grafických stránek jedním z požadavků zakulacení rohů jednotlivých elementů. Doposud se tento efekt vytvářel prostřednictvím obrázků na pozadí, které se umisťovaly mnohdy i do několika přes sebe položených obalových elementů `<div>`.

Nová specifikace jazyka CSS 3 umožňuje nastavení těchto zakulacených rohů prostřednictvím zdrojového kódu, což značně usnadňuje práci a rovněž zajišťuje zdrojový kód přehlednější. (Clark, a další, 2012)

Základní zápis efektu zakulacených rohů je:

```
border-radius: LevýHorní, PravýHorní, LevýDolní, PravýDolní;
```

Velikost zaoblení se udává v pixelech.

3.4.3.1.2 Stíny

Dalším vylepšením možností pro úpravu vzhledu elementů je stínování.

Často se vývojáři snaží o dojem prostorového efektu prostřednictvím stínů, které vytvářejí představu, že obsah stránky vystupuje do popředí. To je vhodné i v případě zviditelnění důležité oblasti webu.

Tyto efekty bylo často nutné simulovat prostřednictvím obrázků na pozadí, což byl problém například u obsahu, který není jednotný – například text.

CSS přináší novou alternativu pro vytváření stínů:

- Stíny objektů `box-shadow`
- Stíny pro text `text-shadow`

Obě tyto alternativy mají stejný způsob zápisu:

```
box-shadow: horizontální_posun, vertikální_posun, ostrost;
```

3.4.3.1.3 Přejechy

Rozšířenou možnost nastavení přináší atribut `background`, kterému je možné nově místo jedné barvy nastavit i barevný přechod. Způsob zápisu pro jednotlivé prohlížeče je však velmi nejjednotný. Všeobecně by se dalo napsat:

```
background:  
gradient  
(princip_přechodu, odkud, kam, výchozí_barva, cílová_barva);
```

Nastavení těchto kaskádových stylů pro prohlížeč Google Chrome:

```
#ukazka {  
border-radius: 10px;  
box-shadow: 5px 5px 15px black;  
background: -webkit-gradient(linear, left top, left bottom,  
from(aquamarine), to(blue));  
}
```

Tyto styly mají za výsledek takovýto efekt.



Obrázek 10 - Stíny, okraje a přechody pomocí CSS

3.4.3.1.4 Vícenásobné pozadí

Atribut `background` získává i rozšířenou možnost pro nastavení obrázku na pozadí. Tím je možnost použití více jednotlivých obrázků k nastavení na pozadí jednoho elementu a jejich pozicování dle představ tvůrce. Dříve bylo potřeba vytvářet více do sebe vložených obalových prvků a těm jednotlivě nastavovat různá pozadí.

Tvorba vícenásobného pozadí vypadá následovně:

```
#ukazka {
    background-image: url(první_obrázek), url(druhý_obrázek);
    background-position: pozice_prvního, pozice_druhého;
    background-repeat: opakování_prvního, opakování_druhého;
}
```

3.4.3.2 Rozšířené pseudotřídy

Přínosem nových možností kaskádových stylů je i rozšíření stávajících možností pseudotříd. Ty se do této doby omezovaly pouze na efekty, které identifikovaly akci uživatele (njetí myši, navštívení odkazu, ap.).

Novinkou jsou pseudotřídy, které umožňují identifikovat jednotlivé elementy podle vztahu k jejich okolí a zjednodušují práci například při vytváření přehledného uživatelského prostředí v tabulkách. To se do nedávna muselo dělat složitými skripty, které počítaly jednotlivé prvky a následně jim přidělovaly různě nastavené třídy, což navíc zahlcovalo zdrojový kód množstvím atributů `class`.

Seznam nových selektorů vypadá následovně:

- `:nth-of-type` selektor vyhledá všechny n-té elementy vybraného typu,
- `:first-child` selektor vyhledá prvního potomka zvoleného elementu,
- `:nth-child` selektor najde konkrétního potomka zvoleného elementu prostřednictvím jeho pořadí,
- `:last-child` selektor najde posledního potomka zvoleného elementu,
- `:nth-last-child` selektor najde konkrétního potomka zvoleného elementu prostřednictvím jeho pořadí od konce,
- `:first-of-type` selektor najde první element zvoleného typu,

`:last-of-type` selektor najde poslední element zvoleného typu,

`:after` selektor vybere element následující.

3.4.3.3 Písma a transformace

Do dnešní doby bylo zvykem veškeré prvky na stránce, ať už texty, obrázky nebo celé bloky obsahu, umisťovat horizontálně. To už dnes neplatí. S příchodem kaskádových stylů třetí generace je možné jednotlivé elementy na stránce transformovat. Objekty se dají otáčet, měnit jejich velikost, zkosit je apod.

`translate()` umožňuje posunutí prvku,

`rotate()` umožňuje otáčení prvku,

`scale()` umožňuje změnu velikosti prvku,

`skew()` umožňuje deformaci prvku,

`matrix()` umožňuje kombinaci všech předchozích metod v jedné.

3.4.3.4 Vícesloupcový layout

Další novinku, kterou CSS3 přináší, je podpora sloupců, která je známa z tištěných publikací. Takto upravený text je snáze čitelný a v obsahu se lépe orientuje. To především na obrazovkách s vysokým rozlišením, kde se obsah stránky roztahuje na celou šířku.

K vytváření vícesloupcového layoutu složí atribut `column-count`, jehož hodnotou je celé číslo, které určuje počet sloupců. (Hogan, 2012)

Další vlastnosti, které nastavují vzhled těchto sloupců jsou:

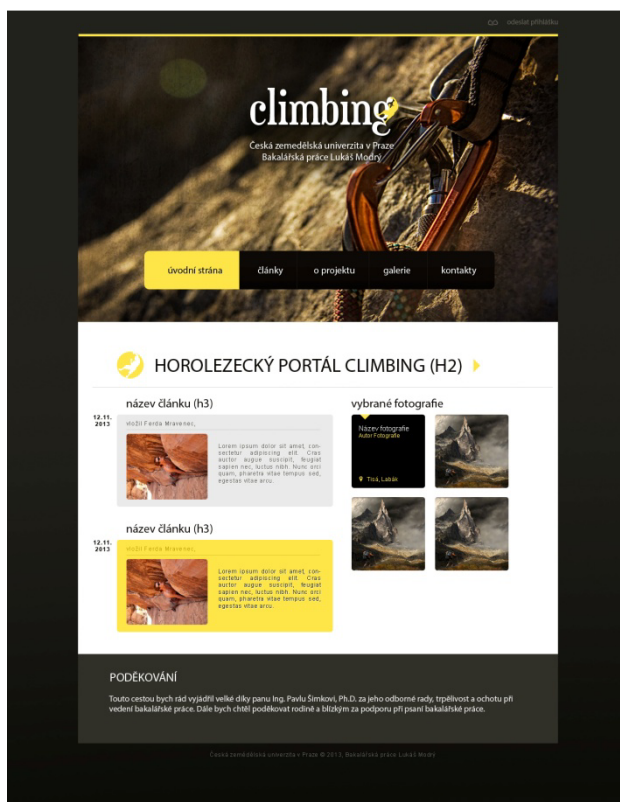
- `Column-gap`, která určuje mezeru mezi jednotlivými sloupci,
- `Column-rule`, která určuje vzhled těchto sloupců.

V praxi by vytvoření dvousloupcového layoutu mohlo vypadat následovně:

```
p {
    column-count: 2;
    column-gap: 20px;
    column-rule: 1px solid red;
}
```

4 Praktická část práce

Předchozí část pojednávala o jazycích HTML a CSS a o přínosech jejich posledních specifikací. V praktické části práce jsou vytvořeny dvě obdobné webové prezentace. První z nich je napsána za pomoci technologie XHTML1.1 a CSS2.1, druhá prostřednictvím HTML5 a CSS3. Zdrojový kód těchto stránek je kompletně psán bez použití WYSIWYG⁵ aplikací a vychází z předem vytvořeného grafického návrhu v aplikaci Adobe Photoshop CS3.



Obrázek 11 - Grafický návrh pro tvorbu webových prezentací

Stránka je navržena jako jednoduchý blog s horolezeckou tematikou. Obsahem webu jsou články a fotografie, které nabízejí ideální možnosti pro znázornění rozdílů mezi oběma specifikacemi.

⁵ What you see is what you get

Součástí prezentace je 5 podstránek:

- „Úvodní strana“ s náhledy posledních článků a nejlepších fotografií,
- „Články“ se seznamem všech publikovaných článků,
- „O projektu“ s popisem práce,
- „Galerie“ s fotografiemi,
- „Kontakty“ s formulářem pro napsání vzkazu.

4.1 Změna v zápisu DOCTYPE

Zásadní rozdíl mezi oběma specifikacemi nalezneme hned na prvním řádku každého HTML dokumentu, tím je *DOCTYPE*.

Ve verzi XHTML1.1 je zápis dokumentového typu velmi složitý a obsahuje mnoho dalších prvků, jako je například odkaz na *.dtd* dokument na serveru www.w3.org. Nevýhodou tohoto zápisu je, že si ho vývojáři musí buď zapamatovat, nebo neustále dohledávat.

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

Oproti tomu v jazyce HTML5 došlo ke zjednodušení tohoto zápisu, který je odlehčen o nepotřebné části, které jsou dnes již součástí zažitého standardu.

```
<!DOCTYPE html>
```

Výhodou zjednodušeného zápisu typu dokumentu je i zpětná kompatibilita se staršími prohlížeči, které si s ní dovedou poradit. S výjimkou Internet Exploreru 6 (a starší) prohlížeče nemají problém s vykreslováním takto deklarovaných stránek. (Mikula, 2012) Je tedy výhodné využívat rovnou nový zápis *DOCTYPE* z verze HTML5.

4.2 Využití nových strukturálních značek

Při tvorbě webových prezentací v XHTML1.1 se vývojáři nejčastěji spoléhají na blokový element `<div>`, ne jinak je tomu i při tvorbě webové prezentace, která je součástí této bakalářské práce.

Základní struktura webu je rozdělena do několika základních částí:

- horní panel s odkazem na formulář s přihláškou,
- hlavička webu s logem a názvem stránky, která rovněž slouží jako hlavní grafický prvek webu,

- hlavní menu, které je součástí hlavičky a slouží k navigaci skrz obsah stránky,
- obsahová část neboli prostor pro jednotlivé články a fotografie,
- patička webu s právy na sdílení obsahu.

Veškeré tyto části stránky jsou vytvořeny pomocí elementů `<div>`, které jsou od sebe odlišeny identifikátory, které je jednoznačně definují. Kostra stránky vypadá následovně:

```
<div id="all">
  <div id="top"></div>
  <div id="header">
    <div id="mainmenu"></div>
  </div>
  <div id="content"></div>
  <div id="thanks"></div>
  <div id="footer"></div>
</div>
```

Při psaní složitějších webových stránek je třeba dbát na přehlednost zdrojového kódu, proto je nutné volit sémanticky správné názvy identifikátorů jednotlivých prvků `<div>` a dodržovat jednotnou strukturu kódu pomocí tabulátorů, které na vzhled samotné stránky nemají následný vliv.

HTML5 přináší nové strukturální elementy pro zlepšení přehlednosti zdrojového kódu, rovněž i pro zlepšení sémantiky jazyka a přístupnosti pro zrakově postižené uživatele. Kostra stránky v HTML5 vypadá následovně:

```
<section id="all">
  <nav id="top"></nav>
  <header id="banner">
    <nav id="mainmenu"></nav>
  </header>
  <section id="content"></section>
  <aside id="partners"></aside>
  <footer></footer>
</section>
```

I zde jsou použity specifické identifikátory pro snazší výběr elementů prostřednictvím kaskádových stylů a stále je nutné dbát na přehlednost zdrojového kódu využíváním správné a jednotné struktury kódu.

