

**Česká zemědělská univerzita v Praze**

**Provozně ekonomická fakulta**

**Katedra informačních technologií**



**Bakalářská práce**

**HTML5 jako pokročilý prostředek pro tvorbu webové  
prezentace**

**Ondřej Čáslavka**

© 2015 ČZU v Praze

# ČESKÁ ZEMĚDĚLSKÁ UNIVERZITA V PRAZE

Katedra informačních technologií

Provozně ekonomická fakulta

## ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Čáslavka Ondřej

Informatika

Název práce

**HTML5 jako pokročilý prostředek pro tvorbu webové prezentace**

Anglický název

**HTML5 as a advanced tool for web pages creation**

---

### Cíle práce

Bakalářská práce bude založena na seznámení s HTML a CSS a jejich základními vlastnostmi. Bude analyzován přehled nových prvků a rozdílů s HTML5 a CSS3. Práce bude mít praktickou ukázkou, možnosti využití HTML5 jazyka a formátovacího prostředku CSS3 bude reprezentovat ukázková stránka. Dále bude analyzována použitelnost HTML5 jazyka v prostředí vybraných internetových prohlížečů.

### Metodika

Metodika bakalářské práce bude založena na studiu a analýze odborných zdrojů. Praktická část bude realizována formou vytvoření webové stránky dle základních pravidel. Na základě poznatků bude formulován závěr bakalářské práce.

### Harmonogram zpracování

1. Studium odborných informačních zdrojů, stanovení dílčích cílů a postupu řešení bakalářské práce: 07/2013
2. Zpracování teoretických východisek práce: 08/2013 – 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

Prohlašuji, že jsem bakalářskou práci na téma „*HTML5 jako pokročilý prostředek pro tvorbu webov*“ 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, 30. 11. 2015

.....

vlastnoruční podpis

## **Resumé**

Bakalářská práce pojednává o využití nových možností značkovacího jazyka HTML5 a formátovacího jazyka CSS3 a věnuje se porovnáním se staršími verzemi těchto specifikací.

Teoretická část se zabývá analýzou zdrojů, které se týkají tvorby a použití webových prezentací. Dále obsahuje shrnutí funkcí a vlastností, které přináší nové standardy HTML5 a CSS3 a je proveden test podpory těchto vlastností v rámci nejpoužívanějších webových prohlížečů.

Praktická část bakalářské práce zahrnuje vytvoření ukázkové webové prezentace za pomoci nové specifikace HTML5 a CSS3. Následně, součástí praktické části práce je popis hlavních prvků, které se na ukázkové webové prezentaci vyskytují.

## **Klíčová slova**

HTML5, CSS3, Flash, JavaScript, Internet Explorer, Chrome, Mozilla Firefox

## **Abstract**

The bachelor thesis discusses about of the use new possibilities markup language HTML5 and CSS3 formatting and engages in comparison with older versions of these specifications.

The theoretical part thesis analyzes the sources that deal with the production and use of websites. It also contains a summary of the functions and features that bring new standards HTML5 and CSS3 and a test is performed to support these properties within popular web browsers.

The practical part includes the creation of a sample web page, using the new specification HTML5 and CSS3. Then, the practical part contains a description of the main elements in the demonstration web presentation occur.

## **Keywords**

HTML5, CSS3, Flash, JavaScript, Internet Explorer, Chrome, Mozilla Firefox

Obsah	
1 Úvod.....	7
2 Cíl a metodika.....	8
2.1 Cíl práce .....	8
2.2 Metodika práce.....	8
3 Teoretická část .....	9
3.1 World Wide Web .....	9
3.2 HyperText Transfer Protocol .....	9
3.2.1 SSL .....	10
3.2.2 Stavové zprávy HTTP .....	11
3.4 Historie a vývoj HTML.....	11
3.4.1 Vývojáři HTML5.....	14
3.5 Porovnání HTML5 a HTML 4.01 .....	14
3.5.1 Deklarace typu dokumentu .....	14
3.5.2 Strukturování .....	15
3.5.3 Přehled nových elementů HTML5 .....	17
3.5.3.1 Popis strukturálních elementů .....	18
3.6 CSS.....	27
3.6.1 Historie CSS .....	28
3.6.2 Význam CSS .....	28
3.6.3 Základní selektory .....	29
3.6.4 Pseudotřídy .....	30
3.6.5 Nové grafické vlastnosti v CSS3.....	32
3.7 Webové prohlížeče a HTML5.....	34
3.7.1 Podíl prohlížečů na trhu .....	34
3.7.2 Test jednotlivých webových prohlížečů.....	36
4 Praktická část .....	38
4.1 Zápis DOCTYPE .....	39
4.2 Využití strukturálních značek .....	39
4.2.1 Hlavička.....	40

4.2.2 Tělo dokumentu.....	41
4.3 Navigační a odkazová lišta.....	45
4.4 Canvas .....	49
4.5 Video .....	52
4.6 Články .....	53
4.7 Formulář .....	55
4.8 Geolokace.....	58

# 1 Úvod

Pro některé uživatele je internet místo pro komunikaci s přáteli, s rodinou a pro jiné zdroj obživy nebo zábavy. V současnosti jsou na poli internetu webové prezentace považované jako samozřejmost. Drtivá většina uživatelů internetu neví, jaká technologie se využívá k tvorbě webových stránek. Webové prezentace neboli webové stránky jsou vytvářeny pomocí značkovacího jazyka HTML, který definuje celou kostru webové stránky, určuje její strukturu a vytváří celý obsah. Bez použití značkovacího jazyka HTML nelze vytvořit žádné webové stránky.

Nedílnou součástí webové prezentace jsou také kaskádové styly, neboli CSS. Tento formátovací jazyk je nepostradatelný k tomu, aby webové stránky získaly takzvané graficky přívětivé prostředí. Pomocí funkcí v CSS lze webovou stránku upravit tak, aby obsah webové stránky nebyl jen strohý text.

K webovým stránkám patří také webové prohlížeče. Tyto prohlížeče překládají kód psaný v HTML a následně zobrazí výsledek, kterým je webová stránka. Na trhu je velké množství webových prohlížečů, které nejsou plně kompatibilní se všemi verzemi specifikace HTML a CSS, případně dalšími programovacími jazyky, které se vpisují přímo do HTML kódu jako například JavaScript nebo podobná Java. Tato nekompatibilita způsobuje, že se webové stránky nezobrazují korektně a některé funkce nepracují správně nebo vůbec. Problém je také se zpětnou kompatibilitou. Není pravidlem, že nová verze webového prohlížeče automaticky správně zobrazí starší verzi kódu a podobně.

Bakalářská práce je rozdělena na dvě části a to na část praktickou a část teoretickou. V teoretické části se práce zabývá historií vývoje jazyků HTML a CSS, věnuje se rozdílům mezi jazyky HTML4.01 a HTML5. Na konci teoretické části jsou předloženy výsledky analýzy uživatelsky neoblíbenějších webových prohlížečů a test podpory nových vlastností HTML5 a CSS3 vybranými webovými prohlížeči.

Praktická část se primárně zabývá tvorbou webové prezentace za použití nové verze značkovacího jazyka HTML5 a formátovacího jazyka CSS3. Následně popisuje jednotlivé části vlastností a funkcí ze kterých je webová stránka vytvořena.

## 2 Cíl a metodika

### 2.1 Cíl práce

Bakalářská práce je primárně založena na seznámení s HTML a CSS a jejich základními vlastnostmi jako je přehled základních prvků a porovnání s HTML5 a CSS3. Práce obsahuje praktickou ukázkou webové stránky vytvořené pomocí specifikace HTML5 jazyka a formátovacího prostředí CSS3.

### 2.2 Metodika práce

Metodika bakalářské práce je založena na studiu a analýze odborných informačních zdrojů. Vlastní práce spočívá ve vytvoření webové prezentace za pomoci technologií HTML5, CSS3 a částečného využití Javascriptu. Na základě získaných poznatků z analýzy v rámci teoretické i praktické části, jsou formulovány závěry bakalářské práce.

Práce obsahuje 3 základní kapitoly:

- 3. Teoretická část
- 4. Praktická část
- 5. Závěr

„Teoretická část“ obsahuje stručné seznámení s vývojem služby WWW, protokolu HTTP a jazyka HTML a CSS. Dále je zaměřena na seznámení s novými prvky v HTML5 a CSS3. Poté je proveden průzkum v rámci nejpoužívanějších internetových prohlížečů a vyhodnocení vybraných prohlížečů.

„Praktická část“ je zaměřena na vytvoření webové prezentace prostřednictvím jazyka HTML5 a CSS3, která bude demonstrovat vybrané nové prvky těchto specifikací. Zároveň tyto prvky budou popsány a bude vysvětlena jejich funkčnost.

V kapitole „Závěr“ bude vyhodnocena praktická část práce a zvážení použitelnosti specifikace HTML5 a CSS3 ve webových prohlížečích.



## 3 Teoretická část

Součástí teoretické části je historie vývoje jednotlivých jazyků HTML a CSS, porovnávání specifikace HTML4.01 a HTML5. Vysvětluje nové vlastnosti a funkce jazyků HTML5 a CSS3. V neposlední řadě se také věnuje základní problematice tvorby webových stránek a dále seznamuje se základními vlastnostmi služby WWW a protokolu HTTP. Teoretická část je zakončena analýzou oblíbenosti jednotlivých prohlížečů.

### 3.1 World Wide Web

Služba WWW<sup>1</sup> je označením pro celosvětovou síť mezi sebou propojených hypertextových dokumentů v rámci internetu. Celá služba je založena na architektuře *client/server* kde fungování WWW zajišťuje protokol HTTP<sup>2</sup>. V roce 1989 vznikl v rámci organizace CERN<sup>3</sup> požadavek na snadno ovladatelný a všem přístupný vývěskový systém. Vědci z CERNU proto požádali o spolupráci Tima Bernerse-Lee<sup>4</sup>. Po krátkém vývoji se v roce 1990 zrodila služba *World Wide Web*, která po dalším vývoji za několik let vypracovala v nejpoužívanější službu Internet současnosti. (Sklenák, 2001)

Cíl této koncepce bylo poskytnout uživatelům prohlížení dokumentů po částech, tedy dokument se nemusel načíst celý, ale jen určitá část. Dokumenty na internetu jsou uloženy v počítačích, které jsou vzájemně propojené a uživatel si může mezi nimi číst a prostřednictvím hypertextových odkazů. (Slavoj, 2014)

### 3.2 HyperText Transfer Protocol

Zkráceně HTTP je protokol, pomocí kterého mezi sebou komunikují klienti a servery systému WWW. Jedná se o obecný aplikační protokol. HTTP se nemusí používat jen na přenos dokumentů, ale lze jej použít na přenos libovolných binárních dat a pro komunikaci mezi klienty a aplikačními bránami neboli *proxy server*. První verze HTTP vznikla s vývojem systému WWW. Požadavek pro přenos hypertextových dokumentů byl jednoduchý protokol bez složitého spojování. Původní verze HTTP umožňovala klientovi pouze navázat spojení se serverem, zaslat požadavek, přečíst odpověď a nakonec uzavřít spojení. Funkce serveru byla přijmout požadavek, zpracovat jej a odeslat požadovaný

---

<sup>1</sup> World Wide Web

<sup>2</sup> HyperText Transfer Protocol

<sup>3</sup> Evropské centrum jaderného výzkumu

<sup>4</sup> Tim Berners-Lee (8. 6. 1955) zdroj [www.biographyonline.net](http://www.biographyonline.net)

dokument nebo případně zprávu o chybě. Za krátkou dobu se zjistilo, že takto jednoduchý protokol nedostačuje, protože klient nebyl schopen rozeznat atributy zaslaného dokumentu jinak než z přípony dokumentu. Další problém byl, že klient neznal velikost a ani čas poslední modifikace dokumentu. Jelikož jsou při přenosu zpráv v rámci Internetu používány textové hlavičky, tak prvním vylepšením protokolu spočívalo v doplnění hlaviček do požadavku klienta a odpovědi serveru. Textová hlavička ve tvaru „Jméno: hodnota“ umožňuje neomezený budoucí vývoj protokolu doplňováním dalších hlaviček. Základní sada hlaviček popisuje typ a atributy dokumentu a vychází ze standardu MIME <sup>5</sup>. Ostatní hlavičky slouží pro přenos pomocných informací a předávání parametrů mezi klientem a serverem. (Reschke, a další, 2007)

Problém po zavedení hlaviček do požadavků a odpovědí nastal při rozlišování mezi původní verzí protokolu od verze s hlavičkami. Řešení bylo rozlišit, zda se jedná o nový nebo původní požadavek a odpověď protokolu. Toto rozlišení bylo dosaženo doplněním verze protokolu za jménem požadovaného dokumentu. Tato podoba HTTP protokolu je používána do dnes. (Margaret Rouse, 2006)

### 3.2.1 SSL

SSL<sup>6</sup> je obecný transportní protokol, který zajišťuje šifrovanou přenosovou cestu. SSL lze použít jak pro HTTP (HTTPS<sup>7</sup>), tak i FTP<sup>8</sup> (FTPS<sup>9</sup>), telnet<sup>10</sup> apod. Protokol zajišťuje soukromí a spolehlivost pro komunikující aplikace, chrání data před odposloucháním, paděláním a zfalšováním. (Jikos, 2012)

SSL je složen z:

- *Record Protocol*- nižší vrstva, zodpovědná za „zabalení“ dat protokolů do vyšší vrstvy
- *Handshake Protocol* – zajišťuje vytvoření bezpečného spojení a ověření stran mezi klientem a serverem (určeno na základě ověření a odsouhlasení šifrovacího algoritmu a klíčů)

---

<sup>5</sup> Multipurpose Internet Mail Extensions, RFC1521

<sup>6</sup> Secure Sockets Layer

<sup>7</sup> Hyper Text Transfer Protocol Secure

<sup>8</sup> File Transfer Protocol

<sup>9</sup> File Transfer Protocol Secure

<sup>10</sup> Telecommunication Network

### 3.2.2 Stavové zprávy HTTP

Stavový kód protokolu HTTP je součástí hlavičky odpovědi serveru na klientský požadavek. Tento kód říká, jak byla odpověď serverem zpracována. Určuje, jestli byl požadavek vyřízen buď kladně, záporně nebo došlo k chybě. Poté záleží jen na klientovi, aby se rozhodl na základě stavového kódu jaké patřičné kroky je zapotřebí provést.

