

Univerzita Hradec Králové
Fakulta informatiky a managementu
Katedra informatiky a kvantitativních metod

Využití HTML5 a CSS3 při tvorbě přístupného webu

Bakalářská práce

Autor: Tomáš Skořepa
Studijní obor: Aplikovaná informatika

Vedoucí práce: Ing. Monika Borkovcová

Hradec Králové

srpen 2015

Prohlášení:

Prohlašuji, že jsem bakalářskou práci zpracoval samostatně a s použitím uvedené literatury.

V Hradci Králové dne 18.8.2015

Tomáš Skořepa

Poděkování:

Děkuji vedoucí bakalářské práce Ing. Monice Borkovcové za metodické vedení práce a pedagogickému sboru Speciální základní školy v Poděbradech za poskytnutí výplňových materiálů do praktické části práce.

Anotace

Bakalářská práce se zaměřuje na přínosy HTML5 a CSS3 do oblasti přístupnosti webu. V posloupnosti seznamuje čtenáře se základy problematiky přístupnosti, handicapovanými uživateli Internetu, asistivními technologiemi, metodikou WCAG 2.0 a postupy testování přístupnosti. Pro naplnění zásad metodiky představuje nové techniky HTML5 ve strukturování dokumentů, zpřístupnění obrázků, tabulek a formulářů, přičemž hodnotí jejich podporu ze strany webových prohlížečů a asistivních technologií. Uvádí specifikaci WAI-ARIA, jakožto nástroj pro obohacení přístupnosti, a detailně se zabývá zpřístupněním médií, zejména metodami transkripce a internacionalizace s formátem WebVTT. V souvislosti s použitelností a responzivním web designem popisuje vlastnost media queries CSS3 a využití CSS frameworků.

Výsledkem vlastní práce autora je tvorba webového projektu, na kterém je uvedena řada příkladů z teoretické části a především zachycen kompletní proces tvorby přístupného webu. V závěru práce jsou uvedeny výsledky z testování v majoritních zástupcích desktopových a mobilních prohlížečů, validátoru WAVE a hlasové čtečky JAWS.

Annotation

Title: Utilisation of HTML5 and CSS3 technologies for development of accessible web sites

The focus of this bachelor thesis is to determine the benefits of HTML5 and CSS3 in web accessibility. First of all, the reader is introduced to the basics of accessibility, disabled users on the Internet, assistive technology, WCAG 2.0 methodology and testing procedures of web accessibility. In order to fulfill the principles of the methodology, the author introduces new HTML5 techniques in structuring documents, accessing images, tables and forms while evaluating their support from web browsers and assistive technologies. Furthermore, he states specification WAI-ARIA as a tool to enhance accessibility and he deals in great detail with accessing media, especially methods of transcription and internationalization with

the WebVTT format. With connection to web usability and responsive web design, the feature of CSS3, media queries, is described as well as the utilization of CSS frameworks.

The result of the author's work is the creation of a web project in which he lists numerous examples from the theoretical part, particularly the complete process of creating an accessible web site. The testing results of the most used desktop and mobile browsers, the WAVE evaluation tool and the JAWS screen reader are listed at the end of the author's work.

Obsah

1	Úvod.....	1
2	Cíl práce a metodika práce	2
3	Teoretická východiska	4
3.1	Úvod do přístupnosti HTML5	4
3.2	WAI-ARIA	4
3.3	Kaskádové styly	5
3.3.1	CSS3.....	6
3.3.2	CSS frameworky	6
3.4	Responzivní webdesign	7
3.5	Přístupný JavaScript.....	7
3.6	Stav podpory HTML5 webovými prohlížeči	8
3.7	Použitelnost.....	9
4	Přístupnost a použitelnost.....	10
4.1	Problematika přístupnosti webu	10
4.1.1	Definice přístupnosti.....	10
4.1.2	Přístupnost jako standardní součást webu	11
4.1.3	Motivace k dodržování zásad přístupnosti.....	11
4.2	Handicapovaní uživatelé Internetu	14
4.2.1	Zrakově postižení uživatelé.....	14
4.2.2	Pohybově postižení uživatelé	18
4.2.3	Sluchově postižení uživatelé	18
4.2.4	Uživatelé s poruchami učení a soustředění.....	19
4.3	Asistivní technologie	19
4.3.1	Hlasové čtečky	19

4.3.2	Zvětšovací program MAGic.....	20
4.4	Pravidla a metodiky tvorby přístupného webu.....	21
4.4.1	WCAG 2.0	21
4.4.2	Vyhláška č. 64/2008 Sb.....	22
4.5	Testování přístupnosti	22
4.5.1	Testování nástrojem WAVE.....	23
4.5.2	Testování hlasovou čtečkou JAWS	23
4.5.3	Validátory	23
5	Optimalizace HTML5 a CSS3 pro prohlížeče.....	25
5.1	Strukturování dokumentů	25
5.1.1	Elementy rozdělující obsah	26
5.1.2	Elementy toku obsahu	29
5.1.3	Nadpisy.....	31
5.1.4	Outline algoritmus.....	32
5.1.5	Zástupný obsah.....	33
5.2	WAI-ARIA	33
5.2.1	Orientační role specifikace ARIA.....	33
5.2.2	Stavy a vlastnosti specifikace WAI-ARIA.....	34
5.2.3	Vlastnosti živých oblastí	35
5.2.4	ARIA label a describe	36
5.2.5	Atribut aria-required	37
5.3	Techniky responzivního webdesignu.....	37
5.3.1	CSS3 media queries.....	37
5.3.2	Meta element viewport	38
5.3.3	Mřížkové layouty	38
5.4	Zpřístupnění obrázků	39

5.4.1	Element figure a figcaption.....	39
5.4.2	Popis vizuálně bohatého obrázku	40
5.4.3	Tabulky a grafy	41
5.4.4	Text ve formě obrázku.....	41
5.4.5	Obrázky s funkcionalitou.....	41
5.4.6	Dekorativní obrázky	42
5.4.7	Ikony	42
5.4.8	CAPTCHA	43
5.5	Přístupnost médií a internacionalizace v HTML5.....	44
5.5.1	Popisné video	44
5.5.2	Titulky.....	45
5.5.3	Legendy	45
5.5.4	Transkripce.....	45
5.5.5	Znaková řeč a znakové jazyky	46
5.5.6	Kombinovaná transkripce.....	46
5.5.7	Nativní přehrávač.....	47
5.5.8	Element track.....	49
5.5.9	WebVTT.....	49
5.6	HTML5 a přístupné tabulky	52
5.6.1	Obecná pravidla tvorby tabulek.....	53
5.6.2	Element caption	53
5.6.3	Atribut aria-describedby	54
5.6.4	Kombinace headers/ID	54
5.7	HTML5 a přístupné formuláře	55
5.7.1	Element label.....	56
5.7.2	Typy elementu input	56

5.7.3	Nové formulářové atributy HTML5.....	57
6	Návrh přístupných webových stránek.....	60
6.1	Postup testování	60
6.2	Návrh základní struktury.....	60
6.2.1	Tvorba wireframe.....	60
6.2.2	Sémantika HTML5 v základní struktuře webu.....	61
6.2.3	Definice ARIA rolí	62
6.3	Hlavní navigace	63
6.3.1	Testování.....	65
6.4	Strukturování obsahu	65
6.4.1	Struktura stránky s aktualitami	65
6.4.2	Strukturování článků.....	66
6.5	Dokončení návrhu.....	67
6.5.1	Výběr barev pro grafický návrh s ohledem na přístupnost	67
6.5.2	System mřížky a responzivní webdesign	68
6.5.3	Testování.....	68
6.6	Tvorba legend videa	69
6.6.1	Testování.....	70
6.7	Interaktivní transkripce.....	71
6.7.1	Testování.....	71
6.8	Přístupná tabulka	72
6.8.1	Testování.....	73
6.9	Přístupný formulář	73
6.9.1	Testování.....	74
7	Shrnutí výsledků.....	76
8	Závěry a doporučení	78

9	Seznam použité literatury.....	80
9.1	Literární zdroje	80
9.2	Internetové zdroje	81
10	Seznam obrázků	85
11	Seznam ukázek kódů	86
12	Přílohy.....	1

1 Úvod

Značkovací jazyk HTML je základem pro tvorbu webových stránek již od počátku Internetu a stojí velkou mírou za úspěchem webu. Svou jednoduchostí a podporou u různých druhů zařízení si vydobyl nenahraditelnou pozici a Internet by bez něj nebyl takový, jaký jej známe dnes. Protože oblast webových technologií se stala jednou z nejrychleji se rozvíjejících větví informatiky, byl základ v podobě HTML dokumentu postupně obohacován novými technologiemi, jako jsou kaskádové styly a JavaScript. Z jednoduchých statických webů nastal přerod do komplexních webových aplikací, texty byly doplněny obrázky a multimédií, konzumace webového obsahu se rozšířila ze stolních počítačů na mobilní telefony, televizní obrazovky a tablety.

Když Tim Berners-Lee, zakladatel technologie WWW, definoval hlavní myšlenky webu, neopomenul jeho univerzálnost. Web měl být ze své podstaty přístupný pro každého uživatele Internetu nezávisle na jeho schopnostech a zobrazovacím zařízení. Prudký vývoj webových technologií sice přinesl nové možnosti, zároveň však kladl větší nároky na znalosti tvůrců stránek. V důsledku nesémantického strukturování obsahu, nepřístupně napsaného JavaScriptu a honby za atraktivním vzhledem webu, způsobily rychlé změny řadu potíží všem uživatelům, o to více těm, kteří nevidí, neslyší nebo mají omezené pohybové schopnosti.

Potřeby handicapovaných uživatelů jsou v prostředí Internetu často opomíjeny. Dle každoročních průzkumů organizace WebAIM, přisuzují uživatelé hlasových čteček důvody nepřístupnosti nedostatečnému povědomí webových kodérů o této problematice a neznalosti správných technických postupů.

Dne 28. října 2014 se dočkala finální publikace specifikace HTML5. Ta posunula HTML od relativně jednoduchého značkovacího jazyka k sofistikované platformě pro tvorbu webu a mimo jiné má za cíl nastolit řád ve strukturování obsahu a poskytnout prostředky pro zlepšení přístupnosti. Inspiraci čerpá ze specifikace WAI-ARIA, která napravovala nedostatky předchozích verzí.

HTML5 však samo o sobě k tvorbě webu nestačí a není také jedinou technologií, která může ke zlepšení přístupnosti na Internetu přispět.

2 Cíl práce a metodika práce

Cílem bakalářské práce je se seznámit čtenáře se zásadami tvorby přístupného webu, benefity přístupnosti a představit vybrané techniky, které do této oblasti přinášejí moderní technologie.

Protože rozhodování, jaký postup při tvorbě webu zvolit, není vždy černobílé, musí mít webový kodér obecné povědomí o problematice přístupnosti. Po teoretických východiscích bude tudíž následovat uvedení přínosů přístupnosti a vyvrácení častých mýtů. Představeni budou handicapovaní uživatelé Internetu, jejich konkrétní potřeby a asistivní technologie, které při práci s webem používají. Čtenář bude dále seznámen s existencí metodiky WCAG 2.0 a vyhláškou Ministerstva vnitra ČR pro dodržení zásad přístupnosti. V závěru této části bude popsán postup testování přístupnosti pomocí validátoru WAVE a hlasové čtečky JAWS.

V další obsahové části budou uvedeny postupy správného strukturování dokumentu pomocí nových elementů HTML5 a pro obohacení přístupnosti budou představeny techniky specifikace WAI-ARIA. V souvislosti s použitelností bude vyjádřena myšlenka responzivního webdesignu a použití systémů mřížky. Naznačena bude také jejich implementace s využitím CSS3 vlastnosti media queries a CSS frameworků. Závěrečné kapitoly budou věnovány konkrétním technikám zpřístupnění obrázků, tabulek a formulářů. Protože HTML5 má ambice zlepšit přístupnost multimédií podporou tvorby alternativního obsahu, vynechány nebudou metody transkripce a internacionalizace s formátem WebVTT.

V poslední obsahové části bude představen webový projekt realizovaný v praktické části a podrobně popsány budou řešení typických prvků webu se vztahem k přístupnosti. Na projektu bude uvedena řada příkladů použití technik z teoretické části a testována bude jejich funkčnost v majoritních zástupcích desktopových a mobilních verzí webových prohlížečů. Tato zjištění budou v konečné fázi doplněna o výsledky testování hlasovou čtečkou JAWS.

Z výsledků práce by mělo vyplynout, jaké nové techniky pro vylepšení přístupnosti zmíněné technologie přinesly a jaká je jejich reálná použitelnost.

Vzhledem k povaze tématu práce, vztahující se k přístupnosti webu v souvislosti s moderními technologiemi, je těžké nalézt vhodnou českou literaturu. Na trhu se sice vyskytuje dostatek knih zabývajících se problematikou HTML5 a CSS3, ale přístupnosti věnují z pravidla maximálně jednu kapitolu. Práce proto čerpá zejména z literatury zahraniční a odborných internetových zdrojů. Jmenovitě se jedná o knihy „Pro HTML5 accessibility“ autora Joshue O. Connora, „Accessibility Handbook“ autorky Katie Cunningham a zpřístupnění multimédií se věnuje kniha „HTML5 audio a video“ od Silvie Pfeiffer. Z české scény bude citován zejména David Špinar, autor knihy „Tvoříme přístupné webové stránky“, a zakladatel blogu Poslepu.cz, Radek Pavlíček. Většina použité literatury byla vydána mezi lety 2011 a 2014. Hlavními internetovými zdroji jsou především oficiální dokumenty specifikací organizace W3C.

Práce je primárně určena webovým kodérům a vývojářům, kterým by měla poskytnout přehled v oblasti tvorby přístupných webových stránek. Svým obsahem pokrývá požadavky na znalost oblasti přístupnosti pro profesi webového kodéra dle draftu pro Národní soustavu kvalifikací. Získané znalosti by kodérům měly zjednodušit a zefektivnit práci, ale především pomoci obohatit jejich weby o nové funkce a prvky přístupnosti. Výsledkem jejich práce tak mohou být uživatelsky komfortnější webové stránky, které plně využívají potenciálu prohlížečů a splňují zásady přístupnosti.

3 Teoretická východiska

3.1 Úvod do přístupnosti HTML5

Ačkoli jazyk HTML5 podporují webové prohlížeče v různé míře už celé roky, skrývá v sobě akt vydání jeho finálního doporučení hlubší symbolický význam. Specifikaci HTML5 totiž dělí 15 let od vydání předchozí verze HTML 4.01, což v oblasti vývoje Internetu poslední dekády představuje opravdu dlouhý časový úsek.

Jazyk HTML4 byl snadno srozumitelný a v kombinaci s jinými jazyky, jako jsou kaskádové styly a JavaScript, byli vývojáři schopni realizovat své představy a posouvat hranice interakce s uživatelem. S příchodem Web 2.0 – dynamicky se měnícím obsahem a nárůstem klientského zpracování, narážel vývoj náročných webových aplikací stále častěji na omezení značkovacích jazyků. Pro běžného uživatele představovala vlna nových interakčních modelů mnoho komplikací, tím více pro uživatele zdravotně handicapovaného. [3]

Již dlouho před příchodem HTML5 se proto řada webových kodérů začala zajímat o potřeby těchto uživatelů a hledat cestu ke zpřístupnění obsahu webových stránek co nejširšímu publiku. Na nedostatky předchozích verzí se organizace W3C v HTML5 rozhodla reagovat přidáním mnoha nových elementů k obohacení sémantiky. Značení obsahu webových stránek sémantickými elementy dodává informacím význam pro zobrazovací zařízení uživatelů a ty tak můžou uživateli prezentovat obsah v podobě vhodné právě pro něj. Tato myšlenka je důležitá především u rozličných zařízení, které pro práci s webem používají handicapovaní uživatelé. [11]

Významné změny pro přístupnost přineslo HTML5 v podobě nových elementů do strukturování HTML dokumentů, zpřístupnění obrázků, tabulek, formulářů i multimédií. V práci jim proto bude věnována patřičná pozornost.

3.2 WAI-ARIA

WAI je zkratkou Web Accessibility Initiative, jedné z pracovních skupin W3C, a zkratka ARIA zastupuje Accessible Rich Internet Applications. Specifikace

WAI-ARIA se připojuje k běžně používané kombinaci vývojářů – HTML, CSS a JavaScriptu, aby pomohla zpřístupnit dynamický webový obsah všem uživatelům. Její podpora zatím není kompletní, řada věcí je však použitelná již dnes. [43]

ARIA umožňuje obohatit HTML elementy o atributy s obsahem speciálně určeným pro hlasové čtečky. Protože vývoj ARIA předcházela vzniku HTML5, překrývá některé jeho sémantické prvky. Ne všechny hlasové čtečky podporují HTML5, ale poradí si s ARIA atributy. Pokud chce tudíž vývojář tvořit skutečně přístupné stránky, musí čerpat z obou specifikací. [9]

Díky WAI-ARIA, mohou vývojáři popsat nezvyklé struktury navigací, přidat přístupné panely progresu, které nevidomé informují například o průběhu stahování, zahrnout přístupné posuvníky, pro změnu vstupních hodnot ve formulářích, a vytvořit přístupnou funkcionalitu drag-and-drop. [3]

3.3 Kaskádové styly

Kaskádové styly (zkráceně CSS z Cascading Style Sheets) jsou samostatný jazyk spravovaný organizací W3C. CSS doplňuje jazyk HTML pro definování vizuálního stylu elementů použitých na stránkách. S jeho pomocí dostává kódér pod kontrolu například vzhled a rozmístění fontů, pozadí, okrajů, odsazení a barev elementů HTML. [11]

Jedním z hlavních konceptů CSS je oddělení definic stylů od obsahu strukturovaného pomocí HTML (textů, obrázků a dalších elementů). Jak k CSS dodává Staníček [12]: *„Umožňuje formátovat dokumenty, definuje způsob jejich prezentace na koncových zařízeních, popisuje podobu stránek a styl jednotlivých prvků a přitom nijak neovlivňuje obsah dokumentů samotných.“* Tato vlastnost je velmi důležitá pro fungování hlasových čteček, které CSS téměř úplně ignorují. Po odstranění stylů se původní dokument nezmění.