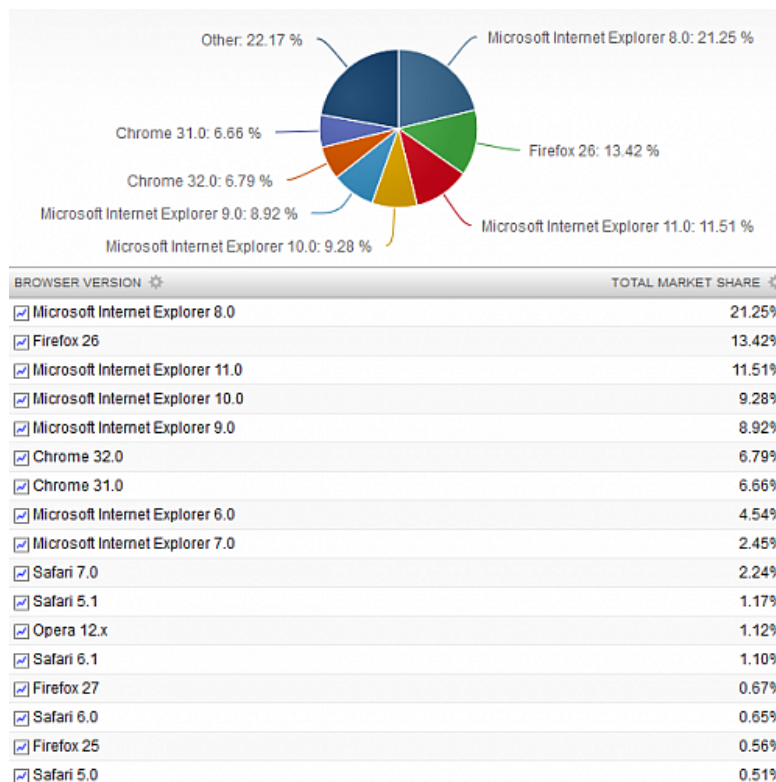
Přínosem nových strukturálních značek s doplněnou sémantikou je přesnější popis obsahu stránek, což umožňuje vyhledávacím robotům snáze analyzovat jednotlivé části webové prezentace a docílit tak lepších a korektnějších výsledků vyhledávání na základě dotazu uživatele. Oproti elementům `<div>`, které nemají žádný sémantický význam, tento význam nové značky mají. (Greguš, 2013)

Na počátku roku 2014 stále webové vyhledávače nedávají těmto značkám velkou váhu, což dokazuje i výrok ve veřejné diskusi zaměstnance společnosti Google, Johna Muellera, který ve volném překladu zní:

„Náš systém indexování a procházení webu v současné době nedělá žádné zvláštní opatření pro značky HTML5, takže neexistuje žádný bonus, ale ani žádná ztráta pro stránky psané v jazyce HTML5.“ (Meuller, 2012)

Strukturální značky HTML5 jsou podporovány ve všech posledních verzích používaných internetových prohlížečů⁶. Největším problémem je nedostatečná podpora u prohlížečů Internet Explorer 8 a starší, které jsou stále velmi rozšířené mezi uživateli. Podle statistik na webu extrawindows.cnews.cz je podíl uživatelů k lednu 2014 následovný:

⁶ Internet Explorer 11, Opera 19, Mozilla Firefox 27, Apple Safari 5, Google Chrome 32



Obrázek 12 - Podíly internetových prohlížečů, Leden 2014,

zdroj: <http://extrawindows.cnews.cz/>

Z toho plyne, že je stále více jak 25% uživatelů internetu s prohlížečem, který nové strukturální značky nepodporuje. (Urban, 2014)

Pro dosažení většinové podpory internetových stránek je lepší se používání těchto značek, které mají rapidní vliv na funkci a vzhled webové prezentace, ještě na nějaký čas vyvarovat a zvolit plně podporovanou, byť sémanticky horší variantu s používáním elementu `<div>`.

4.3 Doplnění sémantiky jazyka a přístupnost webu v HTML5

xHTML1.1 se potýká s nedostatečnou sémantikou jazyka, který nedostatečně popisuje obsah webu. Naopak jazyk HTML5 se velmi zaměřuje na rozšíření přístupnosti webových stránek pro zrakově postižené uživatele, kteří využívají různá zařízení pro převod textu do mluveného slova. K tomu využívá množství prvků, které doplňují sémantiku jazyka pro přesnější popsání obsahu stránek.

Na stránce psané v jazyce HTML5 jsou tyto prvky použity v celém rozsahu zdrojového kódu podle předepsaných standardů WAI-ARIA.

Používání nových atributů k doplnění sémantiky stránek není ovlivněno podporou jednotlivých prohlížečů. Ty z velké většiny tyto prvky nepodporují, ale nemají s jejich aplikováním větší problém. Jediným problémem je, že se jejich funkce nijak neprojeví.

Naopak většina nových zařízení sloužících ke čtení obrazovky dovede tyto prvky plně využívat a usnadnit čtení stránek zrakově postiženým uživatelům. Je tedy vhodné je využívat, je-li požadavkem stránek podpora pro tyto uživatele.

4.4 Přechody

S vývojem internetu je kladen stále větší důraz na grafické prostředí internetových stránek. Moderní grafické aplikace typu Adobe Photoshop návrhářům nabízejí nepřehledné množství možností, jak docílit dokonalého uživatelského prostředí, ať už jsou to různé piktogramy zastupující kategorie stránek, efektivní image obrázky v hlavičce nebo pestrobarevné hlavní menu.

Jednou z možností, jak udělat internetovou prezentaci atraktivnější oku uživatele, je vytváření barevných přechodů na pozadí jednotlivých prvků webu. Takové přechody vytvářejí dojem prostorovosti a narušují tak pocit, že je stránka pouhým plochým textem na obrazovce stránky.

Před příchodem CSS3 byl tento efekt nutný aplikovat pomocí obrázků umístěných na pozadí jednotlivých elementů. Vytváření obrázků s přechodem nese značná úskalí. Jedním z nich je datová velikost, která při použití kruhového přechodu, kde je potřeba většího rozměru podkladových dat s velikostí přesahující několik set kilobajtů.

Jedním z řešení velikosti obrázků je jejich komprese, která má bohužel za následek snížení kvality grafických dat a konkrétně u přechodů je tento efekt znát u jejich vykreslování, kde vznikají viditelné pruhy, či kruhy v závislosti na typu přechodu.

Použití obrázku na pozadí celé stránky vypadá následovně.

```
body {  
    background-image: url('body-background.jpg');  
    background-repeat: repeat-x;  
}
```

Nová specifikace kaskádových stylů problém s velikostí dat a nutnou kompresí obrázků řeší novinkou v podobě možnosti definování přechodu přímo v nastavení kaskádových stylů. Hlavní výhodou je, že přechod je vykreslován jádrem prohlížeče a není potřeba stahovat nutné grafické podklady.

```
body {  
    background: linear-gradient(#24231d, #090907);  
}
```

Nevýhodou složitějších přechodů je jejich náročnost na vykreslování, hlavně na slabších zařízeních a chytrých telefonech. Problém může vzniknout i při zápisu složitějších přechodů o větším množství barev, který je velmi nepřehledný.

Dalším velkým problémem používání přechodů kaskádových stylů je jejich nedostatečná podpora v jednotlivých prohlížečích. Plná podpora těchto přechodů stále není ucelená. Přestože ve většině posledních verzí prohlížečů funguje standardní zápis, který je zmíněný výše, je třeba využívat specifická jádra pro jednotlivé prohlížeče a jejich starší verze.

Apple Safari

```
background: -webkit-linear-gradient(#24231d, #090907);
```

Opera

```
background: -o-linear-gradient(#24231d, #090907);
```

Mozilla Firefox do verze 15

```
background: -moz-linear-gradient(#24231d, #090907);
```

Podpora těchto přechodů tedy stále není úplná a je lepší využívat obrázky na pozadí, které se na všech zařízeních i ve všech prohlížečích zobrazují totožně. Mimo jiné je i zápis kaskádových stylů podstatně přehlednější a mnohem kratší.

4.5 Zakulacené rohy

Webová stránka jako taková je složena z obdélníkových tvarů, které mají v základu ostré hrany. Často se návrháři stránek snaží tento hranatý dojem z webu rozbít zakulacením hran, což napomáhá i zajímavější grafické podobě stránek.

Před příchodem CSS3 se tento efekt obvykle vytvářel prostřednictvím obrázku na pozadí. S tím bohužel byla spojena náročnost na velikost dat potřebných ke stažení všech těchto pozadí. Dalším problémem byla schopnost takového pozadí přizpůsobit se obsahu elementu, který se při překročení určité délky zvětšoval.

Jednou z možností, jak tento problém vyřešit, je obalování obsahového bloku dalším a jednotlivé kousky pozadí (horní zakulacení, dolní zakulacení) umisťovat zvlášť. Efekt zakulacených rohů je názorně ukázán ve více částech webu, nejlépe je to vidět u náhledu článku.

Horolezení v Itálii

19.2.
2014

autor je Lukáš Modrý



Mauris aliquet pretium placerat. Integer urna lacus, mattis sed blandit sed, rhoncus et velit. Fusce auctor velit metus, eu ultricies risus elementum sed. In pharetra nisl vitae mattis pharetra. Morbi tristique nec nisl nec vehicula. Pellentesque ut dui quis arcu ultrices mattis. Ut eu feugiat velit. Morbi id aliquet risus, in faucibus nunc. Vestibulum quis mollis arcu.

Obrázek 13 - Zakulacené rohy u náhledu článku

Základem tohoto prvku je obrázek umístěný na pozadí odkazu. Zakulacené rohy jsou již součástí tohoto obrázku a je jistota, že se na všech zařízeních zobrazí zcela totožně.

Odkaz je nastaven následovně:

```
.clanek-hp .content a {  
    display: block;  
    width: 520px;  
    height: 200px;  
    background-image: url('clanek-bg.jpg');  
    background-repeat: no-repeat; }
```

Nutnost používání obrázků na pozadí řeší jednoduchý atribut CSS3 s názvem `border-radius`, kterému přibude ještě potřeba nastavení barvy na pozadí pomocí `background-color`. Následné vygenerování zakulacených rohů zajistí samotný webový prohlížeč za pomoci kaskádových stylů a není potřeba využívat jakýkoliv obrázek do pozadí při dosažení stejného vzhledu.

```
.left .clanek article a {
    display: block;
    width: 470px;
    height: 170px;
    background-color: #eaeaea;
    border-radius: 10px;
}
```

Tvorba zakulacených rohů je podporována ve všech dnes používaných prohlížečích. Podpora v Internet Exploreru se objevuje až s verzí 9. U starších verzí prohlížečů se zakulacení rohů nezobrazí, což však nenarušuje samotnou funkčnost webu. Pokud není zásadním požadavkem dosažení zakulacení rohů u starších prohlížečů, je vhodné atribut `border-radius` využívat pro snížení datové velikosti stránky.

4.6 Průhlednost

Další novinkou v CSS3 je možnost nastavení alfa kanálu, který zajišťuje průhlednost jednotlivých elementů. Předchůdce tuto možnost vůbec nenabízí a vývojáři jsou nuceni využívat průhledné obrázky nastavené na pozadí elementů. Nejčastěji používaný formát je 32 bitový PNG soubor.

Názorná ukázka průhlednosti se nachází v galerii po najetí myší. Ve specifikaci XHTML1.1 je použit transparentní obrázek s 70% průhledností. To má za následek nutnost stáhnutí 2 různých obrázků pro dosažení žádaného efektu. Jiným problémem je samotná nutnost vytváření 2 různých fotografií.

CSS3 přináší alternativní řešení v podobě vlastnosti `opacity`. Ten umožňuje nastavení průhlednosti pomocí kaskádových stylů a o samotné zpracování se následně stará prohlížeč. Zápis průhlednosti vypadá následovně:

```
.galerie:hover img { opacity: 0.7; }
```

Průhlednost prostřednictvím kaskádových stylů je podporována ve většině dnešních internetových prohlížečů s výjimkou Apple Safari, kde se tato vlastnost vůbec nezobrazuje. Ostatní prohlížeče se zobrazením průhledných objektů nemají problém a dokonce Internet Explorer 8 (a nižší) podporuje alternativu v podobě používání filtru `filter: Alpha(opacity=50)`.

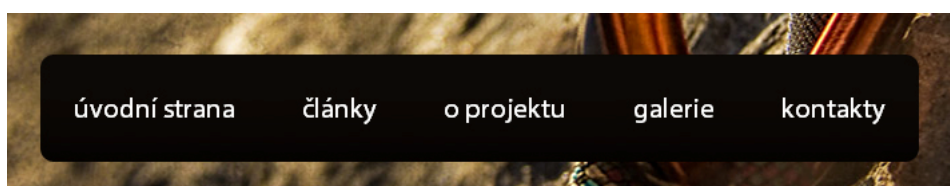
4.7 Pseudotřídy

Rozšíření se objevuje i u tzv. pseudotříd, které v původní specifikaci identifikovaly interakci uživatele se stránkou. To už dnes neplatí a možnosti těchto tříd jsou podstatně rozšířenější.

Při tvorbě webových prezentací je občas potřeba jednoznačně identifikovat některé elementy ve stránce podle jejich pořadí, případně umístění, ve struktuře dokumentu.

K identifikování jednotlivých objektů se velmi často využívá klasický element `class`, jak je tomu i v ukázkové prezentaci u prvku hlavního menu. Problémem může být generovaný obsah stránek například z databáze, kde programátoři musí vytvářet speciální algoritmy pro počítání vypsanych záznamů a potřebnou matematickou funkcí dopočítávat prvky s danou třídou.

To především komplikuje složitost a přehlednost zdrojového kódu a rovněž nutí vývojáře mnohonásobně krát používat potřebné třídy s přednastavenými styly.



