

Vysoká škola ekonomická v Praze

Fakulta informatiky a statistiky

Katedra informačního a znalostního inženýrství

Autor : **Petr Stehlík**
Vedoucí bakalářské práce : **PhDr. Otakar Pinkas**
Oponent bakalářské práce : **Ing. David Chudán**

TÉMA BAKALÁŘSKÉ PRÁCE

Praktické využití jazyka HTML 5 a CSS 3

ROK : 2011

Prohlášení

Prohlašuji, že jsem bakalářskou práci zpracoval samostatně a že jsem uvedl všechny použité prameny a literaturu, ze kterých jsem čerpal.

V Praze dne

.....

podpis

Abstrakt

Tato bakalářská práce pojednává o dvou nových jazycích, HTML 5 a CSS 3, které jsou pravděpodobně budoucností webových stránek. Oproti jejich předchozím verzím obsahují mnoho novinek. V současnosti jsou oba jazyky ve stádiu vývoje, ale již se začínají objevovat v praktickém použití na webu.

Cílem práce bylo prozkoumat použitelnost HTML 5 a CSS 3 a zjistit, zda se vyplatí již dnes na tyto jazyky přejít. Toho jsem docílil teoretickým průzkumem možností jazyků a jejich následnou aplikací. Pro zodpovězení otázky funkčnosti a použitelnosti stránek, napsanými těmito jazyky, jsem porovnal jejich grafický vzhled a funkčnost v různých prohlížečích.

Výsledkem práce je zjištění, že HTML 5 i CSS 3 lze pro výhody, které přináší, využívat již dnes. Je jen otázkou, jak naložit s kompatibilitou starších prohlížečů. To, že starší prohlížeče nové prvky nepodporují lze obejít pomocí různých skriptů.

Bakalářská práce obsahuje teoretickou a praktickou část. V teoretické části popisují nejdůležitější nové možnosti HTML 5 a CSS 3 a v praktické části ověřuji jejich použitelnost.

Abstract

This bachelor thesis deals with two new languages, HTML 5 and CSS 3. They are probably the future of the Web as they contain many useful novelties, compared to their previous versions. At present both languages are still under the development, but they have already begun to be used in practice on the Web.

The aim of this thesis was to find out the usability of HTML 5 and CSS 3 and to determine whether it is really worth to use them today. I worked it out by the theoretical research of languages possibilities and their application. To answer the question of functionality and usability of web sites written by these languages I compared their graphic design and functionality in different browsers.

The result is that HTML 5 and CSS 3 can be effectively used today. There is only the question how to deal with the compatibility with older browsers. The fact that older browsers do not support new features can be bypassed by using different scripts.

The bachelor thesis has theoretical and practical part. In the theoretical part I describe the most important features of HTML 5 and CSS 3. In the practical part I verify their usability.

Poděkování

Děkuji vedoucímu práce panu PhDr. Otakaru Pinkasovi za vedení práce a cenné připomínky, které byly velmi nápomocné při tvorbě této práce.

Obsah

Úvod.....	8
Teoretická část	9
1. Seznámení s HTML 5	9
1.1 Historie HTML 5	9
1.2 Co řeší HTML 5	9
1.3 Kdo vyvíjí HTML 5?	10
1.4 Průběh schvalování specifikace	10
1.5 Struktura HTML	10
1.6 Syntaxe.....	11
1.7 Jak přejít na novou verzi HTML 5.....	12
2. Nové vlastnosti HTML 5	13
2.1 Strukturování stránky.....	13
2.1.1 Problém u starších prohlížečů.....	15
2.2 Nové elementy	16
2.2.1 Video.....	16
2.2.2 Audio	18
2.2.3 Canvas.....	18
2.2.4 Geolokace	18
2.2.5 Formuláře.....	19
3. CSS 3	22
3.1 Úvod do CSS 3	22
3.2 Syntaxe CSS a selektory	22
3.2.1 Odkazování pomocí hodnoty atributů.....	23
3.2.2 Odkazování pomocí pseudotříd	23
3.2.3 Odkazování podle pořadí	23
3.2.4 Odkazování pomocí matematických operací	23
3.3 Vendor prefixy	24
3.4 Nové vlastnosti CSS 3	24
3.4.1 Nové fonty	24
3.4.2 Barvy.....	25
3.4.3 Border-radius	26

3.4.4 Box-shadow	27
3.4.5 Text-shadow.....	27
3.4.6 Gradienty	27
3.4.7 Vícenásobné pozadí	30
3.4.8 Transformace a posun prvku.....	30
3.4.9 Media Queries.....	31
3.5 Starší prohlížeče.....	31
3.5.1 Vytvoření jednoduchého tlačítka s efekty CSS 3	32
Praktická část	34
4. Struktura dokumentu.....	34
4.1 Stará a nová struktura dokumentu	34
4.2 Problém s Internet Explorerem	35
5. Nový vzhled.....	38
5.1 Styl pro větší zobrazovací zařízení a novější prohlížeče	38
5.1.1 Pozadí.....	38
5.1.2 Kulaté rohy	41
5.1.3 Písmo	41
5.1.4 Jiné počáteční písmeno v odstavci.....	42
6. Nové funkce.....	43
6.1 Animace	43
6.2 Formuláře.....	44
6.3 Video.....	46
6.4 Geolokace	46
7. Validita.....	48
8. Test prohlížečů.....	49
8.1 Grafická podoba.....	49
8.2 Funkčnost.....	49
8.3 Použité prvky	49
Závěr	50
Bibliografie	51
Seznam obrázků.....	52
Seznam příloh	53
Seznam použitých termínů a zkratk	54

Příloha 1 – použité nástroje z webu	56
Příloha 2 – stará struktura HTML dokumentu.....	57
Příloha 3 – nová struktura HTML dokumentu.....	58
Příloha 4 – spustitelné soubory stránek	na přiloženém CD

Úvod

V roce 1999 byla ukončena tvorba jazyka HTML verzí 4.01 a vývoj tvorby webu se začal ubírat jiným směrem. V roce 2007 se však vrátila pozornost k jazyku HTML a začalo se pracovat na jeho páté verzi. Ta řeší některé nedokonalosti starší verze a přináší mnoho novinek. HTML 5 by mělo nabídnout přehrávání videa, zvuku, podporu kreslení, tvorbu animací i her. Dále umožňuje snazší práci s formuláři a zlepšuje sémantiku kódu. Spolu s HTML 5 se vyvíjí nová verze Kaskádových stylů, pomocí níž bude konečně možné vytvořit některé grafické prvky na webu, které se již dlouho používají, ale kterých kódem pomocí starší verze nelze dosáhnout.

HTML 5 a CSS 3 jsou dlouho očekávané technologie, pomocí nichž se pravděpodobně budou v následujících letech tvořit webové stránky. Jazyky jsou pořád ještě ve stádiu návrhu, ale již dnes se implementují. Zajímalo mě využití těchto technologií v praxi.

Cílem mé práce je prozkoumat použitelnost HTML 5 a CSS 3 a zjistit, zda se vyplatí již dnes na tyto jazyky přejít. Toho se snažím docílit teoretickým průzkumem možností jazyků a jejich následnou aplikací. Pro zodpovězení otázky funkčnosti a použitelnosti stránek porovnávám jejich grafický vzhled a funkčnost v různých prohlížečích.

Toto téma je velmi rozsáhlé a jeho celkový popis a průzkum není možné v této práci obsáhnout. V práci jsou popsány a ukázány základní funkce jazyků.

Bakalářská práce je rozdělena na teoretickou a praktickou část. V teoretické části jsem se zaměřil na historii a důvod vývoje HTML 5 a přistoupil jsem k popisu jednotlivých nových funkcí jazyků HTML 5 a CSS 3. V praktické části jsem se rozhodl přetvořit stávající webovou prezentaci s využitím nových technologií a zároveň prakticky vyzkoušet nové funkce HTML 5.

Teoretická část

1. Seznámení s HTML 5

1.1 Historie HTML 5

Historie HTML 5 sahá do prosince 1997, kdy W3C (*World Wide Web Consortium*) vydalo HTML 4.0 a následně se další vývoj prakticky zastavil (v prosinci 1999 byl vydán nový standard HTML 4.01, který pouze upravuje drobné chyby předchozí verze). Ani ne za dva měsíce vydalo W3C XML 1.0 (*eXtensible Markup Language*). XML se stalo celosvětově používanou technologií pro výměnu a ukládání dat. Za tři měsíce nato pořádalo W3C konferenci na téma, zda má nebo nemá upustit od dalšího vývoje jazyka HTML.

V následné diskuzi bylo dohodnuto, že další rozšíření a vývoj HTML 4.0 a jeho konverze do XML aplikací by byla obtížná. Jednou z možností, jak se vyhnout těmto omezením, bylo vyvinout další verzi HTML založenou na XML tagách (*elementech*).

Změněná HTML pracovní skupina W3C se tedy začala soustředit na XML. V roce 1998 vydala návrh na specifikaci, kde přeformulovala HTML do XML bez přidání nových tagů a atributů, a tím vznikla specifikace známá později jako XHTML 1.0 a 1.1. Následujícím krokem byly webové formuláře, a tak byl v roce 1999 vydán první návrh XHTML Extended Forms, opět HTML pracovní skupinou. Ta se rozhodla, že při použití formulářů nelze zajistit zpětnou kompatibilitu s prohlížeči pro starší verze jazyka HTML. Cílem bylo, aby nové XHTML Extended Forms bylo založeno na přesně definovaných pravidlech a v roce 2002 W3C vydalo XHTML 2.0 [1].

Problém se zpětnou kompatibilitou způsobil, že používanější technologií bylo HTML 4.01 a XHTML 1.0. Společnosti Mozilla Corporation a Opera navrhly W3C, aby se upustilo od dalšího vývoje XHTML a soustředilo se na rozšíření HTML v otázce aplikací. W3C trvalo na svém, a tak XHTML Extended Forms bylo přejmenováno na XForms a přesunuto do nové pracovní skupiny WHATWG (*Web Hypertext Application Technology Working Group*). WHATWG založily Apple, Mozilla a Opera a v říjnu 2003 tato skupina vydala XForms 1.0, zatímco W3C pokračovalo dále na technologii XHTML. V roce 2007 W3C přiznalo, že XHTML 2 bylo krokem vedle, a upustilo od dalšího vývoje. Pracovní skupiny HTML a WHATWG se spojily a v roce 2008 byl publikován první návrh HTML 5, který vychází z práce odvedené WHATWG [2].

1.2 Co řeší HTML 5

Proč tedy pokračovat s HTML? HTML 5 se snaží být multiplatformní – dnes je potřeba mít v prohlížeči nainstalován Flash či Silverlight. Web je dnes plný různých animací a videí a bez stažení příslušného zásuvného modulu (*pluginu*) video či animace fungovat nebudou. Jsou uživatelé, kteří si prohlíží obsah webu s vypnutým flashem, protože se ve velké míře používá při tvorbě reklam. Dalším problémem je prohlížení internetu z mobilních zařízení, televize apod. Nová verze by měla nabídnout ucelenou technologii pro tvorbu internetových stránek. S její pomocí by mělo jít přehrát video, zvukovou stopu, tvořit animace

i hry. Dále je snaha zlepšit sémantiku webu a důrazněji oddělit vzhled stránky od jeho obsahu. Mezi novinky také patří lepší práce s formuláři, kdy formulář sám kontroluje vstupní data a nemusí se tedy psát skript.

1.3 Kdo vyvíjí HTML 5?

Za vývojem HTML 5 stojí několik organizací: W3C, WHATWG, Khronos Group – stojí za 3D grafikou pro prohlížeče, IETF – síťový protokol websockets, ECMA (*European association for standardizing information and communication systems*) – použití JavaScriptu [3]. Tyto organizace se starají o vývoj, ale na HTML 5 se celkem podílí 422 nezávislých odborníků a 70 odborníků z různých organizací. Česká republika má také dva zástupce [4].

1.4 Průběh schvalování specifikace

Specifikace je důležitý dokument, který přesně definuje vlastnosti jazyka a jak používat jeho konkrétní funkce.

Problémem HTML 5 je, že ještě není hotová specifikace a většina návrhů je pouze v prvních dvou stádiích, takže ještě může dojít k mnoha změnám. Při tvorbě stránek v HTML 5 je tedy vhodné zjistit v jakém stádiu je element, který chceme použít. Schvalování specifikace je dlouhý proces a má tato stádia:

- First public working draft – První veřejně vydaná verze, na které se stále pracuje a může se kdykoliv změnit. Je i možné, že funkce se zamítne a nebude již podporována vůbec.
- Working draft – Když návrh projde prvním kolem First public working draft, tak se přesune do Working draft, kde se na něm začne detailně pracovat. Může dojít ještě k velkým změnám.
- Last Call – V tomto stupni schvalování je poslední možnost zaslat návrh o jeden i dva stupně zpět k dalšímu projednávání.
- Candidate recommendation – Finální zpracování a implementace.
- Recommendation – Schválený návrh, pouze připomínky [5].

1.5 Struktura HTML

HTML soubor je text, který obsahuje elementy, které se zapisují do ostrých závorek. Ten určuje, co ve stránce bude prohlížeč číst jako kód. Každý element může obsahovat atribut, který slouží pro jeho upřesnění – nese nějakou vlastnost.

V HTML 5 většinou zůstaly elementy z předchozí verze a přidalo se mnoho nových jako `<audio>`, `<video>`, `<canvas>` apod.

1.6 Syntaxe

Při vývoji jazyka se zkoumalo, co který prohlížeč dokáže a nedokáže zobrazit správně. To, co šlo zjednodušit, aniž by se narušil vzhled stránek ve starších prohlížečích, není ve verzi HTML 5 vyžadováno. Například dle syntaxe XHTML musí být nepárové tagy jako `<img>` ukončeny zpětným lomítkem. HTML 5 už to však nevyžaduje. HTML 5 se syntaxí příliš nezabývá. Pouze tam, kde hrozí, že se daný prvek vykreslí nesprávně, zahlásí validátor chybu [6].