Charakteristická kaskáda v názvu CSS značí hierarchickou posloupnost v prioritě aplikace stylů. *„Když webový prohlížeč provádí syntaktickou analýzu šablony stylů, postupuje odshora dolů, přičemž přednost dává deklarácím, které najde níže. Pokud tedy spolu dvě deklarace kolidují, později uvedená deklarace přepíše tu dřívější,“* vysvětluje Lazaris [8]. Výchozí styl definovaný prohlížečem je nejprve přepsán

externími styly (obvykle soubor .css). Ty jsou dále přepsány interními styly (typicky definované v elementu `head` celé stránky) a v dokumentu se ještě mohou nacházet styly aplikované na konkrétní element. Poslední slovo má vždy uživatel, který může konečný stav předefinovat styly vlastními. Například k vylepšení přístupnosti používají slabozrací nástroje pro zvětšení písma a lidé se sníženým barvocitem rozšíření pro zvýšení kontrastu. [44]

Z hlediska přístupnosti a asistivních technologií nejsou pravidla pro definování CSS tak zásadní, jako strukturování obsahu pomocí specifikací HTML a WAI-ARIA. Nicméně, správné stylování pomáhá všem návštěvníkům webových stránek k lepší orientaci v obsahu. Vhodné rozmístění prvků webu uvítají především uživatelé s kognitivními poruchami. Využití responzivního webdesignu zase umožní slabozrakým zvětšovat web a volba správného kontrastního poměru mezi textem a pozadím zamezí nutnosti zvyšovat kontrast obrazu lidem s poruchou barvocitu. Při dodržování zásad přístupnosti bude využití CSS jedině přínosem. [3]

3.3.1 CSS3

V současné době prohlížeče plně podporují CSS úroveň 2.1 a organizace W3C pracuje na dokončení specifikace CSS3. Návrh CSS3 se od svých předchůdců odlišuje snahou o modularitu, která spočívá v rozdělení funkcionalit do samostatných standardů nazývaných moduly. Díky tomuto postupu jsou funkcionality CSS2.1 postupně nahrazovány či rozšiřovány moduly CSS3 a kodéři je mohou využívat, aniž by museli čekat na dokončení celé specifikace. [9]

CSS3 přináší mnoho nových funkcí od tvorby animací, transformací, přechodů, nových stylů pro stínování až po sofistikovanější vrstvení pozadí. Z hlediska přístupnosti se často jedná o funkce nepodstatné a při nesprávném použití škodlivé. V práci bude proto věnována pozornost především mřížkovým systémům a využití media queries pro změny velikosti obsahu. [3]

3.3.2 CSS frameworky

K trendům dnešní doby bezesporu patří využívání CSS frameworků. Obecně frameworky slouží k usnadnění kódovací práce tím, že poskytují hotová řešení

k typickým prvkům webových stránek. Využitím CSS frameworku sice šetří kodér čas, ale využívá také prověřených a optimalizovaných postupů.

V praktické části bude využit především systém mřížky frameworku Bootstrap verze 3.3.5 pro vyvážené uspořádání prvků na stránce a základ pro responzivní navigaci.

3.4 Responzivní webdesign

Protože v současnosti stále vzrůstá počet odlišných rozměrů zařízení, na kterých uživatelé prohlíží web, bylo potřeba vývoj stránek na tuto změnu adaptovat. Populárním řešením se stala filosofie responzivního webdesignu a jeho myšlenku vystihuje Lazaris tak, že [8]: *„je to způsob psaní kódu jazyka CSS takovým způsobem, aby se šířka stránky a její obsah přizpůsobily velikosti okna webového prohlížeče, který uživatel používá.“* Kromě stolních počítačů se poté stránky ideálně rozloží na obrazovkách tabletů a smartphonů. [9]

K dosažení tohoto cíle se využívá zejména přelomové vlastnosti CSS3 – media queries neboli dotazů na médium. Bloky s pravidly @media obsahují definice kaskádových stylů, které se aplikují na stránku v závislosti na vlastnostech používaného zařízení. Podle druhu média můžeme měnit uspořádání sloupců stránky a především na mobilních zařízeních ty přebytečné skrýt. [8]

„Dobře vytvořený responzivní web může výrazně pomoci i přístupnosti,“ dodává Pavlíček [33]. Důvodem je, že si uživatel může přečíst obsah pohodlně na libovolném zařízení, protože responzivní design zamezuje vzniku horizontálního posuvníku. Jeho vlastnosti ocení zejména slabozrací uživatelé, kteří pro prohlížení webu používají softwarovou lupu.

3.5 Přístupný JavaScript

Mnoho let byl JavaScript vnímám výhradně jako překážka přístupnosti. Příčinou bylo, že vývojáři používali špatné skriptovací techniky především za účelem oživit své dosavadní stránky. Bariéry, které tím vytvořily, posiloval fakt, že mnoho prohlížečů mělo podporu JavaScriptu defaultně vypnutou a asistivní technologie byly v jeho podpoře pozadu. [4]

V současné době je situace jiná. Asistivní technologie umí těžit z přínosů JavaScriptu a dle posledního průzkumu organizace WebAIM jej má defaultně zapnutý drtivá většina (97,6% respondentů) uživatelů hlasových čteček. [45]

JavaScript umožňuje webovým kodérům obohatit možnosti interakce s uživatelem, zlepšit informační procesy a je-li psán s ohledem na přístupnost, může významně pomoci také handicapovaným uživatelům. [3]

3.6 Stav podpory HTML5 webovými prohlížeči

Nové verze majoritně používaných webových prohlížečů Chrome, Firefox, Safari a Opera a Internet Explorer podporují většinu elementů HTML5. Přesto je vhodné ještě před použitím nového elementu jeho podporu ověřit na některém ze specializovaných webů, jako je například caniuse.com nebo html5accessibility.com. Výsledek bude následně potřeba vždy pečlivě otestovat.

S jistotou však problém nastane u prohlížečů Internet Explorer ve verzi nižší než 9, tedy včetně stále nezřídka používané verze 8. Když prohlížeč narazí na element, jehož sémantiku nezná, zachází s ním jako s řádkovým elementem. Většina sémantických elementů HTML5 je však bloková a prohlížeč tak rozhází strukturu dokumentu. Vývojáři proto přišli s řešením v podobě javascriptového polyfillu HTML5 Shiv. „*Tento skript umožňuje aplikovat kaskádové styly na elementy jazyka HTML5 ve starších verzích prohlížeče Internet Explorer, které tyto elementy běžně nepodporují,*“ vysvětluje Lazaris [8].

Ukázka kódu 1 Deklarace cesty k polyfillům HTML5 Shiv a Respond [zdroj: autor]

```
<!--[if lt IE 9]>  
  <script src="content/js/html5shiv.js"></script>  
  <script src="content/js/respond.js"></script>  
<![endif]-->
```

Obdobný problém činí CSS3, jejichž podporu řeší polyfill Respond.js. Přidání kódu z Ukázky 1 do elementu `head` HTML dokumentu polyfilly zprovozní. Cesty k javascriptovým souborům jsou opatřeny podmínkou, kterou splní pouze prohlížeče Internet Explorer ve verzi nižší než 9. Ostatní prohlížeče budou instrukci k načtení souboru ignorovat. [9]

3.7 Použitelnost

Cílem použitelnosti je co nejvíce zpříjemnit práci uživateli, který web používá. *„Použitelný web je takový web, který se návštěvníkům dobře používá. Kde se dobře orientují, rychle naleznou, co hledají. Kde se neztrácí, nedělají zbytečné chyby. Jsou to weby, ze kterých mají uživatelé dobrý pocit,“* definuje Špinar [13]. Klíčem k dosažení tohoto cíle je, aby uživatel měl možnost procházet web pohodlně na různých druzích zařízení a vždy byla zaručena intuitivnost jeho uživatelského rozhraní a konzistentnost obsahu.

Do použitelnosti se promítá už samotný návrh a design připravovaných stránek či webové aplikace. Kodér by se měl vyvarovat použití nezvyklých prvků, nad jejichž funkcí bude muset uživatel přemýšlet, a standardní prvky umisťovat tam, kde je uživatel očekává. Ve výsledku by měl uživatel pouhým pohledem pochopit, k čemu stránka slouží a jak ji má používat. [7]

Při hledání vztahu mezi přístupností a použitelností je obvykle přístupnost prezentována jako podmnožina použitelnosti. *„Já osobně na přístupnost a použitelnost nahlížím jako na dvě související a vzájemně se ovlivňující množiny sad požadavků/pravidel, kdy tyto množiny mají neprázdný průnik a některá pravidla tak patří do obou skupin,“* vyjádřil svůj pohled na problematiku Pavlíček [36].

4 Přístupnost a použitelnost

4.1 Problematika přístupnosti webu

4.1.1 Definice přístupnosti

Pojem přístupnost můžeme přirovnat k pojmu bezbariérovost, která garantuje vozíčkáři možnost pohybu po budově. V oblasti webů je paralelou například schopnost slabozrakého uživatele používat web k získání hledané informace. Stejně jako vozíčkáři využívají technických prostředků pro překonání schodišť, existují také techniky, které umožňují hendikepovaným uživatelům překonávat nástrahy na webu.

V literatuře lze nalézt mnoho definic přístupnosti. Souhrnnou definici vyjádřil David Špinar v přední české publikaci věnované přístupnosti takto [13]: *„Přístupná webová stránka je použitelná pro každého uživatele Internetu, a to nezávisle na jeho postižení, schopnostech, znalostech či zobrazovacích možnostech.“*

Znamená to, že webová stránka neklade uživateli při používání žádné zásadní překážky. Ze své podstaty by měla být použitelná pro kohokoli, v různých pracovních podmínkách. Bere ohled na uživatele dočasně i trvale hendikepované, na jejich hardwarové a softwarové vybavení a předpokládá možnost práce v prostředí se zhoršenou viditelností (zatemnělou místnost, zhoršenou čitelnost monitoru vlivem svitu Slunce). Kromě uživatelů slabozrakých, nevidomých, tělesně či sluchově postižených pomáhá lidem s poruchami učení či soustředění a bere v potaz jejich znalosti a zkušenosti. [4]

Pro přístup k webu slouží uživatelům nejrůznější technologie či zařízení jako stolní počítače, notebooky, tablety, mobily, ale také technologie specializované právě pro uživatele s nejrůznějším zdravotním postižením. Příkladem může být hmatový displej, počítač s hlasovou čtečkou či softwarovou lupou, nebo ozvučený mobilní telefon. [34]

Kdokoli by měl být schopen web využívat bez ohledu na výše uvedené předpoklady.

4.1.2 Přístupnost jako standardní součást webu

Je důležité si uvědomit, že přístupnost vychází ze samotných základů webu. Není žádným přidaným doplňkem, ale prolíná se systematicky stavbou webu již od jeho struktury. Promyšlený design webových stránek, sémanticky správně tvořená struktura poskytne uživatelům výraznou přidanou hodnotu. Jelikož web je ze své podstaty univerzální, problémy, na které uživatelé při jeho používání narážejí, přivozují sami tvůrci nedodržením stanovených zásad.

Předešlý odstavec poukazuje také na fakt, že možnost číst webovou stránku a ovládat její funkce je základním právem každého uživatele Internetu. Nikomu nesmí být upřena možnost vnímat obsah webu jen proto, že nevlastní optimální hardwarové zařízení, nemá zkušenosti s nejnovějšími technologiemi či trpí některým typem zdravotního postižení. [13]

Úlohou přístupnosti webu je zajistit rovné podmínky pro uživatele s různými schopnostmi. Z tohoto důvodu ukotvilo mnoho zemí přístupnost ve svých právních rádech.

4.1.3 Motivace k dodržování zásad přístupnosti

Mnoho tvůrců webových stránek se domnívá, že úsilí vynaložené na striktní dodržování zásad přístupnosti se nevyplatí s ohledem na počet uživatelů, kteří tuto snahu ocení. Motivací proč je vhodné udržovat svůj web přístupný je však hned několik.

4.1.3.1 Rozšíření spektra návštěvníků

Ten, kdo se v běžném životě s handicapovanými uživateli příliš nese setkává, by mohl nabýt dojmu, že tvoří silně minoritní skupinu uživatelů. Mnoha zrakově postiženým však právě možnost neopouštět pohodlí svých domovů zlepšuje kvality života. Vyřízení úředních žádostí či objednání nákupů elektronicky je pro ně mnohdy snazší než podstupovat rizika pohybu v neznámém prostředí, ale oceňují také jistou formu anonymity, které jim internetové prostředí nabízí. Dopisování či hovor po internetu totiž zastiňuje handicap uživatele, který druhá strana tak nemá šanci zaznamenat. [13]

Některé studie proto dokládají, že uživatelé se specifickými potřebami přístupnosti mohou tvořit až 30% podílu na celkovém počtu uživatelů konkrétní webové stránky. Investice do předělání nepřístupného webu se tedy vyplatí a při vzniku webu nového by tyto úmysly měly být samozřejmostí. [40]

4.1.3.2 Znamka kvality

Nepřístupný web je vizitkou špatně odvedené práce webdesignéra, jelikož se jedná o podstatný prvek webových stránek. Především v západních zemích je jakýkoli akt diskriminace vnímán velmi negativně a může způsobit ztrátu dobré pověsti firmy. Zpřístupnění webu naopak vyvolává pozitivní dojmy z kvalitně odvedené práce tvůrce webu. [3]

4.1.3.3 Lepší viditelnost webu

Nutným předpokladem návštěvy webu uživatelem je, že se dozví, kde ho nalézt. Je běžnou praxí, že uživatelé při hledání informací na Internetu využívají vyhledávačů. Ty se stávají majoritním zdrojem návštěvnosti a je proto důležité, aby roboti, které používají vyhledávače (např. Google) při indexování webových stránek, co nejnáze rozpoznávali obsah webu. [24]

Přístupné stránky jsou takzvaně „robot-friendly“, protože vyhledávače jim dokonale rozumí. Zjednodušeně řečeno, roboti vnímají pouze text stránky, jeho sémantické vyznačení a odkazy. Nerozeznají obsah obrázků či souborů, pokud je o něm neinformujeme. Přístupné stránky zajistí přízeň robotů, protože [13]:

- veškeré informace jsou přítomny v textové podobě nebo v podobě alternativ grafických prvků
- odkazy jsou funkční a robot je bez komplikací používá pro pohyb mezi stránkami
- obsah nezůstává před roboty ukrytý například v JavaScriptu a Flashových prvcích, které lze nyní plnohodnotně nahradit pomocí plně indexovatelných HTML5 prvků
- obsah stránky je promyšleně členěn

- je dodrženo sémantické značení textu, proto robot rozezná název stránky od jednotlivých řádů nadpisů a přiřkládá jim různou prioritu.

Existují i další faktory, ovlivňující pozici webu ve vyhledávačích. Zpřístupnění webu by však mělo být prvotní úlohou před započítáním investice do SEO služeb a tím zlepšení pozice webu ve výsledcích vyhledávání. [42]

4.1.3.4 Lepší použitelnost webu

Přístupný web je použitelný i pro hendikepované. Přístupnost je tedy v zásadě specializací použitelnosti, která má za cíl klást zásady pro tvorbu webu tak, aby jej efektivně mohl používat běžný uživatel. Při orientaci na použitelném webu nemusí běžný uživatel narazit na žádný problém, uživatel postižený však ano. Naproti tomu web přístupný pro uživatele s hendikepem poslouží bez problémů i uživateli běžnému. [4]

Příkladem pravidla, ze kterého těží hendikepovaní mnohonásobně více, ale zároveň slouží k lepší použitelnosti pro majoritní skupinu je třeba zásada rozdělování rozsáhlých obsahových celků do menších částí, oddělování navigačních prvků od informačních, zvýraznění odkazů a jasný popis toho, kam vedou. Naopak obrázky bez alternativních popisků jsou sice použitelné, pro zrakově postiženého však nepřístupné. [13]

4.1.3.5 Podpora základních lidských práv

Postupem času, kdy se z Internetu stávalo stále využívanější informační médium, si mnoho zemí uvědomilo, že spolehnout se na komerční efekt přístupnosti je velmi nejisté. Aby předešli případné diskriminaci, rozhodli se ukotvit zásady přístupnosti ve svých právních rádech. [4]

Konkrétní znění zákonů se liší v jednotlivých zemích, ale také pro různé druhy organizací. *„Někde platí jen pro instituce veřejné správy a samosprávy, jinde pro všechny weby financované z veřejných rozpočtů, jinde platí pro všechny subjekty, které zaměstnávají zaměstnance,“* vysvětluje Špínar [40].

Česká republika se mezi tyto země definitivně zařadila v roce 2008 uveřejněním vyhlášky č. 64/2008 Sb., o formě uveřejňování informací souvisejících s výkonem

veřejné správy prostřednictvím webových stránek pro osoby se zdravotním postižením, ve Sbírce zákonů. S platností od 1. 3. 2008 tedy ukládá zákon institucím veřejné správy povinnost udržovat své weby přístupné tak, aby poskytovaly informace všem občanům bez rozdílu. [37]

4.2 Handicapovaní uživatelé Internetu

Pojem handicap je u nás ve vztahu k lidem s postižením či omezením často vnímán v rozporu s evropskými konvencemi.

„Handicap vzniká v situaci, kdy náročnost prostředí převyšuje fyzické či smyslové schopnosti jedince (např. bariérový přístup, chybějící informace v Braillově písmu, apod.). Není tedy vrozenou či získanou vlastností, ale dočasným stavem způsobeným kombinací určitých specifických potřeb jednotlivce a nevhodně řešeným prostředím, které tyto požadavky nesplňuje,“ uvádí jedna z definic [14].

Početná skupina populace trpí některou z forem vad zraku a pociťují problémy při čtení příliš malého textu či textu umístěném na nevhodně zvolené barvě pozadí. Některé z vad lze korigovat optickými pomůckami, jiné jako třeba poruchy barvocitu ne. Návštěvníci webu mohou pracovat v rušném prostředí s rukou v sádře, velké procento teenagerů trpí poruchou soustředění či učení, zastoupení obyvatel ČR ve věku 65 a více let představovalo ke konci roku 2013 17,4%. Handicapovaných je zkrátka mnoho a tvůrce webu by tento fakt neměl opomíjet. [19]

V následujících kapitolách budou podrobněji představeni uživatelé se specifickými handicapem a bude poukázáno na jejich konkrétní potřeby.

4.2.1 Zrakově postižení uživatelé

Rozlišujeme mnoho druhů a úrovní rozličných postižení zraku. V povědomí široké veřejnosti utkvěla představa, že lidé s těžkým postižením zraku nejsou schopni zužitkovat prostředky tohoto smyslu vůbec. Někteří však jsou schopni zachytit část světla, hran či tvarů a celá problematika nabývá na složitosti. Pro přiblížení rozličných potřeb jsou uživatelé uvedeni v kategoriích tak, jak je dělí Špínar. [13]

4.2.1.1 Zcela nevidomí a těžce zrakově postižení uživatelé

Bezesporu se jedná o nejvíce zohledňovanou skupinu při tvorbě pravidel přístupnosti. Hlavním důvodem je, že web je ze své podstaty především vizuální médium a tito uživatelé se na svůj zrak nemohou spolehnout vůbec. K jejich zohlednění přispívá také fakt, že na jejich práva poukazuje řada sdružení nevidomých. V České republice je jím například Sjednocená organizace nevidomých a slabozrakých (SONS) či občanské sdružení Okamžik.

Podle výzkumu Světové zdravotnické organizace (World Health Organization, WHO) z roku 2010 je poté ve světě nejčastějším důvodem oslepnutí šedý zákal (51%) a zelený oční zákal (8%). [30]

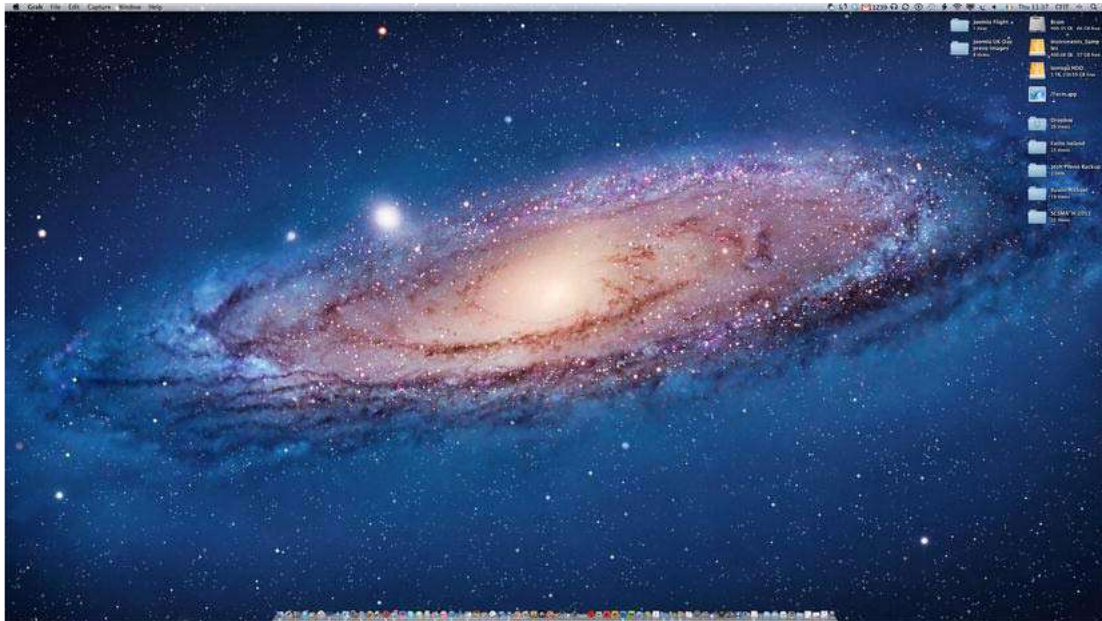
Vývojář by měl mít neustále na paměti, že tito uživatelé používají pro reprodukci obsahu stránky hlasový výstup nebo braillovský řádek a zařízení, kterým stránky ovládají, je nejčastěji klávesnice, nikoliv myš. Fungování hlasových čteček a dalších asistivních technologií bude přiblíženo v samostatné kapitole. [13]

Z hlediska počítačů a webu jsou právě nevidomí skupinou, která reálně nejvíce výtěžila z technologického pokroku. Rozvoj asistivních technologií jim poskytuje lepší začlenění do společnosti a širší okruh uplatnění v zaměstnání.