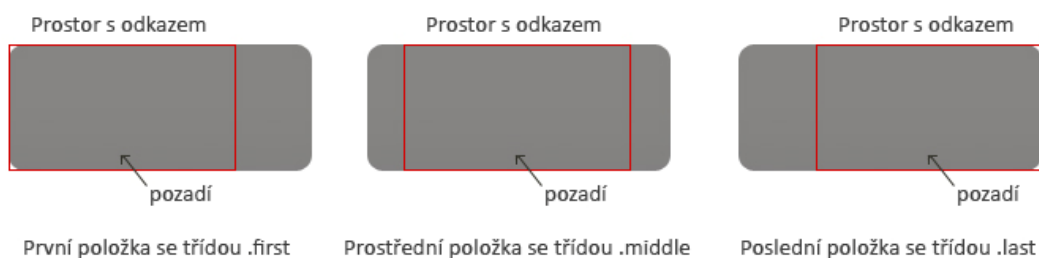
Obrázek 14 – Využití pseudotříd u hlavního menu

Záměrem práce je vytvořit oblast menu s položkami hlavního ovládacího panelu se zakulacenými rohy na počátku a konci znázorněné na obrázku výše.

Jednotlivé položky hlavního menu jsou rozděleny do 3 různých kategorií:

- 1) První položka se třídou `first`
- 2) Prostřední položky se třídou `middle`
- 3) Poslední položka se třídou `last`

Na pozadí všech hlavních odkazů je umístěn totožný obrázek s předkresleným pozadím se zakulacenými rohy, který je větší než zobrazovaná oblast. Základní myšlenkou je vložení unikátní pozice pozadí první a poslední položce, zatímco prostřední budou nezaoblené, tudíž umístěny na střed. Rozložení pozadí je znázorněno na následujícím obrázku.



Obrázek 15 - Náhled umístění pozadí v hlavním menu

Tyto třídy jsou nastaveny následovně:

```
#mainmenu a {
    display: block;
    line-height: 80px;
    height: 80px;
    background-image: url('mainmenu-bg.png');
    background-repeat: no-repeat;
}

#mainmenu a:hover {
    background-image: url('mainmenu-bg-hover.png');
}

#mainmenu a.first { background-position: left; }

#mainmenu a.middle { background-position: center; }

#mainmenu a.last { background-position: right; }
```

Při tvorbě obdobného prvku za pomoci HTML5 a CSS3 odpadá díky výše zmíněnému atributu `border-radius` varianta umístování obrázku na pozadí. Je stále potřeba pomocí identifikátoru jednoznačně určit první a poslední odkaz hlavního panelu. K tomu jsou v ukázce využity pseudotřídy `:first-of-type` pro výběr prvního

prvku daného typu a `:last-of-type` pro výběr posledního prvku daného typu. Takové kaskádové styly budou vypadat následovně:

```
#mainmenu li:first-of-type a {
    border-radius: 10px 0px 0px 10px;
}

#mainmenu li:last-of-type a {
    border-radius: 0px 10px 10px 0px;
}
```

Díky nim není potřeba umisťovat a upravovat speciální třídy `.first`, `.last` a `.middle` přímo do HTML kódu, ale rovnou tyto prvky nastavit pomocí kaskádových stylů 3. generace.

Podobný problém vzniká při tvorbě tabulek s podmínkou přehlednosti jednotlivých řádků. Záměrem bakalářské práce je vytvořit přehled jednotlivých obtížností lezeckých cest a přehledně je zanést do tabulky.

Země	UIAA 2	UIAA 3	UIAA 4	UIAA 5	UIAA 6	UIAA 7	UIAA 8	UIAA 9	UIAA 10
Česko	10	15	20	50	45	40	20	10	3
Slovensko	5	10	30	50	40	20	15	20	25
Německo	15	30	40	50	40	15	10	15	0
Rakousko	5	20	50	50	40	5	20	15	0

Obrázek 16 - Tabulka s odlišnou barevností řádků

Přehlednost v tabulce je zajištěna barevným rozlišením sudých a lichých řádků. Ve specifikaci xHTML1.1 a CSS2.1 neexistuje žádná funkce pro výběr konkrétních elementů. Vývojáři se musí uchýlit k využití jednoznačného označení konkrétních řádků prostřednictvím tříd, což vypadá následovně:

```
.lichy {
    background-color: #fde64a;
}

.sudy {
    background-color: #f6f3ce;
}
```


Nevýhodou tohoto zápisu je nutnost opakovaného psaní tříd `.sudy` a `.lichy` ke všem řádkům, které je potřeba obarvit. Tento nedostatek je velkým problémem zejména při vytváření velkých tabulek s množstvím dat.

Specifikace CSS3 přináší možnost používání pseudotříd, které slouží k identifikaci sudých a lichých potomků libovolného elementu, v tomto případě elementu řádky tabulky `<tr>`.

```
table tr:nth-child(odd) {
    background-color: #fde64a;
}

table tr:nth-child(even) {
    background-color: #f6f3ce;
}
```

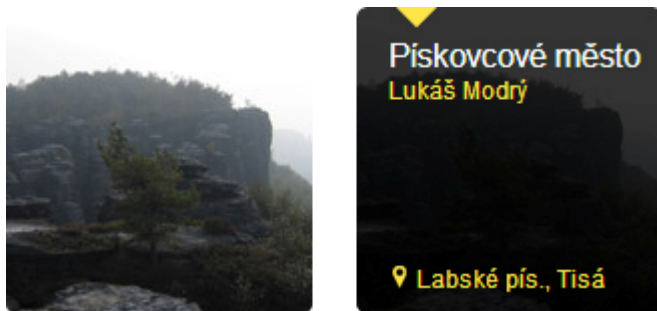
Pseudotřídy nabízejí mnoho dalších možností pro upřesnění výběru elementů. Ty jsou popsány v kapitole „*Nové pseudotřídy*“.

Tyto identifikátory jsou podporovány ve všech prohlížečích od dob Internet Exploreru 9, pro jednoduchost zdrojového kódu je tak vhodné je využívat. Hlavní výhodou je odstranění nutnosti využívání specifických tříd pro elementy, které se pravidelně opakují, nebo je snadné určit jejich pořadí pomocí jednoduché matematické funkce. V případě starších prohlížečů se rozčlenění prvků pomocí tříd neprojeví.

4.8 Mnohonásobné pozadí

Při tvorbě uživatelsky přívětivého prostředí je občas potřeba překrývat jeden obrázek jiným. Příkladem mohou být masky pro překrývání fotografií. Vývojáři, kteří tvoří stránky v xHTML1.1, musí často sáhnout po velmi nepohodlném řešení. Tím je obalení jednoho blokového elementu dalším se stejnými rozměry a na pozadí vkládání jednotlivé transparentní PNG obrázky.

Záměrem v samotné práci je vytvoření krycí masky při najetí na obrázek.



Obrázek 17 - Mnohonásobné pozadí

V praxi to vypadá následovně:

```
<a class="photo p1" href="cesta k souboru" title="Popisek souboru">
    <span class="photo-mask"></span>
    <span class="hover-mask"></span>
    <span class="jmeno">Pískovcové město</span>
    <span class="autor">Lukáš Modrý</span>
    <span class="misto">Labské pís., Tisá</span>
</span>
</a>
```

Základem je hypertextový odkaz identifikovaný třídou `.photo`, který je nastaven jako blokový a pomocí atributu *šířka* a *výška* zvětšen na potřebný rozměr. Do pozadí je vložen náhled konkrétní fotografie, která se zobrazí po kliknutí na odkaz.

```
.photo {
    display: block;
    width: 153px;
    height: 153px;
}

.p1 {
    background-image: url('../images/photo-nahled.jpg');
}
```

V ukázce jsou využity dvě různé masky pro překrytí obrázku. První z nich slouží k zakulacení rohů a nalézá se pod třídou `.photo-mask`. Její nastavení prostřednictvím kaskádových stylů je:

```

.photo-mask {
    display: block;
    width: 153px;
    height: 153px;
    background-image: url('photo-mask.png');
    background-repeat: no-repeat;
}

```

Další vrstvu tvoří `span` identifikovaný třídou `.hover-mask`. Tento element je v základu nastaven skrytý, po najetí na odkaz se však pomocí změny hodnoty atributu `display` zobrazí a s ním i krycí vrstva, která překrývá konkrétní fotografii s náhledem.

```

.photo:hover .hover-mask{
    display: block;
    width: 153px;
    height: 153px;
    background-image: url('hover-mask.png');
    background-repeat: no-repeat;
}

```

Specifikace HTML5 však přináší značné usnadnění v podobě mnohonásobného pozadí, které lze vrstvit přes sebe. Totožný efekt po najetí myši vypadá v HTML5 následovně:

```

.photo:hover{
    background-image:url('hover-mask.png'),
    url('../images/photo-nahled.jpg');
}

```

Zmizela vrstva, která zajišťovala zakulacení rohů fotografie a to díky atributu `border-radius`, který je nastaven již u definování základního vzhledu odkazu. O tom bylo psáno již dříve. Šlo by využít i totožnou variantu, jako v zastaralém xHTML1.1 a přidat další pozadí k atributu `background-image`.

```

.photo:hover{
    background-image:
    url('photo-mask.png'),
    url('hover-mask.png'),
    url('../images/photo-nahled.jpg');
}

```

Je nutné nezapomenout, že pseudotřída `:hover` zobrazí konkrétní nastavení stylů až po najetí myši, je tedy potřeba vlastnost zakulacených rohů nastavit i pro třídu `.photo`.

Mnohonásobné pozadí je podporované ve všech prohlížečích a nevzniká problém s jejich použitím při tvorbě webových prezentací. Rapidně také zjednodušuje strukturu stránek a zlepšuje přehlednost a složitost zdrojového kódu. Je velmi vhodné obracet se na variantu s mnohonásobným pozadím jednotlivých prvků. Jediný problém nastává v aplikacích Internet Explorer 8 a nižší.

4.9 Formátování textu

Základní informační část většiny prezentací tvoří texty. K usnadnění čitelnosti stránek je samozřejmě nutné dodržovat pravidla typografie. Tvůrci webových stránek se setkávají i s problémem viditelnosti a přehlednosti stránek.

4.9.1 Font-face

Součástí grafického návrhu mohou být i nestandardní rodiny fontů. Zde vzniká problém s jejich podporou na koncových zařízeních většiny uživatelů, které konkrétní typy písma nemají instalované ve svém počítači, a tudíž je nemohou zobrazit.

K tomu slouží direktiva `@font-face`, která byla implementována už v dřívější specifikaci CSS2.1. Zásadním problémem však při používání vlastnosti je rozdílná podpora formátů jednotlivých písem. Zatímco společnost Microsoft již v páté verzi svého Internet Exploreru využívá formát písma EOT (Embedded Open Type), ostatní výrobci webových prohlížečů využívají formáty TrueType a OpenType.

S používáním písem je třeba si uvědomit, že i na ně se vztahují autorská práva. Některé fonty nejsou volně k dispozici a je nutné si je koupit, případně dodržovat licenční ujednání s nimi spojená.

Funkce `@font-face` funguje na principu poskytnutí fontu ke stažení a webová stránka a její majitel se stává distributorem fontu. Je tedy nutné dbát zvýšené pozornosti na práva sdílení při používání písem na webových prezentacích.

Zápis pravidla pro vytvoření vlastní rodiny fontů je:

```
@font-face {
    font-family: 'mojepismo';
    src: url('../fonty/mojepismo.eot');
    src: url('../fonty/mojepismo.eot?#iefix')
    format('embedded-opentype'),
    url('../fonty/mojepismo.woff') format('woff'),
    url('../fonty/mojepismo.ttf') format('truetype'),
    url('../fonty/mojepismo.svg#signarita_zhairegular')
    format('svg');
}
```

Direktiva `@font-face` je díky implementaci v CSS2 podporována většinou internetových prohlížečů a není problém s jejím použitím.

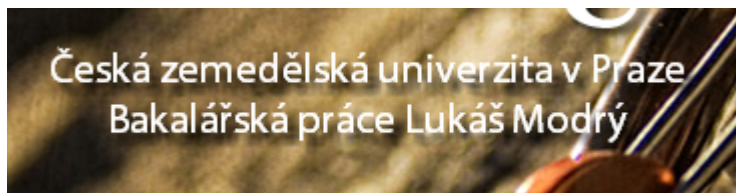
4.9.2 Stínování textu

Pro zvýraznění názvu stránky bylo potřeba vložit pod text stín. K tomu slouží nový atribut CSS3 s názvem `text-shadow`. Pomocí čtyř vlastností lze nastavit vertikální a horizontální posun vůči zdrojovému textu. Třetí hodnota značí míru rozostření stínu a čtvrtá hodnota je pro nastavení barvy stínu.

```
#header h1 {
    text-shadow: 5px 5px 5px black;
}
```

Stínování textu slouží k narušení jednodlosti webové stránky, která následně nepůsobí plošně. To usnadňuje i čitelnost textu, který tolik nesplývá s pozadím.

Takto nastavený text v hlavičce webové stránky vypadá následovně:



Obrázek 18 - Stínování textu

Stejným principem lze nastavit stín objektu pomocí vlastnosti `drop-shadow`.

Stínování textů a objektů je podporováno ve všech aktuálních verzích prohlížečů. V případě, že prohlížeč tyto atributy nepodporuje, je jednoduše nezobrazí.

