

ZÁPADOČESKÁ UNIVERZITA V PLZNI
FAKULTA PEDAGOGICKÁ
KATEDRA VÝPOČETNÍ A DIDAKTICKÉ TECHNIKY

**VYUŽITÍ NOVÝCH MOŽNOSTÍ NÁVRHU WEBOVÝCH
STRÁNEK POMOCÍ HTML5 V PRAXI**
BAKALÁŘSKÁ PRÁCE

Kateřina Hegnerová

B1001 Přírodovědná studia, obor Informatika se zaměřením na vzdělávání

Vedoucí práce: Mgr. Tomáš Jakeš

Plzeň, 2014

Prohlašuji, že jsem bakalářskou práci vypracovala samostatně
s použitím uvedené literatury a zdrojů informací.

V Plzni,

.....

OBSAH

Úvod	3
1 HISTORIE A VÝVOJ JAZYKŮ HTML A HTML5.....	4
1.1 HTML 2.0 A HTML 3.X.....	4
1.2 HTML 4.0.X.....	5
1.3 XHTML.....	5
1.4 HTML 5	5
2 PŘEDSTAVENÍ HTML5 JAKO KONCEPT JEDNOTLIVÝCH ROZHRANÍ	7
2.1 MULTIMEDIÁLNÍ ROZHRANÍ	7
2.1.1 Audio/Video.....	7
2.1.2 Peer to peer	7
2.2 GRAFICKÝ OBSAH	7
2.2.1 2D a 3D grafika a animace	7
2.3 PRÁCE V REŽIMU OFFLINE A WEBOVÉ APLIKACE	8
2.3.1 Webové aplikace	8
2.3.2 File API	8
2.3.3 Web Storage	8
2.4 PŘÍPOJNÁ ZAŘÍZENÍ	9
2.4.1 Geolokace	9
2.4.2 Vstup/Výstup	9
2.5 FORMULÁŘOVÉ PRVKY	10
2.5.1 Input TYPE.....	10
2.6 KONEKTIVITA	11
2.6.1 Web Socket.....	11
2.7 VÝKON A INTEGRACE	11
2.7.1 Drag and Drop	11
2.7.2 Web Workers.....	11
3 VYBRANÉ ELEMENTY JAZYKA HTML5 A PŘÍKLADY JEJICH POUŽITÍ	13
3.1 CANVAS	13
3.1.1 Základní práce s plátnem.....	13
3.1.2 Souřadnicový systém plátna.....	14
3.1.3 Kreslení přímek.....	14
3.1.4 Základní práce s textem.....	15
3.2 ROZHRANÍ AUDIO/VIDEO.....	17
3.2.1 Soubor kodeků	17
3.2.2 Video.....	18
3.2.3 Podpora video kodeků v prohlížečích.....	19
3.2.4 Audio.....	21
3.2.5 Podpora audio kodeků v prohlížečích	21
3.3 ELEMENTY SESKUPENÍ.....	22
3.3.1 <HR>	22
3.3.2 Elementy pro zobrazení seznamu	23
3.3.3 <figure> a <figcaption>.....	23
3.4 DRAG AND DROP	24
3.4.1 Začínáme	24
4 POROVNÁNÍ STRUKTURY WEBOVÉHO DOKUMENTU (X)HTML A HTML5	27
4.1 ZÁKLADNÍ STRUKTURA DOKUMENTU HTML4.01	27

4.1.1	DOCTYPE	27
4.1.2	Head.....	28
4.1.3	Body.....	29
4.2	ZÁKLADNÍ STRUKTURA DOKUMENTU HTML5	30
4.2.1	DOCTYPE	30
4.2.2	Head.....	31
4.2.3	Body.....	31
5	WEBOVÉ STRÁNKY ILUSTRUJÍCÍ VYUŽITÍ TECHNOLOGIE HTML5.....	35
5.1	ROZVRŽENÍ STRÁNKY	35
5.2	GRAFICKÉ ZPRACOVÁNÍ DOKUMENTU	36
5.3	LOGICKÁ ČÁST DOKUMENTU	37
5.3.1	Pages.....	37
5.3.2	Scripts	38
	ZÁVĚR.....	40
	RESUMÉ.....	41
	SEZNAM LITERATURY	42
	SEZNAM OBRÁZKŮ	I

Úvod

Tato bakalářská práce se bude věnovat popisu novinek ve specifikaci jazyka HTML5. V úvodu představíme evoluční linii postupného vývoje značkovacího jazyka HTML, který dospěl až do současné verze.

Dále budou představy rozhraní (komponenty), které nově tato specifikace využívá. Ty budou přehledně uspořádány do podkapitol, součástí bude i popis jejich účelu a použití.

Na vybraných komponentech bude ukázáno, jak je používat skrze podrobně popsané návody. Samozřejmě budou doprovodné obrázky, které doplní případné nejasnosti.

Dále budou popsány prvky a atributy používané nejen v předchozích verzích jazyka HTML.

Na závěr bude bakalářská práce doplněna o funkční webové stránky naprogramované dle specifikace HTML5, kde si bude možno vybrané komponenty a vlastnosti vyzkoušet.

1 HISTORIE A VÝVOJ JAZYKŮ HTML A HTML5

Již od počátku vývoje prvních počítačů lidstvo toužilo po záznamu mluveného slova a vzájemném sdílení informací. V průběhu vývoje psaného textu a jazyka lidé došli k různým způsobům šíření příběhů, poezie i svých vědeckých objevů. Významným pomocníkem byl zcela jistě objev knihtisku, který je znám od roku 1447 (Gutenberg Johann). Ten výraznou měrou napomohl a urychlil záznam psaného textu. V průběhu 20. století došlo k prudkému rozvoji informačních a sdělovacích technologií, které umožnily šíření informací téměř v reálném čase. Na počátku 90. let se tyto možnosti rozhodl využít Tims Berners-Lee (1991) pracující na známé univerzitě v CERNu (Conseil Européen pour la recherche nucléaire). Jeho cílem bylo použít je k celosvětovému usnadnění komunikace mezi svými spolupracovníky, obzvláště pak sdílení výsledků získávaných v průběhu výzkumu Vysokých energií. Pro vytvoření souhrnného komunikačního kanálu využil všechny své dosavadní zkušenosti a znalosti v té době známých technologií a počítačových sítí. Vytvořil nový způsob popisu struktury dokumentu a definoval metody formátování textu. Svůj objev popsal v materiálu HTML tags¹. Zároveň vytvořil první nástroje (např. webový prohlížeč) pro vytváření a zobrazování takto uložených informací. [1] [2]

1.1 HTML 2.0 A HTML 3.X

Studenti a profesori po celém světě si velmi rychle uvědomili potenciál a snadnost použití tohoto nástroje. V krátkém čase začalo vznikat mnoho modifikací a vylepšení, které umožňovaly např. snadné zobrazení tabulek, obrázků a dalších prvků. Začaly se objevovat první potíže s kompatibilitou, které bylo třeba řešit vytvořením standardu, byl nazván HTML 2.0. Tento standard postačil ale jen na velmi krátkou dobu, rychlost vývoje získávala na obrátkách a již pouhé dva roky poté vznikl návrh standardu HTML 3.0, který tento jazyk obohatil o možnosti obtékání obrázku textem, styly dokumentů a možnost zápisu matematických vzorců. V této době již existovala celá řada aplikací sloužící k prohlížení takto vytvářených stránek, nicméně implementace podpory HTML 3.0. se pro softwarové firmy stala příliš náročnou a i když například ve své době oblíbený a velmi pokročilý prohlížeč Netscape implementoval některé prvky specifikace HTML 3.0., nebyla tato verze přijata jako standard. [2] [3]

¹ Více informací o HTML tags na <http://www.w3.org/History/19921103-hypertext/hypertext/WWW/MarkUp/Tags.html>

Vynálezce HTML Berners-Lee se mezitím staral o dohled nad rozšiřováním jazyka HTML a v říjnu roku 1994 založil konsorciu W3C, jako organizaci, která má dohlížet na další vývoj webu jím definovaných standardů. 14. ledna 1997 tak vznikla specifikace HTML 3.2. Ta obsahovala rozšíření vzniklé od počátku roku 1996. [2] [3]

1.2 HTML 4.0.x

V červenci 1997 byla vytvořena verze HTML4.0, ve které se autoři snažili dosáhnout původního záměru, kdy by jednotlivé prvky měly označovat význam daných částí dokumentů. Designová vrstva je zde již řízena a určována za pomoci stylů. Naproti tomu jsou ze specifikace vyjmuty některé značky umožňující určování vzhledu stránky. V roce 1999 byla komunitou W3C představena verze HTML 4.0.1., která obsahovala pouze opravy některých chyb. [2] [3]

1.3 XHTML

V posledních letech 20. století se metoda formátování informací pomocí značek začíná ukazovat použitelnou i v jiných oblastech informatiky. Vzniká tak formát ukládání informací XML, který se dnes používá například pro definování konfiguračních souborů počítačových programů nebo pro přenos dat mezi různými aplikacemi. Dalším příkladem může být využití při exportování dat z databáze. Jednotlivé značky zde mohou mít různý význam a definují je vývojáři nejrůznějších aplikací. Z této myšlenky strukturovaného jazyka, bez předem definovaných značek, vzniká standard XHTML 1.0, který neobsahoval téměř žádné značky formátovacího významu a grafický vzhled dokumentu je zde nutný definovat téměř výhradně pomocí kaskádových stylů. Další vývoj ukázal, že tak radikální změny původního konceptu jazyka HTML s příliš nízkou přidanou hodnotou v podobě nových prvků není dostatečnou motivací pro autory webových prohlížečů, aby jej implementovali, a pro samotné vývojáře, kteří jej odmítli používat. Navíc, na rozdíl od jazyků založených na HTML, XML striktně vyžaduje dodržení syntaxe, kdy například zapomenutí ukončení značky může způsobit nepřeložení celého XML dokumentu. [2] [3]

1.4 HTML 5

Mezi tím skupiny pohybující se v okolí vývoje webových prohlížečů (Mozilla, Opera Software) zakládají iniciativu WHATWG (The Web Hypertext Application Technology

Working Group)². Jejich záměrem bylo připravit podklady pro vznik nové verze HTML a její následné schválení organizací W3C³. V březnu 2007 se nakonec W3C rozhodlo spojit své síly s WHATWG za účelem vytvoření nové verze HTML. V roce 2010 již webové stránky obsahují značné množství aktivních technologií, které významně rozšiřují jejich prezentační schopnosti (flash, JavaScript). Ty při svém provozu nevyužívají jazyku HTML, ale vlastních způsobů zobrazování. Vzniká tak velké množství aplikací, které jsou provozovány v prostředí webového prohlížeče a jazyk HTML využívají pouze jako prostředníka. Webové stránky tak obsahují spousty cizích prvků, což je zdrojem nestability a vzájemné nekompatibility webových prohlížečů. Některé webové aplikace ke svému běhu vyžadují konkrétní webový prohlížeč a jeho určitou verzi. Jazyk HTML tak byl doplněn novou řadou funkcí, s cílem eliminovat potřebu použití jiných technologií (Flash, JavaScript apod.). Stejně tak byla usnadněna validace (HTML5 sjednocuje různé typy validace do jediné). Interpretace jazyka byla upravena tak, aby bylo možné zobrazení například chybně zanořených, nebo překřížených značek. [2] [3]

² Více informací o WHATWG na <http://www.whatwg.org/>

³ Více informací o W3C na <http://www.w3.org/>

2 PŘEDSTAVENÍ HTML5 JAKO KONCEPT JEDNOTLIVÝCH ROZHRAŇÍ

Nedílnou součástí HTML5 tvoří různá rozhraní umožňující rozšiřování samotného obsahu webových stránek o další interaktivní a multimediální prvky. Rozhraní se nazývají komponenty, a přidáním dalších lze rozšiřovat základ daný standardem HTML5.

