

Učební pomůcka pro předmět webové technologie

Bc. Jana Fibichrová

Diplomová práce
2016

 Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky
akademický rok: 2015/2016

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Jana Fibichrová**
Osobní číslo: **A14558**
Studijní program: **N3902 Inženýrská informatika**
Studijní obor: **Učitelství informatiky pro střední školy**
Forma studia: **prezenční**

Téma práce: **Učební pomůcka pro předmět webové technologie**
Téma anglicky: **Lecture Notes for the Web-based Technology Course**

Zásady pro vypracování:

1. Seznamte se s webovými technologiemi (HTML 5, CSS, JQuery, Bootstrap) v aktuální verzi.
2. Zpracujte popis těchto technologií.
3. Zpracujte podklady pro výuku – teoretickou i praktickou.
4. Realizujte sadu vzorových příkladů.
5. Navrhněte sadu 60 testovacích úkolů včetně řešení.
6. Uveďte možné směry rozvoje učební pomůcky.

Rozsah diplomové práce:

Rozsah příloh:

Forma zpracování diplomové práce: **tištěná/elektronická**

Seznam odborné literatury:

1. **BROWN, Tiffany B, Kerry BUTTERS a Sandeep PANDA. HTML5 okamžitě: [ovládněte HTML5 za víkend]. 1. vyd. Brno: Computer Press, 2014, 256 s. ISBN 978-80-251-4296-7.**
2. **PÍSEK, Slavoj. HTML: začínáme programovat. 4., aktualiz. vyd. Praha: Grada, 2014, 181 s. ISBN 978-80-247-5059-0.**
3. **CASTRO, Elizabeth a Bruce HYSLOP. HTML5 a CSS3: názorný průvodce tvorbou WWW stránek. 1. vyd. Brno: Computer Press, 2012, 439 s. ISBN 978-80-251-3733-8.**
4. **HOGAN, Brian P. HTML5 a CSS3: výukový kurz webového vývojáře. Vyd. 1. Brno: Computer Press, 2011, 272 s. ISBN 978-80-251-3576-1.**
5. **LUBBERS, Peter, Brian ALBERS a Frank SALIM. HTML5: programujeme moderní webové aplikace. Vyd. 1. Brno: Computer Press, 2011, 304 s. ISBN 978-80-251-3539-6.**
6. **LAZARIS, Louis. CSS okamžitě. 1. vyd. Brno: Computer Press, 2014, 168 s. ISBN 978-80-251-4176-2.**
7. **DOMES, Martin. 333 tipů a triků pro CSS. 2., aktualiz. vyd. Brno: Computer Press, 2011, 272 s. ISBN 978-80-251-3366-8.**

Vedoucí diplomové práce:

doc. Ing. Zdenka Prokopová, CSc.

Ústav počítačových a komunikačních systémů

Datum zadání diplomové práce:

5. února 2016

Termín odevzdání diplomové práce:

20. května 2016

Ve Zlíně dne 5. února 2016



doc. Mgr. Milan Adámek, Ph.D.
děkan



doc. Mgr. Roman Jašek, Ph.D.
ředitel ústavu

Jméno, příjmení: Bc. Jana Fibichrová

Název bakalářské/diplomové práce: Učební pomůcka pro předmět webové technologie

Prohlašuji, že

- beru na vědomí, že odevzdáním diplomové/bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že diplomová/bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk diplomové/bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce;
- byl/a jsem seznámen/a s tím, že na moji diplomovou/bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – diplomovou/bakalářskou práci nebo poskytnout licenci k jejímu využití jen připouští-li tak licenční smlouva uzavřená mezi mnou a Univerzitou Tomáše Bati ve Zlíně s tím, že vyrovnání případného přiměřeného příspěvku na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše) bude rovněž předmětem této licenční smlouvy;
- beru na vědomí, že pokud bylo k vypracování diplomové/bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky diplomové/bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem diplomové/bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji,

- že jsem na diplomové/bakalářské práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně, dne 20. 5. 2016


.....
podpis diplomanta

ABSTRAKT

Cílem práce je vytvoření učební pomůcky pro předmět webové technologie, která byla zpracována v programu PSPad. Teoretická část je zaměřena na vybrané programy, které umožňují zápis různých jazyků. Následuje popis základní struktury HTML5 dokumentu a jeho náležitosti. Každý záznam vytvořený v HTML dokumentu je potřeba objasnit za použití kaskádových stylů. Závěr teoretické části je věnována struktuře Bootstrap. Praktická část je zaměřena hlavně na praktické znalosti využití HTML, CSS a Bootstrap. Tyto jazyky nám pomáhají k zobrazení názorného popisu a vysvětlení kódu. Výsledný zápis je vyobrazen přes webový prohlížeč. Součástí práce jsou také prezentace pro výuku se vzorovými příklady.

Klíčová slova: Jazyk, Kaskádové styly, PSPad, Webová stránka, Značky, Příkazy.

ABSTRACT

The aim of this thesis is to create learning aids for the web-based technology which was developed in PSPad. The theoretical part is focused on selected programs that allow writing of different languages. The next part is a description of the basic structure HTML5 document and its elements. Each record created in the HTML document is needed to be clarified by the use of cascading style sheets. The ending of the theoretical part is devoted to the structure Bootstrap. The practical part is focused mainly on practical use of the knowledge of HTML, CSS and Bootstrap. These languages help us to show illustrative description and explanation of the code. The final entry is shown via a Web browser. This thesis also includes presentations for teaching with sample examples.

Keywords: Language, Cascading Style Sheets, PSPad, Website, Tags, Commands.

Tímto bych chtěla poděkovat doc. Ing. Zdence Prokopové, CSc., která mě vedla po celou dobu mojí diplomové práce. Vždy mi poskytovala cenné rady, připomínky a čas, který semnou strávila. A také bych chtěla poděkovat svojí rodině, která mi byla neustále oporou po dobu mého studia.

Prohlašuji, že odevzdaná verze bakalářské/diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.

OBSAH

ÚVOD.....	10
I TEORETICKÁ ČÁST.....	11
1 PROGRAMY PRO ZOBRAZENÍ HTML	12
1.1 PSPAD EDITOR	12
1.2 ADOBE DREAMWEAVER	13
1.3 NOTEPAD++	13
1.4 NETBEANS	14
1.5 APTANA STUDIO	15
2 HISTORIE HTML	17
2.1 HTML 5	18
2.2 PLÁNOVÁNÍ WEBOVÉ STRÁNKY	19
2.3 WEBOVÁ STRÁNKA	20
2.4 STRUKTURA HTML DOKUMENTU	21
2.5 ZNAKOVÁ SADA	22
3 HISTORIE CSS	24
3.1 CSS3.....	24
3.2 OBSAH HTML	24
3.3 ZADÁNÍ HODNOT V CSS	26
3.4 CSS SYNTAXE A SELECTOR	26
3.4.1 Identifikátor	28
3.4.2 Prvek třída	28
3.4.3 Seskupení prvků	29
3.4.4 Selector *	29
3.4.5 Selector atributu.....	30
3.4.6 Pseudotřída	30
3.5 VLOŽENÍ CSS DOKUMENTU	31
3.6 TESTOVACÍ VLASTNOSTI	32
3.7 CSS FRAMEWORK	33
4 BOOTSTRAP	36
4.1 VYUŽITÍ BOOTSTRAP.....	36
4.2 CO OBSAHUJE BOOTSTRAP?	36
4.3 POUŽITÍ GRID	37
4.3.1 Použití Grid systému	37
II PRAKTICKÁ ČÁST	39
5 VÝUKA WEBOVÉ TECHNOLOGIE.....	40
5.1 PROSTŘEDÍ PSPAD	40
5.2 VYTVOŘENÍ HTML DOKUMENTU	44
5.3 ZNAČKY V HTML DOKUMENTU.....	48
5.3.1 < HTML>	48
5.3.2 Hlavička dokumentu <head>.....	48

5.3.3	Titulek stránky <title>	48
5.3.4	Tělo dokumentu <body>	49
5.4	KOMENTÁŘ V HTML	50
6	FORMÁTOVÁNÍ TEXTU	52
6.1	NADPISY	52
6.2	ODSTAVCE	53
6.2.1	Zalomení řádku	53
6.2.2	Zarovnání dokumentu	54
6.2.3	Zvýraznění textu	55
6.3	SEZNAMY	57
6.4	OBRÁZEK	59
6.5	ODKAZY	61
6.6	TABULKY	62
6.7	FORMULÁŘE	64
6.8	AUDIO A VIDEO	67
6.9	GEOLOKAČNÍ ROZHŘANÍ	71
6.9.1	Zeměpisná šířka a délka	71
6.9.2	Informace o poloze	72
6.9.3	Ověření podpory pro geolokaci	72
6.9.4	Získání polohy	73
6.9.5	Ošetření chyb	74
6.9.6	Aktualizovaná poloha	75
6.10	MICRODATA	76
6.10.1	Funkce	77
6.10.2	Schéma.org	77
6.10.3	Kontrola microdat	79
7	KASKÁDOVÉ STYLY	80
7.1	KOMENTÁŘE V CSS	81
7.2	BARVA	81
7.3	POZADÍ	83
7.4	OHRANIČENÍ	85
7.5	OKRAJE	88
7.6	VÝPLŇ	89
7.7	VÝŠKA A ŠÍŘKA	89
7.8	OBTÉKÁNÍ	90
7.9	ZOBRAZENÍ	91
7.10	TEXT	92
7.11	FONT PÍSMO	93
7.12	SEZNAMY	94
7.13	STÍNY	95
7.14	SLOUPCE	96

8	BOOTSTRAP	98
8.1	ZÁKLADNÍ STRUKTURA BOOTSTRAP GRID	100
8.2	TYPOGRAFIE.....	101
8.3	TLAČÍTKA.....	103
8.4	FORMULÁŘE	104
8.5	POZADÍ.....	105
8.6	UMÍSTĚNÍ	105
8.7	OBRÁZKY	106
8.8	TABULKY	107
8.9	ROZBALOVACÍ NABÍDKA	108
9	PODKLADY PRO VÝUKU	109
10	SMĚRY ROZVOJE UČEBNÍ POMŮCKY.....	110
	ZÁVĚR	111
	SEZNAM POUŽITÉ LITERATURY.....	112
	SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK.....	118
	SEZNAM OBRÁZKŮ	120
	SEZNAM TABULEK.....	124

ÚVOD

Cílem mojí práce je vytvoření komplexního celku „Učební pomůcka pro předmět webové technologie“ na střední škole. Hlavní důraz je kladen na praktické ukázky jednotlivých částí.

Teoretická část je zaměřena na vybrané programy, které umožňují zápis různých jazyků. Z vybraných programů najdeme jeden, který je vytvořen českým tvůrcem. Mezi českými uživateli je velmi oblíbený pro svoji funkčnost, jednoduchost a hlavně volný ke stažení. Ostatní zvolené programy jsou celosvětově využívány pro tvorbu stránek.

Pro zrealizování kvalitních webových stránek je potřeba se pořádně zamyslet a vytvořit si plán a uspořádání stránek. Následuje popis základní struktury HTML5 dokumentu a jeho náležitosti. Jelikož jsou to stavební kameny pro tvorbu webových stránek. Potřebujeme znát dopodrobna strukturu, co všechno zahrnuje a jaký je význam jednotlivých částí.

Ke každému záznamu vytvořeného v HTML dokumentu bude potřeba objasnit použití kaskádových stylů. Pomocí zápisu kaskádových stylů můžeme webovou stránku upravit do lepšího vzhledu než obyčejná stránka s textem a obrázky. K tomu bude zapotřebí šablona (stejnopis), která se skládá z mnoha příkazů. Každým příkazem říkáme prohlížeči jakou vlastnost má přiřadit určitému prvku. Každá vlastnost je pro CSS dokument klíčová.

V neposlední řadě je popsána struktura Bootstrap. Jaké má uplatnění pro webové technologie a její efektivní rozložení pomocí HTML a kaskádových stylů. Pro vývojáře poskytuje jednoduché a čisté řešení pro budování rozhraní. Přizpůsobí se snadno a obsahuje krásné a funkční vestavěné komponenty.

Praktická část je zaměřena hlavně na praktické znalosti využití HTML, CSS a Bootstrap.

Každá z jednotlivých struktur je objasněna dopodrobna od jejího stažení k funkčnímu zápisu. Zobrazení názorného popisu a vysvětlení kódu. Výsledný zápis je vyobrazen přes webový prohlížeč.

I. TEORETICKÁ ČÁST

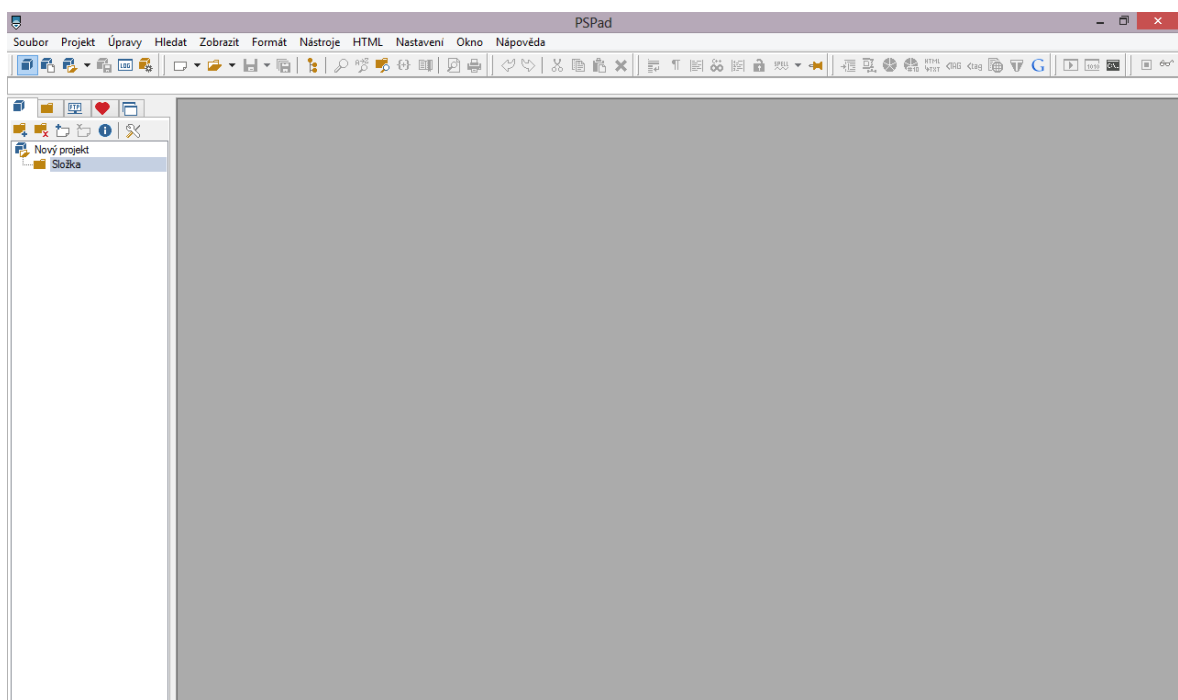
1 PROGRAMY PRO ZOBRAZENÍ HTML

Ke tvorbě a editaci webových stránek jsou zapotřebí HTML editory, které budou mít užitečné a funkční programy. Nástroje pro tvorbu a zjednodušení tvorby HTML a CSS stránek.

Pro tvorbu HTML stránek, je potřeba najít vhodný editor pro tvorbu. Provádět úpravy kódu lze pomocí množství nástrojů, aplikací nebo softwarovým balíčkem. Každý editor HTML má své výhody i nevýhody pro uživatelské používání.[1]

1.1 PSPad editor

PSPad editor [2] je určený programátorům nejrůznějších programovacích jazyků. Víceúčelový editor je původem z České republiky a je ke stažení zcela zdarma. Univerzální editor je určený pro MS Windows. Program nemá sice plnohodnotné vývojové prostředí (Obr.1), ale i přesto je mezi uživateli velmi oblíbený. Nabízí nespočetnou řadu funkcí, které jsou užitečné pro uživatele. Zaměřuje se na jednoduchost, funkčnost, přehlednost a rychlost. Umožňuje práci s více dokumenty najednou a porovnání obsahu souborů mezi sebou. Prostředí nabízí a podporuje přes 30 různých šablon (např. HTML5, CSS, PHP, Java, JavaScript, SQL, C++ apod.) Pro naši další práci budeme využívat právě tento program. [1] [2] [3] [4] [5]



Obrázek 1: Prostředí programu PSPad [vlastní zpracování]

1.2 Adobe Dreamweaver

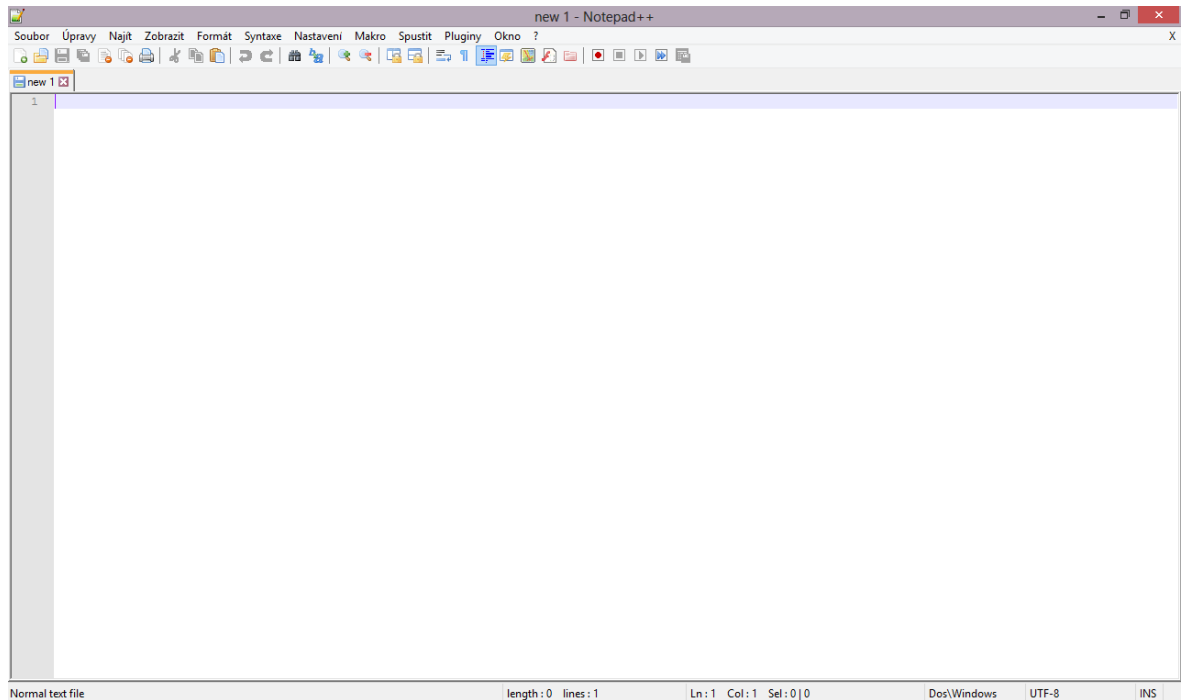
Program Adobe Dreamweaver [6] patří do produktové řady od společnosti Adobe. Celosvětový nejpobulárnější program pro tvorbu a úpravu profesionální webových stránek (Obr. 2). Určen je i pro začáteční uživatele návrhářům i vývojářům. Program nabízí vizuální rozvržení prostředí, tak zjednodušeného prostředí pro psaní přímého kódu. Poskytuje velké množství různých speciálních nástrojů a formátů, které usnadňují práci. Součástí balíčku je i integrovaná knihovna jQuery, která nabízí tvorbu pro operační systém Android a iOS. Program lze získat také v české verzi. Za roční předplatné si zaplatíme 290,17 Euro (kurz Eura 27,040Kč/1Euro, cca 7842Kč). [1] [6] [7] [8]



Obrázek 2: Vývojové prostředí Adobe Dreamweaver [1]

1.3 Notepad++

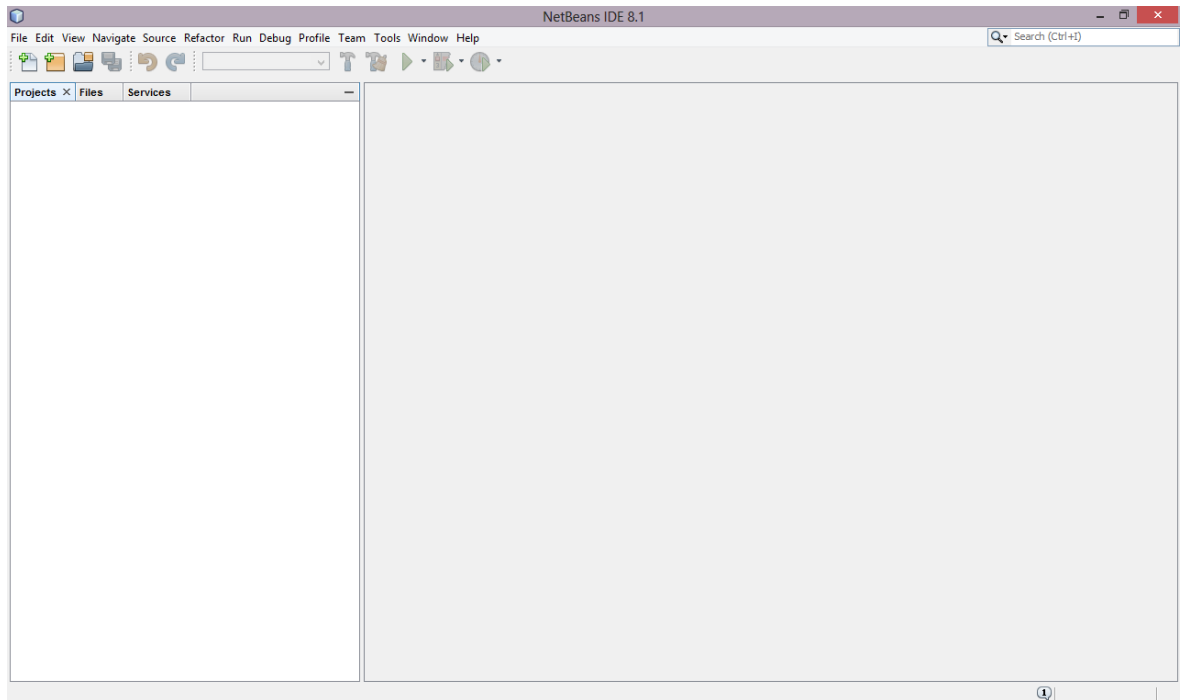
Notepad++ je vylepšená verze poznámkového bloku Windows. Nabízí mnoho podporovaných jazyků včetně HTML, C, C++, CSS a mnoho dalších. Jedná se jednoduchý a přehledný editor (Obr. 3), který umožňuje otevření více dokumentů současně, zvýrazňování syntaxe nebo práce s makry. V nabídce je také čeština a je zcela zdarma ke stažení. [1] [9] [10]



Obrázek 3: Vývojové prostředí Notepad++[vlastní zpracování]

1.4 NetBeans

Vývojové prostředí NetBeans využívají převážně pokročilejší uživatelé. Editor (Obr. 4) je speciálně zaměřen na programovací jazyk Java, ale pomocí pluginů ho lze rozšířit o jiné programovací jazyky. Podpora zahrnuje také pro SQL, CSS, JavaScript, XHTML apod. NetBeans disponuje mnoha vlastnostmi, které usnadňují práci. Obsahuje nástroje pro dynamickou tvorbu webu, práci v panelech, široké možnosti nastavení, napovídání funkcí, metod a další funkce. Editor lze zcela zdarma stáhnout, ale nepodporuje češtinu. [1] [11]

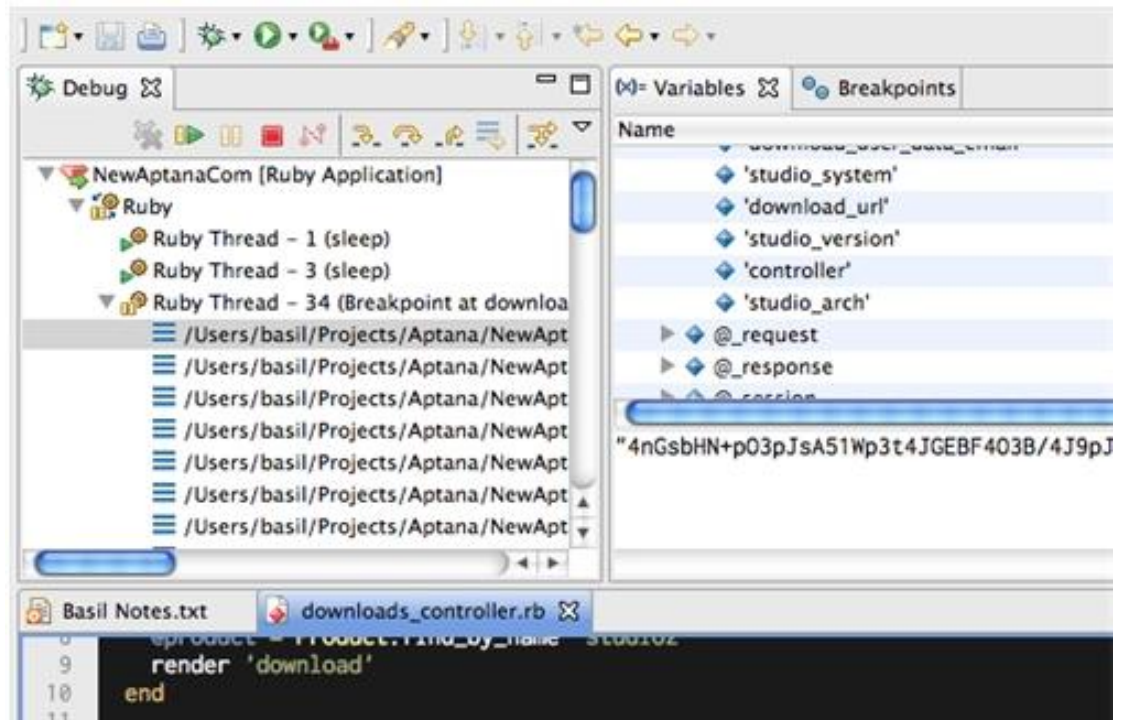


Obrázek 4: Vývojové prostředí NetBeans[vlastní zpracování]

1.5 Aptana studio

Aptana studio je mezi českými uživateli méně známý, ale celosvětově jde o velmi populární program, využívaný pokročilými amatéry, ale i profesionály (Obr. 5). Aplikace je určená především pro tvůrce webů a kodérům HTML5, CSS3, PHP, Ruby, Rails a Python. V rámci tohoto prostředí lze vytvořit kompletní web. Zahrnuje informace o podpoře pro každý prvek webového prohlížeče. Program lze bezplatně získat stažením[12].

[1] [12] [13]



Obrázek 5: Vývojové prostředí Aptana[1]

2 HISTORIE HTML

Historie HTML jazyka odstartovala ve 40. letech 20. století, kdy začínala vývojem hypertextů, tzn. odkazů ve webových stránkách. Vannevar Bush jako první představil Memex, hypertextový systém. Ted Nelson a Douglas Engelbartn věnovali veškerý svůj život ke globálnímu hypertextovému systému Xanada. Populární HyperCardem v roce 1987 se zasloužil Bill Atkinson. [14]

První počátky Html jazyka vytvořil Tim Berners-Lee v roce 1991, jako součást projektu WWW (World Wide Web). Zkratka HTML znázorňuje **H**yper **T**ext **M**arkup **L**anguage. Umožňoval vědcům, kteří se zabývají fyzikou vysokých energií, sdílení výsledků výzkumu po celém světě. Na švýcarsko-francouzských hranicích nedaleko Ženevy vznikl celý projekt s názvem CERNu (Centre Européenne pour la Recherche Nucléaire, Evropské centrum jaderného výzkumu). Součástí tohoto dvouletého projektu, mělo se vyřešit problémy se sdílením informací pro velkou instituci, jakou byla CERN. HTML Tagy umožňovala rozčlenit text do několika logických úrovní, kde bylo možné použít zvýraznění textu a zařadit ho do odkazu a obrázky. [14] [15] [16]

Návrh jazyka, který vytvořil Bernars-Lee nepředpokládal, že by někdo musel znát. Pro firmu NextStep byla napsána první verze www softwaru, která obsahovala jak prohlížeč, tak i integrovaný editor webových stránek. V roce 1991 byl veřejně uvolněn software včetně specifikací jazyka HTML 1.0. Marc Anderssen psal dříve se svými kolegy populární prohlížeč Mosaic, považoval za příliš obtížné aplikovat do programu editor HTML. Kvůli tomuto rozhodnutí, ne každý provozuje na svém počítači NextStep. Pro autory profesionálních stránek je v dnešní době důležité, aby znaly jazyk HTML.

Začali se zvyšovat požadavky uživatelů na WWW, a proto se různí producenti snažili obohacovat o nové HTML prvky. Prohlížeče neměli zatím žádnou grafiku a byli jen řádkové. Standard návrhu HTML 2.0 zahrnuje dvě úrovně spolu s grafickým internetovým prohlížečem, které se poprvé objevují v roce 1993. První úroveň (Level 1) rozšiřuje předchozí verzi. Druhá úroveň (Level 2) definuje práci s formuláři. Webový prohlížeč Mosaicu byl následníkem nového grafického prohlížeče Netscape, který byl představen veřejnosti o rok později. [14] [15]

V roce 1995 bylo zavedeno neoficiální rozšíření verze 2.0, známé jako HTML 3.0. Zahrnovalo rozšíření o vytváření tabulek a matematických vzorců. Objevily se prvky, které umožňovali lépe kontrolovat výsledný vzhled textu. Obtékání obrázků textem a styly dokumentu šlo mnohem lépe. Prohlížeč Mosaic a Netscape podporovaly některé prvky,

jako např. tabulky. Pro kompletní podporu se nabízel experimentální prohlížeč s názvem Arena, který se příliš nerozšířil. [14] [15]

V roce 1996 mělo Html 3.0 tak velký skok vpřed, že nikdo nezachytil a nedokázal zavést prohlížeč s jeho podporou. Přední softwarové firmy W3C (WorldWide Web Consortium) koordinovali vývoj standardů webu. Shodli se na vlastnostech, o které rozšíří Html 2.0. Vzniklo Html 3.2, které zdaleka neobsahovalo vše z verze 3.0, ale zůstaly jen ošetřené tabulky. Nejnovější prohlížeče podporovaly ostatní nové prvky, opakoval se stejný postup jazyka a sjednotil se na průnik nejrozšířenějších prohlížečů. [14] [15]

V červenci 1997 vyšla další verze 4.0, kdy tato verze byla vylepšena o rozšíření tabulek, skriptování, formuláře, doplněná o rámy a kaskádové styly. Od této doby byl samotný vývoj jazyka na dlouhou dobu zastaven. V dalších dvou letech byly sice vydány nové standardy, které jen opravovaly drobné chyby z předchozí verze. Zanedlouho po vydání verze HTML 4.0 zveřejnila firma W3C ještě jeden standard s názvem XML (Extensive Markup Language), který byl mnohem složitější. Standard XML ovlivnilo další vývoj jazyka, kdy vzniklo XHTML 1.0 odvozenou syntaxi od XML v roce 2000. V podstatě význam zůstal stejný jako u předchozích dostupných značek, došlo jen na malé kosmetické změny, kdy se drobně změnila syntaxe. [15] [17]

2.1 HTML 5

V roce 2004 mladá skupina lidí chtěla povýšit dosavadní webovou platformu na novou úroveň. Založili skupinu WHATWG (Web Sypertext Application Working Group) a vytvořili novou specifikaci a názvem HTML5. Zaměřili se na oblasti, které podle nich nejvíce strádali. Rozhodli se zaměřit a vypracovat nové funkce cílené na webové aplikace. Objevil se nový termín Web 2.0 přibližně v té samé době. Pro mnohé to skutečně bylo, jakoby šlo o druhý a nový web. Statické weby museli uvolnit místo těm více dynamickým a sociálním webům, které vyžadovali mnohem více funkcí. Společnost W3C se opět zapojila do vývoje HTML v roce 2006. První pracovní návrh HTML5 byl zveřejněn v roce 2008 a o rok později pak XHTML 2. Tvůrci prohlížečů řeší praktické problémy, kdy implementují nové funkce. Jejich zpětnou vazbou bylo experimentování prohlížečů a tak mohli vylepšovat specifikaci webové platformy. [17] [18]

Během roku se stále pracovalo na specifikaci pro verzi HTML 5.0. Koncem roku 2014 byla schválena konečná podoba. Přidalo se mnoho nových novinek. Mezi nejdůležitější bylo přidání několika nových elementů a zjednodušení zápisu. Další novinkou by měla být

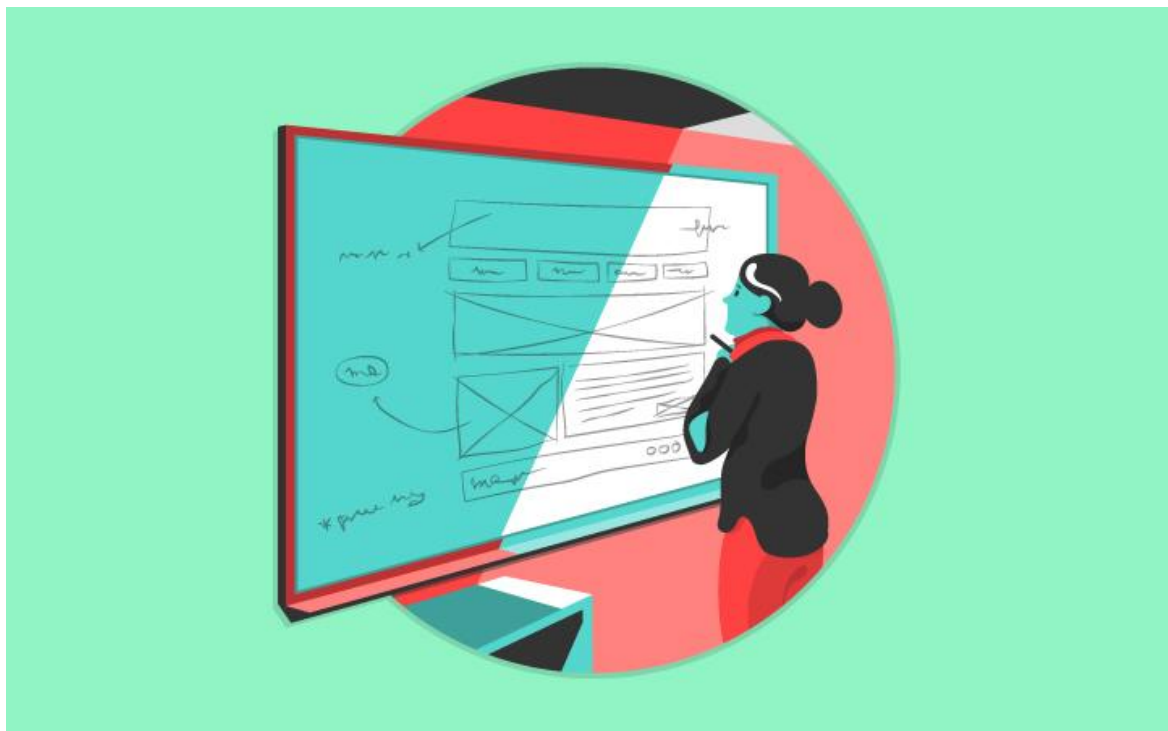
možnost přehrávat multimédia přímo ve webovém prohlížeči. Nejčastěji používané prohlížeče počítají s podporou některých novinek v HTML 5.0. [17] [18]