Validní dokument:

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8">
<link rel="stylesheet" href="style/reset.css" type="text/css">
<LINK REL="stylesheet" HREF="style/reset.css" TYPE="text/css">
<Link Rel="stylesheet" Href="style/reset.css" Type="text/css">
<link rel="stylesheet" href="style/reset.css" type="text/css" />
<link rel=stylesheet href=style/reset.css type=text/css>
<link rel='stylesheet' href='style/reset.css' type='text/css'>
<title></title>
</head>
<body>
</body>
</html>
```

V HTML 5 lze psát v názvech tagů i atributů malá i velká písmena (nebo jejich kombinace). Lze uvádět hodnoty atributů jak v závorkách, tak i bez nich.

Konkrétní změny v syntaxi:

- Některé elementy se již nemusí uvádět. Například: `<html>`, `<body>`.
- Nepárové elementy nemusí být ukončovány zpětným lomítkem.
- Některé párové tagy se nemusí uzavírat. Například `<p>` či `<li>`.
- Elementy a atributy se mohou psát malým či velkým písmem, hodnoty atributů nemusí být v uvozovkách. Pokud hodnoty budou mít mezeru, uvozovky jsou potřeba.
- Text mimo element je validní.
- Element `<form>` nemusí mít atribut `action`.
- `text-tarea` nemusí mít atributy `cols` a `rows`.
- Některé prvky, které nebyly podporovány, již jsou podporovány.
- Do odkazů (`<a>`) je možné vkládat blokové prvky.
- „&” se nemusí zapisovat jako `&`, pokud je jako text [6].

Další příklad validního dokumentu:

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

```

<link rel="stylesheet" href="style/reset.css" type="text/css">
<title></title>
</head>
<ul class="bullet">
<li class="current_page"><a href="index.html">O mně</a>
<li><a href="obcanske-pravo.html">Občanské právo</a>
<li><a href="trest.html">Trestní právo</a>
<li><a href="mediace.html">Mimosoudní řešení sporů</a>
<li><a href="ceny.html">Ceny</a>
<li><a href="kontakt.html">Kontakt</a>
</ul>
<p>Zastupuji své klienty před soudy všech stupňů, v občanskoprávních
i v trestních věcech.

```

Validní HTML 5 dokument bez tagů `<html>`, `<body>` a bez ukončeného elementu `<p>`, `<li>`.

1.7 Jak přejít na novou verzi HTML 5

Přejít na novou verzi HTML 5 je velmi jednoduché, stačí pouze uvést správnou DTD (*deklarace typu dokumentu*) a to pouze tagem `<!DOCTYPE html>`. V HTML 5 se tedy DTD zapisuje kratším způsobem, než v předchozí verzi. Kód HTML 4 není třeba upravovat, zpětná kompatibilita se staršími prohlížeči je zajištěna a stránka se vykreslí ve standardním zobrazovacím režimu. To platí ovšem pro určité prvky. Starší prohlížeče samozřejmě některé části HTML 5 nepodporují. Jedná se o novinky jako je video, audio či zjištění aktuální polohy.

2. Nové vlastnosti HTML 5

2.1 Strukturování stránky

HTML 5 vylepšuje strukturu stránek přidáním několika nových elementů. Přidáním těchto elementů se zdrojový kód stává více přehledným pro vývojáře, prohlížeče a roboty (viz *Obrázek 3*). V předchozí verzi HTML se místo těchto nových značek musely používat elementy `<div>` a pro jejich správné umístění použít identifikátory (viz *Obrázek 2*). Roboti stránky indexují a díky novější struktuře budou schopni nabídnout relevantnější obsah. S použitím těchto nových elementů snadno zjistí, kde se nachází například hlavní obsah dané stránky. Tím vyhledávače budou moci nabídnout uživateli relevantnější obsah – sami rozpoznají, jaké je hlavní sdělení a co ne [7]. Uživatel se v dokumentu může snadněji pohybovat. Nové elementy mu umožňují přeskočit navigaci či přejít rovnou na další článek bez toho, aby autor stránek musel použít speciální odkazy [4]. Další výhoda je pro nevidomé – nové elementy nabízejí lepší orientaci na stránce. Umožní například obrazové čtečky rychle přejít k navigaci (`<nav>`) nebo obsahu. Před příchodem tohoto elementu bylo k přeskočení navigace potřeba vytvořit odkaz. Bylo prohledáno velké množství stránek a nejpoužívanější elementy byly zařazeny do jazyka HTML 5 [3]. Mezi tyto elementy patří:

- `<header>` – hlavička vlastní stránky, kam patří nadpis, informace o verzi, logo stránky atd.
- `<footer>` – patička stránky, která obsahuje informace o autorovi stránky, autorská práva či datum vytvoření.
- `<nav>` – část stránky určená pro odkazy, a to celého webu i jednotlivých dokumentů.
- `<aside>` – boční panel, který se týká hlavního obsahu. Může v něm být umístěna například reklama.
- `<section>` – různé přirozené úseky dokumentu, například kapitoly.
- `<article>` – představuje článek dokumentu jako zprávu, text, komentáře atd.
- `<hgroup>` – sémantické rozložení nadpisů. Nadpisy určují rozložení dokumentu.

Každý nadpis je v dokumentu uzlem a dohromady tvoří osnovu dokumentu. Kdybychom na stránkách, například pod hlavním nadpisem stránky, měli slogan, tak v HTML 4 bychom neměli možnost, jak slogan sémanticky do stránky zařadit. Kdybychom mu přiřadili nadpis `<h2>`, tak ostatní nadpisy by byly `<h3>`, což by bylo nevhodné. `<h2>` jim tedy dát nemůžeme, abychom dodrželi správnou strukturu dokumentu. `<hgroup>`, v případě HTML 5, sváže slogan s nadpisem a vše bude v pořádku. Nadpisu tedy přidělíme `<h1>`, sloganu `<h2>` a obalíme je tagem `<hgroup>`. Ostatním hlavním nadpisům na stránce přidělíme `<h2>` [8].

- `<time>` – slouží pro určení času, kdy vyšel nějaký článek nebo celá stránka. Když se vloží mezi element `<article>` a `</article>`, označuje datum článku, jinak datum celé stránky. Například: `<time datetime="2011-10-25" pubdate>25. října</time>`. První datum je čitelné pro roboty, uživateli se nezobrazí, to druhé je již pro samotného uživatele.

V HTML 4 byly na výběr nadpisy <h1> až <h6>, přičemž stránka by měla obsahovat nadpis <h1> pouze jednou. Elementy <article>, <section> atd. jsou v HTML 5 definovány jako nová část dokumentu a každá tato část může obsahovat vlastní nadpis <h1>. Tato novinka přišla především kvůli generovanému obsahu, kdy se obsah obalí tagem <article> a nemusí se zkoumat, v jaké „nadpisové úrovni“ právě je [8].

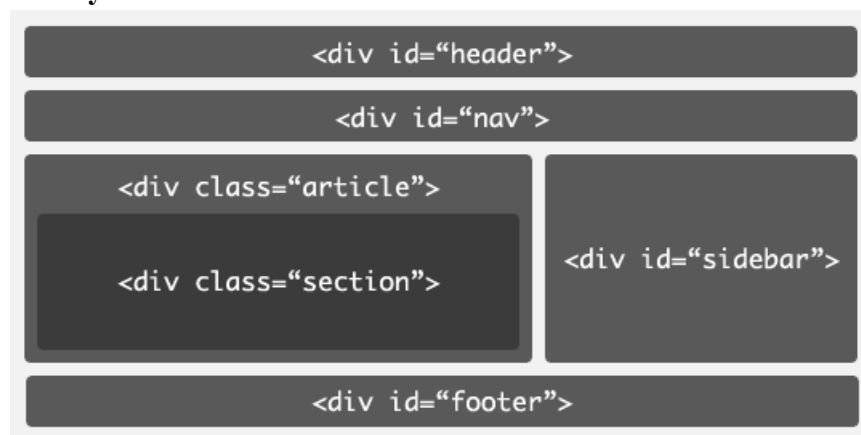
Správnost osnovy nadpisů dokumentu lze ověřit například na adrese:
<http://gsnedders.html5.org/outliner/> (viz Obrázek 1).

1. JUDr. Eva Stehlíková, právní služby
1. *Untitled Section*
1. *Untitled Section*
2. Advokátka JUDr. Eva Stehlíková

Obrázek 1 – osnova dokumentu

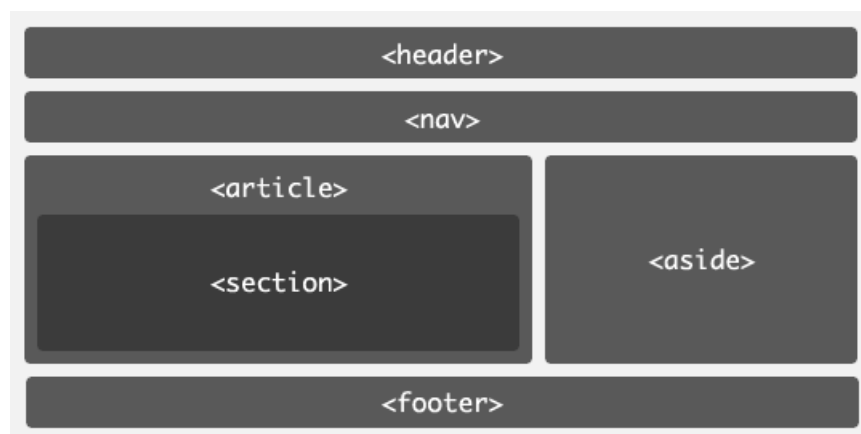
Zdroj: <http://gsnedders.html5.org/outliner/process.py>

Použití sémantických elementů



Obrázek 2 – struktura webu pomocí divů v HTML 4

Zdroj: <http://interval.cz/podklady/1999-2008/hunt/1434/structure-div.png>



Obrázek 3 – struktura webu bez použití divů, s použitím nových elementů v HTML 5

Zdroj: <http://interval.cz/podklady/1999-2008/hunt/1434/structure-html5.png>

2.1.1 Problém u starších prohlížečů

V případě, že prohlížeč nějaký prvek nezná, vykreslí ho řádkově. Pro blokové prvky z HTML 5 je potřeba nastavit styl zobrazení:

```
article, aside, details, figcaption, figure,
footer, header, hgroup, menu, nav, section {
    display: block;
}
```

Problém je s Internet Explorerem (IE) starší verze, nežli je IE 9. Tyto prohlížeče neznámé prvky zkrátka ignorují. Neznámé prvky vloží IE 8 (a starší) do DOMu (*Dokument object model*) jako prvek bez vlastností a bez dětí. Všechny prvky, které měly být dětmi, se chovají jako sourozenci. To lze vyřešit pomocí JavaScriptu. Například, když IE nerozpozná

element `<header>`, je potřeba na stránce vytvořit pomocí JavaScriptu prvek header dříve, než je skutečně na stránce použit. Poté bude IE chápat styl klasicky a styly lze normálně definovat [8].

```
<html>
<head>
...
...
<link rel="stylesheet" href="style.css" type="text/css">
<!--[if lt IE 9]>
<script>document.createElement("header");</script>
<![endif]-->
</head>
<body>
...
<header>
<a href="index.html"> </a>
...
</header>
...
</body>
</html>
```

Skript se musí uvádět na začátek stránky, nejlépe do hlavičky. Prohlížeč potom bude schopen sestavit správný DOM. Skript je zabalen do podmínky, kterou bude číst pouze prohlížeč IE, který je starší nežli IE 8. Ostatní prohlížeče budou zápis ignorovat. Do souboru style.css je tedy třeba uvést:

```
header {
    display: block;
}
```

Je rozumné zprovoznit všechny HTML 5 prvky najednou. Existuje skript, který zprovozní všechny HTML 5 značky najednou. Stačí ho jen stáhnout a vkopírovat do svého kódu. Lze ho získat na této adrese: <http://html5doctor.com/html-5-reset-stylesheet/>.

2.2 Nové elementy

2.2.1 Video

Jednou z nejvíce očekávaných novinek je element `<video>`. Možnost umístit na web video je záležitostí několika posledních let, takže se v HTML 4.01 s touto možností nepočítalo. Jak tedy vložit video na stránky a umožnit jeho přehrávání? Možností je více. Lze využít QuickTime, Windows Media či další, ale nepoužívanější technologií je v současnosti Flash [4].

Video v HTML 5 má poskytovat přímo prohlížeč bez různých zásuvných modulů a zároveň má obsahovat API (*rozhraní pro programování aplikací*) pro jeho přehrávání. Výhodou je, že video by mělo fungovat na jakémkoli zařízení bez nutnosti další instalace.

Vložení videa na stránky je velmi jednoduché:

```
<video width="320" height="240" controls="controls">
  <source src="klip.mp4" type="video/mp4" />
  <source src="klip.ogg" type="video/ogg" />
  Je nám líto, ale Váš prohlížeč nepodporuje přehrávání videa v HTML 5.
</video>
```

Kód pro vložení videa je daleko kratší a přehlednější. Vše se vkládá mezi tagy `<video>`, takže se video bude nechat dát vyhledávači indexovat.

Prohlížeč neví, jak je video velké, a proto by se měly atributy `width` a `height` uvádět. Prohlížeč bude vědět, kolik místa si má při načítání na stránce na video vynechat. `<controls>` zobrazí nebo nezobrazí ovládací lištu pro spuštění či zastavení videa a ovládání hlasitosti. Element `<source>` určuje umístění videa. Prohlížeč začne automaticky přehrávat první formát, který dokáže přehrát. Je vhodné vložit také nějaký text upozorňující na to, že daný prohlížeč tag `<video>` nepodporuje [9].

Další atributy elementu `<video>` jsou:

- **audio** – můžeme nastavit hodnotu `muted` a video bude ztlumené.
- **autoplay** – atribut nemá hodnotou, video se začne po načtení stránky samo přehrávat.
- **loop** – atribut nemá hodnotu, video se bude přehrávat stále dokola.
- **poster** – URL adresa obrázku reprezentující video.
- **preload** – video se začne samo od sebe načítat.

Video – problém s kodeky

Video nebo zvukovou stopu můžeme označit jako kontejner, který obsahuje video, zvuk a metadata nesoucí informace o videu. To vše je v jednom souboru, který má příponu .ogg, .avi, .mov, .flv atd. Video či audio je velký soubor (jedna minuta videa může mít i 1,5 GB) a je potřeba ho kódovat a dekodovat. Tím získáme například pětinou velikost, kterou již lze použít. Pro následné přehrávání těchto souborů potřebujeme příslušný kodek. Kodeků pro audio i video existuje velká řada, například: MPEG-4, H.264 nebo Xvid [10].

HTML 5 mělo nabídnout jeden kodek, který měl být společný pro všechny prohlížeče. Bohužel se firmy nedokázaly dohodnout, který kodek použít. Používání pouze jednoho kodeku je velmi lákavé – pravděpodobně by se stal (díky budoucímu rozšíření HTML 5) nejpoužívanějším kodekem vůbec. Spory o to, který kodek použít, jsou opravdu na dlouho: Open source kodek OGG Theora, který podporuje Mozilla se skoro vůbec nepoužívá a není tak kvalitní jako H.264, protože vychází ze starého formátu VP3 od On2 Technologies. U H.264 je zase problém s placenou licenci, i když část poplatků byla uvolněna. On2 Technologies koupil Google a formát VP7 a VP8 (také produkt On2 Technologies, který se používá u stream videa) uvolnil [11].

Podpora v prohlížečích není jednotná a ještě se bude měnit. Nyní formát .ogg podporuje Firefox, Opera, a Chrome. MPEG-4 podporuje Internet Explorer, Chrome, a Safari [9].

V dnešní době jsou podporovány tři formáty videa. Pokud chceme poskytnout video pro více prohlížečů, mělo by být umístěno na webu ve více formátech [12].

Formát	IE	Firefox	Opera	Chrome	Safari
Ogv	Ne	3.5+	10.5+	5.0+	Ne
MPEG 4	9.0+	Ne	Ne	5.0+	3.0+
WebM	Ne	4.0+	10.6+	6.0+	Ne

2.2.2 Audio

Pro přehrávání audia se doposud používaly různé pluginy – ne pouze jeden pro všechny prohlížeče. To by měl u HTML 5 vyřešit element `<audio>`. Atributy jsou zde stejné jako u videa – `controls`, `source` (atribut `width` a `height` zde samozřejmě není). Kodek opět není jednotný – využívá se .ogg, .mp3, a .wav. Přehrávač bude opět přehrávat první jím rozeznatý formát [9].