2.1 MULTIMEDIÁLNÍ ROZHRAŇÍ

2.1.1 AUDIO/VIDEO

Na rozdíl od předcházejících verzí jazyka HTML usnadňuje HTML5 použití audiovizuálních děl jako součást webových prezentací. Snaží se definovat standardy, které by odstranily nutnost použití přehrávačů třetích stran. Nabízí k tomu speciálně navržené značky `<audio>` a `<video>`, jejichž použití bude detailněji popsáno v kapitole 3.2.

2.1.2 PEER TO PEER

Implementace peer to peer protokolu umožňuje komunikaci mezi počítači koncových uživatelů. Ukázkou použití může být i technologie Web RTC, vytvořená společností Google (nově připravovaný standard W3C), která by měla usnadnit přenos zvuku a obrazu v reálném čase a to vše bez potřeby instalace dalších pluginů. Bohužel, i v tomto případě se pozvolna schyluje k souboji velkých softwarových společností (obdobně jako v případě video kodeků), kvůli čemuž v současné době není možné zaručit vzájemnou kompatibilitu prohlížečů. V současné době jako příklad můžeme uvést norskou službu `Appear.in`⁴, která umožňuje vytvořit video konferenci pomocí několika jednoduchých kroků. Tato služba nefunguje například v IE či Safari. [4] [5]

2.2 GRAFICKÝ OBSAH

2.2.1 2D A 3D GRAFIKA A ANIMACE

Pro vytváření grafického obsahu HTML5 nabízí komponentu Canvas, který jako vývojový nástroj využívá rozhraní umožňující práci s JavaScriptem. S použitím této komponenty nás podrobněji seznámí kapitola 3.1.

⁴ Více info o službě `Appier.in` na <https://appear.in/>

2.3 PRÁCE V REŽIMU OFFLINE A WEBOVÉ APLIKACE

2.3.1 WEBOVÉ APLIKACE

Práce v režimu offline je velice zajímavou novinkou ve specifikaci HTML5. Vstoupíme-li prvně na stránky, internetový prohlížeč na straně uživatele uloží veškerý obsah, který má být zobrazen. To vše se děje, připojujeme-li se online. Offline aplikace taktéž umožňují prohlížení stránek. Vše funguje tak, že prohlížeč se stránky zeptá, jaká data a soubory jsou potřeba, a ty poté stáhne. Takto jednoduše můžeme prohlížet předem uložený obsah i na místech bez dostupnosti internetového připojení. [1] [6]

2.3.2 FILE API

Před příchodem HTML5 nebyla v oblasti webových technologií na straně uživatele umožněná běžná práce se soubory. Žádná z předchozích verzí HTML neumožňovala sama o sobě například vytvořit jednoduchý grafický editor, ve kterém by si uživatel mohl oříznout fotografii a výsledek uložit na disk, nebo odeslat na server, nebo když se například uživatel rozhodl svoji fotografii vložit do svého profilu na nějaké sociální síti, musel jej předem upravit nebo na server odeslat v plné velikosti, a teprve ten se postaral o vytvoření náhledu nebo zmenšení fotografie. Tyto a mnohé další nevýhody způsobované nemožností standardní práce se soubory odstraňuje právě File Api. [1] [7]

2.3.3 WEB STORAGE

Již v první polovině 90. let vývojáři webových aplikací s oblibou využívali možnost vytváření a ukládání malých souborů na straně uživatele, takzvaných cookies. Ty dávají možnost ukládání dat a nejrůznějších nastavení, ale mohou být i snadno zneužity pro účely sledování činnosti uživatele nebo zasílání nevyžádané pošty. Tato vlastnost není příliš oblíbená, a proto v současné době mnoho zkušenějších uživatelů využívá možnosti zakázání cookies. Další jejich nevýhodou je jejich omezená velikost (max. 4 KB) a nutnost předávat cookies při každém požadavku, což zvyšuje síťový provoz. Jazyk HTML5 stále umožňuje používání cookies, ale navíc definuje zcela nové rozhraní, které se nazývá webové úložiště. Webové úložiště se snaží odstranit většinu nevýhod svého předchůdce. Například dává vývojářům možnost zvolit, zda data mají být uložena na straně uživatele (Local Storage) nebo zda mají být zasílána spolu s požadavky na stranu serveru (Session Storage). Díky možnosti ukládání informací v počítači koncového uživatele bylo možné

zrušit omezení týkající se velikosti uchovávaných dat, aniž by došlo k nárůstu síťového provozu. Díky jednoduchému a přehlednému aplikačnímu rozhraní jsou ukládané hodnoty snadno přístupné a lze je uchovávat i po zavření záložky nebo celého prohlížeče. Na rozdíl od mnoha jiných vlastností jazyka HTML5 je v současné době rozhraní Web Storage možné používat ve všech prohlížečích. [1] [8]

2.4 PŘÍPOJNÁ ZAŘÍZENÍ

2.4.1 GEOLOKACE

Tato funkce snadno a rychle umožňuje zjištění a odeslání naší polohy. Jedná se také o jednu z nových vlastností specifikace HTML5. Je mnoho způsobů jak zjistit údaje o aktuální pozici návštěvníka stránek - můžeme tak učinit pomocí IP adresy či GPS. Geolokační rozhraní umožňuje zjištění polohy pomocí bezdrátových sítí, proto ne vždy odpovídá lokalizovaná poloha přesné poloze. Takové rozhraní využívají např. Google mapy. [1] [9]

2.4.2 VSTUP/VÝSTUP

Vstupní a výstupní zařízení jsou vcelku novinkou ve specifikaci HTML5. Nabízí podporu celoobrazovkového režimu či podporu webových notifikací. Jedná se o rozhraní, jež umožňuje zobrazit upozornění mimo kontext stránky, například na domovské stránce prohlížeče či v rohu prohlížeče. Tato funkce není stále podporována všemi prohlížeči, výjimkou jsou Google Chrome a MozillaFirefox, u nichž jest podpora v posledních čtyřech verzích. U vstupních zařízení může být jmenován přístup webové kamery, jenž skýtá velký potenciál, ale její podpora je stále v začátcích. Stejně tak na tom je i podpora vstupu herního zařízení. Tento způsob ovládání aplikací může přinést ovoce a může být žádoucí v budoucím rozvoji HTML5 aplikací. V dnešní době, k nespokojenosti vývojářů, se přístup tohoto zařízení děje skrze emulaci myši nebo klávesnice. Události pomocí ukazatelů jsou další zatím téměř nepodporovanou vlastností v HTML5. Toto rozhraní poskytuje komplexní ovládání vstupních zařízení, ať již pracujeme s myší nebo na tabletu prstem. Je určeno k usnadnění psaní webových stránek, ty pak fungují správně - nehledě na to, na jaké hardwarové konfiguraci bude spouštěno. [10]

2.5 FORMULÁŘOVÉ PRVKY

S formuláři se při procházení webových stránek setkáváme téměř na každé stránce. Ať se jedná o vyhledávání, nakupování nebo sdílení příspěvků, za vším můžeme hledat formuláře. Tento mocný prostředek, který napomáhá komunikaci, obsahuje několik elementů, bez nichž by byl tento fakt nemožný. Základním stavebním prvkem jsou zcela jistě input pole či jiná vstupní pole, která umožňují vstup dat tlačítka, checkboxy. Data jsou po zadání zkontrolována a odeslána na server k dalšímu zpracování. Standard HTML5 tyto schopnosti rozšiřuje. Přidává či přejmenovává nové atributy. Většina nových funkcí rozšiřuje ty staré a zpřijemňuje tak uživatelskou práci s formuláři. Zavádí automatickou validaci hodnoty či kontrolu právě platných znaků (například jsou-li zadávány pouze číselné hodnoty), tím není potřeba kontrolu provádět JavaScriptem. Jako příklad můžeme uvést atribut pro práci s input polem, který vyžaduje vyplnění pole (*required*) - to je bráno jako povinné, čímž uživatel ušetří čas při jeho případném nevyplnění. Atribut *patter* umožňuje snadnou a rychlou validaci pole, využití se hodí k ověření, zda uživatel zadal např. jen znaky a čísla. Další volbou může být element list a datalist, jenž umožňuje nápovědu volby u vstupního pole. [1] [11]

2.5.1 INPUT TYPE

Obecně v HTML se u formulářů používá pro pole parametr *type*. Sděluje nám účel použití daného elementu. Textová pole mají vždy atribut *types* hodnotou *text*. Mnoho těchto prvků ale není určeno jen k zadávání textu. Protože formuláře mohou zajišťovat i zpětnou vazbu čtenářů autorova díla, bylo do specifikace HTML přidáno několik k tomu určených elementů. Například *email* (pole do stránky zapisujeme `<input type="email">`) je součástí již od rané verze HTML2.0 a umožňuje vložení autorova emailu. Specifikace HTML5 zavádí kontrolu tohoto pole, zda je opravdu adresa platná, je-li zapsána v požadovaném tvaru, tj. zda obsahuje tečku a zavináč. Série elementů umožňuje práci s datem (*datetime, date, month, week, aj.*). Nabídka *color* umožňuje nastavit a vybrat barvu prvku. Zcela jistě novým prvkem v HTML5 je prvek *range*, umožňující nastavení rozsahu dle zadaných hodnot *min* a *max*. [11]

2.6 KONEKTIVITA

2.6.1 WEB SOCKET

Programátoři při vytváření aplikací často využívají možnosti síťových socketů. Ty vytvářeným procesům umožňují komunikaci v reálném čase. Pomocí dvojice IP adres a portů se zde vytváří komunikační kanál, který slouží k obousměrnému přenosu informací. Vývojáři webových aplikací tuto možnost po dlouhou dobu postrádali a byli nuceni používat metodu „heart beat“, kdy se klient v pravidelných intervalech ptal serveru, zda pro něho má nové zprávy. Websocket se snaží tuto metodu nahradit a umožnit tak mnohem snadnější vytváření online aplikací, jakými jsou například nejrůznější chaty a sociální sítě. Komponenta vytváří komunikační kanál, který je otevřený po celou dobu používání aplikace. [1]

2.7 VÝKON A INTEGRACE

2.7.1 DRAG AND DROP

Práce v prostředí jakéhokoliv desktopového prostředí nabízí několik způsobů ovládání. Zahrnuje práci jak s klávesnicí, tak i s myší, kde lze využít právě funkci drag and drop. Cíle specifikace HTML5 je tyto funkce zevšednit i v rámci prohlížení webového dokumentu. Drag and drop umožňuje daný objekt přesunout myší do cílového místa a to vše za podpory událostí, jež tuto funkci obsluhují. Více se budeme funkcionalitě drag and drop věnovat v kapitole 3.4 .