4.2.1.2 Uživatelé s vadou zraku

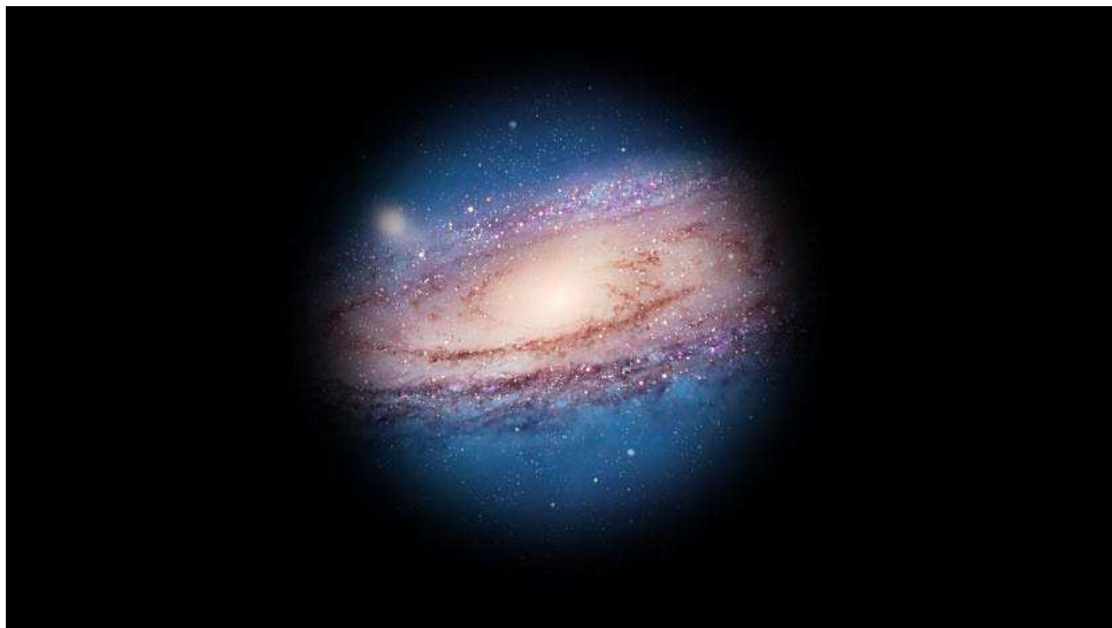
Do této skupiny řadí Špinar [13] uživatele, kteří trpí některou z poruch vidění, ale nepřišli o zrak úplně. Přesto není jejich vadu možné napravit brýlemi či jinými korekčními pomůckami.

Názornější představu o nejčastějších zrakových postiženích se pokouší podat následující simulace na snímcích pracovní plochy.



Obrázek 1 Plocha, pozorovaná uživatelem s relativně dobrým zrakem [3]

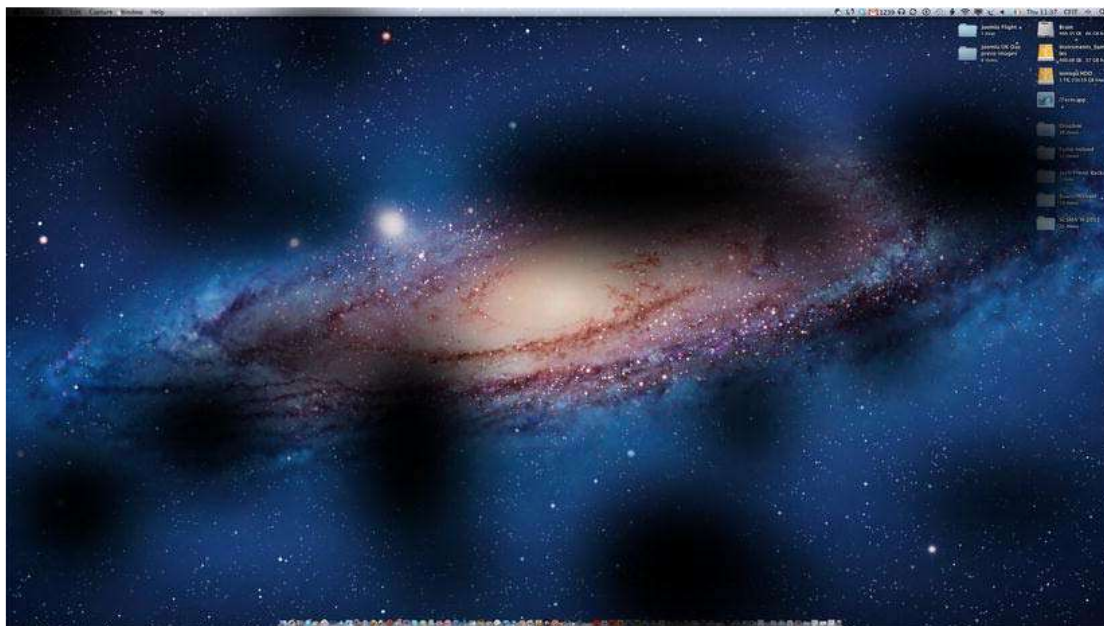
4.2.1.2.1 Zelený zákal



Obrázek 2 Plocha, pozorovaná uživatelem trpícím zeleným zákalem (simulace) [3]

Osoba trpící zeleným zákalem ztrácí periferní vidění a může mít problémy s rozpoznáním změn dynamického obsahu. [3]

4.2.1.2.2 Retinopatie



Obrázek 3 Plocha, pozorovaná uživatelem trpícím retinopatií (simulace) [3]

Retinopatie je označení pro patologické změny sítnice a jejích cév. Projevuje se rozmlženými až neproniknutelnými výpadky v zorném poli. Nejčastějšími příčinami vzniku jsou oběhové poruchy nebo cukrovka. [4]

Pro seznámení s dalšími poruchami je vhodné využít jednoho z mnoha zdarma dostupných simulátorů. Jeden takový připravila společnost Light House, která prostřednictvím filtrů aplikovaných na libovolné video ze serveru YouTube simuluje oční vadu, se kterou se nemocný potýká každý den. [29]

Tito lidé používají pro prohlížení webu zvětšovací programy a vyžadují vhodnou volbu barev popředí a pozadí textu. Pro zvýšení kontrastu často využívají doplňků do webových prohlížečů či funkcí operačních systémů pro přenastavení barev.

4.2.1.3 Uživatelé se sníženým barvocitem

Forem sníženého barvocitu je celá řada. Odhaduje se, že některou z forem trpí 8-10% mužské populace a jedná se tak o nejčastější poruchu zraku. Všeobecně známé jsou Ishimarovy testy, které jsou tvořeny obrazci z kuliček blízkých odstínů určité barvy. Člověk se zdravým zrakem odhalí v obrazci znaky (nejčastěji čísla) tvořené jedním odstínem, člověk se sníženým barvocitem znaky nepřečte.

Postižení mohou mít problémy se splýváním jen určité škály barevných odstínů, úplná barvoslepost je velmi vzácná. [4]

Pro tvůrce webu z těchto vad vyplývá v podstatě vždy totéž. Barvy textu a jeho pozadí musí splňovat dostatečný kontrastní poměr. Praktické využití softwaru pro výběr barev grafického návrhu s ohledem na přístupnost bude předvedeno v praktické části.

4.2.1.4 Dočasně zhoršené viditelnostní podmínky

Může se stát, že uživatel pracuje v ostrém slunečním svitu. Zvláště lesklé povrchové úpravy displejů nedokáží rozptýlit přímé světlo tak, jako matná úprava. Pro prevenci čitelnosti zde opět platí zvýšená pozornost na dostatečný kontrastní poměr barvy textu a pozadí. [13]

4.2.2 Pohybově postižení uživatelé

Rozlišujeme řadu motorických omezení, které se projevují mnoha směry. Může se jednat o postižení přítomné od narození či dočasné postižení způsobené vlivem úrazu (například ruka v sádře). Většinou znemožňuje uživatelům použití standardní myši pro ovládání webu. Tito uživatelé používají specifické typy kompenzačních pomůcek, jako jsou velkoplošná tlačítka, trackbally a speciální klávesnice. V případech úplného ochrnutí se může jednat o zařízení sledující pohyb očí (tzv. eye tracking) či software pro ovládání počítače hlasem. [4]

Nemusíme však testovat použitelnost webu s každou z těchto pomůcek, většina z nich totiž vychází z fungování klávesnice. Majoritní skupinu těchto uživatelů tedy uspokojíme, budou-li veškeré ovládací funkce, odkazy a formuláře na stránkách přístupné použitím klávesnice. [13]

4.2.3 Sluchově postižení uživatelé

V dnešní době nevyužíváme web zdaleka jen jako vizuální médium a měli bychom mít na paměti, že vše, co hodláme sdělit pomocí zvukové složky, musíme nabídnout ve formě textového přepisu. Obsah zvukové stopy může být kromě sluchově postižených nepřístupný také pro uživatele pracujícího v hlučném prostředí či uživatele bez reprodukčního zařízení. [4]

S příchodem audia a videa do HTML započal vývoj tzv. technologií alternativního obsahu. Jejich cílem je dodat obsah audio stopy sluchově postiženému například ve formě legendy či titulků, jako alternativy pro uživatele mluvící jiným jazykem. [10]

4.2.4 Uživatelé s poruchami učení a soustředění

Přestože uživatelů s poruchou učení či soustředění je velmi mnoho, jedná se o nejméně probádanou oblast. Potřeby u konkrétních poruch mohou být značně odlišné a ne vždy je z výzkumu prokazatelné, jaké postupy fungují a jaké ne. [13]

Obecné závěry ukazují, že tito uživatelé potřebují webové stránky se snadno pochopitelnou strukturou, přehlednou navigací, pečlivě členěným obsahem pomocí odstavců a výstižných nadpisů. Především dyslektici ocení vhodnou volbu typografie písma. [4]

4.3 Asistivní technologie

V letech 2011 až 2013 se Ministerstvo vnitra České republiky účastnilo projektu ATIS4all (Assistive Technologies and Inclusive Solutions for All). V jeho výstupech lze nalézt definici pojmu asistivní technologie, která zní [28]: „*Asistivní technologie (AT) je jakýkoliv nástroj, zařízení, software nebo systém, využívající zpravidla moderní technologie (zejména senzory, aktuátory, informační a komunikační technologie) s cílem posílit, udržet nebo zlepšit funkční schopnosti jedinců se speciálními potřebami, a tím jim usnadnit každodenní život a zlepšit kvalitu jejich života, samostatnost a soběstačnost.*“ Za osoby se speciálními potřebami jsou zde považováni senioři, zdravotně postižení a chronicky nemocní lidé.

Asistivní technologie se na rozdíl od asistenčních technologií nespolehnají na asistenci živého člověka (jako například podporu pracovníka dispečinku). Tato práce se bude soustředit výhradně na asistivní technologie spojené s informačními technologiemi.

4.3.1 Hlasové čtečky

Hlasové čtečky jsou využívány především zcela nevidomými a vážně zrakově postiženými uživateli. Pomocí klávesnice umožňují uživateli interagovat

s prostředím operačního systému či webovými stránkami. Hlasová čtečka (neboli screen reader) rozpoznává sémantiku stránky a prostřednictvím zvukového výstupu předčítá uživateli její obsah. Základním předpokladem je, že uživatel se pomocí klávesnice dostane na veškerý obsah (například přeskokováním v navigačním menu pomocí tabulátoru), aby jej screen reader mohl přečíst. [3]

4.3.1.1 Hlasová čtečka JAWS

Celosvětově nejpoužívanější a nejpopulárnější hlasovou čtečkou je screen reader JAWS americké společnosti Freedom Scientific. Poskytuje hlasový výstup či výstup v Braillově písmu pro běžné počítačové programy a oslavil již 20 let své existence. Za tuto dobu stanovil řadu nepsaných standardů ve výpočetní technice pro nevidomé uživatele. [45]

Při načtení stránky začne JAWS lineárně procházet sémantické značky a postupně předčítá zjištěné informace uživateli. K interakci s webovým obsahem používá tzv. virtuální kurzor, proto uživatel nepracuje přímo se stránkou, ale s její virtuální verzí (snapshotem). K ovládání funkcí je využíváno mnoha klávesových zkratk a uživatel má tak možnost například regulovat rychlost čtení nebo postupně procházet všechny nadpisy, tabulky, seznamy, obrázky či formulářové prvky na stránce. Pokud uživatel hledá určitou informaci, nemusí tedy procházet stránku od začátku, ale efektivně přenést kurzor na požadované místo. U prvků jako jsou tabulky nebo formuláře předkládá čtečka informace tak, aby uživatel neztrácel orientaci. [3]

Právě tato čtečka ve verzi JAWS 16 Home Edition bude v praktické části použita pro testování přístupnosti webových stránek. Zkušební verze programu je ke stažení zdarma ze stránek společnosti GALOP včetně počeštěného hlasového výstupu. Ceny plných verzí se pohybují v řádech desítek tisíc korun. [23]

4.3.2 Zvětšovací program MAGic

Kromě vestavěné možnosti přiblížení obrazovky v prohlížečích používají slabozrací uživatelé specializované programy – softwarové lupy. Softwarové lupy

jsou určeny pro zvětšování obrazovky jako celku nebo její části, ale mohou poskytovat řadu doplňkových funkcí.

Jedním ze zástupců softwarových lup je program MAGic, který je v české zkušební verzi opět zdarma dostupný na stránkách společnosti GALOP. Tato lupa umožňuje až šedesátinásobné zvětšení obrazu, zvýraznění karety a kurzoru myši, změnu kontrastu barev a ve verzi s hlasovou podporou používat rovněž hlasový výstup. [22]

4.4 Pravidla a metodiky tvorby přístupného webu

S rozvojem Internetu docházelo k řadě soudních sporů, kdy občané a organizace bojující za práva menšin obviňovali provozovatele webů z diskriminace různých skupin handicapovaných uživatelů. Jako reakce na tyto události vznikaly v průběhu vývoje různé metodiky tvorby webových stránek tak, aby byly bezbariérové. [13]

K prvotnímu počínání z roku 1999, metodice Web Content Accessibility Guidelines 1.0 (WCAG 1.0), od skupiny Web Accessibility Initiative (WAI), utvořené v rámci konsorcia W3C, se postupně přidala také česká metodika Blind Friendly Web (BFW) z prostředí Sjednocené organizace nevidomých a slabozrakých (SONS). Autorem metodiky je Radek Pavlíček, který se podílel také na přípravě Vyhlášky č. 64/2008 Sb. o přístupnosti webových stránek orgánů veřejné správy z roku 2008. Současným trendům v přístupnosti aktuálně nejlépe odpovídá metodika Web Content Accessibility Guidelines 2.0 (WCAG 2.0) od skupiny WAI, která bude blíže představena v další části práce. [13]

4.4.1 WCAG 2.0

WCAG 2.0 je celosvětově nejznámější a nejuznávanější metodikou přístupnosti a pro postup při tvorbě webových stránek je také aktuálně doporučena sdružením SONS. Podle Radka Pavlíčka je však důležité si uvědomit, že soulad s metodikou ještě automaticky nezaručuje přístupnost webu. Tvůrci stránek musí správně porozumět smyslu jednotlivých pravidel a ne jen slepě tyto pravidla dodržovat. Technickou přístupností totiž automaticky nedocílíme přístupnosti reálné. [38]

Oproti metodice WCAG 1.0 respektuje nová verze situaci na poli prohlížečů a asistivních technologií tím, že jasně nestanovuje, co je, a co není přístupné. WCAG 2.0 Definuje 4 principy, které jsou základem přístupnosti webu[54]:

- Vnímatelnost – Informace a komponenty uživatelského rozhraní musí být reprezentovány způsobem, kterým je uživatel schopen je vnímat.
- Ovladatelnost – Uživatel musí být schopný používat součásti uživatelského rozhraní.
- Srozumitelnost – Informace a ovládání uživatelského rozhraní musí být uživateli srozumitelné.
- Robustnost – Obsah musí být dostatečně robustní, aby mohl být interpretován širokým spektrem přístupových zařízení, včetně asistivních technologií. Obsah musí zůstat přístupný i při použití nových technologií zobrazování.

Každý z principů je v metodice dále rozveden do návodů, jakými technikami obsah přizpůsobit pro handicapované uživatele k naplnění daného principu. V práci bude ze znalosti této metodiky vycházeno. Její kompletní lze nalézt zde [52].

Pro testování přístupnosti webu připravila organizace WebAIM kontrolní seznam zásad metodiky WCAG 2.0. Oproti oficiální dokumentaci se zaměřuje přímo na technologii HTML a technické splnění požadavků metodiky. K plnému znění metodiky je užitečným pomocníkem a lze jej nalézt zde [47].

4.4.2 Vyhláška č. 64/2008 Sb.

Spadá-li vytvářený web pod informační systém veřejné zprávy, je nutné, aby splňoval nároky Ministerstva vnitra České republiky. V tomto případě je potřeba se seznámit s vyhláškou č. 64/2008 Sb. a metodickými pokyny, které ministerstvo uveřejnilo na svém webu. [31]

4.5 Testování přístupnosti

Testování je nedílnou součástí vývoje webových stránek. V jeho průběhu lze odhalit nedostatky a zjistit celkovou přístupnost webu ještě před jeho publikací.

Protože největším bariérám v používání Internetu čelí tradičně těžce zrakově postižení uživatelé, zaměřuje se testování přístupnosti primárně na ně.

4.5.1 Testování nástrojem WAVE

Pro testování přístupnosti je k dispozici řada nástrojů a rozšíření do webových prohlížečů. Jeden z nejdoporučovanějších volně dostupných nástrojů je WAVE od organizace WebAIM [46].

WAVE je vhodný pro prvotní kontrolu přístupnosti webu. Vyhodnotí stránku v zadané URL adrese a upozorní na potenciální problémy. Jeho algoritmy odhalí například problémy vzniklé nesprávným strukturováním obsahu, redundancí odkazů, špatným kontrastním poměrem barev a mnohé další. Pozitivně hodnotí využití elementů spojených se zkvalitněním přístupnosti. Na odhalené nedostatky upozorňuje umístěním ikon přímo do webové stránky. [3]

4.5.2 Testování hlasovou čtečkou JAWS

Pro důkladnější testování je nutné se vžít do pozice nevidomého a používat web tak, jak jej používá on, tedy s hlasovou čtečkou. V praktické části bude k tomuto účelu využita čtečka JAWS a v testování bude postupováno podle českého překladu návodu od organizace WebAIM [26].