2.2 Plánování webové stránky

Pro zrealizování dobré stránky je potřeba se pořádně zamyslet. Naplánovat co všechno budeme chtít zobrazit na webových stránkách. Plánování někdy může zabrat i více času, ale dobrý plán zvýší použitelnost a urychlí následné uspořádání stránky. Hlavně uvidíme nakreslenou vizualizaci reálně před sebou. [19] [20]

Postup plánování:

1. Zamyslíme se, proč vytváříme webové stránky? Co všechno chceme vytvořením dosáhnout?
2. Pro jaké návštěvníky je ta webová stránka určena? Jak jim přizpůsobíme funkčnost a obsah stránky?
3. Kolik stránek bude potřeba pro realizaci? Jaké bude uspořádání jednotlivých stránek a textu? V jakém pořadí budeme chtít, aby byly jednotlivé stránky uspořádány?
4. Návrh naší webové stránky si nakreslíme na papír nebo na magnetickou tabuli, kde je možnost úprav mazáním.
5. Pojmenujeme si jednotlivé stránky výstižně a jednoduše. [19] [20]



Obrázek 6: Ukázka návrhu HTML stránky[19]

2.3 Webová stránka

Webové stránky se stávají pro mnohé složitější, ale základní struktura zůstává pozoruhodně jednoduchá. Než začneme, musíme si uvědomit, že bez kódu jazyka HTML není možné vytvořit webovou stránku. Kód jazyka HTML popisuje jeho význam a také uchovává obsah stránky. [19]

Webová stránka skládá se ze tří částí:

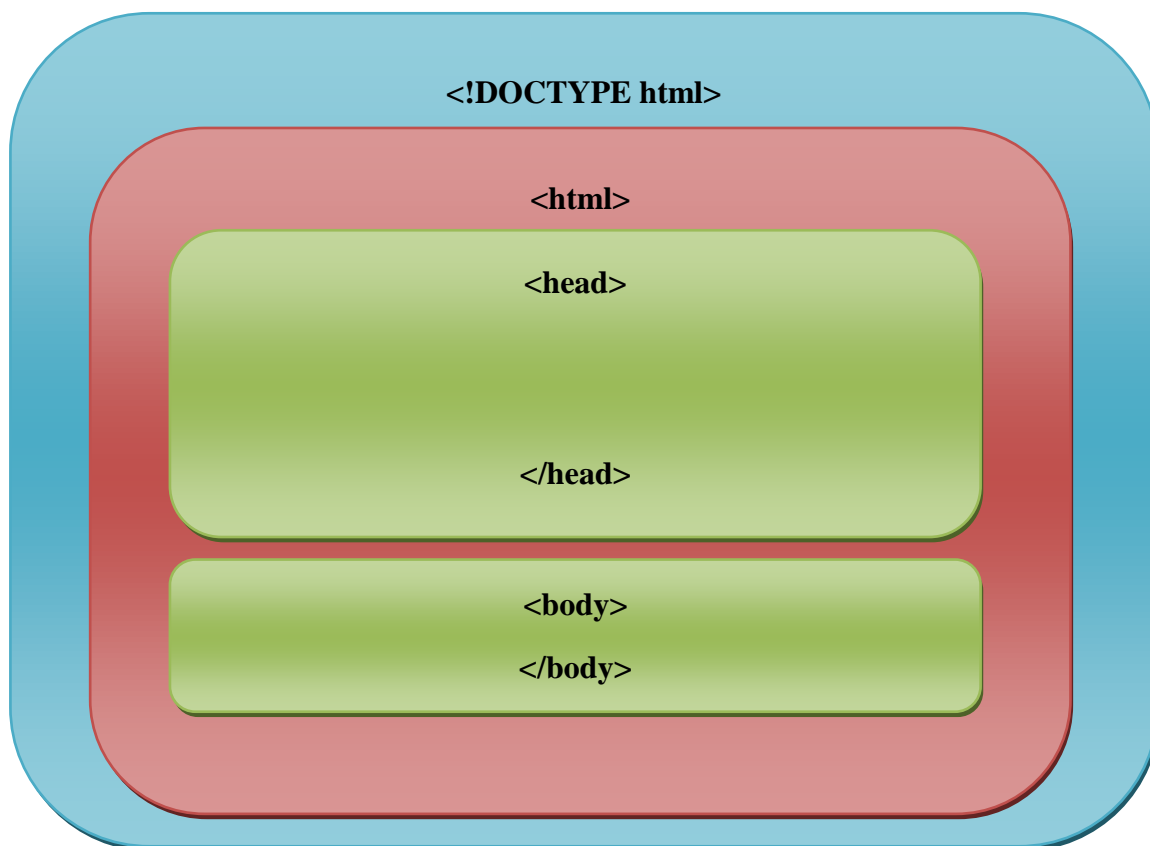
- **Textový obsah:** zobrazovaný prostý text na stránce, abychom informovali ostatní účastníky, na co se naše stránka zaměřuje (např. Rodinná dovolená, produkty)
- **Odkazy na jiné soubory:** slouží na odkazování na jiné stránky HTML, a stejně k nim patří také šablony stylů (ovlivnění vzhledu stránky) a soubory s kódem JavaScript jazyka (ovlivnění chování stránky), také slouží k nahrávání zvukových, obrázkových videosouborů.
- **Značky:** podepisujeme s nimi náš textový obsah a definujeme odkazy. Např. Písmeno M v češtině znamená **značkování**, ale v jazyku HTML definuje, neboli zkracuje slovo **markup**. [19]

2.4 Struktura HTML dokumentu

Správná struktura Html5 dokumentu (Obr. 7) musí mít ucelenou strukturu, bez které by nebylo možné, aby prohlížeč dokumentu porozuměl (tzv. musí ho správně zobrazit). Struktura je poněkud jednoduchá, skládá se ze základních značek (tagů). [22]

```
<!DOCTYPE html>           }   Typ dokumentu Html5  
  
<html lang='cs'>  
  
<head>  
<title> </title>  
<meta charset='utf-8'>  
<meta name='description' content=' '>  
<meta name='keywords' content=' '>  
<meta name='author' content=' '>  
<meta name='robots' content='all'>  
<!-- <meta http-equiv='X-UA-Compatible' content='IE=edge' -->  
<link href='/favicon.png' rel='shortcut icon' type='image/png'>  
</head>  
  
<body>                     }   Tělo  
  
</body>  
  
</html>
```

Hlavička



Obrázek 7: Grafická zobrazení HTML struktury

2.5 Znaková sada

HTML5 přináší nové použití znakové sady *DOCTYPE*, která přiměje prohlížeč k tomu, že bude přizpůsobovat stránku vyhovujícím standardům. Stránky můžeme zobrazovat v různých režimech, například nestandardní (tzv. quirk režim), téměř standardní a standardní režim (tzv. ne-quirk). Pro ověření dané stránky říká DOCTYPE prohlížeči, jaký režim a jaká pravidla se mají použít. Při použití některých ze starších prvků nebude výsledná stránka ověřena. [17] [21] [22]

Oproti dřívějšímu použití znakové sady bylo mnohdy nezapamatovatelné. Pro ukázkou bude použita sada z HTML4:

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

V novém HTML5 bylo velmi elegantně upraveno, tak aby bylo lépe zapamatovatelné:

`<!DOCTYPE html>`

V kořenovém prvku se také podstatně zjednodušilo na prostou informaci. Ukázka delšího zápisu kódového dokumentu oproti dřívějšímu zápisu:

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

Zjednodušení řádku

`<meta charset="UTF-8">` [17] [21] [22]

3 HISTORIE CSS

Oficiálně jazyk vznikl ve společnosti W3C koncem roku 1996. Mezi ostatní tvůrce nepřišla příliš velká podpora prohlížečů tím a bránila v jeho rozšíření. Až po vydání verze 4 jazyka HTML, nastalo využití pro CSS (Cascading Style Sheets). Tvůrci Html stránek umožňovali do nového deklarativního jazyka používat grafické prvky, které v tehdejším HTML nešly ani vytvořit. CSS1 můžeme definovat např. velikost textu, barvu textu, styl textu, rozestupy mezi jednotlivými prvky, barvu pozadí, a rozmístění prvků apod. [20] [23]

V roce 1998 byla vydána druhá verze jazyka (CSS2), která přinesla např. lepší absolutní a relativní pozice, první media typy a mnoho jiných vylepšení. CSS1 byla nahrazena po dvou letech od vzniku právě zmiňovanou CSS2. Zanedlouho i tato verze byla prohlášena za zastaralou, a proto W3C vydala CSS2.1, která řeší a opravuje některé nedostatky. [20] [23]

3.1 CSS3

CSS3 je rozšíření stávajícího jazyka CSS2.1, tak jak to bylo u HTML5. Jsou přidány nové definované prvky, které ulehčí práci vývojářům. Ve zmiňované nové verzi jsou opět řešeny nějaké poznatky problémů z předchozí verze a snaha o odstranění problémů. Nově jsou definovány např. více obrázků na pozadí, přechody, vlastnosti pro definici zaoblených rohů. [20] [23]

3.2 Obsah HTML

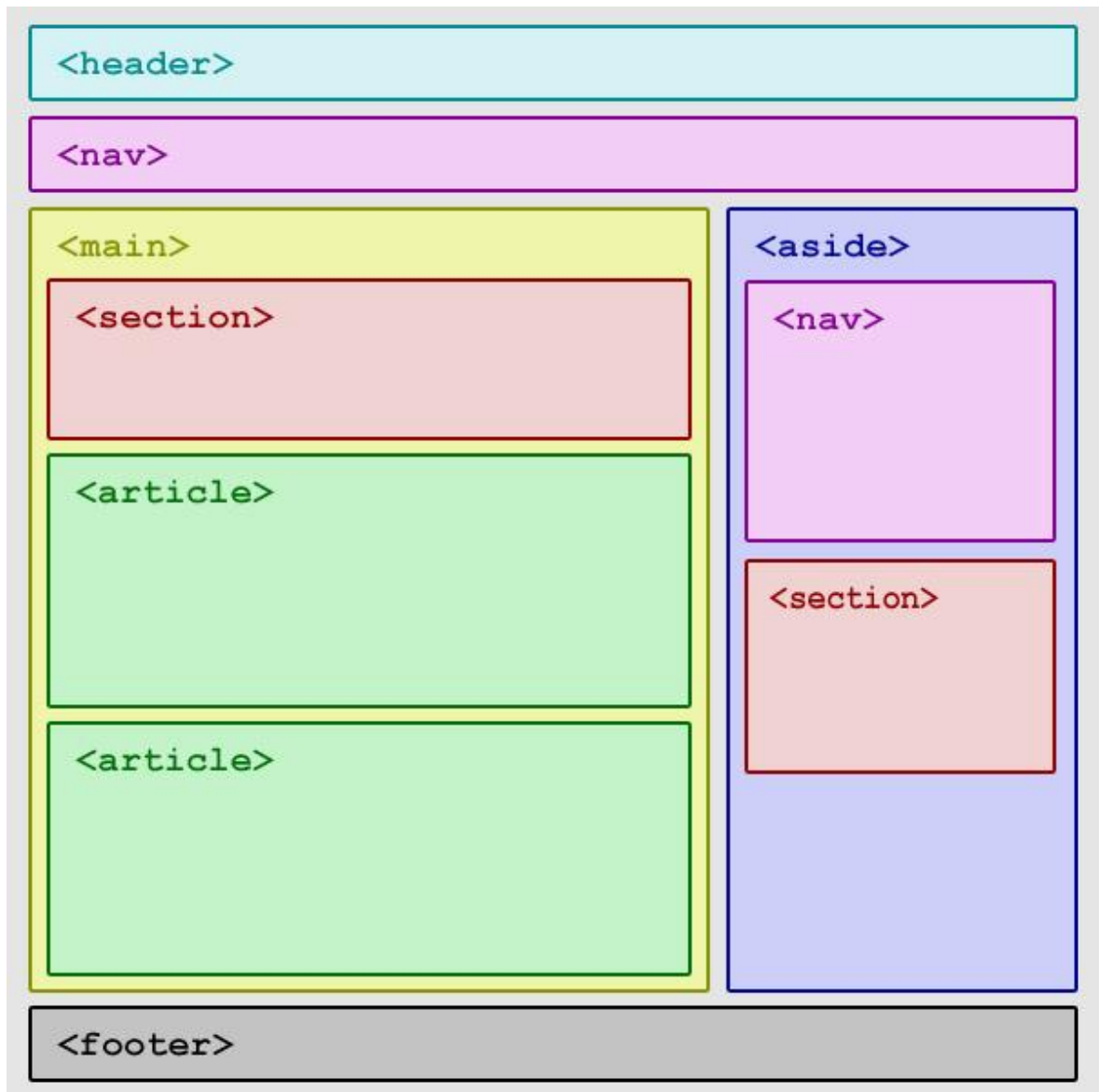
HTML5 zavedlo mnoho nových elementů, které jsou z typů obsahu, například sekce. Obsah elementů definují a popisují nové sémantické značky. Použití sémantického značení má své výhody:

- zkrátit a zjednodušit strukturu
- zlepšit vzhled
- začlenění audio a video [17] [24]

Na rozdělení obsahu stránky je možné aplikovat kaskádové styly. Snaží se o oddělení obsahu a formy, proto je nutné k aplikaci stylů používat jazyk CSS. Rozdělení obsahu stránky (Obr. 8):

- **<header>** - Obsah záhlaví
- **<nav>** - Navigace

- `<main>` - Obsah stránky
- `<section>` - Sekce webové stránky
- `<article>` - Textový obsah článku
- `<aside>` - Postranní nebo navigační panel
- `<footer>` - Zápatí stránky [17] [24]



Obrázek 8: Obsah HTML stránky [24]

3.3 Zadání hodnot v CSS

Webové stránky přiřadí vybraným místům vlastnosti selektorem tzv. kaskádové styly. Samotné vlastnosti by bez hodnot neznamenalý vůbec nic. Podíváme se, jaké hodnoty máme k dispozici:

- *Klíčová slova* – CSS si rezervují slova pro dané vlastnosti, např. *italic* (kurziva).
- *Reálná a celá čísla* – vlastnost číselné hodnoty může nabývat zapisováním arabských číslic (např. 1, 2, 3 apod.). V číselné hodnotě se může vyskytnout desetinná čárka, ale v CSS zapisujeme pomocí tečky (např. 2.5).
- *Jednotky délky* – Jedná se o obrazové body nebo anglicky pixely, kterými je základní jednotka délky v rámci CSS lze použít. Jednotky
- *Procenta* – Skládá se z čísla se znakem procent. Mezi hodnotou v procentech neděláme mezeru (např. 50%).
- *Odkaz URL* – hodnotu v CSS je poněkud odlišné a to na základě praktického zápisu například: `url(obrazky/obrazek.jpg)`.
- *Barvy* – K zápisu barev nejčastěji využíváme dva druhy zápisů:
 - Klíčová slova – psaná forma zápisu např. `blue`
 - Hexadecimální zápis v RGB – barva pro žlutou vypadá následovně `#FFFF00`, dbáme na to, aby se nevyskytovala mezera mezi znakem mřížky. [25]

3.4 CSS Syntaxe a Selector

Pomocí zápisu kaskádových stylů můžeme webovou stránku upravit do lepšího vzhledu než obyčejná stránka s textem a obrázky. K tomu bude zapotřebí šablona (stejnopis), která se skládá z mnoha příkazů. Každým příkazem říkáme prohlížeči jakou vlastnost má přiřadit určitému prvku, například modrou barvu k nadpisu apod. Každá vlastnost je pro CSS klíčová, jde o odborný termín. [25]

Každá vlastnost může nabývat jen v určité hodnotě, vlastnost i hodnota mají konkrétní názvy. Přidělení hodnoty vlastnosti oddělujeme středníkem. Pokud budeme chtít přidělit více vlastností, k tomu budeme potřebovat selektor. Pochází z anglického názvu `select`, česky vybrat. Označuje místo v dokumentu HTML, kterému chceme přidělit vlastnost, a zobrazí se na WWW stránce. [25]

Lze zápis zapsat takhle:

```
selektor {
vlastnost: hodnota;
vlastnost: hodnota;
}
```

Pokud budeme chtít přidělit stejné vlastnosti ještě k dalšímu selektoru, bude to vypadat následovně:

```
selector1, selector 2{
vlastnost: hodnota;
vlastnost: hodnota;
}
```

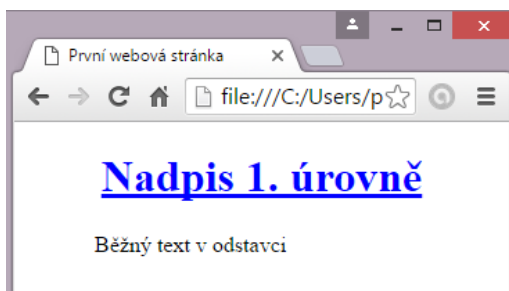
Na praktické ukázce si vyzkoušíme: nadpis podtrhneme, přidáme modrou barvu a zarovnáme na střed. U běžného textu si odsadíme na 50 pixelů a přidáme kurzívu (italic) (Obr. 9, 10, 11). [25]

```
<body>
  <h1>Nadpis 1. úrovně</h1>
  <p>Běžný text v odstavci</p>
</body>
```

Obrázek 9: Ukázka zápisu HTML kódu

```
/* CSS Document */
h1 {
  text-decoration: underline;
  color: blue;
  text-align: center;
}
p {
  text-indent: 50px;
  font-style: italic;
}
```

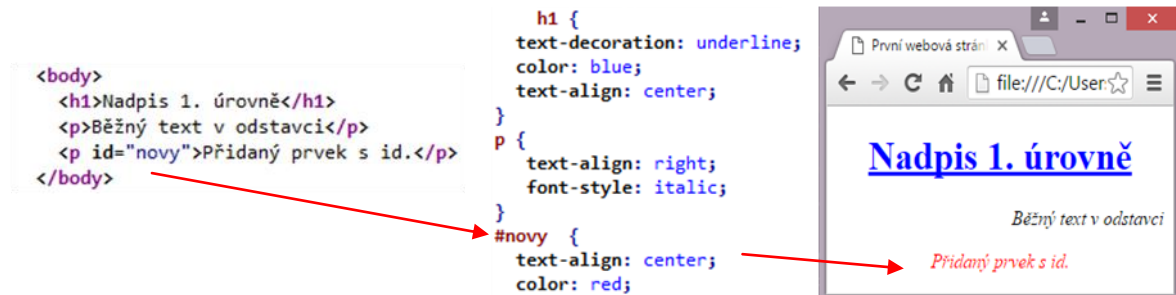
Obrázek 10: Ukázka zápisu v CSS dokumentu



Obrázek 11: Výsledné zobrazení v prohlížeči

3.4.1 Identifikátor

Prvek id používá atribut id elementu HTML pro vybrání konkrétního prvku. Id prvek by měl být jedinečný v rámci stránky. Chceme vybrat prvek s konkrétním ID, musíme napsat znak (#), následovaný id prvku (Obr. 12). Název ID nemůže začínat číslem. [26]



Obrázek 12: Ukázka vložení Syntaxe - Identifikátor

3.4.2 Prvek třída

Prvky vybíráme s konkrétním atributem třídy. Budeme chtít vypsát prvky s určitou třídou, musíme napsat první tečku (.), následně konkrétní prvek. Důležité je přiřadit prvek se stejným názvem, u kterých budeme chtít stejnou změnu (Obr. 13). [26] [27]



Obrázek 13: Ukázka použití Selector - Prvek třída

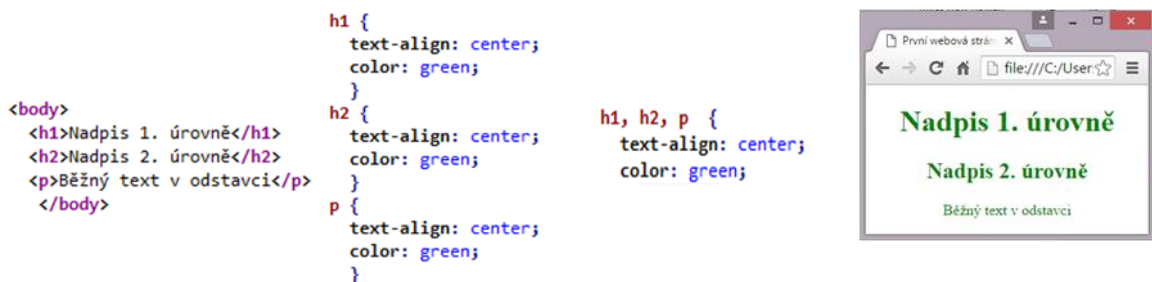
Html elementy mohou obsahovat více než jednu třídu. Před prvek a tečku musíme přidat ještě upřesnění elementu (Obr. 14). V našem případě půjde o element <p>. [26]



Obrázek 14: Ukázka použití Selector - Prvek třída

3.4.3 Seskupení prvků

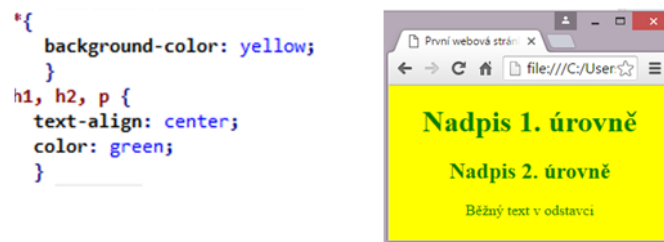
Máme prvky se stejnými definicemi stylů, můžeme je seskupit do jednoho. Ušetří nám místo v dokumentu a bude mít lepší přehled. Při změně stylů u všech nebudeme muset to měnit tři krát, ale pouze jednou, což nám ušetří čas. Skupiny se od sebe oddělí čárkou (Obr. 15). [26] [27] [28]



Obrázek 15: Ukázka použití seskupení prvků

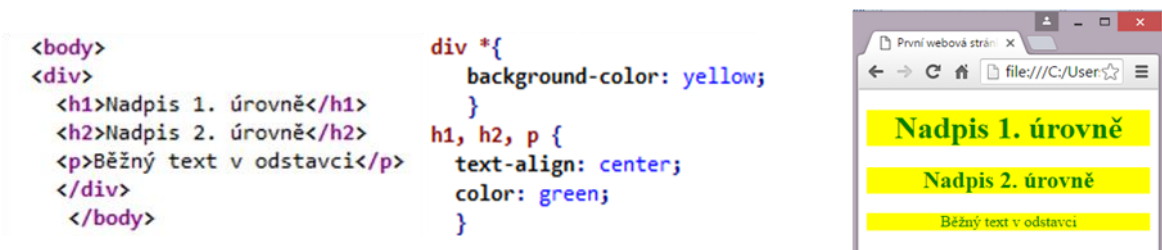
3.4.4 Selector *

Selector hvězdička (*) vybere všechny elementy a přiřadí zvolený styl (Obr. 16).



Obrázek 16: Ukátka použití Selector *

Můžeme vybrat také část elementů uvnitř jiného elementu. Mezi značkou div a selektorem * musí být mezera, jinak nebude selektor funkční (Obr. 17). [29]



Obrázek 17: Ukázka použití Selector *

3.4.5 Selector atributu

Selector atributu vybíráme na základě elementů hodnoty atributu. Využívají se k tomu hranaté závorky k přítomnosti samotného atributu. Mezi elementem a hranatými závorkami by neměla být mezera (Obr. 18). [30]



Obrázek 18: Ukázka použití Selector atributu

3.4.6 Pseudotřída

Pseudotřída je nějaký stav a vyznačuje se tím, že se v něm nachází element. Pro platnost vždy píšeme před pseudotřídou dvojtečku. Využití se nachází při posunu ukazatelem myši na odkaz (element a) a proběhne nastavená změna. [30]



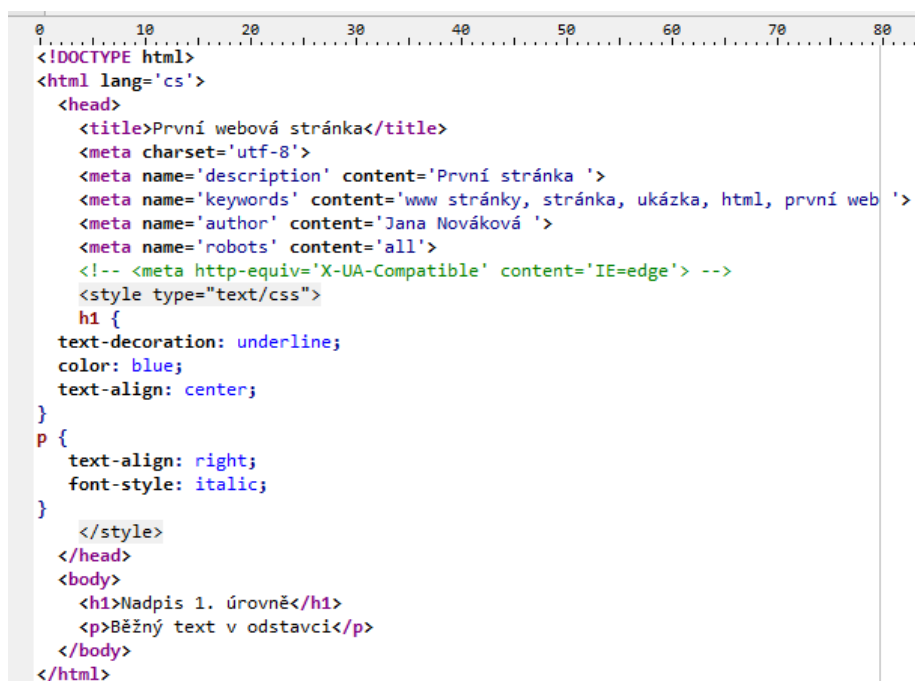
Obrázek 19: Ukázka použití Pseudotřída

3.5 Vložení CSS dokumentu

Kód CSS zapisujeme podobně jako jazyk HTML do běžných textových souborů. Máme dva způsoby vložení jazyka CSS. [25] [26]

- **Interní šablona**

Vložení párové značky <style> s parametrem type do hlavičky <head>. Mezi tyto značky bychom vkládali kaskádové styly na úpravu naší HTML stránky. Tento způsob můžeme využít pouze, na tento dokument, tzn. řádkový styl. Pokud budeme chtít použít na jiné dokumenty, musíme využít externí šablonu. Využívá se především pro rychle zobrazení stylů (Obr. 20). [25] [26]



```
0      10      20      30      40      50      60      70      80
<!DOCTYPE html>
<html lang='cs'>
  <head>
    <title>První webová stránka</title>
    <meta charset='utf-8'>
    <meta name='description' content='První stránka '>
    <meta name='keywords' content='www stránky, stránka, ukázka, html, první web '>
    <meta name='author' content='Jana Nováková '>
    <meta name='robots' content='all'>
    <!-- <meta http-equiv='X-UA-Compatible' content='IE=edge'> -->
    <style type="text/css">
      h1 {
        text-decoration: underline;
        color: blue;
        text-align: center;
      }
      p {
        text-align: right;
        font-style: italic;
      }
    </style>
  </head>
  <body>
    <h1>Nadpis 1. úrovně</h1>
    <p>Běžný text v odstavci</p>
  </body>
</html>
```

Obrázek 20: Ukázka Interní Šablony

- **Externí šablona**

Vytvoření externí šablony je používanější, praktičtější způsob umístění stylů. Šablona neobsahuje žádný element style, ale zde půjde jen o zápis pravidel CSS. Zápis se neliší ničím od interní šablony. Pravidla CSS se selektory a jejich vlastnostmi zapisujeme do textového souboru s příponou css (např. style.css).

Textový soubor s příponou snadno připojíme do dokumentu HTML. Prohlížeč bude využívat styly a elementy z externí šablony (Obr. 21). Výhodou této šablony jsou oddělené styly v samotném souboru od dokumentu HTML. [25] [26]

```

/* CSS Document */
h1 {
  text-decoration: underline;
  color: blue;
  text-align: center;
}
p {
  text-indent: 50px;
  font-style: italic;
}

```

Obrázek 21: Ukázka externí šablony

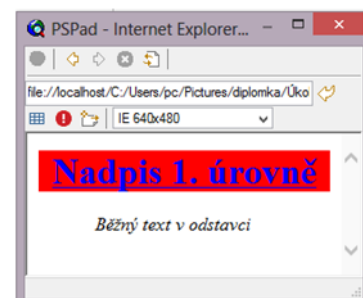
- **Inline styl**

Vlastností stylů jsou zapsány přímo v HTML pro konkrétní tag. Používá se výjimečně, ztrácí výhodu oddělení od HTML kódu a využití samostatně v dokumentu CSS. [31]

```

<body>
  <h1 style="background-color: red;">Nadpis 1. úrovně</h1>
  <p>Běžný text v odstavci</p>
</body>

```








Obrázek 22: Ukázka Inline Stylu

3.6 Testovací vlastnosti

Jazyk CCS se vyvíjí, ale objevuje se nový problém. Některé z prohlížečů nepodporují nové vlastnosti jazyka. Většina novinek z jazyka CSS3 už podporují prohlížeče Firefox, Internet Explorer, Chrome, Safari a Opera. Tohle je obrana před tím, že některé nedokončené nové vlastnosti se můžou změnit. Výrobci prohlížečů zavedli tzv. testovací vlastnost, která umožňuje zavést vlastní podporu vlastností. Tato podpora nekoliduje se staršími prohlížeči ani s finální specifikací. [20] [32]

Každý prohlížeč má přidělenou svoji vlastní testovací předponu (Tab. 1):

-webkitt-	prohlížeč s jádrem WebKit Safari a Chrome 
-moz-	Firefox 
-ms-	Internet Explorer 
-o-	Opera 
-khtml-	Konqueror 

Tabulka 1: Předpony testovacích vlastností [32]

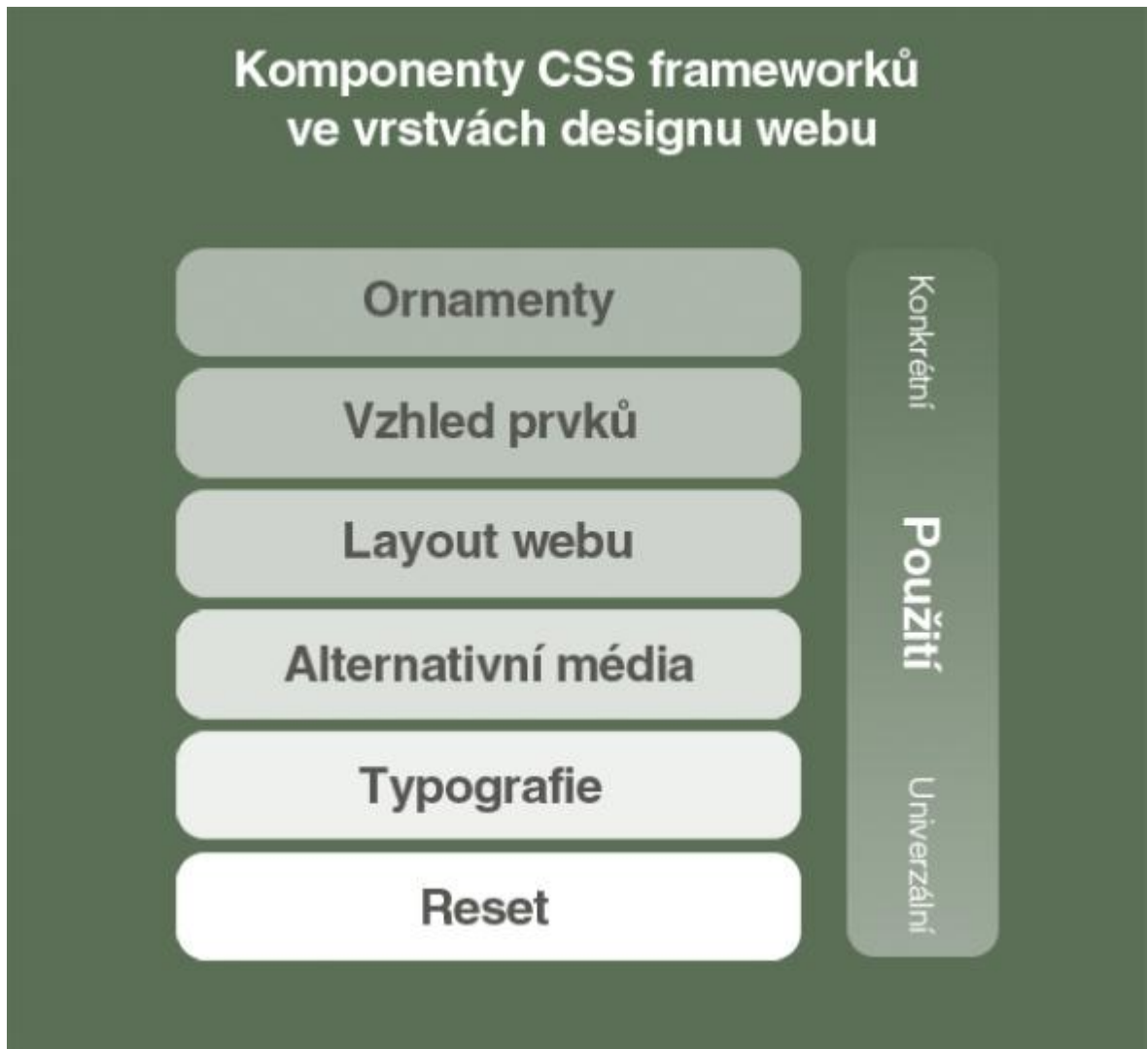
Předpony používáme k příslušné vlastnosti před názvem jazyk CSS. Není vždy důležité je používat a psát vždy a všechny. Nejčastěji využívané předpony jsou webkit a moz. [20] [32]

3.7 CSS framework

Pro komunikaci s kódery ocení designéři šablony postavené na gridu a platformě. Pro rychlé prototypování získají programátoři jednoduchý nástroj. Usnadníme si práci s obsahem kvalitní typografií. [33]

Design webu obecně velmi dobře odráží způsob, jakým můžeme přemýšlet, komponentech CSS frameworků. Skládá se z několika vrstev. Na povrchu se nachází estetické (ornamenty), které jen někdo využije a hodnocení vzhledu bývá subjektivní. Konkrétní využití jsou v horních patrech. Univerzální platnost spíše naopak spodní patro. Velmi široce jsou používány prvky úplně dole, které vývojářům a uživatelům prokážou velkou službu.