2.7.2 WEB WORKERS

Téměř každý uživatel internetu se při používání webových aplikací a procházení webových stránek někdy setkal s nepříjemnou situací, kdy webový prohlížeč jakoby zamrzl, zšedl a po nějaké době zobrazil obávanou informaci o tom, že „Skript na této stránce je buď zaneprázdněn, nebo přestal reagovat“ (MozillaFirefox) nebo nám jiný prohlížeč řekl „Ale toto je nepříjemné“ (Google Chrome). Aby uživatelé netrávili svůj cenný čas bojem s webovými prohlížeči, bylo vytvořeno rozhraní Web workers. To vývojářům umožňuje složitě úkoly svěřit procesu worker, jenž se o vykonání postará na pozadí, nejlépe bez vědomí uživatele. Bohužel v mnoha případech se tomu nelze zcela vyhnout, neboť vykonávání obtížných úloh na pozadí také spotřebovává systémové

prostředky počítače a může tak dojít ke zpomalení celého systému. Naproti tomu v dnešní době většina moderních procesorů obsahuje více jader, což právě umožňuje plnohodnotné paralelní vykonávání mnoha úkolů zároveň. Správním použitím rozhraní Web workers tedy lze aplikaci v mnoha ohledech zefektivnit. V současné době Web workers podporují téměř všechny prohlížeče. [1] [12]

3 VYBRANÉ ELEMENTY JAZYKA HTML5 A PŘÍKLADY JEJICH POUŽITÍ

3.1 CANVAS

Touto komponentou můžeme realizovat větší část grafických prvků bez potřeby externích prostředků. Canvas se sestává z kreslicí plochy - plátna, které tvoří prostor umožňující realizaci většiny grafických prvků. Plátno je možno použít i pro zobrazení vektorové grafiky, kde jsou jednotlivé obrazce zaznamenány pomocí matematických vztahů tvořených souborem vstupních dat. Oproti bitmapové grafice, kde je nutné přenášet úplné souřadnice každého bodu, ze kterého se obraz skládá, je zde, například v případě přímky, třeba zaznamenat pouze souřadnice počátečního a koncového bodu, a všechny ostatní body zobrazené mezi těmito dvěma hraničními jsou dopočítány. Pro přenos a uchování takto tvořeného obrazce, tedy oproti bitmapovému zobrazení, potřebujeme pouze zlomek datového prostoru. Nevýhodou je však výrazně vyšší výpočetní náročnost při zobrazení. Tento způsob zaznamenávání umožňuje snadným způsobem kreslit různé druhy vektorů a přímek. U přímek můžeme měnit jejich velikost, barvu či konečné zaoblení. Canvas umožňuje i pokročilou práci s vektorovými obrázky či sofistikované formátování textu. [1]

3.1.1 ZÁKLADNÍ PRÁCE S PLÁTNEM

Pro vytváření zdrojového kódu, definování jednotlivých objektů a řízení událostí se zde používá JavaScript, jehož výkonný kód je součástí projektu. Ten může být přiložen v externím souboru, nebo může být přímo součástí souboru, ve kterém se nachází HTML kód. Komponentu Canvas přidáme na stránku jednoduchým napsáním HTML značky `<canvas></canvas>`. Tento element je možné rozšířit o základní parametry definující šířku a výšku kreslicího plátna. Další možnosti vzhledu lze realizovat pomocí kaskádových stylů. Canvas zároveň definuje prostor, ve kterém bude zdrojový kód interpretován. Pro správnou funkci komponenty je třeba přidělit Canvasu unikátní identifikátor, který vytváří vazbu mezi HTML elementem a částí realizovanou JavaScriptem. Část obsahující script je třeba uzavřít mezi značku `<script></script>`. [13]

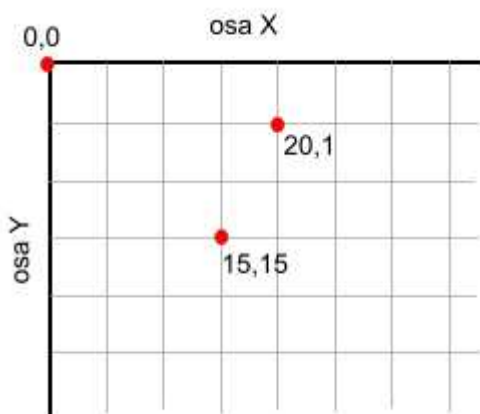
Tento odstavec bude věnovaný základním definicím, které jsou třeba pro správné použití plátna v JavaScriptu. K možnostem výše popsaného objektu Canvas přistupujeme pomocí reference, která je získávána metodou `getElementById`, jejímž vstupním

parametrem je již zmíněný unikátní identifikátor. Canvas může pracovat ve více pracovních režimech, které volíme definováním kontextu. To provedeme voláním metody `getContext`. [1] [13]V následujícím příkladu například říkáme, že budeme pracovat ve 2D:

```
<canvas id='identifikator_canvasu' height="496" width="496"></canvas>
<script>
var canvas = document.getElementById('identifikator_canvasu');
var context = canvas.getContext('2D');
</script>
```

3.1.2 SOUŘADNICOVÝ SYSTÉM PLÁTNA

Jako v mnoha programovacích jazycích umožňující práci s grafikou, tak i zde je důležité určit si, odkud se dané prvky začnou vykreslovat či případně počítat. Abychom se mohli na plátně snadno orientovat, použijeme standardní souřadnicový systém x, y . Ten v Canvasu začíná v horním levém rohu, kde se nachází bod o souřadnicích $0,0$. Názornou ukázkou je možné vidět na obrázku 1. [13]



Obrázek 1 - souřadnicový systém Canvasu

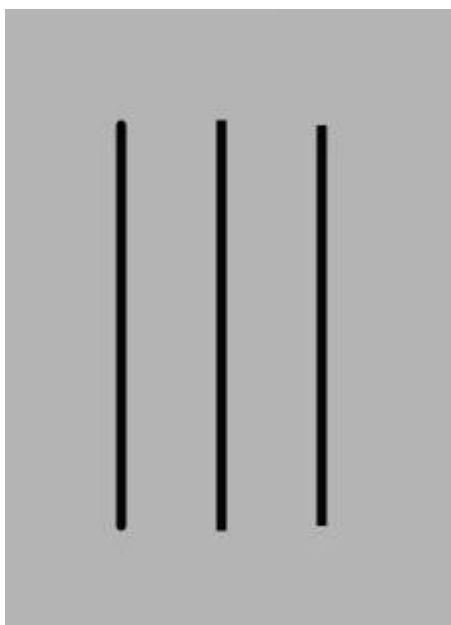
3.1.3 KRESLENÍ PŘÍMEK

Kreslení přímek je velice jednoduchou záležitostí. V první řadě musíme oznámit kreslicímu plátnu, že se chystáme kreslit. Realizujeme tak napsáním metody `begin.Path()`. V dalším kroku přesuneme kreslicí pero na počáteční souřadnice přímky – `moveTo(0,0)`. Metoda `lineTo(20,20)` udává směr a tudíž i konec přímky. Plátno nám také umožňuje lehce nastavit požadovanou barvu – `strokeStyle('#002080')`, tloušťku čáry – `lineWidth() = 3`; či nastavit tvar konce přímky – `lineCap() = round, butt, square`. Nakonec použijeme

metodu *stroke()*; , která výsledný objekt skutečně na plátno vykreslí. Níže je ukázka zdrojového kódu s ukázkou vygenerovaných přímk (viz obr. 2). [13]

```
<canvas id="platno" width="578" height="300"></canvas>
<script>
Var canvas=document.getElementById('platno');
Var context=canvas.getContext('2d');

context.beginPath();
context.moveTo(50,50);//počáteční pozice pera
context.lineTo(50,250);//konečná pozice přímky
context.lineWidth=5;//šířka přímky
context.strokeStyle='#383838';//barva přímky
context.lineCap='round';//konce přímek - round, square, butt
context.stroke();
</script>
```



Obrázek 2 - Ukázka nakreslených přímek a jejich konců

3.1.4 ZÁKLADNÍ PRÁCE S TEXTEM

Text na stránce je možné formátovat několika způsoby. Jedním z nich je standardní použití formátování, které nabízí kaskádové styly. Tento způsob je velmi elegantní, na rozdíl od úpravy textu pomocí HTML5 nám ovšem neumožňuje tolik úprav. Jednou z mnoha výhod může být například zobrazování daného textu na určitých souřadnicích.

Pro samotné vysáání textu nám stačí tyto tři metody. První *context.font = 'italic 40px Arial'*; nám umožňuje nadefinovat textu styl, je možno vybírat z kurzívy, tučného písma či normálního text (normal, bold, italic). Druhým parametrem metody je velikost

písma, v tomto případě jest uvedena v pixelech, v neposlední řadě musíme zadat druh písma. `FillStyle = 'red'`; umožňuje obarvit text na požadovanou barvu. Tu je možné zadávat slovně (red, purple, green, ...), v hexadecimálním kódu (#109000) nebo v RGB (16,144,0).

V neposlední řadě je značně důležité zadat text, který se bude vypisovat, a na jakých souřadnicích se zobrazí. K tomuto účelu použijeme metodu `fillText('Ahoj světe!',305,100)`; Jejím prvním parametrem je vypisovaný text, dále pak souřadnice, na kterých text nalezneme. [1] [13] V důsledku text a konečné zobrazení na stránce ukazuje obrázek 3. Zdrojový kód by mohl vypadat nějak takto:

```
context.font='italic 40pt Arial';//font style, size, family
context.fillStyle='#F9F9F9';//barva výplně napsaného textu (slovy, hex, RGB)
context.fillText('Ahoj světe!',103,70);// výplň - normal, italic, bold
```



Obrázek 3 - Zobrazení textu na stránce

Písmu je možno přidat rozdílnou barvou a tloušťku okrajů. HTML 5 umožňuje i tento obrys písma umístit na své souřadnice. Zdrojový kód doplníme o metody umožňující nastavit barvu obrysové čáry - `strokeStyle = '#109000'`; tloušťku obrysové čáry - `lineWidth = 3`; a nakonec text, který má být vypsán na daných souřadnicích - `strokeText('Ahoj světe!',300,100)`; Konečné zobrazení níže uvedeného kódu ukazuje obrázek 4. [13]

```
context.font='italic 40pt Arial';//font style, size, family
context.strokeStyle='black';//barva obrysu textu (slovy, hex, RGB)
context.fillStyle='#F9F9F9';//barva výplně napsaného textu (slovy, hex, RGB)
context.lineWidth=3;//tloušťka obrysu
context.strokeText('Ahoj světe!',100,70);// obrys - normal, italic, bold
context.fillText('Ahoj světe!',103,70);// výplň - normal, italic, bold
```



Ahoj světe!

Obrázek 4 - Konečné zobrazení nadpisu ve stránce

3.2 ROZHRANÍ AUDIO/VIDEO

Práce s hudbou na webových stránkách nebyla vždy jednoduchou záležitostí. Bylo často třeba mít excelentní přehled o tom, jaký prohlížeč podporuje daný formát. Přitom hudba, je-li použita s mírou a u hodících se stránek, může perfektně dobarvit obsah stránek. Se stejným problémem jsme se potýkali i při použití videa. Nebylo jednoznačně možné říci, v jakém formátu daný soubor má být, aby se z něj mohli těšit majitelé všech webových prohlížečů. Bylo zapotřebí různých zásuvných modulů či pluginů třetích stran. To vše měl změnit příchod specifikace jazyka HTML5.