Při základním testování je vždy ověřeno alespoň správné strukturování stránek pomocí nadpisů, relevantní alternativní textové popisky u obrázků, přístupnost navigace, tabulek a formulářů. [4]

4.5.3 Validátory

Jedno z pravidel metodiky WCAG 2.0 se týká kompatibility s přístupovými zařízeními a konkrétním kritériem úspěšnosti, které se váže k validátorům, je syntaktická analýza. Cílem tohoto kritéria je zajistit, aby zařízení používané čtenářem webu mohlo korektně parsovat obsah. Kód zvoleného značkovacího jazyka by proto měl být syntakticky správně zapsán, aby nedocházelo k chybám na straně prohlížeče a ještě častěji u asistivních technologií. [38]

Nicméně, ne vždy validátory vyhodnotí jako správný postup ten, který je v praxi mnohem přínosnější pro uživatele. Typickým příkladem je tvorba zástupného obsahu do elementu, který asistivní technologie nemusí podporovat. Zde dochází ke křížení elementů, což validátory vyhodnotí jako syntaktickou chybu. V některých případech je proto nutné rozhodnout o prospěšnějším řešení a myslet za hranice validátoru. [3]

5 Optimalizace HTML5 a CSS3 pro prohlížeče

Připomeňme si, že asistivní technologie nejsou browsery uzpůsobené uživatelům se zdravotním postižením, ale speciální programy, které zpřístupňují prostředí běžných webových prohlížečů. Konečný výstup z asistivních technologií je tedy zásadně založený na tom, co jim webový prohlížeč dodá. [35]

Pavlíček [35] předkládá, že: „*Základem dobré přístupnosti je kvalitní strukturování webové stránky.*“ V této části práce budou rozebrány postupy, jak optimalizovat HTML5 a CSS3, aby prohlížeče poskytovaly asistivním technologiím informace v podobě, které rozumí.

5.1 Strukturování dokumentů

Strukturální značky dodávají funkcionalitám asistivních technologií smysl. Díky nim hlasové čtečky nejsou jen strohé předčítače textu řádek po řádku, ale umožňují uživateli pracovat s webem efektivně například přeskokováním mezi nadpisy.

Před příchodem HTML5 bylo k oddělení jednotlivých bloků stránky nutné používat element `div`. Protože `div` sám o sobě nenesl žádnou sémantickou informaci, dodávala se právě pomocí nadpisů. [5]

Také na problémy, které vznikají tímto postupem, upozorňuje Pavlíček [35]: „*Nadpisy jsou v současné době používány nejen ke strukturování hlavního obsahu stránky, ale i k uvození jejich dalších částí – menu, vyhledávání, zápatí, aktualit atp. To sebou nese dvě potencionální nevýhody – jistý zmatek v hierarchii nadpisů a uvození jednotlivých částí stránek pouze shora. V praxi to znamená, že uživatel sice ví, kde ta která část stránky začíná, ale už neví, kde končí.*“

Problém je o to větší, pokud je pod jedním nadpisem umístěno více vzájemně nesouvisejících obsahových částí. Uživatel poté v rámci celé stránky ztrácí orientaci. Jazyk HTML5 proto přináší spoustu nových sémantických značek, které umožňují členit dokument na více obsahových částí.

Podle účelu je specifikace HTML5 rozděluje do kategorií. Práce se bude zabývat jen kategoriemi, které přináší zásadní novinky do oblasti přístupnosti. Celkem se jedná o tyto kategorie [48]:

- Elementy rozdělující obsah
- Elementy toku obsahu
- Elementy nadpisů
- Elementy pro import zdrojů
- Elementy pro interaktivní obsah

Zjednodušené příklady použití některých z těchto elementů budou předvedeny na fiktivních webových stránkách Speciální základní školy v Poděbradech.

Pro účely porovnání, je zde ukázka zápisu záhlaví stránky s navigací v předešlých verzích HTML.

Ukázka kódu 2 Zápis záhlaví s navigačním panelem ve starších verzích HTML [zdroj: autor]

```
<div id="zahlaví">
  <h1>Speciální ZŠ Poděbrady</h1>
  <ul id="nav">
    <li><a href="#">Aktuality</a></li>
    <li><a href="#">O škole</a></li>
    <li><a href="#">Terapie</a></li>
    <li><a href="#">Třídy</a></li>
    <li><a href="#">Zaměstnanci</a></li>
    <li><a href="#">Kontakt</a></li>
  </ul>
</div>
```

5.1.1 Elementy rozdělující obsah

Do této kategorie řadí specifikace HTML5 čtyři elementy – `nav`, `section`, `article` a `aside`. Tyto elementy definují různé části dokumentu a mohou mít vlastní hierarchickou strukturu. Ke každému elementu lze například připojit vlastní záhlaví a zápatí. [1]

„Struktura dokumentu se nikde nezobrazuje, ale stejně jako všechny sémantické elementy je důležitá pro vyhledávací roboty a předčítače textu,“ podotýká Elizabeth Castro [2].

5.1.1.1 Element nav

Zkratka „nav“ pochází z anglického slova „navigation“ a element `nav` tak označuje část stránky obsahující navigaci. Navigace obecně odkazuje na jiné stránky nebo na části v rámci dané stránky. V případech, kdy element `nav` obaluje seznam odkazů, doporučuje W3C obalit odkazy navíc do značky pro list - `ul`. [48]

Protože element `nav` nemusí nutně značit jen hlavní navigaci webu, ale také vedlejší navigaci v postranním sloupci nebo navigaci v obsahu článku, může se jich v rámci jedné stránky vyskytovat více. Uživatel čtečky obrazovky je pak schopen přeskakovat mezi jednotlivými navigacemi přesně na obsah, který hodlá číst.

W3C dále podotýká, že ne každou skupinu odkazů je vhodné obalit pomocí `nav`. Element je primárně určen pro významné navigační bloky, což se netýká například typických odkazů jako „Mapa webu“ a „Prohlášení o webu“ v patičce stránky. K tomuto účelu je vyhrazen samotný element `footer` a použití `nav` by zde bylo nadbytečné. [48]

Nyní lze porovnat zápis kódu pomocí elementů `div` a identifikátorů ve starých verzích HTML (Ukázka 2) s rovnocenným zápisem v HTML5 (Ukázka 3).

Ukázka kódu 3 Zápis záhlaví s navigačním panelem v HTML5 [zdroj: autor]

```
<header>
  <h1>Speciální ZŠ Poděbrady</h1>
  <nav>
    <ul>
      <li><a href="#">Aktuality</a></li>
      <li><a href="#">O škole</a></li>
      <li><a href="#">Terapie</a></li>
      <li><a href="#">Třídy</a></li>
      <li><a href="#">Zaměstnanci</a></li>
      <li><a href="#">Kontakt</a></li>
    </ul>
  </nav>
</header>
```

5.1.1.2 Element section

„Sekce jsou logickými oblastmi stránky a element `section` zde slouží jako náhrada značek `div` zneužívaných k popisování logických částí stránky,“ vysvětluje Hogan [5]. Element `div` stále najde své využití pro potřeby stylování vzhledu stránky pomocí CSS. Pro dodání sémantického významu a seskupování obsahu využijeme `section`.

Téma každé sekce by mělo být předem oznámeno. Typicky se to řeší nadpisem (element `h1-h6`) v podobě přímého potomka sekce a opět to pomáhá uživatelům asistivních technologií v lepší orientaci po stránce. Mimo prostředí webu jsou vhodným přirovnáním k sekcím kapitoly v knize, záložky v šanonu nebo očíslované úseky této práce. Na webu je využijeme spíše k vyznačení souvisejících shluků obsahu. Části, pro které jsou vyhrazené samostatné elementy jako `header`, `footer`, `nav` nebo `aside` je vhodnější vyznačit jimi. [48] [3]

5.1.1.3 Element article

Záměrem specifikace elementu `article` bylo poskytnout celistvou obsahovou jednotku. Tiffany Brown [1] jej vystihuje takto: *„Reprezentuje kompletní nebo samostatný obsahový prvek v dokumentu, stránce, aplikaci nebo na webových stránkách – takový obsah je možné distribuovat nezávisle nebo používat opětovně (například na více stránkách).“*

Právě jeho distribuovatelnost ho odlišuje od elementu `section`. Předejít záměně těchto elementů se snaží Hogan [5] vlastní poučkou: *„O sekci přemýšlejte jako o části dokumentu. O článku přemýšlejte jako o vlastním obsahu, například časopisovém článku, příspěvku na blogu nebo položce aktualit.“*

Další záměna by mohla nastat s elementem `main`. Jejich úlohy v hierarchii dokumentu jsou však opět rozdílné. Pokud je hlavní obsah stránky (vyjímaje záhlaví, zápatí, navigace a postranní panely) pouze jednodílné kompozice, bude její kód obalen jen elementem `main`. Použití dalšího elementu `article` by zde bylo nadbytečné. V případě více článků na stránce, bude `main` obsahovat více elementů `article`. [48]

Pokud jsou v jednom elementu `article` zanořeny další elementy `article`, reprezentují vnitřní elementy obsah související s obsahem elementu vnějšího. Autor obsahu vnějšího elementu není zděděn elementy vnitřními. [48]

5.1.1.4 Element aside

„Element aside reprezentuje ekvivalent pro postranní sloupec v novinách nebo časopise. Označuje obsah, který se vztahuje k hlavnímu článku, ale nelze ho použít samostatně,“ definuje Tiffany Brown [1].

Může obsahovat citaci, diagram, poznámku, odkazy na příbuzná témata nebo jiné doplnění článku. Lze jej také použít pro umístění reklamy nebo speciální nabídky související s obsahem článku. Nikdy by však neměl zastupovat element `section` při tvorbě postranního panelu celé stránky. [5] [9]

5.1.2 Elementy toku obsahu

Jedná se o elementy používané v části body HTML dokumentu. V následujících podkapitolách z nich budou představeny ty nejvýznamnější pro tvorbu struktury stránky. Elementy související se značkováním obrázků, tabulek, formulářů a multimediálního obsahu budou představeny v samostatných kapitolách, které se dané tématice věnují.

5.1.2.1 Element header

Element `header` typicky využijeme pro tvorbu záhlaví. Může obsahovat různé typy obsahu od loga společnosti po vyhledávací pole, ale není nutně využíván pouze pro hlavní záhlaví stránky. Obdobně jej lze využít pro vyznačení záhlaví jednotlivých sekcí stránky nebo článků. [5]

V případě článku může element `header` obalovat jeho nadpis, datum publikace a jméno autora.

5.1.2.2 Element main

Element `main` se objevil až ve specifikaci HTML 5.1 a je tak jedním z nejnovějších elementů HTML5. Používá se k vyznačení hlavní části dokumentu nebo aplikace. Tou je míněna taková část, která je v rámci dokumentu unikátní – neobsahuje části, které se opakují při procházení webu, jako patičku a hlavní záhlaví s logem webu, navigaci a postranní panely. Typicky se jedná o části, které chceme skrýt také při tisku. [48]

Toho využívají některé webové prohlížeče pro chytré telefony nebo RSS čtečky k heuristické analýze ve funkci tzv. „čtenářského režimu“, kdy vyfiltrují vše kromě hlavního obsahu. [1]

„Webové prohlížeče mohou tento element používat pro vylepšení přístupnosti - například přeskočí navigaci pro nevidomé uživatele,“ vysvětluje Tiffany Brown [1]. Autoři specifikace nicméně prozatím doporučují obohatit element `main` o atribut ARIA role - `role="main"`, z důvodu nejisté podpory ze strany prohlížečů. Obecným předpokladem dále je, že na každé stránce se vyskytuje nanejvýš jeden element `main` a nikdy není potomkem elementu `article`, `aside`, `footer`, `header` nebo `nav`. [48]

5.1.2.3 Element footer

Element `footer` reprezentuje zápatí pro svého nejbližšího předchůdce typu sekce. Pokud je jeho nejbližším předchůdcem element `body`, je zápatí aplikováno na celou stránku. Typicky obsahuje informace o autorovi, odkazy související s dokumentem a údaje o ochraně autorských práv. Kontaktní údaje na autora by navíc měly být obaleny elementem `address` a při využití zápatí v článku, není syntakticky chybné zopakovat informace ze záhlaví. Vše záleží na konkrétní potřebě. Zápatí se nemusí nutně vyskytovat až na konci sekce, ve většině případů je to však vhodné. [48]

Příklad zápisu patičky webu ve starších verzích HTML lze vidět na následující ukázce.

Ukázka kódu 4 Zápis zápatí ve starších verzích HTML [zdroj: autor]

```
<div id="zapati">
  <ul id="nav">
    <li><a href="#">Mapa webu</a></li>
    <li><a href="#">Prohlášení o webu</a></li>
    <li><a href="#">Ochrana soukromí</a></li>
  </ul>
</div>
```

Ukázka 5 poté zobrazuje ekvivalentní zápis v HTML5 bez nutnosti vyznačení navigace, jelikož sémantický význam zde přebírá element `footer`.

Ukázka kódu 5 Zápis zápatí v HTML5 [zdroj: autor]

```
<footer>
  <ul>
    <li><a href="#">Mapa webu</a></li>
    <li><a href="#">Prohlášení o webu</a></li>
    <li><a href="#">Ochrana soukromí</a></li>
  </ul>
</footer>
```

Rámování sekcí záhlavím a zápatím má obrovský význam pro přístupnost. Uživatel asistivní technologie pozná, kde sekce začíná a kde končí.

5.1.2.4 Element address

Element `address` reprezentuje kontaktní informace pro nejbližšího předka `article` nebo `body`. Není tedy určen pro označení poštovní adresy, jak by se mohlo na první pohled zdát. Pokud se však nejedná o relevantní kontaktní informaci. [9]

Typicky bývá zahrnut v patičce webu společně s dalšími informacemi, které do ní náleží, a obsahuje email a jméno autora. Při použití u článků mohou poté webové prohlížeče indexovat sekce podle shody kontaktních informací. [48]

5.1.3 Nadpisy

Podle průzkumu organizace WebAIM z roku 2014 používá většina uživatelů screen readerů (65,6% respondentů) k hledání informací na stránce primárně nadpisy. Pro nevidomé jsou tak nadpisy stále velmi důležité orientační body na stránce. [45]

Na rozdíl od oblasti stránky určují nadpisy pouze začátek navigačního bloku. O to důležitější je volit výstižné názvy a správnou úroveň nadpisů. Na stránce by se měl vyskytovat pouze jeden nadpis úrovně 1, pro uvození hlavního obsahu. Pomocí nadpisů úrovně 2 je vhodné obsah rozdělit do majoritních částí. Úroveň 3 lze používat dle potřeby všude, kde další členění zlepší přehlednost obsahu. V dlouhých článcích s hlubokou hierarchií využijeme nadpisů nejnižší úrovně 4-6. [3]

5.1.4 Outline algoritmus

Zatímco ve značení úrovní nadpisů nenastaly žádné změny, přišlo HTML5 s alternativním algoritmem pro tvorbu osnovy (tzv. outline algoritmus). Tato možnost nastala s příchodem nových elementů sekcí. V každé sekci lze nyní vytvořit novou hierarchii nadpisů. [9]

Ukázka kódu 6 Zápis s použitím pouze nadpisů úrovně 1 [15]

```
<h1>Level 1</h1>
<nav>
  <h1>Level 2</h1>
</nav>
<section>
  <h1>Level 2</h1>
  <article>
    <h1>Level 3</h1>
    <aside>
      <h1>Level 4</h1>
    </aside>
  </article>
</section>
```

V závislosti na zanoření elementu sekce se mění úroveň nadpisu, který se v něm nachází. Nový zápis v Ukázce 6 bude tedy generovat identickou osnovu jako již zaběhlý zápis v Ukázce 7.

Ukázka kódu 7 Zápis s použitím nadpisů různé úrovně [15]

```
<h1>Level 1</h1>
<nav>
  <h2>Level 2</h2>
</nav>
<section>
  <h2>Level 2</h2>
  <article>
    <h3>Level 3</h3>
    <aside>
      <h4>Level 4</h4>
    </aside>
  </article>
</section>
```

Ačkoli je outline algoritmus znám od specifikace HTML5 z roku 2011, jeho implementace na poli webových prohlížečů a asistivních technologií není potvrzena. Autoři specifikace tudíž doporučují využívat původní struktury s nadpisy první až šesté úrovně. Bez implementace algoritmu by uživatel viděl všechny nadpisy jako nadpisy 1. úrovně. [48]

5.1.5 Zástupný obsah

Některé z elementů pro multimédia jsou navrženy tak, aby umožnily vestavět zástupný obsah pro zařízení, která daný element neznají či neumí jeho obsah v této formě reprodukovat. Typicky s nimi mají potíže starší prohlížeče a hlasové čtečky. V Ukázce 8 lze vidět příklad použití zástupného obsahu pro element `canvas`, který v HTML5 slouží k dynamickému vykreslování bitmap a grafických primitiv.

Ukázka kódu 8 Tvorba zástupného obsahu animace v elementu `canvas` [zdroj: autor]

```
<canvas id="triangle_fallback" width="200" height="200">
  <h1>Váš prohlížeč nepodporuje ukázky animací.</h1>
  <p>Alternativní obrázkovou ukázkou sestrojení rovnostranného
trojúhelníku naleznete níže.</p>
</canvas>
```

Zástupný obsah by se měl ideálně informačně co nejvíce přiblížit informacím z elementu, odkázat na přístupnější zdroj či alespoň podat uživateli hlášení, že daný prvek může ignorovat, pokud se jednalo například o element pro čistě dekorativní účel. V příkladu bude starší prohlížeč ignorovat `canvas`, kterému nerozumí, nicméně bude schopen naparsovat značky v něm obsažené. Zařízení, která rozumí `canvasu`, budou ignorovat zástupný obsah a `canvas` vykreslí. Nabízené řešení nemusí být validní, ale přispěje k lepší použitelnosti. [3]

5.2 WAI-ARIA

5.2.1 Orientační role specifikace ARIA

Většina webových stránek má podobnou strukturu. Jsou kódovány lineárně v pořadí: záhlaví, navigace, hlavní obsah a zápatí. Právě v tomto pořadí předčítají hlasové čtečky obsah stránek uživateli, a protože se části záhlaví, navigace, zápatí na všech stránkách opakují, musí uživatel jejich obsah vyslechnout znovu a znovu. [3]

Řešením je využití HTML5 rolí, převzatých ze specifikace WAI-ARIA, které umožní zasadit pomyslnou kotvu v blízkosti hlavního obsahu. Tyto orientační body umožní hlasové čtečky zanalyzovat a kategorizovat části stránky. Nevidomý uživatel pak

může přeskočit rovnou na hlavní obsah nebo se orientovat podle rejstříku částí stránky, který čtečka podle rolí vytvoří. [5]

Užitečné byly role zejména ve spojení s elementy `div`, kterým dodávaly sémantický význam. Jejich využití se však uplatní i v HTML5 minimálně pro zpětnou kompatibilitu se starými verzemi asistivních technologií a webových prohlížečů. *„Ačkoliv se orientační role a elementy jazyka HTML5 překrývají, předčítače textu jsou zatím daleko více nakloněné specifikaci ARIA. Můžeme tedy psát kód jazyka HTML běžným způsobem a přístupnost vylepšovat přidáváním rolí ARIA,“* vysvětluje Elizabeth Castro [2].