Je tedy velmi vhodné používat vlastnosti `text-shadow` a `drop-shadow` v případě potřeby.

Vlastnosti pro vytváření stínování v CSS3 nevycházejí z žádného prvku v předchozí specifikaci. V CSS2.1 neexistuje žádná možnost pro vytváření stínů.

4.9.3 Transformování textu

Nové kaskádové styly nabízejí možnost transformování objektů pomocí vlastnosti `transform`, která je podporována ve všech prohlížečích.

```
transform: rotate(45deg);  
-ms-transform: rotate(45deg); /* IE 9 */  
-webkit-transform: rotate(45deg); /* Opera, Chrome a Safari */
```

Výše zmíněný kód má za následek otočení objektu o 45 stupňů.



Obrázek 19 - Transformace objektů

Nevýhodou je složitost přesného umístování transformovaného objektu na stránku. Při použití transformování objektů je problém se zachováním původních rozměrů a polohy, je tedy vhodné tyto elementy následně umístit na web pomocí absolutní pozice.

Z toho důvodu je vhodnější volit alternativní možnosti pro efekt transformovaných objektů, než vlastnost `transform`. I přestože je funkce pro transformaci podporována v nových prohlížečích.

Předchůdce CSS3 žádnou funkci pro transformaci objektů nenabízí a je třeba ji doplnit alternativami, jako použití JQuery. Případně používat obyčejné obrázky umístěné do obsahu stránky.

4.9.4 Více sloupcový layout

Pro usnadnění čitelnosti obsahu textů v článku, který je roztažený na celou šířku stránky, je cílem vytvoření dvousloupcového konceptu.

Ve specifikaci CSS2.1 neexistuje žádná přímá vlastnost pro rozdělení odstavce do více sloupců. Proto byla zvolena alternativa rozdělení textu pomocí vlastnosti obtékání a umístění dvou odstavců s rozděleným textem vedle sebe.

```
.columnLeft {
    width: 450px;
    float: left;
    border-right: 1px dashed #fde64a;
    padding-right: 20px;
}

.columnRight {
    width: 450px;
    float: right;
}
```

Nevýhodou je nutnost manuálního určení konce sloupce a nutnost vytvoření 2 speciálních identifikátorů pro specifické nastavení každého sloupce.

Vlastnost `column-count`, která je součástí specifikace CSS3, nabízí vyřešení tohoto problému v jednom řádku. V ukázce je dále nastavena mezera mezi sloupci `column-gap` a vzhled oddělení sloupců od sebe `column-rule`.

```
.column {
    -webkit-column-count: 2;
    -webkit-column-gap: 20px;
    -webkit-column-rule: 1px dashed #fde64a;
}
```

Výhodou je kompaktnost zdrojového kódu, dále není potřeba určovat konec jednoho sloupce a začátek dalšího. O to se stará prohlížeč.

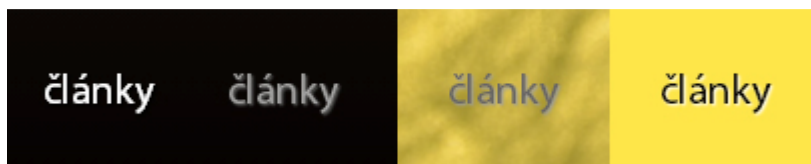
Rozdělení textů do sloupců je užitečným nástrojem. Z důvodu nedostatečné podpory v prohlížeči Mozilla Firefox a Internet Explorer je však nevhodné jej využívat a volit alternativní varianty.

4.10 Dynamické efekty

Moderní webové stránky mají množství grafických prvků, které slouží pouze k upoutání očí uživatele. Dříve se množství jednoduchých animací vytvářelo pomocí knihovny JQuery, popř. zásuvného modulu Flash. Používání těchto možností vyžaduje

vyšší znalost programování v JavaScriptu, případně vytváření animací v aplikaci Adobe Flash. Vytváření těchto alternativ není předmětem této bakalářské práce.

CSS3 přináší možnost vytváření jednoduchých animací přímo prostřednictvím nastavení vzhledu elementů. Jednou z nich je prolínání mezi dvěma styly pomocí vlastnosti `transition`, která byla použita po najetí na položku v hlavním menu.



Obrázek 20 - Prolínání po najetí myši

```
#mainmenu a {  
    -webkit-transition: all 1s; /* Saf3.2+, Chrome */  
    -moz-transition: all 1s; /* FF4+ */  
    -ms-transition: all 1s; /* IE10 */  
    -o-transition: all 1s; /* Opera 10.5+ */  
    transition: all 1s;  
}
```

Podpora efektů v kaskádových stylech je podporována ve všech prohlížečích s nutností použití jednotlivých vykreslovacích jader. V případě starších prohlížečů, které vlastnost nepodporují, se efekt pouze nezobrazí, funkčnost stránek však zůstane nepozměněna. Není tedy problém s používáním těchto efektů.

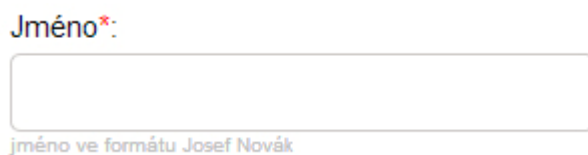
4.11 Využití formulářových prvků

Většina interakce člověka s webovou stránkou je založena na používání webových formulářů, pomocí kterých uživatelé mohou zasílat serveru potřebné informace. Často je nutné uživatelem vložená data filtrovat.

Při používání formulářů v xHTML1.1 je třeba využívat některý z objektově orientovaných jazyků, jako například PHP nebo JavaScript, který je použit v této ukázce.

Pro ulehčení práce s formulářem je zde umístěna nápověda s formátem požadovaného textu. Často se totiž uživatelé setkávají s otázkou, v jakém tvaru je po nich požadováno vyplnění těchto polí.

Ve verzi xHTML1.1 je na ukázkovém webu použit jednoduchý element `` umístěný pod vkládacím polem, ve kterém je textová nápověda k danému prvku.



Obrázek 21 - Popisek formulářového pole

Ve zdrojovém kódu stránky prvek vypadá následovně:

```
<label>Jméno<span>*</span>:</label>
<input type="text" name="name" />
<span class="napoveda">jméno ve formátu Josef Novák</span>
```

Často se na internetu objevují i mnohé skripty, které vkládají nápovědu přímo dovnitř pole jako hodnotu *value*. Nevýhodou takové varianty je nutnost zapnutých skriptů, které zajišťují funkčnost. Rovněž narůstá složitost zdrojového kódu a jeho náročnost na načítání. Z důvodů minimalizace takovýchto skriptů nebyla tato varianta v samotné práci zvolena.

HTML5 přináší nový mocný nástroj s názvem *placeholder*. Tento atribut formulářového elementu funguje na totožném principu, jako výše zmiňovaný skript. Je součástí samotné specifikace a jeho funkci zajišťuje samotný webový prohlížeč, nikoli dodatečný modul naprogramovaný v jiném jazyce.



Obrázek 22 - Atribut Placeholder

Zdrojový kód a samotné formátování stránky prostřednictvím kaskádových stylů je díky HTML5 o poznání jednodušší a přehlednější:

```
<label>Jméno<span>*</span>:</label>
<input type="text" name="name" placeholder="Josef Novák">
```

Atribut *placeholder* je plně podporovaný ve všech posledních verzích nejčastěji používaných internetových prohlížečů⁷. V případě starších aplikací, které atribut nepodporují, ho nezobrazí, ale funkčnost samotného formuláře zůstane nepozměněna,

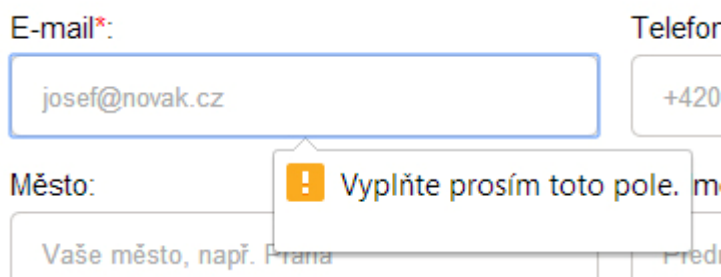
⁷ Internet Explorer 11, Opera 19, Mozilla Firefox 27, Apple Safari 5, Google Chrome 32

tudíž je vhodné jej používat na nově vytvářených webových stránkách psaných v HTML5.

Požadavkem zpracování kontaktního formuláře je ověřit povinné údaje, které uživatel musí zadat pro úspěšné odeslání požadavku. Podmínkou kontrolující tyto údaje, psanou v JavaScriptu, je zajištění kontroly vložených dat v povinných formulářových polích:

```
if(document.kontakt.name.value==""){
window.alert("Pole jméno je povinné!");
return false;
} else if(document.kontakt.mail.value=="") {
window.alert("Pole e-mail je povinné!");
return false;
} else if(document.kontakt.subject.value=="") {
window.alert("Pole předmět je povinné!");
return false;
} else if(document.kontakt.message.value=="") {
window.alert("Pole zpráva je povinné!");
return false;
}
}
```

Výhodou HTML5 je atribut `required`, který ověří naplnění povinných polí již při pokusu o odeslání formuláře.

The image shows a portion of a web form. At the top, there are two input fields: 'E-mail*' containing 'josef@novak.cz' and 'Telefor' containing '+420'. Below the 'E-mail*' field, there is a red error message box with a white exclamation mark icon and the text 'Vyplňte prosím toto pole. m'. Below the error message, there is a 'Město:' label and an input field containing 'Vaše město, např. Praha'. The error message is positioned over the 'Město:' label and its input field.

Obrázek 23 - Chybové hlášení HTML5 s atributem `required`

Dále při vytváření formuláře bylo potřeba ověřit, zdali zadaná e-mailová adresa a URL adresa uživatele je platná a napsána ve správném formátu.

Norma emailové adresy je popsána v dokumentu RFC 822. Ten tvar emailové adresy popisuje jako posloupnost povolených znaků před znakem @, které mezi sebou

mohou být odděleny tečkou, kterou však nesmí začínat. Za znakem @ se nachází název domény, kde je emailová adresa hostována. (Pelletier, Nedatováno)

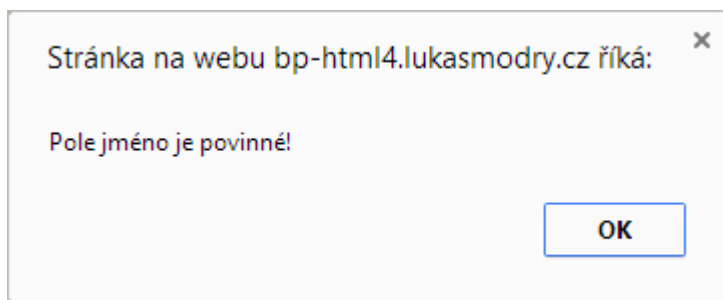
O rozšíření jazyka JavaScript o regulární výrazy se zmiňuje Miloslav Ponkrác ve svém článku o kontrole emailové adresy pomocí JS na serveru interval.cz následovně: „Regulární výrazy byly do JavaScriptu přidány ve verzi 1.2 a to jako objekt typu *RegExp* a také jako metoda *search*, u které je možné kontrolovat, zda text vyhovuje regulárnímu výrazu.“ (Ponkrác, 2000)

K ověření platnosti těchto adres zadané v kontaktním formuláři je ve verzi XHTML1.1 použit právě JavaScript.

Za pomoci podmínky se složitými regulárními výrazy to vypadá následovně:

```
if (regmail.test(document.kontakt.mail.value)==false) {  
window.alert("E-mailová adresa je ve špatném formátu");  
return false;  
} else if(regurl.test(document.kontakt.web.value)==false  
&& document.kontakt.web.value!="") {  
window.alert("URL adresa je ve špatném formátu");  
return false;  
}
```

Samotné chybové hlášení informující uživatele o nastalém problému je vyjádřeno prostřednictvím vyskakovacího okna, které poskytuje samotný prohlížeč.



Obrázek 24 - Chybové hlášení pomocí JS alert

Vytváření regulárních výrazů však svádí k vytváření chyb v zápisu, a proto HTML5 přináší novinku v podobě automatického ověřování těchto polí pomocí nastavení typu formulářových polí, konkrétně na hodnotu `email` a `url`. Ověření zajistí samotný prohlížeč.

E-mail*: lukas(a)email.cz

Telefon: +420 789 123 4

Město: Vaše město, např. Praha

! Do e-mailové adresy zahrňte znak @. V adrese lukas(a)email.cz chybí znak @.

Obrázek 25 - Chybové hlášení pro ověření platnosti e-mailové adresy

O kontrolu formátu vložených dat se stará samotný prohlížeč stránek a odpadá nutnost tvorby složitých regulárních výrazů za pomoci JavaSkriptu, případně PHP, což je mocným nástrojem formulářů tvořených v HTML5.