3.2.1 SOUBOJ KODEKŮ

Máme-li v plánu vložit na naše stránky video či audio, je třeba zvolit správný formát, tedy potřebný kodek. Právě ale způsob komprese videa se v historii vývoje HTML5 stala jednou z nejproblematičtějších otázek. Brzy se ale zjistilo, že výrobci webových prohlížečů nejsou ochotni nalézt společnou cestu, a tak hlavně z počátku podporoval každý nějaký jiný formát. Zatímco společnost Apple a Google preferovaly komerční a po dlouhou dobu licenčními poplatky zatížený kodek H.264, Mozilla a Opera preferovaly open source kodek OggTheora. Vznikla tak nepřehledná situace, která nejvíce ztěžovala život koncovým uživatelům. Ti byli nuceni měnit své oblíbené prohlížeče nebo do nich nesnadnou cestou instalovat nejrůznější rozšíření. I velké softwarové společnosti si nakonec uvědomily chybu svého jednání, a nakonec našli společnou cestu. Google ustoupil a rozhodl se podporovat oba formáty. Společnost Apple však nadále podporuje pouze H.264. Hlavním důvodem sporů byla rozdílná patentová politika obou autorů formátů. Zatímco kodek H.264 byl komerčním produktem společnosti CISCO zatížen poplatky za autorská práva (v současné době otevřen pod licencí BSD), OggTheora byl od počátku otevřeným formátem vyvíjeným neziskovou organizací Xiph.org za finanční podpory Mozilly. Prohlížeč Google Chrome navíc podporuje kodek VP8, který byl původně

vyvinut společností On2 Technologiesale. Po převzetí společností Google byl také uvolněn pod licencí BSD. HTML5 se snaží o sjednocení podporovaných formátů.

V případě audia byla situace velmi podobná. Zde se jedná o trojici formátů AAC, MPEG3 a OggVorbis. Zatímco MP3 a AAC jsou široce používané hudební formáty využívající ztrátovou kompresi (vytvořené skupinou MPEG), OggVorbis je otevřený projekt, jehož cílem je vytvoření svobodného formátu pro šíření digitálních médií. Není náhodou, že stejně jako OggTheora nadací Xiph.org. je šířen pod BSD licencí. Z toho lze také odvodit, že zatímco hudba v tomto formátu půjde snadno přehrát v prohlížečích Mozilla a Opera, u ostatních se setkáme s problémy. [1] [14]

Na závěr této podkapitoly je nutné říci, že je na každém autorovi webových stránek, pro který z formátů se rozhodne, a každý autor stránek si musí rozhodnout, jaká je cílová skupina jeho projektu.

3.2.2 VIDEO

Prvek `<video>` vložíme do stránky napsáním `<video></video>`. Jako u ostatních elementů podporuje značka video atributy, jedná se o ovládací prvky. Atribut *autoplay* zajistí přehrávání videa ihned po načtení stránky. Pokud chceme, aby se klip začal přehrávat opět od začátku po jeho skončení, musíme uvést hodnotu *loop*. *Controls* zobrazí ovládací prvky přehrávače. Hodnotou *volume* volíme nastavení hlasitosti v rozsahu 0,0 až 1,0.

Samotný video soubor je nutno před použitím náležitým způsobem připravit. Obecně lze jen doporučit použití takového formátu, který využívá většina webových prohlížečů a v současnosti již i mobilní zařízení. Pro převod videa do cílového formátu lze použít některý z mnoha volně dostupných programů (FormatFactory, FreeStudio, aj.). Důležitým parametrem je pak volba vhodného datového toku. Pokud zvolíme příliš vysoký, dosáhneme vyšší kvality obrazu, ale z řad diváku tak vyloučíme skupinu, pro kterou je dostupné pouze pomalé připojení k internetu. Chceme-li zaujmout větší cílovou skupinu uživatelů internetu, je třeba volit kompromis a vybrat takovou hodnotu datového toku, jenž nebude tolik náročná na přenos dat a stále se bude těšit pěknému obrazu. Bohužel v tomto případě se nevyhneme časově náročnému experimentování, protože výsledná kvalita záleží na typu videozáznamu, jeho barevnosti, ale také na velikosti

přehrávaného okna (rozlišení). [1] [14] Výsledná podoba vložené značky video do stránky vypadá takto:

```
<video width="320" height="240" controls autoplay='true'>
<source src="video.ogv">
  Tento prohlížeč nepodporuje video tag.
</video>
```

3.2.3 PODPORA VIDEO KODEKŮ V PROHLÍŽEČÍCH

Na následujících srovnávacích testech je ukázána podpora videa v různých desktopových prohlížečích. Pro ukázkou byly použity prohlížeče Google Chrome verze 33.0.1750.154 (viz obr. 5), Opera verze 20 (viz obr.6), MozillaFirefox verze 28 (viz obr.9), Safari verze 5.1.7 (viz obr.7) a IE verze 11 (viz obr.8). U prohlížečů byly zvoleny jejich dostupné aktuální verze a byly aktualizovány k 19. 3. 2014.

GoogleChrome 33



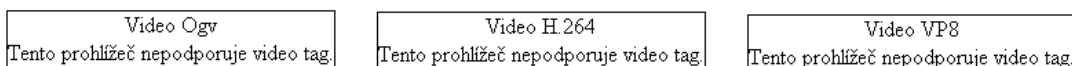
Obrázek 5 - Ukázka podpory kodeků v prohlížeči Google Chrome

Opera 20



Obrázek 6 - Ukázka podpory video kodeků v prohlížeči Opera

Safari 5.1



Obrázek 7 - Tato verze Safari nepodporuje žádný z testovaných kodeků

Internet Explorer 11



Obrázek 8 - Podpora elementu videa v Internet Explorer v. 8

MozillaFirefox 28



Obrázek 9 – MozillaFirefox nepodporuje kodekH.264

S největší podporou se můžeme setkat u kodeku Ogv, jenž je podporován třemi prohlížeči z pěti testovaných. Nejmenší podporu má video kodek H.264 s podporou jen u dvou testovaných prohlížečů. Bohužel navzdory podpoře formátu Ogv je stále problematické zajistit adekvátní převod videa do tohoto formátu. Při testu byl použit krátký open source film Big Buck Bunny⁵.

⁵ Více informací o krátkém animovaném filmu Big Buck Bunny na <http://www.bigbuckbunny.org/>

3.2.4 AUDIO

HTML5 podporuje vkládání audio souborů na stránku ve třech různých formátech, jak již bylo popsáno výše. Jedná se o soubory s audiokodeky AAC, MPEG3 a OggVorbis. Pro vložení do stránky, jak je možné vidět níže, je nutné vepsat značku `<audio>`/`</audio>`. Atributy, jež tento element podporuje, jsou například *autoplay* – umožňující hraní skladby ihned po načtení stránky. *Controls* nám zobrazí ovládací prvky přehrávače. *Loop* zajistí opětovné přehrání skladby v momentě, kdy skončí. [14] Výsledné vložení na stránku ukazuje kód:

```
<audio controls>
<source src="audio.ogg" type="audio/ogg">
  Tento prohlížeč nepodporuje audio tag.
</audio>
</div>
```

3.2.5 PODPORA AUDIO KODEKŮ V PROHLÍŽEČÍCH

Tato podkapitola se bude věnovat podpoře audio značky v již dříve zmíněných prohlížečích. Testy proběhly za stejných podmínek jako testování video značky. Byly použity nejaktuálnější možné verze prohlížečů. Jedná se o Google Chrome verze 33.0.1750.154 (viz obr. 14), Opera verze 20 (viz obr. 10), MozillaFirefox verze 28 (viz obr. 11), Safari verze 5.1.7 (viz obr. 13) a IE verze 11 (viz obr. 12).

Opera 20



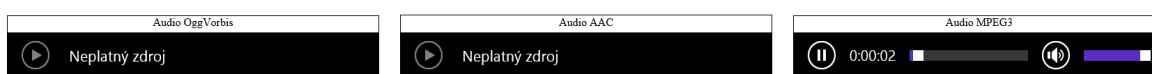
Obrázek 10 - V prohlížeči Opera je podporován jen kodek OggVorbis

MozillaFirefox 28



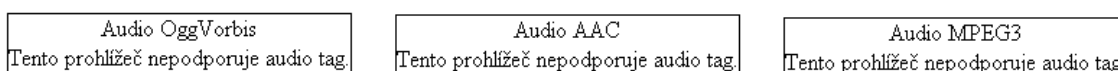
Obrázek 11 - V prohlížeči MozillaFirefox jsou v podpoře OggVorbis a MPEG3

Internet Explorer 11



Obrázek 12 - Internet Explorer podporuje jen MPEG3

Safari 5.1.7



Obrázek 13 - Prohlížeč Safari této verze nepodporuje žádný z audio kodeků

Google Chrome 33



Obrázek 14 - Google Chrome podporuje audio kodeky OggVorbis a MP3

V novějších prohlížečích podpora audio kodeků roste. Zvláště pak formátu OggVorbis, který je bohatě podporován už i ve starších verzích prohlížečů. Při testování kodeku AAC bylo nutné příklad vyzkoušet na videu, poněvadž dle specifikace HTML5 je zvuk AAC podporován jen jako součást kontejneru MP4. Navzdory specifikaci se nepodařilo prokázat funkčnost kodeku AAC v prohlížeči Safari. Při testování bylo největším úskalím zajištění programů potřebných pro konverzi s danými kodeky. Po delší analýze a testování byly použity programy Free WebMConverter a FormatFactory.

3.3 ELEMENTY SESKUPENÍ

Důležitou součástí úpravy textu jsou elementy umožňující seskupit určitou část obsahu do logicky uspořádaného oddílu. Některé elementy jsou ve specifikaci jazyka HTML5 nové, jiné se liší od předchozí verze přidánými či upravenými atributy. Mezi tyto elementy, dle specifikace W3C, patří například `<hr>`, `<p>`, `<pre>`, `<blockquote>`, elementy umožňující tvorbu seznamu - ``, ``, ``, `<dl>`, `<dt>`, `<dd>`. Dále jsou to elementy `<figure>`, `<figcaption>` nebo `<main>`. [15]

3.3.1 <HR>

Tento element je v jazyce HTML již dobře znám. Specifikace HTML5 změnila význam této horizontální čáry na tematickou přestávku v textu (viz obr. 15). V nové verzi jazyka již u tohoto elementu nejsou podporovány žádné atributy. Zápis se provádí prostým způsobem: `<hr>`.

Obrázek 15 - Zobrazení elementu hr na stránce

3.3.2 ELEMENTY PRO ZOBRAZENÍ SEZNAMU

Pro vložení seznamu do stránky je používán atribut ``. Ve specifikaci HTML5 je stále podporován, ale bez dříve známých atributů *compact* a *type*, jímž se dal nastavit druh odrážky.

Stejně i tak použití číslovaného seznamu zůstává téměř stejné. Již není podporován atribut *compact* umožňující menší velikost seznamu. V podporu vstoupil atribut *reversed* zajišťující sestupné číslování položek.

Atribut *type* u elementu položky seznamu `` již HTML5 neobsahuje.

Definiční seznam skládající se z elementů `<dl>`, `<dt>`, `<dd>`. Tag `<dl>` umožňuje vytvořit seznam definic, popis seznamu, `<dt>` slouží pro zadání definovaného termínu a `<dd>` pro samotnou definici. [1] Ukázka seznamů vložených na stránce ukazuje obrázek 16.

- HTML3.2 3. HTML3.2 W3C
- HTML4.01 2. HTML4.01 Konsorcium vyvíjející webové standardy pro WWW.
- HTML5 1. HTML5 WHATWG
Pracovní skupina snažící se o implementaci nástrojů pro snadnější psaní aplikací.

Obrázek 16 - Ukázka vlastností seznamů v HTML5

3.3.3 <FIGURE> A <FIGCAPTION>

Tyto elementy jsou v HTML5 naprostou novinkou. Element `<figure>` umožňuje na stránku vložit jakýkoliv obsah v podobě obrázku, grafu či i zdrojového kódu. `<figcaption>` pak slouží k vložení titulku obrázku. [1] [16] Jak se element zobrazí na stránce je ukázáno na obrázku 17. Zápis kódu vypadá takto:

```
<figure>

<figcaption>Tento obrázek vznikl jako první v rámci pokusů s
HDR.</figcaption>
</figure>
```



Tento obrázek vznikl jako první v rámci pokusů s HDR.