```
<audio controls="controls">
  <source src="audio_stopa.mp3" type="audio/mpeg" />
  <source src="audio_stopa.ogg" type="audio/ogg" />
  Je nám líto, ale Váš prohlížeč nepodporuje přehrávání zvuku v HTML 5.
</audio>
```

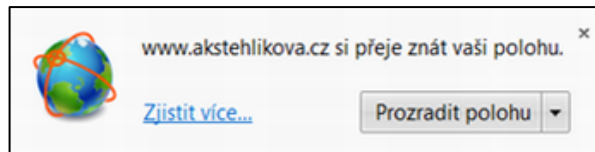
2.2.3 Canvas

Prvek `<canvas>` se používá pro vytváření grafiky do jednoduchých webových her, kreslení grafů a jiných obrázků. Canvas používá JavaScript, pomocí něhož dokáže nakreslit široké spektrum obrázků.

Problematika Canvasu je však velmi rozsáhlá a v této práci nebude probírána.

2.2.4 Geolokace

V dnešní době se na internet přistupuje čím dál více z mobilních zařízení. Pomocí API Geolokace lze zjistit aktuální polohu přímo v prohlížeči. Funkce zjištění polohy umožňuje objekt `navigator.geolocation`. Polohu prohlížeč nezjišťuje sám od sebe. Uživatel tuto možnost musí sám povolit (viz Obrázek 4). Pokud prohlížeč nemá jiný údaj, nežli IP adresu, použije tu. Tento údaj může být velmi nepřesný (například až 100 kilometrů). Má-li zařízení Wi-Fi, Firefox zjistí polohu a sílu signálu okolních Wi-Fi sítí a určí polohu. Mobilní zařízení používají BTS (*Systém základových stanic*) a GPS (*globálního pozičního systému*). Firefox navíc využívá databázi IP adres a Wi-Fi sítí, které potom porovnává [13].



Obrázek 4 – povolení zjištění polohy

Metoda pro získání aktuálních souřadnic pozice je `getCurrentPosition()`. Jednoduchý kód na zjištění polohy:

```
function ukazPolohu(mojePozice) {  
    alert(mojePozice.coords.latitude + ; + mojePozice.coords.longitude);  
}  
if (navigator.geolocation) {  
    navigator.geolocation.getCurrentPosition(ukazPolohu);  
} else {  
    alert("ERR: Prohlizec nepodporuje zjišťování polohy.");  
}
```

Jsou zavedeny dva atributy – `coords` a `timestamp`. Timestamp udává, kdy byly údaje naměřeny. Pomocí `coords` lze zjistit například následující údaje (v předchozím případě bylo použito pouze `latitude` a `longitude`):

- `latitude` – zeměpisná šířka
- `longitude` – zeměpisná délka
- `accuracy` – přesnost
- `speed` – rychlost

Dále lze nastavit zjišťování polohy pomocí přesnějšího zařízení (například u mobilního telefonu pomocí GPS), čas, po který se prohlížeč bude informace snažit zjistit a ukládání informací o zjištěné poloze. Toto API podporuje Firefox od verze 3.5, Chrome od verze 5.0, IE 9 a Opera od verze 10.6 [5].

2.2.5 Formuláře

Málokterá stránka se obejde bez toho, aby si zjišťovala od uživatelů nějaká data. Často je snaha od potenciálních zákazníků získat různé údaje – například kontaktní informace jako e-mail, telefon či adresu bydliště. Nebo je potřeba jen na dané stránce zřídit účet na jméno a heslo. K tomu potřebujeme data ukládat do databáze. Aby vše fungovalo tak, jak má, je potřeba ukládat jen ta data, která potřebujeme a samozřejmě data validní. To znamená, že pole ve formulářích se musí kontrolovat, abychom do databáze uložili opravdu jen to, co uložit chceme.

HTML 5 přidává do formulářů nové atributy, které zajišťují, že data z formuláře budou validována a odpadne používání JavaScriptu. Pro starší prohlížeče, které nezajišťují podporu nových atributů, se atributy budou tvářit jako klasický `input type="text"`.

Majitelům chytrých mobilních telefonů s dotykovým displejem nabídne při zadávání vstupních hodnot tu nejlepší možnost pro zadávání – upraví klávesnici.

Příklad použití nových vlastností ve formuláři

```
<form action="zpracuj.php" method="post">
Email:<br /> <input type="email" placeholder="Váš e-mail" required="true"
name="email" value="" /><br />
<input type="submit" value="Odeslat" />
</form>
```

Nejzajímavější nové atributy [14]

autocomplete

Předvyplňuje formuláře za nás. Autocomplete sděluje, zda má nebo nemá formulářové pole předvyplnit.

autofocus

Při návštěvě některých stránek (například Google) se kurzor ihned po načtení stránky přesune na element, který obsahuje atribut autofocus (na Googlu do pole pro vyhledávání).

Toto se však některým uživatelům, kteří používají při pohybu na webu klávesnici, nemusí příliš líbit. Použití autofocusu jim znemožní vrátit se zpět na stránku, ze které přišli pomocí klávesy backspace nebo posunout stránku pomocí mezerníku níže.

placeholder

Text v poli, který slouží jako nápověda, a hned po kliknutí na toto pole zmizí.

required

Určuje, že pole je povinné. Je-li vstupní hodnota prázdná nebo neplatná, formulář se neodešle. Kurzor se potom přesune na první chybně vyplněné pole.

Nové vstupní typy (input type)

Typy pro kalendář

Po kliknutí na dané vstupní pole by se měl zobrazit kalendář a dát na výběr datum, podobně jako na pracovní ploše Windows. Vstupní typy jsou: `date`, `month`, `week`, `time`, `datetime`, `datetime-local`.

Number

Pro zadávání čísel se zobrazí šipka nahoru a dolů a pro mobilní zařízení se zobrazí číselná klávesnice. Atributem `min` a `max` se nastaví minimální a maximální hodnota, kterou lze zadat. Atribut `step` určuje, o kolik se má hodnota při kliknutí myši zvětšit nebo zmenšit.

Range

Posuvník sloužící pro zadávání čísel.

Email

Pole pro email, kde se kontroluje, zda je e-mail ve správném tvaru. Mobilní zařízení nabídnou klávesnici s tlačítkem zavináče.

Search

Tlačítko hledání.

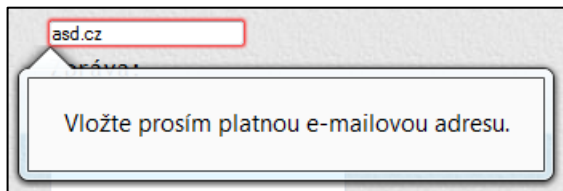
Color

Pro formulář, kde chceme nechat uživatele vybrat barvu.

Nové vstupní typy ještě nejsou zcela dobře podporovány, takže o dobrém praktickém využití se zatím mluvit bohužel nedá.

Chybové zprávy u formulářů

Při vyplnění (nebo naopak nevyplnění, když je údaj povinný) pole chybnou hodnotou se objeví chybová zpráva (viz *Obrázek 5*). Obsah chybových zpráv i jejich vzhled lze u formulářů měnit [4].



Obrázek 5 – chybová zpráva

3. CSS 3

3.1 Úvod do CSS 3

Každá webová stránka má svůj obsah, který má určitou formu. CSS (*Kaskádové styly*) je jazyk pro popis způsobu, jakým se má HTML, XHTML nebo XML dokument zobrazit. Jazyk odděluje obsahovou část od té, která definuje podobu stránky. Tím, že se podoba striktně oddělila od obsahu, se dosáhlo lepšího zpracovávání a vyhledávání dokumentů. Nemluvě o větší přehlednosti kódu a snazší změně vzhledu stránek [15].

CSS bylo navrženo organizací W3C a poslední verze je CSS 2.1, která byla dokončena v roce 2011. Díky této verzi lze dosáhnout grafických prvků, se kterými se setkáváme běžně na webu. Díky CSS 2 lze nastavit umístění a velikost částí dokumentu, zvolit barevné pozadí, vkládat na stránky obrázky a podobně. Postupem času tyto funkce přestávaly stačit. Módou se staly kulaté rohy, stíny a další grafické novinky. Bohužel pomocí CSS 2 toho nelze dosáhnout. Tyto prvky se musí pro CSS 2 vytvářet v grafických editorech. Jejich výstupem jsou obrázky, které se vkládají na stránky. To zvyšuje velikost přenesených dat do prohlížeče návštěvníka stránky. Tvorba těchto prvků také vyžaduje od autora stránek více času stráveného nad projektem. Dalším příkladem může být počet fontů použitelných na webu, který je velmi omezený. Tohle je jen zlomek toho, co se nyní od internetových stránek vyžaduje. Web přestává být statický a dostává se do něj čím dál tím více animací. Animace jsou tvořeny nejčastěji pomocí Flashe. Jedná se tedy o další technologii, kterou musí tvůrce internetových stránek ovládat. Odpovědí na tyto problémy by měla být třetí verze Kaskádových stylů. Co tedy nového CSS 3 umožňuje? Zavádí novinky jako kulaté rohy, vržený stín, přechod jedné barvy do druhé či spolu s HTML 5 možnost tvorby animací.

3.2 Syntaxe CSS a selektory

CSS 3 má stejnou syntaxi zápisu jako starší verze. Styl lze zapsat přímo do hlavičky stránky mezi tagy `<style>` a `</style>`. Druhou možností je zápis stylu do souboru. Je potřeba vytvořit soubor (v tomto příkladu `reset.css`), do kterého se styl bude zapisovat a do hlavičky stránky uvést k souboru správnou cestu:

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

Každý prvek, kterému se přiřadí určitá vlastnost, musí obsahovat zápis, kterým na prvek odkážeme. Nebo se lze odkázat i přímo, ale potom se nastaví vlastnost pro všechny prvky stejného typu (například pro všechny nadpisy úrovně `h1`). Odkazy jsou třídy a identifikátory – `class` a `id`. Třidu můžeme použít pro více prvků najednou. Identifikátory jsou určeny pouze pro jeden prvek na stránce. Například:

```
h1{ color: red;} /* platí pro všechny nadpisy h1 */
#menu{color: green;} /* identifikátor*/
.text{color: white;} /* třída */
```

Pomocí tříd a identifikátorů lze nastavit pro každý prvek jinou vlastnost.

3.2.1 Odkazování pomocí hodnoty atributů

V CSS 3 lze prvky nově cílit pouze podle části atributů:

- `E[att^="val"]` – atribut může na určitou hodnotu začínat.
- `E[att$="val"]` – atribut končí na danou hodnotu.
- `E[att*="val"]` – obsahuje hodnotu.

`E` je element a `val` je hodnota atributu.

3.2.2 Odkazování pomocí pseudotříd

Pseudotřídy byly zavedeny již ve starší verzi Kaskádových stylů. Používají se například na efekty u odkazů po přejetí myši (`a:hover`). Do CSS 3 jich však přibylo mnohem více.

- `E:root` – odkazuje na kořenový adresář (v HTML to je `<html>`).
- `E:empty` – vybere prázdný element (například `<h1></h1>`).
- `E:target` – odkazuje na prvek, který je cílem `<a>` uvnitř stránky.
- `E:enabled` a `E:disabled` – zacílí prvek u formuláře, kde je povoleno/ zakázáno zapisovat.
- `E:checked` – zaškrtnuté pole u formulářů.

3.2.3 Odkazování podle pořadí

Odkazováním dle pořadí prvků je možné zacílit prvek, který je určitým potomkem.

- `E:last-child`, `E:only-child` – vybere posledního, jediného potomka prvku. V CSS 2 již je definován `E:first-child` (například první položka v seznamu).
- `E:first-of-type`, `E:last-of-type`, `E:only-of-type` – prvek, který je první, poslední, jediný svého typu.

3.2.4 Odkazování pomocí matematických operací

Pomocí parametru „`n`“ lze odkazovat na prvek v určitém pořadí. Tím například odpadne nutnost přiřazovat každému řádku tabulky nějakou třídu. Jednoduše se odkáže na `n`-tý prvek.

- `E:nth-last-child(n)`
- `E:nthchild(n)`

- `E:nth-of-type(n)`
- `E:nth-last-of-type`

„n“ může být jakékoli číslo nebo i matematická operace. Například `E:nthchild(1)` – bude vyhovovat každý prvek. `E:nthchild(3n)` – vyhovuje třetí, šestý, devátý... Číslování pro „n“ začíná od nuly, takže `E:nthchild(2n+2)` bude vyhovovat druhému, čtvrtému, šestému (atd.) prvku [6].

Tento seznam není zdaleka vyčerpávající. S podporou nových selektorů mají problém starší verze IE. Jak prohlížeč podporuje selektory lze ověřit na adrese <http://tools.css3.info/selectors-test/test.html>.

3.3 Vendor prefixy

Většina prvků ještě není hotová a může se změnit. Tento fakt je v prohlížečích vyjádřen „vendor prefixem“. Neboli prvek, který ještě není hotový, obsahuje prefix. Například kulaté rohy:

```
-webkit-border-radius: 20px;
-moz-border-radius: 20px;
border-radius: 20px;
```

Mozilla má prefix `-moz-`, prohlížeče na jádru Webkit jako Safari, Google Chrome mají `-webkit-`, Opera `-o-` a IE (od verze 9) `-ms-`.

Až budou konkrétní vlastnosti hotové, prefixy budou postupně mizet. Některé prohlížeče již u některých vlastností prefixy neuvádí. V tomto příkladu prohlížeče Safari, IE 9, Chrome, Opera a Firefox 4 už prefix odstranily [16]. Pokud je použit prvek s prefixem, tak by měl být vždy uveden i zápis bez prefixu. Tím se zajistí kompatibilita do budoucna [6].

3.4 Nové vlastnosti CSS 3

3.4.1 Nové fonty

Vždy, když bylo potřeba zvolit na web nějaký font, bylo jejich množství velmi omezené. Většinou se vytvářely odkazy a loga pomocí obrázků. Při změně designu bylo v tomto případě nutné obrázky vytvořit celé znovu [6]. Další možností bylo obejít tento problém použitím nějakého JavaScriptu. CSS 3 umožňuje použít vlastní písmo pomocí zápisu `@font-face` (viz *Obrázek 6*). Podpora v prohlížečích je velmi dobrá, nové fonty podporuje většina prohlížečů. Je jen třeba poskytnout více souborů a prohlížeč si vybere ten, kterému rozumí. Písmo se vykresluje na různých operačních systémech trochu odlišně, ale většinou vypadá hezky [17].

Když chceme přidat nový font, je třeba:

- Příslušný font nahrát v různých formátech na server.
- Pomocí `@font-face` pojmenovat a uvést cestu fontu.

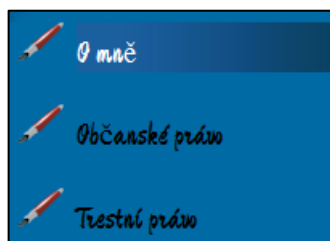
- Použít font „postaru“ ve `font-family` [6].

Problém může být s licenci. Každé písmo nelze ve `@font-face` použít. Když se písmo umístí na server a použije se na stránkách, tak každý, kdo stránku navštíví, si font automaticky stahuje. I když je font koupený, je tímto způsobem je nabízen třetí osobě. Proto je třeba (pokud nechceme porušovat licence) vybírat fonty, které jsou licenčně uvolněné. Existují stránky, kde lze fonty, které jsou zdarma, stáhnout. Pro získání více formátů se může použít generátor (<http://www.fontsquirrel.com/fontface/generator>). Pokud chceme použít nějaký placený font, můžeme využít hostingů, kterým platíme za použití písma. Nepříjemnosti můžu nastat s češtinou – písma nemusí obsahovat českou diakritiku [16].

Použití písma:

```
@font-face {
  font-family: 'devonshire';
  src: url(../fonts/devonshire-regular-webfont.eot);
  src: url(../fonts/devonshire-regular-webfont.svg);
  src: url(../fonts/devonshire-regular-webfont.svgz);
  src: url(../fonts/devonshire-regular-webfont.ttf);
  src: url(../fonts/devonshire-regular-webfont.woff);
}

nav a, nav a:visited {
  font-family: devonshire;
}
```



Obrázek 6 – nový font

Při používání nového druhu písma se uvádí nějaký základní font, kdyby se nový font z nějakého důvodu nenačetl.

3.4.2 Barvy

Barvy se mohou zapsat pomocí `rgb` (`rgb(255, 255, 255)`) nebo hexadecimálně (`#ffffff`). Dalším způsobem je využít pojmenované barvy jako `red`, `green`, `yellow` atd. Nově je seznam pojmenovaných barev rozšířen o dalších 147 barev a přibýly další možnosti, jak definovat barvy: `RGBA`, `HSL` a `HSLA`.

RGBA

RGBA je klasický model RGB (red, green, blue) rozšířený o čtvrtou složku A – alfa, která slouží k zapsání průhlednosti. Má rozsah 0 – 1, přičemž 0 je zcela průhledná a 1 zcela neprůhledná. Například černá barva s 50% průhledností se zapíše takto:

```
rgba(0, 0, 0, 0.5)
```

Před tento zápis je třeba uvádět i zápis pro starší prohlížeče, protože ty tento zápis neznají. Nový prohlížeč vykreslí starší zápis, který překreslí novým zápisem. Starší prohlížeč bude rgba ignorovat.

HSL

V HSL (*hue, saturation, lightness*) se mění odstín, sytost a světlost.

- Odstín se zadává od 0 do 359, kdy 0 je červená, 120 zelená a 240 modrá.
- Sytost 0% až 100%, 100% je sytá barva.
- Světlost 0% až 100%, kdy 0 je černá barva a 100 bílá. 50% je „normální“ barva.

HSLA přidává jako rgba čtvrtou složku průhlednosti, která má hodnoty v intervalu od nuly do jedné:

```
hsla(100, 100%, 50%, 0.5)
```

Možnost vyjádřit průhlednost lze i mimo modely rgba a hsla – pomocí vlastnosti `opacity` [6].

3.4.3 Border-radius

V CSS 2.1 bylo obtížné vytvořit kulaté rohy. Aby však bylo možné kulaté rohy vytvořit, bylo potřeba použít buď nějaký skript, nebo vytvořit kulaté rohy pomocí obrázků. To se neobešlo bez několika `divů` navíc. Tím však byl kód delší a méně přehledný. Nově lze kulaté rohy zapsat jednoduše:

```
border-radius: 10px;
```

Samozřejmě nesmíme zapomenout na prefixy. Tímto zápisem nastavíme všem rohům zakulacení 10 pixelů. Zakulacení však můžeme nastavit pro každý roh jině. Potom budeme mít čtyři hodnoty. Pouze Mozilla vyžaduje zapsat každý roh zvlášť – `topleft` pro levý horní roh, `topright` pro horní pravý atd [18].

3.4.4 Box-shadow

Umožňuje přidělit prvku vržený stín. Prvek, kterému se stín přidělí, potom vypadá jako by vystupoval ze stránky a docílí se tím 3D efektu [16].

```
-webkit-box-shadow: inset -50px 20px 5px 10px #000000;  
-moz-box-shadow: inset -50px 20px 5px 10px #000000;  
box-shadow: inset -50px 20px 5px 10px #000000;
```

`Insert` znamená, že stín bude uvnitř prvku. Jeho vynecháním se vykreslí okolo. První hodnota potom znamená posunutí stínu po ose x – záporná hodnota posouvá doleva a kladná doprava, druhá po ose y, třetí hodnota udává velikost rozmazání stínu a čtvrtá vzdálenost, do které se má stín vykreslit. Poslední je barva, kterou můžeme zapsat buď hexadecimálně nebo i ve formátu RGBA.

3.4.5 Text-shadow

Přidává podobně jako `box-shadow` stín, ale tentokrát textu.

```
text-shadow: 2px 2px 5px #000000;
```

První hodnota posune stín po ose x, druhá po y, třetí udává rozmazání stínu a čtvrtá barvu. Stín funguje v nejnovějších prohlížečích kromě IE 9.

3.4.6 Gradients

Další prvek, kterého je dosaženo pouze pomocí obrázku, je přechod mezi barvami. Pomocí gradientů můžeme tedy přecházet mezi dvěma nebo více barvami. K dispozici jsou dva typy gradientů: lineární a radiální.

Lineární gradienty

U lineárního gradientu se jedná o přechod po přímce. Nastaví se směr a barvy, od které, po kterou se má přejít (případně barvy další).

U gradientů je však problém s podporou.

Syntaxe dle CSS 3:

```
background-image: linear-gradient(250deg, #FFFFFF 0%, #006aa4 100%);
```

První hodnota znamená směr (úhel) gradientu. Druhá počáteční barvu přechodu a procento vyjadřuje, odkud bude barva začínat. Další hodnota je potom pro konečnou barvu. Místo úhlu lze použít také `top`, `left`, `right` nebo `bottom`. Úhel lze také vynechat. Gradient potom bude začínat od shora. I procenta přechodu se mohou vynechat. Potom bude defaultně nastavena hodnota 0 % pro první a 100 % pro druhou barvu.

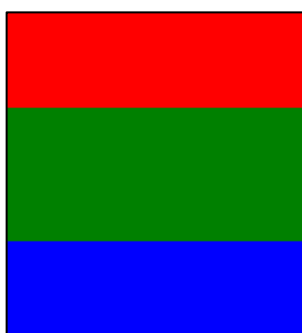
Další barva se přidá jednoduše:

```
background-image: -moz-linear-gradient(45deg, #FFFFFF 0%, #006aa4 60%, #000000 100%);
```

Je však potřeba nastavit procenta pro přechod. Pokud se vynechají, přechod bude rovnoměrně plynulý.

Přechod nemusí být plynulý, ale může být ostrý (viz *Obrázek 7*). Tam kde jedna barva bude končit, bude začínat jiná:

```
background: -moz-linear-gradient(top, #ffffff 20%, #006aa4 20%, #006aa4 40%, #000000 40%);
```



Obrázek 7 – ostrý přechod

Prohlížeče založené na jádru WebKit přišly s gradienty již dříve. Nové prohlížeče podporují novou syntaxi CSS 3. Pro ty starší je třeba uvést jinou syntaxi. Pro starší prohlížeče IE a Opera existují proprietární filtry. Lze však použít pouze dvě barvy a přechod může být pouze vodorovný nebo svislý. IE verze 9 a Opera 11.01 podporují SVG (*škálovatelná vektorová grafika*) pozadí [6].

Aby se docílilo použití gradientů v širokém spektru prohlížečů, je potřeba opravdu mnoho kódu. To by stálo hodně času. Existují však naštěstí nástroje, které kód jednoduše vygenerují. Například: <http://www.colorzilla.com/gradient-editor/>. Grafické návrhy stránek se většinou vytvářejí v programech, jako je Adobe Photoshop. Nástroje, které potřebný kód vygenerují, mají podobné uživatelské rozhraní jako právě Photoshop. Lze se tedy v grafickém návrhu podívat na nastavení přechodu a pouze ho vyplnit do nástroje pro gradient. Pokud je použit pouze vertikální přechod o dvou barvách a kód bude vygenerován, bude vypadat následovně:

```
background: #ffffff; /* Old browsers */
/* IE9 SVG, needs conditional override of 'filter' to 'none' */
background:
url(data:image/svg+xml;base64,PD94bWwgdmlld0JveD0iMCAwIDEgMSIgcHJlc2VydmVBc3B1Y3RSYXRpbz0ibm9uZSI+CiAgPGxpbmVhckd
```

```

yYWRpZW50IGlkPSJncmFkLXVjZ2ctZ2VuZXJhdGVkIiBncmFkaWVudFVuaXRzPSJlc2VyU3BhY2
VPblVzZSIgeDE9IjAlIiB5MT0iMCUiIHgyPSIwJSIgeTI9IjEwMCUiPgogICAgPHN0b3Agb2Zmc
2V0PSIwJSIgc3RvcC1jb2xvcj0iI2ZmZmZmZiIgc3RvcC1vcGFjaXR5PSIxIi8+CiAgICA8c3Rv
cCBvZmZzZXQ9IjEwMCUiIHN0b3AtY29sb3I9IiMwMDZhYTQiIHN0b3Atb3BhY2l0eT0iMSIvPgo
gIDwvbGluZWFyR3JhZGllbnQ+CiAgPHJlY3QgeD0iMCIgeT0iMCIgd2lkdg9IjEiIGhlaWdodD
0iMSIgZmlsbD0idXJsKCNcmFkLXVjZ2ctZ2VuZXJhdGVkKSIgLz4KPC9zdmc+);

background: -moz-linear-gradient(top,  #ffffff 0%, #006aa4 100%);
/* FF3.6+ */

background: -webkit-gradient(linear, left top, left bottom, color-
stop(0%,#ffffff), color-stop(100%,#006aa4));
/* Chrome,Safari4+ */

background: -webkit-linear-gradient(top,  #ffffff 0%,#006aa4 100%);
/* Chrome10+,Safari5.1+ */

background: -o-linear-gradient(top,  #ffffff 0%,#006aa4 100%);
/* Opera 11.10+ */

background: -ms-linear-gradient(top,  #ffffff 0%,#006aa4 100%);
/* IE10+ */

background: linear-gradient(top,  #ffffff 0%,#006aa4 100%);
/* W3C */

filter: progid:DXImageTransform.Microsoft.gradient(
startColorstr='#ffffff', endColorstr='#006aa4',GradientType=0 );
/* IE6-8 */

```

Radiální gradienty

U radiálního gradientu je přechod barev od jednoho bodu do všech stran. Radiální gradienty jsou podporovány Firefoxem od verze 3.6. Opera 11.10 podporuje pouze lineární – radiální je možné vytvořit pomocí SVG (Platí i pro IE9). Do dřívějších verzí IE radiální gradient nelze dostat ani pomocí filtrů [6].

Opakování gradientů

Gradienty na pozadí můžeme také opakovat a vytvořit tak pruhované pozadí (viz *Obrázek 8*). Lze opakovat lineární, radiální i oba dohromady. Například:

```
background: -moz-repeating-linear-gradient(top,  #ffffff 20%, #006aa4 30%);
```

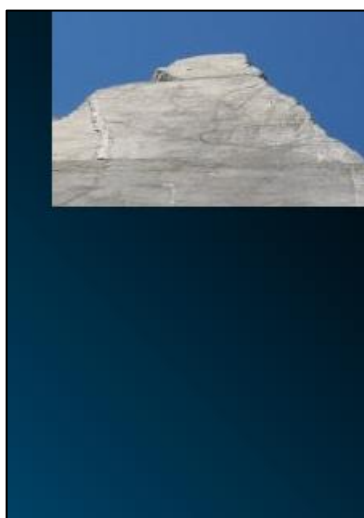


Obrázek 8 – opakování gradientu

3.4.7 Vícenásobné pozadí

Pozadí se v CSS 3 může kombinovat. Lze na pozadí nastavit více obrázků a to i s použitím gradientů (viz *Obrázek 9*). Zápisy pro jednotlivé obrázky se pouze oddělují čárkami. Obrázky se skládají na sebe tak, že obrázek uvedený prvně, je nahoře.

```
background: url(..../images/obrazek.jpg) no-repeat top right,  
-moz-linear-gradient(45deg, #FFFFFF 0%, #006aa4 60%, #000000 100%);
```



Obrázek 9 – vícenásobné pozadí

3.4.8 Transformace a posun prvku

Pomocí transformace lze stránky udělat dynamičtější. Tyto funkce můžeme především využít při tvorbě odkazů. Daný prvek můžeme zkosit, natočit, zvětšit či posunout.

Příklad: Zvětšení navigace:

```
nav a:hover {  
    -webkit-transform: scale(1.2, 1.2);
```

```

-moz-transform: scale(1.2, 1.2);
-ms-transform: scale(1.2, 1.2);
-o-transform: scale(1.2, 1.2);
-transform: scale(1.2, 1.2);
}

```

3.4.9 Media Queries

Na webové stránky dnes přistupuje velké množství zařízení. Z nezanedbatelné části to jsou zařízení s menšími obrazovkami. Jedná se o netbooky, různé tablety a mobilní telefony. Je dobré stránky pro tyto zařízení přizpůsobit. Pomocí Media Queries můžeme určit, jak bude pro jednotlivé zařízení stránka vypadat. Neboli, pro ně napsat samostatný styl.