Zásadní problém při využívání těchto polí však nastává s podporou internetových prohlížečů. Většina posledních verzí atribut *required* a nové formulářové prvky pro zadávání emailu a URL adresy podporují, výrobci aplikace Opera, která má velké zastoupení na straně uživatelů, však ve své poslední verzi podporu těchto novinek nemá.

Pole input nastavené jako typ *email* nebo typ *url* se v prohlížeči Opera jeví jako klasické pole pro vkládání textu, což zajišťuje funkci formuláře jako takového, ale neprojeví se žádné ověření vložených dat. Jedinou možností je následné vytvoření alternativy pro ověření těchto hodnot pomocí regulárních výrazů, což vede zpět k problému se složitostí jejich vytváření.

Vývojáři webu mohou bez obav tyto nové hodnoty atributů na svých webech využívat s kombinací alternativního ověření dat před odesláním samotného formuláře na server například za pomoci PHP nebo JavaScriptu.

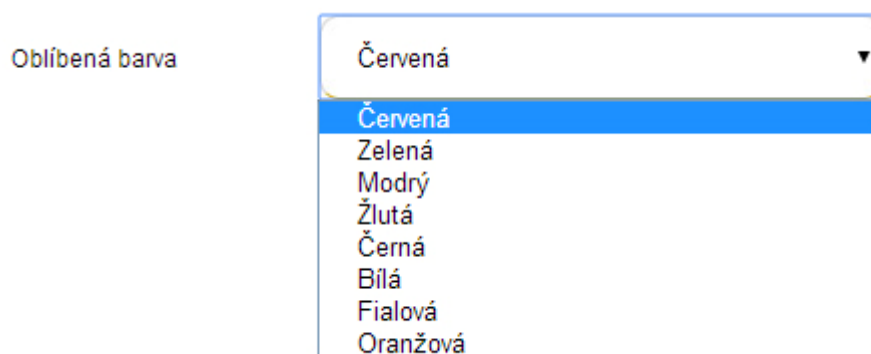
Stejně tak u několika aktualizací prohlížečů (Firefox, Internet Explorer) není implementována podpora pro typ *number*, který se rovněž jeví jako prosté textové pole. Jeho podpora je nedostatečná a vytvoření podmínky na kontrolu číselné hodnoty není nijak složité, tudíž je lepší se tomuto typu prozatím úplně vyvarovat.

Formuláře v HTML5 přinášejí řadu nových možností pro interakci člověka s počítačem. Tyto elementy jsou stále ještě ve vývojové fázi a jejich podpora v prohlížečích je velmi nedostatečná.

Mezi nové možnosti patří formulářový typ pro výběr barvy z předdefinované palety *color*. Po kliknutí na pole barvy se uživateli objeví nabídka palety barev, kterou již zná z aplikací jako malování. Náhled samotného prvku je k dispozici v kapitole „*Přínosy HTML5*“.

Podpora palety barev je velmi nedostatečná a plnou podporu dostává pouze v posledních verzích aplikací Google Chrome a Opera. Z toho vyplývá, že je velmi nevhodné tyto prvky v dnešní době aplikovat do webových prezentací a naopak volit alternativní varianty pro výběr barvy. Jednou z možností je nabídnout uživateli předepsanou paletu barev ve formulářovém prvku `<select>`, která je součástí specifikace HTML již delší dobu nebo zvolit možnosti knihoven JQuery.

V ukázkové variantě webu psaném v xHTML1.1 je použita právě varianta s elementem `select`, kde jsou uživatelé omezeni na pár předdefinovaných barev, nikoli celou paletu RGB.



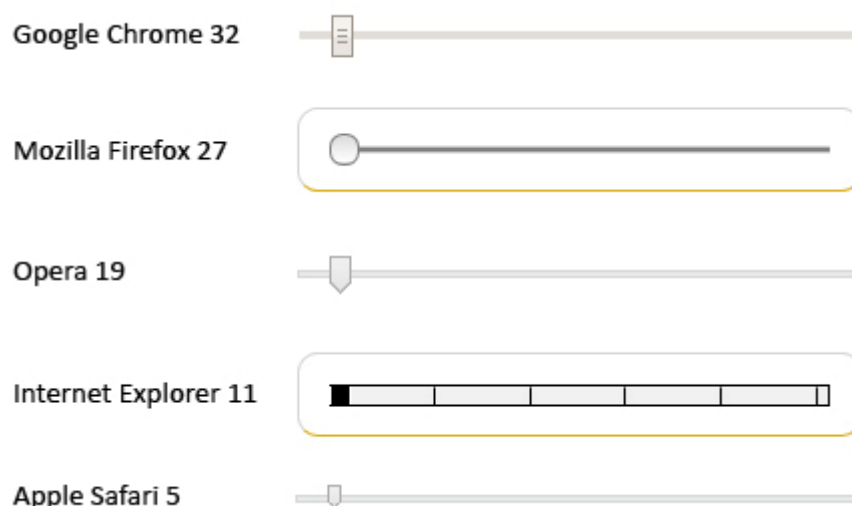
Obrázek 26 - Select pro výběr barvy

Stejně omezenou podporu u prohlížečů má i nový prvek pro výběr a vkládání data a času `datetime`, který je rovněž popsán v kapitole „*Přínosy HTML5*“.

Další částí formuláře je prvek pro výběr určité hodnoty z předepsaného rozsahu. xHTML1.1 neobsahuje žádnou možnost pro výběr z daného rozsahu, proto je zvoleno pouze pole pro vložení textu.

Oproti tomu HTML5 nabízí formulářový prvek `<range>`. Ten vloží do stránky přednastavený jezdec, pomocí kterého uživatelé mohou zvolit konkrétní hodnotu.

Používání prvku je podporováno ve všech webových prohlížečích. Jediným rozdílem je základní vzhled jezdců, který se v jednotlivých prohlížečích mění.



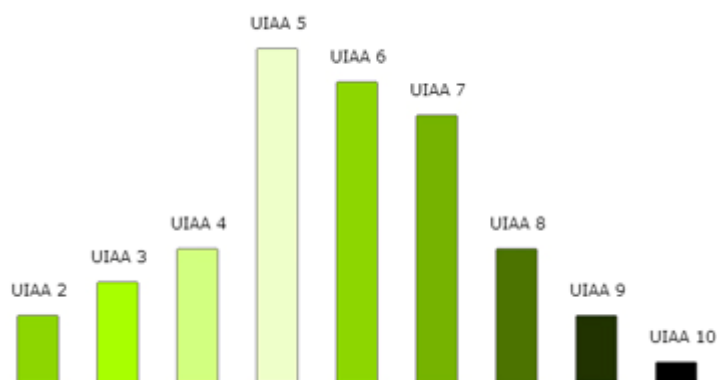
Obrázek 27 - Formulářový element Range

Dalším užitečným nástrojem, který se nenachází v předchozí specifikaci HTML, je element `<meter>` pro určení stavu zpracování nějaké položky. V ukázce se jedná o naznačení kroků při vyplňování přihláškového formuláře.

Tento prvek stále není podporován v posledních verzích prohlížečů Internet Explorer 11 a Safari 5, tudíž je nevhodné jej na stránkách využívat a raději použít alternativní možnosti, například jazyku JavaScript, případně manuální vkládání obrázků do stránky.

4.12 Grafika za pomoci elementu Canvas

Cílem ukázky použití grafiky na webu je vytvoření grafu četnosti lezeckých cest podle jejich obtížnosti. V době XHTML1.1 neexistovala žádná konkrétní funkce jazyka HTML pro kreslení objektů na stránce a vývojáři tak byli omezeni pouze na vkládání obrázků, případně využívání zásuvných modulů. V ukázce je využit právě obrázek vytvořený za pomoci grafického softwaru.



Obrázek 28 - Názorná ukázka grafu

Tento graf je do článku vložen pomocí klasického elementu pro vkládání obrázků a zdrojový kód vypadá následovně:

```

```

Velmi spekulovaným a mocným nástrojem HTML5 je bezpochyby element canvas, který slouží pro kreslení vektorové 2D grafiky generované přímo prohlížečem stránek za pomoci JavaScriptu.

Totožný graf lze s využitím plátna nakreslit přímo pomocí zdrojového kódu stránky a není potřeba žádné aplikace určené pro grafiku. V ukázce zdrojového kódu je jen první sloupec grafu s nadpisem UIAA 2.

```
window.onload = function() {
    var platno = document.getElementById("graf");
    var context = platno.getContext("2d");
    context.beginPath();
    context.rect(30, 250, 30, 50);
    context.lineWidth = 1;
    context.strokeStyle = "#000000";
    context.stroke();
    context.fillStyle = "#8dd700";
    context.fill();
    context.font="12px Verdana";
    context.fillText("UIAA 2", 25, 235);
}
```

Kreslení objektů na plátno využívá X a Y osu plátna, díky které je možné se po něm orientovat a vytvářet jednotlivé obrazce. Výhodou plátna je možnost změny

jednotlivých částí grafiky bez nutnosti využití grafické aplikace a nahrávání aktualizovaných obrázků na server.

Vývojáři webových aplikací disponují mocným nástrojem pro generování interaktivních a automaticky se aktualizujících grafických prvků za pomoci JavaScriptu.

Plátno je podporováno ve všech posledních verzích prohlížečů⁸ a není tak nutné se použití elementu `<canvas>` obávat.

4.13 Použití multimédií

Součástí článků je možnost vložení zvukové stopy nebo video souboru. Ve verzi psané v XHTML1.1 je nutné spoléhat se na zásuvné moduly (většinou Flash Player), které nabízejí možnost přehrávání videa. Nejvhodnější variantou je vložení video souborů prostřednictvím služeb třetí strany.

V ukázkovém článku jsou k tomuto účelu použity služby sítě YouTube⁹, kde je uloženo video určené ke sdílení.

Specifikace HTML5 přináší novou funkci pro přehrávání video souborů prostřednictvím webových aplikací. Tím je element `<video>`.

```
<video width="320" height="240" controls>
  <source src="video/video.mp4" type="video/mp4">
</video>
```

Nevýhodou tohoto elementu je nutnost uložení videa na serveru, který má omezenou datovou kapacitu. Pomocí úplné adresy je možné využívat i cizí datová úložiště. Mimo jiné jsou zde kladeny vyšší nároky na hardware serveru.

Dále podpora elementu pro přehrávání video souborů je odlišná v jednotlivých prohlížečích. Zásadní rozdíl je v podpoře jednotlivých video formátů.

⁸ Internet Explorer 11, Opera 19, Mozilla Firefox 27, Apple Safari 5, Google Chrome 32

⁹ www.youtube.com

Prohlížeč	MP4 (H264)	WebM (VP8)	Ogg (Theora)
Internet Explorer	Ano	Ne	Ne
Google Chrome	Ano	Ano	Ano
Mozilla Firefox	Ano, omezeně	Ano	Ano
Apple Safari	Ano	Ne	Ne
Opera	Ne	Ano	Ano

Tabulka 1 - Podpora video formátů v prohlížečích

Obdobně je na tom element pro vkládání zvukových souborů `<audio>`. Rovněž je problém s podporou formátů zvukových dat pod jednotlivými prohlížeči.

Prohlížeč	MP3	WAV	Ogg
Internet Explorer	Ano	Ne	Ne
Google Chrome	Ano	Ano	Ano
Mozilla Firefox	Ano, omezeně	Ano	Ano
Apple Safari	Ano	Ano	Ne
Opera	Ne	Ano	Ano

Tabulka 2 - Podpora zvukových formátů v prohlížečích

Z důvodu nejednotné podpory přehrávaných formátů je prozatím vhodné spoléhat se na služby třetích stran. Ty zajišťují totožné zobrazení ve všech prohlížečích na všech platformách. Dále jsou sníženy náklady spojené s hardwarovými nároky. Služby třetích stran poskytují stabilní prostředí pro sdílení video souborů prostřednictvím internetu.

5 Výsledky a jejich zhodnocení

Uživatelské nároky na podobu a funkci webových stránek jsou velké. Vývojáři se potýkají s nepřehledným množstvím problémů k řešení při tvorbě efektivní internetové prezentace. Díky návrhům HTML5 a CSS3, které doplňují předchozí specifikace o nové funkce, je realizace těchto úkolů jednodušší.

S vývojem jazyka se vyvíjejí i internetové prohlížeče. S tím je spojena i snaha o získání majoritní pozice na poli těchto aplikací, díky čemuž je velká většina nových funkcí podporována v posledních aktualizacích jednotlivých prohlížečů.

5.1 Analýza webů vytvořených v praktické části

Po celkové analýze a porovnání obou vytvářených webových prezentací bylo zjištěno, že stránky vytvářené prostřednictvím specifikace xHTML1.1 jsou o více, než 6% dat větší, než jejich alternativa psaná v HTML5.

Základní rozdíl v datové velikosti stránek vytváří nutnost využívání množství obrázků ve staré specifikaci, které jsou potřebné pro vytvoření jednotlivých prvků a efektů. Ty jsou v HTML5 a CSS3 nahrazeny automatickými funkcemi. Do celkové velikosti stránek však není zahrnuta velikost vloženého video souboru v nové specifikaci, který je na starém webu aplikován prostřednictvím služby třetí strany.