Obrázek 17 - Takto vypadá element vyobrazený na stránce

3.4 DRAG AND DROP

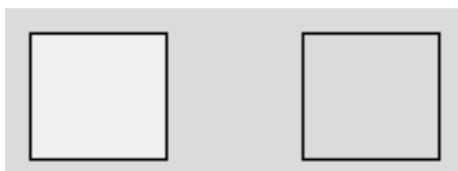
Internetové společnosti, mezi které patří například Google, se snaží o to, aby uživatelé při své práci pokud možno neopouštěli svět jejich služeb. HTML5 jim dává možnosti, jak v prostředí webového prohlížeče vytvářet aplikace, které byli uživatelé doposud zvyklí vidat pouze na svém desktopu. Funkce drag and drop patří právě mezi tyto nové možnosti. Příkladem může být například balík kancelářských aplikací Google Docs (před nedávnem se stal součástí skupiny služeb nazvaných Google Drive), kde můžeme vytvářet rozsáhle texty doplněné o obrázky a použití drag and drop nám umožňuje s nimi pracovat stejným způsobem jako v běžném textovém editoru. V neposlední řadě drag and drop využijeme při vytváření her.

3.4.1 ZAČÍNÁME

Proto, aby tato technika byla funkční, je třeba vytvořit si objekty, které budeme přetahovat. Pro názornost byl zvolen jednoduchý příklad – přetahování barevné kostky z jedné pozice na druhou. [17]Základní kód vypadá takto:

```
<div id="zdroj">
  <div id="pretah" width="88" height="31"></div>
</div>
<div id="cil"></div>
```

Dále je třeba připravit si oblast, ze které budeme objekt přesouvat, a cílové pole, ke kterému se `<div id="pretah">` přichytí. Objekty jsou tvořeny pomocí značky `div` a aplikací kaskádových stylů. Prozatím je kód neaktivní, jsou to jen obyčejné `divy` (viz obr. 18):

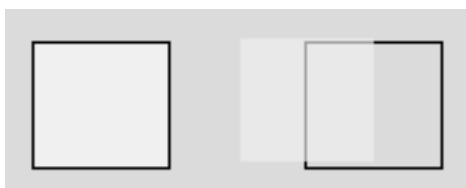


Obrázek 18 - Divy bez aplikace Drag and drop

V prvním kroku je nutné objekt, který chceme přetáhnout, utvořit chytatelným. Přiřadíme mu vlastnost `draggable="true"`:

```
<div id="pretah" draggable="true"></div>
```

Tento atribut může nabývat celkem tří hodnot – `true`, `false` a `auto`. `True` specifikuje, že element může být chycen, `false` chytit nelze. Ve většině webových dokumentů je tento atribut nastaven na `true`, což nám umožňuje s obrázkem volně manipulovat. K žádnému elementu se však nepřichytí. Není-li tento atribut nastaven, jeho defaultní hodnota je `auto`. Jeho použití je nutnou, ne však postačující podmínkou pro úspěšné dokončení operace drag and drop. [17] Na obrázku 19 lze vidět přetah prvku.

Obrázek 19 - Bílý prvek má nastaveno `draggable="true"`, lze ho tedy přetahovat

Celé tažení prvku je nutné obsluhovat řadou událostí, dokud se prvek nedostane do cílového objektu. První podstatnou funkcí je nastavení události `ondragstart`, metoda přiřazená k této události se začne vykonávat, jakmile tažení prvku započne. Atribut přiřadíme prvkům, jež mají být taženy:

```
<div id="pretah" draggable="true" ondragstart="drag(event)"></div>
```

JavaScriptová funkce přiřazená události `ondragstart` obsahuje objekt `dataTransfer` umožňující využívat rozhraní drag and drop. Metoda `setData(format, Data)`; nastaví data a jejich formát, která jsou předána ke zpracování. Na místo formátu uvádíme, zda se bude zpracovávat text nebo URL odkaz. [17] Parametrem `Data` oznámíme data, která chceme zpracovat, udáváme id přetahovaného elementu:

```
function drag(ev) {
  ev.dataTransfer.setData("Text", ev.target.id);
}
```

Při přetažení elementu musíme zjistit oblast, ve které lze daný element upustit. Událost se vyvolá při přesouvání nad cílovými prvky. Prvky, v kterých můžeme přetahovaný objekt upustit, specifikujeme událostí `ondragover`:

```
<div class="cil" ondragover="allowDrop(event)"></div>
```

Této události přiřadíme metodu, která potvrdí, že prvek může být na tomto místě umístěn. V této metodě můžeme definovat jakékoliv chování, které se uskuteční, dosáhne-li přetahovaný prvek cílové oblasti (např. změna barvy pozadí, aj.):

```
functionallowDrop(ev){
ev.preventDefault();
}
```

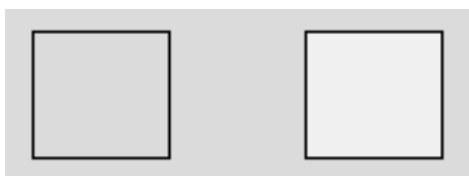
Pro dokončení operace drag and drop zbývá už jen zajistit, aby se prvek přichytil k cílovému prvku. To zajistíme přidáním události `ondrop` opět k cílovému prvku. Událost se vyvolá u cílového prvku, jakmile je přenos přetahovaného prvku dokončen:

```
<div id="cil" ondragover="allowDrop(event)" ondrop="drop(event)"></div>
```

Události jsme přiřadili tuto JavaScriptovou metodu:

```
functiondrop(ev){
ev.preventDefault();
var data=ev.dataTransfer.getData("Text");
ev.target.appendChild(document.getElementById(data));
}
```

Metoda udělá to, že data, která jsme si nastavili v metodě u události `ondragstart`, v tuto chvíli vrátíme tím, že se zobrazí v cílové oblasti a zároveň jsou smazána v oblasti startovní. V posledním řádku kódu náš prvek vložíme do stránky. [17] [18]Dokončený přesun je možno vidět na obrázku 20.



Obrázek 20 - Dokončený přesun pomocí Drag and Drop

4 POROVNÁNÍ STRUKTURY WEBOVÉHO DOKUMENTU (X)HTML A HTML5

Již Tim Berners-Lee, jako součást návrhu první podoby jazyka HTML, definoval základní strukturu HTML dokumentu. Ta se do současné doby téměř nezměnila. Každý HTML dokument, bez ohledu na použitou verzi jazyka HTML, je rozdělen do několika základních částí. Doctype, který definuje verzi použitého jazyka a obsahuje sadu informací důležitých pro správnou interpretaci HTML dokumentu prohlížečem. Hlavička, která popisuje obsah (informace o autorovi, klíčová slova pro vyhledávače a další užitečné informace) a tělo, do kterého autoři webových stránek umísťují svůj obsah. I HTML5 se drží této základní konvence, ale obsahovou část HTML dokumentu člení na další pododdíly. Dále v této kapitole popíšeme na jaké a za jakým účelem. Vznikne tak přehledné srovnání původní a dnes používané základní struktury HTML dokumentů. [2]

4.1 ZÁKLADNÍ STRUKTURA DOKUMENTU HTML4.01

4.1.1 DOCTYPE

Pro správné sestavení HTML dokumentu je důležité uvádět tento parametr. Obsahuje informace o verzi použitého HTML a DTD. DTD je zkratkou Document Type Definition a prohlížeč nebo validátor tak snadno rozpozná verzi použitého jazyka, tedy informaci o tom, které elementy a atributy jsou v dokumentu použity.

Specifikace jazyka HTML4.01 umožňuje tři základní typy DTD: Strict, Transitional a Frameset. Při použití typu Strict prohlížeč očekává pouze základní sadu značek a parametrů určených specifikací jazyka HTML4.01. Tato varianta je ze všech tří nejspornější, ale není zde umožněna zpětná kompatibilita. Oproti tomu typ Transitional umožňuje použití prvků známých z předchozích verzí jazyka HTML. Mohli bychom říci, že validátor je zde tolerantnější k chybám vznikajícím vkládáním značek, které již HTML4.01 nepodporuje. Třetí možnost umožňuje užití tak zvaných rámců nebo vložených rámců. [19] [20]

Doctype v případě HTML 4.01 definujeme následujícím způsobem:

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

Druhý způsob definuje HTML4.01 Transitional:

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

Třetím způsobem lze doctype deklarovat takto:

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

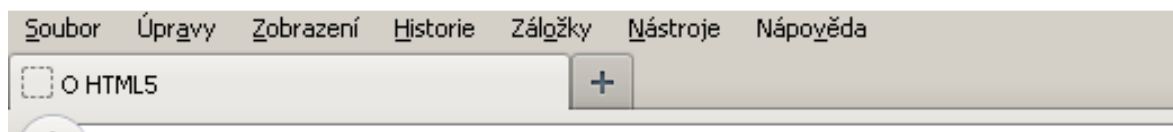
4.1.2 HEAD

Head neboli hlavička obsahuje důležité informace o HTML dokumentu. Zpravidla to je jméno autora, informace o druhu použitého kódování a titulek. Může ale obsahovat i celou řadu meta značek, určených například pro webové katalogy a vyhledávače, nebo značky umožňující připojení externích souborů se styly nebo JavaScriptové knihovny. [15]

Titulek vkládáme pomocí párové značky `<title>` a při spuštění webové stránky se jeho obsah zobrazí v jejím záhlaví. Jak je vidět z obrázku 21, každá stránka může tedy mít svůj vlastní popis:

`<title>`

```
<title>O HTML5</title>
```



Obrázek 21 - Ukázka praktického použití

`<meta>`

Metaznačky jsou nepárové elementy, které mohou sloužit k nejrůznějším účelům. Význam konkrétní metaznačky pak určuje parametr *name*. Často se používají například pro identifikaci autora, vložení stručného popisu obsahu nebo klíčových slov, které usnadňují práci internetových katalogů a vyhledávačů. Téměř povinností je pak metaznačku použít pro definování kódování a znakové sady, kterou projekt využívá. Jak vidíme, metaznačky jsou velmi důležitou součástí stránek, ačkoliv po překladu zobrazení ve webovém prohlížeči téměř nejsou vidět. [15] [21]

Informace o autorovi:

```
<meta name="author" content="Hegnerová Kateřina">
Popis obsahu stránek:
<meta name="description" content="Stránky zaměřené na HTML5">
```

Klíčová slova charakterizující náš dokument:

```
<meta name="keywords" content="HTML5, CSS, XML, XHTML, JavaScript">
```

Kódování:

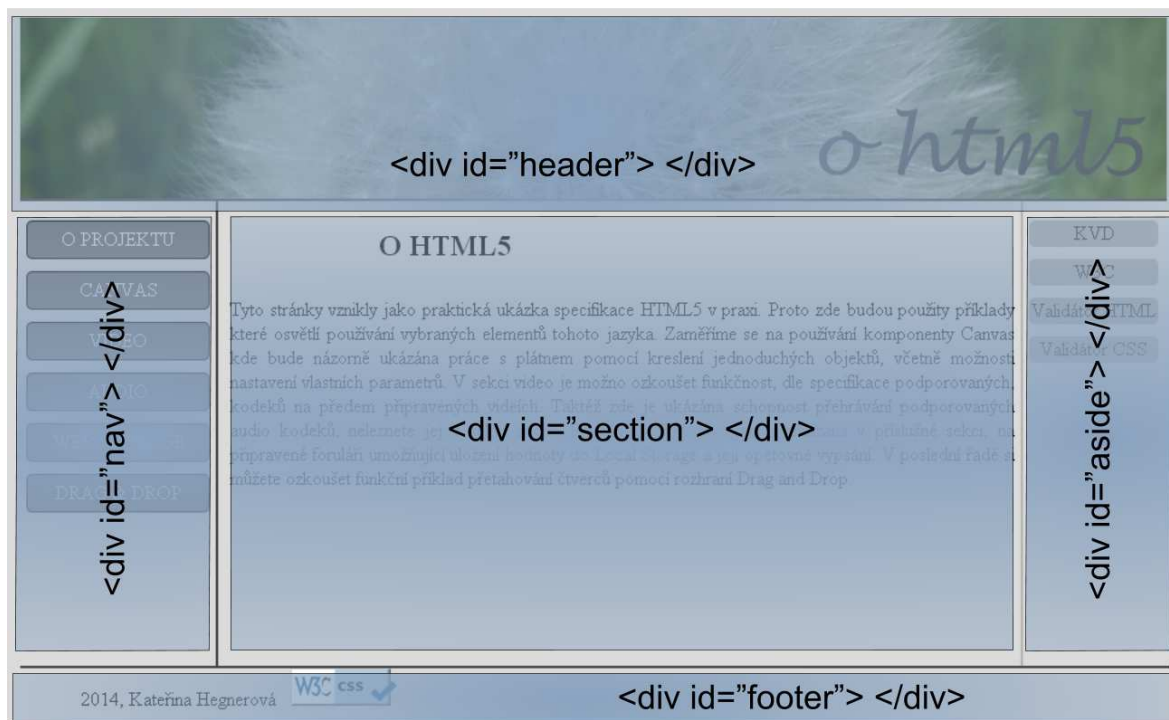
```
<meta http-equiv="content-type" content="text/html; charset=UTF-8">
```

Připojení externího dokumentu s kaskádovými styly:

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

4.1.3 BODY

Tato párová značka zapouzdřuje část určenou pro viditelný obsah a lze o ní říci, že co není uvedeno uvnitř, nebude na stránce viditelné. Samotná značka již nepodporuje žádný z atributů, se kterými se setkáváme v dřívějších verzích HTML, kdy zde bylo možné například definovat barvu pozadí. Úpravy designu, jak je vidět na obrázku 22, je třeba realizovat pomocí kaskádových stylů. Tělo dokumentu je často členěno do několika sekcí, jež ulehčují orientaci nejen prohlížečům. Tato úprava spočívá v používání párové značky `<div>` a následně její stylováním. To provádíme pomocí třídy kaskádových stylů či unikátního identifikátoru, jednoznačně odkazujícího na užitý kaskádový styl. Třídy a identifikátory kaskádových stylů zpravidla pojmenováváme podle účelu jejich použití. Například `<div id="menu">` prohlížeči říká, jaký kaskádový styl má aplikovat na obsah zapouzdřený touto párovou značkou, a vývojář zase při pozdějších úpravách kaskádových stylů lehce nalezne grafickou definici oblasti určené pro navigační prvky. [15]



Obrázek 22 - Rozvržení dokumentu dle HTML4.01

4.2 ZÁKLADNÍ STRUKTURA DOKUMENTU HTML5

HTML5 využívá stejnou základní strukturu členění HTML dokumentů jako všechny předchozí verze jazyka HTML (doctype, hlavička a tělo). Definice struktury těchto částí ale vychází z dlouhodobých zkušeností vývojářů a ve většině případů se snaží o zjednodušení a zpřehlednění. K největším změnám došlo v části určené pro obsah (tělo), kdy HTML5 ve své specifikaci zavádí nové prvky umožňující přehlednější členění obsahu a navigace. To přináší výrazné usnadnění práce s HTML dokumentem jak pro samotného vývojáře, tak i pro webové prohlížeče, kterým je tím umožněno snadnější a rychlejší vyhledávání.

Změny, které HTML5 přináší, podrobněji představím v následujících několika podkapitolách.

4.2.1 DOCTYPE

Změnou si prošla i specifikace typu dokumentu DOCTYPE, ve kterém došlo k radikálnímu zjednodušení poměrně komplikovaného zápisu použité verze jazyka HTML. Trojice již výše popsaných typů – Strict, Transitional a Frameset, byla nahrazena jediným. [19] Zatím co dříve se uvádělo například:

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


V HTML5 není číslo verze nutné, protože se předpokládá neustálý vývoj a plně postačuje:

```
<!DOCTYPE html>
```

4.2.2 HEAD

I přesto, že se jazyk HTML neustále vyvíjí, podoba elementu *head* se téměř nezměnila. Pozastavíme se pouze u způsobu definování znakové sady a použitého kódování, které prošlo velmi přínosnou inovací. [1]

Dříve bylo nutné jej zapisovat následujícím způsobem:

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

Nyní je dostačující pouze:

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

4.2.3 BODY

Odborníci komunity W3C provedli celou řadu analýz webových stránek, které vedly k závěru, že velké množství stránek je tvořeno opakujícím se vzorem struktury stránky. Proto byly do specifikace přidány elementy, které jednoznačně popisují jeho obsah a zvyšují přehlednost celé stránky. Jedná se především o blokové elementy *section*, *article*, *header*, *nav*, *aside* a *footer*. [22]

<header>

Součástí téměř jakýchkoliv stránek jsou vždy informace o webu, na kterém se právě nacházíme, doprovázené většinou úvodní fotografií umístěnou v horní části stránky (viz obr. 23). Pro jasné rozpoznání této části stránky slouží právě tento element *<header>*. Jedná se o párovou značku a uzavírá vše, co v záhlaví má být. Tento element se také používá při psaní obsahu v elementu *<article>* jako záhlaví článku [1] [22]:

```
<header>

</header>
```



Obrázek 23 - Vložení obrázku do elementu header

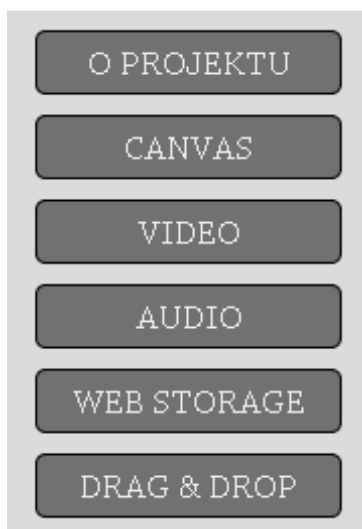
<nav>

Prvek `<nav>` je ve specifikaci HTML5 novým elementem umožňující označení a seskupení odkazů sloužících k navigaci mezi jednotlivými kategoriemi, do kterých bývá projekt členěn. Element může obsahovat odkazy, formátovací značky nebo seznamy. Pomocí CSS stylů lze definovat jeho pozici a design. Samotný účel, pro který byla tato značka vytvořena, jej však neumožňuje využívat pro vložení obsahu, který je pro uživatele zavádějící (například reklamní bannery), ačkoliv technicky to možné je. [1] [22]

Použití tohoto elementu je vysvětleno na tomto příkladu:

```
<nav>
<ul>
<li><a href="index.html">O projektu</a></li>
<li><a href="pages/canvas.html">Canvas</a></li>
<li><a href="pages/video.html">Video</a></li>
<li><a href="pages/audio.html">Audio</a></li>
<a href="pages/web_storage.html"><li>Web Storage</a></li>
</ul>
</nav>
```

V příkladu je menu vytvořeno použitím jednoduchého seznamu odkazů, umožňujícího procházení webu. Celý seznam je pak uzavřen v obalové párové značce `<nav>`. Na obrázku 24 je ukázáno použití kaskádových stylů:



Obrázek 24 - Výsledná podoba menu po použití kaskádových stylů

<section>

Tento obalový element je také novinkou ve specifikaci HTML5. Jeho úkolem je sdružovat obsah podobného obsahu. A proto vše, co bude uvnitř tohoto elementu,

bychom měli být schopni pojmenovat jedním vše říkajícím nadpisem. Pod tuto značku můžeme zahrnout v podstatě veškerý obsah tj., jak obrázky, videa či grafy, tak i text či tabulky. Implementaci provedeme, jak je uvedeno níže, uzavřením požadovaného obsahu do `<section></section>` [1] [22].

```
<section>
<article>
<h2>Web Storage</h2>
<p>Ukládání a vytváření dat v prohlížečích bylo a je nedílnou součástí
při procházení...</p>
</article>
<article>
<h2>Canvas</h2>
<p>je novou komponentou jazyka HTML. Byla implementována ve
specifi...</p>
<video>Video obsah</video>
</article>
</section>
```

<article>

Pro vkládání textového obsahu je možné použít mnoho stylových elementů. HTML5 představuje nový element `<article>`, který umožňuje textový obsah, včetně jeho formátování, jednoduše sdružit pod jeden element, a umožnit tak vyhledávačům jeho snadné nalezení. [1] [23] [22] Takto lze element použít:

```
<article>
<h1>Nadpis</h1>
<p>Odstavctextu.</p>
</article>
```

<aside>

Při procházení většiny webových dokumentů se setkáváme s nejčastějším uspořádáním stránky, kdy je zakomponován i postranní panel. Tento element se nazývá `<aside>` a udává prostor na stránce, který je vyhrazen například reklamám. Dále jsou zde umístěny odkazy na podobné články, jiné potřebné informace, nebo jen doplňující informace k článku. Použití této značky není povinné, plánujeme-li však ve webovém dokumentu prezentovat výše uvedený obsah, použití je doporučeno. [1]

<footer>

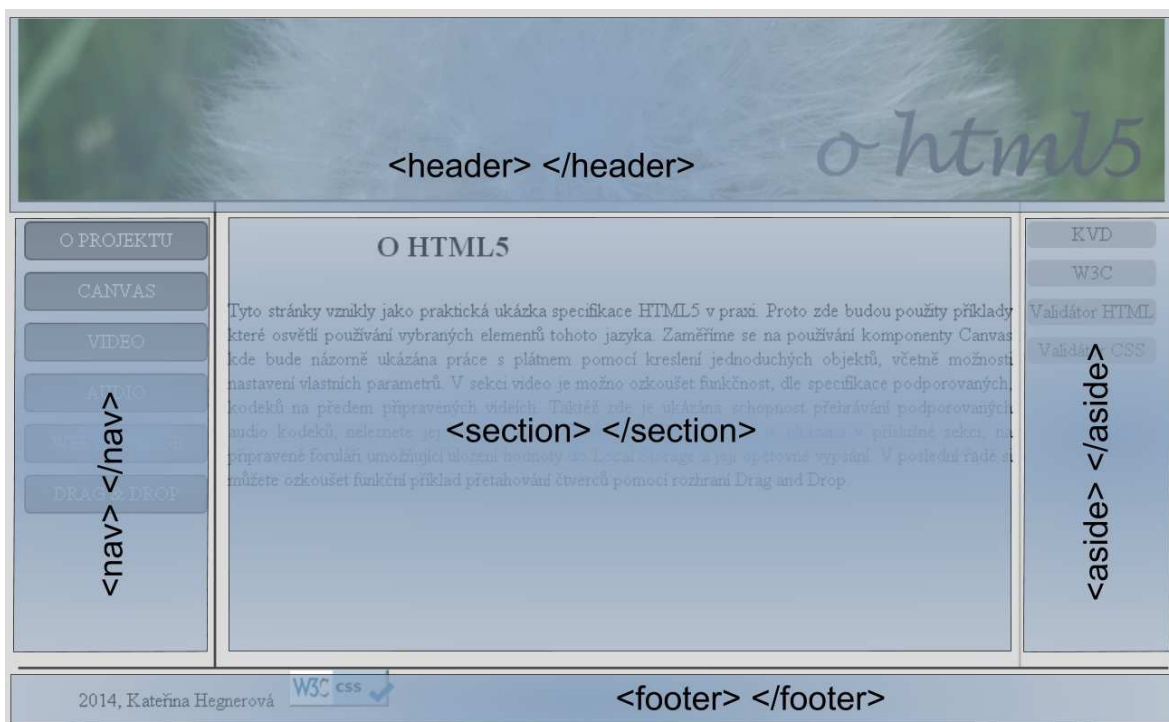
Toto je poslední strukturální prvek, který by dle HTML5 neměl v HTML dokumentu chybět. V zápatí nejčastěji můžeme nalézt informace o tvůrci webového dokumentu, je zde prostor i pro vkládání počítadla návštěvnosti, hodnocení validátoru, či zde můžeme

zobrazit použité zdroje (viz obr. 25). Na rozdíl od předchozích verzí, kde bylo nutné tyto věci zakomponovat do obsahu stránek, je toto snadné a elegantní řešení (viz obr. 26). [23] [22]

```
<footer>2014, Kateřina Hegnerová</footer>
```