```

<link rel="stylesheet" href="normal.css" media="screen and (min-width:
1100px)">

```

```

<link rel="stylesheet" href="telefon.css" media="screen and (max-device-
width: 480px)">

```

Media Queries lze používat i v samotném stylu CSS:

```

@media screen and (max-device-width: 480px) { nějaká hodnota }

```

Podpora u desktopových prohlížečů je poměrně dobrá: Opera verze 7 a novější, Firefox 3.5 a novější, Safari od verze 3, Chrome a IE od verze 9. U mobilních zařízení je to Opera Mobile, Opera Mini a mobilní Webkit.

Jsou dva způsoby, jak grafiku stránek řešit. Pro hotový web lze použít speciální zápis pomocí Media Queries a vytvořit pro ně speciální styl. Mobilní prohlížeče, které nepodporují funkci Media Queries, budou stránku zobrazovat tak, jak nechceme (načtou špatný styl). Toto je jednoduché a rychlé řešení. Nebo při tvorbě nových stránek můžeme vytvořit nějaký základní styl pro všechna zařízení. Nebudeme používat složité grafické prvky, ale pouze základní prvky a co nejméně obrázků. Pro zařízení s většími zobrazovacími zařízeními lze použít vlastní styl, kde už budou použita různá vylepšení. Ve výsledku tedy velká zobrazovací zařízení použijí styl zapsaný dle Media Queries. Ostatní použijí základní styl. Pro IE starší verze nežli IE 9 bude navíc potřeba jednoduchá podmínka [16]:

```

<!--[if lt IE 9]>
<link rel="stylesheet" type="text/css" href="normal.css">
<![endif]-->

```

3.5 Starší prohlížeče

Nové funkce starší prohlížeče samozřejmě nepodporují. Když prohlížeč narazí na neznámý prvek, jednoduše ho ignoruje. Zůstává otázkou, co s tím. Zda řešit, aby v každém prohlížeči stránky vypadaly zcela identicky. Jak tedy tento problém řešit? Můžeme stránky nechat zobrazit správně pouze novým prohlížečům. Druhou možností je nové funkce

nepoužívat a místo nich stylový prvek udělat pomocí obrázku. Tím se budou stránky načítat déle a kód už nebude tak přehledný. Nebo je zde třetí možnost – použití JavaScriptu, který funkci umožní. U každého projektu zůstává na jeho tvůrci, jak si s problémem poradí. Použitím novinek CSS 3 zaručíme, že se stránka zobrazí rychle s grafickými vylepšeními. Při použití JavaScriptu pro starší prohlížeče se stránka bude déle načítat. Také musíme vzít v úvahu čas strávený nad jednotlivým projektem. Zda je potřeba mít grafickou podobu ve všech prohlížečích v každém detailu stejnou, anebo jestli nám postačí některé prvky nechat bez například kulatých rohů nebo přechodu. Při aplikování JavaScriptu na celé stránky může dojít k nepříjemně dlouhému načítání stránek [16].

3.5.1 Vytvoření jednoduchého tlačítka s efekty CSS 3

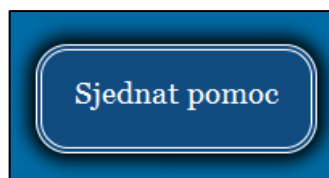
Cílem je vytvořit tlačítko pomocí efektů CSS 3 (viz *Obrázek 10*). Napiše se tedy jednoduchý kód a do html stránky se vloží:

```
<a href="kontakt.html">Sjednat pomoc</a>
```

Do stylu:

```
html body div#container footer a {
    width: 167px;
    height: 60px;
    float: left;
    margin: 0 20px auto 20px;
    background-color: #104b80;
    border: 4px double;
    color: #ffffff;
    font-size: 14pt;
    font-weight: normal;
    text-decoration: none;
    font-family: georgia;
    text-align: center;
    line-height: 50px;
    -webkit-border-radius: 20px;
    -moz-border-radius: 20px;
    border-radius: 20px;
    -webkit-box-shadow: 0px 0px 10px 5px #000000;
    -moz-box-shadow: 0px 0px 10px 5px #000000;
    box-shadow: 0px 0px 10px 5px #000000;
}
```

Protože píšu stránku v HTML 5, mohu odkazu `<a>` přiřadit téměř cokoliv. Odpadla mi nutnost mu přidělovat další `<div>`. Tímto jsem prvku přidělil zakulacené rohy a stín. Odkaz bude vypadat takto:



Obrázek 10 – tlačítko v CSS 3

Ale co podpora v prohlížečích? Podpora kulatých rohů je ve Firefoxu od verze 3.5, ve Chromu od 4.0, Safari 3.1 a v Opeře 10.5 [19]. Hodně uživatelů používá IE, který poskytuje tyto vlastnosti až od verze 9. Ve starších verzích se tlačítko zobrazí bez stínu a kulatých rohů. Vše se naštěstí nemusí dodělovat pomocí obrázků. Existuje několik JavaScriptů, které tyto vlastnosti zprovozní ve starších prohlížečích IE. Jedním řešením je CSS 3 PIE (<http://css3pie.com/>). Stačí script stáhnout a do stylu připsat pouze:

```
-pie-border-radius: 20px;  
-pie-box-shadow: 0px 0px 10px 5px #000000;  
behavior: url(pie/PIE.htc);
```

Takto tlačítko získá stín i kulaté rohy. Já jsem měl problém s tím, že tlačítko sice mělo kulaté rohy i stín, ale oboje se vykreslilo až za pozadí. S ostatními prvky jako hlavička nebo patička stránky mi PIE fungoval bez problému. PIE umožní používat funkce CSS 3 pro IE 6 až 8. Poskytuje: `boreder-radius`, `box-shadow`, `border-image`, `linear-gradient` jako obrázek na pozadí a více obrázků na pozadí.

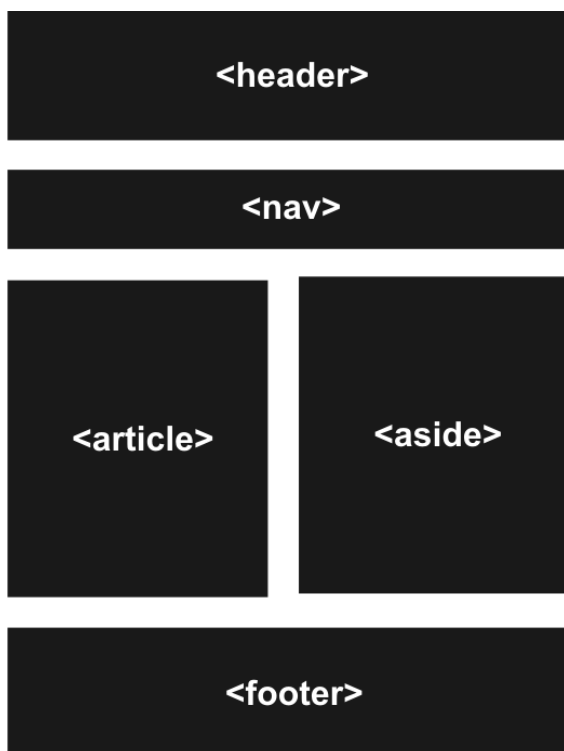
Praktická část

V praktické části bych chtěl upravit stávající stránky tak, aby využívaly nových funkcí HTML 5 a CSS 3 a přitom byly přístupné pro širší spektrum prohlížečů a nejen pro nejnovější verze. Celé stránky (staré i nové) se nacházejí na přiloženém CD.

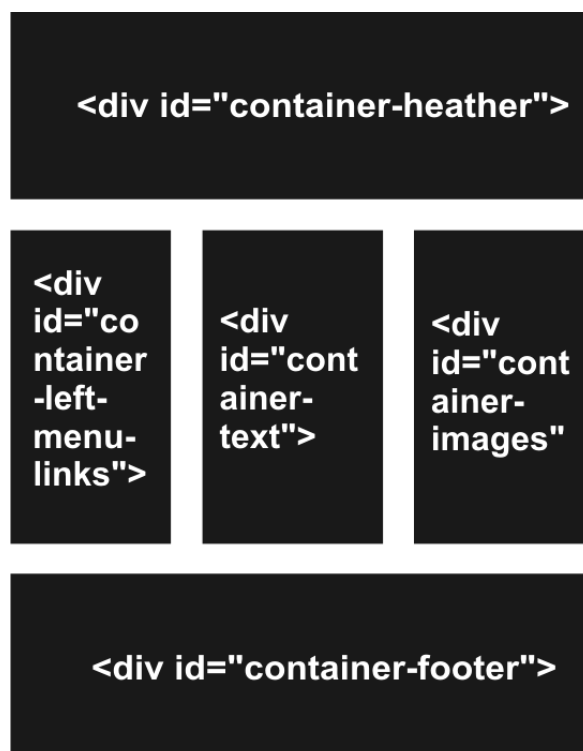
4. Struktura dokumentu

Do HTML 5 přibylo několik nových elementů. Po přechodu na HTML 5 dokument zůstává validní a prohlížeč ho dokáže zobrazit. Chtěl jsem využít nových funkcí, takže jsem dokument přepsal tak, jak by měl dokument psaný v HTML 5 vypadat (viz *Obrázek 11*). Nový kód je díky novým elementům kratší, přehlednější a měl by být lépe indexován. Kód staré a nové titulní strany webu se nachází v příloze.

4.1 Stará a nová struktura dokumentu



Obrázek 11 – nová struktura



Obrázek 12 – stará struktura

Co je nového na samotném kódu? Nejdříve zkrácený DOCTYPE – `<!DOCTYPE html>`, který určuje, že se jedná o HTML 5.

Poté jsem přidal sémantické elementy `<nav>`, `<aside>`, `<header>`, `<footer>`, `<article>` a `<section>`. Nadpis „JUDr. Eva Stehlíková, právní služby“ doposud nenesl

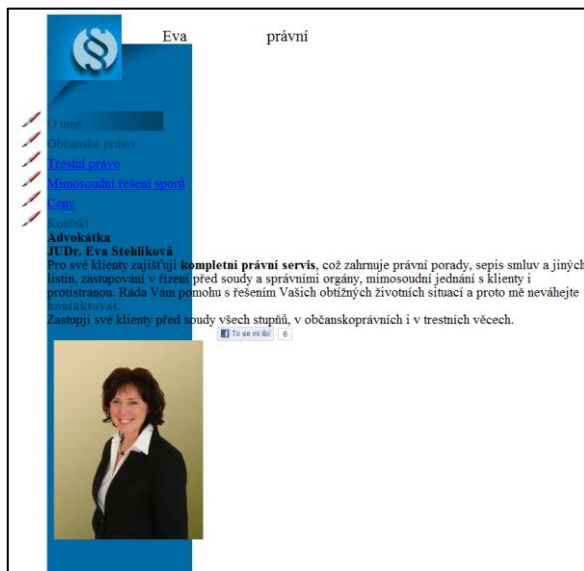
pro vyhledávače žádný sémantický význam – elementem `<h1>` byl obalen další nadpis na stránce. Nyní se však nadpis ocitl v samostatné části dokumentu, takže nadpisu lze přidělit `<h1>`.

4.2 Problém s Internet Explorerem

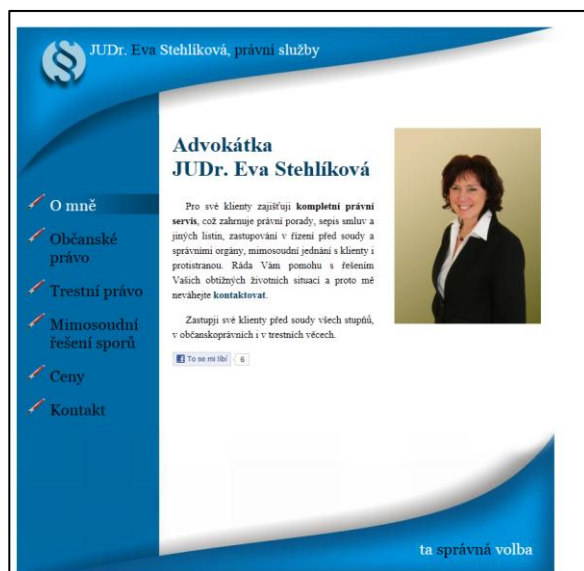
Nastal však problém se staršími prohlížeči. Layout stránky se zcela rozsypal (viz *Obrázek 13*). Bylo potřeba, aby se elementy jako `<header>`, `<footer>` atd. zobrazovaly blokově. Vytvořil jsem nový soubor pro styly, který jsem nazval `reset.css`. Do něj jsem vložil následující kód:

```
article, aside, details, figcaption, figure,
footer, header, hgroup, menu, nav, section {
    display: block;
}
```

Bohužel prohlížeče Internet Explorer starší nežli verze 9, elementy ignorovaly. Bylo potřeba přidat JavaScript, který prvek na stránce vytvoří. IE potom element bude chápat klasicky a přiřadí mu daný styl (blokové zobrazení). Zápis `<!--[if lt IE 9]>` je podmíněný komentář. Říká, že jiný prohlížeč, nežli prohlížeče starší než IE 9, podmínku přeskočí. JavaScript jsem stáhl z internetové adresy <http://remysharp.com/2009/01/07/html5-enabling-script/>. Script lze napsat ručně, ale tenhle řeší všechny HTML 5 elementy najednou.



Obrázek 13 – před přidáním skriptu v IE 8



Obrázek 14 – po přidání skriptu v IE 8

Stránka se vykreslila správně v prohlížečích IE 6 a novějších (viz *Obrázek 14*). Pro ostatní prohlížeče není skript potřeba – stačí styl.

Další užitečná adresa je: <http://html5doctor.com/html-5-reset-stylesheet/>, kde se nachází styl, který umožní starším prohlížečům zobrazovat prvky z HTML 5 korektně. Je to

vlastně rozšířený původní styl `reset.css`. Stažený styl jsem zkopíroval do stylu `reset.css`, takže nyní obsahuje následující kód:

```
/*
html5doctor.com Reset Stylesheet
v1.6.1
Last Updated: 2010-09-17
Author: Richard Clark - http://richclarkdesign.com
Twitter: @rich_clark
*/

html, body, div, span, object, iframe,
h1, h2, h3, h4, h5, h6, p, blockquote, pre,
abbr, address, cite, code,
del, dfn, em, img, ins, kbd, q, samp,
small, strong, sub, sup, var,
b, i,
dl, dt, dd, ol, ul, li,
fieldset, form, label, legend,
table, caption, tbody, tfoot, thead, tr, th, td,
article, aside, canvas, details, figcaption, figure,
footer, header, hgroup, menu, nav, section, summary,
time, mark, audio, video {
    margin:0;
    padding:0;
    border:0;
    outline:0;
    font-size:100%;
    vertical-align:baseline;
    background:transparent;
}

body {
    line-height:1;
}

article,aside,details,figcaption,figure,
footer,header,hgroup,menu,nav,section {
    display:block;
}

nav ul {
    list-style:none;
}

blockquote, q {
    quotes:none;
}

blockquote:before, blockquote:after,
q:before, q:after {
    content:'';
    content:none;
}

a {
    margin:0;
    padding:0;
    font-size:100%;
    vertical-align:baseline;
```

```

        background:transparent;
    }

    /* change colours to suit your needs */
    ins {
        background-color:#ff9;
        color:#000;
        text-decoration:none;
    }