V hlavičce odpovědi spolu se stavovým kódem je stavové hlášení ve formě slovního popisu stavového kódu. Zde se nachází pět kategorií podle charakteru stavového hlášení. Kód je trojice čísel, kde v prvním čísle je určena kategorie odpovědi a zbylá čísla ji pouze blíže specifikují. (Elizabeth, a další, 2006)

- 1xx - Informační kódy
- 2xx - Úspěch vyřízení požadavku
- 3xx - Přesměrování
- 4xx - Chyba na straně klienta
- 5xx - Chyba na straně serveru

### 3.4 Historie a vývoj HTML

První definice jazyka HTML byla vytvořena v roce 1991 a jejím autorem byl Tim Berners-Lee. Projekt vznikl jako součást projektu WWW, který měl umožnit vědcům zabývajícím se fyzikou vysokých energií komunikaci a sdílení výsledků výzkumu po celém světě. Ne náhodou proto celý projekt vznikl v CERNu. Jednalo se o výsledek dvouletého projektu, který měl vyřešit problémy se sdílením informací ve velké instituci jako je CERN. Tato verze umožňovala text rozčlenit do několika logických úrovní, použít několik druhů zvýraznění textu a zařadit do textu odkazy a obrázky. Berners-Lee při návrhu HTML nepředpokládal, že by autoři webových stránek museli tento jazyk znát. První verze WWW softwaru byla napsána pro operační systém NextStep a obsahovala jak prohlížeč, tak i integrovaný editor webových stránek. Software vyvinutý v CERNu včetně specifikací jazyka HTML byl veřejně uvolněn v roce 1991. Požadavky uživatelů na WWW vzrůstaly, a tak producenti různých prohlížečů obohacovali HTML o některé nové prvky. Aby byla zachována kompatibilita mezi jednotlivými modifikacemi HTML, vytvořil

Berners-Lee pod hlavičkou IETF<sup>11</sup> návrh standardu HTML 2.0, který zahrnoval všechny v té době běžně používané prvky HTML. Verze HTML 2.0 má zároveň dvě úrovně. První z nich (Level 1) pouze málo rozšiřuje předchozí verzi HTML. Level 2 navíc definuje práci s formuláři. Následující rozšíření jazyka je známo jako HTML+, zahrnují zejména rozšíření HTML o vytváření tabulek a matematických vzorců. Rovněž se zde objevují prvky, které umožňují precizněji kontrolovat výsledný vzhled textu například lepší obtékání obrázků textem a styly dokumentů. Dave Raggett<sup>12</sup> z laboratoří Hawlett-Packard HTML+ formalizoval a vytvořil jeho definici jako DTD<sup>13</sup> v jazyce SGML. Na jaře roku 1995 tak vznikl návrh standardu HTML 3.0. SGML je metajazyk, který slouží k definici jiných jazyků tedy i HTML. Této definici se pak říká DTD. Vývoj standardů Webu v té době již koordinovalo konsorcium W3C<sup>14</sup>, jehož členy jsou mimo jiné přední softwarové firmy. Členové W3C se tedy shodli na vlastnostech, o které rozšíří HTML 3.0, a vytvořili tak HTML 3.2. HTML 3.2 však zdaleka neobsahuje vše z HTML 3.0. Z verze 3.0 zůstaly v podstatě jen okleštěné tabulky. Ostatní nové prvky HTML 3.2 jsou jen jakousi směskou vlastností a funkcí, kterou v té době podporovaly nejnovější prohlížeče. Kromě tabulek přibýly ve verzi HTML 3.2 zejména možnosti lepší kontroly formátování včetně mnohem volnějšího výběru použitých druhů písma. Další podstatné rozšíření se týkalo podpory Java appletů. HTML 3.2 neslo kódové označení Wilbur a od ledna 1997 bylo doporučením konsorcia W3C což znamenalo, že by jej měli všichni používat, aby byla ve Webu zajištěna kompatibilita. Na jaře roku 1997 zveřejnilo W3C další plány na rozšíření HTML pod kódovým názvem *Cougar*. *Cougar* v sobě zahrnuje HTML 3.2 společně s běžně používanými konstrukcemi, jako jsou rámy, skripty a obecné vkládání objektů. Dalšími novinkami byla podpora vícejazyčných dokumentů. (Bordás, 2006)

Na začátku července 1997 uveřejnilo W3C návrh HTML 4.0, který vznikl drobnými úpravami *Cougaru* a vytvořením jediného komplexního dokumentu popisujícího návrh standardu. V prosinci 1997 pak bylo HTML 4.0 přijato jako standard W3C. Od té doby se na dlouhou dobu samotný vývoj jazyka HTML v podstatě zastavil. V prosinci 1999 byl vydán nový standard HTML 4.01, který pouze opravuje některé drobné chyby v předchozí specifikaci. Krátce po vydání specifikace HTML 4.0 zveřejnilo W3C ještě

---

<sup>11</sup> Internet Engineering Task Force

<sup>12</sup> Dave Raggett (8. 6. 1955) zdroj: <http://www.w3.org/People/Raggett/profile.html>

<sup>13</sup> Document type definition

<sup>14</sup> World Wide Web Consortium

jeden standard – jazyk XML<sup>15</sup>. Ten se od svého vydání v roce 1998 stal ve světě informačních technologií všude přítomným formátem pro výměnu a ukládání dat. XML ovlivnilo i další vývoj jazyka HTML. Na začátku roku 2000 byla vydána specifikace jazyka XHTML 1.0, která měla svoji syntaxi odvozenou od XML, nikoli od SGML jako HTML. V zásadě však šlo jen o kosmetickou změnu. Repertoár dostupných značek a jejich význam zůstal stejný, jen se drobně změnila syntaxe. (Jiří Kosek, 2012)

Bohužel historie ukázala, že samotné XHTML nebylo pro další vývoj webu nejlepší cestou. Jazyky XHTML 1.0 a 1.1 nepřidaly oproti HTML 4.01 téměř žádnou novou funkčnost. Kromě jisté módnosti tak nebyl pragmatický důvod pro masový přechod od HTML k XHTML. Nenapomohl k tomu ani fakt, že statisticky nejpoužívanější prohlížeč Internet Explorer neuměl XHTML stránky zpracovávat úplně korektně. Na půdě W3C pak začala pracovní skupina XHTML vyvíjet další verzi jazyka XHTML 2.0, která přinášela mnohé zajímavé vlastnosti. Bohužel za cenu porušení kompatibility s předchozími verzemi jazyka HTML a XHTML. Pro výrobce prohlížečů to byl příliš revoluční krok, preferovali konzervativnější přístup postupného vylepšování stávajících funkčních technologií HTML4.01 a XHTML1.0. Nespokojenost výrobců prohlížečů pak dospěla tak daleko, že někteří z nich v roce 2004 založili společnou pracovní skupinu WHATWG<sup>16</sup>. Na její půdě pak společně připravovali specifikaci platformy pro webové aplikace běžící v prohlížeči. Kromě rozšíření jazyka HTML specifikace zahrnovala i definice důležitých rozhraní pro využití ve skriptovacím jazyku JavaScript<sup>17</sup>. Rozštěpení dalšího vývoje HTML samozřejmě nebylo žádoucí. Po dlouhých a složitých diskusích a jednáních nakonec i uvnitř W3C převážil názor, že XHTML 2 je slepá cesta, protože výrobci prohlížečů jej ignorují. V roce 2007 se proto síly W3C a WHATWG spojily a pracovní skupina HTML vzala jako základ specifikace vytvořené ve WHATWG a na jejich základě začalo vznikat HTML5. (Garth, neuvědno)

HTML5 navazuje na HTML 4.01 a přidává pro vývojáře webových aplikací spoustu užitečných funkcí, kvůli kterým je výhodné tuto nejnovější verzi jazyka používat. Zároveň zachovává možnost používat pro zápis stránek syntaxi XML a používat tak

---

<sup>15</sup> eXtensible Markup Language

<sup>16</sup> The Web Hypertext Application Technology Working Group

<sup>17</sup> Skriptovací jazyk

víceméně XHTML. Finální specifikace HTML5 byla vydána 28. října 2014. (Jiří Kosek, 2012)

### 3.4.1 Vývojáři HTML5

U vývoje HTML5 je zřejmé, že je nutné určité rozdělení odpovědností, proto jsou zde tři organizace, které jsou zodpovědné za specifikaci HTML5. (Albers, a další, 2011)

- WHATWG (Web Hypertext Application Technology Working Group) – Organizace založená v roce 2004 několika jednotlivci pracujícími pro společnosti vyvíjející prohlížeče, jako je Apple, Mozilla, Google, Opera. Vyvíjí HTML a rozhraní API pro vývoj webových aplikací a otevřeně spolupracuje s tvůrci prohlížečů a dalšími zainteresovanými stranami.
- W3C (WorldWide Web Consortium) – Součástí konsorcia W3C je i pracovní skupina zodpovědná za HTML, která má momentálně na starosti dodání specifikace HTML5.
- IETF (Internet EngineeringTaskForce) – Součástí této organizace jsou skupiny odpovědné za Internetové protokoly, jako je HTTP (Hypertext Transfer Protocol). HTML definuje nové rozhraní API WebSocket, které staví na protokolu WebSocket, jež vyvíjí jedna z pracovních skupin IETF.

## 3.5 Porovnání HTML5 a HTML 4.01

Tato část se zabývá základními rozdíly mezi novým a starým standardem a dále se věnuje popisu základních elementů v HTML5. U každého elementu je znázorněna jednoduchá ukázka zdrojového kódu.

### 3.5.1 Deklarace typu dokumentu

Jeden z hlavních rozdílů mezi HTML5 a HTML4.01 je změna tzv. DTD neboli deklaráce typu dokumentu. Každý správně formátovaný XHTML dokument by měl na začátku obsahovat informace o verzi použitého HTML a typu DTD. DTD je jinými slovy

návod pro prohlížeč zpracovávající dokument. Říká mu, jaké elementy dokument používá a jak s nimi zacházet. Element DOCTYPE není součástí XHTML dokumentu, není elementem XHTML a nemusí mít koncovou značku. (Janovský, a další, 2014)

Deklarace typu HTML 4.01

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

Delší a méně srozumitelná deklarace s odkazem na stránky *www.w3.org*.

Deklarace typu HTML5

```
<!DOCTYPE html>
```

Deklarace typu HTML5 se stala jednodušší a lépe zapamatovatelnou.

### 3.5.2 Strukturování

Obsah HTML dokumentu se nachází v elementu `<body>`. Většinou se však `<body>` dále rozděluje do několika sekcí s různým významem, které tvoří tzv. *layout*. Před příchodem HTML5 se sekce vytvářely pomocí elementů `<div>`, HTML5 však nyní přišlo s novými značkami právě pro tvorbu *layoutu*, které vždy dávají dané sekci určitý význam. To pomůže např. vyhledávačům lépe pochopit obsah stránky. (Hogan, 2011)

Layout se s postupem času vyvíjel. V předchozích verzích se k jeho tvorbě využívaly rámce nebo tabulky. S příchodem dynamických stránek se skládal z elementů `<body>` odlišených prostřednictvím popisných atributů `id` a `class`. Ty jsou nyní nahrazeny novými elementy. Z hlediska ergonomie je výraznou změnou přesunutí navigačního menu nahoru, které nejčastěji vypadá jako vodorovný řádek. Dříve byla navigační oblast nejčastěji jako levý sloupec. Součástí *layoutu* zůstala hlavička a patička. Objevila se nová oblast v podobě pravého sloupce, který obsahuje doplňující obsah k článku. Tyto změny v koncepci webové stránky se občas označují termíny jako web 2.0 a web 3.0. (Slavoj, 2014)

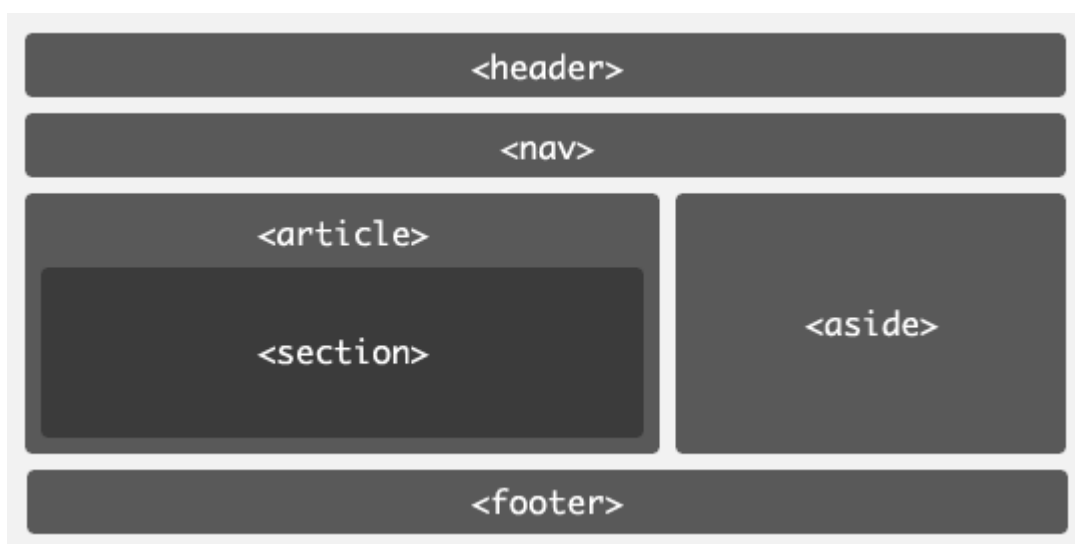
Příklad struktury stránky vytvořený pomocí elementu `<div>`.



Obrázek 1- Struktura pomocí <div>

Schéma znázorňuje klasický dvousloupcový *layout* vyznačený pomocí elementů <div> s atributy `id` a `class`. Obsahuje hlavičku <header> a patičku <footer>, dále navigační lištu <nav> pod hlavičkou. Hlavní obsah je tvořen článkem <article> a vedlejším sloupcem umístěným vpravo <sidebar>. Důvod použití elementu <div> je dáno z důvodu, že verze HTML4.01 postrádá sémantické prvky pro přesnější vyznačení částí dokumentu. HTML5 řeší tento problém pomocí nově zavedených elementů, které reprezentují jednotlivé části dokumentu. Znázorněno na Obr. 2.