Další úsporou kombinace HTML5 a CSS3 je množství znaků obsažených ve zdrojovém kódu. Zde je úspora ve prospěch nových technologií rovněž 6%. Součástí tohoto výpočtu není zdrojový kód nutný pro definování grafu pomocí elementu canvas, jehož náročnost na zdrojový kód je o poznání větší. Není však součástí všech webů a jeho názorné použití by zkreslovalo samotné výsledky.

Specifikace	Datová velikost	Počet obrázků	Počet znaků
xHTML1.1 CSS2.1	6,30 MB	77	42 779
HTML5 CSS3	5,93 MB (20,63 MB s videem)	55	40 051 (42 965 s elementem Canvas)

Tabulka 3 - Velikost specifikací

Další částí analýzy stránek je zhodnocení rychlosti načítání jednotlivých stránek. Zde vychází opět výhodněji a rychleji nová technologie oproti zastaralému předchůdci. Prostřednictvím webu Pingdom¹⁰ byly zjištěny následující hodnoty:

Název stránky	xHTML1.1+CSS2.1	HTML5+CSS3
Úvodní strana (index)	542,2 kB za 1,74 s	439,9 kB za 1,34 s
Články	422,1 kB za 1,86 s	304,2 kB za 1,44 s
Články 1 – Transformace, Sloupce	732,0 kB za 1,70 s	608,3 kB za 1,24 s
Článek 2 – Canvas	400,7 kB za 438 ms	248,3 kB za 313 ms
Článek 3 – Video	755,5 kB za 1,22 s	13,65 MB za 3,8 s
Galerie	863,2 kB za 1,94 s	778,8 kB za 408 ms
Kontakty	451,2 kB za 2,25 s	359,3 kB za 1,35 s

Tabulka 4 - Rychlost načítání stránek prostřednictvím webu Pingdom

Pomocí tohoto testu bylo zjištěno, že stránky psané v HTML5 se načítají podstatně rychleji, především z důvodu menší náročnosti na stahování podkladových obrázků. Součástí zjištění je i výpočet náročnosti jednotlivých prvků každé stránky na samotné načtení. Průměrně nejvíce času k zobrazení potřebují samotné obrázky. Poměr těchto dat se v porovnání se zbytkem pohybuje kolem 90% u specifikace xHTML1.1 a 70% u specifikaci HTML5. Malý rozdíl je i v množství časové náročnosti na zobrazení HTML značek použitých ve zdrojovém kódu, to především u specifikace HTML5. Zde se poměr vůči zbytku pohybuje kolem 20% celkového času nutného k zobrazení dané stránky.

5.2 Přínosy realizace stránek pomocí HTML5 a CSS3

Největší výhodou využívání nových specifikací je snížení prostorové velikosti zdrojového kódu, který je podstatně čitelnější. S tím je spojena i složitost jednotlivých návrhů řešení určitých problémů při realizaci stránek díky snížení počtu identifikátorů

¹⁰ <http://tools.pingdom.com>

a množství nových vlastností a funkcí v kaskádových stylech. Stránky psané v HTML5 a CSS3 jsou snáze upravovatelné.

Z ekonomického hlediska je tvorba a úprava webových stránek rychlejší a snazší. Díky tomu dochází k úspoře výdajů spojených s lidským faktorem, popř. mají vývojáři více času a prostoru na korektní definice jednotlivých částí stránky. Nové vlastnosti kaskádových stylů snižují datovou náročnost webových stránek, díky tomu dochází k dalším možným úsporám na výdajích spojených s provozem webových stránek.

Nové specifikace jsou obohaceny o množství nových funkcí, které v dřívější verzi nebyly k dispozici. Jazyk HTML5 je obohacen o množství sémanticky významných elementů a atributů, díky čemuž jsou stránky čitelnější. To by v budoucnosti mohlo znamenat korektnější výsledky hledání prostřednictvím internetových vyhledávačů. Sémanticky popsaný obsah stránek zároveň usnadňuje čitelnost obsahu zrakově postiženým uživatelům využívajících čtečky obrazovky.

Dalším přínosem jazyka HTML5 je minimalizace potřeby využívání zásuvných modulů. Uživatelé už nejsou nuceni ke stahování množství plug-inů pro správnou funkčnost webových stránek. Stačí jim pouze aktualizovaný webový prohlížeč. Internetové prohlížeče množství prvků nyní zpracovávají nativně.

Nespornou výhodou specifikací HTML5 a CSS3 je její původ ve starších verzích, díky tomu weboví vývojáři dovedou využívat nové funkce bez nutnosti přeškolení a studia množství odborných zdrojů. Přechod na nové verze je tak ekonomicky nenáročný.

Celkovým přínosem při plné podpoře specifikací HTML5 a CSS3 je možnost tvorby efektivních webových stránek s jednotnou podporou v prohlížečích a na všech platformách. Za pomoci nových funkcí je umožněno vytvářet a prohlížet internet snadněji a rychleji. Internetové prezentace postavené na HTML5 jsou přístupné zrakově postiženým uživatelům.

5.3 Nevýhody realizace stránek v HTML5 a CSS3

V dnešní době jsou nové specifikace stále ve fázi vývoje. Jednotlivé části těchto návrhů jsou v odlišných stádiích vývoje.

Přestože se tvůrci webových prohlížečů předhánějí v množství podporovaných značek HTML5 a vlastností CSS3, zpětná podpora ve starších verzích těchto prohlížečů je nereálná. Největším problémem na poli starých internetových prohlížečů je Internet Explorer 8, jenž je nejnovější variantou prohlížeče podporovaného na operačním systému Windows XP. Množství uživatelů tohoto prohlížeče je stále velmi velké. V době vydání Internet Exploreru 8 se o specifikacích HTML5 a CSS3 ještě mnoho nevědělo.

Množství nových funkcí HTML5 je úzce spjato s používáním JavaScriptu. To má za následek nutnost větší odbornosti při vývoji složitějších webových aplikací vytvářených v HTML5 a pozice „*Kodér webových stránek*“ a „*Programátor webových aplikací*“ se stávají jedním a tím samým člověkem.

5.4 Podpora v nových verzích prohlížečů

Většina standardně využívaných elementů a vlastností je již v dnešní době plně podporována ve všech tradičně používaných prohlížečích internetových stránek. Nejlepší podpory dosahuje prohlížeč Google Chrome 32, naopak největší nedostatky při zobrazování webových prezentací psaných za pomoci nových specifikací má Apple Safari 5.

6 Závěr

V dnešní době, kdy jsou návrhy nových specifikací stále ve stádiu vývoje a podpora nových prvků je stále nekompletní, je vhodné využívání těchto specifikací v omezené míře. Vhodné je využívání jen těch prvků, které nemají vliv na vzhled a funkčnost webových stránek. U ostatních je třeba vždy vytvářet alternativní obsah, který nahradí nefunkční prvky stránek. S tím je spojena i větší ekonomická náročnost na tvorbu takovýchto prezentací.

Jednotlivé specifikace byly vzájemně porovnány. Webová prezentace psaná v jazyce XHTML1.1 a CSS2.1 je prostorově mnohem náročnější a zdrojový kód stránek je nepřehledný a složitý.

Naopak kompaktnost a efektivnost zdrojového kódu specifikací HTML5 a CSS3 je na vysoké úrovni. Díky tomu je takto napsaná stránka mnohem čitelnější jak pro vývojáře, tak pro roboty internetových vyhledávačů popř. pro čtečky obrazovky. Jsou však kladeny vyšší nároky na využívání JavaScriptu pro vytváření interaktivních aplikací.

Jazyk HTML5 je ve vývoji už delší dobu, což má za následek ustálení valné většiny nových elementů a funkcí. V případě možnosti ověření softwarového vybavení cílového uživatele je vhodné tyto prvky využívat, nutné je však poskytnout alternativu ostatním uživatelům, je-li cílovou skupinou webové prezentace široká veřejnost.

Množství funkcí specifikací HTML5 a CSS3 je zpětně kompatibilní. Především v případě kompaktnějšího zápisu některých vlastností. V případě nedostatečné podpory se tyto prvky zobrazí jako podobný prvek ze staré specifikace (to se týká především formulářové elementy), popř. se nezobrazí vůbec, což ale nemá vliv na funkčnost webové stránky.

Seznam zkratek

WWW – World Wide Web

W3C – World Wide Web Consortium

HTML – Hyper Text Mark-up Language

xHTML – Extensible Hyper Text Markup Language

XML – Extensible Markup Language

CSS – Cascading Style Sheets (Kaskádové styly)

HTTP – Hyper Text Transfer Protokol

URL – Unique Resource Locator

CERN – Conseil Européen pour la recherche nucléaire (Evropská organizace pro jaderný výzkum)

DTD – Document Type Declaration

ISO – International organization for Standardization

UTF – UCS Transformation Format

SEO – Search Engine Optimization

IE – Internet Explorer

FTP – File Transfer Protocol

WYSIWYG – What you see is what you get

RGB – barevné schéma Red, Green, Blue

Seznam použitých zdrojů

Albers, Brian, Salim, Frank a Lubbers, Peter. 2010. *HTML5 - Programujeme moderní webové aplikace.* Brno : Computer Press, 2010. ISBN 978-80-251-3539-6.

Castro, Elizabeth a Hyslop, Bruce. 2012. *HTML5 a CSS3 - názorný průvodce tvorbou WWW stránek.* Brno : Computer Press, 2012. ISBN 978-80-251-3733-8.

Clark, Richard, a další. 2012. *Beginning HTML5 and CSS3.* New York : Apress, 2012. ISBN 978-1-4302-2874-5.

Connolly, Dan. 2011. A Little History of the World Wide Web. W3. [Online] W3, 10. 7 2011. [Citace: 12. 2 2014.] <http://www.w3.org/History.html>.

Cooper, Marcel. 2010. Error 404 a další chyby, co číhají na webu. *Živě.cz.* [Online] Mladá fronta a. s., 10. 8 2010. [Citace: 20. 12 2013.] <http://www.zive.cz/clanky/error-404-a-dalsi-chyby-co-ctihaji-na-webu/sc-3-a-153345/default.aspx>.

Dudek, Jan. 2002. CSS2 selektory, pseudotřídy a pseudoelementy. *Interval.cz.* [Online] ZONER software, a.s., 12. 12 2002. [Citace: 15. 1 2014.] <http://interval.cz/clanky/css2-selektory-pseudotridy-a-pseudoelementy/>.

Foundation, World Wide Web. 2014. Web Inventor and Founding Director of the World Wide Web Foundation. *World Wide Web Foundation.* [Online] World Wide Web Foundation, 2014. [Citace: 20. 12 2014.] <http://www.webfoundation.org/>.

Friendly, Blind. 2014. Blind Friendly. *Blind Friendly.* [Online] TyfloCentrum Brno, o. p. s. a SONS ČR, 2014. [Citace: 2013. 12 12.] <http://www.blindfriendly.cz>.

Google. 2014. Stavové kódy HTTP. *Google.* [Online] Google Inc., 2014. [Citace: 2. 12 2014.] <https://support.google.com/webmasters/answer/40132?hl=cs>.

Greguš, Mojmír. 2013. Segmentace webové stránky a sémantika. *H1.cz s.r.o.* [Online] H1.cz s. r. o. a QUISMA Company, 29. 3 2013. [Citace: 18. 2 2014.] <http://blog.h1.cz/aktualne/html5-a-seo/>.

Hickson, Ian. 2014. Web Hypertext Application Technology Working Group. *whatwg.org*. [Online] whatwg, 2014. [Citace: 17. 2 2014.] www.whatwg.org.

— . **2009.** Web socket protocol in "last call"? (E-mail archive). *The Internet Engineering Task Force*. [Online] 30. 10 2009. [Citace: 1. 12 2013.] <http://www.ietf.org/mail-archive/web/hybi/current/msg00784.html>.

Hogan, Brian P. 2012. *HTML a CSS 3 - Výukový kurz webového vývojáře*. Brno : computer press, 2012. ISBN 978-80-251-3576-1.

Janovský, Dušan. nedatováno. Historie CSS. *Jak psát web*. [Online] nedatováno. [Citace: 10. 11 2013.] <http://www.jakpsatweb.cz/css/css-historie.html>.

— . **nedatováno.** HTTP Protokol. *Jak psát web*. [Online] nedatováno. [Citace: 15. 12 2013.] <http://www.jakpsatweb.cz/server/http-protokol.html>.

Kosek, Jirka. 2013. HTML Guru. *HTML Guru*. [Online] 2013. [Citace: 10. 10 2013.] <http://htmlguru.cz/>.

Kubíček, Michal. 2010. *Velký průvodce SEO - Jak dosáhnout nejlepších pozic ve vyhledávačích*. Brno : Computer Press, 2010. ISBN 978-80-251-2195-5.

Longman, Addison Wesley. 1998. A history of HTML. W3C. [Online] W3, 1998. <http://www.w3.org/People/Raggett/book4/ch02.html>.