CSS framework se představuje jako hrad. Základy tvoří resetovací pravidla položené na pevno. [33]



Obrázek 23: Komponenty CSS frameworků [33]

Jednotlivé vrstvy si popíšeme:

- **Ornamenty** – své využití ornamenty naleznou v konkrétních případech a vzhled bude velmi individuální. Obsahuje sadu ikon, vzhled tlačítek a jiné sady CSS pravidel, které slouží k dekorativním a specializovaným účelům.
- **Vzhled prvků** – vzhled formulářů, typickým příkladem je systém pro definici. Blueprint CSS a specializovaný formulářový framework jsou příkladem obsahu. Pravidla vzhledu pro microformáty najdeme mezi pluginy s názvem YEML framework.
- **Layout webu** – definuje systém pro rozvržení stránky. Tato část patří mezi nejpopulárnějších Frameworku ze všech součástí CSS, ale taky nejvíce přeceňovaná.

- **Alternativní média** – obecně platná pravidla integrují některé frameworky pro vzhled tiskových sestav. Například pro mobilní zařízení bychom mohli definovat obecná pravidla podobně jako pro tisk.
- **Typografie** – resetovaným prvků přiřazují pravidla pro nový vzhled. V této části určíme, jak se nám bude výsledný text číst.
- **Reset** – jedná se o výchozí nastavení prohlížečů. Při stavbě webových stránek jde o první a zásadní úkon. [33]

4 BOOTSTRAP

Produkt Bootstrap byl vyvinut Mark Otto a Jacob Thornton na Twitteru jako framework. Mezi interními nástroji podporuje konzistenci. Před vznikem byla využívána řada knihoven, což vedlo k velké náročnosti na údržbu (tzv. nekonzistentní). Bootstrap byl zahájen v srpnu 2011 a mezi uživateli se dostalo velké oblibě.

Pro tvorbu webových aplikací a webu je bootstrap volně ke stažení. Produkt je velmi intuitivní, elegantní, výkonný, mobilní rámec pro rychlejší a snadnější vývoj webových aplikací. Návrhářské šablony jsou založené na HTML a CSS. Upravují typografii, formuláře, tlačítka, a další komponent rozhraní. Rozšíření JavaScriptu jsou stejně volitelná. [34] [35]

4.1 Využití Bootstrap

Pro vývojáře poskytuje jednoduché a čisté řešení pro budování rozhraní. Přizpůsobí se snadno a obsahuje krásné a funkční vestavěné komponenty. Poskytuje webové přizpůsobení a také open source. Vlastnosti HTML a CSS prvků vyžadují přítomnosti HTML5 *doctype* (`<!DOCTYPE html>`).

- **První mobilní přístup** - rámec Bootstrap 3 se skládá z mobilních stylů v celé knihovně, místo toho v samotných souborech. Navržení je přizpůsobeno tak, aby reagovala na mobilní zařízení.
- **Podpora prohlížeče** – podporuje všechny populární známé prohlížeče
- **Snadné začít** – znalostmi HTML a CSS můžeme začít s bootstrap.
- **Vnímavý návrh** – přizpůsobí se počítačům, tabletům a mobilním telefonům. [35]

4.2 Co obsahuje Bootstrap?

- **Struktura** – poskytuje základní strukturu bootstrap, styly odkazů a pozadí.
- **CSS** – přichází s nastavením CSS, základních prvků HTML zařízení a vylepšených tříd.
- **Komponenty** – obsahuje mnoho opakovatelně použitelných komponentů, které poskytnou rozbalovací nabídku, navigaci, upozornění apod.
- **JavaScript Plugin** – obsahuje mnoho vlastních jQuery pluginů. Můžeme je zahrnout všechny nebo jeden po druhém.
- **Přizpůsobení** – můžeme upravit komponenty Bootstrap, méně proměnné jQuery paginy a dostat své vlastní verze. [34]

4.3 Použití Grid

Grafický design, mřížka se rovná struktuře, obvykle dvou dimenzí. Skládá se z řady vertikálních a horizontálních linií, které jsou použity pro strukturování obsahu. Velmi často je používán k návrhu struktury layoutu i obsahu designu tisku. Nabízí velmi efektivní způsob při návrhu internetových stránek. Rychle a efektivně konzistentní rozložení pomocí HTML a CSS. [34] [36]

Bootstrap zahrnuje první mobilní distribuční soustavu, která se vhodně rozděluje až na 12 sloupců. Při zvětšování velikosti dosahuje právě zmiňovaných 12 sloupců (Tab. 2). Obsahuje předdefinované třídy pro snadné rozložení a rozvržení. Bootstrap 3 začíná nejprve na menších obrazovkách, jakou jsou mobilní telefony. Postupně se rozšiřuje na větší obrazovky tablety, notebooky a stolní počítače. [34] [36]

Span 1	Span 1	Span 1	Span 1	Span 1	Span 1	Span 1	Span 1	Span 1	Span 1	Span 1	Span 1
Span 4				Span 4				Span 4			
Span 4				Span 8							
Span 6						Span 6					
Span 12											

Tabulka 2: Rozdělení Bootstrap na 12 sloupců [36]

4.3.1 Použití Grid systému

Grid systémy se používají pro tvorbu rozvržení stránek prostřednictvím řádků a sloupců (Tab. 3). Vytvoříme každému svůj obsah.

- Řádky musí být umístěny uvnitř této značky (**.container**), pro správné zarovnání a výplně.
- Řádky pro vytvoření horizontálních skupin sloupců.
- Obsah by měl být umístěn uvnitř sloupců.
- Předdefinované třídy mřížky (**.row** a **.col-xs-4**), umožňují rychlé rozvržení mřížky.
- Vytvoření mezer mezi obsahem sloupců lze pomocí výplně (**padding**).
- Sloupce jsou tvořeny pomocí zadáním počtu 12 dostupných sloupců. [34]

	Malé přístroje Telefony (<768px)	Malé zařízení Tablety (≥768px)	Střední zařízení Notebook (≥992px)	Velké zařízení Stolní počítače (≥1200px)
<i>Chování Grid</i>	Horizontální po celou dobu	Zhroucení na začátku, horizontální nad zlomové body	Zhroucení na začátku, horizontální nad zlomové body	Zhroucení na začátku, horizontální nad zlomové body
<i>Maximální šířka</i>	Žádný (auto)	750px	970px	1170px
<i>Třída předpona</i>	.col-xs-	.col-sm-	.col-md-	.col-lg-
<i>Sloupce</i>	12	12	12	12
<i>Maximální šířka sloupce</i>	Auto	60px	78px	95px
<i>Šířka mezery</i>	30px (15px z každé strany sloupce)	30px (15px z každé strany sloupce)	30px (15px z každé strany sloupce)	30px (15px z každé strany sloupce)
<i>Zasouvateľné</i>	Ano	Ano	Ano	Ano
<i>Posuny</i>	Ano	Ano	Ano	Ano
<i>Uspořádaní sloupce</i>	Ano	Ano	Ano	Ano

Tabulka 3: Vlastnosti Bootstrap [34]

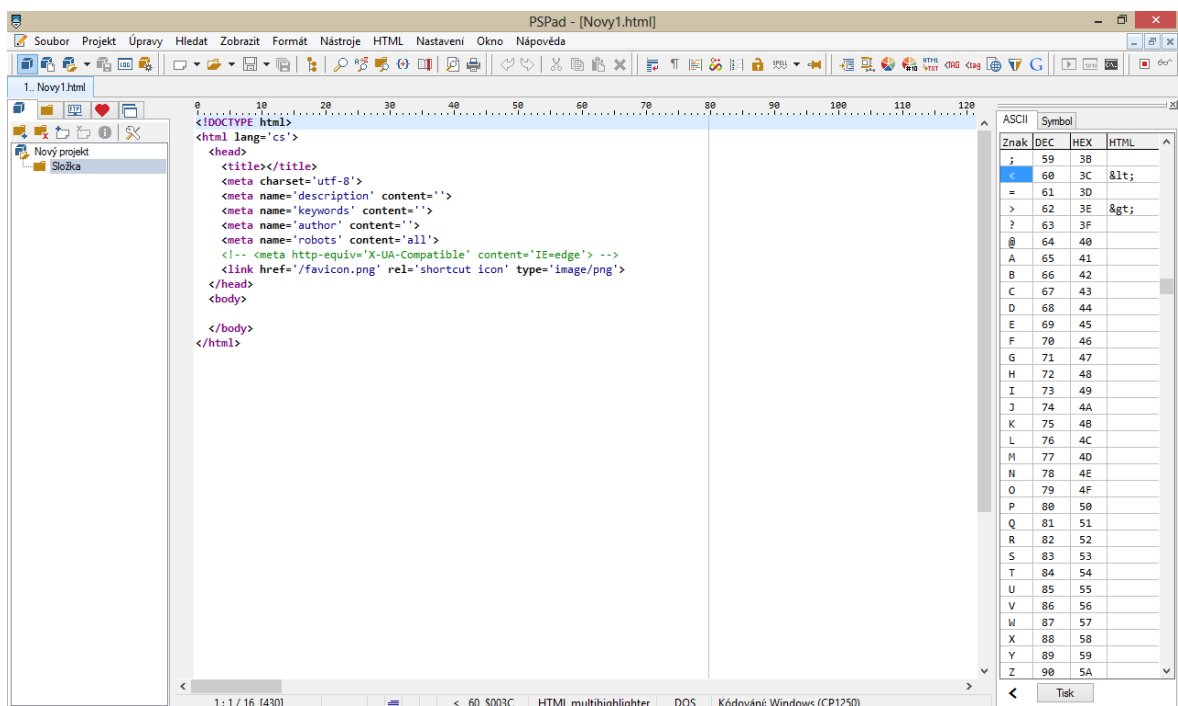
II. PRAKTICKÁ ČÁST

5 VÝUKA WEBOVÉ TECHNOLOGIE

Po spuštění programu PSPad se seznámíme s prostředím a ovládáním jednotlivých částí programu. Na obrázku číslo 24. můžeme vidět různé textové nabídky se základními i pokročilými nástroji. Pro okamžité zpřístupnění nástrojů slouží panel nástrojů s ikonami.

5.1 Prostředí PSPad

Nejpodstatnější část programu je samotné obsahové okno s bílou popsanou plochou. Tato plocha vychází ze šablony HTML5, kterou si následně popíšeme, kde ji nalezneme. Právě sem budeme zapisovat kód, kterým budeme tvořit WWW stránku. [25]



Obrázek 24: Výukové prostředí PSPad programu

- **Lišta s panely**

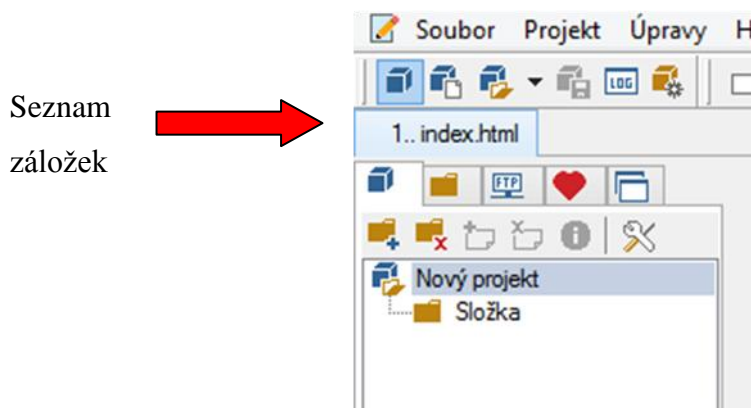
Lišta se nachází v horní části programu. Tento panel slouží ke všem potřebným úkonům v programu, jako jsou např. otevřít projekt, otevřít soubor, průzkumník kódu, hledání kódu a mnoho jiným prvků (Obr. 25).



Obrázek 25: Lišta s panely

- **Panel souborů**

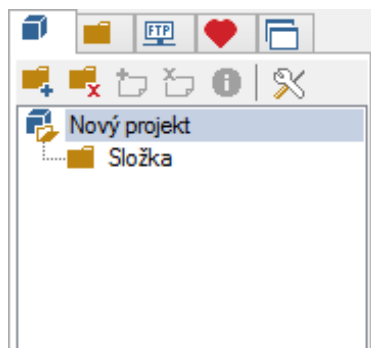
Zobrazuje seznam aktuálních souborů ve tvaru záložek (Obr. 26).



Obrázek 26: Panel souborů

- **Nástrojový panel**

Uspodňuje rychlou orientaci v prostředí programu PSPad. Umožňuje nám přehled, jak v otevřených dokumentech, tak rychlé vyhledávání ve struktuře počítače nebo připojení k FTP. FTP slouží k připojení s možností správy (přidání, úpravy, mazání apod.).



Obrázek 27: Nástrojový panel

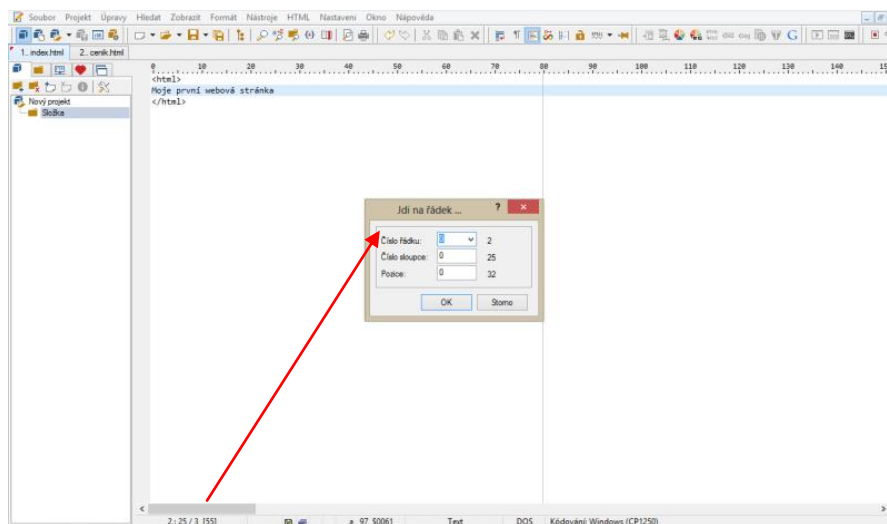
- **Stavový řádek**

Nalezneme ho v dolní části obrazovky programu a slouží k zobrazování stavu aplikace (Obr. 28).






Obrázek 28: Stavový řádek

1. Ukazatel pozice v souboru (řádek:sloupec (počet řádků) [počet znaků]), kliknutí vyvolá dialog "Jdi na řádek"(Obr. 29).



Obrázek 29: Ukazatel pozice v souboru

2. Indikace změněného souboru
3. Typ bloku pro označování textu normální /řádkový /sloupcový , kliknutím se mění typ bloku
4. Alternativní označení bloku - umístit kurzor na začátek, F8, umístit na konec, F8
5. Indikace zamknutého posunu
6. Aktuální znak a jeho hodnota v desítkové a šestnáctkové soustavě, kliknutí (Obr. 30) vyvolá ASCII tabulku (kompletní mapa znaků, které nemůžeme najít na klávesnici). [37]

5.2 Vytvoření HTML dokumentu

Pro tvorbu naší první webové stránky budeme potřebovat program PSPad. S prostředím programu jsme se seznámili (viz. výše) do podrobných detailů, co všechno je důležité k programu znát. Textový dokument od HTML dokumentu se liší jen tím, že do obsahu stránky se zahrnují navíc informace o vzhledu a formátování stránky, které do obyčejného dokumentu nepatří a jsou určeny jen pro prohlížeč. Informace se předávají pomocí značek uzavřených mezi znaky `<>` (Obr. 32). [38]

Pro mnohé z Vás nebudete znát jakým způsobem zobrazit zmiňované znaky. Pokud používáte českou klávesnici lze docílit znaku stisknutím **Pravý Alt** a klávesou nad pravý alt, která vyobrazuje zmiňovaný znak `<`, respektive **Pravý Alt + >**. [38]

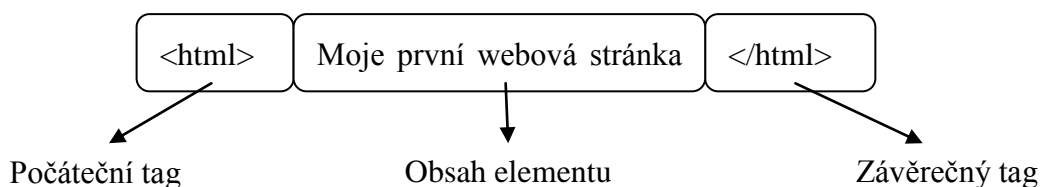
Druhým způsob, který by se mohl využít, je zmiňován ve stavovém řádku č. 6. Klikem se nám zobrazí ASII tabulka, kde můžeme najít požadovaný vzor. Rychlým levým dvojklikem na požadovaný znak, zobrazí se nám na pracovní ploše.

Spustíme program a pokusíme si vytvořit výstup naší práce a vložíme následující text:

```
<html>
```

Moje první stránka

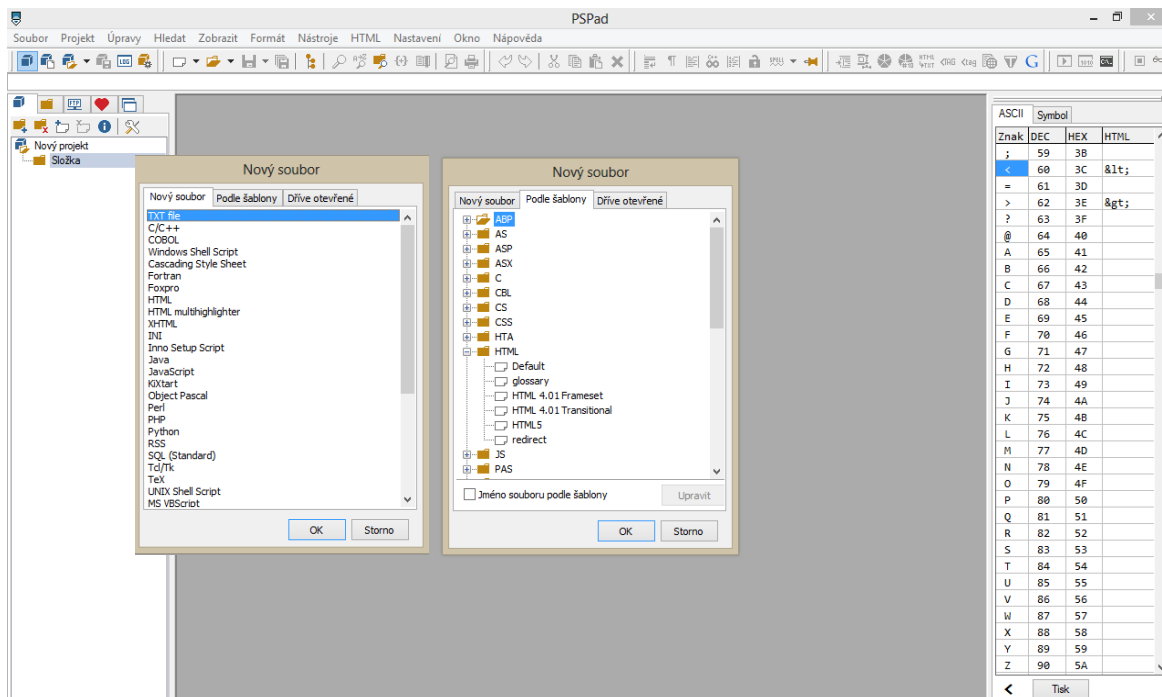
```
</html>
```



Obrázek 32: Názorná ukázka formátovacích značek

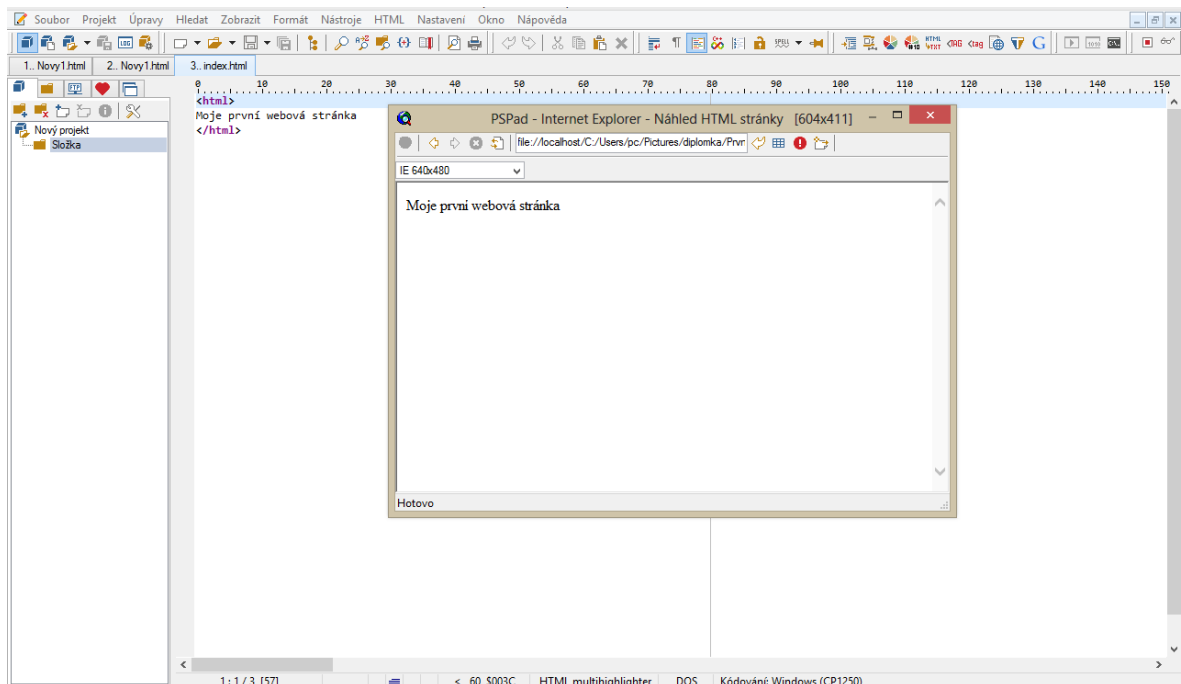
Ve značné většině se tyto značky ukončují podle lomítka před značkou `</html>`. Těmto formátovacím značkám se často říká tagy nebo také elementy. Značka `<html>` vyznačuje základ, který tvoří jazyk pro používání webové stránky v HTML.

Po spuštění programu musíme vytvořit nový soubor, který najdeme **Soubor => Nový** nebo klávesovou zkratkou `Ctrl+n`. Zobrazí se nám dialogové okno, kde můžeme si vybrat různé šablony nebo prázdný soubor (TXT file), který použijeme v tomto případě. Pro budoucí práci budeme využívat HTML5 soubor, nalezneme ho v šablonách a ve složce HTML (Obr. 33).



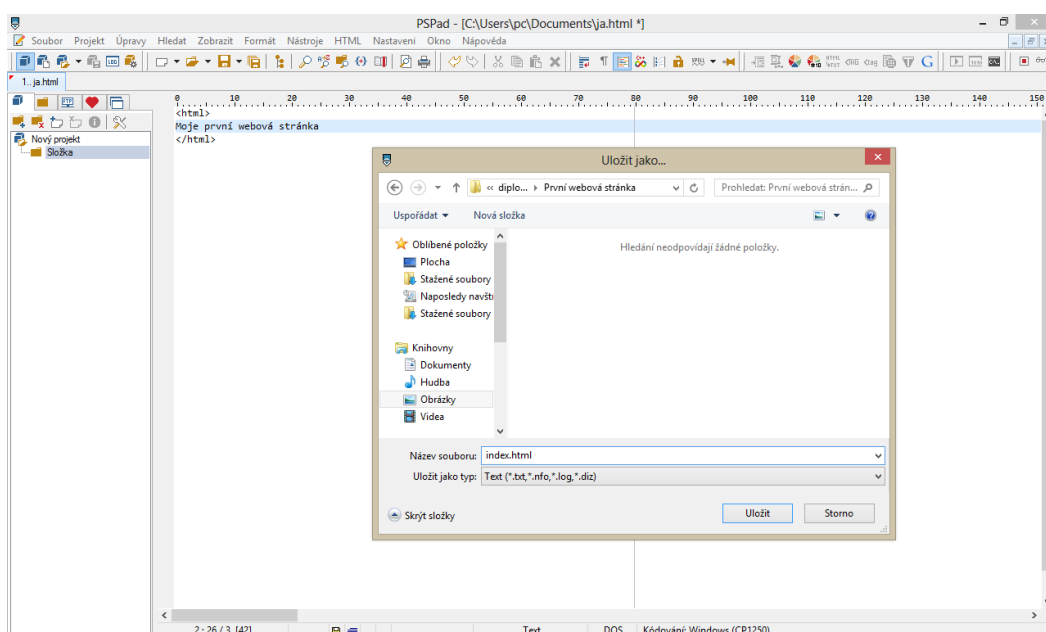
Obrázek 33: Vytvoření nového dokumentu v programu PSPad

Po vybrání příslušného souboru (txt file) napíšeme z výše uvedený text do prázdného souboru. Máme dva způsoby, jakým lze prohlédnout v prohlížeči text. První způsob je velmi jednoduchý, nemusíme dokument ukládat. Najdeme si v horní liště HTML a v rozbalovací nabídce klikneme na náhled HTML stránky (F10). Vyskočí okno, kde uvidíme výsledek naší vytvořené stránky (Obr. 34). [24] [39] [40] [25] [18]



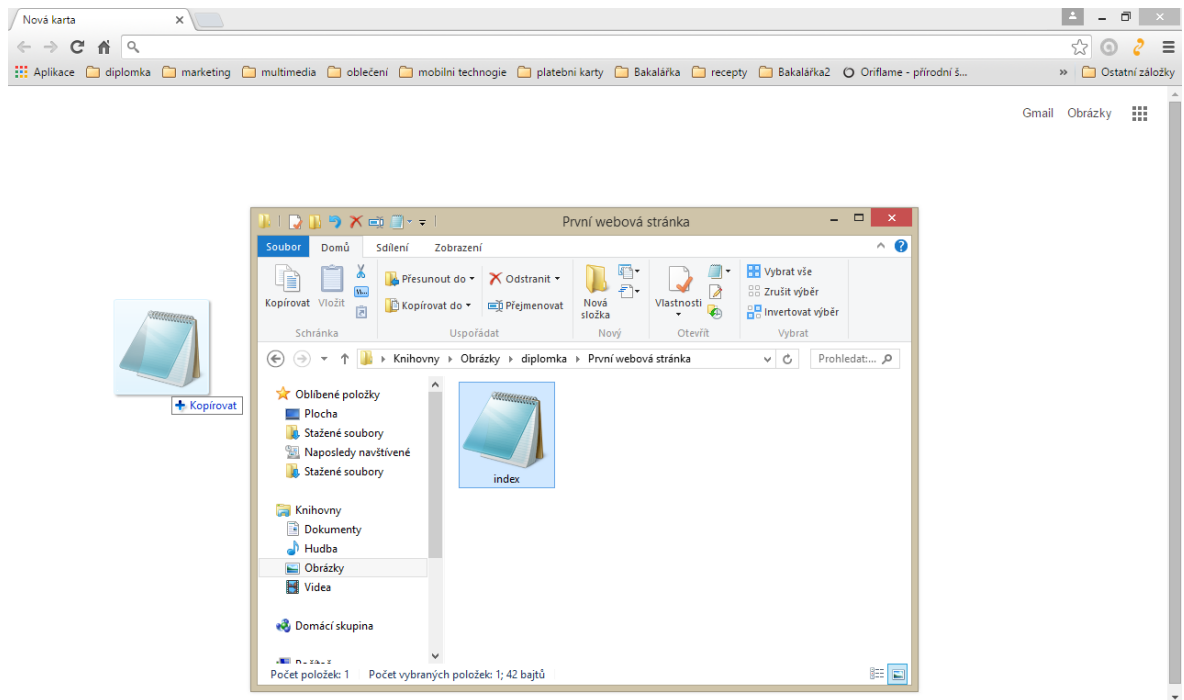
Obrázek 34: Ukázka rychlého zobrazení v programu

Pokud chceme vidět, jakým způsobem v prohlížeči se nám to zobrazí (např. Google Chrome, Internet Explorer atd.). Výsledný text musíme uložit na pevný disk, abychom mohli vidět výsledek. Proto je důležité vytvořit si složku např. První webová stránka a uložíme náš první dokument s názvem **index.html** do souboru. Soubor se vždy musí ukládat v textovém formátu HTML. U programu PSPad najdeme v horní liště **Soubor** => **Uložit jako** (F12) najdeme naši složku a potvrdíme uložit (Obr. 35). [24] [39] [40] [25] [18]

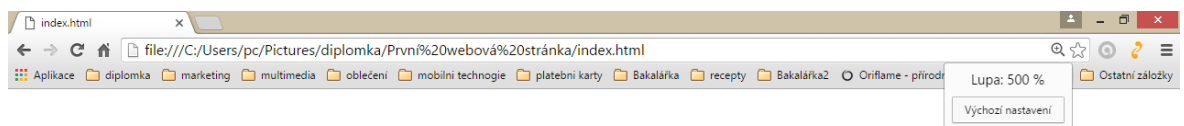


Obrázek 35: Ukázka uložení webové stránky

Nalezneme v počítači naši uloženou složku s požadovaným názvem. Ve složce se nám objevil soubor s názvem index. Otevřeme si internetový prohlížeč, v tomto případě byl použit Google Chrome. Jednoduchým přetažením souboru ze složky do prohlížeče (Obr. 36) zobrazí se nám výsledek (Obr. 37). [24] [39] [40] [25] [18]



Obrázek 36: Ukázka zobrazení v prohlížeči



Moje první webová stránka

Obrázek 37: Zobrazení v prohlížeči se zvětšením na 500%

5.3 Značky v HTML dokumentu

5.3.1 <HTML>

Celý HTML dokument obsahuje tag, který je rozdělen na hlavičku (<head>) a tělo (<body>).

```
<!DOCTYPE html>
```

5.3.2 Hlavička dokumentu <head>

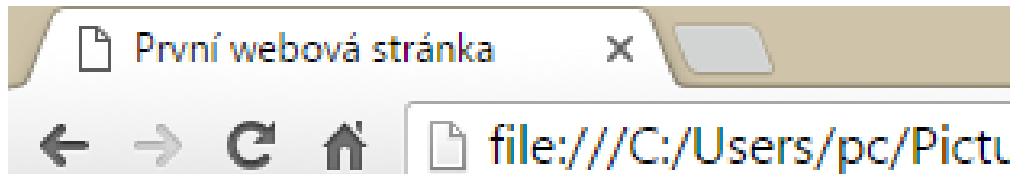
- obsahuje doplňující informace pro prohlížeč (např. název stránky, definice stylů, popis obsahu, atd.)
- obsah hlavičky se nezobrazuje na stránce, ale nemělo by se na ní zapomínat
- jedná se o párovou značku a tu vždy ukončíme </head >
- doplňující informace v Html5, které se nám zobrazí v šabloně a mi je musíme vypsát do příslušných volných odkazů mezi znak ' ':
 - <meta charset='utf-8'> **Znaková sada**
 - <meta name='description' content=' **Popis stránky** '>
 - <meta name='keywords' content=' **Klíčová slova stránky** '>
 - <meta name='author' content=' **Autor stránek** '>
 - <meta name='robots' content='all'> **Index celé naší webové stránky**
 - <link href='/favicon.png' rel='shortcut icon' type='image/png'> **Připojení ikony před adresou v prohlížeči**
 - <link href="styl.css" type="text/css" rel="stylesheet"> **Připojení stylů CSS**
[24] [39] [40] [25] [18]

5.3.3 Titulek stránky <title>

- patří k důležitému údaji, který je v hlavičce
- slouží k pojmenování dané www stránky
- vkládáme pouze běžný text mezi značky <title> a </title>
- nemůžeme sem umístit například obrázek nebo video
- název se nám zobrazí vždy na horní liště prohlížeče (Obr. 38) [18] [22] [25] [38] [40]

Příklad:

```
<html>  
<head>  
  <title> První webová stránka </title>  
</head>  
</html>
```



Obrázek 38: Zobrazení v horní liště prohlížeče

5.3.4 Tělo dokumentu `<body>`

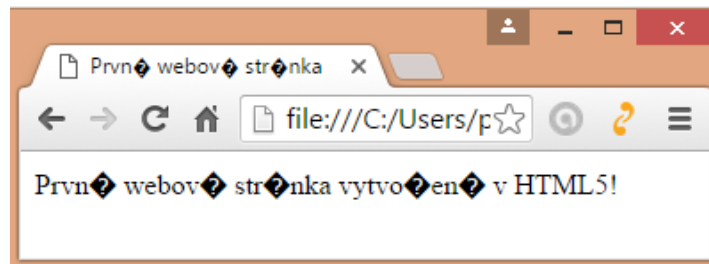
- zde je uložen veškerý obsah stránky (Obr. 39), který se nám zobrazí v prohlížeči (např. text, tabulky, obrázky, videa, atd.)
- Jedná se také o párovou značku a tu vždy ukončíme `</body>` [18] [22] [25] [38] [40]

```
<!DOCTYPE html>  
<html lang='cs'>  
  <head>  
    <title>První webová stránka</title>  
    <meta charset='utf-8'>  
    <meta name='description' content='První stránka '>  
    <meta name='keywords' content='www stránky, stránka, ukázka, html, první web '>  
    <meta name='author' content='Jana Nováková '>  
    <meta name='robots' content='all'>  
    <!-- <meta http-equiv='X-UA-Compatible' content='IE=edge' -->  
    <link href='/favicon.png' rel='shortcut icon' type='image/png'>  
  </head>  
  <body> <!-- Začátek těla dokumentu -->  
  První webová stránka vytvořená v HTML5!  
  </body> <!-- Konec těla dokumentu -->  
</html>
```

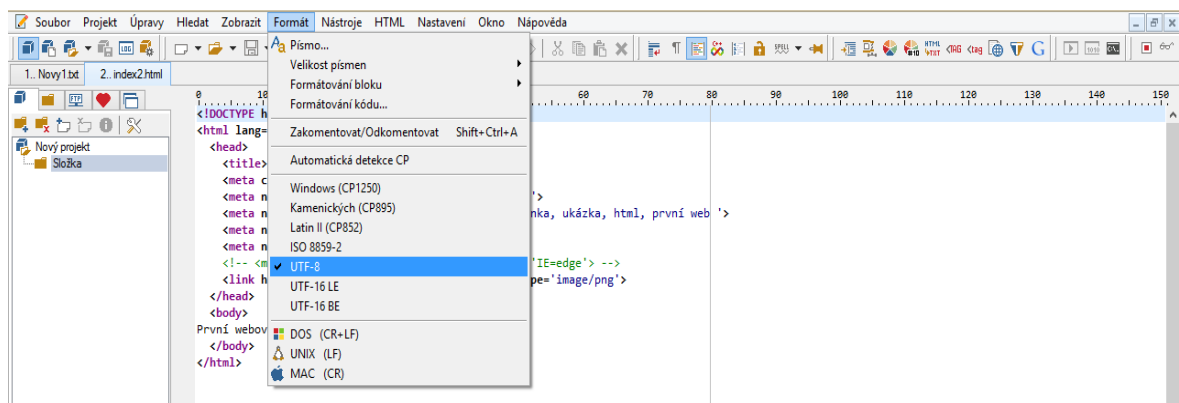
Obrázek 39: Tělo dokumentu HTML

Poznámka:

Pro zobrazení plnohodnotného textu v prohlížeči je potřeba v programu PSPad změnit kódovací stránku na UTF-8. Nalezneme ji **Formát** → **UTF-8**(Obr. 41). Pokud neuděláme tento příslušný krok, zobrazí se nám výsledný text v následující podobě (Obr. 40):



Obrázek 40: Nesprávně vložený formát



Obrázek 41: Výběr formátu UTF-8

5.4 Komentář v HTML

Další nedílnou součástí zdrojového kódu v HTML představují komentáře. Je to část kódu, které nemají žádný vliv na vzhled stránky a po zobrazení stránky nepůjdou nikde vidět. Umožňují autorovi stránky zapisovat přímo do dokumentu HTML. Autor ve složitých a dlouhých dokumentů se někdy nevyzná.