*Příklad struktury stránky vytvořený pomocí nových elementů HTML5:*



Obrázek 2 - Struktura pomocí elementů HTML5



Kód struktury využívající nových elementů HTML5 (Obr. 2) by mohl vypadat takto.

```
<body>
  <header>...</header>
  <nav>...</nav>
  <article>
    <section>
      ...
    </section>
  </article>
  <aside>...</aside>
  <footer>...</footer>
</body>
```

### 3.5.3 Přehled nových elementů HTML5

Verze HTML5 přinesla nové elementy, které zpřehledňují kód. Zde je výčet nově přidaných elementů a seříděné podle jejich funkce. (Albers, a další, 2011)

**Skripty:**

```
<template>
```

**Sekce:**

```
<section>; <nav>; <article>; <aside>; <header>; <footer>; <main>; <hgroup>
```

**Členění obsahu:**

```
<figure>; <figcaption>
```

**Sémantika na úrovni textu:**

```
<data>; <time>; <mark>; <ruby>; <rt>; <rp>; <bdi>; <wbr>
```

**Vložený obsah:**

```
<embed>; <video>; <audio>; <source>; <track>; <canvas>; <svg>; <math>
```

**Formuláře:**

```
<datalist>; <keygen>; <output>; <progress>; <meter>
```

**Interaktivní elementy:**

```
<details>; <summary>; <menuitem>; <menu>
```

### 3.5.3.1 Popis strukturálních elementů

#### Article

Tento element reprezentuje část obsahu stránky, který tvoří samostatnou a nezávislou část. Může se použít například pro článek nebo komentář. `<Article>` může obsahovat další prvky jako `<header>`, `<footer>` nebo jiný `<article>`. Element je párový. (Churý, 2012)

Ukázka zdrojového kódu:

```
<article>
  <header>
    <h2>Název článku</h2>
    <time datetime="2015-01-01">01. ledna 2015</time>
  </header>
  <p>Obsah, nebo popis článku</p>
</article>
```

Jedná se o článek v elementu `<article>`, který obsahuje hlavičku `<header>` a nadpis druhé úrovně `h2`, kde je uveden název článku a časový údaj. Ten je demonstrován pomocí elementu `<time>` s atributem `<datetime>`. Po uzavření hlavičky následuje element `<p>` s obsahem daného článku.

#### Section

Do elementu `<section>` by se měl zabalovat obsah, který nejde ze stránky vyjmout, neboli kdyby se vyjmul, narušila by se logika stránky. Proto není `<section>` alternativou `<article>`. Například `<article>` lze použít pro články, které jsou samostatné a nezávislé na svém okolí, protože když se jeden odstraní, nenaruší logiku stránky. Element `<section>` by se také neměl používat v rámci vázání kaskádových stylů. Element je párový. (Hogan, 2011)

Ukázka zdrojového kódu:

```
<section>
  <h2>Nadpis</h2>
  <p>Text</p>
</section>
```

Element `<section>` obsahuje nadpis druhé úrovně `h2` a samotný text obalený v elementu `<p>`.

## Nav

Element `<nav>` představuje hlavní navigaci, neboli menu stránky `<nav>` není `<menu>`, element `<nav>` lze tedy použít pro navigaci, která bude obsahovat odkazy na hlavní části stránky nebo na stránky s podobným obsahem. Prvek nesmí být potomkem elementů `<address>`. Element je párový. (Churý, 2012)

Ukázka zdrojového kódu:

```
<nav>
<ul>
  <li><strong>Navigace</strong></li>
  <li><a href="http://web1.com">Úvodní stránka</a></li>
  <li><a href="http://web2.com/?akce=diskuze">Diskuze</a></li>
  <li><strong>Odkazy</strong></li>
  <li><a href="http://web3.com" target="_blank">web3.com</a></li>
</ul>
</nav>
```

Element `<nav>` obsahuje další element `<ul>`, který má v sobě elementy `<li>`. Tato forma zápisu se používá většinou pro vytvoření navigační lišty na stránkách. Element `<li>` obsahuje element `<a>` s atributem `href`. Ten slouží k odkazu na danou stránku pomocí zadané adresy.

## Menu

Element `<menu>` již je obsažen v HTML4.01. Nyní je v HTML5 přepracovaný a slouží jako seznam ovládacích prvků pro formuláře. Tento prvek nesmí být potomkem elementů `<a>` ani `<button>`. Element je párový. (Churý, 2012)

Možné atributy:

- **type** = neboli druh nabídky, hodnotami jsou: `toolbar` (panelová nabídka) nebo `context` (kontextová nabídka)
- **label** = popisek v nabídce, hodnotou je libovolný text

Ukázka zdrojového kódu:

```
<form action="" method="post">
  <strong>Barva:</strong>
  <menu>
    <li><input type="checkbox">Bíla</li>
    <li><input type="checkbox">Černá</li>
  </menu>
</form>
```

Element `<form>` obsahuje atribut `method`, který slouží k předávání dat. V elementu `<menu>` jsou dva elementy `<li>` a uvnitř elementu `<input>` je atribut `type`, který má hodnotu `checkbox` což znamená, že na stránce zobrazuje jako zaškrťovací pole s textem červená nebo bíla.

## Header

Element `<header>` představuje hlavičku své sekce nebo hlavičku celé stránky. Lze jej využít například v článku pro vložení nadpisu (elementy `h1-h6`), spolu s obsahem, jako je například úvodní materiál nebo navigační pomůcky pro článek, v případě celé stránky může obsahovat logo, uživatelské rozhraní a podobně. Prvek nesmí být potomkem elementů `<header>`, `<address>` a `<footer>`. Ukázku najdete níže s elementem `<footer>`. Element je párový. (Churý, 2012)

## Footer

Element `<footer>` představuje patičku své sekce nebo patičku celé stránky. Lze jej použít například v článku pro představení autora nebo pro odkazy na související dokumenty, informace o autorských právech apod. Prvek nesmí být potomkem elementů `<header>`, `<address>` a `<footer>`. Element je párový. (Churý, 2012)

Ukázka zdrojového kódu:

```
<article>
  <header>
    <h2>Název článku</h2>
    <time datetime="2015-01-01">01. ledna 2015</time>
  </header>
  <p>Obsah nebo popis článku</p>
  <footer>
    <a href="http://web1.com">Článek</a>
  </footer>
</article>
```

Element `<article>` obsahuje další dva elementy `<header>` a `<footer>`. Tyto dva elementy slouží spíše pro lepší orientaci a přehlednost v samotném kódu. Každý z nich může mít vložené další elementy.

### **hgroup**

Tento element patří k nově přidaným v HTML5. Používá se čistě pro seskupení nadpisů a podnadpisů. Na rozdíl od `<header>` by neměl obsahovat nic než nadpisy `h1-h6`. Element je párový. (Churý, 2012)

Ukázka zdrojového kódu:

```
<article>
  <hgroup>
    <h1>Nadpis první úrovně</h1>
    <h2>Nadpis druhé úrovně</h2>
  </hgroup>
  <p>Text</p>
</article>
```

Element `<article>` obsahuje element `<hgroup>`, který obsahuje dvě úrovně nadpisu `h1` a `h2`. Sloučí pouze k seskupení nadpisů a lepší přehlednosti v kódu.

## Figcaption a figure

Element `<figcaption>` umožňuje vložení doplňkového textu, například k obrázku nebo zdrojovému kódu. `<figcaption>` je potomek `<figure>`. V ukázce zdrojového kódu je vidět jak `<figure>` a `<figcaption>` fungují. Elementy jsou párové. (Churý, 2012)

Ukázka zdrojového kódu:

```
<figure>
  
  <figcaption>Obrázek není předmětem tohoto kódu </figcaption>
</figure>
```

V elementu `<figure>` se nachází obrázek vložený pomocí tagu `img` s atributem odkazu `src` a atributem `alt`. Dále ve `<figure>` je potomek element `<figcaption>`. Tento element slouží jako doplňkový text k obrázku s konkrétním textem.

## Canvas

Element slouží jen jako plátno pro dynamické vykreslování obrázků, grafiky, her apod. Zjednodušeně je to kontejner pro grafiku, pro skutečné vykreslení je nutné použití skriptu. Element je párový. (Churý, 2012)

Možné atributy:

- **width** = šířka plátna (hodnotami jsou délkové míry)
- **height** = výška plátna (hodnotami jsou délkové míry)

## time

Element `<time>` označuje časové údaje, nabízí nastavení vlastního formátu data a času, použití by měla nastínit ukázka. Element je párový. (Churý, 2012)

Možné atributy:

- **datetime** = datum, čas nebo datum i čas, které element představuje; hodnotou je časový formát
- **pubdate** = označuje datum publikování například článku, v případě použití tohoto atributu by měl být celý element `<time>` potomkem elementu `<article>`; bez hodnoty

Ukázka zdrojového kódu:

```
<time datetime="2015-1-1"> 1.ledna 2015 </time>
<time datetime="2015-1-13T20:00" pubdate> 13.ledna 2015 ve 20:00 </time>
<time datetime="2015-1-13T20:00+09:00"> 13.ledna 2015 ve 20:00 v Tokiu
</time>
```

V tomto případě je tu element `<time>` s třemi různými způsoby zápisu. První případ obsahuje pouze datum, druhý případ využívá atributu `pubdate` jakožto publikované datum článku a poslední případ obsahuje datum, kde se přičítá +9h k aktuálnímu času.

## a

HTML5 zásadně nezasahuje do elementu `a`, ale jen ho rozšiřuje o další možnosti. Nyní už se nemusí odkazovat pouze obrázkem nebo textem, ale lze do hypertextového odkazu zabalit jakékoliv, i blokové elementy. Element je párový. (Churý, 2012)

Možné atributy:

- **href** = adresa stránky, na kterou se má odkazovat; hodnotou je URL
- **target** = způsob otevření stránky, na kterou se odkazuje; hodnotami jsou `_blank` (nový panel), `_top` (nové okno), `_parent` (nadřazený rám) nebo `_self` (aktuální panel)
- **rel** = druh odkazu; hodnotami jsou `alternate`, `stylesheet`, `start`, `next`, `prev`, `contents`, `index`, `glossary`, `chapter`, `section`, `subsection`, `appendix`, `help` a `bookmark`
- **hreflang** = jazyk stránky, na kterou odkaz směřuje; hodnotami jsou národní zkratky

Ukázka zdrojového kódu:

```
<a href="http://WEB1e.com" target="_blank" hreflang="cs">
  <h2>HTML5 - nové vlastnosti</h2>
  <p> vlastnosti jazyka HTML 5</p>
</a>
```

Zde je znázorněný příklad elementu `a`, který využívá atributu `href`. Tento atribut slouží k vložení URL adresy. Uvnitř elementu je i element `h2` jakožto nadpis a atribut `p` jako pouhý text.

## **aside**

Element `<aside>` definuje nějaký obsah, který by měl být spojen s hlavním obsahem stránky, lze jej tedy využít například pro boční panel na stránce. Nesmí být potomek elementu `<address>`. Element je párový. (Churý, 2012)

## **cite**

Přestože element `<cite>` byl obsažen již v předchozích verzích HTML, tak v novém HTML5 se upřesňuje jeho využití, a to pro označení titulku díla, ze kterého je citováno. Element je párový. (Churý, 2012)

## **details**

Element `<details>` se používá k popisu podrobností o dokumentu nebo části dokumentu, ze kterého může uživatel získat další informace. Element nesmí být potomkem elementu `<button>`. Element je párový. (Churý, 2012)

Možné atributy:

- **open** = obsah bude uživateli viditelný; bez hodnoty

## **hr**

Element `<hr>` je už asi taky všem známý, ale až v HTML5 dostal zvláštní sémantický účel představující "tematickou přestávku textu". Využit lze pro rozdělení textu na více částí. Element není párový. (Churý, 2012)

## **small**

Element `<small>` je také znám z předchozích verzí HTML, nyní mu však přibyl nový sémantický význam, a to význam drobné poznámky nebo doplňku k textu, který bude zobrazen malým drobným písmem. Element je párový. (Churý, 2012)

## **summary**

Element `<summary>` je pouze titulek, který se používá především pro prvek `details`. Element je párový. (Churý, 2012)

## **mark**

Element `<mark>` představuje zvýraznění nebo označení určitého textu. Lze ho využít například pro zvýraznění vyhledávaných slov ve výsledku hledání. Výchozí nastavení



zobrazení je pro text červená barva a žlutá barva pro jeho pozadí. Element je párový. (Churý, 2012)

Ukázka zdrojového kódu

```
<details>
  <summary>Další informace</summary>
  <p>Další informace například zdroj dat k článku </p>
</details>
```

Element `<details>` obsahuje element `<summary>`, který má roli titulku a element `<p>` jako samostatný text.

### **meta**

Element `<meta>` je víceúčelový element, který je již znám z předchozích verzí HTML, nyní byly jeho atributy trochu pozměněny, z nabídky jeho atributů zmizel `http-equiv=content-language` a přibyl `charset`, který slouží pro nastavení kódování v HTTP hlavičce dokumentu. Jedná se pouze o ukázkou nového zápisu oproti předchozímu. (Churý, 2012)

Ukázka kódu pro HTML 4.01

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

Ukázka kódu pro XHTML 1.1

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

Ukázka kódu pro HTML5

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

### **video**

V minulosti se video do webů vkládalo pomocí *Adobe Flash*. Dnes se však dává přes nové funkce HTML5. Video byla jedna z prvních funkcí HTML5, která začala masově z trhu vytlačovat *Flash*.

`<video>` je párový element, sloužící k vložení videa do HTML dokumentu. Je podporován všemi moderními prohlížeči, ve starších prohlížečích se zobrazí alternativní text, uvedený mezi elementy `<video>` a `</video>`. (Churý, 2012)

Element má následující atributy

- **autoplay** - Atribut je typu `boolean`. Pokud je uveden, spustí se video automaticky, ihned po načtení. Pokud ne, musím na video kliknout pravým tlačítkem a zvolit Přehrát (pokud nemáme zapnuté `controls`)
- **controls** - Atribut je typu `boolean`. Pokud je uveden, zobrazí se přehrávač s ovládacími prvky. Pokud ne, zobrazí se pouze samotné video a jediná možnost ovládání je přes kontextové menu
- **height** - Nastavení výšky přehrávače
- **loop** - Atribut je typu `boolean`. Pokud je uveden, je zvuková stopa přehrávána s opakováním stále dokola
- **muted** - Atribut je typu `boolean`. Pokud je uveden, je zvuk vypnutý
- **poster** - URL obrázku, který se zobrazí na pozadí přehrávače, dokud uživatel video nespustí
- **preload** - Umožňuje nastavit načítání média do paměti, což urychlí jeho přehrání po kliknutí, ale zpomalí načtení stránky. Při hodnotě `auto` se po načtení stránky načte do paměti celý video soubor. Při hodnotě `metadata` se načte pouze hlavička. Při hodnotě `none` se soubor do paměti nenačte
- **src** - Specifikuje umístění video nebo audio souboru. Soubor se může vložit přímo, pomocí atributu `src` nebo pomocí elementu `source`
- **width** - Nastavení šířky přehrávače.