Obrázek 25 - Element footer na stránce



Obrázek 26 - Rozvržení dokumentu dle HTML5

5 WEBOVÉ STRÁNKY ILUSTRUJÍCÍ VYUŽITÍ TECHNOLOGIE HTML5

Účelem této kapitoly je představit a popsat webové stránky, které vznikly jako ukázka sloužící pro demonstrování funkčnosti HTML5 (viz příloha DVD). Kapitoly jsou členěny dle logického uspořádání stránky, tj. na část grafickou (prezentační) a logickou, jež řeší funkčnost skriptů a stránky jako takové.

5.1 ROZVRŽENÍ STRÁNKY

Grafický vzhled celé webové prezentace se zpravidla snažíme navrhnout tak, aby následné procházení bylo pro uživatele co nejpříjemnější. Jak například ukazuje volba šířky stránky, kterou se snažíme optimalizovat pro zobrazení na různých monitorech a zařízeních s cílem co nejvíce eliminovat nutnost použití horizontálního posuvníku. Stejně tak je důležité zamyslet se nad umístěním ovládacích a navigačních prvků, podobou záhlaví, menu a podobně. Protože se jedná o věc, která se špatně předělává v konečné fázi tvorby stránek, musíme si toto členění rozvrhnout předem. V mém dokumentu jsem zvolila šířku celého dokumentu 950 pixelů, tuto velikost jsem ozkoušela a doladila tak, aby se i na zařízeních (ne telefony) s menším rozlišením nezobrazoval horizontální posuvník, který značně ruší prohlížení stránek. Dalším krokem je navrhnout si uspořádání prvků na stránce, tzv. layout. Při návrhu bychom neměli brát v úvahu jen estetickou stránku věci, ale i funkčnost. Proto musíme dbát na to, aby jeden prvek nebyl rušen druhým. Ovládání stránek(y) musí být srozumitelné a nesmí docházet k tomu, že uživatel ze samého hledání zapomene, co vlastně hledal. Dokument by měl být navržen tak, aby z jakékoliv hierarchické struktury byl umožněn přístup kamkoliv jinam. Barevnost by neměla rušit samotný obsah sdělení, proto jsem zvolila klasické uspořádání viditelné na většině webových stránek, menu umístěné vlevo, vpravo obsah rychlých odkazů na potřebné webové dokumenty, které jsou tematicky laděny, záhlaví a zápatí. Pro názorné vysvětlení rozvržení HTML5 dokumentu jsem do návrhu zakomponovala i postranní panel (*aside*) s volbou rychlých odkazů. V první řadě je nutné si uvědomit, že webový dokument je prostředek, který umožňuje prezentaci našich sdělení, ale i přesto veškeré dění v dokumentu realizuje uživatel. Pokud ho stránka nezaujme během několika vteřin, stránku opustí. Snažíme se tedy o co nejzajímavější obsah, který je doplněn nadpisy, které stránky velmi zpřehlední a jasně sdělí, co se v článku ukrývá. Dalším zpřehledněním dosáhneme vložení popisku obrázku `<title="Popis po najetí myši.">`, text se zobrazí po

najetí myši na element. Často na stránkách vidáme oživení, a to v podobě použití různých fontů. Chceme-li ponechat přehlednost dokumentu, nedoporučuje se používat více jak dva druhy fontu. Stejně tak i použití různých kreativních písem, některé vypadají pěkně a nabízí se pro použití, ale musíme mít na paměti, že nikdy nemůžeme vědět, jaké fonty jsou nainstalovány na klientském počítači. Pak se může stát, že se text zobrazí v základním používaném fontu, zobrazí se místo efektního písma obrázky, v nejhorším případě se text nezobrazí vůbec. Potom tedy, není-li to opravdu nutné, neměníme defaultní nastavení fontu a můžeme použít tyto základní písma: Times New Roman CE, Arial CE a Courier. Dále je žádoucí, aby obsah stránek byl čitelný, pokud možno i při rozlišné volbě velikosti písma v prohlížečích.

5.2 GRAFICKÉ ZPRACOVÁNÍ DOKUMENTU

Barevnost i rozvržení a umístění, v podstatě veškeré grafické zpracování webového dokumentu určuje, jak dlouho dobu (a zda vůbec) uživatel věnuje pozornost našim stránkám. Proto je vhodné zvolit korespondující barvy a prvky, které nebudou procházení rušit. Design stránek byl realizován pomocí kaskádových stylů verze 3 (CSS3), která přináší do tvorby stránek několik usnadnění, například tvorbu kulatých rohů - tuto grafickou novinku jsem využila u navigačního menu. V předchozích verzích tato vlastnost není k dispozici a kulatých rohů bylo možné dosáhnout jen vložením patřičného obrázku. Jedná se o hodnotu *border-radius*. Hodnotu lze udávat v pixelech, čím je vyšší, tím zakulacenější rohy jsou. Navigační menu je implementováno pomocí HTML seznamu, jako seznam je i stylován. Font je ponechán neměněn, tedy náleží základní znakové sadě prohlížeče. Efekt zvětšení písma je řešen taktéž pomocí kaskádových stylů. Ty umožňují použití tzv. pseudotříd, které na určité prvky přidají speciální vlastnosti. Jednou z nich jsou dynamické pseudotřídy aplikované na odkazy, zejména například *a:hover* zaručující změnu parametrů po najetí myši. Dokument dodržuje členění obsahu dle specifikace HTML5. Všechny blokové elementy tvořící layout stránky jsou odděleny přímkou v černé barvě, ta doplňuje příjemně stránky a jasně odděluje bloky dat. Protože se jedná o stránky představující HTML5, po spuštění ve starších prohlížečích Internet Exploreru se nezobrazily kaskádové styly přiřazené novým blokovým prvkům. Ty v těchto prohlížečích nejsou podporovány.

5.3 LOGICKÁ ČÁST DOKUMENTU

Při návrhu webových stránek jsem se rozhodla jednotlivé kapitoly rozčlenit do samostatných odkazů. Tento navigační prvek je umístěn v elementu `<nav>` a je tvořen pomocí seznamu, který je dále stylován. Menu je viditelné na každé stránce, je tedy možno se lehce přemístit do jiné sekce. Adresářová struktura stránek je logicky členěna do adresářů: `img`, `pages`, `scripts` a `source`. Samotný index je v kořenu, stejně tak i soubor s kaskádovými styly. Adresář `img` obsahuje jediný obrázek, který je použit v záhlaví stránky. Všechny použité stránky, jenž se zobrazí po kliknutí na patřičný odkaz v menu, jsou uloženy v adresáři `Pages`. Jedná se o stránky, které prezentují vybrané komponenty specifikace HTML5.

5.3.1 PAGES

Audio.html

V této stránce testuji podporu audio kodeků v prohlížečích. Po rozkliknutí je možno pustit krátkou ukázkou zvukového souboru ve formátech: OggVorbis, MPEG3 a AAC. Na stránce naleznete i jeden video soubor, který prokazuje právě funkčnost AAC kodeku, jenž je podporován jen jako audio stopa video záznamu.

Video.html

Do dokumentu jsem umístila videa ve, dle specifikace, podporovaných formátech: AggTheora, VP8 a H.264. Vídeu lze klasicky spouště kliknutím na play na ovládací liště. Podporu lze ověřit puštěním této stránky v různých prohlížečích.

Canvas.html

Tato komponenta umožňuje tvorbu grafického obsahu v reálné interaktivně s uživatelem. Jako ukázkou jsem zvolila jednoduché malování pomocí obdélníku a kruhu. U obou objektů je možno navolit si souřadnice zobrazení, barvu vykreslovaného objektu a dle druhu i další parametry. Vykreslování probíhá v elementu `canvas`. Volba vstupních parametrů je realizována pomocí formuláře. Formulář obsahuje elementy typu `input`, výběr objektu je realizován prvkem typu `checkbox`. Pro zadání šířky a výšky obdélníka jsem použila `input` typu `number`. Stejně je tomu tak i při zadávání poloměru kruhu. Objektům je možno pomocí posuvníku, `input` typu `range`, nastavit pozici, na které se na

plátně vykreslí. Pro výběr barvy je nutné použít `<input type="color">`: prvek na místo vloží tzv. colorpicker, ten se otevře buď prohlížečový nebo systémový. Hodnotu lze zadat i ručně. Všechny formulářové popisy jsou vloženy pomocí prvku `<label for="kruh">`, jenž umožňuje přiřadit popisek k danému elementu. Podporu některých nových formulářových prvků se mi nepodařilo prokázat, např. input pro vložení barvy, který nefunguje v prohlížečích Safari 5.1.7 a některých verzích Internet Exploreru, lze tam ale uvést slovní hodnotu barvy (např. *red*). Local Storage funguje ve všech testovaných prohlížečích, v Internet Exploreru verze 11 je nutné pro správné fungování webový dokument testovat online.

Web_storage.html

Stránka se zabývá funkčností komponenty Local Storage. Local Storage je, stejně jako v případě obyčejných cookies, podporován ve všech prohlížečích. Zvyšuje velikost úložného prostoru až na 5MB. Součástí stránky je jednoduchý formulář demonstrující funkčnost této komponenty. Do formuláře je nutné zadat klíč a hodnotu tohoto klíče. Takto jsou data do Local Storage ukládána. Je-li zadáno více dat, je možné zobrazit počet uložených položek. Dále je do formuláře implementována možnost mazání uložených položek. V neposlední řadě lze hodnotu, dle zadaného klíče, vypsát.

Drag.html

Drag and Drop představuje rozšíření v ovládání stránek. Tento způsob ovládání je znám z klasického prostředí desktopových systémů. V prostředí internetu se jedná o významný posun kupředu v ovládání aplikací. Na stránce je možno vyzkoušet si funkčnost Drag and Drop pomocí přemísťování prvků.