Lubbers, Peter, Albers, Brian a Salim, Frank. 2011. *HTML 5 - Programujeme moderní webové aplikace*. Brno : computer press, 2011. ISBN 978-80-251-3539-6.

Malý, Martin. 2009. Web Sockets. *Zdroják.cz*. [Online] Devel.cz Lab s.r.o., 14. 12 2009. [Citace: 12. 2 2014.] <http://www.zdrojak.cz/clanky/web-sockets/>. ISSN 1803-5620.

Meuller, John. 2012. Forum Google. *Forum Google*. [Online] 21. 3 2012. [Citace: 2. 18 2014.] [https://productforums.google.com/forum/#!searchin/webmasters/authorname\\$3Ajohnmu%7Csort:date/webmasters/ihmDyu5ykr/Kq0VngOh71oJ](https://productforums.google.com/forum/#!searchin/webmasters/authorname$3Ajohnmu%7Csort:date/webmasters/ihmDyu5ykr/Kq0VngOh71oJ).

Míkula, Jan. 2012. HTML5: Doctype. *Prohlížeče*. [Online] 2012. [Citace: 1. 1 2014.] <http://prohlizece.info/clanky/html5-doctype/>. ISSN 1802-3584.

Pavlíček, Mgr. Radek. 2005. Metodika Blind Friendly Web 2.3. *blindfriendly.cz*. [Online] TyfloCentrum Brno, o. p. s. a SONS ČR, 31. 3 2005. [Citace: 30. 1 2014.] <http://blindfriendly.cz/metodika>.

Pelletier, Ray. Nedatováno. RFC 822. *Request for Comments*. [Online] Association Management Solutions, Nedatováno. [Citace: 18. 2 2014.] <http://tools.ietf.org/html/rfc822>.

Peters, Simon a Kesteren, Anne van. 2014. Differences HTML4 and HTML5. *Differences HTML4 and HTML5*. [Online] 2014. [Citace: 30. 12 2013.] <http://rawgithub.com/whatwg/html-differences/master/Overview.html>.

Pilgrim, Mark. 2014. Dive into HTML5. *Dive into HTML5*. [Online] 2014. [Citace: 2. 2 2014.] <http://diveintohtml5.info/>.

Ponkrác, Miloslav. 2000. JavaScript - kontrola správnosti e-mailové adresy 3. *Interval*. [Online] ZONER software, a.s, 14. 7 2000. [Citace: 18. 2 2014.] <http://interval.cz/clanky/javascript-kontrola-spravnosti-e-mailove-adresy-3/>.

Schools, W3. 2014. HTML References. *W3Schools*. [Online] 2014. [Citace: 13. 10 2013.] <http://www.w3schools.com/tags/default.asp>.

Urban, Petr. 2014. Podíl Windows XP se v lednu zvýšil. *ExtraWindows*. [Online] Extra Publishing, s. r. o., 6. 2 2014. [Citace: 18. 2 2014.] <http://extrawindows.cnews.cz/clanky/podil-windows-xp-se-lednu-zvysil-osmicky-narostly-jen-o-009>. ISSN 1804-9907.

Seznam obrázků

Obrázek 1 - Časová osa vývoje HTML a CSS.....	14
Obrázek 2 - Logická struktura stránky pomocí nových sémantických značek	19
Obrázek 3 - Pole pro výběr barvy [Google Chrome 33.0].....	21
Obrázek 4 - Formulářový prvek pro vkládání data [Google Chrome 33.0].....	21
Obrázek 5 - Formulářový element typu range[Google Chrome 32].....	22
Obrázek 6 - Element <meter>.....	22
Obrázek 7 - Progress bar s nastavenou hodnotou na 50 / 100.....	22
Obrázek 8 - Progress bar s nenastavenou aktuální hodnotou.....	23
Obrázek 9 - Základní použití elementu canvas	25
Obrázek 10 - Stíny, okraje a přechody pomocí CSS.....	33
Obrázek 11 - Grafický návrh pro tvorbu webových prezentací	36
Obrázek 12 - Podíly internetových prohlížečů, Leden 2014,.....	40
Obrázek 13 - Zakulacené rohy u náhledu článku.....	43
Obrázek 14 – Využití pseudotříd u hlavního menu	45
Obrázek 15 - Náhled umístění pozadí v hlavním menu	46
Obrázek 16 - Tabulka s odlišnou barevností řádků.....	47
Obrázek 17 - Mnohonásobné pozadí	49
Obrázek 18 - Stínování textu	52
Obrázek 19 - Transformace objektů	53
Obrázek 20 - Prolínání po najetí myši	55
Obrázek 21 - Popisek formulářového pole	56
Obrázek 22 - Atribut Placeholder	56
Obrázek 23 - Chybové hlášení HTML5 s atributem required.....	57
Obrázek 24 - Chybové hlášení pomocí JS alert.....	58
Obrázek 25 - Chybové hlášení pro ověření platnosti e-mailové adresy.....	59
Obrázek 26 - Select pro výběr barvy	60
Obrázek 27 - Formulářový element Range	61
Obrázek 28 - Názorná ukázka grafu	62

Seznam tabulek

Tabulka 1 - Podpora video formátů v prohlížečích.....	64
Tabulka 2 - Podpora zvukových formátů v prohlížečích.....	64
Tabulka 3 - Velikost specifikací	65
Tabulka 4 - Rychlost načítání stránek prostřednictvím webu Pingdom.....	66

Přílohy

Webové prezentace

Webové prezentace vytvářené v „Praktické části práce“ jsou k dispozici online na adresách:

- Varianta webové prezentace psaná v xHTML1.1 a CSS2.1:
 - o <http://bp-html4.lukasmodry.cz>
- Varianta webové prezentace psaná v HTML5 a CSS3:
 - o <http://bp-html5.lukasmodry.cz>

Dále jsou k dispozici na přiloženém DVD.

Přehled HTML značek

Značka	Definice značky
<!-- -->	Vkládání poznámky do zdrojového kódu
<!DOCTYPE>	Definování typu dokumentu
<a>	Vkládání hypertextového odkazu
<abbr>	Definování zkratky
<acronym>	Definice akronymu Není podporován v HTML5, alternativou je <abbr>
<address>	Definování kontaktních informací autora webového dokumentu
<applet>	Definování zásuvného modulu Není podporován v HTML5, alternativou je <object>
<area>	Definování oblasti uvnitř obrázkové mapy
<article>	Strukturální značka identifikující obsah stránky Přidáno v HTML5
<aside>	Strukturální značka identifikující vedlejší obsah stránky Přidáno v HTML5

<code><audio></code>	Vkládání zvukových souborů Přidáno v HTML5
<code></code>	Definování tučného textu
<code><base></code>	Specifikování báze URL adresy nebo cíle pro všechny relativní URL adresy v dokumentu
<code><basefont></code>	Specifikování základní barvy, velikosti a písma pro veškeré texty v dokumentu Není podporován v HTML5, alternativou jsou kaskádové styly
<code><bdi></code>	Izolování části textu s jiným formátem Přidáno v HTML5
<code><bdo></code>	Změna směru textu
<code><big></code>	Definování velkého textu Není podporováno v HTML5, alternativou jsou kaskádové styly
<code><blockquote></code>	Definování citované oblasti z jiného zdroje
<code><body></code>	Definice těla HTML dokumentu
<code>
</code>	Zalomení řádku
<code><button></code>	Vložení tlačítka
<code><canvas></code>	Plátno pro kreslení 2D objektů prostřednictvím skriptů Přidáno v HTML5
<code><caption></code>	Nadpis tabulky
<code><center></code>	Umístění textu na střed Není podporováno v HTML5, alternativou jsou kaskádové styly
<code><cite></code>	Definování názvu díla
<code><code></code>	Vložení úryvku zdrojového kódu
<code><col></code>	Specifikování sloupců ve skupině <code><colgroup></code>
<code><colgroup></code>	Specifikování skupiny sloupců pro formátování v tabulce

<code><command></code>	Vložení tlačítka, na které se může uživatel odvolat Přidáno v HTML5
<code><datalist></code>	Specifikování seznamu předdefinovaných možností Přidáno v HTML5
<code><dd></code>	Definování popisu / hodnoty v popisovacím listu
<code></code>	Definování textu, který má být z textu vymazán
<code><details></code>	Definování dalších detailů, které mohou být uživateli zobrazeny nebo skryty Přidáno v HTML5
<code><dfn></code>	Definování termínu
<code><dialog></code>	Definování dialogového boxu nebo okna Přidáno v HTML5
<code><dir></code>	Definování seznamu Není podporováno v HTML5, alternativou je
<code><div></code>	Strukturální značka pro členění obsahu
<code><dl></code>	Definování seznamu popisů
<code><dt></code>	Definování terminu / názvu v seznamu popisů
<code></code>	Definování zdůrazněného textu
<code><embed></code>	Definování kontejneru pro zásuvné moduly Přidáno v HTML5
<code><fieldset></code>	Skupina souvisejících formulářových prvků
<code><figcaption></code>	Definování názvu prvku umístěného v elementu <figure> Přidáno v HTML5
<code><figure></code>	Strukturální značka pro vkládání vlastního obsahu, například obrázku Přidáno v HTML5
<code></code>	Definování písma, barvy a velikosti textu Není podporováno v HTML5, alternativou jsou kaskádové styly

<code><footer></code>	Strukturální značka identifikující zápatí dokumentu Přidáno v HTML5
<code><form></code>	Vložení webového formuláře
<code><frame></code>	Definování samotného okna ve elementu <code><frameset></code> Není podporováno v HTML5
<code><frameset></code>	Definování skupiny elementů <code><frame></code> Není podporováno v HTML5
<code><h1></code> až <code><h6></code>	Nadpisy úrovní 1 až 6
<code><head></code>	Hlavička HTML dokumentu
<code><header></code>	Strukturální značka identifikující záhlaví dokumentu Přidáno v HTML5
<code><hgroup></code>	Skupina nadpisů dokument Přidáno v HTML5
<code><hr></code>	Horizontální čára
<code><html></code>	Definování kořene HTML stránky
<code><i></code>	Definování textu psaného kurzívou
<code><iframe></code>	Definování rádkového elementu <code><frame></code>
<code></code>	Vkládání obrázků
<code><input></code>	Vložení formulářového elementu pro vkládání dat
<code><ins></code>	Definování části textu, která má být vložena do dokumentu
<code><kbd></code>	Definování vstupu z klávesnice
<code><keygen></code>	Definování párů klíčových polí pro formuláře Přidáno v HTML5
<code><label></code>	Popisek polí ve formuláři
<code><legend></code>	Popisek elementu <code><fieldset></code>
<code></code>	Položka v seznamu

<code><link></code>	Definování vazby mezi dokumentem a externím zdrojem (například souboru s kaskádovými styly nebo JavaScriptem)
<code><main></code>	Specifikování hlavní části dokumentu Přidáno v HTML5
<code><map></code>	Definování obrazové mapy
<code><mark></code>	Definování podbarveného textu Přidáno v HTML5
<code><menu></code>	Definování seznamu ovládacích prvků
<code><meta></code>	Definování meta dat o HTML dokumentu
<code><meter></code>	Vložení elementu pro zobrazení stavu vykonávání zvoleného úkolu Přidáno v HTML5
<code><nav></code>	Strukturální značka identifikující navigaci stránky Přidáno v HTML5
<code><noframes></code>	Definování alternativního obsahu pro uživatele, jejichž prohlížeč nepodporuje používání framů Není podporováno v HTML5
<code><noscript></code>	Definování alternativního obsahu pro uživatele, jejichž prohlížeč nepodporuje skripty
<code><object></code>	Definování zásuvného modulu
<code></code>	Vložení třízeného seznamu
<code><optgroup></code>	Definování skupiny souvisejících možností v elementu <code><select></code>
<code><option></code>	Definování možnosti v elementu <code><select></code>
<code><output></code>	Definování výsledku kalkulace Přidáno v HTML5
<code><p></code>	Vložení odstavce
<code><param></code>	Definování parametrů objektu
<code><pre></code>	Definování předdefinovaného textu

<code><progress></code>	Vložení elementu pro zobrazování průběhu zpracování zvoleného úkolu Přidáno v HTML5
<code><q></code>	Definování krátké citace
<code><rp></code>	Definování alternativního obsahu pro uživatele, jejichž prohlížeč nepodporuje element <code><ruby></code> Přidáno v HTML5
<code><rt></code>	Definování významu / výslovnost znaků pro jihoasijskou typografii Přidáno v HTML5
<code><ruby></code>	Definování obsahu psaného v jihoasijské typografii Přidáno v HTML5
<code><s></code>	Definování obsahu, který není delší dobu platný
<code><samp></code>	Definování ukázky výstupu programu
<code><script></code>	Definování skriptu
<code><section></code>	Strukturální značka identifikující nespecifický obsah stránky Přidáno v HTML5
<code><select></code>	Vložení formulářového pole pro výběr z přednastavených možností
<code><small></code>	Definování malého textu
<code><source></code>	Definování zdroje medií Přidáno v HTML5
<code></code>	Definování jednořádkového obsahu v dokumentu
<code><strike></code>	Definování přeškrtnutého textu Není podporováno v HTML, alternativou je <code></code>
<code></code>	Definování důležitého textu
<code><style></code>	Definování stylů stránky
<code><sub></code>	Definování popisného textu
<code><summary></code>	Definování viditelné hlavičky pro element <code><details></code> Přidáno v HTML5