Vložení video souboru do HTML stránky

```
<video src="video/video.mp4" width="320" height="240" controls="controls">
  Váš prohlížeč nepodporuje vkládání video souborů
  <a href="video/video.mp4">stáhněte</a>.
</video>
```

V elementu `<video>` jsou atributy `src` v kterém je stanovena cesta k souboru, dále atributy `width` a `height` určují šířku a výšku okna s videem a atribut `controls` zobrazí ovládací prvky videa/audia. Pokud prohlížeč z nějakého důvodu nedokáže přehrát video, ukáže se text, že video není podporováno. V poslední řadě je odkaz na stáhnutí videa, který je tvořen elementem `<a>` a tributem `href` s odkazem na video.

## 3.6 CSS

Kaskádové styly umožňují definovat vlastnosti určitých elementů např. barvy textu, pozadí, zarovnání, velikost písma, přidání efektů a jiné. HTML kód nedisponuje ničím podobným. Pomocí CSS lze jednoduše připravit stránku se stejnými nadpisy, odstavci, seznamy atd. CSS lze zapisovat přímo do HTML dokumentu nebo do externího souboru s příponou `.css`, který se připojí k HTML stránce.

Názorný příklad jak lze připojit externí souboru `.css`

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

Odkaz na soubor `.css` se vkládá do elementu `<head>`.

Příklad vložení prostřednictvím prvku `<style>`

```
<!DOCTYPE html>
<head>
<title>Vložení prostřednictvím prvku</title>
    <style type="text/css">
        body {
            color: white;
            background-color: black;
        }
    </style>
</head>
```

Pomocí prvku `<style>`, se stylový předpis vloží přímo do stránky. Použitý atribut `type`, má stejnou hodnotu jako u odkazu `text/css`.

Příklad vložení atributem `style`

```
<p style="color: red; text-align: left">Tento jediný
odstavec bude červený a zarovnaný vlevo</p>
```

Každému prvku na stránce, lze přidat atribut `style` a v jeho hodnotě předepsat hodnoty CSS vlastností.

### 3.6.1 Historie CSS

CSS ( *Cascading Style Sheets* ) neboli „kaskádové styly“ vznikly kolem roku 1996. Tento jazyk byl navržen standardizační organizací W3C, autor prvotního konceptu byl Hakon Wium Lie<sup>18</sup>. Snahou CSS bylo oddělit informaci o obsahu od formátovacích pokynů. Přibližně o rok později vyšla první verze kaskádových stylů CSS 1.0.

#### 3.6.1.1 CSS 1

Mezi první prohlížeč, který podporoval CSS, byl Internet Explorer verze 3 (rok 1996). Jednalo se zatím o omezenou podporu v rámci změny písma a barvy.

Teprve větší podpory se CSS1 dočkalo, až ve čtvrtých verzích Internet Explorer a Netscape Navigator (1998). Tyto prohlížeče se snaží podporovat celé CSS1 včetně pozicování. Přesto realita byla jiná. To byl i jeden z hlavních důvodů, proč se kaskádové styly nedostaly dlouho do oblíbenosti širokých vrstev webmasterů. (Albers, a další, 2011)

#### 3.6.1.2 CSS 2

Druhá verze kaskádových stylů přináší více vlastností pro stylování několika dalšími parametry. Konečná verze se ustanovila přibližně kolem roku 2000. Ovšem s příchodem CSS 2 nastaly problémy v korektním zobrazování nových prvků v různých webových prohlížečích a jejich verzích. Postupem času se však tyto problémy téměř vyřešily. (Elizabeth, a další, 2006)

#### 3.6.1.3 CSS 3

Vývoj byl zahájen roku 2005 konsorciem W3C, jedná se o třetí verzi kaskádových stylů. CSS3 bylo oficiálně dokončeno v roce 2015, ale ještě před dokončením, některé prohlížeče podporovaly nové vlastnosti. (Albers, a další, 2011)

### 3.6.2 Význam CSS

Před nástupem CSS se HTML dokumenty formátovaly jen pomocí prostředků jazyka HTML, který nabízí poměrně široké spektrum značek a atributů. CSS k formátovacím prostředkům se představuje jako vhodná alternativa, popřípadě doplněk, ovšem význam CSS je širší. (Elizabeth, a další, 2006)

---

<sup>18</sup>Hakon Wium Lie (1965) zdroj: [https://www.evi.com/q/hakon\\_wium\\_lie\\_biography](https://www.evi.com/q/hakon_wium_lie_biography)

## Seznam využití CSS v praxi

- širší formátovací možnosti
- snadná tvorba a údržba konzistentního stylu
- oddělení struktury a stylu
- vyšší přístupnost dokumentu
- dynamická práce se styly

### 3.6.3 Základní selektory

Další podstatná část CSS jsou selektory, které určují, na které elementy dokumentu se použijí vlastnosti nastavené v deklaraci. Selektory berou dokument HTML jako stromovou strukturu, složenou z jednotlivých elementů. Selektor si poté vybere určité uzly stromu, na které aplikuje své pravidlo. (Etemad, a další, 2011)

#### Výběr elementu podle jména

Nejjednodušším selektorem je název elementu. Jakmile se použije pro obecný element, například `<h3>` (nadpis 3. úrovně), poté se pravidlo aplikuje pro všechny elementy `<h3>` použité v HTML dokumentu. (Jahoda, 2014)

#### Příklad použití pro element `<h3>`

```
h3 {font-size:14px; font-weight: normal; color:#0e0a07;}
```

Element `<h3>` má atributy `font-size`, tedy velikost písma konkrétně v pixelech, dále `font-weight`, který určuje tučnost písma a poslední atribut `color` nastaví barvu písma.

#### Třída jako selektor

Použití třídy jako selektor je vhodné, pokud chceme stejný element zobrazit v různých případech rozdílně. Proto lze u každého elementu, který patří do těla dokumentu, neboli v elementu `<body>`, určit jeho třídu použitím atributu `class`. (Etemad, a další, 2011)

#### Příklad použití třídy jako selektor

```
<div class="priklad1">...</div>
```

V definici stylu lze selektor s názvem elementu doplnit o název třídy. Název třídy se od jména elementu odděluje tečkou

```
div. priklad1 { font-size:25px; color: red; margin-left: 5px; }
```

Element `<div>` má atribut `class`. Tento atribut obsahuje hodnotu `priklad1`, tato hodnota slouží k rozeznání v CSS, které poté stylují všechny elementy `<div>` se stejným názvem atributu `class`.

### Identifikátor elementu jako selektor

V tomto případě se jedná o tzv. unikátní identifikátor `id`. Měl by se používat v případě, když se na stránce vyskytuje pouze jednou. (Jahoda, 2014)

Příklad `id` jako selektor

```
#footer {margin: 10px; padding: 10px; font-size: 25px;}
```

Identifikátor `id` názvem `footer` nastyluje vnitřní okraj prvku `padding` a vnější okraj prvku `Margin` o 10px z každé strany a velikost písma bude 25px.

### 3.6.4 Pseudotřídy

Pseudotřídy jsou typ selektorů, které nejsou přímo svázány s určitým elementem. Prohlížeč si domyslí podle aktuální situace, na které části stránky se musí aplikovat. Nová specifikace CSS3 rozšiřuje pseudotřídy, tedy staré zůstávají a jsou přidány další nové. (Sládek, 2010)

- **E:link** - vlastnosti nenavštíveného odkazu
- **E:visited** - vlastnosti navštíveného odkazu
- **E:focus** - vlastnosti při označení pomocí klávesnice
- **E:hover** - vlastnosti při najetí kurzorem nad element (Od CSS 2 lze pseudotřídou `:hover` použít na jakýkoli element)
- **E:active** - vlastnosti právě při kliknutí

**E:root**

Vybere kořenový element dokumentu. V HTML bude vždy `<html>`. Který je na stránce vždy jeden. Podpora v IE<sup>19</sup> až od verze 9.

**E:empty**

Vybere prázdný element. Např. tedy prázdný element `<span></span>`. Podporu znovu nabízí všechny prohlížeče kromě IE8.

**E:target**

Vybere element, který cílíme v URI. Takže když odkazujeme konkrétní část dokumentu přes `dokument.html#cast-dokumentu`, takto ji lze uživateli zvýraznit. Podporu nabízí Mozilla, Webkit, IE9 a částečně i Opera.

**E:enabled x E:disabled**

Vybere element formuláře, do kterého je povolené/zakázané zapisovat. Podporu nabízí znovu všechny prohlížeče kromě IE8.

**E:checked**

Vybere `checkbox`, který je zaškrtnutý. Podporu nabízí znovu všechny prohlížeče kromě IE8.

**E::selection**

Po najetí na požadovaný text kurzorem, se vybraný text zvýrazní, typicky se podbarvuje určitou barvou. Pomocí této pseudotřídy `::selection` se barva může ovlivnit. Podporu nabízí všechny internetové prohlížeče vyjma IE8. U Mozilly FireFox se musí použít `-moz-::selection`.

**E:not(s)**

Vybere element `<E>`, který neodpovídá jednoduchému selektoru `s`. Podporu nabízí všichni kromě IE8.

---

<sup>19</sup> IE – Internet Explorer (Webový prohlížeč)

### 3.6.5 Nové grafické vlastnosti v CSS3

#### Box-shadow

Tato vlastnost umožňuje vytvořit u objektů vržený stín, který je určen pomocí čtyř parametrů.

- **x-offset** – horizontální posun stínu.

Záporná čísla posunují stín doleva, kladná doprava

- **y-offset** – vertikální posun stínu.

Záporná čísla posunují stín nahoru, kladná dolů

- **blur radius** – udává poloměr rozmazání, tj. okraj stínu, který přechází do ztracena
- **shadow color** – definuje barva stínu

Příklad Box-shadow

```
box-shadow: 5px 6px 20px #CCC;
```

Vytvoří šedivý stín ve směru ze shora o 5px, zleva o 6px a rozmáže stín v hodnotě 20px.

#### Text-shadow

Stejná vlastnost jako u `box-shadow`, jen s rozdílem, že se využívá pro stínování písma.

Příklad Text\_shadow

```
text-shadow:5px 10px 15px #CCC;
```

Vytvoří šedivý stín textu ve směru ze shora o 5px, zleva o 10px a rozmáže stín v hodnotě 15px.

#### Border-radius

Tato vlastnost dokáže u objektů vytvořit zakulacené rohy. Dříve se pro tyto účely používaly obrázky a podobně. `border-radius` poskytuje více možností zápisu.

Příklad Border-radius

```
border-radius:8px;
```

Tato vlastnost vytvoří zakulacení rohů v hodnotě 8px.



## Opacity

Neboli průhlednost, umožňuje nastavení průhlednosti požadovaného objektu. Hodnota v tomto případě je určena od 0 do 1, což představuje 0% až 100%.

Příklad opacity

```
<div style="opacity:0.7; width:50px; height:20px;
```

Tato vlastnost vytvoří u objektu průhlednost o opacitě 0.7 a o šířce 50px a výšce 20px.

## Transform

Pomocí této vlastnosti se může nastavit pohyb, rotaci nebo zkosení objektu. `transform` používá pět metod.

- **Translate()** - Posune objekt na určené souřadnice x,y.

Příklad `Translate()`

```
transform: translate(10em,5em);
```

- **Rotate()** - Otáčí objekt ve směru hodinových ručiček.

Příklad `Rotate()`

```
transform: rotate(20deg);
```

- **Scale()** – Slouží ke zvětšení objektu. Hodnota 1 představuje standartní velikost, hodnota 2 dvojnásobnou a hodnota 0,5 poloviční apod.

Příklad `Scale()`

```
transform: scale(2,4);
```

- **Skew** - Otáčí objekt v zadaném úhlu po daných osách.

Příklad `Skew`

```
transform: skew(30deg,15deg);
```

- **Matrix()** - Kombinaci všech předchozích metod. Posun, rotace, zvětšení.

Příklad `Matrix()`

```
transform:matrix(0.366,0.5,-0.5,0.366,0,0);
```

## Multi-column

Modul, který umožňuje do předem definovaných sloupců vložit text. Pro představu je obdobné jako v novinové vazbě.

Modu se skládá z několika vlastností.

- **column-width** – Určuje šířku sloupce
- **column-count** – Určuje počet sloupců
- **column-rule** – Určuje vlastnost, jak mají být dva sloupce odděleny
- **column-gap** – Určuje velikost mezery mezi sloupci
- **column-fill** – Určuje způsob vykreslení textu do sloupců
- **column-span** – Nastaví výjimku, která vyjme daný prvek ze sloupcového rozložení

Příklad Multi-column

```
.MultiCollumn {  
  column-count: 3;  
  column-gap: 20px;  
  column-rule: 4px outset red;  
}
```

Tento příklad vytvoří tři sloupce rozdělené červenou čarou o tloušťce 4px a velikostí mezery mezi sloupci 20px.