Vložením si usnadníme orientaci dokumentu jazyka a při zpětném vyhledání nám komentáře usnadní práci. Můžeme označit nejrůznější komentáře, například obrázek, který chceme umístit do dokumentu.

Komentář zapisujeme mezi znaky `<!-- komentář -->`. Hned v hlavičce dokumentu HTML se nám objevil jeden komentář, který značí, že bude přizpůsobivý k nejnovějším módům (Obr. 42). [18] [25] [41]

```
<!-- <meta http-equiv='X-UA-Compatible' content='IE=edge'> -->
```

Obrázek 42: Ukázka vloženého komentáře v hlavičce dokumentu

6 FORMÁTOVÁNÍ TEXTU

V této kapitole se zaměříme na text, aby byl na stránce přehledný, a jakým způsobem ho můžeme vylepšit. Pokud text má být na stránce přehledný, musí být rozčleněn na části. Každá část nadpisu by měla mít svoji značku k přiřazené úrovni. [18] [40] [42]

6.1 Nadpisy

Nadpisy v HTML mají značnou podporu, a proto není divu, že jsou výrazným členícím prvkem v textu. K dispozici v HTML máme 6 typů nadpisů. Budeme se více zajímat o úroveň nadpisu 1 až úroveň 6 (Obr. 43). Pro lepší orientaci na webu nadpisy slouží, jak pro čtenáře, tak pro prohlížeče. Zaměříme se jakou úroveň a kdy je vhodné ji použít. Nadpisy musíme zapsat mezi tělo (<body> a </body>). Jedná se o párové značky a ty se vždy ukončíme </...>

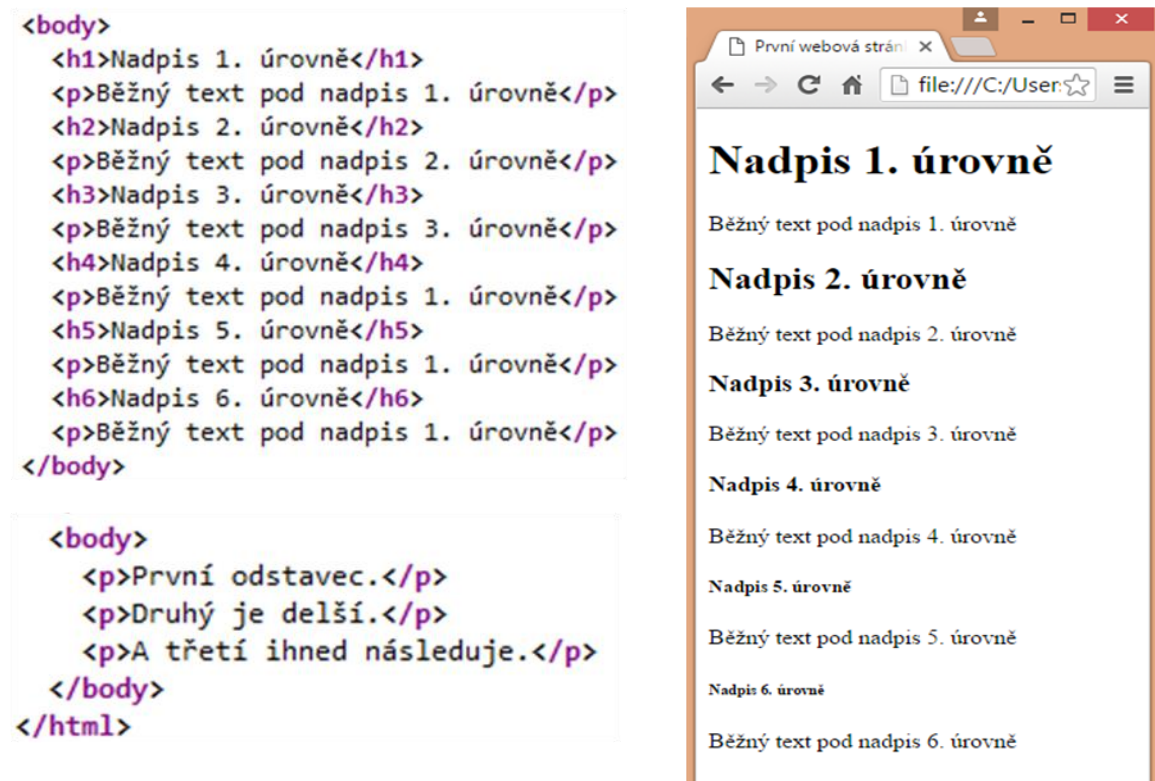
- <h1>
 - úroveň 1 má nejvyšší váhu a může být použit v dokumentu pouze jedenkrát
 - hlavní nadpis obsahuje název webu
 -
- <h2>
 - používá se jako nadpis dané podstránky/článku
 - vždy bude pod nadpisem h1
 - vyskytnout se na stránce může h2 vícekrát, ale není to obvyklé
- <h3> až <h6>
 - ostatní nadpisy vyznačují lepší orientaci v textu. [42]



Obrázek 43: Ukázka nadpisů

6.2 Odstavce

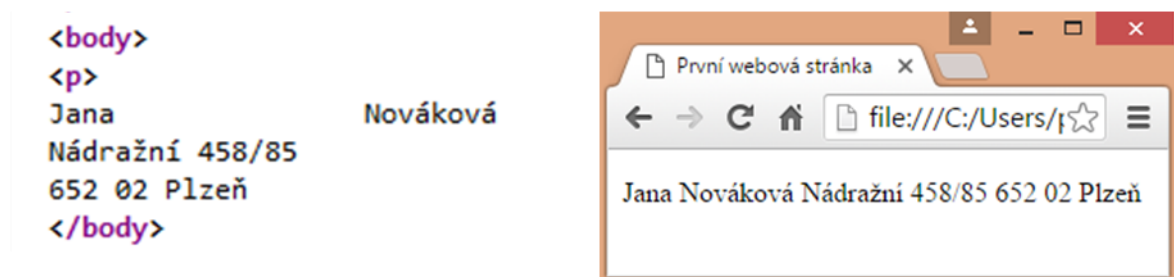
Text již máme rozdělen do nadpisů, ale zdaleka to ještě nestačí. Při psaní textu pod odstavcem je nutné označit přesně, kde odstavec začíná a končí (Obr. 44). Každý odstavec proto musí být uzavřen mezi značky `<p>` a `</p>` (tzv. p jako „paragraf“ = odstavec). Každý zdrojový kód ignoruje všechny mezery a znaky, které se zobrazí v prohlížeči. Pokud se nachází za sebou více mezer, zobrazí se jen jedna mezera. [18] [43]



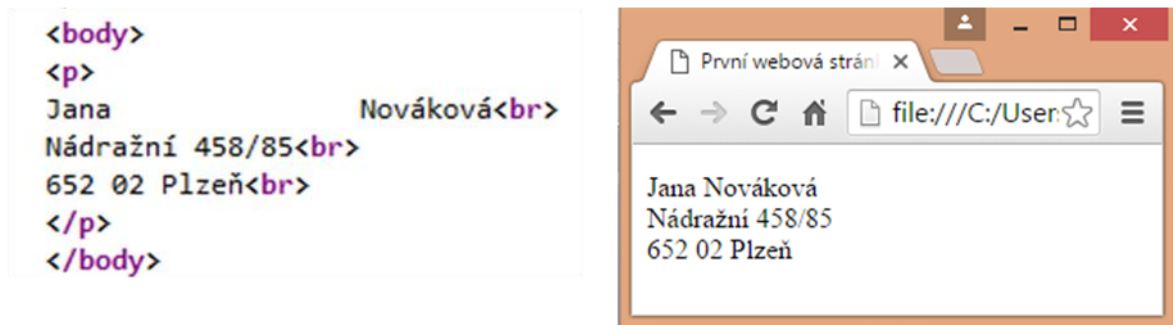
Obrázek 44: Ukázka Nadpisu a odstavců

6.2.1 Zalomení řádku

V některých situacích budeme potřebovat zalomit text v určitém přesně daném místě. Použijeme značku `
`, která docílí požadovaného zlomu v textu. Jde o nepárovou značku a ta nebývá zakončena `</br>` (Obr. 45,46). [18] [43]



Obrázek 45: Ukázka bez zalomení textu



Obrázek 46: Ukázka zalomení řádku

6.2.2 Zarovnání dokumentu

Zarovnáním můžeme ovlivnit celkový vzhled dokumentu, jenom na vyznačených jednotlivých částí. Pokud najdeme nadpis uprostřed stránky, vypadá to mnohem lépe. Pomůžeme si atributem *align*, který využijeme ve většině formátovacích znaků. [18]

K atributu *align* odpovídají způsoby následovně:

- **Left** – doleva
- **Center** – na střed
- **Right** – doprava

Pokud budeme chtít využít atribut u nadpisu, který by se zarovnal na střed (Obr. 47, 49), bude mít následující tvar:

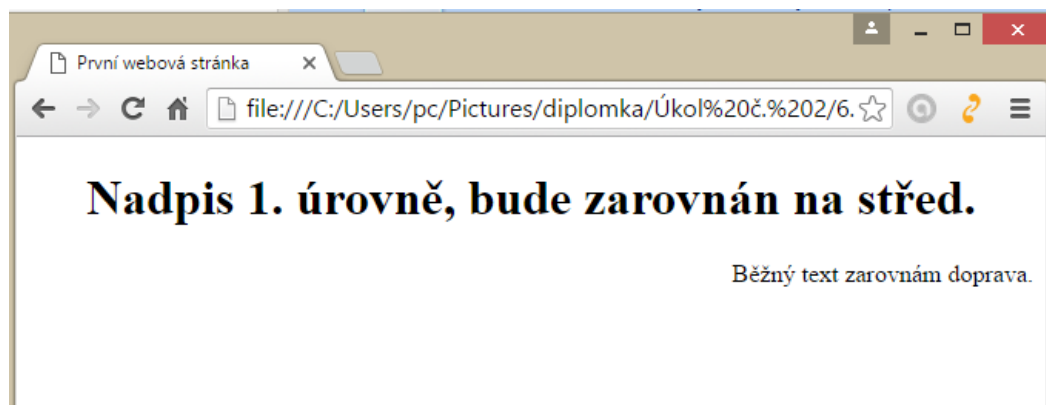
```
<h1 align="center"> Nadpis 1. úrovně, bude zarovnán na střed. </h1>
```

Obrázek 47: Ukázka zápisu zarovnání nadpisu

U běžného textu je tvar skoro stejný, jenom zaměníme za h1 na p (Obr. 48, 49):

```
<p align="right">Běžný text zarovnáám doprava. </p>
```

Obrázek 48: Ukázka zápisu zarovnání běžného textu



Obrázek 49: Zarovnání nadpisu a běžného textu

Pokud budeme chtít nastavit pro určitou část odstavců současně, budeme muset přidat ke stejnému atributu značku <div>. Nastavíme zarovnání pro ohraničované odstavce (Obr. 50). [18]

```
<body>
<div align="right">
<h1> Nadpis 1. úrovně, bude zarovnán na střed. </h1>
<p align="left">U běžného textu je tvar skoro stejný, jenom zaměníme za h1 na p:</p>
<p>Další odstavec se opět řídí značkou DIV. </p>
</div>
</body>
```



Obrázek 50: Zarovnání části odstavců

6.2.3 Zvýraznění textu

Chceme-li zdůraznit nějaké slovo nebo části slov v odstavci, použijeme vybranou značku. Jedná se o párovou značku, kde přesně bude mezi značkami vymezeno, o jakou část si vybereme a zvýrazníme text (Tab. 4,5), (Obr. 51). [18]

- **Základní**

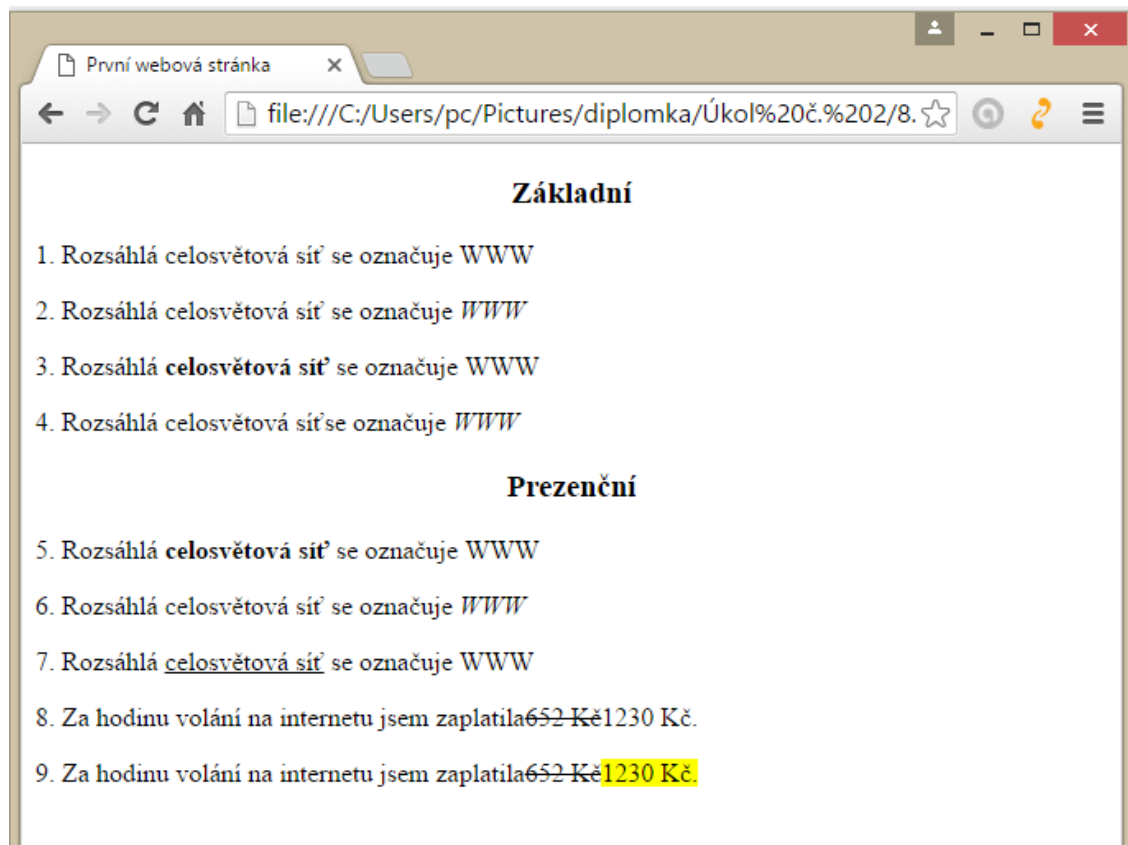
1.	<abbr>	Vyznačuje zkratku v textu, která nebude zobrazena v prohlížeči. (Př.: <p>Rozsáhlá celosvětová síť se označuje<abbr title="World Wide Web"> WWW</abbr></p>)
2.		Vyznačuje část textu, která je vykreslena v prohlížeči jako kurzíva (Př.: <p> Rozsáhlá celosvětová síť se označuje WWW</p>)
3.		Silnější zdůraznění části textu než značka . (Př.: <p> Rozsáhlá celosvětová síť se označuje WWW</p>)
4.	<dfn>	Definuje pojem k označení textu, vykreslí se kurzívou. (Př.: <p> Rozsáhlá celosvětová síť se označuje <dfn> WWW</dfn></p>)

Tabulka 4: Základní zvýraznění textu [44] [45]

- Prezenční

5.	<code></code>	Výsledný text se zobrazí tučně (Př.: <code><p>Rozsáhlá celosvětová síť se označuje WWW </p></code>)
6.	<code><i></code>	Vyznačuje část textu, která je vykreslena v prohlížeči jako kurzíva (Př.: <code><p> Rozsáhlá celosvětová síť se označuje<i> WWW</i></p></code>)
7.	<code><u></code>	Vyznačuje podtržení textu, již tato značka už se moc nepoužívá, zaměňuje se (např. s odkazem, podtržení překlepu). (Př.: <code><p> Rozsáhlá <u>celosvětová síť</u> se označuje WWW </p></code>)
8.	<code><s></code>	Vyznačuje přeškrtnutí textu (např. jiná cena zboží) (Př.: <code><p> Za hodinu volání na internetu jsem zaplatila<s>652 Kč</s>1230 Kč.</p></code>)
9.	<code><mark></code>	Zvýrazní část textu žlutým pozadím (např. zvýraznění části citace) (Př.: <code>p> Za hodinu volání na internetu jsem zaplatila<s>652 Kč</s><mark>1230 Kč.</mark></p></code>)

Tabulka 5: Prezenční zvýraznění textu [44] [45]



Obrázek 51: Ukázka základního a prezenčního textu

6.3 Seznamy

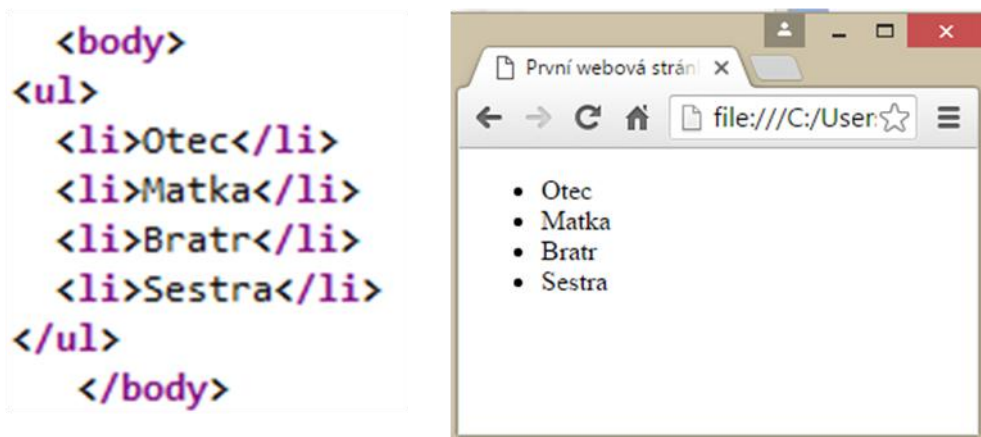
Seznamy jsou nedílnou součástí a velmi silnou zbraní jazyku HTML. Pomocí nich vytváříme strukturovanou část dokumentu např. navigační menu, seznam literatury, číslované kroky (Obr. 54) apod. Seznamy vkládáme opět mezi tělo `<body>`. [18] [25]

Rozlišujeme tři druhy seznamů:

1. Neuspořádané (nečíslované)

- Zahájíme celý seznam tagem ``.
- Vypsání počet položek seznamu, každá položka seznamu musí být zapsána mezi tagy `` a ``.
- Ukončíme seznam tagem `` (Obr. 52). [46] [47]

Příklad:

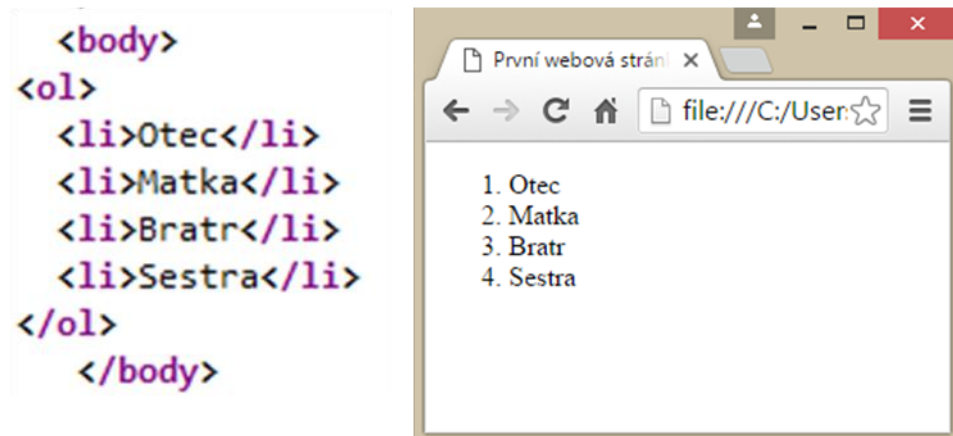


Obrázek 52: Ukázka neuspořádaného seznamu

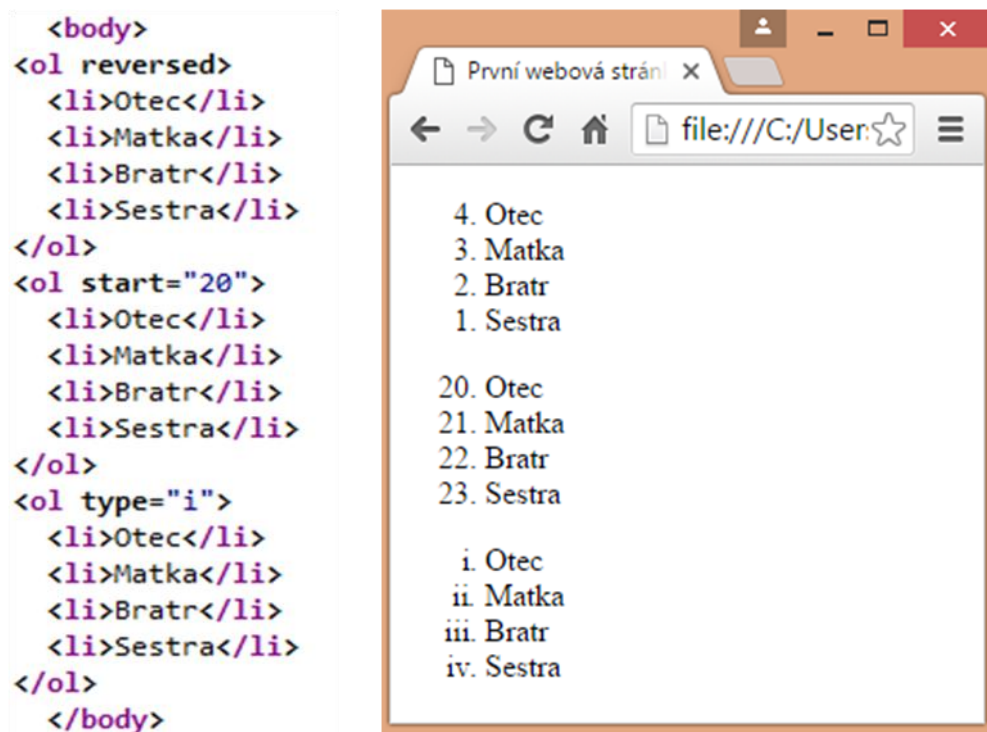
2. Uspořádané (číslované)

- Před každou položku je umístěováno automaticky pořadové číslo.
- Postup je v podstatě identický s neuspořádaným seznamem (Obr. 53).
- Záměna místo `` na ``.
- Nespořádaný seznam můžeme dále upravit přidáním k `` dalších atributů:
 - *reversed* – položky v seznamu jsou číslovány opačně, tedy sestupně
 - *start* – určuje nám, od jaké hodnoty začne první číslo v seznamu (např. `<ol start="20">`)

- *type* – typ číslování (např. 1, A, a, I, `<ol type="i">`), (Obr. 54).
[47] [48] [65]



Obrázek 53: Ukázka uspořádaného seznamu

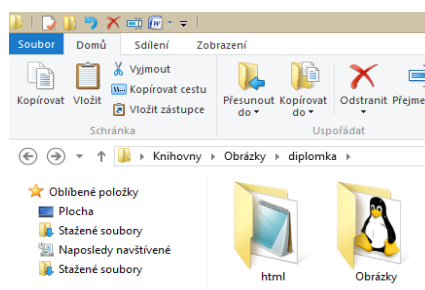


Obrázek 54: Ukázka typů seznamů

Budeme do ní ukládat veškeré obrázky, které budeme chtít použít v dokumentu. [18] [25] [49] [50]

Příklad:

V html složce máme html dokument a potřebujeme se dostat do složky Obrázky, abychom mohli zobrazit obrázek tucnak.png. Pokud budeme chtít ze složky vystoupit (značí dvě tečky (..) a lomítko (/)) a dostat se do složky obrázky (Obr. 57. 58), výsledek bude vypadat následovně: [18] [25] [49] [50]



Obrázek 57: Ukázka souborů ve složce

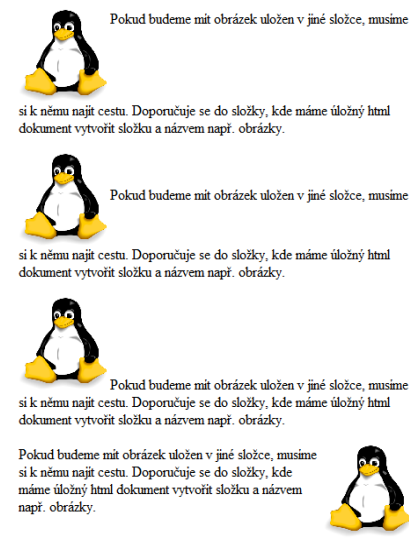
```
<body>

</body>
```

Obrázek 58: Zobrazení vystoupení a zobrazení z jedné složky do druhé

Chceme-li využít obrázek a text, musíme určit, kam má být umístěn okolní text. Slouží k tomu atribut align a pět možných způsobů (Obr. 59) zarovnání:

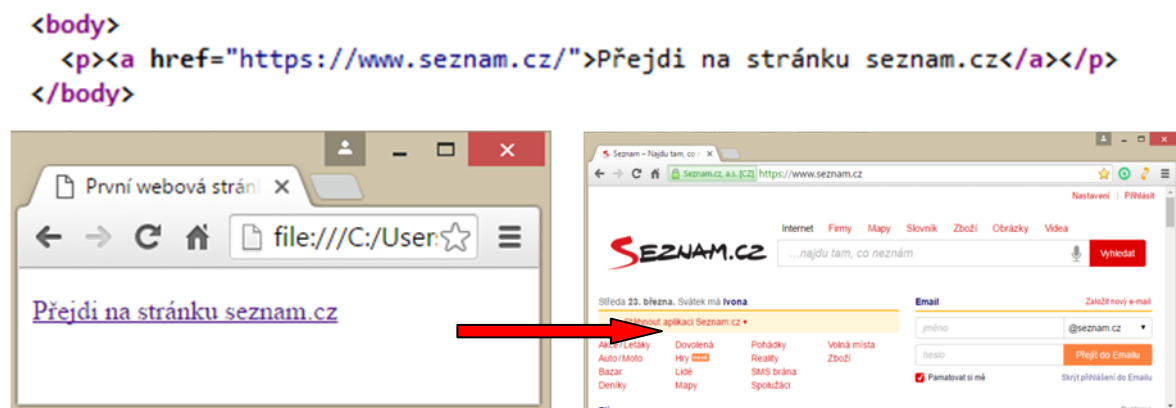
- align="top" – k horní hraně obrázku
- align="middle" – ve středu obrázku
- align="bottom" – k spodnímu okraji obrázku
- align="right" – obrázek obtékám textem z prava
- align="left" – obrázek obtékám textem z leva [18] [25] [49] [50]



Obrázek 59: Ukázka umístění textu kolem obrázku.

6.5 Odkazy

Pro určení cíle odkazu budeme potřebovat parametr `<a>`, který definuje hypertextový odkaz. Nejdůležitějším atributem `<a>` je `href`, tím vyznačujeme cílovou adresu odkazu (Obr. 60). Nejčastěji tvorba odkazů, jsou uvedeny přímo v textu (Tab. 6). Zde můžeme také vložit naše uložené stránky a přepínat si mezi nimi (Obr. 61). Dále můžeme zveřejnit na internetu dokument (MS World, Excel, PDF), hudbu a video.



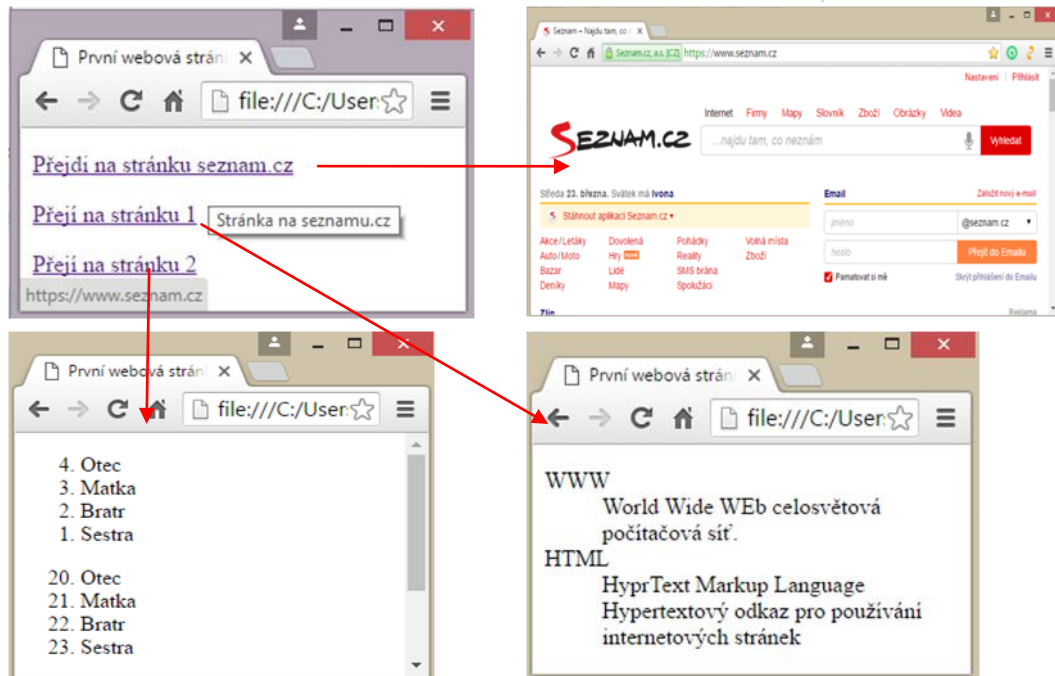
Obrázek 60: Ukázka odkazu na webovou stránku.

K atributu můžeme přidat *title*, který značí popis odkazu. Když najedeme na odkaz myší. [25] [51] [52]

```

<body>
  <p><a href="https://www.seznam.cz/" title="Stránka na seznamu.cz">Přejdi na stránku seznam.cz</a></p>
  <p><a href="9-3.html">Přejí na stránku 1</a></p>
  <p><a href="9-2.html">Přejí na stránku 2</a></p>
</body>

```



Obrázek 61: Ukázka více odkazů na webové stránky.