5.3.2 SCRIPTS

Adresář obsahující ve stránkách použité skripty. Jedná se zejména o Canvas, Local Storage a Drag and Drop.

canvas_kresli.js

V tomto scriptu se nachází metody realizující vykreslování objektů, zejména obdélníku a kruhu. Metoda `functionkresliKruh()`; je vyvolána událostí onclick a obstarává vykreslení kruhu na plátno. Ta obsahuje sadu metod, jenž nastaví vykreslení obrysově

čáry (implicitně je černá) a barvu výplně (je zadávána uživatelem). Metoda *context.arc()*; kreslí samotný kruh. Pro vykreslení obdélníku jsou metody obdobné, o vykreslení se stará metoda *kresliObdelnik()*. Podrobná ukázka kódu je dostupná v elektronické příloze na DVD.

Canvas.js

Volání funkcí realizující logiku rozhodování. Jsou zde nadefinovány dvě metody. *vyberTvar()*; ta se provede po stisknutí tlačítka „Vložit na plátno“, v ní se volají výše uvedené metody zajišťující vykreslení daných objektů. *ClearCan()*; je volána při použití tlačítka „Smazat“, vymaže veškerý obsah plátna, který byl doposud nakreslen.

Drag_drop.js

Obsahuje tři metody realizující Drag and Drop. Tyto metody jsou podrobně popsány v kapitole 3.4.

Storage.js

Skript skládající se ze čtyř metod realizující práci s Local Storage. *Save()*; uloží zadané hodnoty z pole input polí do Local Storage (klíč, hodnota). Metoda *load()*; umožní po kliknutí na příslušné tlačítko a zadání klíče, vypsát příslušnou hodnotu. Skrip umožňuje i zjistit aktuální počet uložených položek - *numberItem()*; Nakonec *clearLoc()*; realizuje smazání veškerých údajů v úložišti, po vyprázdnění je vypsána hláška o úspěšném provedení operace.

ZÁVĚR

Snahou této bakalářské práce bylo představit nové možnosti návrhu webových stránek pomocí HTML5. Otestovat podporu komponent používaných ve specifikaci HTML5 a zjistit rozdílnosti mezi starší a novější specifikací jazyka. V úvodu se práce věnuje historii, bylo zajímavé sledovat důvody a příčiny vedoucí ke vzniku jazyka HTML5.

Dále byly představeny nejzajímavější komponenty specifikace jazyka HTML5. Řešení pomocí těchto rozhraní je velice přínosné nejen pro návštěvníka webových stránek, ale i samotného tvůrce webu.

V další části dokumentu byla testována kompatibilita vybraných komponent. Testování bylo velice přínosné pro ověření kompatibility, ale bylo mnohdy provázeno řadou problémů. Zejména funkčnost audio a video obsahu se oproti specifikaci značně lišila. Na rozdíl od toho je práce s komponentou Canvas velice snadná.

Další kapitola mé práce se věnuje testování podpory nových blokových elementů tvořící strukturu dokumentu. Vše jsem testovala v běžně dostupných prohlížečích. Potvrdilo se, že podpora nové specifikace jazyka není podporována v těch starších a překvapivě je mnohdy problematická i v současných verzích.

Při zpracování poslední části práce se nevyskytly žádné problémy. Webové stránky, které jsou výstupem, snadno a přehledně prezentují použití elementů jazyka HTML5. Zejména práce s JavaScriptem se ukázala nepostradatelnou a snadnou součástí tvorby většiny interaktivních doplňků moderního webu. Spolu s kaskádovými styly se nová verze specifikace ukázala jako přínosným prostředkem pro tvorbu webových dokumentů.

RESUMÉ

The purpose of this bachelor's thesis is briefly and clearly introduces the new HTML5 language specification. The work is a component of which is language enriched. The performance is realized by a more detailed presentation of each of them. From the previous chapter were selected several important elements and are described by working with them. For each element is characterized by what to do to make it possible to use and last but not least, worked as it should. Marginally I describe the operation and functioning components Canvas. At present, it is often used in Web documents and nothing prevents its further development and use. Most problematic, nowadays, is to support audio and video codecs in different browsers. Nevertheless, this problem is solvable. Another part of this study is comparing the differences in the structure of web document earlier versions of HTML and HTML5 versions. Indicates work with new block elements. The whole issue is described in detail and accompanied by illustrating examples. At the end created a website that meets all preconditions HTML5. Part of these pages are practical examples of components from previous chapters. Pages are enriched with Cascading Style Sheets version 3. It is still recognize that support for the latest version of HTML is still not sufficient.

SEZNAM LITERATURY

1. LUBBERS, P. B. ALBERS a F. SALIM. *HTML 5 Programujeme moderní webové aplikace*. Brno: Computer Press, a.s. 2011. ISBN 978-80-251-3539-6.
2. World Wide Web Consortium (W3C). *A Little History of the World Wide Web* [online]. 10. 7. 2011 [cit. 2014-03-02]. Dostupné z: <http://www.w3.org/History.html>
3. Web Hypertext Application Technology Working Group (WHATWG). *History* [online]. 3. 3. 2014 [cit. 2014-03-04]. Dostupné z: <http://www.whatwg.org/specs/web-apps/current-work/multipage/introduction.html#history-1>
4. ML. J. M. <http://www.root.cz>. In: *Videohovory v prohlížeči: WebRTC čeká ještě trnitá cesta* [online]. 10. 5. 2013 [cit. 2014-02-10]. Dostupné z: <http://www.root.cz/clanky/videohovory-v-prohlizeci-webrtc-ceka-jeste-trnita-cesta/>
5. KASÍK, P. <http://technet.idnes.cz/>. In: *Konečně videokonference pro lidi: zdarma, nic neinstalujte, stačí odkaz* [online]. 7. 3. 2014 [cit. 2014-03-10]. Dostupné z: http://technet.idnes.cz/videokonference-pres-prohlizec-appear-in-f13-/sw_internet.aspx?c=A140306_175305_sw_internet_pka
6. W3C. World Wide Web Consortium (W3C). In: *Web application* [online]. 4. 2. 2014 [cit. 2014-02-19]. Dostupné z: <http://www.w3.org/TR/html5/webappapis.html>
7. SMOLA, M. <http://www.root.cz>. In: *Čtení souborů javascriptem pomocí FILE API* [online]. 22. 8. 2012 [cit. 2014-02-02]. Dostupné z: <http://www.root.cz/clanky/cteni-souboru-javascriptem-pomoci-file-api/>
8. W3C. World Wide Web Consortium (W3C). In: *Web Storage* [online]. 30. 7. 2013 [cit. 2014-03-02]. Dostupné z: <http://www.w3.org/TR/webstorage/>
9. World Wide Web Consortium (W3C). *Geolocation API Specification* [online]. 24. 10. 2013 [cit. 2014-01-05]. Dostupné z: <http://www.w3.org/TR/geolocation-API/>
10. W3C. World Wide Web Consortium (W3C). In: *Gamepad* [online]. 25. 2. 2014 [cit. 2014-02-27]. Dostupné z: <http://www.w3.org/TR/gamepad/>
11. Web Hypertext Application Technology Working Group (WHATWG). *Forms* [online]. 1. 4. 2014 [cit. 2014-04-02]. Dostupné z: <http://www.whatwg.org/specs/web-apps/current-work/multipage/forms.html#forms>
12. W3C. World Wide Web Consortium (W3C). In: *Web Workers* [online]. 1. 5. 2012 [cit. 2014-02-06]. Dostupné z: <http://www.w3.org/TR/workers/>
13. PILGRIM, M. <http://kniha.html5.cz/>. In: *ŘÍKEJME TOMU (PLOCHA NA) KRESLENÍ* [online]. [cit. 2013-12-10]. Dostupné z: <http://kniha.html5.cz/canvas.html>
14. SILVIA, P. L. LAZARIS a E. WEYL. *HTML5 audio a video: kompletní průvodce*. Brno: Zoner Press, 2011, 350 s.. ISBN 978-80-7413-147-9.
15. HLAVENKA, J. et al. *Vytváříme WWW stránky*. Brno: CP Books, a.s. 2005. ISBN 80-251-0801-5.
16. W3C. World Wide Web Consortium (W3C). In: *Grouping content* [online]. 4. 2. 2014 [cit. 2014-02-10]. Dostupné z: <http://www.w3.org/TR/html5/grouping-content.html#grouping-content>

17. W3C. World Wide Web Consortium (W3C). In: *Drag and drop* [online]. 4. 2. 2014 [cit. 2014-04-19]. Dostupné z: <http://www.w3.org/TR/html5/editing.html#dnd>
18. PILGRIM, M. a H. MARTIN. In: *Dive into HMTL5* [online]. [cit. 3-02-2014]. Dostupné z: <http://kniha.html5.cz/>
19. World Wide Web Consortium (W3C). *Document Type Definition* [online]. 24. 12. 1999 [cit. 2013-12-05]. Dostupné z: <http://www.w3.org/TR/html4/sgml/dtd.html>
20. W3C. Index of ElementsWorld Wide Web Consortium (W3C). In: *Index of Elements* [online]. [cit. 2014-04-05]. Dostupné z: <http://www.w3.org/TR/html401/index/elements.html>
21. World Wide Web Consortium (W3C). *HTML/Elements/metav* [online]. 8. 11. 2012 [cit. 2013-12-12]. Dostupné z: <http://www.w3.org/wiki/HTML/Elements/meta>
22. World Wide Web Consortium (W3C). *structural elements* [online]. 14. 3. 2014 [cit. 2014-03-15]. Dostupné z: http://www.w3.org/wiki/HTML_structural_elements
23. LAZARIS, L. E. WEYLOVÁ a A. GOLDSTEINOVÁ. *HTML5 a CSS3 pro webové designéry*. Brno: Zoner Press, 2011, 286 s.. ISBN 978-80-7413-166-0.

SEZNAM OBRÁZKŮ

Obrázek 1 - souřadnicový systém Canvasu	14
Obrázek 2 - Ukázka nakreslených přímek a jejich konců	15
Obrázek 3 - Zobrazení textu na stránce	16
Obrázek 4 - Konečné zobrazení nadpisu ve stránce	17
Obrázek 5 - Ukázka podpory kodeků v prohlížeči Google Chrome	19
Obrázek 6 - Ukázka podpory video kodeků v prohlížeči Opera	19
Obrázek 7 - Tato verze Safari nepodporuje žádný z testovaných kodeků.....	20
Obrázek 8 - Podpora elementu videa v Internet Explorer v. 8	20
Obrázek 9 – MozillaFirefox nepodporuje kodekH.264.....	20
Obrázek 10 -V prohlížeči Opera je podporován jen kodek OggVorbis	21
Obrázek 11 - V prohlížeči MozillaFirefox jsou v podpoře OggVorbis a MPEG3.....	21
Obrázek 12 - Internet Explorer podporuje jen MPEG3.....	21
Obrázek 13 - Prohlížeč Safari této verze nepodporuje žádný z audio kodeků	22
Obrázek 14 - Google Chrome podporuje audio kodeky OggVorbis a MP3	22
Obrázek 15 - Zobrazení elementu hr na stránce	23
Obrázek 16 - Ukázka vlastností seznamů v HTML5	23
Obrázek 17 - Takto vypadá element vyobrazený na stránce	24
Obrázek 18 - Divy bez aplikace Drag and drop	25
Obrázek 19 - Bílý prvek má nastaveno draggable="true", lze ho tedy přetahovat	25
Obrázek 20 - Dokončený přesun pomocí Drag and Drop	26
Obrázek 21 - Ukázka praktického použití	28
Obrázek 22 - Rozvržení dokumentu dle HTML4.01	30
Obrázek 23 - Vložení obrázku do elementu header	31
Obrázek 24 - Výsledná podoba menu po použití kaskádových stylů.....	32
Obrázek 25 - Element footer na stránce	34
Obrázek 26 - Rozvržení dokumentu dle HTML5.....	34