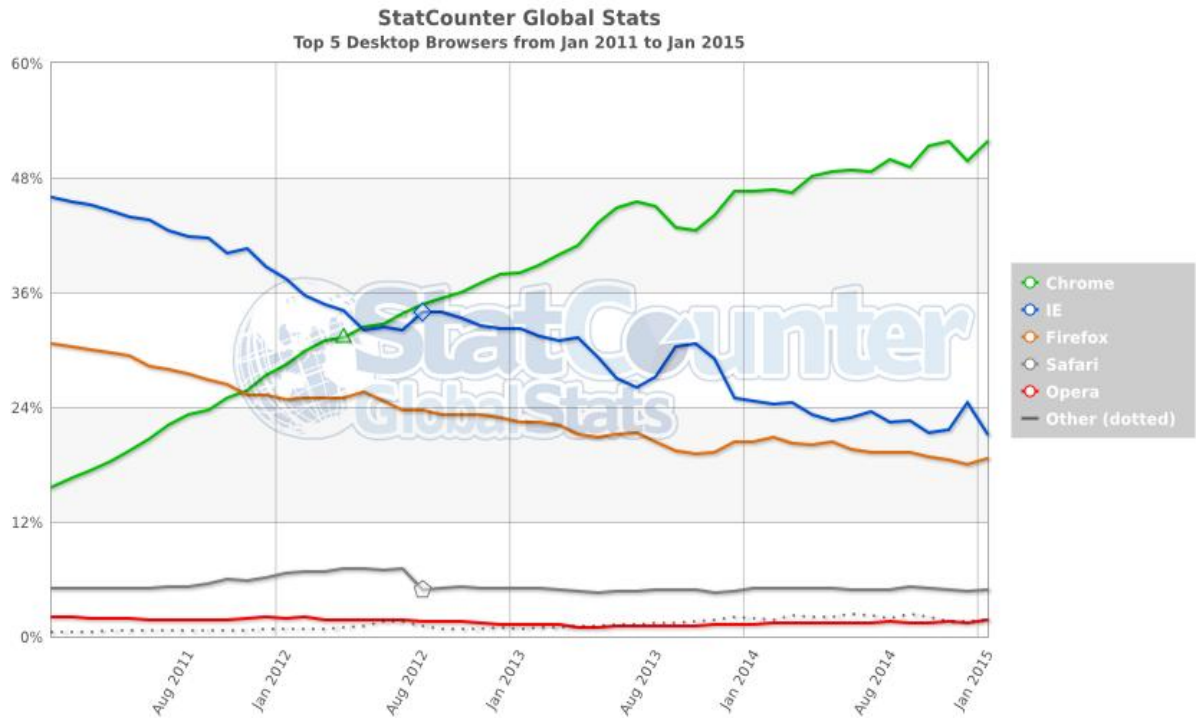
## 3.7 Webové prohlížeče a HTML5

Od chvíle kdy byla specifikace HTML5 oficiálně dokončena, dalo by se říci, že poslední verze vydaných webových prohlížečů plně podporují nové vlastnosti HTML5 a CSS3. Aby toto tvrzení bylo relevantní, nejpoužívanější webové prohlížeče podle statistiky ze serveru „[www.gs.statcounter.com](http://www.gs.statcounter.com)“ byly podrobeny testu. Test spočívá v analýze podpory nových vlastností HTML5.

### 3.7.1 Podíl prohlížečů na trhu

Ještě před oficiálním dokončením vývoje HTML5 v sobě měli nové verze webových prohlížečů podporu nových vlastností. Webové stránky, které byly vytvořeny využitím nových funkcí se dokázaly zobrazit v internetových prohlížečích, ale ne vždy korektně. Z velké většiny záleželo na použitém prohlížeči a jeho verzi a také jaká vlastnost nového standardu byla použita.

Tyto skutečnosti a jiné, mohly napomoci k tomu, že se popularita nejpoužívanějších prohlížečů v postupu času měnila. Graf (Graf 1) znázorňuje procentuální zastoupení nejpoužívanějších prohlížečů celosvětově, za období od ledna 2011 do ledna 2015.



**Graf 1 – Top 5 webových prohlížečů (od ledna 2011 do ledna 2015)**

Jedná se o prohlížeče:

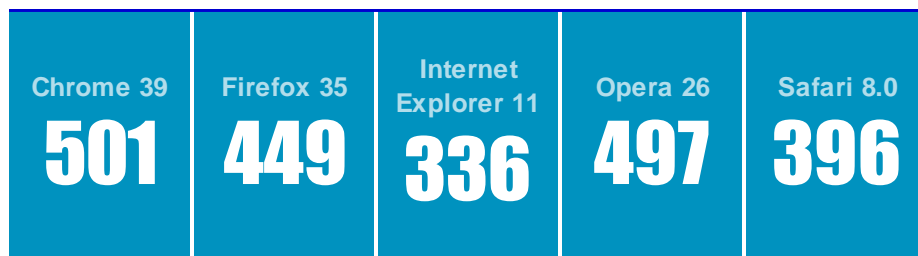
- Chrome
- Internet Explorer
- Firefox
- Safari
- Opera
- Ostatní (všechny ostatní desktopové prohlížeč, pohybující se okolo 1%)

Z grafu (graf 1) lze vyčíst, že na začátku roku 2011 měl Internet Explorer největší podíl na trhu, ale postupně v roce 2012 prohlížeč Chrome, porazil IE a dostal na první pozici. Dokonce ani nové verze Internet Exploreru nedokázaly porazit oblíbenost prohlížeče od Googlu. Třetí místo si udržel Firefox i přes postupný pokles a drží se jen přibližně 3% pod IE. Safari a Opera si drží svoji pozici konstantně.

### 3.7.2 Test jednotlivých webových prohlížečů

Následující test porovnává nejpoužívanější webové prohlížeče podle statistiky v předchozí kapitole. Tento test vycházel z webové stránky „*html5test.com*“. Tato stránka porovnává podporu vlastností HTML5, ať se jedná o sémantiku, multimédia, 3D, grafiku, efekty a jiné.

Na základě počtu podporovaných prvků se určí hodnocení, který prohlížeč má nejlepší podporu v oblasti všech dosavadních vlastností HTML5.

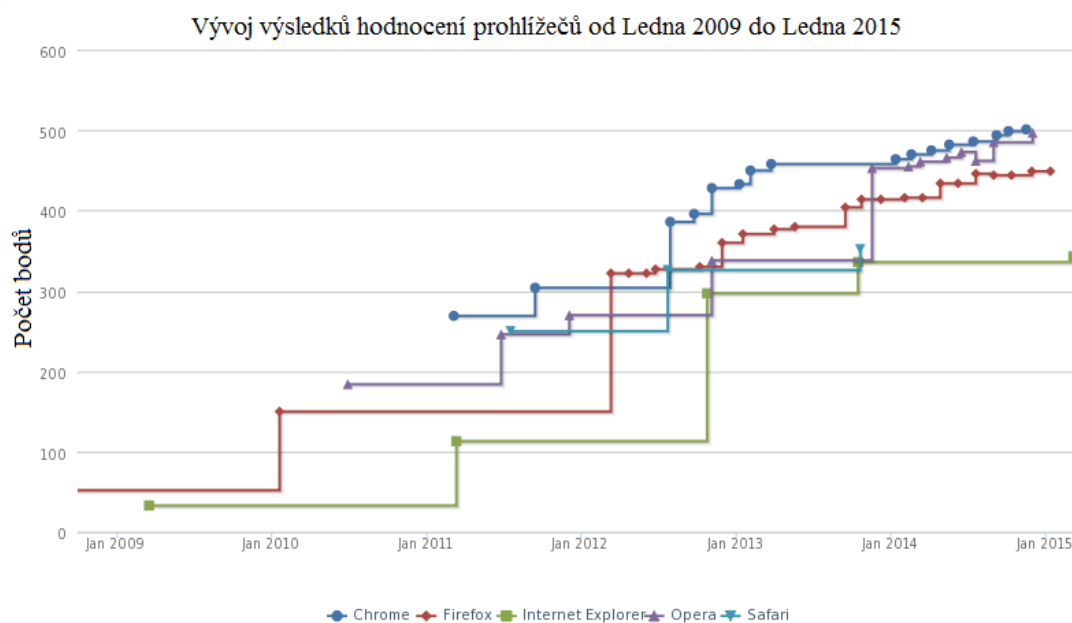


Obrázek 3 - Výsledek porovnání Top 5 prohlížečů

Přiložený obrázek (Obrázek. 3) zobrazuje konečné vyhodnocení všech pěti prohlížečů v aktuálně vydaných verzích. Největší možné skóre je 555 bodů. Podrobná tabulka s výsledky je přiložena v příloze k bakalářské práci.

S nejlepším hodnocením vyšel webový prohlížeč Chrome od společnosti Google. Dále je zřejmé, že podle tohoto testu se Internet Explorer umístil nejhůře ze všech porovnaných prohlížečů, tento výsledek by mohl vysvětlovat, proč Internet Explorer nemá takové zastoupení v dnešní době. Ovšem je to pouhá spekulace.

Pro příklad je přiložen graf, jak se postupně vyvíjelo hodnocení na základě vydávaných verzí webových prohlížečů za uplynulých let od ledna 2009 do ledna 2015.



Graf 2 - Vývoj bodů v rámci časového období

## 4 Praktická část

Předchozí teoretická část se věnovala shrnutí základních vlastností HTML a CSS a pojednávala o přínosech jejich nejnovějších specifikací. V této části se práce zaměřuje na praktické využití výše zmíněných specifikací, formou vytvoření demonstrativní webové prezentace, bez použití aplikací typu WYSIWYG<sup>20</sup>.

Webová stránka je navržena jako ukázka vybraných funkcí a vlastností HTML5 a CSS3. Je rozdělena do 6 stránek (záložek), kde každá obsahuje rozdílné vlastnosti.



Obrázek 4 - „Úvodní strana“

Výčet podstránek webové prezentace:

- „Úvodní strana“ s informativním textem o projektu,
- „Canvas“ element s jednoduchým kreslením,
- „Video“ s logem ČZU před přehráním samotného videa,
- „Články“ se dvěma tématy,
- „Formulář“ určený například pro registraci,
- „Geolokace“ pro určení oblasti výskytu.

<sup>20</sup> What you see is what you get

## 4.1 Zápis DOCTYPE

Každý HTML dokument musí mít na prvním řádku deklaraci typu dokumentu, neboli DOCTYPE. Značka DOCTYPE určuje, jak má být (podle jaké definice) HTML dokument zpracováván. Předchozí verze HTML musely obsahovat složitý zápis, kde byl odkaz na *.dtd* dokument ze serveru *www.w3.org*.

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

Nová specifikace HTML5 zavedla zjednodušení zápisu.

```
<!DOCTYPE html>
```

Výhoda nového zjednodušeného zápisu typu dokumentu je bezesporu zpětná kompatibilita se staršími prohlížeči, kromě IE6 a jeho starších verzí, které si s novou deklarací neporadí.

## 4.2 Využití strukturálních značek

Po stanovení deklarace typu dokumentu DOCTYPE následuje párový element `<html>`. Tento element může, ale i nemusí být v HTML dokumentu obsažen. Webové prohlížeče si dokáží poradit i bez tohoto elementu. Element `<html>` navíc nemá žádné atributy.

Obecná kostra ukázkové webové prezentace vypadá následovně

```
<!DOCTYPE html>
<html>
  <head>
    <title>...</title>
  </head>
  <body>...</body>
</html>
```

Struktura webové prezentace je rozdělena do pěti částí:

- Hlavička s logem stránky
- Navigační lišta s hlavním menu určená k pohybu na stránce
- Menu s odkazy na stránky W3School na levé části stránky
- Obsahová část stránky (kontejner pro jednotlivé ukázky nových vlastností specifikace HTML5 a CSS3)
- Patička webové stránky

Jednotlivé části struktury webové prezentace jsou popsány v následujících kapitolách.

### 4.2.1 Hlavička

Element `<head>` je hlavička dokumentu, obsahuje informace určené majoritně pro prohlížeč. Z důvodu, že je webová stránka primárně psaná v češtině, je vložen do hlavičky element `<meta>`. Tento element obsahuje atribut `charset` s hodnotou `utf-8` kvůli správnému kódování dokumentu.

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

Vložením hodnoty `utf-8` se zobrazí správně česká diakritika.

Následně hlavička obsahuje povinný údaj, který je definován jako element `<title>`. Tento element specifikuje titulek stránky.

```
<title>Bakalářská práce - Čáslavka Ondřej</title>
```

Poslední použitý element `<link>` připojí soubor `style.css`. Jedná se o CSS dokument, který je určený pro stylování HTML stránky.

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

Atribut `rel` vyjadřuje typ vztahu, poté atribut `type` slouží k určení typu přiloženého dokumentu a atribut `href` slouží jako odkaz, kde se soubor nachází.

Celá hlavička dokumentu tedy vypadá názorně.

```
<head>
<meta charset="utf-8">
  <title>Bakalářská práce - Čáslavka Ondřej</title>
  <link rel="stylesheet" type="text/css" href="style.css">
</head>
```

V rámci záložky „Canvas“ je hlavička dokumentu obohacen o odkaz na JavaScriptový soubor, který se nachází na serveru, kde jsou stránky uloženy.

Hlavička stránky „Canvas“ je definována následovně.



```

<head>
  <meta http-equiv="content-type" content="text/html; charset=UTF-8">
  <title>Bakalářská práce - Čáslavka Ondřej</title>
  <link rel="stylesheet" type="text/css" href="style.css">
  <script type="text/javascript" src="malovani.js"></script>
</head>

```

## 4.2.2 Tělo dokumentu

V elementu `<body>` se nacházejí jednotlivé části webové stránky, které jsou tvořeny pomocí nových strukturálních značek a odlišeny identifikátory. Přínos těchto nových strukturálních značek doplněný sémantikou umožňuje vyhledávacím robotům jednodušeji analyzovat části webové stránky. To má například za následek lepší výsledky při vyhledávání dotazu uživatele.

Rozdělení stránky vypadá následovně.

```

<header id="header" role="banner"> ... </header>
  <section id="all" role="main">
    <nav role="navigation" class="mainmenu"> ... </nav>
    <aside class="leftLinks"> ... </aside>
    <section id="content" role="main">
      <section class="right" role="main"> ... </section>
      <header> ... </header>
      <article class="clanekbig"> ... </article>
    </section>
    <aside id="thanks" role="complementary"> ... </aside>
  </section>
<footer role="contentinfo"> ... </footer>

```

Pozadí webové prezentace je deklarováno přímo v elementu `<body>`, které je vytvořeno pomocí CSS vlastnosti `background` s pomocí nové funkce `linear-gradient`. Tato funkce vytvoří barevný přechod, který se zobrazí na celé stránce.

Zápis vypadá následovně.

```

body {
background: linear-gradient(to right,#2E8EEE, white, #2E8EEE);
font-family:Myriad Pro;
}

```

Parametr této vlastnosti obsahuje 3 barvy a směr přechodu (výchozí nastavení přechodu je shora dolů). Funkce `font-family` nastaví primárně celému elementu `<body>` výchozí písmo `Myriad Pro`.

Výsledek funkce `linear-gradient`.



Obrázek 5 - Pozadí stránky

Aby nedošlo k problému vykreslování v rámci jednotlivých prohlížečů, jsou využity specifické zápisy pro konkrétní webové prohlížeče a jejich starší verze.

Apple Safari

```
background: -webkit-linear-gradient(to right,#2E8EEE, white, #2E8EEE);
```

Opera

```
background: -o-linear-gradient(to right,#2E8EEE, white, #2E8EEE);
```

Mozilla Firefox (do verze 15)

```
background: -moz-linear-gradient(to right,#2E8EEE, white, #2E8EEE);
```

**„header“**

Hlavička stránky neboli element `<header>` je na stránce formou obrázku, který je vložen za pomoci kaskádových stylů. Celková struktura tohoto elementu je v tomto případě pouze logo, které po kliknutí odkáže zpět na „Úvodní stránku“ webové prezentace.