    /* change colours to suit your needs */
    mark {
        background-color:#ff9;
        color:#000;
        font-style:italic;
        font-weight:bold;
    }

    del {
        text-decoration: line-through;
    }

    abbr[title], dfn[title] {
        border-bottom:1px dotted;
        cursor:help;
    }

    table {
        border-collapse:collapse;
        border-spacing:0;
    }

    /* change border colour to suit your needs */
    hr {
        display:block;
        height:1px;
        border:0;
        border-top:1px solid #cccccc;
        margin:1em 0;
        padding:0;
    }

    input, select {
        vertical-align:middle;
    }

```

Nyní se stránka zobrazuje v prohlížečích tak, jak má.

5. Nový vzhled

Současný design stránek je tvořený z velké části obrázky, což není zcela vhodné. Nejprve bylo potřeba upravit HML kód, kvůli jinému rozmístění prvků na stránce, a bylo třeba přepisovat styl v souboru `style.css`. Rozhodl jsem se napsat jeden styl pro všechna zařízení a druhý pro zařízení se zobrazovacím zařízením větší, než 900 pixelů, který jsem nazval `normal.css`. Do základního stylu jsem zapisoval společné hodnoty a do druhého pouze prvky, s kterými by mohly mít nedesktopové prohlížeče problém. Speciální vzhled se zapíše pomocí atributu `media`. V hlavičce stránky tedy zápis vypadá takhle:

```
<link rel="stylesheet" href="style/normal.css" media="screen and (min-width: 900px)">
```

S tímto zápisem má problém IE starší verze, nežli verze 9. Mohl bych použít podmíněný komentář:

```
<!--[if lt IE 9]>
    <link rel="stylesheet" type="text/css" href="style/normal.css">
<![endif]-->
```

Je, ale otázkou, jestli se vyplatí řešit design do detailu pro starší prohlížeče obecně. Kód by byl zbytečně delší a neobešel bych se bez velkého množství skriptů. Dle mého názoru postačí, když starší prohlížeče zobrazí stránku se základními grafickými prvky. To je zajištěno již uvedeným zápisem:

```
<link rel="stylesheet" href="style/normal.css" media="screen and (min-width: 900px)">
```

Když prohlížeč nerozumí atributu `media`, celý element ignoruje a stránka se načte pouze se stylem `style.css`. To platí i v případě, že má uživatel nízké rozlišení monitoru. V opačném případě stránka bude čerpat i styl ze souboru `normal.css`.

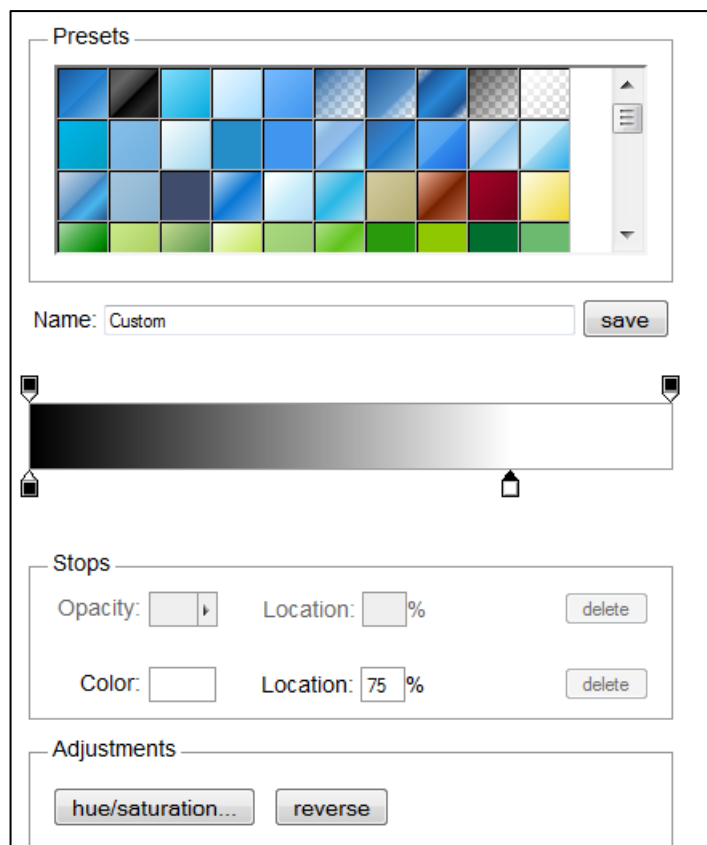
5.1 Styl pro větší zobrazovací zařízení a novější prohlížeče

5.1.1 Pozadí

Na pozadí stránky jsem použil přechod. Bez použití CSS 3 bych jej musel vytvářet pomocí obrázku například v programu Adobe Photoshop. Vytvořil bych obrázek s výškou, jakou chci mít a šíří 1 pixel. Tento obrázek bych nechal opakovat po ose x. Zápis by vypadal takto:

```
background: url('../images/body-background.png') repeat-x #FFFFFF;
```

V CSS 3 mohu toto přechodové pozadí vytvořit pomocí gradientu:



Obrázek 15 – tvorba gradientů

Zdroj: <http://www.colorzilla.com/gradient-editor/>

Pro IE 9 bych musel vypnout filtry a do hlavičky stránek připsat tento styl:

```
<!--[if gte IE 9]>
    <link rel="stylesheet" href="style/filter-none.css"
type="text/css">
<![endif]-->
```

Ve stylu se pouze vypne filtr:

```
.gradient {
    filter: none;
}
```

Protože starším prohlížečům nabídnu základní zobrazení, nemusím se zabývat filtry (jejich zapínáním i vypínáním). Z CSS kódu vypustím poslední řádek a styl ze souboru filter-none.css nepoužiji.

5.1.2 Kulaté rohy

V poslední době jsou v módě kulaté rohy. Dvěma prvkům na stránce jsem tedy kulaté rohy přidělil. Kdybych chtěl kulaté rohy dříve, musel bych to řešit pomocí obrázků a do stránky by mi přibýlo několik elementů `<div>`. Kód lze opět nechat vygenerovat. Například na stránce <http://css3generator.com/>. Kód pro kulaté rohy:

```
-webkit-border-radius: 10px;  
-moz-border-radius: 10px;  
border-radius: 10px;
```

Kdybych chtěl přidat kulaté rohy i do IE 6 až 8, musel bych použít nějaký skript. Možností je více. Můžu pouze stáhnout jeden skript a připsat následující kód:

```
-pie-border-radius: 10px;  
behavior: url(pie/PIE.htc);
```

PIE je skript, který umožňuje použít některé novinky z CSS 3 v IE od verze 6. Skript jsem stáhl z adresy <http://css3pie.com/about/>. Potíž s ním byla v tom, že se prvek, na který byl aplikován PIE, zobrazil za všemi prvky stránky (po ose z) a nebyl vidět. Na fóru bylo doporučeno přidat vlastnost `z-index`, ale v tomto případě to nepomohlo.

Dalším skriptem je `DD_roundies`, který funguje podobně jako PIE. (http://www.dillerdesign.com/experiment/DD_roundies/) V tomto případě se kulaté rohy vykreslily. Při zmenšování okna prohlížeče zůstal prvek s rohy na svém místě, i když ostatní prvky se hýbaly, tak jak měly. Když text „sleze“ ze svého pozadí, stránka přestane být čitelná.

Možnost vytvoření kulatých rohů bez použití JavaScriptu je `Spanky Corners` (<http://tools.sitepoint.com/spanky/index.php>). Ten však vytvoří obrázky stejně, jako bychom to mohli udělat sami v Photoshopu. Pro přechodové pozadí není vhodný, protože je třeba nastavit barvu na pozadí. Využívá jednoho CSS hacku, takže do HTML není potřeba přidávat další kód.

5.1.3 Písmo

CSS 3 umožňuje použít své vlastní písmo. Některé fonty jsou dostupné zdarma a jiné jsou placené. Použil jsem font, který je zdarma. Stáhl jsem ho ze stránky <http://www.google.com/webfonts#ChoosePlace:select>. Potom jsem využil stránky <http://www.fontsquirrel.com/fontface/generator>, kde jsem si nechal vygenerovat více formátů fontu. Potom stačilo font umístit na stránky a do souboru se stylem zapsat následující kód:

```
@font-face {  
    font-family: 'devonshire';  
    src: url(../fonts/devonshire-regular-webfont.eot);  
    src: url(../fonts/devonshire-regular-webfont.svg);  
    src: url(../fonts/devonshire-regular-webfont.svgz);  
    src: url(../fonts/devonshire-regular-webfont.ttf);  
    src: url(../fonts/devonshire-regular-webfont.woff);
```

```
}
```

Nyní už písmo můžu používat klasicky zápisem: `font-family: devonshire;`. Je dobré uvádět i další druh písma, kdyby se vlastní písmo z nějakého důvodu nenačetlo.

5.1.4 Jiné počáteční písmeno v odstavci

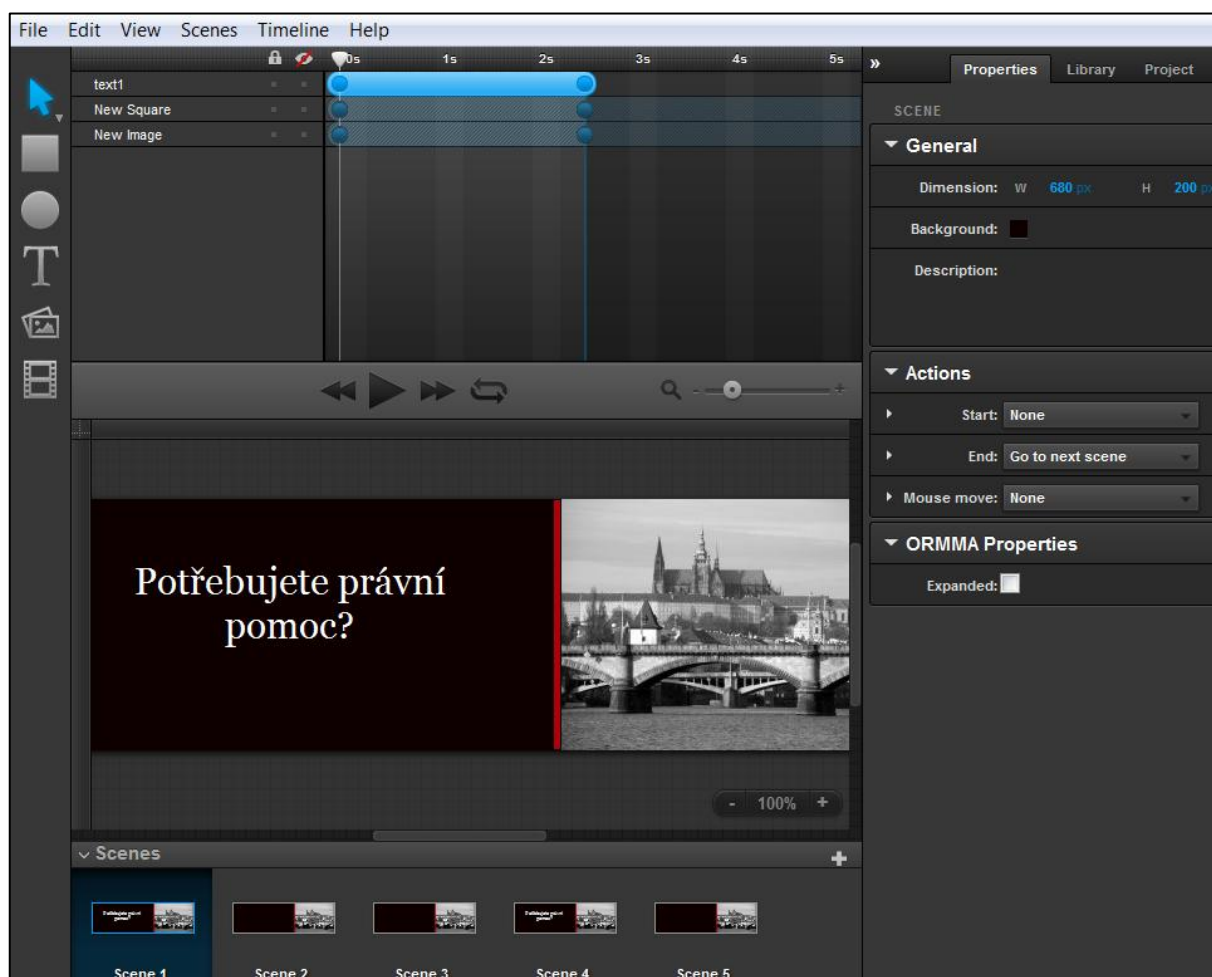
Díky selektorům lze změnit styl každému prvnímu (nebo jakémukoli) prvku. Já jsem prvnímu písmenu v odstavci změnil velikost a font:

```
article p:first-letter {  
font-family: devonshire;  
font-size: 14pt;  
font-weight: normal;  
}
```

6. Nové funkce

6.1 Animace

Do hlavičky stránky jsem se rozhodl umístit animaci. Využil jsem nástroje Sencha Animator (viz Obrázek 16) – (<http://www.sencha.com/store/animator/>), což je nástroj na tvorbu animací pro prohlížeče pracujících na jádru WebKit.

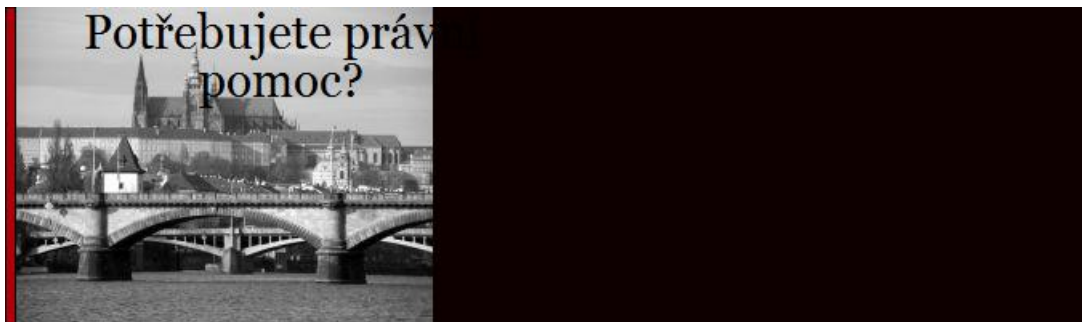


Obrázek 16 – Sencha Animator

Pracovní prostředí je uživatelsky přívětivé a intuitivní. Nalezneme zde časovou osu, panel na vytváření prvků a panel s vlastnostmi. Sencha Animator se občas zasekl a občas nešel vybrat nějaký nástroj.

Po exportu projektu vznikne poměrně nepřehledný HTML soubor, který jsem rozdělil na jeden soubor se stylem a na dva soubory s JavaScriptem. Zbýlý HTML kód jsem vložil do místa, kde jsem chtěl mít animaci. HTML kód je seznam, který obsahuje všechny části jako obrázky a text. Vlastní animace funguje na prohlížečích na jádře WebKit a na dotykových zařízeních od Applu. Testoval jsem ji v prohlížeči Google Chrome 15.0.874.121 m, kde se zobrazovala, ale s určitými problémy. Animace sice fungovala jak měla, ale některé prvky na stránce se nechovaly správně. Navigace měla po přejetí myši změnit barvu a podtrhnout se.

K tomu ale vlivem animace nedošlo. V ostatních prohlížečích se ani nespustila a všechny prvky v animaci se umístily vlevo (viz Obrázek 17).



Obrázek 17 – nefunkční animace v prohlížeči Firefox 7.0.1

Po troše pátrání jsem objevil program Adobe Edge, který má velmi podobné pracovní prostředí a jeho animace fungují napříč prohlížeči. V něm jsem tedy vytvořil stejnou animaci, která už fungovala ve více prohlížečích. Oba dva programy jsou placené, ale program lze 30 dnů zkusit zdarma.

6.2 Formuláře

Na stránce obsahující kontaktní informace jsem vytvořil formulář. Pole pro jméno a e-mail musí být vyplněny. Pole pro e-mail navíc obsahuje `input type="email"`, který zajistí, že bude vyplněna platná e-mailová adresa. V polích se zobrazuje nápověda, co má uživatel vyplnit. Nápověda zmizí po kliknutí myši do pole. Kód vypadá následovně:

```
<form action="" method="post">
Jméno:<br /> <input type="text" placeholder="Vaše jméno" required="true"
name="jmeno" value="" /><br />
Email:<br /> <input type="email" placeholder="Váš e-mail" required="true"
name="email" value="" /><br />
Zpráva:<br /><textarea id="zprava" name="zpráva" rows="5"
cols="20"></textarea><br />
<input type="submit" value="Odeslat" />
</form>
```

Starší prohlížeče budou atribut `placeholder` ignorovat. Zprovoznit jej můžeme způsobem, který se používal doposud – pomocí JavaScriptu:

```
<input type="text" value="Vaše jméno"
onfocus="if(this.value==this.defaultValue)this.value=''"
onblur="if(this.value=='')this.value=this.defaultValue" />
```

Nebo můžeme využít jQuery. Pomocí javaskriptové knihovny Modernizer zjistíme podporu pro atribut placeholder. Pokud není placeholder podporován, spustí se jQuery. Pro zprovoznění je potřeba jQuery, Modernizer a následující kód:

```
<script src="jquery.js"></script>
<script src="modernizr.js"></script>

<script>
$(document).ready(function() {

if(!Modernizr.input.placeholder) {

    $('[placeholder]').focus(function() {
        var input = $(this);
        if (input.val() == input.attr('placeholder')) {
            input.val('');
            input.removeClass('placeholder');
        }
    }).blur(function() {
        var input = $(this);
        if (input.val() == '' || input.val() ==
input.attr('placeholder')) {
            input.addClass('placeholder');
            input.val(input.attr('placeholder'));
        }
    }).blur();
    $('[placeholder]').parents('form').submit(function() {
        $(this).find('[placeholder]').each(function() {
            var input = $(this);
            if (input.val() == input.attr('placeholder')) {
                input.val('');
            }
        })
    });
}

});

</script> [20]
```

Pomocí jQuery lze pro starší prohlížeče formulář také validovat. Na adrese <http://flowplayer.org/tools/download/index.html> si lze sestavit vlastní skript podle toho, co všechno budeme potřebovat. Pro kontrolu formuláře potom stačí vložit na konec stránky skript:

```
$("#myform").validator();
```

Dále jsem vytvořil samostatný styl pro formulář a v daném stylu ve třídě error upravil vzhled chybové zprávy při validaci.

6.3 Video

Aby bylo možné video přehrát v různých prohlížečích, musel jsem záznam nejprve převést do více formátů. Potom stačilo na stránky napsat kód pro vložení videa a video fungovalo:

```
<video width="320" height="240" controls="controls" autoplay="">
<source src="video/video.mp4" type="video/mp4" />
<source src="video/video.ogv" type="video/ogv" />
<source src="video/video.webm" type="video/webm" />
Zde je umístěno video, ale Váš prohlížeč bohužel nepodporuje video v HTML
5.
</video>
```

U videa se zobrazí ovládací lišta a `autoplay=""` znamená, že se video začne přehrávat samo od sebe. Prohlížeče, které nepodporují tag `<video>`, zobrazí pouze text se zprávou, že video nelze zobrazit.

6.4 Geolokace

HTML 5 nabízí možnost zjištění aktuální polohy zařízení, ze kterého se právě připojujete. Možností, kterých API pro geolokaci využívá je více. Pokud například aktuální polohu zjišťujeme pouze podle IP adresy, může být informace velmi nepřesná.

Mým cílem bylo umístit na stránky mapu, která by vyhledala trasu z aktuální polohy na adresu kanceláře. Z internetové adresy <http://api4.mapy.cz/> jsem stáhl mapu od Seznamu a vložil ji do stránek. V kódu bylo umístěno pole pro vyhledávání místa, z kterého chceme trasu vyhledat a tlačítko pro odeslání. Tomuto poli jsem přiřadil atribut `name="prvni"`.

```
<input type="text" value="" id="mapy-route-frm" name="prvni"><input
onclick="route()" type="button" value="Hledat">
```

Dále jsem upravil kód pro zjištění aktuální polohy tak, aby se aktuální poloha načetla do pole. Úplný výsledný kód, který má na starosti zjištění aktuální polohy vypadá takto:

```
function showPosition(position) {
    var element = document.getElementsByName('prvni')[0];
    element.value= (position.coords.latitude + " " +
position.coords.longitude);
}
if (navigator.geolocation) {
    navigator.geolocation.getCurrentPosition(showPosition);
} else {
    alert("ERR: Prohlizec nepodporuje Geolocation API");
}
```

Při načítání stránky prohlížeč položí dotaz, zda chceme prozradit naši polohu. Když svolíme, souřadnice se načtou do pole. Po kliknutí na tlačítko „hledej“ se zobrazí trasa. Ta vede z aktuální polohy do kanceláře. Poli, do kterého se aktuální poloha načítá, jsem změnil

input type na z text na hidden, protože není potřeba aby se uživateli poloha zobrazovala. Při testování z počítače připojeného pomocí ADSL modemu se mi má aktuální poloha umístila 13 kilometrů od skutečné pozice. Informace pro zjištění polohy nebyly dostatečné. Při testování zjištění polohy na jiné soukromé Wi-Fi síti se poloha zobrazila s přesností desítek metrů.

7. Validita

HTML soubor je podle validátoru HTML 5 (<http://html5.validator.nu/>) validní, ale styl validní není.

Validátor CSS 3 (http://jigsaw.w3.org/css-validator/#validate_by_uri+with_optionss) zahlásil 12 chyb a z toho 3 typy zpráv:

- background -moz-linear-gradient(top,#000000 0%,#ffffff 75%) není hodnota background-color : -moz-linear-gradient(top,#000000 0%,#ffffff 75%)
- Vlastnost -webkit-border-radius neexistuje : 10px
- Vlastnost -webkit-transform neexistuje : scale(1.2,1.2)

8. Test prohlížečů

Prohlížeče jsem testoval pomocí prohlížečů nainstalovaných na počítači a nástroje Adobe BrowserLab (<https://browserlab.adobe.com/>), který udělá screenshot z řady prohlížečů. Získané obrázky lze mezi sebou porovnávat.

8.1 Grafická podoba

Stránka se zobrazuje stejně v prohlížečích: Firefox 8.0., Firefox 7.0, Firefox 5.0, IE 9.0, Safari 5.1, Opera 9.5, Opera 11.52, Chrome 14.0 a Chrome 13.0.

V prohlížečích IE 7.0, IE 8.0 se stránka načte bez grafických vylepšení, jako jsou kulaté rohy, stín písma atd. To je způsobeno tím, že prohlížeč čte pouze styl ze souboru `style.css`, takže grafická podoba je správná. Prohlížeč Opera 9.5 zobrazí stránku bez kulatých rohů a s větší velikostí fontů pro navigaci. Prohlížeč neumí zobrazit nový font Devonshire a pouze aplikuje větší velikost fontu na font starý.

8.2 Funkčnost

Funkčnost jsem testoval v prohlížečích Firefox 7.0, Firefox 8.0, IE 9.0, IE 8.0, Opera 9.5, Opera 11.52 a Chrome 15.0.

Animace se zobrazuje správně ve všech prohlížečích kromě IE 8.0 a Opery 9.5, kde se nezobrazí vůbec. Z testovaných prohlížečů nepodporuje geolokaci IE 8.0 a Opera 9.5. Kontrola při odeslání formuláře funguje ve všech prohlížečích díky aplikovanému JavaScriptu. Video nefunguje v IE 8.0, Opeře 9.5.

8.3 Použité prvky

Zda použité prvky fungují, je zahrnuto v následující tabulce:

	Firefox 7.0	Firefox 8.0	IE 9.0	IE 8.0	Opera 9.5	Opera 11.52	Chrome 15.0
border-radius	Ano	Ano	Ano	Ne	Ne	Ano	Ano
text-shadow	Ano	Ano	Ne	Ne	Ne	Ano	Ano
animace	Ano	Ano	Ano	Ne	Ne	Ano	Ano
video	Ano	Ano	Ano	Ne	Ne	Ano	Ano
geolokace	Ano	Ano	Ano	Ne	Ne	Ano	Ano
transformace	Ano	Ano	Ano	Ne	Ne	Ano	Ano
linear-gradient	Ano	Ano	Ano	Ne	Ano	Ano	Ano
@font-face	Ano	Ano	Ano	Ne	Ne	Ano	Ano
Input type email	Ano	Ano	Ano	Ne	Ne	Ano	Ano
required="true"	Ano	Ano	Ano	Ne	Ne	Ano	Ano
placeholder	Ano	Ano	Ano	Ne	Ne	Ano	Ano
:first-letter	Ano	Ano	Ano	Ano	Ano	Ano	Ano
media screen	Ano	Ano	Ano	Ne	Ano	Ano	Ano

Závěr

Cílem bakalářské práce bylo prozkoumat možnosti jazyka HTML 5 a CSS 3. Nové verze byly obohaceny o mnoho praktických funkcí. Ať se jedná o práci s formuláři, snadnější tvorbou vzhledu, či zvýšení sémantiky kódu. V praktické části bylo cílem prozkoumat použitelnost jazyků již nyní, kdy specifikace není dokončena.

Výsledkem práce je zjištění, že HTML 5 i CSS 3 lze využívat již dnes. Jen na každém tvůrci webových stránek stojí rozhodnutí, jak naložit s kompatibilitou starších prohlížečů. To, že starší prohlížeče nové prvky nepodporují, lze obejít pomocí různých skriptů. Je tedy otázka, zda chceme stránky zatěžovat dalšími skripty a věnovat projektu více času. Další možností je uživatele starších prohlížečů o některé funkce a grafické prvky ochudit. Jaké jsou tedy hlavní důvody pro tvorbu stránek pomocí HTML 5 a CSS 3? Tvorba s pomocí těchto jazyků je rychlejší, kód je srozumitelnější, stránku lze snadněji indexovat a vzhled není řešen pomocí obrázků, takže stránka se zobrazí rychleji.

Oba dva jazyky jsou pořád ve stádiu návrhu a specifikace ještě není hotová. Jazyk se neustále mění a funkce, které již začínají být implementovány, se mohou změnit. Návrh HTML 5 a CSS 3 byl uvolněn, aby měli tvůrci HTML 5 a CSS 3 zpětnou vazbu od webových vývojářů a mohli návrh podle praktických zkušeností upravit. Zcela hotová specifikace je naplánována až na rok 2022. Nejde ale rozhodně o rok vydání, kdy se začne s používáním. Je to rok, kdy by měla být specifikace po všech připomínkách definitivně dokončena. Pro běžnou tvorbu webu se ale pravděpodobně ve větším měřítku začne využívat mnohem dříve. Již dnes HTML 5 a CSS 3 přináší novinky, které zkrátí čas, usnadní práci i množství přenesených bajtů mezi serverem a uživatelem.

Podle mého názoru jsou oba jazyky tím správným krokem, ale kudy se skutečně bude ubírat vývoj, ukáže až čas.

Bibliografie

1. **Pilgrim, Mark.** Jak to všechno bylo? Dive Into HTML5. *Dive Into HTML5*. [Online] [Citace: 20. říjen 2011.] <http://kniha.html5.cz/>.
2. **Kosek, Jiří.** Historie a vývoj HTML. *html guru*. [Online] 2010. [Citace: 24. říjen 2011.] <http://htmlguru.cz/uvod-historie.html>.
3. **Bechynský, Štěpán.** HTML 5 ROADSHOW. Plzeň : 13. říjen 2011.
4. **Hunt, Lachlan.** Seznámení s HTML 5. *Interval.cz*. [Online] 12. 24 2007. [Citace: 29.. října 2011.] <http://interval.cz/clanky/seznameni-s-html-5/>.
5. **W3C.** World Wide Web Consortium Process Document. W3C. [Online] 14. říjen 2005. [Citace: 25. říjen 2011.] <http://www.w3.org/2005/10/Process-20051014/>.
6. **Goldsteinová, Alexis, Lazaris, Louis a Weylová, Estelle.** *HTML5 a CSS3*. Brno : Zoner press, 2011.
7. **Dvořák, Jakub.** HTML 5: nová generace webů. *zive*. [Online] 9. červenec 2009. [Citace: 23. říjen 2011.] <http://www.zive.cz/clanky/html-5-nova-generace-webu/sc-3-a-147815/default.aspx>.
8. **Pilgrim, Mark.** *Dive into HTML 5*. [Online] [Citace: 2011. říjen 2011.] <http://kniha.html5.cz/semantics.html>.
9. **w3schools.** HTML5 Video. *w3schools.com*. [Online] [Citace: 25.. říjen 2011.] http://www.w3schools.com/html5/html5_video.asp.
10. **Wikipedia.** HTML5 video. *Wikipedia*. [Online] 25.. září 2011. [Citace: 2009. říjen 2011.] http://en.wikipedia.org/wiki/HTML5_video.
11. **Václavík, Lukáš.** Google I/O: HTML5 a VP8 jako konečné řešení online videa? *CNEWS.cz*. [Online] 20.. květen 2010. [Citace: 29.. říjen 2011.] Google I/O: HTML5 a VP8 jako konečné řešení online videa?.
12. **w3schools.** HTML5 Tutorial. *w3schools.com*. [Online] 2011. [Citace: 15.. 11. 2011.] <http://www.w3schools.com/html5/>.
13. **Hassman, Martin.** HTML 4 5 6... [Online] 24.. 6. 2010. [Citace: 22.. 11. 2011.] <http://html456.blogspot.com/2010/06/jak-funguje-geolokace-ve-firefoxu.html>.

14. **Sládek, Jan.** Webdesignérův průvodce po HTML5 - taháme data od návštěvníka. *Root.cz*. [Online] 29. 6 2010. [Citace: 29. října 2011.] <http://zdrojak.root.cz/clanky/webdesigneruv-pruvodce-po-html5-tahame-data-od-navstevnika/>.
15. Kaskádové styly. *Wikipedie*. [Online] 27.. 9. 2011. [Citace: 25.. 11. 2011.] http://cs.wikipedia.org/wiki/Kask%C3%A1dov%C3%A9_styly.
16. **Sládek, Jan.** Webdesignérův průvodce po CSS3. *zdroják.cz*. [Online] 14.. 9. 2010. [Citace: 29.. říjen 2011.] <http://zdrojak.root.cz/serialy/webdesigneruv-pruvodce-po-css3/>.
17. **Macek, Petr.** Fonty a web včera, dnes a zítra. *Zdroják.cz*. [Online] 18. 3 2011. [Citace: 10. 11 2011.] <http://zdrojak.root.cz/clanky/fonty-a-web/>.
18. **W3C.** Cascading Style Sheets (CSS) Snapshot 2010. W3C. [Online] 12.. 5. 2011. [Citace: 21.. 10. 10.] <http://www.w3.org/TR/CSS/#css3>.
19. **Jensen, Randy.** CSS3 Generator. *CSS3 Generator*. [Online] [Citace: 10.. listopad 2011.] <http://css3generator.com/>.
20. **La, Nick.** Cross-Browser HTML5 Placeholder Text. *web designer wall*. [Online] 28.. 3. 2011. [Citace: 20.. 11. 2011.] <http://webdesignerwall.com/tutorials/cross-browser-html5-placeholder-text>.

Seznam obrázků

Obrázek 1 – osnova dokumentu.....	14
Obrázek 2 – struktura webu pomocí divů v HTML 4.....	15
Obrázek 3 – struktura webu bez použití divů, s použitím nových elementů v HTML 5.....	15
Obrázek 4 – povolení zjištění polohy	19
Obrázek 5 – chybová zpráva.....	21
Obrázek 6 – nový font	25
Obrázek 7 – ostrý přechod	28
Obrázek 8 – opakování gradientu	30
Obrázek 9 – vícenásobné pozadí	30
Obrázek 10 – tlačítko v CSS 3.....	33
Obrázek 11 - nová struktura.....	34
Obrázek 12 – stará struktura	34
Obrázek 13 – před přidáním skriptu v IE 8	35
Obrázek 14 – po přidání skriptu v IE 8.....	35
Obrázek 15 – tvorba gradientů.....	40
Obrázek 16 – Sencha Animator	43

Obrázek 17 – nefunkční animace v prohlížeči Firefox 7.0.1	44
---	----

Seznam příloh

Příloha 1 – použité nástroje z webu.....	55
Příloha 2 – stará struktura HTML dokumentu.....	56
Příloha 3 – nová struktura HTML dokumentu	57
Příloha 4 – spustitelné soubory stránek.....	na přiloženém CD

Seznam použitých termínů a zkratek

Termín	Zkratka	Význam [zdroj]
Hypertextový značkovací jazyk	HTML	HTML je zkratka Hypertext markup Language (hypertextový značkovací jazyk). HTML je v dnešní době stále nejoblíbenější jazyk pro vytváření jednoduchých www stránek. [http://www.artic-studio.net/slovnicek-pojmu/jazyk-html/]
eXtensible Markup Language	XML	Rozšířitelný značkovací jazyk.
Web Hypertext Application Technology Working Group	WHATWG	Seskupení prohlížečů založil Apple, Mozilla a Opera.
deklarace typu dokumentu	DTD	Definice typu dokumentu (DTD) přitom říká, které elementy a atributy můžeme v dokumentu použít. Navíc je zde definováno, v jakých vzájemných vztazích mohou být jednotlivé elementy použity. DTD je tedy užitečný nástroj, který nám umožní hlídat, zda mají naše dokumenty správnou strukturu. [http://www.kosek.cz/clanky/xml/xml-01.html]
Aplication Programming Interface	API	Aplikační programové rozhraní. Jde o sbírku procedur, funkcí či tříd nějaké knihovny (ale třeba i jiného programu nebo jádra operačního systému), které může programátor využívat. API určuje, jakým způsobem jsou funkce knihovny volány ze zdrojového kódu programu. [http://cs.wikipedia.org/wiki/API]
Document object model	DOM	Objektový model dokumentu je API. Pomocí tohoto rozhraní můžete pracovat se samotnou webovou stránkou. DOM umožňuje přistupovat k jednotlivým objektům XML (XHTML) dokumentu a pracovat s nimi. Těmito objekty jsou elementy, atributy, text, komentáře. [http://www.tvorba-webu.cz/dom/]
Cascading Style Sheets	CSS	Kaskádové styly – "souhrn pravidel a metod pro formátování webových stránek" [http://owebu.blogger.cz/Grafika-CSS/CSS-pro-zacatecniky-co-to-vlastne-CSS-je-1-dil]
HyperText Markup Language	XHTML	Jazyk XHTML vychází z XML. XHTML je jazyk pro vytváření www stránek. XHTML má velice přísná a zároveň jednoduchá pravidla. Díky tomu mohou počítače XML a tedy i XHTML velmi snadno automatizovaně zpracovávat. [http://www.artic-studio.net/slovnicek-pojmu/jazyk-xhtml]
Flash		Flash je technologie, která umožňuje výrobu interaktivních animací a programů nejen pro webové stránky. Flash vyvinula firma Macromedia.[http://www.artic-studio.net/slovnicek-

Termín	Zkratka	Význam [zdroj]
		pojmu/flash]
Silverlight		Microsoft Silverlight je technologie pro internetové prohlížeče. Je to platforma určená pro tvorbu dynamického online obsahu a interaktivní práce s ním. Kombinuje text, vektorovou i bitmapovou grafiku, animace a video. [http://www.microsoft.com/cze/web/silverlight/]
JavaScript	JS	JavaScript je programovací jazyk, který se používá v internetových stránkách. JavaScript je klientský skript. To znamená, že se program odesílá se stránkou na klienta (do prohlížeče) a teprve tam je vykonáván. [http://www.jakpsatweb.cz/javascript/javascript-uvod.html]
Element, tag		Definované značky (tagy) k vytváření a formátování dokumentů pro webové stránky.[http://www.artic-studio.net/slovnicek-pojmu/jazyk-html/]
Atribut		Atributy jsou speciální části v kódu psaného v jazycích HTML nebo XHTML. Atributy jsou jakési parametry, umožňují jednotlivým značkám přiřazovat speciální vlastnosti. [http://www.adaptic.cz/znalosti/slovnicek/atribut/]
Kodek		Kodek je zařízení nebo počítačový program, který dokáže transformovat datový proud (stream) nebo signál.[http://cs.wikipedia.org/wiki/Kodek]
Zásuvný modul, plugin		Software, který nepracuje samostatně, ale jako doplňkový modul jiné aplikace a rozšiřuje tak její funkčnost. [http://cs.wikipedia.org/wiki/Plugin]
Vendor prefix		Předpona vlastnosti v CSS, která patří určitému prohlížeči. Zapisují se u vlastností, které nemají dokončenou specifikaci.

Příloha 1 – použité nástroje z webu

V tabulce je seznam a krátká vysvětlivka užitečných nástrojů a stránek při tvorbě webu pomocí HTML 5 a CSS 3:

Adresa	Popis
http://www.findmebyip.com/litmus/#target-selector	Podpora HTML 5 a CSS 3 v prohlížečích.
http://html5test.com/	Test podpory HTML 5 v konkrétním prohlížeči.
http://www.sencha.com/products/animator/	Sencha Animator – desktopový program pro tvorbu animací pro prohlížeče pracujících na jádru WebKit.
http://labs.adobe.com/technologies/edge/	Adobe Edge – desktopový program pro tvorbu animací od Adobe.
http://css3generator.com/	Generátor kódu pro CSS 3 vlastnost.
http://ukazky.zdrojak.cz/html5/	Detekce podpory HTML 5 pro právě používaný prohlížeč.
http://css3pie.com/	JS umožňující kulaté rohy, stíny, gradienty a více obrázkové pozadí do starších IE.
http://www.dillerdesign.com/experiment/DD_roundies/	JS vytvářející kulaté rohy.
http://tools.sitepoint.com/spanky/index.php	Kulaté rohy pomocí obrázku.
http://prohlizece.info/clanky/css3-selektory-jaka-je-podpora-prohlizecu/	Podpora selektorů v prohlížečích.
http://tools.css3.info/selectors-test/test.html	Test podpory selektorů v konkrétním prohlížeči.
http://www.w3schools.com/html5/default.asp	Popis a praktické možnosti praktického vyzkoušení elementů a atributů HTML 5.
http://www.quackit.com/html_5/tags/	Přehled HTML tagů včetně páté verze.
http://html5doctor.com/html-5-reset-stylesheet/	Zprovoznění HTML 5 tagů v prohlížečích IE.
http://webdesignerwall.com/tutorials/cross-browser-html5-placeholder-text	Zprovoznění placeholderu v prohlížečích bez podpory.
http://www.matiasmancini.com.ar/jquery-plugin-ajax-form-validation-html5.html	Zprovoznění funkcí formulářů v HTML 5 v prohlížečích bez podpory.
http://flowplayer.org/tools/download/index.html	Sestavení jQuery na míru.
http://www.fontsquirrel.com/	Fonty do @font-face zdarma.
http://www.fontsquirrel.com/fontface/generator	Generátor fontů do více formátů.
http://www.google.com/webfonts#ChoosePlace:select	Fonty zdarma na Googlu.
http://www.colorzilla.com/gradient-editor/	Generátor kódu pro gradienty.

Příloha 2 – stará struktura HTML dokumentu

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN">
<html>
  <head>
    <meta http-equiv="content-type" content="text/html; charset=utf-8">
    <meta name="description" content="Advokátní kancelář Eva Stehlíková
Praha právník právnické služby">
    <meta name="author" content="Petr Stehlík">
    <meta name="robots" content="all, follow">
    <style type="text/css"> @import "style.css";
  </style>
    <link rel="icon" href="/favicon.ico" type="image/x-icon">
    <title>Advokát JUDr. Eva Stehlíková
  </title>
  </head>
  <body>
    <div id="container">
      <div id="container-heather">
        <a href="index.html"> </a>
        <p class="text-heather"> <span class="text-white">JUDr.
</span><span class="text-black">Eva</span> <span class="text-white">
Stehlíková,</span><span class="text-black"> právní</span> <span
class="text-white"> služby</span></p>
      </div>
      <div id="container-content">
        <div id="container-left-menu">
          <div id="container-left-menu-img">
          </div>
          <div id="container-left-menu-links">
            <ul class="bullet">
              <li class=
"current_page"><a href="index.html">O mně</a></li>
              <li><a href="obcanske-pravo.html">Občanské právo</a>
</li>
              <li><a href="trest.html">Trestní právo</a> </li>
              <li><a href="mediace.html">Mimosoudní řešení sporů</a>
</li>
              <li><a href="ceny.html">Ceny</a> </li>
              <li><a href="kontakt.html">Kontakt</a> </li>
            </ul>
          </div>
        </div>
        <div id="container-right">
          <div id="container-text">
            <h1>Advokátka <br> JUDr. Eva Stehlíková</h1>
            <p> Pro své klienty zajišťuji <b>kompletní právní
servis</b>, což zahrnuje právní porady, sepis smluv a jiných listin,
zastupování v řízení před soudy a správními orgány,
mimosoudní jednání s klienty i protistranou. Ráda Vám pomohu s řešením
```

```

Vašich obtížných životních situací a proto mě neváhejte
<a href="kontakt.html"><b>kontaktovat</b></a>.
    </p>
    <p>Zastupji své klienty před soudy všech stupňů,
v občanskoprávních i v trestních věcech.
    </p>
    <iframe
src="http://www.facebook.com/plugins/like.php?href=http%3A%2F%2Fakstehlikov
a.cz%2F&layout=button_count&show_faces=false&width=150&acti
on=like&font=arial&colorscheme=light&height=21" scrolling="no"
frameborder="0" style="border:none; overflow:hidden; width:150px;
height:21px;" allowTransparency="true"></iframe>
    </div>
    <div id="container-images">
        
    </div>
    <div class="container-end">
    </div>
</div>
<div class="container-end">
</div>
<div id="container-footer">
    <p class="text-footer"> <span class="text-white">ta</span>
<span class="text-black"> správná</span> <span class="text-white">
volba</span></p>
    </div>
</div>
</body>
</html>

```

Příloha 3 – nová struktura HTML dokumentu

```

<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="content-type" content="text/html; charset=utf-8">
        <meta name="description" content="Advokátní kancelár Eva Stehlíková
Praha právník právnické služby">
        <meta name="author" content="Petr Stehlík">
        <meta name="robots" content="all, follow">
        <!--[if lt IE 9]>
        <script src="js/html5.js"></script>
        <![endif]-->
        <link rel="stylesheet" href="style/reset.css" type="text/css">
        <link rel="stylesheet" href="style/style.css" type="text/css">
        <link rel="icon" href="/favicon.ico" type="image/x-icon">
        <title>Advokát JUDr. Eva Stehlíková
        </title>

```

```

</head>
<body>
  <div id="container">
    <header>
      <a href="index.html"> </a>
      <h1 class="text-heather"> <span class="text-white">JUDr.
</span><span class="text-black">Eva</span> <span class="text-white">
Stehlíková,</span><span class="text-black"> právní</span> <span
class="text-white"> služby</span></h1>
    </header>
    <div id="container-content">
      <aside>
        <div id="container-left-menu-img">
        </div>
        <nav>
          <ul class="bullet">
            <li class=
"current_page"><a href="index.html">O mně</a></li>
            <li><a href="obcanske-pravo.html">Občanské právo</a>
</li>
            <li><a href="trest.html">Trestní právo</a> </li>
            <li><a href="mediace.html">Mimosoudní řešení sporů</a>
</li>
            <li><a href="ceny.html">Ceny</a> </li>
            <li><a href="kontakt.html">Kontakt</a> </li>
          </ul>
        </nav>
      </aside>
      <selection>
        <article>
          <h1>Advokátka <br> JUDr. Eva Stehlíková</h1>
          <p> Pro své klienty zajišťuji <b>kompletní právní
servis</b>, což zahrnuje právní porady, sepis smluv a jiných listin,
zastupování v řízení před soudy a správními orgány,
mimosoudní jednání s klienty i protistranou. Ráda Vám pomohu s řešením
Vašich obtížných životních situací a proto mě neváhejte
<a href="kontakt.html"><b>kontaktovat</b></a>.
          </p>
          <p>Zastupji své klienty před soudy všech stupňů,
v občanskoprávních i v trestních věcech.
          </p>
          <iframe
src="http://www.facebook.com/plugins/like.php?href=http%3A%2F%2Fakstehlikov
a.cz%2F&layout=button_count&show_faces=false&width=150&acti
on=like&font=arial&colorscheme=light&height=21" scrolling="no"
frameborder="0" style="border:none; overflow:hidden; width:150px;
height:21px;" allowTransparency="true"></iframe>
        </article>
      <aside>
        
      </aside>

```

```

        <div class="container-end">
        </div>
    </selection>
    <div class="container-end">
    </div>
</div>
<footer>
    <p> <span class="text-white">ta</span> <span class="text-
black">  správná</span> <span class="text-white">  volba</span></p>
    </footer>
</div>
</body>
</html>

```