<code><sup></code>	Definování popisného textu
<code><table></code>	Vložení tabulky
<code><tbody></code>	Definování těla tabulky
<code><td></code>	Definování buňky v tabulce
<code><textarea></code>	Formulářové pole pro vkládání dlouhého textu
<code><tfoot></code>	Definování zápatí tabulky
<code><th></code>	Definování buněk v hlavičce tabulky
<code><thead></code>	Definování záhlaví tabulky
<code><time></code>	Definování data a času Přidáno v HTML5
<code><title></code>	Titulek HTML dokumentu
<code><tr></code>	Definování řádku tabulky
<code><track></code>	Definování textu nahrávky Přidáno v HTML5
<code><tt></code>	Definování dálnopisu Není podporováno v HTML5, alternativou jsou kaskádové styly
<code><u></code>	Definování podtrženého textu
<code></code>	Vložení netřízeného seznamu
<code><var></code>	Definování proměnné
<code><video></code>	Vkládání video souborů Přidáno v HTML5
<code><wbr></code>	Definování možnosti zalomení řádku Přidáno v HTML5

Přehled CSS vlastností

Vlastnost	Definice vlastnosti
<code>align-content</code>	Zarovnání obsahu Přidáno v CSS3
<code>align-items</code>	Zarovnání prvků v elementu Přidáno v CSS3
<code>align-self</code>	Zarovnání jednoho prvku v elementu Přidáno v CSS3
<code>animation</code>	Vázání animace k elementu Přidáno v CSS3
<code>animation-delay</code>	Zpoždění animace Přidáno v CSS3
<code>animation-direction</code>	Sled průběhu animace Přidáno v CSS3
<code>animation-duration</code>	Čas průběhu animace Přidáno v CSS3
<code>animation-fill-mode</code>	Umístění animace Přidáno v CSS3
<code>animation-iteration-count</code>	Počet opakování animace Přidáno v CSS3
<code>animation-name</code>	Název animace Přidáno v CSS3
<code>animation-play-state</code>	Pozastavení animace Přidáno v CSS3

<code>animation-timing-function</code>	Časování animace Přidáno v CSS3
<code>backface-visibility</code>	Viditelnost zadní strany elementu Přidáno v CSS3
<code>Background</code>	Nastavení pozadí elementu Nové možnosti nastavení v CSS3
<code>background-attachment</code>	Ukotvení pozadí elementu
<code>background-clip</code>	Specifikování oblasti pro umístění pozadí Přidáno v CSS3
<code>background-color</code>	Barva pozadí
<code>background-image</code>	Obrázek na pozadí
<code>background-origin</code>	Relativní pozice pozadí vůči elementu Přidáno v CSS3
<code>background-position</code>	Pozice pozadí elementu
<code>background-repeat</code>	Opakování pozadí elementu
<code>background-size</code>	Velikost pozadí elementu Přidáno v CSS3
<code>Border</code>	Okraj
<code>border-bottom</code>	Spodní okraj
<code>border-bottom-color</code>	Barva spodního okraje
<code>border-bottom-left-radius</code>	Rádus spodního levého okraje Přidáno v CSS3

<code>border-bottom-right-radius</code>	Rádus spodního pravého okraje Přidáno v CSS3
<code>border-bottom-style</code>	Styl spodního okraje
<code>border-bottom-width</code>	Šířka spodního okraje
<code>border-collapse</code>	Nastavení okrajů
<code>border-color</code>	Barva okraje
<code>border-image</code>	Použití obrázku jako okraje Přidáno v CSS3
<code>border-image-outset</code>	Umístění okraje od obsahu Přidáno v CSS3
<code>border-image-repeat</code>	Opakování obrázku použitého v okraji Přidáno v CSS3
<code>border-image-slice</code>	Nastavení vnitřní odsazení okraje Přidáno v CSS3
<code>border-image-source</code>	Zdroj obrázku použitého u okraj Přidáno v CSS3
<code>border-image-width</code>	Šířka obrázku použitého u okraje Přidáno v CSS3
<code>border-left</code>	Levý okraj
<code>border-left-color</code>	Barva levého okraje
<code>border-left-style</code>	Styl levého okraje
<code>border-left-width</code>	Šířka levého okraje

<code>border-radius</code>	Rádus okraje Přidáno v CSS3
<code>border-right</code>	Pravý okraj
<code>border-right-color</code>	Barva pravého okraje
<code>border-right-style</code>	Styl pravého okraje
<code>border-right-width</code>	Šířka pravého okraje
<code>border-spacing</code>	Rozestupy polí
<code>border-style</code>	Styl okraje
<code>border-top</code>	Horní okraj
<code>border-top-color</code>	Barva horního okraje
<code>border-top-left-radius</code>	Rádus levého horního okraje Přidáno v CSS3
<code>border-top-right-radius</code>	Rádus horního pravého okraje Přidáno v CSS3
<code>border-top-style</code>	Styl horního okraje
<code>border-top-width</code>	Šířka horního okraje
<code>border-width</code>	Šířka okraje
<code>Bottom</code>	Pozice elementu od spodní hranice
<code>box-shadow</code>	Stínování elementu Přidáno v CSS3
<code>box-sizing</code>	Rozestup 2 prvků Přidáno v CSS3

<code>caption-side</code>	Umístění nadpisu
<code>Clear</code>	Zakázání obtékání elementu
<code>Clip</code>	Ukotvení obrázku
<code>Color</code>	Barva elementu
<code>column-count</code>	Definování počtu sloupců Přidáno v CSS3
<code>column-fill</code>	Definování způsobu vyplnění sloupce Přidáno v CSS3
<code>column-gap</code>	Mezera mezi sloupci Přidáno v CSS3
<code>column-rule</code>	Pravidla pro definování sloupců Přidáno v CSS3
<code>column-rule-color</code>	Specifikování barvy mezi sloupci Přidáno v CSS3
<code>column-rule-style</code>	Specifikování stylu mezi sloupci Přidáno v CSS3
<code>column-rule-width</code>	Specifikování šířky mezi sloupci Přidáno v CSS3
<code>column-span</code>	Umístění prvku přes více sloupců Přidáno v CSS3
<code>column-width</code>	Šířka sloupců Přidáno v CSS3

Columns	Definování sloupců Přidáno v CSS3
Content	Vložení obsahu
counter-increment	Číslování prvků pomocí počítadla
counter-reset	Vynulování počítadla
Cursor	Vzhled kurzoru
Direction	Změna směru textu
Display	Způsob zobrazení elementu
empty-cells	Skryje okraj a pozadí prázdných buněk
Flex	Všechny flexibilní prvky mají totožné nastavení Přidáno v CSS3
flex-basis	Nastavení pro konkrétní prvek Přidáno v CSS3
flex-direction	Nastavení opačného počítání prvků Přidáno v CSS3
flex-flow	Nastavení opačného pořadí prvků Přidáno v CSS3
flex-grow	Rozlišení velikosti prvků Přidáno v CSS3
flex-shrink	Nastavení zmenšení prvku Přidáno v CSS3
flex-wrap	Zabalení flexibilních prvků Přidáno v CSS3

Float	Obtékání elementu
Font	Volba písma
@font-face	Definování vlastní rodiny fontů
font-family	Definování vzhledu písma
font-size	Velikost písma
font-size-adjust	Nastavení velikosti textu pro všechny prohlížeče
font-stretch	Širší písmena
font-style	Styl písma
font-variant	Nastavení kapitálek
font-weight	Tučnost písma
hanging-punctuation	Umístění interpunkce Přidáno v CSS3
Height	Výška
Icon	Vzhled obrázku s alternativou v podobě ikony Přidáno v CSS3
justify-content	Zarovnání obsahu Přidáno v CSS3
@keyframes	Efekt padajícího objektu Přidáno v CSS3
Left	Pozice elementu od levé hranice
letter-spacing	Rozestup písmen
line-height	Výška řádku

<code>list-style</code>	Styl seznamu
<code>list-style-image</code>	Definování vlastního vzhledu odrážky
<code>list-style-position</code>	Definování polohy odrážky
<code>list-style-type</code>	Definování stylu odrážky
Margin	Vnější odsazení
<code>margin-bottom</code>	Vnější odsazení od spodní hrany
<code>margin-left</code>	Vnější odsazení od levé hrany
<code>margin-right</code>	Vnější odsazení od pravé hrany
<code>margin-top</code>	Vnější odsazení od horní hrany
<code>max-height</code>	Maximální výška
<code>max-width</code>	Maximální šířka
<code>min-height</code>	Minimální výška
<code>min-width</code>	Minimální šířka
<code>nav-down</code>	Definování posunu použitím klávesy dolů Přidáno v CSS3
<code>nav-index</code>	Definování posunu použitím klávesy Přidáno v CSS3
<code>nav-left</code>	Definování posunu použitím klávesy doleva Přidáno v CSS3
<code>nav-right</code>	Definování posunu použitím klávesy doprava Přidáno v CSS3

<code>nav-up</code>	Definování posunu použitím klávesy dolů Přidáno v CSS3
<code>opacity</code>	Průhlednost Přidáno v CSS3
<code>Order</code>	Nastavení pořadí flexibilních prvků Přidáno v CSS3
<code>Outline</code>	Definování vnějšího ohraničení
<code>outline-color</code>	Definování barvy vnějšího ohraničení
<code>outline-offset</code>	Definování vnějšího stylu
<code>outline-style</code>	Definování stylu vnějšího ohraničení
<code>outline-width</code>	Definování šířky vnějšího ohraničení
<code>overflow</code>	Přetékání
<code>overflow-x</code>	Vertikální přetékání
<code>overflow-y</code>	Horizontální přetékání
<code>padding</code>	Vnitřní odsazení
<code>padding-bottom</code>	Vnitřní odsazení od spodní hrany
<code>padding-left</code>	Vnitřní odsazení od levé hrany
<code>padding-right</code>	Vnitřní odsazení od pravé hrany
<code>padding-top</code>	Vnitřní odsazení od horní hrany
<code>page-break-after</code>	Pravidlo pro vkládání zalomení za prvkem Přidáno v CSS3

<code>page-break-before</code>	Pravidlo pro vkládání zalomení před prvkem Přidáno v CSS3
<code>page-break-inside</code>	Pravidlo pro vkládání zalomení uvnitř prvku Přidáno v CSS3
<code>perspective</code>	Definování perspektivního pohledu na prvek Přidáno v CSS3
<code>perspective-origin</code>	Definování způsobu pohledu prvku Přidáno v CSS3
<code>position</code>	Způsob umístění elementu
Quotes	Citace
Resize	Nastavení možnosti změny velikosti v závislosti na obsahu
Right	Pozice elementu od pravé hranice
<code>tab-size</code>	Nastavení velikosti tabulátoru
<code>table-layout</code>	Nastavení šablony pro tabulku
<code>text-align</code>	Zarovnání textu
<code>text-align-last</code>	Zarovnání textu na konci řádku
<code>text-decoration</code>	Zvýraznění textu
<code>text-decoration-color</code>	Barva zvýraznění textu
<code>text-decoration-line</code>	Typ čáry pro zvýraznění textu
<code>text-indent</code>	Odsazení prvního řádku tabulky
<code>text-justify</code>	Definování způsobu zarovnání

<code>text-overflow</code>	Přetékání textu
<code>text-shadow</code>	Síňování textu Přidáno v CSS3
<code>text-transform</code>	Definování velikosti písmen
<code>Top</code>	Pozice elementu od horní hranice
<code>transform</code>	Transformování elementů Přidáno v CSS3
<code>transform-origin</code>	Způsob transformování elementů Přidáno v CSS3
<code>transform-style</code>	Styl transformování elementů Přidáno v CSS3
<code>transition</code>	Efekt pronutí Přidáno v CSS3
<code>transition-delay</code>	Prodleva efektu pronutí Přidáno v CSS3
<code>transition-duration</code>	Délka trvání efektu pronutí Přidáno v CSS3
<code>transition-property</code>	Typ efektu pronutí Přidáno v CSS3
<code>transition-timing-function</code>	Časování efektu pronutí Přidáno v CSS3
<code>unicode-bidi</code>	Směr psaní textu
<code>vertical-align</code>	Vertikální zarovnání

<code>visibility</code>	Viditelnost
<code>white-space</code>	Způsob obalování obsahu
<code>width</code>	Šířka
<code>word-break</code>	Zalamování slov
<code>word-spacing</code>	Velikost mezer mezi slovy
<code>word-wrap</code>	Obalování textu
<code>z-index</code>	Definování priority elementu