Obrázek 6 - „header“ Logo

Definice v HTML kódu je následující.

```

<header id="header" role="banner">
    <a href="../BP2/projekt.html" title="Úvodní stránka" class="logo">
    </a>
</header>

```

Element využívá atributy `id` a `role`. Atribut `id` slouží jako identifikátor v kaskádových stylech. Kdyby se element `<header>` nacházel v HTML dokumentu vícekrát, tak bez identifikátoru by bylo nemožné upravovat každý zvlášť. Atribut `role` má specifickou funkci pro čtečky webu a ulehčuje orientaci v obsahu.

Za pomoci kaskádových stylů se elementu `<header>` nastaví vlastnosti pro zobrazování na stránce. Definice v CSS vypadá následovně.

```

#header {
    position: relative;
    margin-bottom: 0px;
}

```

Nastavení `position` s hodnotou `relative` v tomto příkladu určuje, aby se hlavička s logem překryla s navigační lištou tak, že pod hlavičkou bude vytvořený stín a nebude ho překrývat navigační lišta. Nastavení `margin-bottom` na `0px` změní vnější šířku okraje, tedy na nulovou hodnotu.

Obsahem elementu `<header>` je element `<a>`. Zápis vypadá následovně.

```

<a href="../BP2/projekt.html" title="Úvodní stránka" class="logo"> </a>

```

Element `<a>` obsahuje atribut `href` s hodnotou `../BP2/projekt.html`. Tato hodnota atributu po kliknutí na logo odkáže na úvodní stránku webové prezentace.

Element `a` má definovanou třídu `class` s hodnotou `logo`. Takto definovaná třída je v CSS zapsána takto.

```

#header .logo {
    background-image: url(logo-background-1.jpg);
    background-position: center; height:170px;
    display:block; box-shadow: #00004F 2px 2px 15px;
    border-bottom:2px solid #e44f27;
}

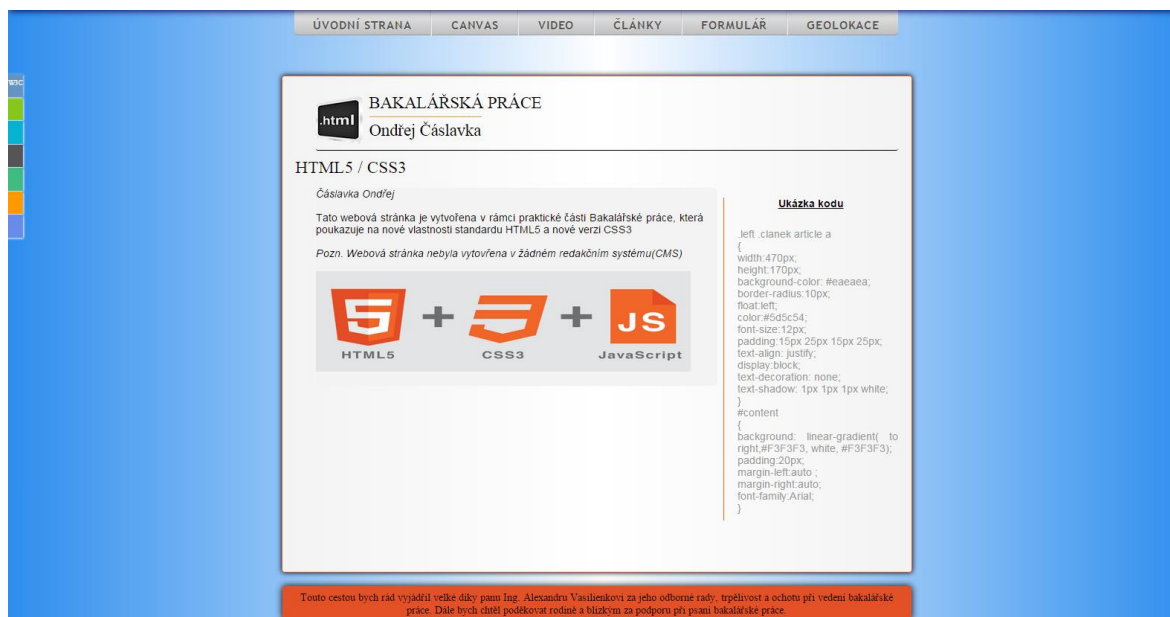
```

Obrázek je nahraný pomocí odkazu `url(logo-background-1.jpg)`, následně vycentrovaný pomocí vlastnosti `background-position:center`. Výška loga je určena parametrem

height:170px. Ohraničení spodní části je definováno border-bottom:2px solid #e44f27 a stíny jsou vytvořeny pomocí box-shadow: #00004F 2px 2px 15px.

## „section“

Webová prezentace obsahuje element <section> s identifikátorem all, ten slouží jako obal pro elementy, které se vyskytují mezi hlavičkou a patičkou stránky. Jedná se o navigační lištu navrchu stránky pod hlavičkou, středovou obsahovou část, odkazovou lištu na levé straně stránky a sekci s poděkováním nad patičkou stránky.



Obrázek 7 - Obsah "section" s identifikátorem „all“

Struktura zapsaná elementu <section> s identifikátorem all ve zdrojovém kódu stránky vypadá následovně.

```
<section id="all" role="main">
  <nav role="navigation" class="mainmenu"> ... </nav>
  <aside class="leftLinks"> ... </aside>
  <section id="content" role="main">
    <section class="right" role="main"> ... </section>
    <header> ... </header>
    <article class="clanekbig"> ... </article>
  </section>
  <aside id="thanks" role="complementary"> ... </aside>
</section>
```

Jednotlivé elementy, které jsou obsaženy v elementu <section> s identifikátorem all jsou popsány v jednotlivých kapitolách, které popisují funkce jednotlivých elementů.

## „footer“

Element `<footer>` označuje patičku své sekce. V případě článku obsahuje informace o autorovi, odkazy na zdroje či copyright<sup>21</sup>. Také ale může obsahovat patičku celé stránky s odkazy na sociální sítě, pravidla použitelnosti a jiné. Zpravidla se také používá kvůli lepší přehlednosti v kódu.



Česká zemědělská univerzita v Praze © 2014, Bakalářská práce Ondřej Čáslavka

Obrázek 8 - Obsah "footer"

Zápis ve zdrojovém kódu na stránce je definován takto.

```
<footer role="contentinfo">
    Česká zemědělská univerzita v Praze © 2014, Bakalářská práce Ondřej
    Čáslavka
</footer>
```

Element `<footer>` obsahuje pouze přidělenou roli s hodnotou atributu `contentinfo` která má význam pro čtečky webu. Konkrétně vrací informaci o copyrightu a zápatí stránky.

Dodatečné vlastnosti elementu `<footer>` jsou zapsány v CSS následovně.

```
footer {
    margin-bottom:50px;
    margin-top:12px;
    text-align: center;
}
```

Odsazení patičky od horní sekce je nastaveno na 12px pomocí funkce `margin-top`, která určuje vnější šířku okraje prvku. Spodní část prvku je odsazená o „50px“ pomocí stejné funkce, ale s označením `margin-bottom`. Text je zarovnán na střed přes funkci `text-align` s hodnotou `center`.

## 4.3 Navigační a odkazová lišta

„Navigační lišta“ slouží jako menu, pomocí kterého se lze pohybovat po stránce. Ohledně přehlednosti je navigační lišta vložena do vrchní části stránky.

Navigační lišta je v HTML kódu definována takto.

---

<sup>21</sup> Autorské právo

```

<nav role="navigation" class="mainmenu">
  <ul>
    <li> ... </li>
  </ul>
</nav>

```

Element `<nav>` označuje navigační menu stránky, role `navigation` slouží primárně pro čtečky webu. Pomocí třídy `mainmenu` je navigační lišta definována v CSS. Element `<nav>` dále obsahuje elementy `<ul>`, které se používají jako odrážkové seznamy a elementy `<li>` jako položky seznamu.

Položky v navigační liště jsou stylizovány přes CSS. Zápis vypadá následovně.

```

nav a {
float:left;
padding: 1.3em 1.6em .3em 1.6em;
text-decoration: none;
color: #555;
text-shadow: 0 1px 0 rgba(255,255,255,.5);
font: bold 1.1em/1 'trebuchet MS', Arial, Helvetica;
letter-spacing: 1px;
text-transform: uppercase;
border-width: 1px;
border-style: solid;
border-color: #fff #ccc #999 #eee;
background: linear-gradient(#f5f5f5, #c1c1c1);
}

```

Výsledek stylizace pomocí CSS.



Obrázek 9 - Navigační lišta

Po najetí kurzorem na lištu se automaticky přemění položka navigační lišty. Tato přeměna je vytvořena v CSS a to pomocí pseudotřídy `a:hover`. Díky této pseudotřídě lze definovat prvek tak, že po označení kurzorem změní vlastnost prvku podle nové definice. Příklad definice pseudotřídy `a:hover` navigační lišty vypadá takto.

```

nav a:hover {
outline: 0;
color: #fff;
text-shadow: 0 1px 0 rgba (0,0,0,.2);
background: linear-gradient (#e5a228, #f1bf26);
transition-duration: 0.4s;
}

```

Výsledek na webové stránce po najetí kurzorem je zobrazen následovně.



Obrázek 10 - Navigační lišta po najetí kurzorem

Celková doba přeměny je definována pomocí CSS funkce `transition-duration`. Tato funkce má parametr, který nadefinuje, za jaký celkový čas dojde postupně k přeměně.

„Odkazová lišta“ na stránce je tvořena z odkazů na webové stránky *W3School*. Celá lišta je vytvořena za pomoci nových vlastností kaskádových stylů. Tato lišta je specifická tím, že je na stránce zafixovaná a zůstává stále na stejném místě.

Definice v HTML vypadá následovně.

```

<aside class="leftLinks">
<ul>
<li> ... </li>
</ul>
</aside>

```

Element `<aside>` s třídou `leftLinks` označuje odkazovou lištu. Pomocí třídy `leftLinks` je lišta definována v CSS. Element `<aside>` dále obsahuje elementy `<ul>`, které se používají jako odrážkové seznamy a elementy `<li>` jako položky seznamu.

Lišta je rozlišena za pomoci barev. Každá z barev má po najetí kurzorem rozdílný obsah pole a odkazuje na jinou stránku. Efekty jsou podobně jako u navigační lišty vytvořeny pomocí CSS vlastností, ale tentokrát s rozdílnými parametry a funkcemi.

Definice v CSS obecně pro blok elementu s třídou `leftLinks` vypadá následovně.

```
.leftLinks {  
left:0px;  
float:left;  
display:block;  
}
```

Pro jednotlivé odrážky seznamu je definice v CSS stanovena takto.

```
.leftLinks ul {  
position: fixed;  
left: 0px;  
list-style: none;  
top: 265px;  
padding:1px;  
}
```

Funkce position s hodnotou fixed stanoví, že lišta je zafixována na stránce a atribut top určuje odsun menu od vrchu stránky o 265px.

Zobrazení a definování položky seznamu je nastaveno pomocí elementu <li> takto.

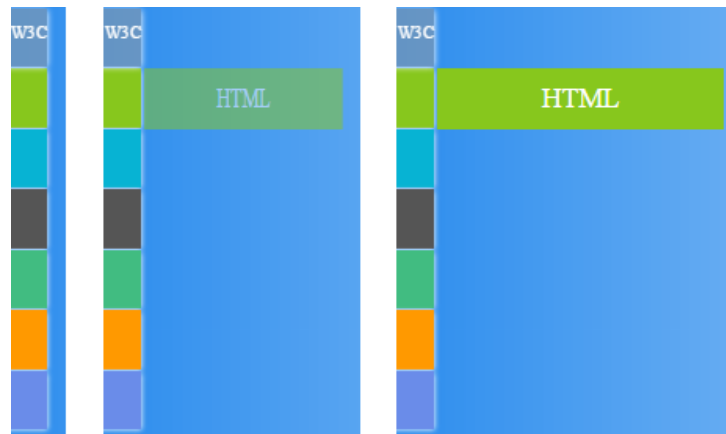
```
.leftLinks li a {  
display:block;  
height:18px;  
width:15px;  
background:rgba(155,155,155,0.5);  
color:#fff;  
padding:8px 6px;  
text-decoration:none;  
text-align:center;  
box-shadow: #FFF 1px 1px 4px;  
}
```

Dále každá položka seznamu má dodatečně zvlášť definovanou barvu, která se přiřazuje ke konkrétní položce. Jako příklad je uvedena položka odkazové lišty „HTML“.

```
.leftLinks li a[class='menu'] {  
box-shadow: #FFF 1px 1px 5px;  
font-size:10px;  
text-align:left;  
}
```

Odkazová lišta na stránce po najetí kurzoru na prvek vypadá názorně.





Obrázek 11 - Odkazová lišta

Efekt vysunutí položky je způsoben funkcí `transform` s hodnotou atributu `rotateY(0deg)`. Položka seznamu je nastavena na `rotateY(95deg)`, která po najetí kurzorem vytvoří vizuální efekt vysunutí. Přitom se jedná o posun na ose Y o 95 stupňů. Celý přechod je nastaven na délku 0.5s. Defínice v CSS vypadá následovně.

```
.leftLinks li span {
width:154px;
float:left;
text-align:center;
background:rgba(155,155,155,0.5);
color:#fff;
margin:-34px 29px;
padding:8px 6px;
opacity:0;
transform-origin:0%;
transform:rotateY(95deg);
transition: .5s;
animation: flip 2s;
}
```

## 4.4 Canvas

Záložka „Canvas“ ve webové prezentaci obsahuje ukázkou využití elementu `<canvas>`, který je nově přidán ve specifikaci HTML5. Element `<canvas>` představuje bitmapové kreslicí plátno závislé na rozlišení, které může být využíváno pro vykreslování grafů, herní grafiky a ostatních vizuálních prvků.

Element `<Canvas>` na ukázkové webové stránce je vykreslován pomocí *JavaScriptu*. Jedná se o jednoduché malování, kde jsou k dispozici nástroje přímka, obdélník a tužka. Jako pozadí při vykreslování je použito logo České zemědělské univerzity v Praze.