Definice rolí a účel jejich použití [2]:

- role `banner` – Oblast záhlaví platného pro celé webové stránky. Přiřazujeme většinou elementu `header`.
- role `search` – Identifikuje oblast z formulářem pro vyhledávání.
- role `navigation` – Označení skupin navigačních elementů. Přidělujeme všem elementům `nav` nebo jinému elementu, který odkazy obaluje.
- role `main` – Hlavní obsah dokumentu. Přidělujeme HTML5 elementu `main`, případně elementům `article` nebo `section`.
- role `complementary` – Samostatně smysluplná část doplňující hlavní obsah. Přidělujeme všem elementům `aside` a elementům `div` s doplňujícím obsahem.
- role `contentinfo` – Identifikuje oblast s informacemi o existenci obsahu, jako jsou autorská práva a odkazy na prohlášení o ochraně soukromí. Lze přirovnat k elementu `footer` pro celé webové stránky.

V případě kombinace ARIA atributů a HTML5 elementů převezmou atributy kontrolu nad nativní funkcí HTML5 elementů. [3]

5.2.2 Stavby a vlastnosti specifikace WAI-ARIA

WAI-ARIA dále nabízí prostředky pro popsání stavů a vlastností. Některé budou detailně rozebrány v souvisejících kapitolách, popis dalších užitečných je uveden zde [51]:

- `aria-checked` (stav) – Indikuje hlasové čtečce stav zaškrtnutí checkboxu nebo radio buttonu.
- `aria-disabled` (stav) – Indikuje, že element je viditelný, ale není možné měnit jeho obsah nebo ho jinak používat. Typicky použitelné u neaktivních tlačítek nebo předvyplněných inputů.
- `aria-expanded` (stav) – Indikuje, zda nabídka (například navigace) je rozbalená nebo sbalená.
- `aria-hidden` (stav) – Informuje uživatele o tom, že element je z rozhodnutí autora pro čtenáře skryt.
- `aria-invalid` (stav) – Informuje uživatele, že zadaná hodnota neodpovídá očekávanému formátu.
- `aria-live` – Indikuje, že v elementu došlo ke změnám, a předá asistivní technologii report o proběhlých změnách.
- `aria-multiline` – Indikuje, zda text box akceptuje víceřádkový vstup.
- `aria-readonly` – Indikuje, že element je určen pouze pro čtení, jinak je použitelný při procházení karetem.
- `aria-required` – Informuje uživatele, že před odesláním formuláře je nutné vyplnit daný element.

Jmenované stavy a vlastnosti jsou plně podporovány hlasovou čtečkou JAWS. [21]

5.2.3 Vlastnosti živých oblastí

Živé oblasti slouží ke zpřístupnění dynamických stránek nevidomým uživatelům. Vidomý uživatel má při procházení webu celistvý přehled nad děním na stránce. Pokud v určité oblasti dojde k dynamické změně, nedělá takovému uživateli potíže zaznamenat a vyhodnotit změny. Z principu hlasových čteček je nevidomý uživatel soustředěn v danou chvíli pouze na jednu oblast. Pokud chce hlasová čtečka nevidomého o změně informovat, znamená to přerušit jeho práci. To může být v danou chvíli rušivé a nežádoucí. Atribut `aria-live` proto nastavuje živým regionům jednu ze tří úrovní priority vyrušení [53]:

- off – Výchozí nastavená úroveň, která nijak neupozorňuje uživatele na nastalé změny. Příkladem jsou časté změny koordinací GPS v pozadí nebo nepodstatné tikající hodiny.
- polite – Uživatel bude na změny upozorněn pouze ve chvíli, kdy čtečka nepracuje. Příkladem je použití u prezentace změn v nadpisech.
- assertive – Změny v tomto regionu by měly být oznámeny uživateli ihned, jak je to možné. Příkladem jsou varovné zprávy při validaci formuláře.

Živé oblasti najdou využití například při přepisu průběhu hokejového utkání v reálném čase, u sledování změn na trhu, předpovědi počasí nebo zdi na Twitteru. [4]

5.2.4 ARIA label a describe

Nativní ovládací prvky HTML nesou dostatečný sémantický význam na to, aby asistivní technologie rozpoznala jejich účel. Chce-li vývojář použít jako ovládací prvek vlastní komponentu, představuje to pro nevidomého problém. Například při vytvoření tlačítka pomocí grafického obrázku nemusí být nevidomému zřejmé, co komponenta představuje a k čemu slouží. [3]

Pro dodání sémantického významu vlastní komponentě zavedla specifikace ARIA tři atributy [53]:

- `aria-labelledby` – Přístupný způsob, jak odkázat na element na stejné stránce, který poskytuje označení komponenty, u které je atribut použit. Například sémantická informace, že se jedná o vlastní vytvořené tlačítko.
- `aria-label` – Definuje řetězec znaků (string), který popisuje element, u něhož je atribut použit. Náhrada atributu `aria-labelledby`, pokud text není umístěn na prohlížené stránce.
- `aria-describedby` – Identifikuje element sloužící k dodatečnému popisu některého z objektů. Zde by byl vyplněn identifikátor elementu s detailním popisem účelu tlačítka a následků jeho použití. Více v kapitole věnované obrázkům.

5.2.5 Atribut aria-required

WAI-ARIA nabízí vlastní způsob, jak oznámit uživateli, která pole je nutná vyplnit před odesláním formuláře. Stačí u vybraného elementu `input` nastavit hodnotu atributu `aria-required` na `"true"`. Asistivní technologie, která tento atribut podporuje, oznámí uživateli po přenesení focusu na pole požadavek na vyplnění. Více o kombinaci ARIA atributu `aria-required` a HTML5 atributu `required` v kapitole věnované formulářům. [3]

5.3 Techniky responzivního webdesignu

5.3.1 CSS3 media queries

K dosažení cílů responzivního webdesignu pomohl vznik nové vlastnosti CSS3 – mediálních dotazů. *„Moc mediálních dotazů tkví v jejich způsobilosti testovat zařízení, zda mají jisté schopnosti. Testování probíhá pomocí výrazů, které se vyhodnocují na pravdu (true) nebo nepravdu (false),“* vysvětluje Kadlec [6]. Zařízení poté rozhodne, zda se na něj blok stylů vztahuje nebo ne.

Struktura mediálního dotazu se skládá ze čtyř komponent: mediálních typů, mediálních výrazů, logických klíčových slov a pravidel. S jejich pomocí můžeme zacílení stylů omezit na libovolné výstupní zařízení. V praxi ovšem mediální dotazy cílí na zlomové body rozlišení, které jsou odvozeny z míst, kde se přibližně začínají stránky rozpadat, a proto vyžadují úpravy. Vybrané hodnoty v Ukázce 9 tedy nejsou náhodné, ale zároveň nepředstavují šířku obrazovek konkrétních zařízení. [8]

Ukázka kódu 9 Využití CSS3 vlastnosti media queries pro rozdělení úhlopříček [zdroj: autor]

```
@media (max-width: 767px) {  
  [...]  
}  
@media (min-width: 768px) {  
  [...]  
}  
@media (min-width: 992px) {  
  [...]  
}
```

V příkladu jsou mediální dotazy pomyslně cíleny na velikosti obrazovek smartphonů (šířka do 767px) tabletů (šířka nad 768px) a desktopů (šířka nad 992px).

5.3.2 Meta element viewport

Do každého dokumentu HTML, pro který je implementován responzivní webdesign, je potřeba vložit nastavení meta elementu `viewport`. U desktopových prohlížečů představuje viewport viditelnou oblast prohlížeče, tedy po celé jeho šířce. Protože v dnešní době také smartphony disponují vysokým rozlišením, je zde problematika komplikovanější. „Navzdory tomu, že mají obrazovky mnohem menší, pokoušejí se zobrazit web v kompletní podobě,“ předkládá Kadlec [6].

Ukázka kódu 10 Nastavení viewportu pro účely responzivního designu [8]

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

Přidání řádku z Ukázky 10 do elementu `head` HTML stránky zajistí korektní chování. Direktiva `width=device-width` nastaví viewport layoutu obrazovky na šířku zařízení v pixelech a direktiva `initial-scale=1.0` nastaví počáteční úroveň přiblížení stránky na 1x, tedy ani oddálené ani přiblížené. „Bez tohoto elementu by se stránka zobrazila se svou obvyklou šířkou, ale zmenšená tak, aby se vlezla na obrazovku,“ potvrzuje Lazaris [8].

5.3.3 Mřížkové layouty

Mřížky pomáhají docílit vyváženosti rozestupů mezi prvky stránky a jejich přehledného uspořádání na stránce, čímž výrazně přispívají k lepší použitelnosti webu. „Dobře implementovaný mřížkový systém (grid systém) způsobuje, že web vypadá méně chaoticky, je lépe čitelný a lépe se také prohledává,“ dodává Kadlec [6].

Správně použitý systém mřížky umožňuje uživatelům lépe odhadnout, kde najít hledanou informaci, jelikož dodává stránkám informační řád a soulad. Mřížky také zajišťují konzistenci návrhu webu při přidávání nového obsahu. [11].

Umístění obsahu v demonstračním webovém projektu se bude opírat o systém mřížky CSS frameworku Bootstrap verze 3.3.5, který při zvětšování viewportu škáluje standardně obsah až do 12 sloupců.

5.4 Zpřístupnění obrázků

V této kapitole budou popsány techniky zpřístupnění grafického obsahu a pokryty budou jak techniky časem prověřené a dosud nepřekonané, tak příspěvky nové sémantiky HTML5.

Díky textovým alternativám obrázků mohou jejich sdělení porozumět všichni uživatelé, kteří z nějakého důvodu nemohou vnímat jeho grafickou podobu. Kromě zrakových postižení může být příčinou nenačtení obrázku z důvodu technických problémů nebo omezeného datového limitu na připojení k Internetu. *„Správné zpřístupnění obrázků je proto jedna z prvních věcí, kterou by se měl každý webdeveloper naučit a potom ji také v praxi uplatňovat,“* dodává Pavlíček [35].

5.4.1 Element figure a figcaption

Novinkou ve značení HTML5 jsou elementy `figure` a `figcaption`. Ve vzájemné spolupráci slouží k anotaci diagramů, fotografií, ilustrací a zdrojových kódů. Jejich revolučnost spočívá ve snaze vytvořit samostatnou jednotku, na kterou je možné odkázat z jakékoli části webu. [3]

Jinými slovy, na obrázek v článku je pouze referencováno, strukturálně je od textu oddělen. Přináší to jistou flexibilitu v umístění obrázku a jeho popisu, který je s ním pevně svázán. Syntaxe spočívá v umístění obsahu do elementu `figure` a vyplnění popisu do jeho potomka `figcaption`, jak je vidět v následující ukázce. [9]

Ukázka kódu 11 Použití elementů `figure` a `figcaption` [zdroj: autor]

```
<figure>
  
  <figcaption>Výstižný popis k tomuto obrázku</figcaption>
</figure>
```

Díky těmto elementům již není nutné využívat ke svázání obrázku s popisem elementu `div`, jak tomu bylo dříve. Asistivní technologie využijí toho, že element

`figcaption` dodává popisu sémantický význam. Příklad formátované samostatné jednotky elementu `figure` lze vidět na Obrázku 4.



Žák v průběhu canisterapie

Obrázek 4 Popisek v elementu `figcaption` svázán s obrázkem elementem `figure` [zdroj: autor]

Platí, že v elementu `figcaption` lze použít veškeré HTML značky, včetně odkazů a ikon. V případech, kdy `figcaption` obsahuje detailní popis obrázku, není nutné nadbytečně vyplňovat atribut `alt`. Naopak je nutné se vyvarovat použití prázdného atributu `alt` současně s nevyplněným elementem `figcaption`. Hlasová čtečka by vyhodnotila obrázek jako čistě dekorativní a nevidomý uživatel by k němu neměl přístup. [9]

5.4.2 Popis vizuálně bohatého obrázku

Obrázky mohou nést vysokou informační hodnotu. Někdy však může být velmi obtížné jejich obsah vystihnout pouze textem. V praxi je často nemožné splnit nároky WCAG 2.0, aby popis obrázku byl jeho přímým ekvivalentem nebo plnou náhradou. Navíc se do popisu např. uměleckého díla může promítat subjektivní úhel pohledu tvůrce popisu.

Místo zachycení každého detailu obrázku je nutné vystihnout jeho myšlenku použitím elementů `alt` nebo `figcaption`. V situacích, kdy by psaní samostatného popisu pomocí `alt` bylo příliš rozsáhlé, může být užitečnější odkázat na příslušnou část dokumentu použitím ARIA atributu `aria-describedby`. Jednoduchou alternativou může být slovní odkázání typu „Pro více informací k tomuto obrázku, přejděte zde“ s připojeným linkem na jinou webovou stránku. V krajním případě, pokud nelze vhodnou textovou náhradou, se doporučuje vyplnit element `alt`

hodnotou `null`. Connor toto východisko považuje za lepší, než ponechání rozhodnutí o naložení s obrázkem na obrazové čtečce. [3]

Atribut `longdesc` pro popis komplexních obrázků je považován za zastaralý a byl vyřazen ze specifikace HTML5.

5.4.3 Tabulky a grafy

Pokud obrázkem je graf nebo tabulka, měl by atribut `alt` obsahovat stručné shrnutí toho, co se čtenáři snaží sdělit. V případě výsledků průzkumu volebních preferencí bychom zde například uvedli, která strana obdržela nejvíce hlasů, která nejméně apod. Jestliže stránka obsahuje detailnější popis, je vhodné na něj odkázat pomocí atributu `aria-describedby`. [3]

5.4.4 Text ve formě obrázku

Typické použití obrázkového textu je v logu stránky. Atribut `alt` by zde měl doslovně zrcadlit text v obrázku, nicméně tak, aby bylo zřejmé, na jakých stránkách se nacházíme. Jak můžeme vidět na Obrázku 5, z použití kódu `alt="FitSpirit"` těžíme výhody také při nedostupných obrázcích, kdy strohý text může suplovat originální obrázkové logo. [49]



Obrázek 5 Porovnání originálu s náhradou loga atributem `alt` [zdroj: autor]

Tento způsob použijeme, kdykoliv by absence obrázku mohla znemožnit přístup k jeho informační hodnotě.

5.4.5 Obrázky s funkcionalitou

Dalším příkladem, kdy nepopisujeme vzhled obrázku, jsou obrázkové odkazy nebo obecně obrázky s funkcionalitou. „Pokud má obrázek nějakou funkci (slouží například jako tlačítko nebo odkaz), pak by textová alternativa měla popisovat tuto funkci spíše než to, co je na obrázku,“ dokládá Pavlíček [35].

Příkladem může být obrázek letenky s nápisem „Zakoupit“ v rezervačním systému aerolinek nebo pouhá ikona tiskárny s funkcí vytištění dané stránky. Zde bude atribut `alt` obsahovat popis funkce odkazu `alt="Zakoupit letenku"` resp. `alt="Vytiskni stránku"`. [49]

5.4.6 Dekorativní obrázky

Obrázky s čistě estetickým účelem, které nepřidávají žádnou informační hodnotu ani neposkytují žádnou další funkci, musí být označovány prázdným atributem `alt` (`alt=""`) nebo lépe, nahrazeny technikami CSS. Tím zaručíme, že budou asistivními technologiemi ignorovány. [49]

5.4.7 Ikony

Používáme-li obrázkovou ikonu jako součást textového odkazu pro podtrhnutí jeho funkce, měla by být s textem pevně svázána. Na některých webech lze pozorovat, že jako samostatný odkaz funguje jak ikona, tak text. To vede v asistivních technologiích k nežádoucímu zdvojování odkazů, kdy focus prochází oba odkazy. [3]



Obrázek 6 Ilustrační ukázka ikon v menu rezervačního systému fitness centra [zdroj: autor]¹

Pokud ikona nese pouze dodatečnou vizuálně-informační hodnotu, vyplňujeme dle W3C doporučení atribut `alt` hodnotou `null`, jak je vidět v Ukázce 12. Pokud by například obrázek varovného vykřičníku doplňoval text ve smyslu upozornění, je nutné jeho atribut `alt` vyplnit hodnotou `alt="Varování!"`. [49]

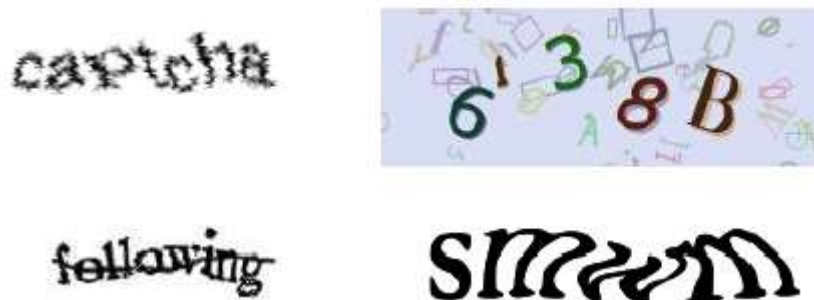
¹ Obrázek navigace je pouze ilustrační ukázkou využití doplňkových ikon. Konkrétně využívá připojení speciálního písma na stránku (tzv. font icons), které místo běžných znaků obsahuje grafické symboly. Výhodou font ikon mimo jiné je, že lze jejich podobu dále pomocí CSS upravovat.

Ukázka kódu 12 Zjednodušený úryvek kódu ikon v menu [zdroj: autor]

```
<ul>
  <li><a href="home.html">Úvod</a></li>
  <li><a href="lessons.html">Lekce</a></li>
</ul>
```

5.4.8 CAPTCHA

Speciálním případem obrázkového textu je tzv. CAPTCHA ochrana, která žádá uživatele o opsání textu z obrázku. Nejčastěji se s ní setkáme při odesílání anonymních formulářů či komentářů na webu. Jedná se o Turingův test, který má rozlišit, zda přispěvatelem je opravdový člověk nebo spamový robot. [4]



Obrázek 7 Příklady CAPTCHA obrázků [4]

Princip CAPTCHA ochrany spočívá v tom, že text na obrázku je nějakým způsobem deformován. Navíc, protože se metody robotů pro automatické rozpoznávání CAPTCHA stále zdokonalují, musí nástroje pro tvorbu CAPTCHA generovat stále více zdeformované texty. Poté již rozpoznání znaků nečiní potíže jen nevidomým nebo dyslektikům, ale také lidem bez zjevných zrakových vad. [50]

5.4.8.1 Náhrady CAPTCHA ochrany

Jednou z alternativ k vizuální CAPTCHA ochraně je audio CAPTCHA a W3C ji doporučuje umístit ihned vedle obrázku. Nevidomý si přehraje zvukovou nahrávku a do formuláře vyplní znaky, které v ní slyšel. [49]

Další možností je položení otázky, jejíž zodpovězení může být snadné pro člověka, ale neřešitelné pro spamového robota. Dotazy by měly být nezávislé na kultuře a zemi, z které uživatel pochází. Příkladem takové otázky je: „Jak se nazývá sport, ve kterém se používá hokejka?“ nebo „Který den následuje po čtvrtku?“. Časté je

použití jednoduché matematiky jako „Kolik je 2 + 3?“. Podobné dotazy dokáže zodpovědět pouze sofistikovaný robot. [4]

5.4.8.2 Služba CAPTCHA Help

Protože bezpečnostní prvky webů často tvoří bariéry handicapovaným uživatelům, vytváří zájmové skupiny různé projekty na jejich podporu. Jedním z takových projektů je služba CAPTCHA Help provozovaná zdarma sdružením CZ.NIC. Prostřednictvím e-mailu či rozšíření do prohlížeče Chrome odešle nevidomý snímek CAPTCHA obrázku operátorovi a ten mu následně v odpovědi vrátí požadovaný text pro zadání. [18]

5.5 Přístupnost médií a internacionalizace v HTML5

Do oblasti audia a videa přináší HTML5 řadu novinek. Samo o sobě se snaží dodat prostředky pro publikaci multimediálního obsahu v prohlížečích bez nutnosti instalovat software třetích stran. Především zavádí nové elementy `video` a `audio`, které by měly usnadnit vývojářům práci. Přesto se stále nevyhneme nejednotné podpoře kodeků napříč prohlížeči a dalším útrapám, kterým se tato kapitola bude věnovat.