Atribut	Popis
<i>href</i>	Definuje cestu ke zdroji URL.
<i>target</i>	<ul style="list-style-type: none"> • <i>_blank</i> – odkaz se otevře v novém okně • <i>_parent</i> – odkaz se otevře ve stejném okně, ze kterého bylo na odkaz, kliknu to • <i>_self</i> – odkaz se otevře ve stejném okně, ze kterého bylo na odkaz, kliknu to

Tabulka 6: Popis atributů odkazu[51]

6.6 Tabulky

Tabulku vložíme do dokumentu pomocí značek `<table>` a `</table>`. Každý samostatný řádek je označen `<tr>` a `</tr>` a v řádku každá buňka označena `<td>` a `</td>`. Pokud budeme chtít vyznačit záhlaví tabulky, vyznačíme ji značkami `<th>` a `</th>`. Ohraničení

tabulky je potřeba k `<table>` přidat atribut `border`. Udává, jakou šířku bude mít rámeček (Obr. 62). [15] [16] [25]



Obrázek 62: Ukázka vytvoření tabulky

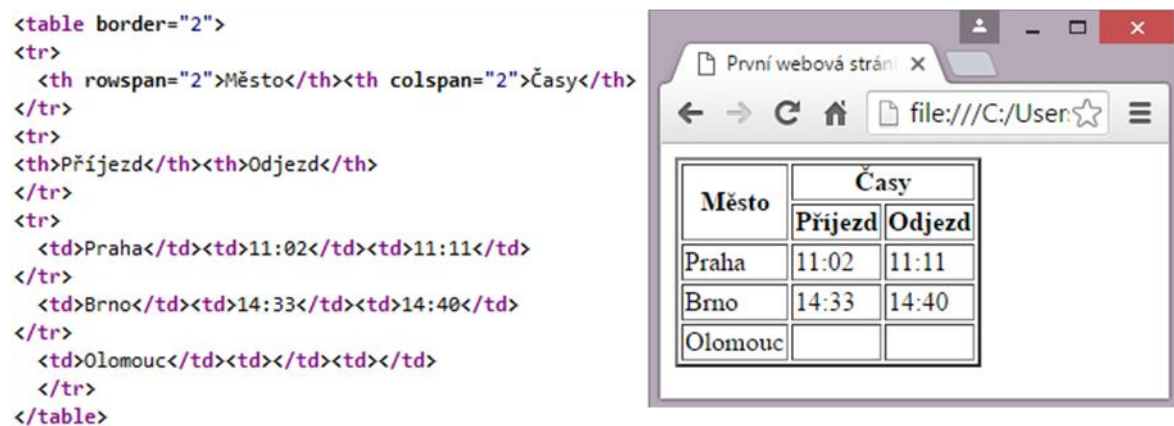
- **Sloučení buněk**

Při složitějším vytváření tabulek bývá někdy zapotřebí sloučit dvě a více buněk dohromady (Obr. 62). Platí pro značky **th** (nadpis) a **td** (buňka). Pro takové případy je dobré znát atributy:

- **colspan** – sloučí daný počet buněk, které jsou vedle sebe, do jedné buňky
- **rowspan** – sloučí daný počet buněk, které jsou pod sebou, do jedné buňky

- **Prázdná buňka**

Prázdnou buňku vyjádříme, že se mezi značky `<td>` a `</td>` vynechá text a zobrazí se nám jen prázdná orámovaná buňka (Obr. 63). [15] [16] [25]



Obrázek 63: Ukázka sloučení a prázdné buňky

6.7 Formuláře

Na mnoha internetových stránkách lze najít nejrůznější ankety, do kterých lze zapisovat, hlasovat o různých službách. Pomocí formulářů můžeme zadávat potřebné údaje na stránku, které budou následně odeslány k dalšímu zpracování na server. Umožňují vzájemnou komunikaci se serverem a uživatelem. Pro vytvoření formuláře je potřeba znát znaku `<form>` a `</form>`. Ve formuláři můžeme vybírat z mnoha ovládacích prvků, které vkládáme mezi značku `<input>` (Obr. 64), (Tab. 8). [16] [25] [40] [53]

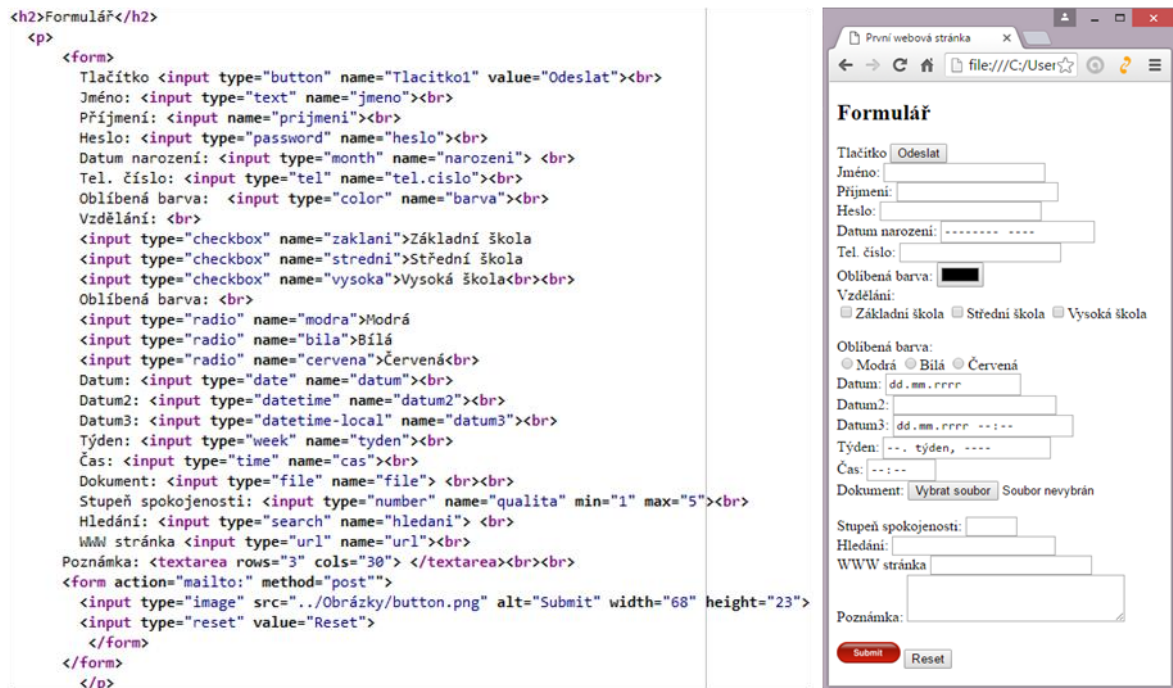
- `<input type="">`

Atribut	Popis
Bottom	Tlačítko
<i>Checkbox</i>	Vybrat jednu nebo více možností
<i>Color</i>	Vybrat barvu
<i>Date</i>	Datum
<i>Datetime</i>	Datum a čas
<i>Datetime-local</i>	Datum a čas předdefinované
<i>Email</i>	Pole pro email
<i>File</i>	Vybrat soubor
<i>Image</i>	Obrázek jako tlačítko
<i>Month</i>	Měsíc a rok
<i>Number</i>	Definovat pole pro zadání čísla (lze ji omezit, jaké čísla mají být přijímány)
<i>Password</i>	Heslo (znaky nejsou vidět)
<i>Radio</i>	Tlačítko na výběr pouze jedné možnosti
<i>Reset</i>	Vymaže všechny hodnoty z formuláře

Tabulka 7: Ukázka popisu atributu „`<input type="">`“ [53] [54]

- Textové pole

Při zadání dlouhého textu více než jeden řádek, musíme použít textové pole (Obr. 64). Značka pro textové pole je `<textarea>`, jedná se o párovou značku. Pole je možné nastavit pomocí atributu `rows` (počet řádků) a `cols` (počet znaků na řádek). [55] [18]



Obrázek 64: Ukázka typů atributu `<intype>` a textového pole.

- Seznamy

Slouží k výběru jedné nebo více hodnot ze seznamu. Vložíme ho pomocí párové značky `<select>`. Ke značce `select` je možné přidávat další aspekty, například vzhled i chování seznamu. Pomůžeme si vložením multiple u značky `<select>` rozhoduje, zda můžeme vybrat jedno nebo více položek.

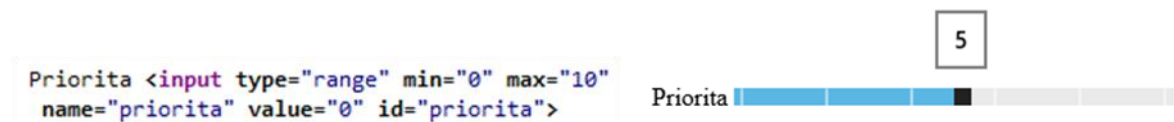
Dále se u téhle značky může objevit `size`, udávající kolik bude počet řádků v seznamu vidět. Pak jednotlivé položky do seznamu umístíme pomocí `<option>`. Seznamy lze ještě uspořádat do kategorií, abychom lépe označili skupiny a jejich podskupiny (Obr. 65). K tomu potřebujeme formátovací značku `<optgroup>` doplněná atributem `label`. [56] [18]



Obrázek 65: Ukázka seznamů

- **Posuvník**

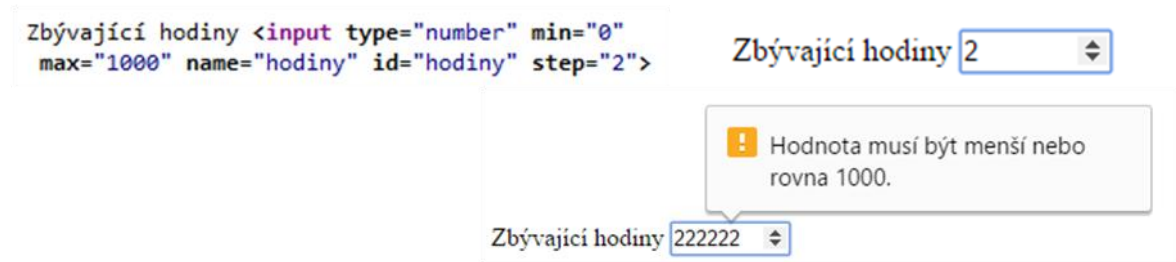
Využití posuvníku pro uživatele spočívá v tom, aby jezdce používali k možnosti snížit nebo zvýšit číselnou hodnotu. Dobrý příkladem využití posuvníku je vizualizovat rychlou prioritu nějakého projektu. Vytvoříme jezdce pomocí typu **range** a rozsah nastavíme atributy **min** a **max** (Obr. 66). [57]



Obrázek 66: Ukázka posuvníku.

- **Obsluha čísel**

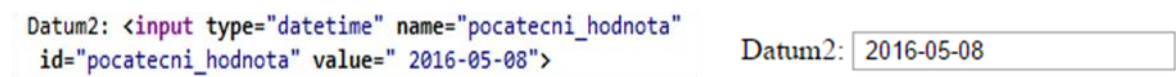
Využití obsluha čísel je velmi jednoduché a umožní snadné drobné změny. Ovládání prvku počítadla umožní šipky pro snížení a zvýšení nastavených hodnot v poli. Nejenom šipky umožní zobrazit hodnotu, ale také psaním číslic do pole. Lze nastavit rozsah hodnot od minimální do maximální. Nelze použít jakékoliv číslo, zobrazí se nám nápověda, do jakých hodnot jsou dostupná. Přidáním parametru **step** (Obr. 67) lze nastavit počáteční hodnotu (např. nastaven na 2, budou i následující hodnoty pokračovat násobením dvakrát. [57]



Obrázek 67: Ukázka Obsluha čísel.

- **Počáteční data**

Počáteční datum je vloženo do pole typu **date**, kde upřesníme počáteční hodnotu, kterou budeme chtít zobrazit (Obr. 68). [57]



Obrázek 68: Ukázka počátečních dat.

6.8 Audio a video

Představíme si zcela nové prvky `<audio>` a `<video>`, které bychom mohli použít v HTML5. Velmi důležité je vybírat správné formáty video souborů a zvukových souborů. [20] [58]

- **<video>**

Pro vložení videa je potřeba napsat párovou značku `<video>` a uzavřít `</video>`. Pro přehrávání videa na prohlížeči je potřeba znát formáty a kodeky (Tab. 9):

Prohlížeč	Kodek WebM přípona .webm	Kodek Ogg Theora přípona .ogg, .ogv	Kodek MPEG-4 (H.264) přípona .mp4, .m4v
<i>Firefox</i>	✓	✓	–
<i>Safari</i>	–	–	✓
<i>Opera</i>	✓	✓	–
<i>Google Chrome</i>	✓	✓	✓
<i>Internet Explorer</i>	–	–	✓

Tabulka 8: Kodeky pro `<video>`. [58]

Pro vložení video souborů, je potřeba si stáhnout soubor s příslušnou příponou. Nebo použít nástroj pro převádění video souborů do příslušné přípony (Obr. 69). Pro vložení videa je potřeba napsat kód ve tvaru:

```
<video src="../../obrázky/video.mp4">
</video>
```

Obrázek 69: Ukázka zápisu video souborů

Element má mnoho atributů, které je možné využívat pro další přehrání videa. Zde v tabulce můžeme najít atributy a jejich popis (Tab. 10). [20] [58]

Atribut	Popis
<i>src</i>	Definuje cestu ke zdroji URL videa
<i>autoplay</i>	Při spuštění stránky, prohlížeč začne sám načítat video soubor, přehrává automaticky od začátku do konce.
<i>controls</i>	K videu se doplní standardní ovládací prvky.
<i>muted</i>	Ztlumení zvuku videa.
<i>loop</i>	Prohlížeč bude video opakovaně přehrávat v cyklu.
<i>poster</i>	Specifikujeme název souboru, když načítá dané video, které by měl prohlížeč zobrazit. Přijímá URL adresu k obrázku, když není uveden atribut, prohlížeč zobrazí první snímek videa.
<i>width</i>	Šířka videa uvedena v pixelech.
<i>height</i>	Výška videa uvedená v pixelech.
<i>preload</i>	Sdělíme webovému prohlížeči, kterou část daného videa měl by zobrazit. Atribut může mít tři hodnoty: <ul style="list-style-type: none"> • none – nenačte prohlížeč nic • metadata – prohlížeč načte pouze délku, velikost apod. • auto – jde o výchozí nastavení, prohlížeč rozhodne sám, co udělá.

Tabulka 9: Popis funkcí atributu <video>. [20]

- **<audio>**

Pomocí jazyka HTML5 budeme se zabývat vkládání zvukových souborů. Pro audio soubory jsou také k dispozici několik souborových formátů (kodeků). V následujícím seznamu najdeme pět různých formátů a jejich přípony (Tab. 11), které podporují prohlížeče. [20] [58]

Prohlížeč	Kodeky				
	WAV přípona .wav	Ogg Vorbis přípona .ogg	MPE3 přípona .mp3	MPEG-4 přípona .mp4	AAC přípona .acc
<i>Firefox</i>	✓	✓	–	–	–
<i>Safari</i>	✓	–	✓	✓	✓
<i>Opera</i>	✓	✓	–	–	–
<i>Google Chrome</i>	✓	✓	✓	✓	–
<i>Internet Explorer</i>	–	–	✓	✓	✓

Tabulka 10: Kodeky pro <audio>. [58]

Pro vložení audio souborů, je potřeba si stáhnout soubor s příslušnou příponou. Nebo použít nástroj pro převádění audio souborů do příslušné přípony (Obr. 70). Pro vložení audia je potřeba napsat kód ve tvaru:

```
<audio src="../../obrázky/Chuckie.mp3" controls>
</audio>
```

Obrázek 70: Ukázka zápisu pro zvukové soubory

Elementy pro audio jsou převážně stejné, jen tady se neuvádí výška a šířka přehrávacího okna. Zde v tabulce můžeme najít atributy a jejich popis (Tab. 12). [20] [58]

Atribut	Popis
<i>src</i>	Definuje cestu ke zdroji URL videa

<i>autoplay</i>	Při spuštění stránky, prohlížeč začne sám načítat audio soubor, přehrává automaticky od začátku do konce.
<i>controls</i>	K audio se doplní standardní ovládací prvky.
<i>muted</i>	Ztlumení zvuku videa, tuto funkci zatím nenabízí všechny moderní prohlížeče.
<i>loop</i>	Prohlížeč bude video opakovaně přehrávat v cyklu.
<i>preload</i>	Sdělíme webovému prohlížeči, kterou část daného audia měl by zobrazit. Atribut může mít tři hodnoty: <ul style="list-style-type: none"> • none – nenačte prohlížeč nic • metadata – prohlížeč načte pouze délku, velikost apod. • auto – jde o výchozí nastavení, prohlížeč rozhodne sám, co udělá.

Tabulka 11: Tabulka funkcí pro <video>. [20]

- **<source>**

Atribut *source* definuje více zdrojů pro jediný prostředek (Obr. 71). Jelikož z výše uvedených kodeků pro video a audio si můžeme všimnout, že ne každý prohlížeč umí přehrát určitý typ kodeku. Tento atribut představuje univerzální základní kodek, který bude podporovaný ostatními prohlížeči. Pokud prohlížeč narazí na element video, zjišťuje, zda tam není atribut *src*. Pokud ne tak hledá elementy *source*.

Postupně prohledává jednotlivé zdroje typu *source* a najde si vhodný zdroj, který přehraje. Ostatní zdroje bude ignorovat. Smyslem vytvoření bylo, aby webový vývojáři nemuseli testovat, který prohlížeč podporuje jednotlivé zdroje. Tento typ atributu je použitelný jak pro video tak i audio (Tab. 13). [20] [58]

Atribut	Popis
<i>src</i>	Definuje cestu ke zdroji URL videa
<i>type</i>	Tento typ atributu pomáhá prohlížeči specifikovat typ videa, zda dokáže přehrát tento typ nebo ne. Hodnota musí odpovídat danému formátu nebo kodeku daného videa (video/mp4, video/webm, video/ogg)
<i>media</i>	Definuje zdroj videa, který bude ušitý na míru pro konkrétní výstupní zařízení. Umožní nám například používat menší videa pro zařízení s různými

	<p>nými velikostmi obrazovek. Atribut media podporuje zatím jen prohlížeče Safari a Opera.</p> <p>Př.:</p> <ul style="list-style-type: none"> • media="handheld" vyjadřuje mediální zdroj, že je vhodný pro mobilní zařízení • media="all and (min-device-height:720px) " vyjadřuje mediální zdroj, že je vhodný pro obrazovky se 720 řádky pixelů nebo s ještě větší šířkou. • Media="green and (min-device-width:100px) " vyjadřuje mediální zdroj, že je vhodný pro obrazovky se 100 řádkami pixelů nebo s ještě větší šířkou.
--	--

Tabulka 12: Tabulka funkcí pro <source>. [20]

```

<video controls="controls" width="320" height="180"> video
<source src="../obrázky/video.ogv" type="video/ogg">
</video>
<audio controls>
<source src="../obrázky/Chuckie.mp3" type="audio/mpeg">
</audio>

```

Obrázek 71: Ukázka zápisu atributu <source>

6.9 Geolokační rozhraní

Pomocí geolokačního rozhraní v HTML5 můžeme uživatele pořádat s jeho souhlasem o jeho polohu. V poslední době je tohle rozhraní velmi využíváno uživateli, jako aplikace při běhání. Před začátkem běhu si spustíme aplikaci a ta po dobu našeho běhu zaznamená vzdálenost, převýšení a trasu kde jsme běželi.

Aplikace si vyžádá od uživatele jeho aktuální polohu, pouze s jeho souhlasem o pozici. Pokud ano, prohlížeč uživatele vrátí informace o poloze. Prohlížeč zařízení dostane informaci o poloze, na zařízení, které podporuje geolokační rozhraní HTML5. Společně s dalšími metadaty udává zeměpisnou polohu (šířku a délku). Souhrn těch dat je možné vytvořit aplikaci, která využívá polohu uživatele. [17]

6.9.1 Zeměpisná šířka a délka

Zobrazení šířky a délky se zapisuje dvěma způsoby:

- V desetinném formátu (např. 25,326547)
- DMS formát, využívá stupně, minuty a sekundy (např. 25° 30' 60")

Geolokační rozhraní informuje také jinými souřadnicemi o přesnosti. Různé typy zařízení můžou obdržet další metadata, například nadmořská výška (*altitude*). Přesnější údaje o výšce (*altitudeAccuracy*), rychlost (*speed*) a směru (*heading*). [17]

6.9.2 Informace o poloze

HTML5 neuvádí, jakým způsobem má geolokační rozhraní určit polohu uživatele aplikace. Aplikační rozhraní nabízí jednoduché získání informací o poloze. Stanoví se společně se souřadnicemi také míra jejich přednosti. Není vždy záruka, že zařízení vrací přesné informace o pozici.

Zařízení vychází z následujících zdrojů informací o pozici:

- IP adresa
- Zaměření polohy
- Využití GSP (GlobalPositioningSystem)
- Podle adresy Mac Bluetooth
- Podle mobilní sítě GSM v telefonu nebo CDMA (Code Division Multiple Access)
- Zadané informace uživatelem [17]

6.9.3 Ověření podpory pro geolokaci

K zjištění polohy uživatele musíme mít na paměti, že spousta uživatelů nepoužívá prohlížeč, který podporuje geolokační rozhraní HTML5. Pro odeslání své polohy většinou nedali souhlas, a proto je důležité vždy ověřit, zda tyto informace je možné získat. V případě výskytu problému je dobré informovat uživatele o těchto problémech.

Ověření podmínky lze provést velmi jednoduše (Obr. 72). Zapisujeme do těla dokumentu mezi značky `<script type="text/javascript">` a `</skript>`. [59]

```
if (navigator.geolocation) {  
    // V geolokaci je možné pokračovat, je dostupná  
} else {  
    // V geolokaci není možné pokračovat, není dostupná  
}
```

Obrázek 72: Ukázka ověření polohy

- **getCurrentPosition()**

Pomocí této metody získáme polohu objektu, navigator.geolocation jsme ověřovali v obrázku č. 71. Tato metoda má další následující syntaxi `getCurrentPosition(showLocation, ErrorHandler, options)`.

- *showLocation*

- metoda callback, zavolá prohlížeč pro získání informací o poloze
- k tomuto objektu jsou přidány dva atributy `coords` a `timestamp` (povinný argument)

- *ErrorHandler*

- metoda callback, zavolá prohlížeče, pokud dojde k chybě při zpracování
- nepovinný argument

- *options*

- **enableHighAccuracy** – hodnota typu boolean, použije nejpřesnější dostupný způsob
- **timeout** – nastavení časového limitu pro prohlížeč, při překročení je vyvolán parametr `ErrorHandler`, čas je uveden v milisekundách
- **maximumAge** – určuje, zda prohlížeč může starou polohu použít z mezipaměti (cache), čas je uveden v milisekundách [59]

6.9.4 Získání polohy

Předání funkce **callback**, objekt disponuje atributy **coords** a **timestamp**. Implementujeme k atributům rozhraní objektu **Position**. Atribut **timestamp** zjišťuje polohu uživatele tzv. časové razítko s časem. Nejedná se o nijak důležitou informaci. Atribut **coords** disponuje se všemi informacemi o poloze. Implementace rozhraní **Coordinates**. Pro tento objekt jsou následující informace.

- *accuracy* – zeměpisná šířka a délka v přesnosti metrů; základní údaj
- *altitude* – nadmořská výška v metrech
- *altitudeAccuracy* – nadmořská výška v přesnosti metrů
- *heading* – udává směr pohybu zařízení, po směru hodinových ručiček (od severu ve stupních)
- *latitude* – zeměpisná šířka ve formátu desetinném, základní údaj
- *longitude* – zeměpisná délka ve formátu desetinném, základní údaj

- *speed* – udává aktuální rychlost pohybu zařízení v metrech za sekundu

Na příkladu si ukážeme vyskakovací okno s hlášením, které nám zobrazí zeměpisnou šířku a délku. Prohlížeč musí mít podporu pro lokalizaci (Obr. 73). Když nebude mít podporu prohlížeče, nezobrazí se nám žádná data. [59] [65]

```
// callback funkce pro zpracování polohy
function showLocation(position) {
    var latitude = position.coords.latitude;
    var longitude = position.coords.longitude;
    alert("Zeměpisná šířka: " + latitude + "\nZeměpisná délka: " + longitude);
}
// Ověření podpory prohlížeče
if (navigator.geolocation) {
    // V geolokaci je možné pokračovat, je dostupná
    navigator.geolocation.getCurrentPosition(showLocation);
} else {
    // V geolokaci není možné pokračovat, není dostupná
    alert("Podpora není dostupná.");
}
```

Obrázek 73: Ukázka získání a ověření polohy

6.9.5 Ošetření chyb

V každé aplikaci se nevyhneme chybám, i tak při lokalizaci může dojít k chybám. Vždy je dobré tyto chyby včas zachytit a informovat o nich. Zvětšuje to komfort uživatele a bezpečnost. Parametr **ErrorHandler** nám zajišťuje funkci **callback**, kterou předala metodě **getCurrentPosition**.

Předává objekt rozhraní **PositionError**, který obsahuje dva argumenty:

- *code* – chyby kódem
- *message* – detailní popis chyby

Definování chyb:

- **permission_denied** (číselná hodnota 1) – selhání z důvodu nedostatečného oprávnění (tzn. uživatel, nedal souhlas o přístup své polohy)
- **position_unavailable** (číselná hodnota 2) – nebylo možné určit polohu zařízení, došlo k interní chybě v lokalizaci
- **timeout** (číselná hodnota 3) – časový limit byl překročen [59]

Ukázka zachycení chyb pomocí *callback*, můžeme provést pomocí konstrukce **switch** (Obr. 74).

```
// ošetření chyb
function errorHandler(error) {
  switch (error.code) {
    case 1:
      document.getElementById("error").innerHTML = "Chyba: Přístup byl odepřen.";
      break;
    case 2:
      document.getElementById("error").innerHTML = "Chyba: Poloha není k dispozici.";
      break;
    case 3:
      document.getElementById("error").innerHTML = "Chyba: Vypršel časový limit.";
      break;
    default:
      document.getElementById("error").innerHTML = "Chyba: " + error.message;
      break;
  }
}
```

Obrázek 74: Ukázka ošetření chyb.

6.9.6 Aktualizovaná poloha

Metoda pro opakovatelnou aktualizaci polohy se nazývá *watchPosition()*. Tato metoda zajišťuje časový interval aktualizaci polohy zařízení. Syntaxe je schodná s metodou **getCurrentPosition**: (*watchPosition(showLocation, ErrorHandler, options)*).

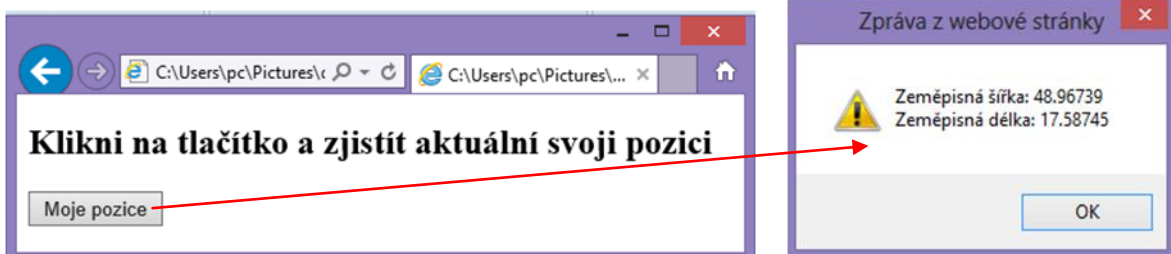
Následující příklad poukazuje na tlačítko (Obr. 75), když na něj klikneme, vyskočí nám okno s informacemi o naší poloze (šířka a délka). Pro zobrazení byl použit internetový prohlížeč Internet Explorer. [59]

```
<body>
  <script type="text/javascript">
var watchID;
var geoLoc;

// callback funkce pro zpracování polohy
function showLocation(position) {
  var latitude = position.coords.latitude;
  var longitude = position.coords.longitude;
  alert("Zeměpisná šířka: " + latitude + "\nZeměpisná délka: " + longitude);
}

// funkce je volána po kliknutí na tlačítko a zajišťuje volání aktualizace polohy
function getLocationUpdate() {
  if (navigator.geolocation) {
    geoLoc = navigator.geolocation;
    watchID = geoLoc.watchPosition(showLocation);
  }else{
    alert("Tvůj prohlížeč nepodporuje geolokaci!");
  }
}
  </script>

  <!-- formulář s tlačítkem pro aktualizaci polohy -->
  <h2> Klikni na tlačítko a zjistit aktuální svoji pozici</h2>
  <form>
    <input type="button" onclick="getLocationUpdate();" value="Moje pozice">
  </form>
</body>
```



Obrázek 75: Ukázka příkladu na geolokaci pomocí tlačítka.

6.10 Microdata

Microdata umožňují označovat obsah tvůrcům webových stránek. Informace dostupné prohlížečům nebo vyhledávačům musí být zpracovatelné strojově. Tři formáty Microdata, Microformáty (definice vlastních slovníků) a RDFa (závislé na slovnících) dosáhnout obdobného výsledku na vašich preferencích.

Ian Hickson cítil, že je ještě jedno místo mezi formáty. Tento formát by měl stačit dnešnímu dynamicky se rozvíjejícímu webu. Dále, aby byl jednoduchý a rozšířitelný mezi webdesignéry, kteří budou používat tento formát bez dlouhého učení. [60]

6.10.1 Funkce

Data se označují pomocí párů *jméno/hodnota*, definování jmen ve slovnících.

Microdata jsou slovníky, které definují jednotlivá jména vlastností. Příkladem může být slovník se jménem „Osoba“, kde jsou vlastnosti „webové stránky“, „jméno“ a „fotografie“. V HTML použijeme tento slovník. Každý slovník má svoji unikátní **namespace** (webová adresa, na které je umístěn) a pak už jen popisujeme, co potřebujeme. Pracují také se **zanořováním**, když budeme chtít vytvořit například položku „Osoba“, kde jsou vlastnosti jméno, fotografie a webové stránky. [60]

- **itemscope**

Značí novou položku, která má nějaké vlastnosti a můžeme je zanořovat do sebe. Položku můžeme dát ke skupině (itemscope) a popsat do detailů.

- **itemtype**

Odkázání na slovník. Absolutní URL adresa musí být hodnota platná. Můžeme vytvářet vlastní nebo již hotové (doporučeno) slovníky.

- **itemprop**

Specifikace jednotlivých vlastností celé položky. Specifikace ve slovníku jsou hodnoty itemprop. [60]

6.10.2 Schéma.org

Výše uvedený atribut **itemtype**, má odkazovat na existující schéma. Ve spolupráci s největšími firmami jako jsou Google, Microsoft (Bing) a Yahoo vytvořili za tímto účelem portál **Schéma.org**. Na odkazu nalezneme veškeré informace a také o dokumentaci akceptovaných microdat vyhledávači.

Na názorné ukázce si vysvětlíme situace, ke které se může dostat většina webdesignerů. Jako jsou autoři vlastních webových stránek nebo blogů, kde jiní uživatelé přispívají příspěvky. Redaktoři by rádi informace o přispívajících předali skrz microdata robotům vyhledavačům. Nejdříve také musíme zjistit na Schéma.org nejbližší objekt autora. Na stránkách nalezneme objekt **Person**, který využijeme v našem případě (Obr. 76, 77). [61]

```

<body>
<!-- Varianta zápisu číslo 1-->
<h2>Zápis Microdat</h2>
  <div class="authinfo" itemscope itemtype="http://schema.org/Person">
<p>Autor:<span itemprop="name">Jana Fibichrová</span></p>

<p></p>

<span itemprop="jobTitle">Student</span>
<p>Telefon:<span itemprop="telephone"> +420 604 444 444</span></p>
<p>E-mail:<a href="mailto:fibichrova@gmail.com" itemprop="email"> fibichrova@gmail.com</a></p>

<p><span itemprop="description">Jana Fibichrová je studentkou navazujícího studia
      na Fakultě aplikované informatiky ve Zlíně.</span></p>

  <p><a href="http://www.fibichrovawww.hys.cz/" itemprop="url">Blog</a></p>
</div>
<!-- Varianta zápisu číslo 2-->
<h2>Zápis Microdat</h2>
  <section class="authinfo" itemscope itemtype="http://schema.org/Person">
<h3>Autor:<span itemprop="name"> Jana Fibichrová</span></h3>

<p></p>

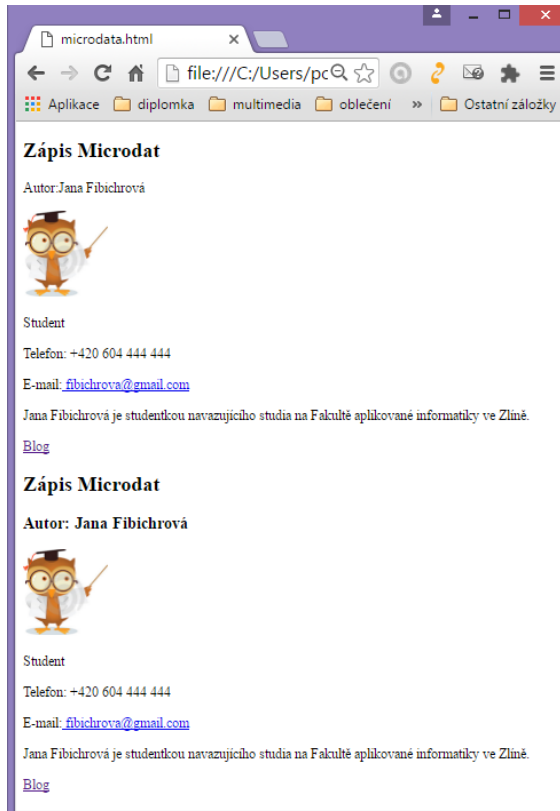
<p itemprop="jobTitle">Student</p>
<p>Telefon: <span itemprop="telephone">+420 604 444 444</span></p>
<p>E-mail:<a href="mailto:fibichrova@gmail.com" itemprop="email"> fibichrova@gmail.com</a></p>

<p itemprop="description">Jana Fibichrová je studentkou navazujícího studia
      na Fakultě aplikované informatiky ve Zlíně.</p>

  <p><a href="http://www.fibichrovawww.hys.cz/" itemprop="url">Blog</a></p>
</section>
</body>

```

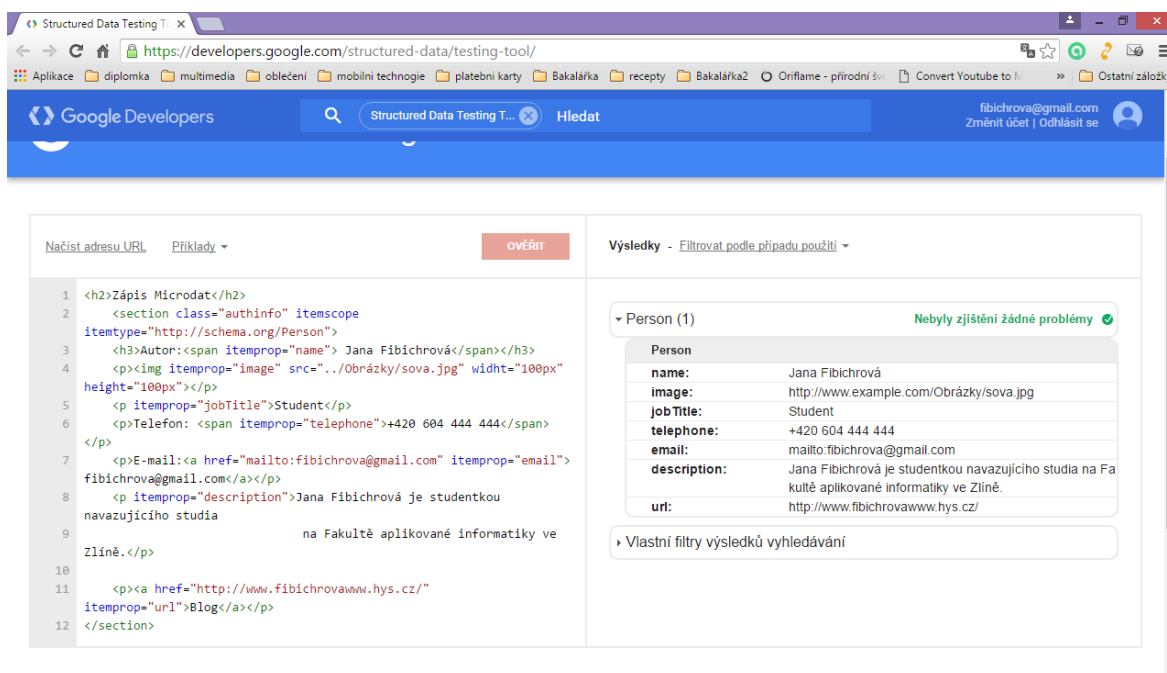
Obrázek 76: Ukázka dvojího zápisu v HTML.



Obrázek 77: Ukázka zobrazení v prohlížeči.

6.10.3 Kontrola microdat

Společnost Google poskytuje nástroj **Rich snippet testing tool**, kde můžeme zjistit funkčnost našich dat. Na výběr máme dvojí kontrolu pomocí vložení zdrojového kódu nebo URL adresy. Pomůže nám zajistit a zkontrolovat správnost stránky a **zobrazit náhled výsledku** ve vyhledávání. Náhled výsledku bude zobrazen jen při správném vyhodnocení roboty vyhledavačů (Obr. 78). [61]



The screenshot shows the Google Developers Rich Snippet Testing Tool interface. The browser address bar displays `https://developers.google.com/structured-data/testing-tool/`. The page title is "Structured Data Testing T...". The interface is divided into two main sections:

- Left Panel (Code Editor):** Contains HTML code for a microdata snippet. The code includes an `<h2>` tag, an `<section class="authinfo" itemscope itemtype="http://schema.org/Person">` tag, and various `<p>` tags with `itemprop` attributes for name, image, jobTitle, telephone, email, and description. The code ends with `</section>`.
- Right Panel (Results):** Shows the results of the test. It indicates "Person (1)" and "Nebyly zjištěny žádné problémy" (No problems detected). Below this, a table displays the structured data extracted from the code:

Person	
name:	Jana Fibichrová
image:	http://www.example.com/Obrázky/sova.jpg
jobTitle:	Student
telephone:	+420 604 444 444
email:	mailto:fibichrova@gmail.com
description:	Jana Fibichrová je studentkou navazujícího studia na Fakultě aplikované informatiky ve Zlíně.
url:	http://www.fibichrovawww.hys.cz/

Obrázek 78: Kontrola zápisu nástrojem Rich snippet testing tool.