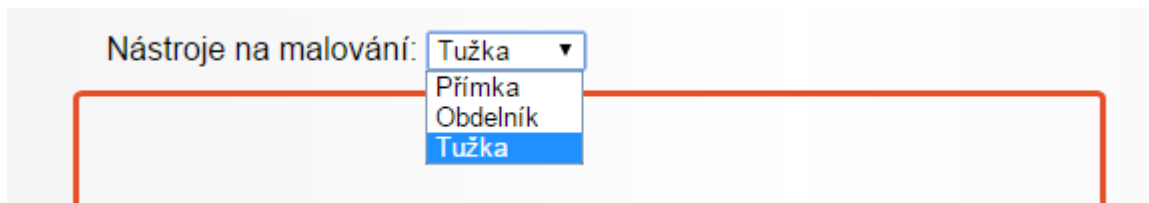
Připojení *JavaScriptového* souboru je provedeno pomocí elementu `<script>` v elementu `<head>`. Odkaz na soubor je připojen v hlavičce na stránce záložky „Canvas“.

```
<script type="text/javascript" src="malovani.js"></script>
```

Atribut `src` uvádí odkaz na místo uloženého souboru `malovani.js` na serveru, kde se nachází webová prezentace.

Pro výběr nástroje v malování je použit element `<label>`. Zdrojová kód vypadá následovně.

```
<label>Nástroje na malování:  
  <select id="dtool">  
    <option value="line">Přímka</option>  
    <option value="rect">Obdélník</option>  
    <option value="pencil">Tužka</option>  
  </select>  
</label>
```



Obrázek 12 - Výběr nástroje v malování

Princip funkčnosti propojení HTML kódu s *JavaScriptem*, je demonstrován na příkladu výběru nástroje „Tužka“.

Aby připojený *JavaScriptový* kód správně fungoval a dokázal si přiřadit správný nástroj pro malování, záleží na vybrané hodnotě atributu `value`. Hodnota `value` v tomto příkladu je `pencil`. *JavaScriptový* kód si pomocí této hodnoty přiřadí funkčnost naprogramované „Tužky“.

Část zápisu kódu pro výběr nástroje „Tužka“ zapsaný v *JavaScriptu* vypadá následovně.

```
Nástroj tužka je definován takto
tools.pencil = function () {
var tool = this;
this.started = false;
```

Samotný vykreslovací prostor je definován v elementu `<div>`, který obsahuje element `<canvas>` s identifikátorem `imageView`. Takto definovaný element si *JavaScript* nalezne pomocí následné funkce.

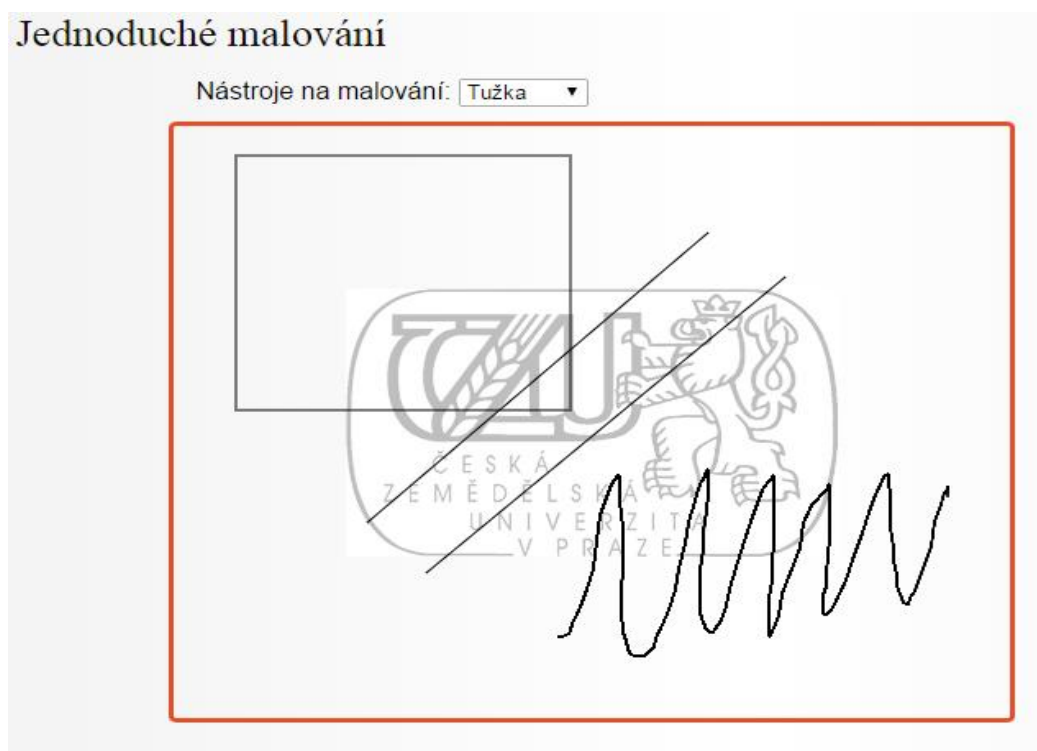
```
canvaso = document.getElementById('imageView');
```

Celý *JavaScriptový* kód s popisem funkcí je přiložen jako příloha k bakalářské práci

Rozměry kontejneru pro vykreslení jsou obsaženy jako atributy v elementu „*canvas*“ s parametry.

```
<canvas id="imageView" width="510" height="400">
```

Výsledek rozměru kreslicího pole vypadá takto.



Obrázek 13 - Jednoduché malování

V případě kdyby byl použit webová prohlížeč, který by nepodporoval nový element `<canvas>`, se zobrazí informativní text o nefunkčnosti aplikace a přidá hypertextové odkazy na stránky výrobců webových prohlížečů.

```
<p>
Bohužel, váš webový prohlížeč nepodporuje tuto aplikaci. Prosím použijte
jeden z níže podporovaných webových prohlížečů.
</p>
<p>Podporované prohlížeče:
    <a href="http://www.opera.com">Opera</a>,
    <a href="http://www.mozilla.com">Firefox</a>,
    <a href="http://www.apple.com/safari">Safari</a> a
    <a href="http://www.konqueror.org">Konqueror</a>
</p>
```

## 4.5 Video

Element `<video>` je nová funkce specifikace HTML5, nahrazuje starý a složitý element `<object>` a umožňuje jednoduché vložení videa přímo na webovou stránku. Jedná se o náhradu *FlashPlayeru*, který díky svým vlastnostem nebyl a není moc populární. Element `<video>` je vložen následovně.

```
<video width="510" height="400" controls="" poster="LOGO_CZU.jpg">
```

Element `<video>` obsahuje atributy `width` a `height`, které určují rozměry okna s videem. Atribut `controls` slouží k zobrazení lišty, pomocí které lze ovládat video (*play*, *stop*, *volume*). Dále atribut `poster` vkládá obrázek, který se zobrazuje místo videa před přehráním.

## Video s logem ČZU



Obrázek 14 - Video s logem CZU

Pomocí odkazu se načte video, které je uloženo na serveru s webovou prezentací. Odkaz je vložen přes atribut `src` s hodnotou atributu `What is HTML5.mp4`, jedná se o název videa, které je uloženo v kořenové složce na serveru.

Celá definice v HTML je zapsána následovně.

```
<video width="510" height="400" controls="" poster="LOGO_CZU.jpg">  
  <SOURCE src="What is HTML5.mp4" type="video/mp4">  
  Váš prohlížeč nepodporuje tento video tag.  
</video>
```

Atribut `type` informuje prohlížeč o typu video souboru. Text, který je vložen mezi elementem `<video>` se zobrazí tehdy, když prohlížeč nepodporuje nový element `<video>`.

## 4.6 Články

Záložka „Články“ slouží jako demonstrace vybraných vlastností nové specifikace kaskádových stylů v rámci případné stylizace textu v ohraničeném prvku. Před příchodem

CCS3 nebylo možné například vytvořit zakulacené rohy nebo přidat stín prvku. Nová specifikace přináší jednodušší a méně náročný způsob jak vytvářet jednoduché efekty jen za pomoci CSS3. Následný popis vysvětluje, jak vytvořit prvek, který je obsažen na ukázkové stránce v záložce „Články“.

Velikost prvku je definována pevnými rozměry pomocí funkce `width` a `height`.

```
.left .clanek article a {  
width:470px;  
height:170px;
```

Barva pozadí prvku je definováno přes funkci `background-color` a barva textu pomocí funkce `color`.

```
background-color: #EFEFEF;  
color: #5d5c54;
```

Efekt zakulacení rohů definuje funkce `border-radius`.

```
border-radius:6px;
```

Výsledek vypadá následovně.



Obrázek 15 – Článek

Pomocí pseudotřídy `a:hover` lze vytvořit efekt, který po přejetí kurzorem přidá prvku další vlastnosti. Přidáním funkce `box-shadow`, `border-color` a `color` lze jednoduše vytvořit interaktivní efekt. Zápis v CSS vypadá takto.

```

.clanek article a:hover {
box-shadow: black 1px 1px 10px;
}
.clanek article a:hover span {
border-color:#e44f27;
color:#e44f27;
}

```

Výsledný efekt přidá stín celému prvku a změní barvu textu názvu článku.



Obrázek 16 - Článek při interakci

Funkce `border-radius`, `box-shadow` a jim podobné se vyskytují v konceptu celé webové prezentace. Pomocí těchto funkcí je vytvořena většina prvků na stránce. Jedná se o efektivní způsob jak graficky upravit vzhled bez použití starých metod, tedy vkládání obrázků, které akorát prodlužovaly dobu načtení webové stránky.

## 4.7 Formulář

Záložka „Formulář“ demonstruje ukázkou vlastností webových formulářů, vytvořených za pomoci nově přidávaných atributů ve specifikaci HTML5.

Formulář ve webové prezentaci se nachází v elementu `<section>` s třídou `form`. Pomocí této třídy je v CSS provedena stylizace na stránce. Zápis vypadá následovně.

```

.form {
background:#f1f1f1;
width:440px;
margin-left:70px;
padding-left:20px;
padding-top:10px;
border-radius: 8px;
}

```

Pomocí tohoto zápisu se nastyluje prvek, ve kterém se nachází jednotlivé okna formuláře. Pomocí funkce `width` se stanoví šířka celého prvku. Odsazení od kraje stránky je posunuto o 70px funkcí `margin-lef` a vnitřní odsazení je nastaveno funkcemi `padding-left` a `padding-top`. Zakulacení rohy prvku jsou vytvořeny pomocí `border-radius`.

Obrázek 17 - Formulář

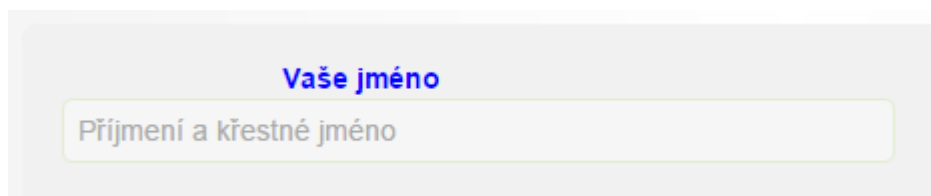
Struktura pole ve formuláři je definovaná přímo v HTML kódu a pomocí atributů jsou poté určeny vlastnosti pole. Formulář demonstruje ukázkové registrační okno, ve kterém je potřeba vyplnit povinné základní údaje k případné registraci.



Atribut `placeholder` umožňuje nastavit výchozí text v poli, který slouží jako nápověda nebo popisek. Po najetí a kliknutí kurzoru text zmizí a je možné do pole zapisovat.

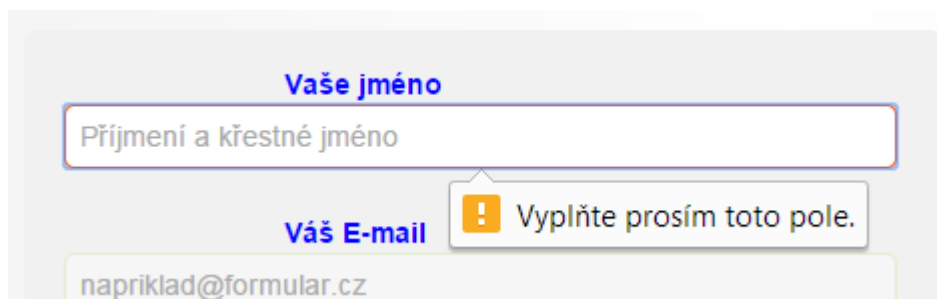
Zdrojový kód prvku vypadá následovně.

```
<p class="contact">
  <label for="name">Vaše jméno</label>
</p>
<input name="name" id="name" required="" type="text"
placeholder="Příjmení a křestné jméno" >
```

The image shows a web form element. At the top, there is a blue label "Vaše jméno". Below it is a text input field with a light gray border and a light gray background. Inside the field, the text "Příjmení a křestné jméno" is displayed in a light gray font, serving as a placeholder. The entire form element is set against a light gray background.

Obrázek 18 - Formulář - "placeholder"

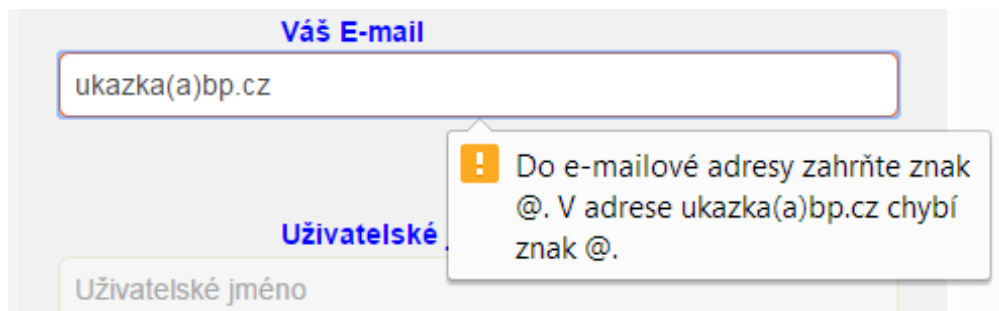
Další výhoda nové specifikace HTML5 je atribut `required`, který ověří naplnění povinných polí. Pokud pole při odeslání požadavku budou prázdné, požadavek se neodešle ke zpracování a vyskočí automatická hláška s informativním textem.

The image shows a web form with two fields. The first field is labeled "Vaše jméno" and contains the placeholder text "Příjmení a křestné jméno". The second field is labeled "Váš E-mail" and contains the text "napriklad@formular.cz". A red border highlights the email field, and a tooltip with a yellow warning icon and the text "Vyplňte prosím toto pole." points to the field, indicating a validation error because the field is empty.