HTML5 má ambice zlepšit přístupnost multimédií, počínaje podporou tvorby vlastních přístupných přehrávačů a podporou tzv. technologií alternativního obsahu. Technologie alternativního obsahu dodávají uživatelům obsah v takovém formátu, v jakém jej dokáží konzumovat. Alternativní reprezentace původního obsahu může být například ve formě titulků (subtitles), legendy (captions) nebo audio popisků k videu. S jejich pomocí bychom měli být schopni publikovat obsah dostupný uživatelům se sluchovým postižením nebo uživatelům, kteří nemluví jazykem použitým v dané zvukové stopě. Účel a podoba jednotlivých možností budou přiblíženy níže. Dále budou představeny funkce, které HTML5 zavádí ve snaze usnadnění přístupu uživatelů k médiím a jejich internacionalizace. [3]

5.5.1 Popisné video

Cílem popisného videa je poskytnout obrazové informace videa ve formě vhodné pro konzumaci zrakově postiženým uživatelem. Technika je založena na výstižném

popisu dění ve videu, který je uživateli poskytován formou textu nebo zvukové stopy. Alternativní zvuková stopa je namluvena mluvčím, který postupně vysvětluje, co se na videu zrovna odehrává. Textový popis je synchronizován s videem tak, aby čtenářova hlasová čtečka byla schopna plynule předčítat dění na obrazovce. [10]

5.5.2 Titulky

Poskytují textový přepis toho, co se říká v audio stopě, primárně za účelem překladu do jazyka uživatele. Jsou hlavním nástrojem k internacionalizaci multimediálního obsahu. Náročnější formou je předabování původní zvukové stopy. [9]

5.5.3 Legendy

Legendy jsou hlavní metodou poskytování alternativního obsahu uživatelům se sluchovým postižením. Oproti titulcům poskytují také popis hudby v pozadí a významných zvukových efektů. V záznamu divadelní hry by se jednalo například o zvuky, na které herci reagují a hudbu, která dokresluje atmosféru na jevišti. Z legend mohou těžit i uživatelé bez sluchového postižení nachází-li se v hlučném prostředí nebo nemají vhodné výstupní zařízení. *„U zdrojů audia jsou legendy vhodné zejména v prostředí sdílených s uživateli bez sluchového postižení – v opačném případě jsou vhodnější transkripce, protože umožňují nezávislou rychlost čtení toho, co bylo řečeno v audio souboru,“* dodává Silvia Pfeiffer [10].

5.5.4 Transkripce

Prostou transkripcí chápeme kompletní textový přepis zvukové stopy videa či audio nahrávky. Uživatelé se sluchovým postižením mohou čerpat informace z audio nahrávky vlastním tempem čtení ve formě podobné článku. Převod řeči na text opět poskytuje vhodný formát pro vyhledávání. Technikou blízké k legendám je tzv. interaktivní transkripce. Ta sice poskytuje ucelený přepis zvukové stopy, nicméně rozdělený do bloků synchronizovaných s videem. V průběhu videa se postupně zvýrazňují nebo posouvají právě předčítané bloky. Transkripci používá například Česká televize ve svém webovém archivu. [10]



Interaktivní transkripce

Klíknutím na text začnete přehrávat video z daného místa.

Lenka Řihová: iSen je vlastně taková neformální komunita rodičů, učitelů a všech dalších, které zajímá využití iPadu ve speciálním školství.

Lenka Řihová: "I" je první písmenko všech Apple produktů a "Sen" znamená to, že si splníme sen - o využívání iPadu - o tom, jak iPady mohou pomáhat.

Lenka Řihová: Ale taky "Special Educational Needs" - Speciální vzdělávací potřeby.

Obrázek 8 Ukázka interaktivní transkripce [zdroj: autor]

5.5.5 Znaková řeč a znakové jazyky

Zejména neslyšící uživatelé denně používají znakovou řeč k běžnému dorozumění. Je proto zřejmé, že překlad v podobě videozáznamu znakové řeči je jim nejbližší. Tento záznam je obvykle pevnou součástí původního videa a vkládá se do spodního pravého rohu tak, jak to známe z vybraných televizních pořadů. Záznam může být i v samostatné stopě, což často přináší výhodu v možnosti skrytí překladu. Na Internetu tento způsob využívá několik kanálů serveru Youtube, převážně těch, které se zabývají tematikou pro handicapované.

5.5.6 Kombinovaná transkripce

Pro hluchoslepé uživatele je jediným řešením, jak konzumovat obsah Braillovo písmo. Je nutné jim poskytnout přepis dění v audio nahrávce i na obrazovce. Technicky se tato potřeba realizuje kombinací transkripce audia a textového popisu videa. [10]

Na závěr je dobré si uvědomit, že z vytváření alternativního obsahu profitují všichni uživatelé Internetu. „*Například největší předností je to, že máme k dispozici text, který přesně reprezentuje, co se děje ve videu, přičemž tento text je nejlepším prostředkem, když potřebujeme něco vyhledat,*“ uvádí Silvia Pfeiffer [10]. V současné době je totiž text stále nepřekonatelným zdrojem pro indexování audiovizuálního obsahu.

Implementace vybraných forem alternativního obsahu pomocí HTML5 budou předvedeny v praktické části.

5.5.7 Nativní přehrávač

Dnešní webové prohlížeče mají vestavěn nativní HTML5 přehrávač. Jeho provedení se v každém z prohlížečů nepatrně liší a lze ho snadno vyvolat umístěním následujícího kódu do elementu `body`.

Ukázka kódu 13 Kód pro vyvolání nativního přehrávače [zdroj: autor]

```
<video controls></video>
```

Podobu nativního přehrávače v prohlížeči Google Chrome lze vidět na Obrázku 8.

5.5.7.1 Atributy elementů médií

K účelnému využití přehrávače je potřeba element `video` resp. `audio` obohatit některými z podporovaných atributů. Funkce těch nejběžněji používaných jsou následující [56]:

- `src` – Předání URL adresy umístění souboru, který chceme přehrávat. Ve zdroji specifikujeme také MIME typ, kterým rozlišujeme typ souboru.
- `controls` – Povel prohlížeči k zobrazení nativních ovládacích prvků přehrávače.
- `preload` – Volbou `none`, `metadata` nebo `auto` rozhodujeme o tom, zda má prohlížeč při načtení stránky automaticky zahájit přednačítání média.
- `autoplay` – Ihned, po načtení dostatečného množství dat pro plynulý chod, zahájí přehrávání.

- `poster` – Adresa obrázku, který se zobrazí v přehrávači před spuštěním přehrávání.

Atribut `loop` dále slouží pro aktivaci opakovaného přehrávání a `muted` pro ztlumení zvuku.

5.5.7.2 Podpora formátů v prohlížečích

Podpora formátů videa se může napříč prohlížeči lišit. Před vlastním testováním je proto vhodné si podporu formátu ověřit na některém ze specializovaných webů. Pro webové prohlížeče je klíčové přiřadit k formátu videa správný MIME typ.

Přehled nejrozšířenějších formátů s odpovídajícími MIME typy [3]:

- MP3: audio/mpeg
- OGG: audio/ogg; video/ogg
- MP4: audio/mp4; video/mp4
- WebM: audio/mp4; video/mp4

Na základě informací z webu caiuse.com je zde sestaven přehled současné podpory nejrozšířenějších formátů videa v nejpoužívanějších prohlížečích. Mimo jiné tento web shromažďuje také informace o podpoře HTML5 elementů.

	WebM	Ogg/Theora	MPEG-4/H.264
IE 11	ne	ne	ano
Firefox 39	ano	ano	ano
Chrome 43	ano	ano	ano
Safari 8	ne	ne	ano
Opera 30	ano	ano	ano

Tab. 1 Současná podpora video formátů [20]

Z tabulky je vidět, že v současnosti má plnou podporu prohlížečů formát MPEG-4. Pro podporu starších prohlížečů je však vhodné poskytnout navíc alespoň formát WebM.

5.5.8 Element track

Pro připojení audio popisků, titulků či legend k multimediálním souborům zavedla specifikace HTML5 nový element `track`. Vkládá se do elementu `audio` resp. `video` a odkazuje na externí textový zdroj, který má být synchronizován s časovou osou média. Pojí se s ním kromě `src` nové atributy [48]:

- `kind` – Indikuje prohlížeči, jak má prezentovat data připojeného souboru. Nabývá hodnot `subtitles` pro titulky, `captions` pro legendy, `descriptions` pro video popisky a `chapters` – názvy kapitol, které usnadňují navigaci po médiu.
- `srclang` – Jazyk použitý v připojeném souboru. Jeho kód se specifikuje podle standardu IETF BCP 47. Tyto kódy se obvykle skládají ze dvou písmen; například `"cs"` pro češtinu.
- `label` – Krátký název textové stopy, který se zobrazuje v nabídce stop.
- `default` – Aktivuje danou stopu, pokud uživatelovy preference neurčí jinak.

Ukázka kódu 14 Připojení alternativních textových souborů [zdroj: autor]

```
<video width="480" height="360" controls>
  <source src="o_auticku.mp4" type="video/mp4">
  <source src="o_auticku.webm" type="video/webm">
  <track kind="subtitles" label="Titulky - české"
    srclang="cs" src="titulky_cz.vtt" default>
  <track kind="subtitles" label="Subtitles - English USA"
    srclang="en-US" src="titulky_enUS.vtt">
</video>
```

Jsou-li textové soubory dostupné, zobrazí nativní přehrávač tlačítko „CC“ (Closed Captioning) pro vypínání/zapínání popisků. Tímto tlačítkem lze zároveň přepínat mezi dostupnými textovými stopami. Stopy s atributem `kind="description"` jsou určeny pro nevidomé a měly by být interpretovány hlasovou čtečkou. [1] [9]

5.5.9 WebVTT

Ve spojení s elementem `track` zavádí HTML5 také nový formát souborů pro tvorbu alternativního textového obsahu – WebVTT (Web Video Text Tracks). Jedná se o klasický textový soubor, jehož syntaxe je inspirována slavným formátem

SubRIP (SRT). Formát WebVTT lze využít pro tvorbu stop titulků, legend, popisků i kapitol. W3C stále pracuje na standardizaci syntaxe mezi webovými prohlížeči, nicméně jeho podpora je již velmi dobrá. [2][9]

5.5.9.1 Základy syntaxe WebVTT

První řádek souboru WebVTT vždy začíná řetězcem `WEBVTT`. Následují skupiny popisků, zvaných *cue*, které se skládají z časového intervalu zobrazení ve videu a samotného textu. Časové údaje jsou uvedeny ve formátu `hh:mm:ss.mls`, kde `hh` označuje hodiny, `mm` minuty, `ss` sekundy a `mls` milisekundy. Počáteční a konečný čas zobrazení každého *cue* je oddělen znaky `-->` a zároveň jsou od sebe jednotlivé *cue* odděleny prázdným řádkem. [1]

Ukázka kódu 15 Začátek souboru WebVTT [zdroj: autor]

```
WEBVTT

00:00:08.100 --> 00:00:13.380
Bylo jednou jedno autíčko
a to se strašně moc nudilo.

00:00:16.146 --> 00:00:22.035
A jednou si autíčko vymyslelo,
že pojedě na výlet <b>do lesa</b>.
```

5.5.9.2 Sémantika popisků

Protože je vznik WebVTT úzce spojen s vývojem HTML5, využívá některých jeho značek k formátování titulků a legend. Těmi základními jsou elementy pro úpravu písma: element `i` pro kurzívu, element `b` pro tučné písmo a element `u` pro podtržení. Pro odlišení dialogů je klíčový element `v`, který označuje hlas postavy. Ve značce `<v Jmeno>` specifikujeme jméno postavy a její projev je možné graficky odlišit. Tyto techniky napomáhají nejen neslyšícím, pro pochopení děje ve videu, ale především lidem s kognitivními poruchami. [55]

Ukázka kódu 16 Formátování textů v souboru WebVTT [zdroj: autor]

```
00:00:42.000 --> 00:00:49.000
Červené autíčko bylo smutné
a křičelo: <v.hlasite Auticko><i>Pomoc! Pomoc! Pomoc!</i></v>
```

Podobně jako v HTML mají elementy svou počáteční a koncovou značku a lze je do sebe vnořovat. Vhodný editor titulků také automaticky doplňuje značky `
` pro odřádkování textu. Použití zmiňovaných značek je vidět na úryvku kódu v Ukázce 16, které odpovídá snímek na Obrázku 9.



Obrázek 9 Formátované WebVTT titulky [zdroj: autor]

Jak upozorňuje Tiffany Brown [1]: „Špatně napsaný kód ve formátu WebVTT způsobí nefunkčnost titulků a legend.“ Je proto vhodné využít některého z nástrojů pro validaci kódu.

5.5.9.3 Vzhled popisků

Odlišení dialogů postav pouhou úpravou stylu písma nemusí být vždy dostatečné. Tvůrci formátu WebVTT proto zavedli jeho sémantiku tak, aby bylo možné přizpůsobit vzhled titulků a legend kaskádovými styly – prostřednictvím pseudoelementu `::cue`. Pseudoelement `::cue` přijímá jako argument jeden nebo více sloučených selektorů jazyka CSS. [55]

Snímku z Obrázku 7 odpovídá úryvek značkování v Ukázce 17 a stylování v Ukázce 16. Zde byla nejprve využita jako argument třída `hlasite`, která má za účel zdůraznit výkřik zvětšením textu. Následně bylo pro odlišení barvy dialogu postavy autíčka použito spojení selektoru elementu `v` a selektoru atributu `voice`.

Ukázka kódu 17 Úryvek CSS pro formátování WebVTT titulků [zdroj: autor]

```
<style>
  video::cue(.hlasite) {
    font-size: 1.2em;
  }
  video::cue(v[voice="Auticko"]) {
    color: #fe5656;
  }
  [...]
</style>
```

Standardně jsou popisky umístovány do spodní části videa a zarovnány na střed. „Ačkoliv formát WebVTT poskytuje rovněž syntaxi pro zarovnávání a rozmístování popisků, výrobci prohlížečů ji zatím neimplementovali,“ vytýká Tiffany Brown [1].

5.6 HTML5 a přístupné tabulky

Tabulky jsou běžným prostředkem pro prezentaci dat a vztahů mezi nimi. Může se jednat o statistické průzkumy, události v kalendáři, sportovní výsledky nebo televizní program. Ve většině případů se záhlaví tabulky vyskytuje nahoře a data pod ním, nebo je umístěno v levé části a data v pravé. [2]

Forma tabulky umožňuje vidícímu uživateli snadno asociovat záhlaví sloupců či řádků s buňkami tabulky a rychle tak pochopit význam dat. Takový uživatel se ve vztazích tabulky brzy zorientuje a je pro něj tato reprezentace vhodnější, než kdyby data byla prezentována formou strohého textu.

Pro nevidomého uživatele představuje tento proces značný problém. Jeho hlasová čtečka prochází data lineárním způsobem, buňku po buňce, a je pro něj tudíž složité pochopit, jaký je účel tabulky a co data v ní znamenají. [3]

Standardně tvoříme tabulku elementem `table` a její řádky elementy `tr`. Řádky tabulky vyplňujeme elementy `th` pro buňky záhlaví nebo elementy `td` pro datové buňky. [Castro, 2012, str. 362] Abychom v HTML5 učinili tabulku přístupnou, musíme explicitně nadefinovat vazbu mezi datovou buňkou a odpovídajícím záhlavím. Díky tomu, budou moci asistivní technologie prezentovat tabulku nevidomému způsobem, kterému rozumí. Například hlasová čtečka nevidomému oznámí, ke kterému záhlaví procházená datová buňka náleží. [3]

5.6.1 Obecná pravidla tvorby tabulek

Před započítím kódování tabulky je doporučeno krátké zamyšlení, zda data v tabulce hodláme reprezentovat tím nejjednodušším způsobem. Složitá tabulka vede ke složitému kódu. Dále je vhodné rozhodnout, zda k prezentaci dat tabulku nutně potřebujeme. Prostý text předkládá nevidomému vždy méně bariér. [3]

Obecný postup tvorby začíná vytvořením vizuálně přehledné tabulky tak, aby si vidící uživatelé byli schopni snadno spojit data se záhlavím a pochopit význam tabulky. Pro nevidomé pokračuje poskytnutím shrnutí významu tabulky a asociací datových buněk s korespondující hlavičkou sloupce resp. řádku. U složitějších tabulek je vhodné přidat výsledky, které z dat vyplývají.

5.6.2 Element caption

Text mezi tagy elementu `caption` označuje titulek rodičovské tabulky. Měl by krátce a výstižně shrnovat její význam. [48]

Typicky bývá zvýrazněn a umístěn na střed nad tabulku, jak lze vidět na Obrázku 10. Pokud nevidomý přenesse focus na tabulku, je mu hlasovou čtečkou automaticky titulek oznámen. Všichni uživatelé tak ihned pochopí, k čemu tabulka slouží.

Informace o aktuálně vypsaných lekcích					
Datum	Čas	Aktivita	Místnost	Kapacita	Instruktor
čtvrtek 20.8.	10:00 - 11:30	Funkční trénink	Venkovní prostory	6/6	J. Bočák
pondělí 24.8.	10:00 - 11:00	Pilates	Malý cvičební sál	4/8	M. Michaelová
úterý 25.8.	16:00 - 17:00	Jumping	Hlavní cvičební sál	6/10	Z. Poláková
středa 26.8.	15:00 - 16:00	CrossFit	Hlavní cvičební sál	3/5	F. Skořepa

Den konání lekce se nachází v prvním sloupci, čas zahájení a ukončení ve sloupci druhém. Třetí sloupec značí, o jaký druh aktivity se jedná a čtvrtý, kde se lekce koná. Pátý sloupec informuje o dostupném počtu míst z původního počtu. V posledním sloupci naleznete jméno vedoucího lekce.

Obrázek 10 Tabulka lekcí v rezervačním systému pro fitness centrum [zdroj: autor]

Hlasové čtečky jako JAWS navíc oplývají funkcí postupného procházení vybraných elementů. Například stiskem klávesy „T“ může nevidomý procházet všechny tabulky na stránce. Pokud s tabulkou není svázán žádný titulek `caption`, oznámí čtečka pouze „tabulka“. Je-li `caption` vyplněn, čtečka titulek přečte. [3]

Ukázka kódu 18 Zjednodušený úryvek zápisu caption [zdroj: autor]

```
<table>
  <caption>Informace o aktuálně vypsanych lekcích</caption>
  [...]
</table>
```

5.6.3 Atribut aria-describedby

Pro delší popis významu tabulky a shrnutí výsledků jejích dat byl dříve využíván element `summary`. Ten je v současnosti považován za zastaralý, ale lze jej důstojně nahradit použitím ARIA atributu `aria-describedby`. [3]

Popis se musí nacházet na stejné stránce jako tabulka a do `aria-describedby` vyplňujeme identifikátor prvku, kterým je popis reprezentován. Tento způsob umožní hlasové čtečky na popis odkázat. Ukázku popisu lze vidět na předešlém Obrázku 10 a odpovídající úryvek zdrojového kódu na Ukázce 19.