7 KASKÁDOVÉ STYLY

V programu PSPad otevřeme nový dokument pro externí šablonu CSS. Postup bude následovný:

1. Vytvoření souboru

Soubor → Nový → Podle šablony → CSS → Default → OK.

2. Vkládání stylů

Do tohoto souboru budeme vkládat námi napsané styly.

3. Uložení souboru

Soubor uložíme pomocí Soubor → Uložit. V následujícím okně zvolíme název šablony (např. styly.css). Nesmíme zapomenout na řádně označenou koncovku css. Uložíme do stejného adresáře jako dokument HTML a potvrdíme.

4. Připojení stylů do HTML souboru

Do hlavičky (<head>) přidáme nepárovou značku link, s ní připojíme externí šablonu. Dbáme na to, aby název se shodoval s uloženým souborem. Celý zápis zapišeme tímto způsobem:

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

Obrázek 79: Ukázka nepárové značky vložené do hlavičky.

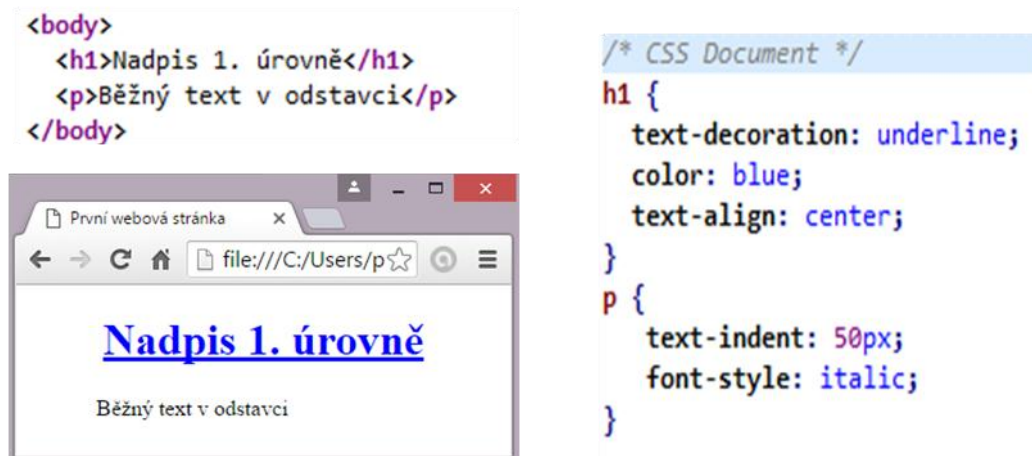
- **link** – nepárová značka, která slouží k propojení externí šablony s HTML dokumentem
- **rel** – říkáme prohlížeči, že cílový dokument je šablona stylů, jeho hodnota je *stylesheet* neboli česky šablona
- **type** – jeho hodnota je *text/css*, říkáme prohlížeči, že půjde o zápis kaskádových stylů
- **href** - URL adresa k souboru s externí šablonou s příponou css. [25]

5. Praktická ukázka

Na praktické ukázce si vše vyzkoušíme, zda bude CSS soubor plně funkční (Obr. 79):

- nadpis podtrhneme (*text-decoration: underline;*),
- přidáme modrou barvu (*color: blue;*)
- zarovnáme na střed (*text-align: center;*)
- běžný text odsadíme na 50 pixelů (*text-indent: 50px;*)

- přidáme kurzívu (*font style: italic;*)



Obrázek 80: Ukázka vložení externího souboru do hlavičky HTML

7.1 Komentáře v CSS

Jak jsme si mohli všimnout při otevření prázdné stránky CSS v programu PSPad, zobrazil se nám šedivý nakloněným písmem text (*/* CSS Document */*). Program ho umísťuje na začátek souboru, ale nezobrazí se v prohlížeči. Je to pouze doplňující informace (Obr. 80) pro ty co tvoří webové stránky, aby se lépe orientovali v šabloně. [25] [26]

```

/* CSS Document */
h1 {
  text-decoration: underline; /* text potřzený*/
  color: blue;                /* barva modrá*/
  text-align: center;        /* text zarovnan na střed*/
}
p {
  text-indent: 50px;         /* text odsazený o 50 pixelů*/
  font-style: italic;       /* text kurzíva*/
}

```

Obrázek 81: Ukázka komentářů v CSS

7.2 Barva

Je několik typů pro zapsání barevného vyobrazení (Obr. 81).

- **Název**

Barvu lze nastavit pomocí plného názvu.

- **Hodnota RGB**

Hodnotu barev lze zadat pomocí vzorce RGB (červená, zelená, modrá). Každý parametr (červená, zelená, modrá) určuje intenzitu barvy mezi 0 až 255. [26]

- **Hodnota RGBA**

Barevné hodnoty RGBA vychází z hodnot barev RGB s alfa kanálem, které představuje písmeno A. Využívá ke specifikaci krytí pro barvu. Parametr A (alfa kanál) se uvádí v čísle v rozmezí 0,0 (zcela průhledné) a 1,0 (zcela neprůhledné). [62]

- **Hexadecimální**

RGB hodnoty lze zadat také pomocí hexadecimální hodnoty barev ve tvaru: #rrggbb. Hodnoty RR (červená), GG (zelená) a BB(modrá) jsou hodnoty mezi 00 až FF (je to stejné jako u RGB modelu mezi 0-255). [26]

- **Hodnota HSL**

Hodnota HSL představuje:

- Hue (barevný tón) – míra barevné škály je od 0 až 360
- Saturation (sytost) – Vyjadřuje se procentem (např. 100% je plně barevný)
- Lightness (světlost) – vyjádřeno procentem (např. 0% je černá, 100% je bílá) [62]

- **Hodnota HSLA**

Tato hodnota barvy se skládá z hodnot HSL, ke které je přidán alfa kanál. Písmeno A specifikuje krytí pro barvu. HSLA představuje vlastnosti stejné jako j HSL. Pro alfa kanál jsou parametry v rozmezí 0,0 (zcela průhledné) a 1,0 (zcela neprůhledné). [62]

- **Průhlednost**

Vlastnost neprůhlednosti představuje značka opacity, která nastavuje hodnotu krytí. Hodnota pro nastavení vlastnosti musí mít číslo mezi 0,0 (zcela průhledné) a 1,0 (zcela neprůhledné). [62]

```

<body>
  <h2>Barvy</h2>

  <p class="navez">Barva na pozadí s barevným označení : Název.</p>
  <p class="rgb">Barva na pozadí s barevným označení - RGB.</p>
  <p class="rgba">BBarva na pozadí s barevným označení - RGBA.</p>
  <p class="hexa">Barva na pozadí s barevným označení - Hexadecimální.</p>
  <p class="hsl">Barva na pozadí s barevným označení - HSL.</p>
  <p class="hsla">Barva na pozadí s barevným označení - HSLA.</p>
  <p class="pruhlednost">Rámeček na počátku okraje ve 3D - Průhlednost.</p>

</body>

```

```

p.navez {background-color: yellow;}
p.rgb {background-color: rgb(204, 0, 102);}
p.rgba {background-color: rgba(0, 0, 255, 0.3);}
p.hexa {background-color: #CC0000;}
p.hsl {background-color: hsl(75, 100%, 60%);}
p.hsla {background-color: hsla(240, 100%, 30%, 1);}
p.pruhlednost {background-color: rgb(192,192,192);opacity:0.6;}

```



Obrázek 82: Ukázka barevného vyobrazení.

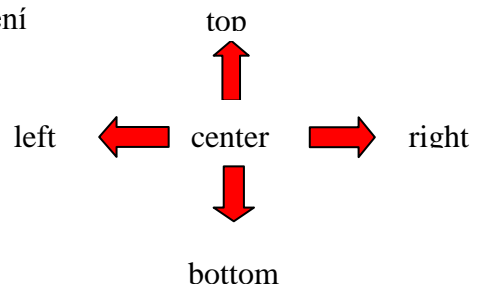
7.3 Pozadí

Vlastnosti stylů, které definují efekty pozadí pro prvky (Obr. 84).

- **background-color** – určí barvu pozadí prvku
- **background-image** – určí obrázek, který bude použitý na pozadí prvku. Obrázek se opakuje po celé ploše, dokud nepokrývá celý prvek.
- **background-repeat** – omezení počtu opakování
 - *repeat-x* - obrázek se nebude opakovat vodorovně
 - *repeat-y* – obrázek se nebude opakovat vertikálně
 - *no-repeat* - nechceme, aby se výsledný obrázek opakoval ani vodorovně ani svisle
- **background-attachment** – určí, zda bude např. obrázek rolován se zbytkem stránky nebo zůstane pevně na místě
 - *scroll* – pozadí se posouvá spolu s prvkem
 - *fixed* – pozadí zůstane pevné, i když se bude posouvat stránka
 - *local* – pozadí se posouvá spolu s obsahem elementovaným

- **background-position** – změna polohy umístění

- *left* – vlevo
- *right* - vpravo
- *top* - nahoře
- *center* - uprostřed
- *bottom* – dolů [26]



Obrázek 83: Umístění polohy

- **background-size** – určuje velikost obrázků na pozadí

- velikost lze specifikovat v délkách, procentech, pixelech
- *contain* - největší velikosti obrázku, aby jeho šířka i výška vešla dovnitř obsahu)
- *cover* - pokrytí obrázkem na pozadí, jak to jen bude možné[62]

```
<body>
  <h2>Pozadí</h2>
  <div id="priklad">
    <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.</p>
    <p>Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat.</p>
    <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.</p>
    <p>Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat.</p>
  </div>
</body>
```

```
#priklad {
  background-color:yellow; /*barva na pozadí*/
  background-image: url("../obrazky/tucnak.png");/*vlození obrázku*/
  border: 1px solid black; /*orámování div*/
  background-size: 100px 80px; /*velikost obrázku*/
  background-repeat: no-repeat; /*žádné opakování obrázku*/
  background-position: left top; /*umístění obrázku vlevo nahoře*/
  background-attachment:local; /*obrázek se posouvá s obsahem*/
}
```

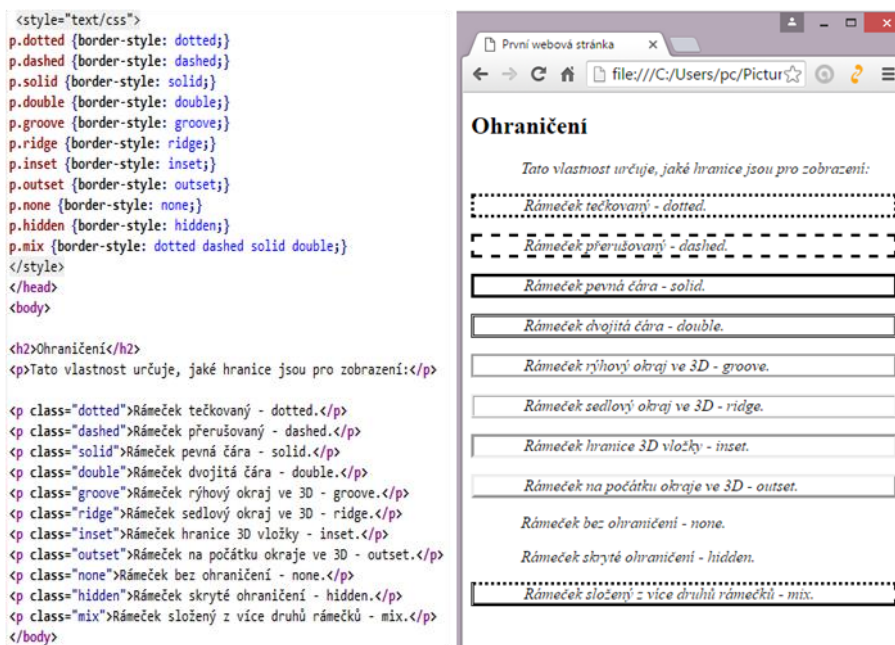


Obrázek 84: Ukázka vložení pořadí

7.4 Ohraničení

Ohraničení umožní označit stránku nebo text různým základním ohraničením (Obr. 85, 86). Lze vybrat na každou stranu jiné označení, píšeme je za sebou a oddělujeme mezerou. [26]

- **dotted** - tečkování
- **dashed** - přerušované
- **solid** – pevná čára
- **double** – dvojitá čára
- **groove** – rýhový okraj 3D
- **ridge** – sedlový okraj 3D
- **inset** – hranice 3D vložky
- **outset** - na počátku okraje 3D
- **none** – bez ohraničení
- **hidden** – skryté ohraničení
- **mix** – více ohraničení v jednom rámečku



Obrázek 85: Ukázka ohraničení

- **Šířka rámečku**

Vlastnost **border-width** určuje šířku čtyř hranic. Šířku lze nastavit v určitých velikostech (px, cm, pt, em atd.) nebo předem definovaných hodnot

- tenká (*thin*),
- střední (*medium*)
- silná (*thick*)

Border-with můžeme nastavit od jedné do čtyř hodnot (pro horní hranici, pravý okraj, spodní hranici a levý okraj). [26]

- **Barva rámečku**

Nastavíme ji pomocí vlastnosti **border-color** od jedné až do čtyř hodnot jak u šířky rámečku. Barvu přidáváme pomocí stanovených hodnot název, hexadecimální, RGB a průhledný (transparent). [26]

- **Jednotlivé strany**

Uvádí přesnější hranici pro každou stranu.

- *border-top-style* – horní hranice
- *border-right-style* – pravý okraj
- *border-bottom-style* – spodní hranice
- *border-left-style* – levý okraj [26]

- **Obrázek v ohrazení**

K ohrazení nám poslouží i správně zvolený obrázek, který bude využit k ohrazení stránky nebo část textu.

- **Border-image** – určení vlastnosti obrázku
 - **Border-image-source** – určuje se cesta k obrázku, který má být použitý (např. `border-image.source: url(border.png);`)
 - **Border-image-slice** – specifikuje, jak se zkrátí výsledný obraz
 - *Fill* – zobrazí uvnitř rámečku obrázek (např. `30px fill`) [62]
 - **Border-image-width** – specifikuje šířku okrajů obrazu v procentech (%), číslo (1), pixely (5px)
 - **Border-image-outset** – určuje oblast hranice, o kterou přesahuje hranici. Určujeme od jedné do čtyř hodnot (nahore, vpravo, dolů, zleva) a nebude-li určena jedna z hodnot, bude nahrazena, již uvedenou hodnotou (např. `5px 15px` jsou určeny pro hodnotu nahore a vpravo, pro hodnotu dolů a zleva se opakuje to stejné, výsledek bude `5px nahore, 15px vpravo, 5px dolů, 15px zleva`). [62]
 - **Border-image-repeat**
 - * *stretch* – obraz je protažený, aby vyplnil prostor
 - * *repeat* – obraz se vedle sebe opakuje, aby vyplnil prostor
 - * *round* – obraz se opakuje tak, že se přepočte obrazec, aby se vešla na celou plochu

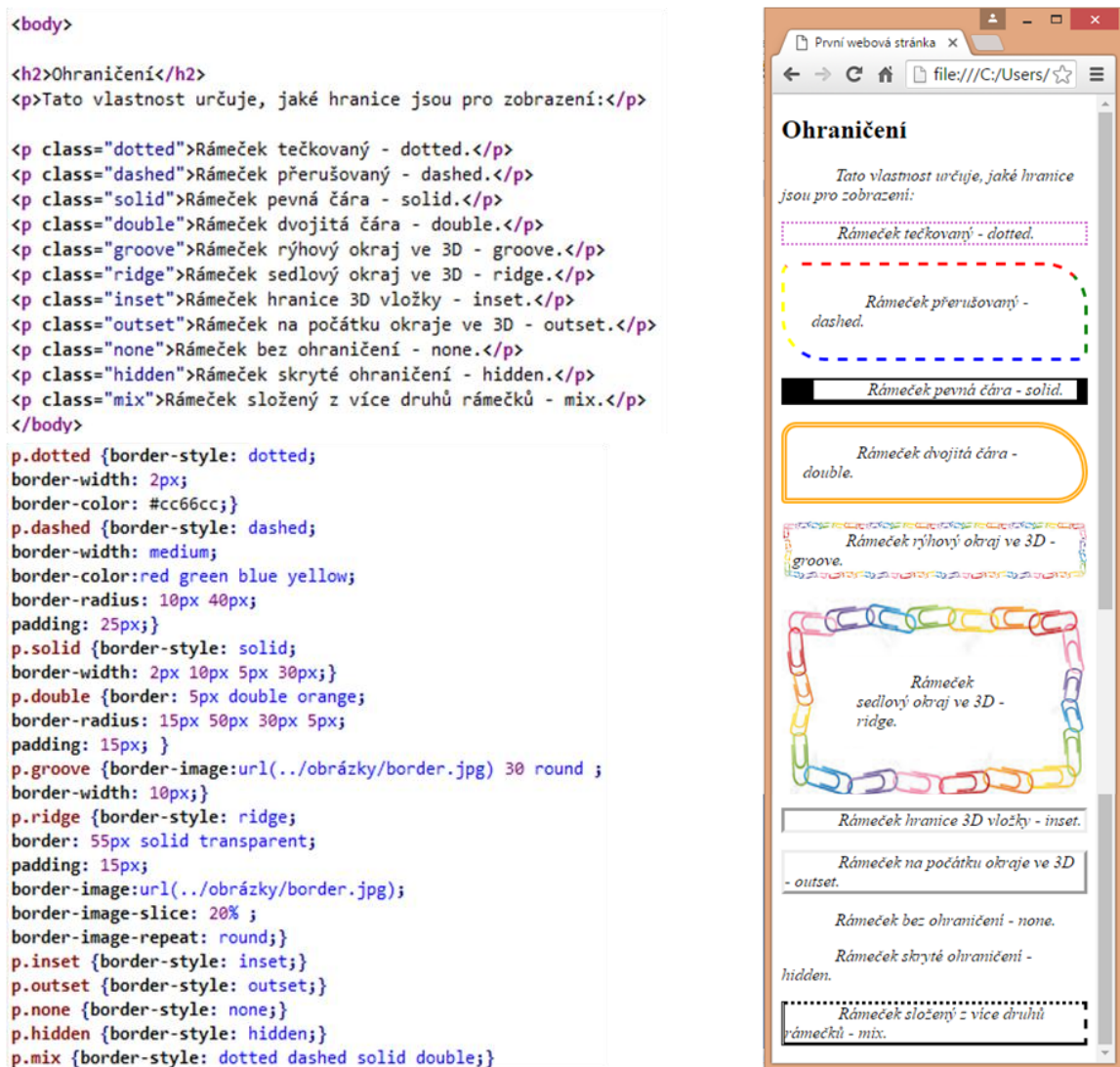
* *space* – obraz se opakuje tak, že prostor je rozdělen kolem dlaždice [62]

- **Zaoblení rohů**

Pro zaoblení rohů rámečku využijeme prvek `border-radius`. Můžeme uvést hodnotu pro každý roh. Pokud není uvedena hodnota pro některý z rohů, bude se opakovat již ze zadaných hodnot.

- **Border-radius**

- *border-top-left-radius* – zaoblení jen horního levého rohu
- *border-top-right-radius* – zaoblení horního pravého rohu
- *border-bottom-right-radius* – zaoblení dolního pravého rohu
- *border-bottom-left-radius* – zaoblení dolního levého rohu [62]



Obrázek 86: Ukázka ohraničení textu

7.5 Okraje

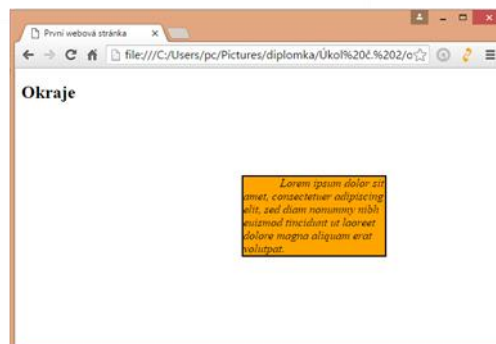
Margin slouží ke generování prostor kolem prvků. Nastavit lze velikost prázdného prostoru mimo hranice (Obr. 85). Můžeme nastavit pro každou stranu rozpětí prvku.

- **Margin** – nastavíme všechny vlastnosti do jedné hodnoty, v jedné vlastnosti můžou být od jedné do čtyř hodnot
 - **margin**: 15px 5px 10px 15px;
 - Horní okraj je 15px
 - Pravý okraj je 5px
 - Spodní okraj je 10px
 - Levý okraj je 15px
 - *Auto* - horizontální vycentrování prvku
 - *Inherit* – zdědí vlastnosti z hlavního elementu
- **Margin-top** – nastavení horního okraje
- **Margin-right** – nastavení pravého okraje
- **Margin-bottom** – nastavení dolního okraje
- **Margin-left** – nastavení levého okraje [26] [66]

```
<body>
  <h2>Okraje</h2>

  <p class="priklad">Lorem ipsum dolor sit amet, consectetur adipiscing
    elit, sed diam nonummy nibh euismod tincidunt ut
    laoreet dolore magna aliquam erat volutpat.</p>
</body>
</html>
```

```
p.priklad {
background-color:orange;
border:2px solid;
Margin-top: 100px;
margin-right: 150px;
margin-bottom: 100px;
margin-left: 300px;
}
```

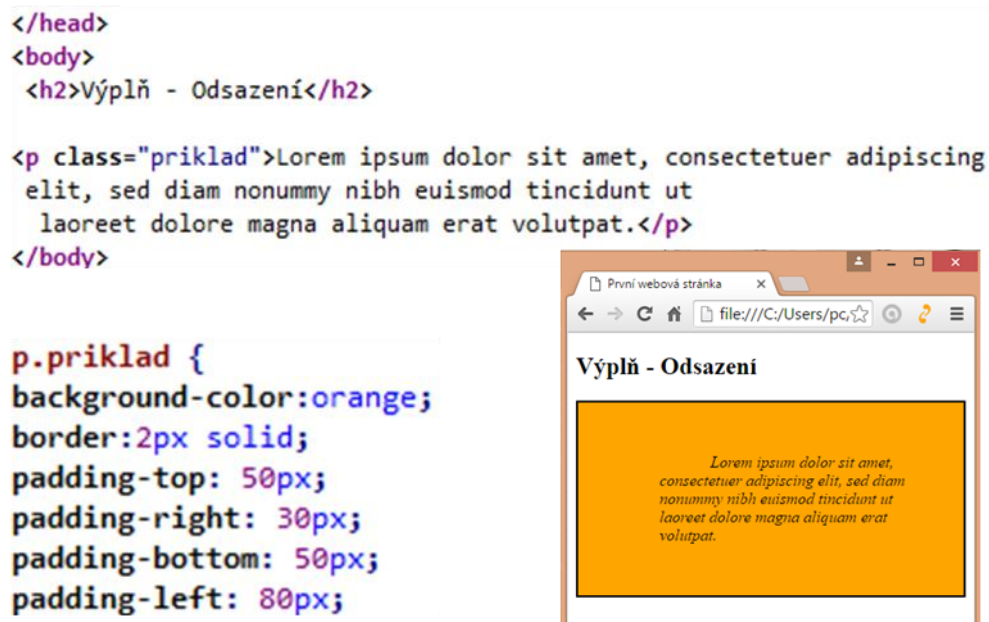


Obrázek 87: Ukázka vložení odkazů.

7.6 Výplň

Padding nastavuje výplň prostoru kolem obsahu. Nastavíme tímto prvkem prázdný prostor mezi obsahem textu a hranicí prvku. Využíváme procento, centimetry a pixely.

- **Padding** – odsadíme text v nastavených hodnotách, můžeme nastavit u všech stran najednou (např. padding: 1cm 2cm 3cm 1cm;)
 - *padding-top* – odsazení nahoře
 - *padding-right* – odsazení vpravo
 - *padding-bottom* – odsazení dolů
 - *padding-left* – odsazení vlevo [26] [66]



Obrázek 88: Ukázka výplně.

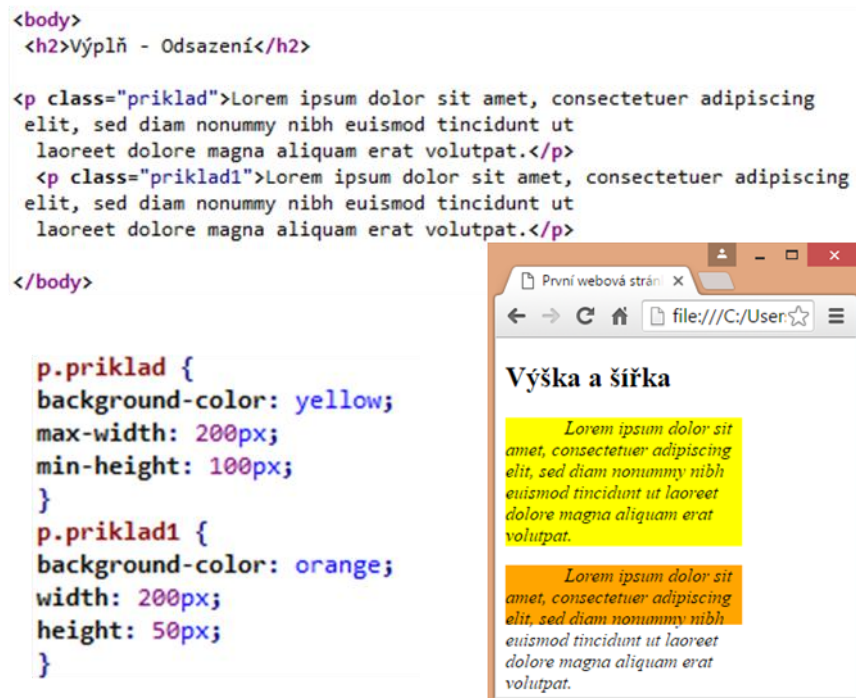
7.7 Výška a šířka

K nastavení **výšky** (*height*) a **šířky** (*width*) jsou používány tyto vlastnosti. Pro stanovení hodnot délky jsou opět využívány centimetry (cm), pixely (px) nebo procento (%) se zarovnáním do bloku apod. Výšku a šířku lze nastavit jako na automatickou (*auto*) hodnotu, tím říkáme, že prohlížeč si sám vypočítá výšku a šířku. [26]

Tyto vlastnosti se používají k nastavení maximální šířky a výšky. Pokud bude okno prohlížeče menší než nastavená maximální výška, prohlížeč přidá vodorovný pokusník na

stránku. Při nastavení výchozí hodnoty *none* se neprojeví žádná maximální šířka a výška (Obr. 89).

- **Max-width** – maximální šířka
- **Min-width** – minimální šířka
- **Max-height** – maximální výška
- **Min-height** – minimální výška [26]



Obrázek 89: Ukázka výšky a šířky.

7.8 Obtékání

Plovoucí umístění změní webové stránky a velmi snadno vytvoří přesně, co chceme.

Obtékání vyřeší:

- **Úpravu lokálního textu**
- **Uspořádání stránky (globální layout)**

Uspořádání řešíme úpravou obtékání pro nějaký obrázek, aby přesně seděl v textu. Rozmístí celé bloky bez využití dalších úprav, jako například do tabulek či pozice. Obtékavý (plovoucí) objekt musí mít většinou nastavenou šířku. Pokud nebude jinak uvedeno, kdy výchozí hodnota bude mít šířku 100% a okolní text nemá kam téci. [25] [30] [63]

Obtékání prvku určuje vlastnost **float** v CSS a nabízí tyto hodnoty:

- *None* – prvek nebude obtékán
- *Right* – prvek bude umístěn k levému okraji a další přiřazené elementy ho budou obtékat zprava
- *Left* - prvek bude umístěn k pravému okraji a další přiřazené elementy ho budou obtékat zleva [25] [30] [63] [64]

V případě, že dva objekty budeme mít obtékané blízko sebe, mají tendenci se skládat vedle sebe. A my budeme chtít, aby se neobtékali vedle sebe, ale až pod jeho koncem.

K tomu je potřeba vlastnost `clear` v CSS, které ukončí jiné obtékání prvku.

- *None* – nečeká se na plovoucí prvky
- *Both* – prvek se zobrazí až pod všemi plovoucími prvky
- *Left* – element nebude obtékat žádný plovoucí prvek umístěný doleva
- *Right* - element nebude obtékat žádný plovoucí prvek umístěný doprava [25] [63] [67]

7.9 Zobrazení

Rozlišujeme druhy elementu v jazyku HTML.

- **Řádkové**
 - Jde o vyobrazení elementů, které se zobrazují v řádku
 - Skládají se jeden za druhým a nelze jim přidělit velikost
 - Například pro element v HTML `<a>`
- **Blokové**
 - Jde o vyobrazení elementů, které se zobrazují v bloku
 - Skládají se jeden pod druhým a lze jim přidělit velikost
 - Například pro element v HTML `<p>`, `<div>` nebo nadpisy

Vlastnost `display` umí změnit celou povahu elementu, které mají následující hodnoty:

- **Display**
 - *Block* – zobrazení prvku jako blok, tzn. řádkový zlomen na konci
 - *Inline* – zobrazení prvku jako řádkový, tzn. bez řádkového zlomu na konci
 - *List-item* – zobrazení prvku jako položka v seznamu (``)
 - *None* – žádné vyobrazení prvku na stránce [25] [68]

7.10 Text

Velmi důležitý odstavec pro formátování textu do různé stylizace a vlastností textu (Obr. 90).

Již něco málo jsme si řekli v HTML. K zarovnání textu využíváme:

- **text-align**
 - *center* – text zarovnan na střed
 - *left* - text zarovnan vlevo
 - *right* – text zarovnan vpravo
 - *justify* – text zarovnan do bloku
- **text-decoration** – využití k nastavení nebo odebrání označených částí textu
 - *none* – odstraní podtržení z odkazů
 - *overline* – nad textem se vytvoří souvislá čára
 - *line-through* – přeškrtnutý text
 - *underline* – podtržené pod textem [26]
- **text-transform** – využívá se k nastavení velkých a malých písmen v textu
 - *uppercase* – všechny text bude napsán velkými písmeny
 - *lowercase* – všechny text bude napsán malými písmeny
 - *capitalize* – každé začáteční písmeno slova bude velkým písmenem
- **text-indent** – odsazení prvního řádku textu[26]
- **letter-spacing** – určíme, jak velkou mezeru budeme mít mezi textem, můžeme jí i do záporných hodnot
- **line-height** – určíme, jak velká mezera bude mezi řádky
- **direction** – určí změnu směru textu psaní
 - *right* – z prava do leva, zkratka rtl
 - *left* – z leva do prava, zkratka ltr, nastavena jako výchozí
- **word-spacing** – roztahuje nebo zužuje mezery mezi písmeny, hodnota může být i záporná [26]
- **text-overflow** – určí vlastnost obsahu, který není zobrazen
 - *clip* – oříznutý text, výchozí hodnota
 - *ellipsis* – vyjádřeno konec textu (...) [62]
- **word-wrap** – zalamování textu
 - *normal* – výchozí nastavení,

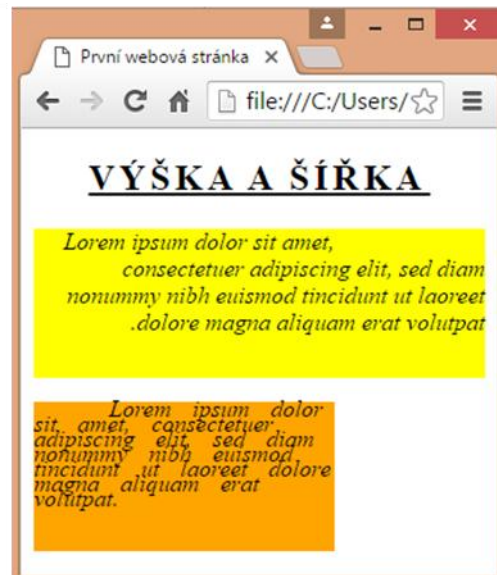
- *break-word* – vynutí si zalomení textu, rozdělí slovo i uprostřed
- **word-break**
 - *keep-all* – text bude zalomen libovolně mezi dvěma písmeny
 - *break-all* – text bude rozdělen mezi dvojicemi písmen [62]

```
<body>
  <h2>Výška a šířka</h2>

  <p class="priklad">Lorem ipsum dolor sit amet, consectetur adipiscing
elit, sed diam nonummy nibh euismod tincidunt ut
laoreet dolore magna aliquam erat volutpat.</p>
  <p class="priklad1">Lorem ipsum dolor sit amet, consectetur adipisci
elit, sed diam nonummy nibh euismod tincidunt ut
laoreet dolore magna aliquam erat volutpat.</p>

</body>
```

```
h2 {
  text-align: center;
  text-decoration: underline;
  Text-transform: uppercase;
  letter-spacing: 3px; }
p.priklad {
  background-color: yellow;
  max-width: 300px;
  min-height: 100px;
  text-indent: 100px;
  direction: rtl;
}
p.priklad1 {
  background-color: orange;
  width: 200px;
  height: 100px;
  line-height: 10px;
  word-spacing: 10px;
}
```



Obrázek 90: Ukázka úpravy textu

7.11 Font písma

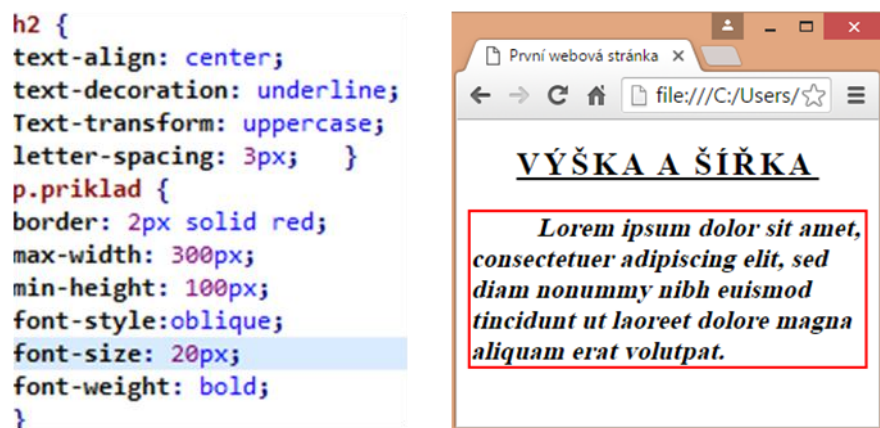
Pro font písma využijeme značku (Obr. 91) :

- **font-family**

Můžeme za sebou napsat několik fontů, pro případ že nebude některý z prohlížečů podporovat první písmo a pokusí se o přečtení dalších. Musíme písmo mezi sebou oddělit čárkou. Pro delší název než jedno slovo musíme písmo uzavřít do uvozovek (např. "Times New Roman "). [90]

- **font-style**

- *normal* – normální styl písma
- *italic* - kurzíva
- *oblique* - nakloněný styl písma jako kurzíva
- **font-weight** - pro nastavení tučného písma
 - *bold* – tužné písmo
- **font-size** - nastavení velikosti písma, pokud neurčíme velikost je nastaveno výchozí nastavení pro text je 16px (1em). [26]

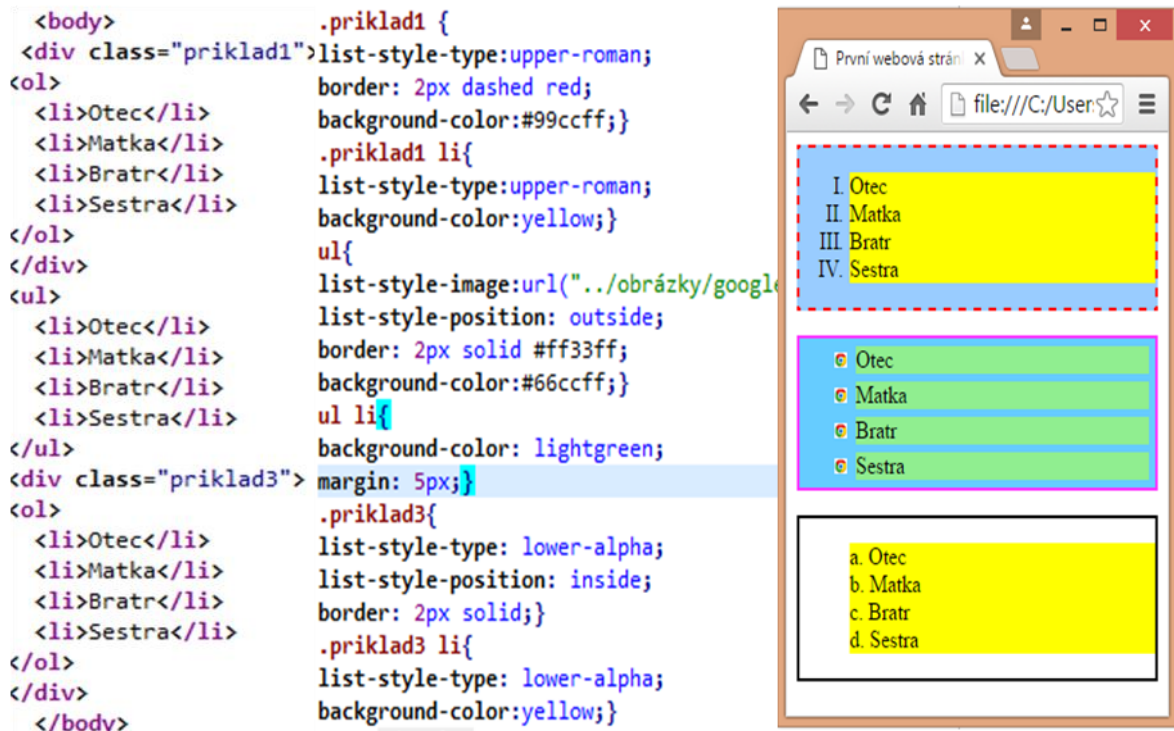


Obrázek 91: Ukázka stylu písma

7.12 Seznamy

Podstatu formulářů jsme si objasnili již v HTML, jaké mají rozdělení a využití. V CSS si popíšeme, co všechno nám umožní dále upravovat (Obr. 92).