Obrázek 19 - Formulář - "required"

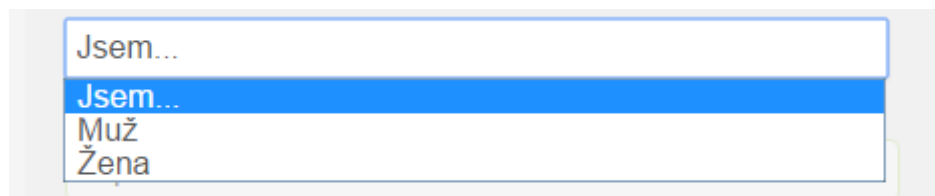
Každý webový prohlížeč má svou specifickou verzi vyskakovací hlášky, ale význam zůstává stejný.

Pomocí atributu `type` lze definovat druh pole. Příkladem je hodnota atributu `email`, kde text v poli musí obsahovat znak `@`. Pokud nebude pole korektně vyplněno s tímto znakem, nelze formulář odeslat ke zpracování a vyskočí informativní hláška.



Obrázek 20 - Formulář - "type"

Ukázkový registrační formulář dále nabízí výběr pohlaví. Tato část obsahuje rozbalovací pole s předem stanovenými hodnotami. Prvek je definován pomocí elementu `<select>`.



Obrázek 21 - Rozbalovací pole

Zdrojový kód je zapsán následovně.

```
<select name="gender" class="select-style gender">
  <option value="select">Jsem...</option>
  <option value="m">Muž</option>
  <option value="f">Žena</option>
</select>
```

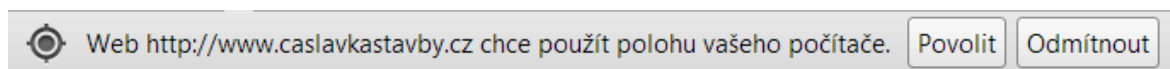
Element `<select>` obaluje tři elementy `<option>`, které jsou obsahey rozbalovacího menu. Atribut `value` je řetězec odesílaný jako hodnota pole. Mezi elementem `<option>` se poté nachází samotný text, který je vidět jako stanovená hodnota v menu.

## 4.8 Geolokace

Jedna z překvapivých novinek v HTML5 je netradiční geolokační rozhraní. Jedná se o rozhraní představující jednoduchou službu v rámci webových aplikací, pracující

s polohou uživatelů. Po svolení uživatelů lze jednorázově nebo opakovaně získat jejich aktualizovanou polohu pomocí dat z GPS<sup>22</sup>, mobilních sítí, Wi-Fi a podobně.

Aby aplikace mohla zjistit aktuální polohu uživatele, je zapotřebí, aby uživatel potvrdil odeslání polohy a měl webový prohlížeč, který podporuje tuto funkci.



Obrázek 22 - Povolení Geolokace

Lokalizace probíhá pomocí různých zdrojů informací o poloze. Každé zařízení provádí lokalizaci pomocí dostupných zdrojů. V rámci lepší přesnosti se zdroje často kombinují. Informace o poloze provádí samotné zařízení, na kterém je spuštěn prohlížeč. Zobrazení mapy s pozicí je implementováno pomocí odkazu na JavaScriptový soubor, ve kterém je naprogramovaná samotná aplikace na určení polohy.

```
<script type="text/javascript"
src="http://maps.google.com/maps/api/js?sensor=false">
</script>
<article>
<p>Hledání vaší pozice: <span id="status">hledám...</span></p>
</article>
```

Element `<article>` obsahuje vložený element `<span>` s identifikátorem `status`. Po nalezení polohy přes *JavaScriptový* soubor se automaticky změní hodnota identifikátoru u elementu `<span>` na `Nalezen!` nebo `Neúspěšné` (záleží na výsledku hledání). Všechno je provedeno automaticky pomocí *JavaScriptu*.

Mapa s polohou je zobrazena a rozvržena dle předem definovaných rozměrů. Do zdrojového kódu stránky se vloží nový element `<section>`, který je vytvořen pomocí *JavaScriptu*. Pokud aplikace nebude schopná nalézt polohu uživatele, mapa se na stránce nezobrazí.

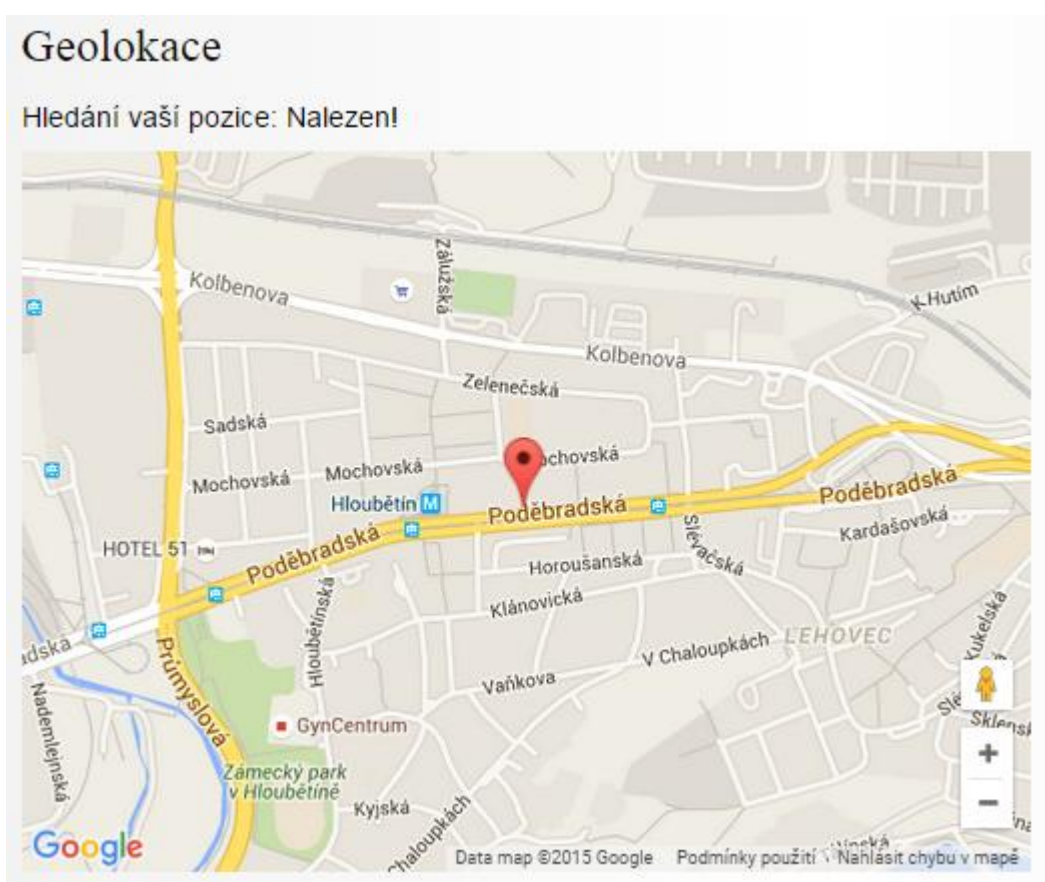
Hledání vaší pozice: Neúspěšné

Obrázek 23 - Neúspěšné nalezení polohy

---

<sup>22</sup> Global Positioning System

V případě, že aplikace nalezne aktuální polohu uživatele, na stránce se zobrazí mapa s pozicí uživatele.



Obrázek 24 - Úspěšné nalezení polohy

Obsah kódu celé *JavaScriptové* aplikace na určení pozice uživatele je obsahem přílohy k bakalářské práci.

## 5 Závěr

V dnešní době po dokončení dlouhého vývoje nových specifikací, by se dalo říci, že HTML5 a CSS3 jsou hodnotným a inovativním rozšířením předchozích verzí. Nabízejí řadu užitečných a jednoduchých novinek, které se dříve musely řešit složitějším způsobem zápisu.

Obecně celková kompaktnost a efektivnost zdrojového kódu je na vysoké úrovni a díky nově vydané specifikaci HTML5 a CSS3 jsou takto napsané stránky mnohem čitelnější jak pro vývojáře, tak i pro roboty internetových vyhledávačů nebo například pro čtečky obrazovky. Velké množství funkcí specifikací HTML5 a CSS3 je zpětně kompatibilní a to hlavně v případě kompaktnějšího zápisu určitých vlastností.

Dle výsledků testu podpory nových funkcí a vlastností ve webových prohlížečích, HTML5 a CSS3 prokázala relativně přijatelnou připravenost současných verzí prohlížečů. Dalo by se říci, že s každou novou verzí webového prohlížeče se zlepšuje podpora nových funkcí a vlastností.

Výsledkem převedení teorie do praxe, byla vytvořena ukázková webová prezentace založená na poznatcích z teoretické části. Ukázková stránka demonstruje vybrané funkce a vlastnosti specifikace, které jsou následně popsány a vysvětleny. Ukázková webová stránka tak následně představuje praktickou část bakalářské práce.

## Seznam zdrojů

- Adaptic.cz. 2012.** <http://www.adaptic.cz/znalosti/slovnicek/www/>. *www.adaptic.cz*. [Online] 10. Únor 2012. [Citace: 2. Listopad 2014.] <http://www.adaptic.cz/znalosti/slovnicek/www/>.
- Albers, Brian, Salim, Frank a Lubbers, Peter. 2011.** *HTML5 - Programujeme moderní webové aplikace*. místo neznámé : Computer Press, 2011. ISBN: 978-80-251-3539-6.
- Bordás, Robert. 2006.** <http://www.leancompany.cz/>. [Online] 2006. [Citace: 3. 7 2015.] <http://www.leancompany.cz/historie.html>.
- Coniseal. 2010.** *Počítačové znalosti*. [Online] 2010. <http://www.coniseal.com/historie-world-wide-web-nebo-internet/>.
- Elizabeth, Naramore a Yann Le, Michael K. 2006.** *PHP5, MySQL, Apache – Vytváříme webové aplikace*. Brno : autor neznámý, 2006. ISBN 80–251–1073–7.
- Etemad, Elika J. a Hickson, Ian. 2011.** <http://www.w3.org/>. [Online] W3C, 2011. [Citace: 20. 7 2015.] <http://www.w3.org/TR/css3-selectors/>.
- Garth. neuveđeno.** *Garth.cz*. [Online] neuveđeno. [Citace: 20. 6 2015.] <http://www.garth.cz/uvod-do-html/historie-html/>.
- Hogan, Brian P. 2011.** *HTML5 a CSS3* . místo neznámé : Computer press, 2011. brožovaná lepená.  
<http://www.garth.cz/uvod-do-html/historie-html/>.
- Churý, Lukáš. 2012.** *Webtea.cz*. [Online] 2012. [Citace: 20. 8 2015.] <http://programujte.com/clanek/2010082200-html5-nove-vlastnosti/>.
- Jahoda, Bohumil. 2014.** *jecas.cz*. [Online] 2014. [Citace: 19. 7 2015.] <http://jecas.cz/css-selektory>.
- Janovský a Dušan. 2014.** <http://www.jakpsatweb.cz>. *Jak psát web*. [Online] 2. 4 2014. <http://www.jakpsatweb.cz/odkazy-html.html>.
- Jikos. 2012.** <http://roger.jikos.cz/ssl-ssh.html#uvod3>. místo neznámé : roger.jikos.cz, 2012. <http://roger.jikos.cz/ssl-ssh.html#uvod3>.
- Jiří Kosek. 2012.** *HTML Guru*. [Online] 2012. [Citace: 10. 7 2015.] <http://htmlguru.cz/uvod-historie.html>.
- Margaret Rouse. 2006.** *SearchWinDevelopment*. [Online] 2006. [Citace: 18. 4 2015.] <http://searchwindevelopment.techtarget.com/definition/HTTP>.

**Reschke, J. F. a Lafon, Y. 2007.** Hypertext Transfer Protocol. [Online] W3, 2007. [Citace: 12. 6 2015.] <http://www.w3.org/Protocols/HTTP/1.1/rfc2616bis/draft-lafon-rfc2616bis-03.html>.

**Sklenák, Vilém. 2001.** *Data, informace, znalosti a Internet.* místo neznámé : Nakladatelství C H Beck, 2001, 2001. 8071794090, 9788071794097.

**Sládek, Honza. 2010.** *zdrojak.cz.* [Online] 10. 12 2010. [Citace: 23. 8 2015.] <https://www.zdrojak.cz/clanky/webdesigneruv-pruvodce-po-css3-selektory/>.

**Slavoj, Písek. 2014.** *HTML: začínáme programovat, 4.* místo neznámé : Grada Publishing, a.s, 2014. 8024750597, 9788024750590.

**Trejbal, Tomáš. 2014.** <http://www.htmlkody.info/HTML-5>. 2014.

## Seznam obrázků

Obrázek 1- Struktura pomocí <div> .....	16
Obrázek 2 - Struktura pomocí elementů HTML5 .....	16
Obrázek 3 - Výsledek porovnání Top 5 prohlížečů .....	36
Obrázek 4 - „Úvodní strana“ .....	38
Obrázek 5 - Pozadí stránky .....	42
Obrázek 6 - „header“ Logo .....	42
Obrázek 7 - Obsah "section" s identifikátorem „all“ .....	44
Obrázek 8 - Obsah "footer" .....	45
Obrázek 9 - Navigační lišta .....	46
Obrázek 10 - Navigační lišta po najetí kurzorem .....	47
Obrázek 11 - Odkazová lišta.....	49
Obrázek 12 - Výběr nástroje v malování .....	50
Obrázek 13 - Jednoduché malování.....	51
Obrázek 14 - Video s logem CZU .....	53
Obrázek 15 – Článek.....	54
Obrázek 16 - Článek při interakci .....	55
Obrázek 17 - Formulář.....	56
Obrázek 18 - Formulář - "placeholder" .....	57
Obrázek 19 - Formulář - "required" .....	57
Obrázek 20 - Formulář - "type" .....	58
Obrázek 21 - Rozbalovací pole.....	58
Obrázek 22 - Povolení Geolokace .....	59
Obrázek 23 - Neúspěšné nalezení polohy.....	59
Obrázek 24 - Úspěšné nalezení polohy.....	60

## Seznam Grafů

Graf 1 – Top 5 webových prohlížečů (od ledna 2011 do ledna 2015) .....	35
Graf 2 - Vývoj bodů v rámci časového období.....	37



## **Přílohy**

Webová prezentace vytvořená v „*Praktické části práce*“ je k dispozici online na adrese:

- <http://www.caslavkastavby.cz/BP2/projekt.html>

Dále je webová prezentace k dispozici na přiloženém CD.