Ukázka kódu 19 Zjednodušený úryvek použití atributu aria-describedby [zdroj: autor]

```
<table aria-describedby="summary">
  [...]
</table>
<p id="summary">
  Den konání lekce se nachází v prvním sloupci, čas zahájení a
  ukončení ve sloupci druhém. Třetí sloupec značí, o jaký druh
  aktivity se jedná a čtvrtý, kde se lekce koná. Pátý sloupec
  informuje o dostupném počtu míst z původního počtu. V posledním
  sloupci naleznete jméno vedoucího lekce.
</p>
```

Atribut je vhodné použít u komplexních či nestandardních tabulek pro popis jejich účelu, nastínění základní struktury buněk, upozornění na skryté vzory a obecně by měl poučit uživatele o správném používání tabulky. [48]

5.6.4 Kombinace headers/ID

Jedním ze způsobů, jak programově asociovat datovou buňku se záhlavím, je kombinace atributu `headers` a identifikátoru elementu `th`. Touto technikou dosáhneme toho že, hlasová čtečka oznámí nevidomému, do jakého sloupce buňka patří ještě před tím, než přečte její obsah. [3]

Konvenci pojmenování identifikátorů si můžeme libovolně zvolit. Důležité je, aby označení byly v rámci stránky unikátní. Vhodné je zvolit ekvivalentní názvy záhlaví, jak lze vidět na Ukázce 20, která odpovídá tabulce z Obrázku 10.

Ukázka kódu 20 Úryvek kódu použití kombinace headers/ID [zdroj: autor]

```
<thead>
  <tr>
    <th id="datum">Datum</th>
    <th id="cas">Čas</th>
    <th id="aktivita">Aktivita</th>
    <th id="mistnost">Místnost</th>
    <th id="kapacita">Kapacita</th>
    <th id="instruktor">Instruktor</th>
  </tr>
</thead>
<tbody>
  <tr>
    <td headers="datum">pondělí 27.7.</td>
    <td headers="cas">10:00 - 11:00</td>
    <td headers="aktivita">Pilates</td>
    <td headers="mistnost">Malý cvičební sál</td>
    <td headers="kapacita">4/8</td>
    <td headers="instruktor">M. Michaleová</td>
  </tr>
  [...]
</tbody>
```

Podobně namapujeme odpovídající sloupce záhlaví ke všem datovým buňkám tabulky.

5.7 HTML5 a přístupné formuláře

Formuláře jsou jednou z oblastí kódování, kdy prohlížeč interaguje s uživatelem. Pro zpřístupnění formulářů je tudíž nezbytné, aby uživatel věděl, co a v jaké formě má do příslušné kolonky vyplnit. Mnoho technik z HTML4 lze pro zpřístupnění formulářů nadále využívat. V této kapitole budou připomenuty ty zásadní z hlediska přístupnosti, doplněné o novinky ze specifikace HTML5.

Vytvořený formulář by měl být vždy použitelný při ovládání klávesnicí. Pro fyzicky handicapované je nepřípustné vyžadovat, aby myši zatrhávali například možnosti check boxů. Pravidlem je umožnit procházení formuláře pomocí šipek nebo tabulátoru. Z důvodu přehlednosti při procházení tabulátorem je vhodné kolonky umísťovat spíše vertikálně pod sebe než za sebou do řádků. [4]

5.7.1 Element label

Element `label` informuje uživatele o tom, co by měl vyplnit do prázdného pole. Funguje podobně, jako popisek formulářové kolonky na papíře, a pro přístupnost je tudíž velmi důležitý. Hlasová čtečka využívá element `label` k tomu, aby postupným čtením popisků provedla zrakově handicapovaného vyplněním celého formuláře. [1]

K programovému spojení popisku s příslušným polem slouží zaběhlá metoda `for/id`. Metoda spočívá v přidání identifikátoru k elementu `input` a atributu `for="identifikator"` s odpovídajícím identifikátorem k elementu `label`. [2]

HTML5 nabízí možnost alternativního zápisu, ve kterém je element `input` s textem popisku obalen elementem `label`. Zápis syntaxe této metody lze vidět v Ukázce 23.

5.7.2 Typy elementu input

Pro vytváření formulářů je klíčový element `input`. Slouží jako kolonka pro vkládání dat (jména, adresy, fotografie, atd.) od uživatele. Pro odlišení druhu těchto dat, obohatila specifikace HTML5 element `input` o nové typy. S každým typem elementu `input` může být napříč prohlížeči nakládáno jinak. Odlišnosti se projevují zejména ve validaci vkládaných dat a specializovaných ovládacích rozhraních, kdy například mobilní prohlížeče přemísťují významnější klávesy pro daný typ pole na výhodnější pozice. Příkladem nových typů elementu `input` jsou [11]:

- URL – Validované pole pro vyplnění URL adresy.
- Email – Pole pro vyplnění emailové adresy. Validace očekává platnou syntaxi ve formátu „neco@domena.xx“.
- Telephone – Pole pro vyplnění telefonního čísla. Ve výchozím stavu není validováno, jelikož syntaxe ve světě není jednotná. Pro zadávání zobrazí mobilní prohlížeče přímo číselník namísto standardní klávesnice.
- Hidden – Typ `hidden` reprezentuje hodnotu, která má být skryta koncovému uživateli.

- Date – Zobrazí rozhraní kalendáře pro zadání data.

Ukázka kódu 21 Zápís nových HTML5 typů elementu input [zdroj: autor]

```
<input type="url" />
<input type="email" />
<input type="tel" />
<input type="hidden" />
<input type="date" />
```

Dobrou zprávou je, že webový prohlížeč, který nezná nový HTML5 input typ nebo atribut, zobrazí výchozí pole input. Ostatní funkcionality tak není nijak narušena. [3]

5.7.2.1 Typ password

Pro vyžádání hesla od uživatele slouží typ password. Projevuje se zejména tím, že vkládané znaky nahrazuje symboly – většinou hvězdičkami (*) nebo tečkami. Uživatel se nemusí obávat, že jeho heslo bude při zadávání vyzrazeno. [48]

Ukázka kódu 22 Zápís elementu input typu password [zdroj: autor]

```
<form>
  <label for="heslo">Heslo:</label>
  <input type="password" id="heslo" name="heslo"/>
</form>
```

Ačkoli je tento typ známý již z HTML4, zajímavá je jeho interpretace hlasovými čtečkami. Nejprve element label informuje nevidomého o účelu prvku. V okamžiku, kdy uživatel přenesse focus na pole input a vkládá znaky, nahrazuje čtečka běžné čtení znaků zvuky klikání. V případě, že prohlížeč nahrazuje znaky hvězdičkami, bude čtečka postupně číst „hvězda, hvězda, hvězda“ a tak dále. Jinými slovy, soukromí nevidomého zůstane zachováno. [3]

5.7.3 Nové formulářové atributy HTML5

5.7.3.1 Atribut placeholder

Nový atribut placeholder může být specifikován na elementech input a textarea. Naznačuje příklad vyplnění kolonky, který má navést uživatele ke

správné formulaci odpovědi. „Text atributu `placeholder` by neměl sloužit jako náhrada elementu `label`,“ upozorňuje Tiffany Brown [1].

Ukázka kódu 23 Použití atributu `placeholder` [zdroj: autor]

```
<label>Email: <input type="email" placeholder="JanNovak@priklad.cz">
</label>
```

Atribut vypíše zašedlý text do pole `input`. Tento text, v závislosti na druhu prohlížeče, zmizí při přenesení focusu na pole nebo při napsání prvního znaku.

Email:

Obrázek 11 Reprezentace atributu `placeholder` v prohlížeči Google Chrome [zdroj: autor]

Některé z prohlížečů, jako například Safari, používají barvu textu v atributu `placeholder` ve špatném kontrastním poměru vůči pozadí. Čtení příliš světlého textu může činit potíže lidem se sníženým barvocitem. Je proto vhodné použít CSS3 selektory k jeho ztmavení. [3]

5.7.3.2 Atribut `required`

Nový HTML5 atribut `required` je aplikovatelný na elementy `input`, `select` a `textarea`. Jedná se zároveň o příklad prázdného atributu. Jeho přidáním k poli požádáme prohlížeč, aby před odesláním formuláře ověřil, zda uživatel toto pole vyplnil. Je-li odesláno pole prázdné, zabrání webový prohlížeč přijetí formuláře. Prohlížeče Opera, Chrome, Firefox a Internet Explorer 10+ navíc nativně zobrazí výstražnou zprávu. Safari do verze 8 nativně pouze zabrání odeslání. Zpětnou vazbu uživateli je nutné doplnit kombinací JavaScriptu a CSS. [1]

Ačkoli podpora HTML5 atributu `required` je velmi dobrá i na poli nejnovějších verzí hlasových čteček, starší technologie jej interpretovat neumí. Řešením je kombinace s atributem `aria-required="true"`, který byl představen v kapitole věnované specifikaci WAI-ARIA. Toto spojení zajistí, že také uživatelé starších asistivních technologií budou informováni o povinnosti vyplnit pole. Nevýhodou tohoto řešení je, že uživatelé hlasových čteček podporujících nové HTML5 atributy budou o povinnosti vyplnění informováni dvakrát. Jednou atributem HTML5, podruhé ARIA atributem. [3]

Naneštěstí existují také uživatelé asistivních technologií, které nepodporují ani jednu z uvedených specifikací. Krajním řešením je zahrnout slovo „[Vyžadováno]“ přímo do textu elementu `label` náležejícímu k povinnému poli. [52]

V závislosti na okruhu uživatelů webu musí každý vývojář zvolit, které řešení je uživatelsky nejpřívětivější.

6 Návrh přístupných webových stránek

S využitím zásad a technik představených v teoretické části budou v této kapitole prezentovány konkrétní ukázky řešení dílčích problémů při tvorbě přístupných webových stránek. Tématem pro realizovaný fiktivní projekt bude návrh webu Speciální základní školy v Poděbradech. Pro demonstrativní účely bude snahou předvést typické prvky, s jejichž tvorbou se webový kodér setkává nejčastěji a aplikovat postupy načerpané z předešlých kapitol. Z důvodu snahy uvést co nejvíce názorných ukázek nemohou některé části zcela odpovídat realitě.

Veškeré výplňové materiály (texty, obrázky, videa) jsou se svolením provozovatelů čerpány z oficiálních stránek školy [39]. Webový projekt je přiložen na CD-ROM a nezávazně provozován na adrese <http://edu.uhk.cz/~skoreto1>.

6.1 Postup testování

V první fázi byl web validován pomocí nástroje WAVE představeného v kapitole 4.5.1 a byly opraveny drobné nedostatky. Dále byla ověřena funkčnost prvků v majoritně používaných webových prohlížečích. Konkrétně se jednalo o verze Google Chrome 44, Opera 31, Mozilla Firefox 40 a Internet Explorer 11. Způsob zobrazení a relevantní funkce byly otestovány také v mobilních verzích prohlížečů Chrome 44, Opera 30 a Firefox 40 pro Android.

V závěrečné fázi bylo provedeno testování hlasovou čtečkou JAWS 16 dle postupu uvedeného v kapitole 4.5.2. Odečítač JAWS byl provozován primárně nad prohlížečem Internet Explorer a Firefox, jelikož mají dle průzkumů organizace WebAIM majoritní podíl využití mezi uživateli hlasových čteček.

Výsledky testování jsou uváděny v souvislosti s dílčími částmi postupu tvorby webového projektu.

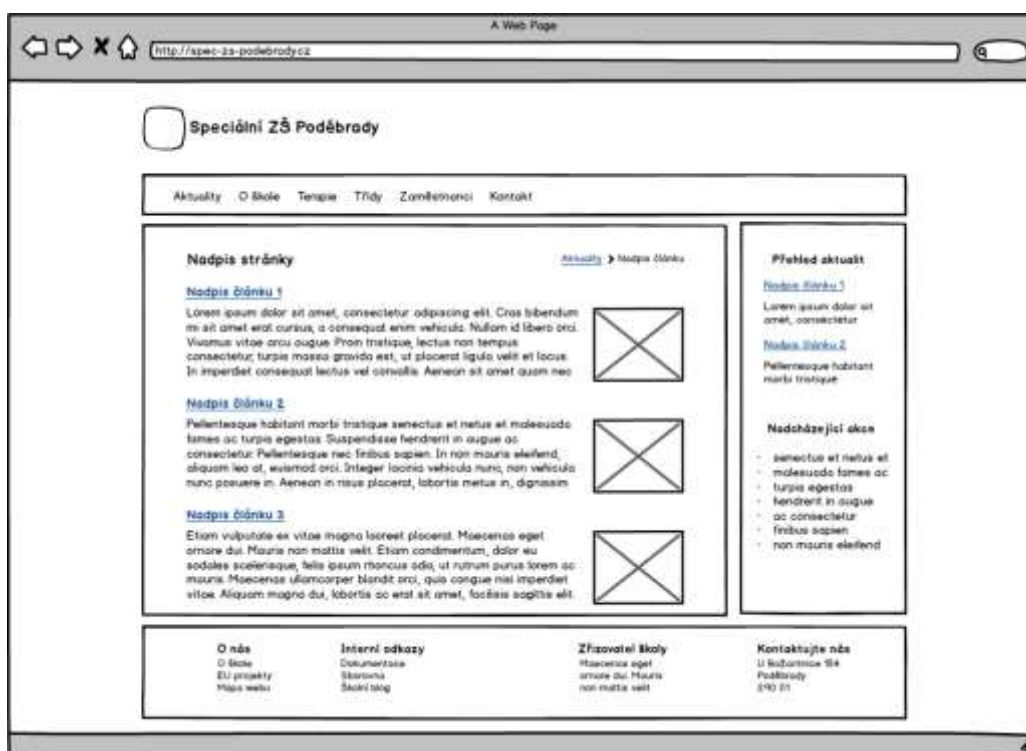
6.2 Návrh základní struktury

6.2.1 Tvorba wireframe

Před započítím samotného kódování struktury webu je vhodné vytvořit si základní koncept rozmístění prvků stránky. K tomuto účelu lze použít papír

s tužkou nebo běžný kreslicí program, ale existuje také specializovaný software na tvorbu tzv. wireframe. Wireframe má charakterem blízko ke skice webu a neobsahuje žádné složité grafické prvky, které by se objevily až v samotném grafickém návrhu. Vytvořením wireframu zpřesníme své představy a porovnáme je s představami zákazníka, čímž efektivně předejdeme neshodám a případné nadbytečné práci. V tuto chvíli tedy nejlépe poslouží systém čar a textů.

V praktické části byl pro vytvoření wireframu použit online nástroj Balsamic Mockups [16], který je ve zkušební verzi poskytován na 30 dní zdarma. Výslednou skicu lze vidět na Obrázku 12.

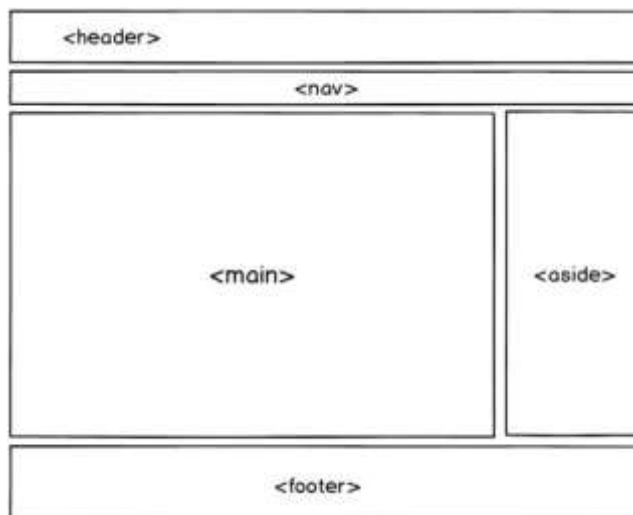


Obrázek 12 Výsledný wireframe [zdroj: autor]

Jedná se o jeden z typických layoutů s horizontálním umístěním navigace a postranním panelem. Z hlediska použitelnosti je vhodné se vyhnout atypickým konstrukcím.

6.2.2 Sémantika HTML5 v základní struktuře webu

Na základě poznatků načerpaných z kapitoly 5.1, věnované sémantickým prvkům HTML5 pro strukturování dokumentu, je třeba vyznačit hlavní části webu.

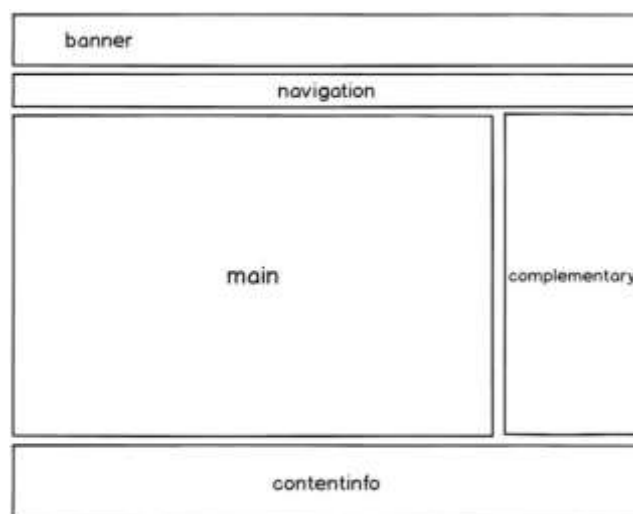


Obrázek 13 Základní HTML5 struktura [zdroj: autor]

Hlavička stránky s logem bude vyznačena elementem `header`, hlavní navigace elementem `nav`, hlavní obsah stránky elementem `main` a patička webu elementem `footer`. K vyznačení postranního panelu s vedlejšími informacemi slouží element `aside`.

6.2.3 Definice ARIA rolí

V tuto chvíli je vhodné doplnit strukturu o definice ARIA rolí, jejichž použití byla věnována kapitola 5.2 o specifikaci WAI-ARIA. Podle ARIA rolí se mohou orientovat asistivní technologie, které nepodporují elementy HTML5. Jejich rozmístění proto v zásadě odpovídá předešlé struktuře.



Obrázek 14 Rozvržení ARIA rolí [zdroj: autor]

Zdrojový kód základní struktury webu bude mít následující podobu.

Ukázka kódu 24 Kombinace HTML5 a ARIA rolí pro základní strukturu webu [zdroj: autor]

```
<!DOCTYPE html>
<head>
  [...]
</head>
<body>
  <header class="header">
    [...]
  </header>
  <nav role="navigation">
    [...]
  </nav>
  <main role="main">
    [...]
  </main>
  <aside role="complementary">
    [...]
  </aside>
  <footer role="contentinfo">
    [...]
  </footer>
</body>
</html>
```

6.3 Hlavní navigace

V současné době je využívání frameworků trendem a jedno z optimalizovaných řešení pro horizontální navigaci nabízí framework Bootstrap. V práci byl využit základ pro tvorbu tzv. „hamburger menu“. Jedná se o responzivní horizontální navigaci, která je na malých úhlopříčkách displeje nahrazena vysunovací vertikální navigací s ovládacím tlačítkem.



Obrázek 15 Standardní rozložení horizontální navigace [zdroj: autor]

Protože Bootstrap potřebuje ke svému fungování knihovnu jQuery je nutné kromě primárních dvou souborů importovat do projektu také ji. Právě zde bude dokázáno, že ani hlasové čtečky nemají v současnosti s obsluhou JavaScriptu potíže. Soubor kaskádových stylů a JavaScriptu v nejnovější verzi 3.3.5 lze volně získat na oficiálních stránkách frameworku Bootstrap [17] a knihovnu jQuery 1.11.3 opět z oficiálního zdroje [25].

Struktura zápisu navigace je do jisté míry předurčena frameworkem Bootstrap a má následující podobu.