- **list-style** – zkrácená verze pro nastavení všech vlastností v jednom
- **list-style-type** –určení typu odrážky
 - *circle* – odrážka ve tvaru kruhu
 - *square* – odrážka ve tvaru krychle
 - *upper-roman* – číslování římskými čísly
 - *lower-alpha* - číslování malým písmen od začátku abecedy
- **list-style-image** – pro značku můžeme vybrat obrázek, který bude vyjádřen jako odrážka
 - *url* – cesta k obrázku (např. `list-style-image: url('obrazek.gif');`)
- **list-style-position** – určení toku obsahu
 - *inside* – uvnitř toku
 - *outside* – vnější tok [26]



Obrázek 92: Ukázka seznamů.

7.13 Stíny

V nové verzi CSS3 se umožňuje přidat vržení stínů k textu, jako jsou například nadpisy nebo odstavce. Lze také přidat stíny k rámečkům nebo obrázkům (Obr. 93).

- **text-shadow** – určený pro písmo, v některých prohlížečích nutno použití testovací vlastnosti (viz kapitola 3.6), doplňuje k textu vržené stíny ve čtyřech hodnotách, které v tomto pořadí zapisujeme:
 - *posun* – x
 - *posun* – y
 - *poloměr rozostření*
 - *barva stínu*[62]
- **box-shadow** – určený pro rámečky a obrázky, v některých prohlížečích nutno použití testovací vlastnosti (viz kapitola 3.6). Doplňuje k textu vržené stíny v šesti hodnotách, které v tomto pořadí zapisujeme:
 - *posun* – x
 - *posun* – y
 - *poloměr rozostření*
 - *rozsah*

- *barva stínu*
- *inset – uvnitř* [62]

```

<h2>Stínování</h2>
<p>Tato vlastnost určuje, přidání stínů.</p>

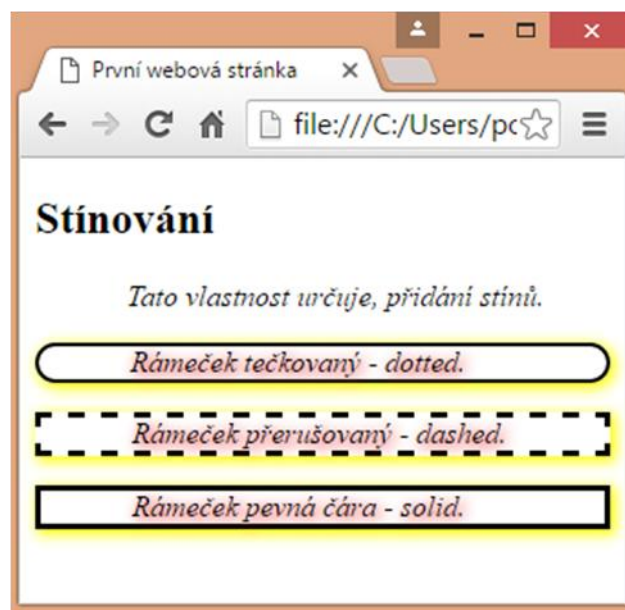
<p class="dotted">Rámeček tečkovaný - dotted.</p>
<p class="dashed">Rámeček přerušovaný - dashed.</p>
<p class="solid">Rámeček pevná čára - solid.</p>

```

```

div * {
text-shadow: 2px 2px 10px red;
box-shadow: 2px 2px 10px 1px yellow;
}
p.dotted {
border-radius: 25px;
border: 2px solid black;
}
p.dashed {border-style: dashed;}
p.solid {border-style: solid;}

```



Obrázek 93: Ukázka stínů.

7.14 Sloupce

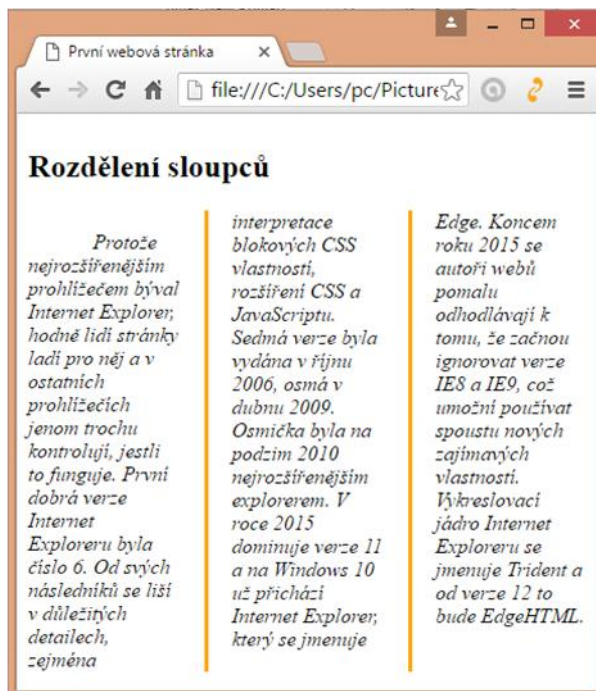
Umožňuje snadné rozložení textu na sloupce a jeho následné úpravy (Obr. 94). Nesmíme zapomenout na testovací vlastnost (viz 3.6), která umožní zobrazení v různých prohlížečích.

- **column-count** - určíme si, na kolik sloupců budeme chtít rozdělit text
- **column-gap** – určíme mezeru mezi sloupci
- **column-rule** – zápis všech column-rule do jednoho řádku v následujícím pořadí
 - *column-rule-width* – určíme šířku mezi sloupci (tloušťka plné čáry)
 - *column-rule-style* – určíme si styl mezi sloupci (např. plná čára solid)
 - *column-rule-color* – určíme barvu mezi sloupci
- **column-span** – určíme například, že nadpis by se měl zobrazit přes všechny sloupce

- **column-width** – určíme šířku sloupce. [62]

```
<body>
<div class="sloupec">
<h2>Rozdělení sloupců</h2>
<p>Protože nejrozšířenějším prohlížečem býval Internet Explorer,
První dobrá verze Internet Exploreru byla číslo 6. Od svých nás:
Koncem roku 2015 se autoři webů pomalu odhodlávají k tomu, že za
Vykreslovací jádro Internet Exploreru se jmenuje Trident a od ve
</div>
</body>
```

```
div {
-webkit-column-count: 3; /* Chrome, Opera, Safari*/
-webkit-column-gap: 40px;
-webkit-column-rule-style: solid;
-webkit-column-rule-width: 3px;
-webkit-column-rule-color: orange;
-webkit-column-width: 100px; }
h2 {
-webkit-column-span: all; }
```

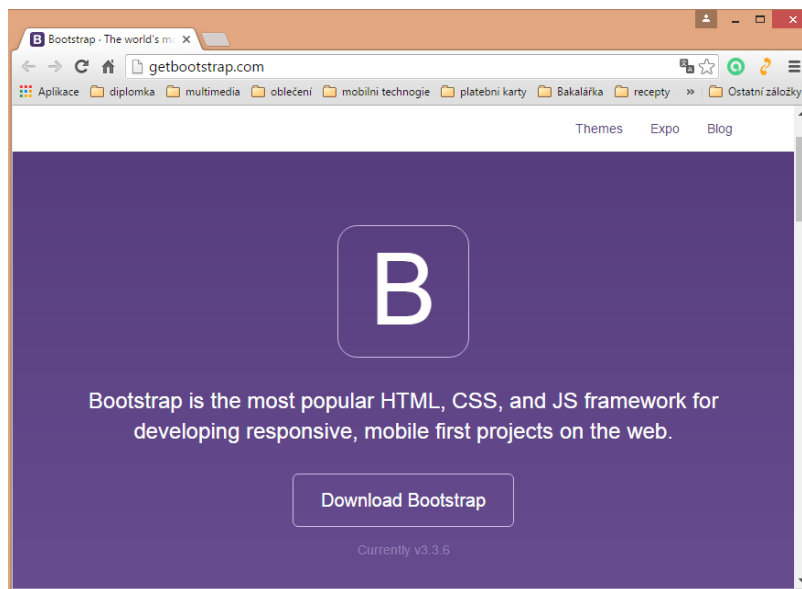


Obrázek 94: Ukázka vložení sloupců.

8 BOOTSTRAP

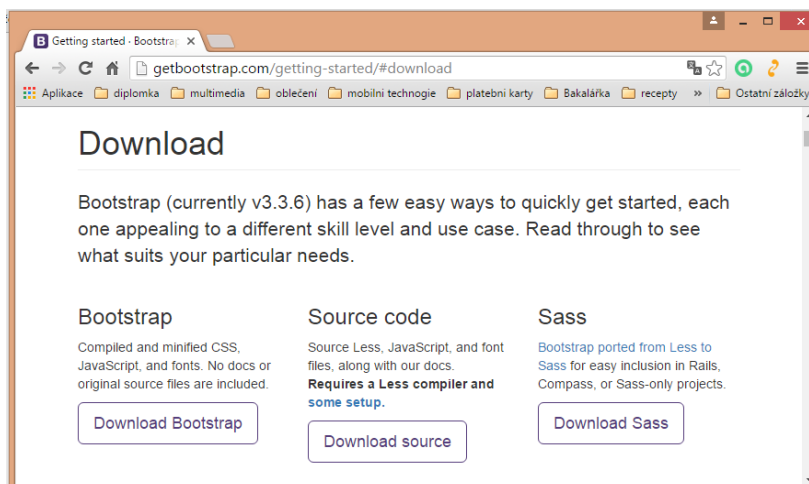
1. Stažení šablony

K využívání bootstrap je potřeba si stáhnout framework z internetového odkazu <http://getbootstrap.com/> (Obr. 95).



Obrázek 95: Ukázka internetového prohlížeče Bootstrap.

Klikneme na tlačítko Download Bootstrap a stránka bude přeměrována na jinou, kde máme na výběr ze tří možností (Obr. 96). Pro další práci budeme používat první variantu Bootstrap. [70]



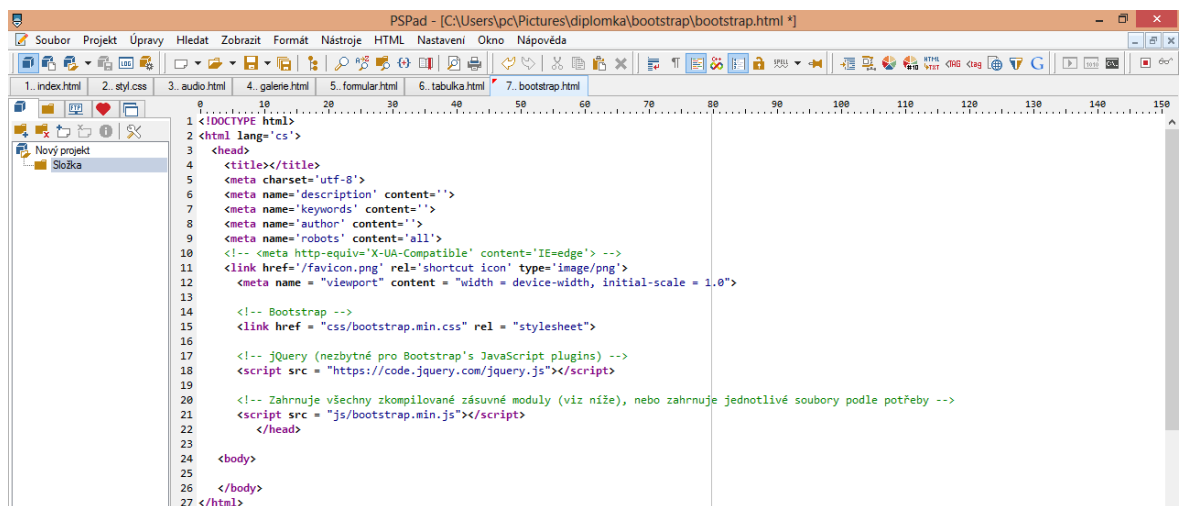
Obrázek 96: Ukázka výběru možností ke stažení.

- **Bootstrap** – stažením získáme verzi Bootstrap CSS, JavaScript a písmo. Žádná dokumentace nebo původní zdrojový kód nejsou zahrnuty.

- **Source code** – stažením získáme nejnovější Bootstrap méně a JavaScript zdrojový kód a písmo spolu s dokumenty.
- **Sass** – snadné začlenění do Rails, kompas, pouze pro projekty. [70]

2. Vložení stylů

Po stažení zazipovaného balíčku rozbalíme celý obsah do prázdné složky. Otevřeme si program PSPad a vytvoříme nový dokument HTML5 (Obr. 97). [70]



Obrázek 97: Vložení stylu do hlavičky dokumentu HTML.

Do hlavičky byla vložena meta značka, aby bylo zajištěno řádné vykreslení na mobilním zařízení (Obr. 98).

```
<meta name = "viewport" content = "width = device-width, initial-scale = 1.0">
```

Obrázek 98: Vložení meta značky do hlavičky dokumentu HTML.

- **Widht (šířka)** – určuje šířku zařízení. Ujistíme se, zda budou zobrazeny různými zařízeními (mobilní telefony, stolní počítače, tablety apod.)
- **Initial-scale** – 1.0 zajišťuje, že webová stránka při zatížení bude vykreslena v měřítku 1:1

Byly přidány odkazy, které směřují k rozbalenému balíčku. Odkazují na soubory uvnitř složek (Obr. 99). JQuery je internetový odkaz pro využití prvků. [70]

```

<!-- Bootstrap -->
<link href = "css/bootstrap.min.css" rel = "stylesheet">

<!-- jQuery (nezbytné pro Bootstrap's JavaScript plugins) -->
<script src = "https://code.jquery.com/jquery.js"></script>

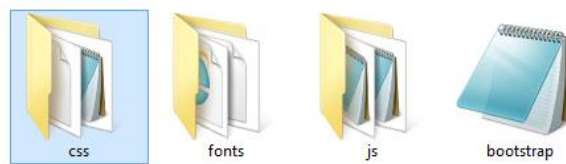
<!-- Zahrnuje všechny zkompileované zásuvné moduly (viz níže), nebo zahrnuje jednotlivé soubory podle potřeby -->
<script src = "js/bootstrap.min.js"></script>

```

Obrázek 99: Vložení odkazů na knihovny.

3. Uložení

Vložíme následující strukturu, kterou uložíme do stejné složky, kde jsme rozbalili balíček Bootstrap (Obr. 100).



Obrázek 100: Ukázka uspořádání ve složce.

8.1 Základní struktura Bootstrap Grid

Uvedeme si základní strukturu mřížky (Obr. 101).

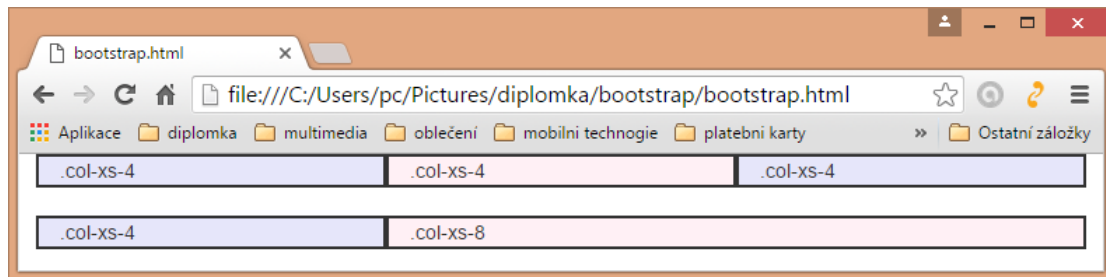
```
<body>
<div class = "container">
  <div class = "row" >
    <div class="col-*-*"></div>
  </div>
  <div class="row">
    <div class="col-*-*"></div>
    <div class="col-*-*"></div>
    <div class="col-*-*"></div>
  </div>
</div>
</body>
```

Obrázek 101: Základní struktura Bootstrap Grid.

Tři stejné šířky sloupců, které začínají od mobilních telefonů až po velké plochy. Dva sloupce šířky ukazují pro tablety až po velké plochy (Obr. 102, 103). [71]

```
<body>
<div class = "container">
  <div class = "row" >
    <div class="col-xs-4" style="background-color:lavender; border: 2px solid;".col-xs-4</div>
    <div class="col-xs-4" style="background-color:lavenderblush; border: 2px solid;".col-xs-4</div>
    <div class="col-xs-4" style="background-color:lavender; border: 2px solid;".col-xs-4</div>
  </div><br>
  <div class="row">
    <div class="col-sm-4" style="background-color:lavender; border:2px solid;".col-sm-4</div>
    <div class="col-sm-8" style="background-color:lavenderblush; border: 2px solid;".col-sm-8</div>
  </div>
</div>
</body>
```

Obrázek 102: Ukázka sloupců v Bootstrap.



Obrázek 103: Ukázka zobrazení sloupců v prohlížeči.

8.2 Typografie

Uvedené prvky v tabulce jsou prvky HTML, které jsou navrženy pro bootstrap (Tab. 13).

[70]

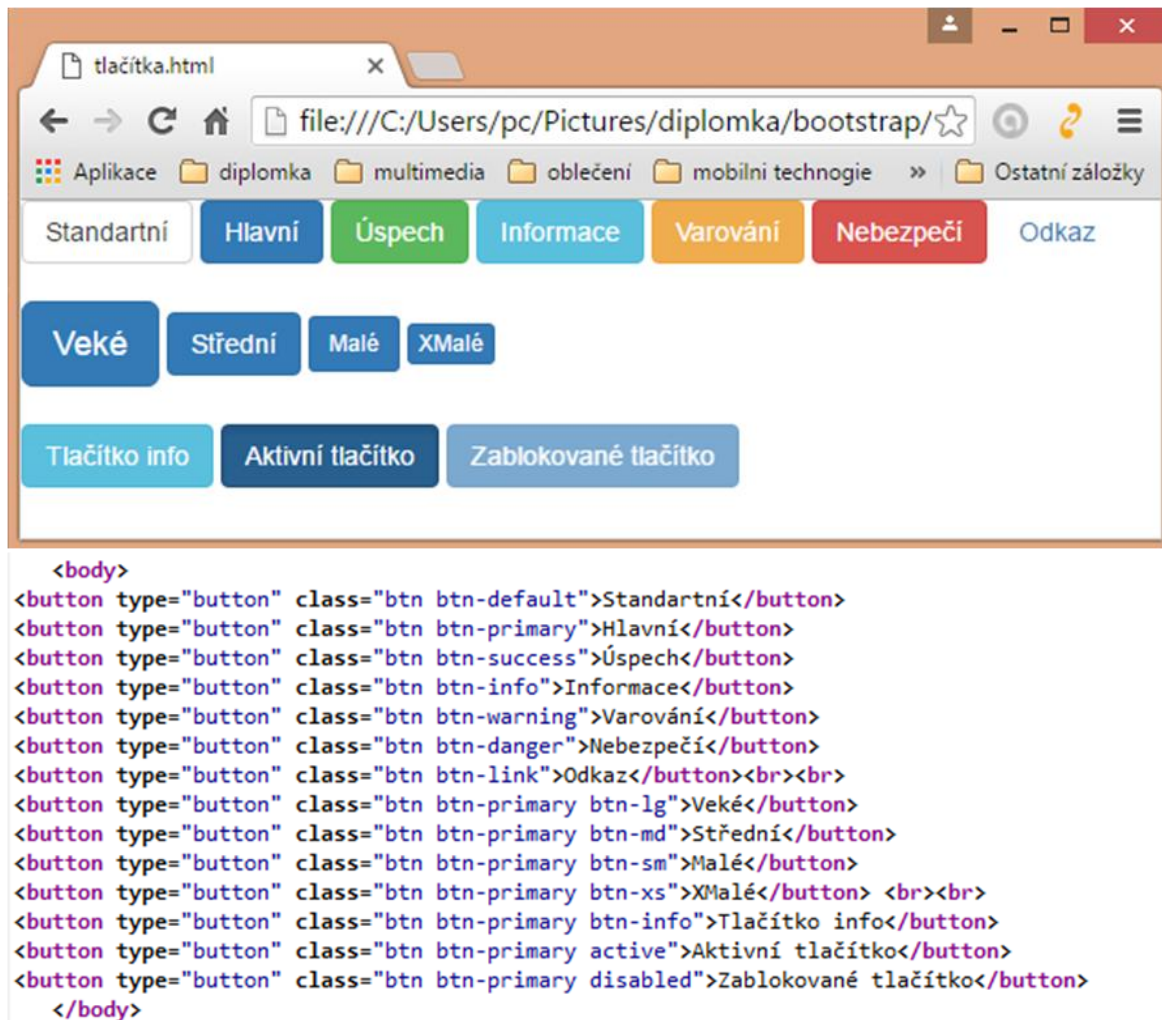
Atribut/třída	Popis
<code><h1></code> až <code><h6></code> <code>.h1</code> až <code>.h6</code>	Úrovně nadpisů.
<code><small></code> <code>.small</code>	Vytvoření světlejší varianty textu (např. Nadpis 36px). Označení menšího textu, nastaven na 85% velikosti.
<code>.lead</code>	Zvýraznění textu, podobné k tučnému písmu.
<code><mark></code> nebo <code>.mark</code>	Zvýraznění textu (např. Nadpis).
<code></code>	Přeškrtnutí textu (např. Nadpis).
<code><s></code>	Přeškrtnutí nepotřebného textu (např. Nadpis).
<code><ins></code>	Označení vloženého textu (např. <u>Nadpis</u>).
<code><u></code>	Podtržení textu (např. <u>Nadpis</u>).
<code></code>	Zvýraznění textu tučným písmem (např. Nadpis).
<code></code>	Zvýraznění textu kurzívou (např. <i>Nadpis</i>).
<code>.text-left</code>	Zarovnání textu doleva.
<code>.text-center</code>	Zarovnání textu na střed.
<code>.text-right</code>	Zarovnání textu doprava.

<i>.text-justify</i>	Zarovnání textu do bloku.
<i>.text-nowrap</i>	Žádné obtékání textu.
<i>.text-lowercase</i>	Označení malým písmem.
<i>.text-uppercase</i>	Označení velkým písmem.
<i>.text-capitalize</i>	Označení každé počáteční písmeno ve slově.
<i>.text-muted</i>	Označuje se šedým textem.
<i>.text-primary</i>	Označení základní barvou modrou.
<i>.text-success</i>	Označení textu barvou zelenou.
<i>.text-info</i>	Označení textu tmavě modrou.
<i>.text-warning</i>	Označení textu oranžovou barvou.
<i>.text-danger</i>	Označení textu červenou barvou.
<code><abbr></code>	Potržení zkratky tečkováním, přejetí myši na označené slovo se nám objeví celá zkratka (např. <code><abbr title="World Health Organization">WHO</abbr></code>).
<i>.initialism</i>	Přiřazení k značce <code><abbr ></code> , která výsledný text zmenší písmo.
<code><address></code>	Kontaktní informace (adresa).
<code><blockquote></code>	Označuje blok textu.
<i>.blockquote-reverse</i>	Označuje blok citace s obsahem zarovnaným doprava.
<code></code>	Označení neuspořádaného seznamu.
<code></code>	Označení uspořádaného seznamu.
<i>.list-unstyled</i>	Odstraní odrážky seznamu, platné pro <code>ul</code> a <code>ol</code> .
<i>.list-inline</i>	Zobrazení seznamu v jedné linii (horizontálně)
<code><dl></code>	Označení popisku, další úroveň <code><dt></code> a <code><dd></code> .
<i>.dl-horizontal</i>	Název a popis jsou v jedné linii, ne pod sebou (horizontálně)

Tabulka 13: Ukázka typografie Bootstrap.

8.3 Tlačítka

V tabulce si popíšeme třídy, které se přidávají ke značce `<button>` (Tab. 14), která označuje tlačítko (Obr. 104). [70]



Obrázek 104: Ukázka vložení tlačítek.

Třída	Popis
<code>.btn</code>	Základní přidání do class, která vyznačuje tlačítko, žádná zobrazení.
<code>.btn-default</code>	Označuje základní tlačítko.
<code>.btn-primary</code>	Označuje hlavní tlačítko, pro použití.
<code>.btn-success</code>	Označení tlačítka úspěch a pozitivní akce
<code>.btn-info</code>	Označení pro informační a emailové zprávy.

<code>.btn-warning</code>	Označení tlačítka jako varování.
<code>.btn-danger</code>	Označení tlačítka jako nebezpečné.
<code>.btn-link</code>	Označení tlačítka jako odkaz.
<code>.btn-lg</code>	Velké tlačítko
<code>.btn-sm</code>	Střední tlačítko
<code>.btn-xs</code>	Malé tlačítko
<code>.btn-block</code>	X malé tlačítko.
<code>.active</code>	Aktivní kliknutí na tlačítko.
<code>.disabled</code>	Zablokované kliknutí na tlačítko.

Tabulka 14: Ukázka vložení atributů tlačítka.

8.4 Formuláře

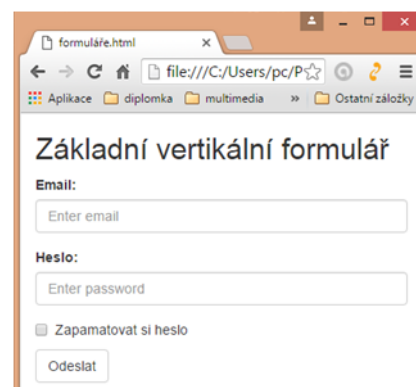
Sestavení formulářů podle pravidel bootstrap (Obr. 106).

- Přiřadit role „formu“ do `<form>`, napomáhá zpřístupnit pro lidi, kteří používají čtečky)
- Přidat ovládací prvky formuláře `<div>` prvek a `class="form-skupina"` (nutné pro optimální rozestup), (Tab. 15).
- Přidat `class="form-control"` do všech prvků `<input>`, `<textarea>` a `<select>`. [70]

```

<body>
<div class="container">
<h2>Základní vertikální formulář</h2>
<form role="form">
<div class="form-group">
<label for="email">Email:</label>
<input type="email" class="form-control" id="email" placeholder="Enter email">
</div>
<div class="form-group">
<label for="pwd">Heslo:</label>
<input type="password" class="form-control" id="pwd" placeholder="Enter password">
</div>
<div class="checkbox">
<label><input type="checkbox"> Zapamatovat si heslo</label>
</div>
<button type="submit" class="btn btn-default">Odeslat</button>
</form>
</div>
</body>

```



Obrázek 105: Ukázka vložení formuláře.

Třída	Popis
<i>.form-inline</i>	Jednotlivé prvky se řadí vedle sebe. <code><form class="form-inline" role="form"></code>
<i>.form-horizontal</i>	Jednotlivé prvky se řadí pod sebou. <code><form class="form-horizontal" role="form"></code>

Tabulka 15: Ukázka vložení horizontálního a vertikálního formuláře.

8.5 Pozadí

Přidání pozadí barvy využívá níže uvedené třídy (Tab. 16), kde se pro použití pozadí využívá počáteční značka (.bg). [70]

Třída	Popis
<i>.bg-primary</i>	Výplň pozadí základní modrou barvou.
<i>.bg-success</i>	Výplň pozadí světle zelenou barvou.
<i>.bg-info</i>	Výplň pozadí světle modrou barvou.
<i>.bg-warning</i>	Výplň pozadí světle oranžovou barvou.
<i>.bg-dander</i>	Výplň pozadí světle černavou barvou.

Tabulka 16: Popis atributů pozadí.

8.6 Umístění

Třída	Popis
<i>.pull-left</i>	Prvek zarovnan doleva.
<i>.pull-right</i>	Prvek zarovnan doprava.
<i>.center-block</i>	Prvek bude zarovnan na stred do bloku.
<i>.clearfix</i>	Vymaze plovouci prvky
<i>.show</i>	Nuti k zobrazeni prvku
<i>.hidden</i>	Prvek zustane skryty.
<i>.sr-only</i>	Skriva prvky pro vsechna zarizeni krome cteni obrazovky.

<i>.sr-only-focusable</i>	Kombinuje se s třídou <i>.sr-only</i> znovu vkládat prvek.
<i>.text-hide</i>	Pomáhá nahradit textový obsah prvku s obrázkem na pozadí
<i>.close</i>	Ikona na zavření je umístěna vpravo ve středu.
<i>.caret</i>	Rozevírací nabídka

Tabulka 17: Popis atributů umístění.

8.7 Obrázky

The screenshot shows a web browser window with the title 'Obrázek'. It displays three examples of a cartoon book character image with different styling options: 'Zabolené rohy' (rounded corners), 'Kruh' (circle), and 'Náhled' (thumbnail). Each image has a yellow background and a link icon. Below the browser window, the HTML code is shown, demonstrating the use of Bootstrap classes like `img-rounded`, `img-circle`, and `img-thumbnail` along with `background-color:yellow;` and `class="thumbnail"`.

Obrázek 106: Ukázka vložení obrázku.

Třída	Popis
<i>.img-rounded</i>	Obrázek bude mít zaoblené rohy.
<i>.img-circle</i>	Obrázek bude zakulacen do kruhu.

<i>.img-thumbnail</i>	Obrázek bude mít náhled.
<i>.img-responsive</i>	Zvětšení obrázku, velikost uvádí nadřazený prvek.

Tabulka 18: Popis atributů obrázku.

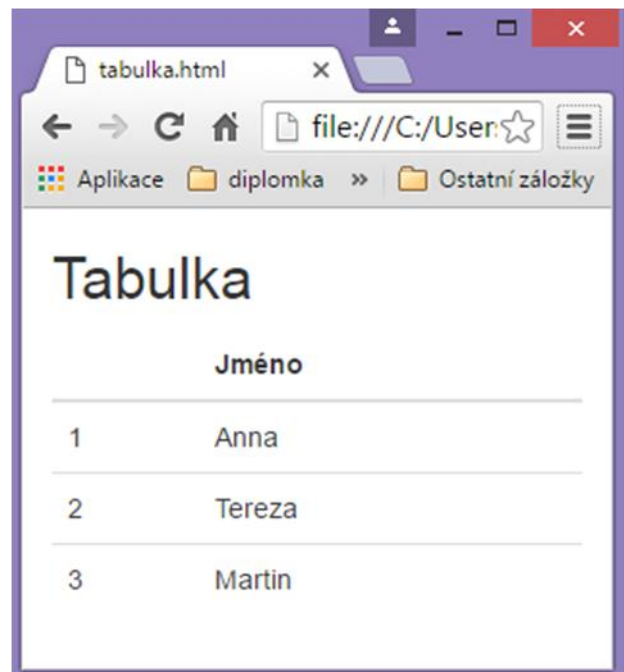
8.8 Tabulky

Tabulku vytvoříme pomocí atributu `<table>`. Zápis do hlavičky tabulky použijeme `<thead>` a ukončíme ji `</thead>`. Tělo tabulky zapíšeme `<tbody>` a `</tbody>` uvnitř těla využíváme značky (`<tr>`,`<th>`,`<td>`), jak v HTML (Obr. 107). [70]

```

<body>
<div class="container">
  <h2>Tabulka</h2>
  <table class="table table-hover">
    <thead>
      <tr>
        <th></th>
        <th>Jméno</th>
      </tr>
    </thead>
    <tbody>
      <tr>
        <td>1</td>
        <td>Anna</td>
      </tr>
      <tr>
        <td>2</td>
        <td>Tereza</td>
      </tr>
      <tr>
        <td>3</td>
        <td>Martin</td>
      </tr>
    </tbody>
  </table>
</div>
</body>

```



Obrázek 107: Ukázka zobrazení tabulky.

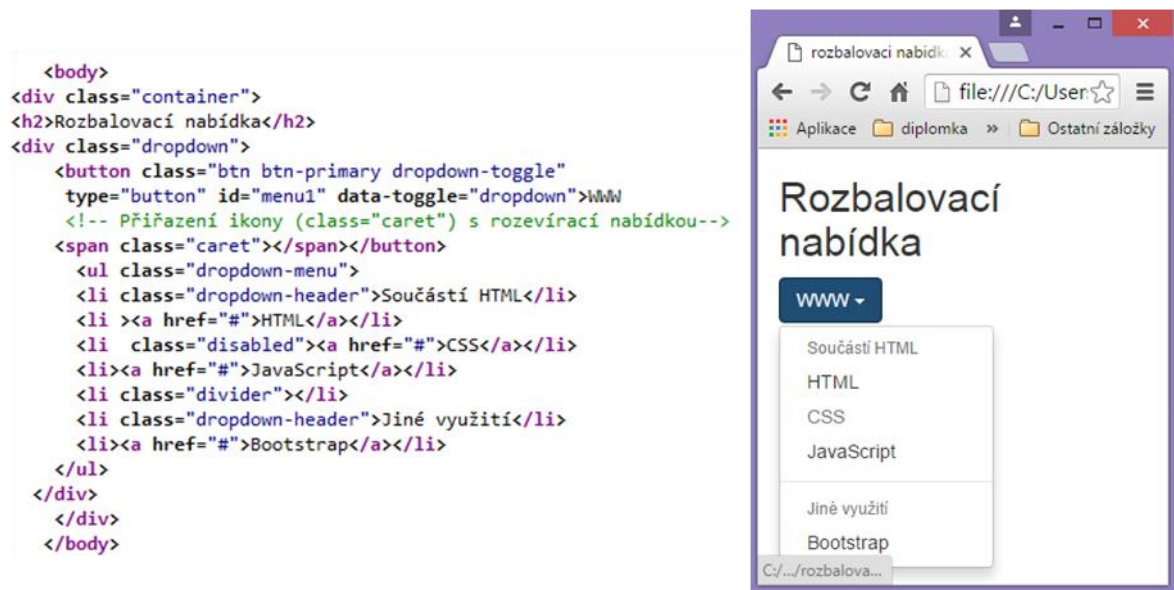
Třída	Popis
<i>.table</i>	Základní styl pro atribut <code><table></code> .
<i>.table-striper</i>	Odděluje jednotlivé řádky pruhem.
<i>.table-bordered</i>	Tabulka je ohraničena, celá kolem dokola.

<i>.table-hover</i>	Umožní větší mezery v tabulce, aby byl text v prostoru ve středu.
<i>.table-condensed</i>	Odstraní velké mezery před a za na polovinu.

Tabulka 19: Popis atributů tabulky.

8.9 Rozbalovací nabídka

Rozbalovací nabídka nabízí mnoho využití pro tvorbu webové stránky. Usnadní místo a zároveň efektivně využije místo (Obr. 108), (Tab. 20). [70]



Obrázek 108: Ukázka rozbalovací nabídky.

Třída	Popis
<i>.dropdown</i>	Označení rozbalovací nabídky
<i>.dropdown-menu</i>	Vytvoří rozbalovací nabídku.
<i>.dropdown-menu.right</i>	Rozbalovací menu bude zobrazeno vpravo.
<i>.dropdown-header</i>	Přidá do rozbalovací nabídky podnadpis označení jednotlivých nabídek.
<i>.dropup</i>	Zobrazí nabídku v horní části.
<i>.disabled</i>	Zablokování kliknutí na určitou nabídku.
<i>.divider</i>	Rozdělení nabídky linkou.

Tabulka 20: Popis atributů rozbalovací nabídky.

9 PODKLADY PRO VÝUKU

Diplomová práce je určena k výuce webových technologií na střední škole. Její součástí jsou diplomové práce vypracované prezentace k výuce, které jsou umístěny na příloženém CD-Romu.

Výuka je rozdělena na 16 týdnů. Z toho důvodu je k dispozici 16 prezentací pro výuku webových technologií. Prezentace pro výuku je rozdělena na dvě části. První část se zabývá látkou, která je obsažena v diplomové práci. Témata jsou vhodně zvolena, tak aby na sebe vhodně navazovala. Důležitým prvkem pro výběr témat bylo vytvoření návrhu webové stránky, která postupně bude realizována v hodinách. Proto výběr byl vhodně zvolen, aby byla návaznost na tvorbu webové stránky.

Druhá část se zabývá zadáním úkolu, které započne již v hodině výuky a následně bude dodělána doma. Úkolem bude předloha již vypracované a upravené výsledné stránky, kde využijí již naučenou látku k docílení stejné úpravy stránky. V následující hodině prezentace poukazuje na opakování, kde bude vysvětlena a předložena názorná ukázka postupu vložené úpravy.

Prezentace jsou pojmenovány ve formátu: cvičeníČíslo. Číslo obsahuje číslici podle probíhajících týdnů výuky. Prezentace obsahují postupy, jak pracovat v prostředí programu PSPad. Postupy jsou v textovém, ale i grafickém zobrazení.

Součástí prezentací jsou i názorné ukázky HTML stránek, kde jsou všechny změny, podle zadaného úkolu. Ve složce obrázky jsou uvedeny potřebné formáty, které jsou použity při cvičení.

Poslední vyučovací hodina je určena k zopakování zadaného úkolu z předešlé hodiny a její postup realizace. V závěru hodiny proběhne konečný test. Z toho důvodu jsou součástí práce také testovací úkoly. K dispozici budou mít 60 úkolů. Otázky budou různého typu. Budou mít na výběr ze 4 odpovědí, ale jen jedna bude správná. Dále budou mít k dispozici obrázky a do prázdného políčka napsat strukturu zápisu.

10 SMĚRY ROZVOJE UČEBNÍ POMŮCKY

Při studiu webové technologie je důležité kombinovat teoretickou a praktickou část diplomové práce. Diplomová práce zahrnuje převážnou část problematiky, která bude probírána v hodinách. Prezentace obsahují postup při tvorbě a názorné ukázky výsledného zadání úkolů. V následujících hodinách jsou prezentace doplněna o správné možné řešení zadaného úkolu. Avšak prezentace obsahují vše, co může při studiu ulehčit práci, které jsou doplněny o obrázky.

Problematika týkající se webových technologií nemohla být probrána všechna z důvodu časového a rozsahového hlediska. Doporučila bych pro následující studium tyto knihy a odkazy. **Tvorba WWW stránek pro úplné začátečníky** (Martin Domes), **HTML začínáme programovat** (Slavoj Písek) a **HTML5 a CSS3** (Autoři: Elizabeth Castro, Bruce Hyslop). Všechny tři knihy jsou napsány v českém jazyce. Knihy obsahují základy pro tvorbu webových stránek. Poslední jmenovaná kniha obsahuje poslední novinky z tvorby www stránek, jak HTML tak CSS3. V knize najdeme postupy krok za krokem, názorné ukázky doplněné o zobrazení v internetovém prohlížeči.

Pro aktuální přehled všech novinek bych odkázala na webovou stránku **www.w3school.com**. Stránka obsahuje komplexní průvodce pro vytvoření webových stránek. Zahrnuje nejen ukázkový zdrojový kód, ale také i výsledné zobrazení.

ZÁVĚR

Hlavní cíl mé práce bylo vytvořit učební pomůcku pro předmět webové technologie. Tato příručka by měla splňovat základy pro tvorbu www stránek. Postupně má práce provází od HTML až do CSS doplněná o rozšíření znalostí v Bootstrap.

Diplomová práce je rozdělena na dvě části. V teoretické části byly vybrány a popsány jednotlivé programy. Tyto programy umožňují zápis jazyka. Na základě statistik v České republice byly vybrány právě tyto programy. Mezi uživateli je velmi oblíbený PSPad. Právě tento program jsem si vybrala pro moji práci. Důvodem bylo, že je zcela volně ke stažení a je možné ho plně při výuce webových technologií využít.

Při tvorbě www. Stránek je dobré z počátku dobré si vytvořit náskres, jak bude naše finální webová stránka vypadat. V práci je uvedeno několik zásad pro realizaci takového náskresu.

Následuje základní popis struktury HTML5 dokumentu a jeho náležitosti. Tato struktura je stavebním kamenem celých webových stránek bez něho je to jen obyčejný textový dokument.

Ke každému záznamu vytvořeného v HTML dokumentu bylo objasněno použití kaskádových stylů. Pomocí zápisu kaskádových stylů jsme mohli webovou stránku upravit do lepšího vzhledu než je jednoduchá stránka s textem a obrázky. K tomu byla zapotřebí šablona (stejnopis), která se skládá z mnoha příkazů. Přímou v programu je uvedena tahle šablona pro HTML5, která tím usnadnila práci při psaní kódu.

V neposlední řadě byla popsána struktura Bootstrap. Jaké má uplatnění pro webové technologie a její efektivní rozložení pomocí HTML a kaskádových stylů. Pro vývojáře poskytuje jednoduché a čisté řešení pro budování rozhraní. Přizpůsobí se snadno a obsahuje krásné a funkční vestavěné komponenty.

V praktické části jsou uvedeny popisy jednotlivých součástí stránek. Každá z jednotlivých struktur byla objasněna podrobně od jejího stažení k funkčnímu zápisu. Tyto jazyky nám pomáhají k zobrazení názorného popisu a vysvětlení kódu. Výsledný zápis byl vyobrazen přes webový prohlížeč. V práci byly použity co nejaktuálnější informace o zápisu jednotlivých struktur. Zvláště věnována byla pozornost novinkám pro tvorbu webových stránek v HTML5.

SEZNAM POUŽITÉ LITERATURY

- [1] Nejlepší HTML editor podle čtenářů Živě.cz. Živě.cz [online]. Josef Kraus, 2012 [cit. 2016-05-09]. Dostupné z: <http://www.zive.cz/clanky/nejlepsi-html-editor-podle-ctenaru-zivecz/sc-3-a-162955/>
- [2] PSPad [online]. Brno: Jan Fiala, 2015 [cit. 2016-05-09]. Dostupné z: <http://www.pspad.com/cz/>
- [3] Používáme PSPad. Zdroják.cz [online]. Tomáš Jukin, 2008 [cit. 2016-05-09]. Dostupné z: <https://www.zdrojak.cz/clanky/pouzivame-pspad/>
- [4] PSPad editor. Stahuj.cz [online]. 2016 [cit. 2016-05-09]. Dostupné z: http://www.stahuj.centrum.cz/vyvojove_nastroje/www-tvorba/ostatni_editory/pspad/
- [5] PSPad editor 4.6.0 #2700. Slunečnice.cz [online]. 2016 [cit. 2016-05-09]. Dostupné z: <http://www.slunecnice.cz/sw/pspad/>
- [6] Adobe Dreamweaver CS3. Slunečnice.cz [online]. 2016 [cit. 2016-05-09]. Dostupné z: <http://www.slunecnice.cz/sw/adobe-dreamweaver/>
- [7] Jednotlivci. Adobe [online]. 2016 [cit. 2016-05-09]. Dostupné z: https://creative.adobe.com/cs/plans?promoid=KSPEO&store_code=cz&single_app=dreamweaver
- [8] Nebojte se tvorby vlastního webu. Slunečnice.cz [online]. Ondřej Špulák, 2008 [cit. 2016-05-09]. Dostupné z: <http://www.slunecnice.cz/tipy/nebojte-se-tvorby-vlastniho-webu/>
- [9] Notepad++. Stahuj.cz [online]. 2016 [cit. 2016-05-09]. Dostupné z: [http://www.stahuj.centrum.cz/vyvojove_nastroje/multifunkcnieditory/notepad/?g\[hledano\]=notepad%2B%2B&g\[oz\]=6.9.0&g\[up\]=Instalace](http://www.stahuj.centrum.cz/vyvojove_nastroje/multifunkcnieditory/notepad/?g[hledano]=notepad%2B%2B&g[oz]=6.9.0&g[up]=Instalace)
- [10] Notepad++ 6.9.1. Slunečnice.cz [online]. 2016 [cit. 2016-05-09]. Dostupné z: <http://www.slunecnice.cz/sw/notepad/>
- [11] NetBeans IDE. Stahuj.cz [online]. 2015 [cit. 2016-05-09]. Dostupné z: http://www.stahuj.centrum.cz/vyvojove_nastroje/multifunkcnieditory/netbeans-ide/
- [12] Aptana Studio 3. Aptana [online]. 2014 [cit. 2016-05-09]. Dostupné z: <http://www.aptana.com/products/studio3.html>

- [13] Aptana Studio. Stahuj.cz [online]. Aptana, Inc., 2014 [cit. 2016-05-09]. Dostupné z: [http://www.stahuj.centrum.cz/vyvojove_nastroje/www-tvorba/ostatni_editory/aptana-studio/?g\[hledano\]=Aptana+Studio&g\[oz\]=3.6.0](http://www.stahuj.centrum.cz/vyvojove_nastroje/www-tvorba/ostatni_editory/aptana-studio/?g[hledano]=Aptana+Studio&g[oz]=3.6.0)
- [14] Historie HTML. Garth.cz [online]. 2008 [cit. 2016-05-09]. Dostupné z: <http://www.garth.cz/uvod-do-html/historie-html/>
- [15] Historie a vývoj HTML. HTML 5 [online]. Jiří Kosek, 2014 [cit. 2016-05-09]. Dostupné z: <http://htmlguru.cz/uvod-historie.html>
- [16] HTML tutorial [online]. W3school: Refsnes Data, ©1999-2016 [cit. 2016-05-09]. Dostupné z: <http://www.w3schools.com/html/default.asp>
- [17] LUBBERS, Peter, Brian ALBERS a Frank SALIM. HTML5: programujeme moderní webové aplikace. Vyd. 1. Brno: Computer Press, 2011. ISBN 9788025135396.
- [18] PÍSEK, Slavoj. HTML: začínáme programovat. 4., aktualiz. vyd. Praha: Grada, 2014. Průvodce. ISBN 9788024750590.
- [19] Stages of web design. Adobe [online]. Adobe, 2015 [cit. 2016-05-09]. Dostupné z: <https://helpx.adobe.com/dreamweaver/how-to/stages-web-design.html?set=dreamweaver--get-started--essential-beginners>
- [20] CASTRO, Elizabeth a Bruce HYSLOP. HTML5 a CSS3: názorný průvodce tvorbou WWW stránek. 1. vyd. Brno: Computer Press, 2012. ISBN 9788025137338.
- [21] HTML5. Webové technologie [online]. Vladimír Zelený, 2015 [cit. 2016-05-09]. Dostupné z: <http://webtech.9e.cz/html5.html>
- [22] Struktura HTML dokumentu - Český HTML 5 manuál. ITnetwork.cz [online]. David Čápka, 2016 [cit. 2016-05-09]. Dostupné z: <http://www.itnetwork.cz/html-css/html-manual/struktura/html-struktura-stranky-cesky-manual/>
- [23] Struktura HTML dokumentu - Český HTML 5 manuál. Zdrojak.cz [online]. Honza Javorek, 2011 [cit. 2016-05-09]. Dostupné z: <https://www.zdrojak.cz/clanky/css-preprocesory-mene-psani-vyssi-efektivita/>
- [24] Semantyka kodu w HTML5. Stanisław Molitorys web developer blog [online]. Stanisław Molitorys, 2014 [cit. 2016-05-09]. Dostupné z: <http://www.blog.molitorys.pl/podstawowe-elementy-html5>
- [25] DOMES. Tvorba WWW stránek: Pro úplné začátečníky. 1. vydání. Brno: Computer Press, 2012. ISBN 978-80-251-2160-3.

- [26] CSS Tutorial [online]. W3school: Refsnes Data, ©1999-2016 [cit. 2016-05-09]. Dostupné z: <http://www.w3schools.com/css/default.asp>
- [27] Vlastní styly: v CSS aneb třídy, identifikátory a složené deklarace. Jak psát web [online]. Slaný: Dušan Janovský, 2014 [cit. 2016-05-09]. Dostupné z: <http://www.jakpsatweb.cz/css/css-tridy-class.html>
- [28] Definice Stylů: v CSS aneb třídy, identifikátory a složené deklarace. WEB [online]. 2013 [cit. 2016-05-09]. Dostupné z: http://owebu.org/cze/css/definice_komb.php
- [29] CSS * Selector. W3school [online]. Refsnes Data, ©1999-2016 [cit. 2016-05-09]. Dostupné z: http://www.w3schools.com/cssref/sel_all.asp
- [30] LAZARIS, Louis. CSS okamžitě. 1. vyd. Brno: Computer Press, 2014. ISBN 9788025141762.
- [31] Styly. CSS [online]. 2013 [cit. 2016-05-09]. Dostupné z: http://owebu.org/cze/css/styly_inline.php
- [32] Podrobnosti o prohlížečích. Jak psát web [online]. Dušan Janovský, 2014 [cit. 2016-05-09]. Dostupné z: <http://www.jakpsatweb.cz/prohlizece-detaily.html>
- [33] CSS frameworky pro masy, díl první. Zdrojak.cz [online]. Martin Michálek, 2009 [cit. 2016-05-09]. Dostupné z: <https://www.zdrojak.cz/clanky/css-frameworky-pro-masy-dil-prvni/>
- [34] Bootstrap Tutorial [online]. Indie, 2016 [cit. 2016-05-10]. Dostupné z: <http://www.tutorialspoint.com/bootstrap/index.htm>
- [35] Bootstrap. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2015 [cit. 2016-05-10]. Dostupné z: <https://cs.wikipedia.org/wiki/Bootstrap>
- [36] Bootstrap 3 Tutorial [online]. Refsnes Data, 2016 [cit. 2016-05-10]. Dostupné z: http://www.w3schools.com/bootstrap/bootstrap_get_started.asp
- [37] Tabulka ASCII znaků. Luky Homepage [online]. Lukáš Tyl, 2006 [cit. 2016-05-10]. Dostupné z: <http://www.tyl.cz/Pocitace/Internet/Tabulka-ASCII-znaku.html>
- [38] Naše první stránka. HTML5 Základy HTML [online]. Jiří Kosek, 2014 [cit. 2016-05-10]. Dostupné z: <http://htmlguru.cz/zaklady-prvni-stranka.html>

- [39] What is the meaning of the meta name "robots" tag. Metatags.org [online]. United States: The Metatags Company Inc, 2016 [cit. 2016-05-10]. Dostupné z: http://www.metatags.org/meta_name_robots
- [40] DELLWIG, Ingo. HTML 4: příručka tvůrce webu. 1. vyd. Praha: Grada, 2002. Moderní programování. ISBN 8024702975.
- [41] Módy prohlížečů: proč psát na začátek stránky !doctype. Jak psát web [online]. Slaný: Dušan Janovský [cit. 2016-05-10]. Dostupné z: <http://www.jakpsatweb.cz/css/mody-prohlizecu.html>
- [42] Nadpisy - Český HTML 5 manuál. Itnetwork.cz [online]. David Čápka, 2016 [cit. 2016-05-10]. Dostupné z: <http://www.itnetwork.cz/html-css/html-manual/text/html-nadpisy-h1-h2-h3-h4-h5-h6>
- [43] Nadpisy - Český HTML 5 manuál. Htmlguru.cz [online]. Jiří Kosek, 2012 [cit. 2016-05-10]. Dostupné z: <http://htmlguru.cz/zaklady-cleneni-dokumentu.html>
- [44] Frázové tagy - Český HTML 5 manuál. Itnetwork.cz [online]. David Čápka, 2016 [cit. 2016-05-10]. Dostupné z: <http://www.itnetwork.cz/html-css/html-manual/text/html-frazove-tagy-cesky-manual>
- [45] Měníme typy písma. Htmlguru.cz [online]. Jiří Kosek, 2014 [cit. 2016-05-10]. Dostupné z: <http://htmlguru.cz/zaklady-zmena-pisma.html>
- [46] Seznamy. Htmlguru.cz [online]. Jiří Kosek, 2014 [cit. 2016-05-10]. Dostupné z: <http://htmlguru.cz/zaklady-seznamy.html>
- [47] Seznamy a slovníček pojmů - Český HTML 5 manuál. Htmlguru.cz [online]. David Čápka, 2016 [cit. 2016-05-10]. Dostupné z: <http://www.itnetwork.cz/html-css/html-manual/seznamy/html-seznamy-a-slovnicek-pojmu-cesky-manual>
- [48] HTML Tag. W3school.com [online]. Refsnes Data, ©1999-2016 [cit. 2016-05-10]. Dostupné z: http://www.w3schools.com/tags/tag_ol.asp
- [49] Obrázky. Htmlguru.cz [online]. Jiří Kosek, 2014 [cit. 2016-05-11]. Dostupné z: <http://htmlguru.cz/zaklady-obrazky.html>
- [50] HTML Tag. W3school.com [online]. Refsnes Data, ©1999-2016 [cit. 2016-05-11]. Dostupné z: http://www.w3schools.com/tags/tag_img.asp
- [51] HTML <a> Tag. W3school.com [online]. Refsnes Data, ©1999-2016 [cit. 2016-05-11]. Dostupné z: http://www.w3schools.com/tags/tag_a.asp

- [52] Odkazy. Htmlguru.cz [online]. Jiří Kosek, 2012 [cit. 2016-05-11]. Dostupné z: <http://htmlguru.cz/zaklady-odkazy.html>
- [53] HTML <input> Tag. W3school.com [online]. Refsnes Data, ©1999-2016 [cit. 2016-05-11]. Dostupné z: http://www.w3schools.com/tags/tag_input.asp
- [54] Příklady na input z html 5. Jakpsatweb.cz [online]. Dušan Janovský, 2015 [cit. 2016-05-11]. Dostupné z: <http://www.jakpsatweb.cz/priklady/input-html5.html>
- [55] HTML <textarea> Tag. W3school.com [online]. Refsnes Data, 2016 [cit. 2016-05-11]. Dostupné z: http://www.w3schools.com/tags/tag_textarea.asp
- [56] HTML <select> Tag. W3school.com [online]. Refsnes Data, 2016 [cit. 2016-05-11]. Dostupné z: http://www.w3schools.com/tags/tag_select.asp
- [57] HOGAN, Brian P. HTML5 a CSS3: výukový kurz webového vývojáře. Vyd. 1. Brno: Computer Press, 2011. ISBN 9788025135761.
- [58] PFEIFFER, Silvia. HTML5 - audio a video: kompletní průvodce. Vyd. 1. Brno: Zoner Press, 2011. Encyklopedie webdesignera. ISBN 9788074131479.
- [59] HTML5 – geolokační rozhraní. Programujte.com [online]. Martin Šimeček, 2012 [cit. 2016-05-11]. Dostupné z: http://programujte.com/clanek/2011052400-html5-geolokacni-rozhrani/#clanek_code_1
- [60] Webdesignérův průvodce po HTML5: Microdata. Zdrojak.cz [online]. Honza Sládek, 2010 [cit. 2016-05-11]. Dostupné z: <https://www.zdrojak.cz/clanky/webdesigneruv-pruvodce-po-html5-microdata/>
- [61] Microdata aneb obohaťte svůj výsledek ve vyhledávání. Programujte.com [online]. Filip Podstavec, 2012 [cit. 2016-05-11]. Dostupné z: http://programujte.com/clanek/2012033001-microdata-aneb-obohatte-svuj-vysledek-ve-vyhledavani/#clanek_code_2
- [62] CSS3 tutorial [online]. Refsnes Data, ©1999-2016 [cit. 2016-05-12]. Dostupné z: http://www.w3schools.com/css/css3_intro.asp
- [63] CSS3 background-size Property. W3school.com [online]. Refsnes Data, ©1999-2016 [cit. 2016-05-12]. Dostupné z: http://www.w3schools.com/cssref/css3_pr_background-size.asp
- [64] Float. Jak psát web [online]. Dušan Janovský, 2012 [cit. 2016-05-12]. Dostupné z: <http://www.jakpsatweb.cz/css/float.html>

- [65] BROWN, Tiffany B, Kerry BUTTERS a Sandeep PANDA. HTML5 okamžitě: [ovládněte HTML5 za víkend]. 1. vyd. Brno: Computer Press, 2014. ISBN 9788025142967.
- [66] DOMES, Martin. 333 tipů a triků pro CSS. 2., aktualiz. vyd. Brno: Computer Press, 2011. ISBN 9788025133668. Dostupné také z: http://toc.nkp.cz/NKC/201103/contents/nkc20112168878_1.pdf
- [67] Clear. Jak psát web [online]. Dušan Janovský, 2012 [cit. 2016-05-12]. Dostupné z: <http://www.jakpsatweb.cz/css/clear.html>
- [68] Display. Jak psát web [online]. Dušan Janovský, 2012 [cit. 2016-05-12]. Dostupné z: <http://www.jakpsatweb.cz/css/display.html>
- [69] CSS font-family Property. W3school.com [online]. Refsnes Data, ©1999-2016 [cit. 2016-05-12]. Dostupné z: http://www.w3schools.com/cssref/pr_font_font-family.asp
- [70] Bootstrap tutorial [online]. India: Rights Reserved, 2016 [cit. 2016-05-12]. Dostupné z: http://www.tutorialspoint.com/bootstrap/bootstrap_environment_setup.htm
- [71] Bootstrap 3 Tutorial [online]. Refsnes Data, ©1999-2016 [cit. 2016-05-12]. Dostupné z: <http://www.w3schools.com/bootstrap/default.asp>

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

HTML	Hyper Text Markup Language
CSS	Cascading Style Sheet
PHP	Hypertext Preprocessor
JAVA	Programovací jazyk
JAVASCRIPT	Programovací jazyk
SQL	Structure Query Language
C	Programovací Jazyk
C++	Programovací jazyk
XHTML	Extensible Hypertext Markup Language
RUBY	Skriptovací programovací jazyk
RAILS	Skriptovací programovací jazyk
PYTHON	Programovací jazyk
WWW	Word Wide Web
CERN	Conseil Européen pour la recherche nucléaire
W3C	World Wide Web Consortium
XML	Extensible Markup Language
WHATWG	The Web Hypertext Application Technology Working Group
JQUERY	Javascriptová knihovna
URL	Uniform Resource Locator
WAV	Waveform audio file format
OGG	Zvukový formát
MPE3	Video formát
MPEG-4	Video formát
WebM	Video formát
AAC	Video formát
GPS	Global Positioning System
RDF	Resource Description Framework

GSM	Globální Systém pro Mobilní komunikaci
RGB	Barevný model
RGBA	Barevný model
HSL	Barevný model
HSLA	Barevný model
PX	Pixel

SEZNAM OBRÁZKŮ

Obrázek 1: Prostředí programu PSPad [vlastní zpracování]	12
Obrázek 2: Vývojové prostředí Adobe Dreamweaver [1]	13
Obrázek 3: Vývojové prostředí Notepad++[vlastní zpracování]	14
Obrázek 4: Vývojové prostředí NetBeans[vlastní zpracování]	15
Obrázek 5: Vývojové prostředí Aptana[1].....	16
Obrázek 6: Ukázka návrhu HTML stránky[19].....	20
Obrázek 7: Grafická zobrazení HTML struktury	22
Obrázek 8: Obsah HTML stránky [24].....	25
Obrázek 9: Ukázka zápisu HTML kódu	27
Obrázek 10: Ukázka zápisu v CSS dokumentu	27
Obrázek 11: Výsledné zobrazení v prohlížeči	27
Obrázek 12: Ukázka vložení Syntaxe - Identifikátor.....	28
Obrázek 13: Ukázka použití Selector - Prvek třída	28
Obrázek 14: Ukázka použití Selector - Prvek třída	28
Obrázek 15: Ukázka použití seskupení prvků	29
Obrázek 16: Ukátka použití Selector *	29
Obrázek 17:Ukázka použití Selector *	29
Obrázek 18: Ukázka použití Selector atributu	30
Obrázek 19: Ukázka použití Pseudotřída.....	30
Obrázek 20: Ukázka Interní Šablony.....	31
Obrázek 21: Ukázka externí šablony	32
Obrázek 22: Ukázka Inline Stylu.....	32
Obrázek 23: Komponenty CSS frameworků [33].....	34
Obrázek 24: Výukové prostředí PSPad programu	40
Obrázek 25: Lišta s panely.....	40
Obrázek 27: Nástrojový panel	41
Obrázek 26: Panel souborů	41
Obrázek 28: Stavový řádek.....	41

Obrázek 29: Ukazatel pozice v souboru	42
Obrázek 30: ASCII tabulka.....	43
Obrázek 31: Změna syntaxe	43
Obrázek 32: Názorná ukázka formátovacích značek.....	44
Obrázek 33: Vytvoření nového dokumentu v programu PSPad.....	45
Obrázek 34: Ukázka rychlého zobrazení v programu	46
Obrázek 35: Ukázka uložení webové stránky.....	46
Obrázek 36: Ukázka zobrazení v prohlížeči	47
Obrázek 37: Zobrazení v prohlížeči se zvětšením na 500%.....	47
Obrázek 38: Zobrazení v horní liště prohlížeče.....	49
Obrázek 39: Tělo dokumentu HTML	49
Obrázek 40: Nesprávně vložený formát	50
Obrázek 41: Výběr formátu UTF-8	50
Obrázek 42: Ukázka vloženého komentáře v hlavičce dokumentu	51
Obrázek 43: Ukázka nadpisů	52
Obrázek 44: Ukázka Nadpisu a odstavců	53
Obrázek 45: Ukázka bez zalomení textu	53
Obrázek 47: Ukázka zápisu zarovnání nadpisu	54
Obrázek 48: Ukázka zápisu zarovnání běžného textu	54
Obrázek 49: Zarovnání nadpisu a běžného textu.....	54
Obrázek 46: Ukázka zalomení řádku.....	54
Obrázek 50: Zarovnání části odstavců.....	55
Obrázek 51: Ukázka základního a prezenčního textu.....	56
Obrázek 52: Ukázka neuspořádaného seznamu.....	57
Obrázek 53: Ukázka uspořádaného seznamu	58
Obrázek 54: Ukázka typů seznamů	58
Obrázek 56: Popis vložení obrázku	59
Obrázek 55: Ukázka definičního seznamu	59
Obrázek 57: Ukázka souborů ve složce.....	60

Obrázek 58: Zobrazení vystoupení a zobrazení z jedné složky do druhé.....	60
Obrázek 59: Ukázka umístění textu kolem obrázku.....	61
Obrázek 60: Ukázka odkazu na webovou stránku.....	61
Obrázek 61: Ukázka více odkazů na webové stránky.....	62
Obrázek 62: Ukázka vytvoření tabulky.....	63
Obrázek 63: Ukázka sloučení a prázdné buňky.....	63
Obrázek 64: Ukázka typů atributu < intype> a textového pole.....	65
Obrázek 65: Ukázka seznamů.....	66
Obrázek 66: Ukázka posuvníku.....	66
Obrázek 67: Ukázka Obsluha čísel.....	67
Obrázek 68: Ukázka počátečních dat.....	67
Obrázek 69: Ukázka zápisu video souborů.....	68
Obrázek 70: Ukázka zápisu pro zvukové soubory.....	69
Obrázek 71: Ukázka zápisu atributu <source>.....	71
Obrázek 72: Ukázka ověření polohy.....	72
Obrázek 73: Ukázka získání a ověření polohy.....	74
Obrázek 74: Ukázka ošetření chyb.....	75
Obrázek 75: Ukázka příkladu na geolokaci pomocí tlačítka.....	76
Obrázek 76: Ukázka dvojího zápisu v HTML.....	78
Obrázek 77: Ukázka zobrazení v prohlížeči.....	78
Obrázek 78: Kontrola zápisu nástrojem Rich snippet testing tool.....	79
Obrázek 79: Ukázka nepárové značky vložené do hlavičky.....	80
Obrázek 81: Ukázka komentářů v CSS.....	81
Obrázek 80: Ukázka vložení externího souboru do hlavičky HTML.....	81
Obrázek 82: Ukázka barevného vyobrazení.....	83
Obrázek 83: Umístění polohy.....	84
Obrázek 84: Ukázka vložení pořadí.....	84
Obrázek 85: Ukázka ohraničení.....	85
Obrázek 86: Ukázka ohraničení textu.....	87

Obrázek 87: Ukázka vložení odkazů.	88
Obrázek 88: Ukázka výplně.	89
Obrázek 89: Ukázka výšky a šířky.	90
Obrázek 90: Ukázka úpravy textu	93
Obrázek 91: Ukázka stylu písma	94
Obrázek 92: Ukázka seznamů.	95
Obrázek 93: Ukázka stínů.	96
Obrázek 94: Ukázka vložení sloupců.	97
Obrázek 95: Ukázka internetového prohlížeče Bootstrap.	98
Obrázek 96: Ukázka výběru možností ke stažení.	98
Obrázek 97: Vložení stylu do hlavičky dokumentu HTML.	99
Obrázek 98: Vložení meta značky do hlavičky dokumentu HTML.	99
Obrázek 99: Vložení odkazů na knihovny.	99
Obrázek 100: Ukázka uspořádání ve složce.	100
Obrázek 101: Základní struktura Bootstrap Grid.	100
Obrázek 102: Ukázka sloupců v Bootstrap.	100
Obrázek 103: Ukázka zobrazení sloupců v prohlížeči.	101
Obrázek 104: Ukázka vložení tlačítek.	103
Obrázek 105: Ukázka vložení formuláře.	104
Obrázek 106: Ukázka vložení obrázku.	106
Obrázek 107: Ukázka zobrazení tabulky.	107
Obrázek 108: Ukázka rozbalovací nabídky.	108

SEZNAM TABULEK

Tabulka 1: Předpony testovacích vlastností [32].....	33
Tabulka 2: Rozdělení Bootstrap na 12 sloupců [36].....	37
Tabulka 3: Vlastnosti Bootstrap [34].....	38
Tabulka 4: Základní zvýraznění textu [44] [45].....	55
Tabulka 5: Prezenční zvýraznění textu [44] [45].....	56
Tabulka 6: Popis atributů odkazu[51].....	62
Tabulka 7: Ukázka popisu atributu „<input type>“ [53] [54].....	64
Tabulka 8: Kodeky pro <video>.[58].....	67
Tabulka 9: Popis funkcí atributu <video>. [20].....	68
Tabulka 10: Kodeky pro <audio>. [58].....	69
Tabulka 11: Tabulka funkcí pro <video>. [20].....	70
Tabulka 12: Tabulka funkcí pro <source>. [20].....	71
Tabulka 13: Ukázka typografie Bootstrap.....	102
Tabulka 14: Ukázka vložení atributů tlačítka.....	104
Tabulka 15: Ukázka vložení horizontálního a vertikálního formuláře.....	105
Tabulka 16: Popis atributů pozadí.....	105
Tabulka 17: Popis atributů umístění.....	106
Tabulka 18: Popis atributů obrázku.....	107
Tabulka 19: Popis atributů tabulky.....	108
Tabulka 20: Popis atributů rozbalovací nabídky.....	108