Ukázka kódu 25 Zápis navigace s využitím základu frameworku Bootstrap [zdroj: autor]

```
<nav class="main-nav" role="navigation">
  <div class="container">
    <div class="navbar-header">
      <button class="navbar-toggle" type="button" data-toggle="collapse"
data-target="#navbar-collapse">
        <span class="sr-only">Přepnout navigaci</span>
        <span class="icon-bar"></span>
        <span class="icon-bar"></span>
        <span class="icon-bar"></span>
      </button>
    </div>
    <div class="navbar-collapse collapse" id="navbar-collapse">
      <ul class="nav navbar-nav">
        <li class="nav-item active">
          <a href="index.html">Aktuality</a></li>
          [...]
        <li class="nav-item"><a href="kontakt.html">Kontakt</a></li>
      </ul>
    </div>
  </div>
</nav>
```

Celá navigace je vyznačena elementem `nav`. Její obsah je pro účely stylování umístěn do dalších elementů `div`. Element `button`, s referencí na seznam odkazů ve spodní části kódu, představuje tlačítko pro rozbalení/sbalení navigace. Jeho součástí je nápis „Přepnout navigaci“, který díky stylování není standardně zobrazen, ale bude interpretován hlasovou čtečkou. Dále obsahuje tři CSS čáry s účelem graficky znázornit funkci tlačítka. V závěru navigace je vypsán vysunovací seznam odkazů.



Obrázek 16 Vertikální rozložení navigace na malých úhlopříčkách [zdroj: autor]

6.3.1 Testování

Navigace responzivně mění rozložení a je plně funkční v nejnovějších verzích prohlížečů Chrome, Firefox, Opera, Internet Explorer a mobilní verzi prohlížečů Chrome a Opera pro Android. Díky polyfillům HTML5 Shiv a Respond.js není problém s jejím použitím ani ve starších verzích prohlížeče Internet Explorer.

Navigace je plně přístupná ve spolupráci s hlasovou čtečkou JAWS v obou variantách rozložení. Po přenesení focusu na ovládací tlačítko vertikálního rozložení oznámí JAWS uživateli, zda je navigace sbalena či rozbalena. Stiskem mezerníku může uživatel navigaci rozbalit a dále procházet její položky pomocí tabulátoru nebo tabulátorem ihned přeskočit z navigace na procházení obsahu. Že s navigací nebudou mít asistivní technologie problém, dokazuje také její podoba s vypnutými CSS styly.



Obrázek 17 Podoba navigace se zakázanými CSS styly [zdroj: autor]

6.4 Strukturování obsahu

6.4.1 Struktura stránky s aktualitami

Hlavní stránka webu bude obsahovat výňatky novinek školy s odkazy na celé články. Podle kapitoly 5.1 je také tento obsah nutné náležitě strukturovat. Při správném značkování elementy HTML5 budou hlasové čtečky nevidomého informovat mimo jiné o začátku a konci aktuality.

Ukázka kódu 26 Strukturování úryvků aktualit do elementů article [zdroj: autor]

```
<main role="main">
  <article>
    <figure>
      
    </figure>
    <header>
      <h2><a href="pribeh.html">Příběh o červeném
autíčku</a></h2>
    </header>
    <p>Prvňáček Honzík Khas odpověděl na výzvu [...]</p>
    <a href="pribeh.html">Číst více</a>
  </article>
  <article>
    [...]
  </article>
  [...]
</main>
```

Protože úryvky článků představují vzájemně nesouvisející jednotky, je vhodné je strukturovat do elementů `article`. Zjednodušený úryvek zdrojového kódu lze vidět v Ukázce 26.

6.4.2 Strukturování článků

Dlouhé texty v podobě článků v hlavní části webu je vhodné dále strukturovat pomocí nadpisů a sekcí. V praktické části se tento koncept týká například stránek „Terapie“ nebo „Třídy“. Zatímco každý z úryvků aktualit strukturujeme zvlášť do elementu `article`, v případě souvisejícího kontextu bude veškerý text umístěn v jednom elementu `article` a dále dle nadpisů rozdělen elementy `section`.

Ukázka kódu 27 Strukturování celých článků [zdroj: autor]

```
<main role="main">
  <article>
    <header>
      <p>Autor: Lenka Říhová | Publikováno: 10. ledna 2015</p>
    </header>
    <p>Většina terapií je zařazena do projektu [...]</p>
    <section>
      <h2>Canisterapie</h2>
      <figure>
        
        <figcaption>Průběh canisterapie</figcaption>
      </figure>
      <p>Dobrou kamarádkou našich dětí je [...]</p>
    </section>
    <section>
      [...]
    </section>
  </article>
```

```
[...]  
</article>  
</main>
```

6.5 Dokončení návrhu

6.5.1 Výběr barev pro grafický návrh s ohledem na přístupnost

Jedním z kritérií přístupnosti hodnocených metodikou WCAG 2.0 je minimální kontrast mezi textem a pozadím. Například pro prioritu AA musí mít běžné písmo do velikosti 18 bodů vůči svému pozadí kontrast minimálně 4,5:1. Při tomto poměru nemají se čtením textu problém ani uživatelé s lehkým postižením zraku.

Pro měření dostatečného kontrastu existuje řada nástrojů. V praktické části byl využit program Colour Contrast Analyser dostupný zdarma na stránkách agentury The Paciello Group [41].



Obrázek 18 Použití nástroje Colour Contrast Analyser [zdroj: autor]

Program vyhodnotí kontrastní poměr zadaných barev a přímo porovná výsledek s kritérii metodiky WCAG 2.0. Tento postup je nutné aplikovat na volbu barev veškerých oblastí s textem, včetně obrázkových textů, jakým může být logo stránek. Dle českých pravidel přístupnosti by barevná kombinace měla vyhovovat prioritě „AA“, pokud splňuje také prioritu „AAA“, je kontrast mimořádně dobrý.

6.5.2 Systém mřížky a responzivní webdesign

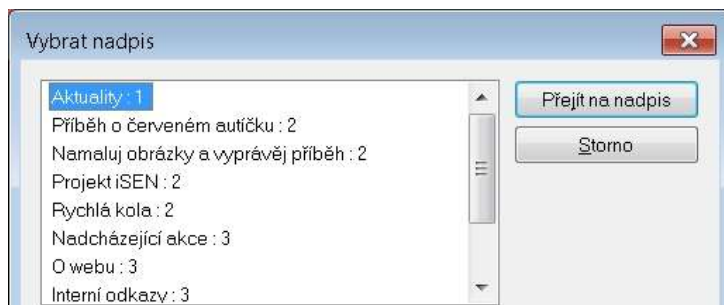
Pro účely stylování byly do elementů s hlavní sémantikou přidány nesémantické elementy `div`. Jejich rozmístění se odvíjí především od systému mřížky frameworku Bootstrap, který byl aplikován. Ten, jak bylo popsáno v kapitole 5.3.3, zaručuje vyvážené rozestupy mezi prvky stránky a výrazně přispívá k lepší použitelnosti webu.

K dosažení responzivního webdesignu využívá nadstavba kaskádových stylů nad frameworkem Bootstrap CSS3 vlastnosti media queries. Jejich konkrétní rozvržení bylo příkladem pro kapitolu 5.3.1. Kapitola 5.3.2 uvedla nastavení viewportu a funkčnosti HTML5 a CSS3 ve starších verzích prohlížeče Internet Explorer dosahuje projekt využitím polyfillů představených v kapitole 3.7.

6.5.3 Testování

Web zachovává konzistentní strukturu minimálně do přiblížení o 200%. Díky systému mřížky je jeho obsah přehledně členěn a zásluhou responzivního webdesignu je použitelný i na mobilních zařízeních. Tam jeho rozvržení převezme jednosloupcovou podobu a vycentruje blokové prvky na střed.

Po příchodu na stránku oznámí hlasová čtečka JAWS titulek stránky, počet oblastí, počet nadpisů a počet odkazů. Veškeré odkazy jsou dostupné z klávesnice. Při jejich procházení pomocí tabulátoru komentuje JAWS, ve které oblasti se uživatel nachází a zda byl již v minulosti odkaz navštíven. Při procházení nadpisů rozeznává JAWS jejich úroveň, jak lze vidět na seznamu nadpisů.



Obrázek 19 Seznam nadpisů v hlasové čtečce JAWS [zdroj: autor]

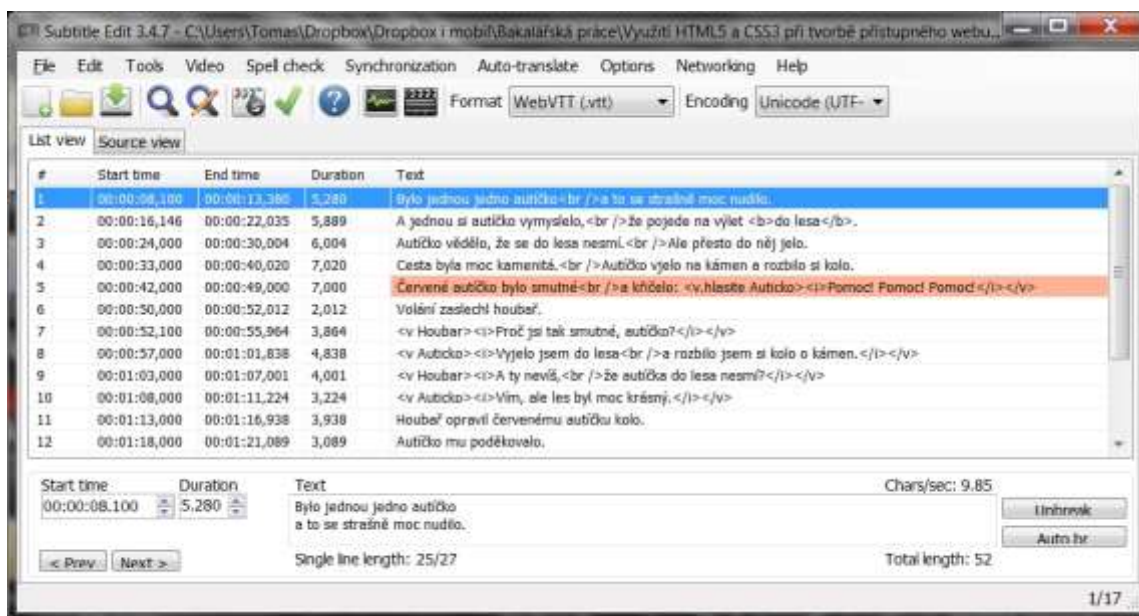
Protože jsou veškeré obrázky opatřeny textovou alternativou nebo popiskem v elementu `figcaption`, bude jejich informační hodnota dostupná i nevidomým.

Uživatel může přeskokovat mezi oblastmi stránky a díky elementu `article` pozná, kde článek začíná a kde končí. V orientaci na webu mu napomůže také drobečková navigace.

6.6 Tvorba legend videa

Na základě znalostí načerpaných z kapitoly 5.5 Přístupnost médií a internacionalizace v HTML5 byla vytvořena ukázka připojení legend k videu využitím elementu `track` a formátu WebVTT. Video přehrává sekvenci obrázků s vyprávěním příběhu „O červeném autíčku“. Bez alternativy zvukové stopy nemá neslyšící uživatel šanci pochopit děj příběhu.

Pro tvorbu legend byl použit opensource editor Subtitle Edit, který podporuje ukládání do formátu WebVTT a je dostupný na serveru GitHub [32].



Obrázek 20 Tvorba legend v editoru Subtitle Edit [zdroj: autor]

Konfiguraci textového editoru je potřeba nastavit tak, aby soubor `.vtt` byl uložen s kódováním UTF-8. Výsledný kód lze nalézt v Příloze 2.

Zdrojový kód připojení souboru legend k videu elementem `track` zobrazuje Ukázka 28 a stylování legend Ukázka 29.

Ukázka kódu 28 Připojení souboru legend k videu [zdroj: autor]

```
<video width="480" height="360"
poster="content/image/poster/o_auticku_poster.png" controls>
  <source src="content/video/o_auticku.mp4" type="video/mp4">
  <track kind="captions" label="Legendy - české" srclang="cs"
src="content/text/captions_cz.vtt" default>
</video>
```

Ukázka kódu 29 Obsah CSS souboru pro stylování legend [zdroj: autor]

```
video::cue(.hlasite) {
  font-size: 1.2em;
}
video::cue(v[voice="Auticko"]) {
  color: #fe5656;
}
video::cue(v[voice="Houbar"]) {
  color: #49ab49;
}
```

6.6.1 Testování

K validaci zápisu legend byl využit bezplatný validátor od vývojáře jménem Anne van Kesteren [27].

```
00:01:48.000 --> 00:01:52.000
Zazvonil zvonec
a pohádky je <b>konec!</b>
```

Kind: subtitles/captions/descriptions ▾

This is boring, your WebVTT is valid! (1ms)

Obrázek 21 Validace kódu WebVTT [zdroj: autor]

V prohlížečích byla ověřena plná podpora HTML5 videa ve formátu MP4 a připojených legend ve formátu WebVTT. Složitější je situace u stylování popisek.



Obrázek 22 Zobrazení legend v nativním přehrávači prohlížeče Opera [zdroj: autor]

Zatímco počítačové i mobilní verze prohlížečů Opera a Chrome podporují elementy pro úpravu písma i použití kaskádových stylů, Firefox dosud neimplementuje kaskádové styly a Internet Explorer ani elementy pro úpravu písma. Nativní přehrávač prohlížeče Internet Explorer navíc popisky nepodbarvuje.



Obrázek 23 Zobrazení legend v prohlížeči Firefox (vlevo) a Internet Explorer [zdroj: autor]

Z Obrázku 23 je tedy patrné, že pro odlišení hlasu vypravěče od hlasů postav lze v prohlížeči Firefox použít alespoň kurzívu. Pro sofistikovanější řešení je nutné využít některou z javascriptových knihoven, jakou je například Captionator.js.

6.7 Interaktivní transkripce

Pro tvorbu interaktivních transkripcí žádnou přímočarou cestu HTML5 nenabízí. S využitím ARIA atributů a JavaScriptu byla v praktické části implementována jednoduchá ukázka interaktivního přepisu videa o projektu iSEN.

Video obsahuje vestavěné anglické titulky, tudíž by se s vloženými českými titulky navzájem překrývaly. Přepis v této formě je snadněji indexovatelný a neslyšícím uživatelům umožňuje získat informace ze zvukové stopy i bez přehrání videa.

6.7.1 Testování

V průběhu přehrávání videa jsou posouvány dialogy k hornímu okraji panelu transkripce. Neslyšící tak pozná, kdo v danou chvíli hovoří. Kliknutím na dialog dojde k posunu času videa na danou pozici.

Interaktivní transkripce je funkční ve všech prohlížečích se zapnutým JavaScriptem. Při vypnutém JavaScriptu bude přepis dostupný alespoň formou prosté transkripce.

6.8 Přístupná tabulka

Dle kapitoly 5.6 věnované přístupnosti tabulek byla navržena tabulka pro přehled tříd školy s odpovědnými pedagogy v sekci „Zaměstnanci“.

Přehled tříd s odpovědnými pedagogy

Třída	Program	Třídní učitelka	Další pedagog ve třídě
1.A přípravný stupeň	speciální škola	Šárka Kolářová	Iva Jelínková, Štěpánka Matoušková
2.A přípravný stupeň	speciální škola	Gabriela Bečvářová	Zuzana Nykodýmová, Jana Férová
3.A přípravný stupeň	speciální škola	Andrea Havlinová	Martina Nováková, Dana Borová
4.A rehabilitační	speciální škola	Lucie Nováková	Žaneta Martinková, Šárka Fialová
5.A kombinované postižení 1	speciální škola	Vendula Šukalová	Monika Fingrová, Lenka Černá
6.A kombinované postižení 2	speciální škola	Martina Běhalíková	Květa Soukalová, Vladka Zmeškalová
7.A auti 1	speciální škola	Jitka Šípková	Míla Holá, Pavlína Reissigová
8.A auti 2	speciální škola	Andrea Marešová	Jitka Stesková, Petra Lochmanová
9.A auti 3	speciální škola	Jitka Teslíková	Marika Bendová
1.B (0-1. ročník)	praktická škola	Kristina Benešová	Hanka Čáповá, Jan Suchánek
2.B (2-3. ročník)	praktická škola	Blanka Kvízová	Libuše Pátová
3.B (4-5. ročník)	praktická škola	Lenka Čvančarová	Kateřina Jirmanová

Tabulka poskytuje přehled aktuálně provozovaných ročníků a druhů tříd s odpovědnými pedagogy. V prvním sloupci je zaznamenán ročník a zaměření třídy. Druhý sloupec rozlišuje, pod který studijní programu třída spadá. Ve třetím sloupci naleznete jméno třídní učitelky a ve čtvrtém jména pomocných pedagogů v dané třídě.

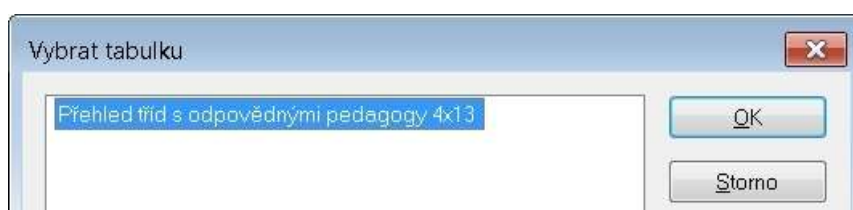
Obrázek 24 Tabulka přehledu tříd s odpovědnými pedagogy [zdroj: autor]

Tabulka je opatřena titulkem v elementu `caption` a jsou nadefinovány vazby mezi jejími datovými buňkami a odpovídajícím záhlavím metodou `headers/ID`. Protože se jedná o poměrně rozsáhlá data, je její smysl shrnut v odstavci pod tabulkou a programově s ní svázán ARIA atributem `aria-describedby`. Zdrojový kód sémantického zpřístupnění této tabulky lze nalézt v Příloze 2.

6.8.1 Testování

V momentě, kdy hlasová čtečka JAWS narazí na tabulku, oznámí tuto skutečnost uživateli společně s počtem jejích sloupců a řádků. Ihned poté přečte její titulek a následuje postupné předčítání záhlaví a obsahu datových buněk.

Uživatel má možnost přenést se na tabulku stiskem tlačítka T nebo vyvolat seznam všech tabulek na stránce. Díky využití elementu `caption` uvidí v seznamu titulky tabulek.



Obrázek 25 Seznam tabulek na stránce v hlasové čtečce JAWS [zdroj: autor]

Dále má možnost procházet tabulku držením kombinace kláves CTR + ALT a stiskem šipek. Značení metodou `headers/ID` zajišťuje, že uživateli je vždy oznámeno, pod jaké záhlaví a řádek buňka patří.

6.9 Přístupný formulář

Na základě teorie z kapitoly 5.7 byl sestaven formulář k aktualitě „Namaluj obrázky a vyprávěj příběh“. Využito bylo nových HTML5 typů elementu `input` pro e-mail, telefon, datum, čas a URL adresu. K vylepšení přístupnosti a použitelnosti byly využity také elementy `fieldset` a `legend`, kterými lze formulář rozdělit na logicky související celky, a formulářová pole byla provázána s popisky metodou `for/id`. Zdrojový kód formuláře z Obrázku 26 lze nalézt v Příloze 3.

Kontakt na Vás

Jméno dítěte [vyžadováno]	<input type="text" value="Pěta Havel"/>
E-mail [vyžadován]	<input type="text" value="marketa.havlova@seznam.cz"/>
Telefon	<input type="text" value="+420 xxx xxx xxx"/>

Informace o příběhu

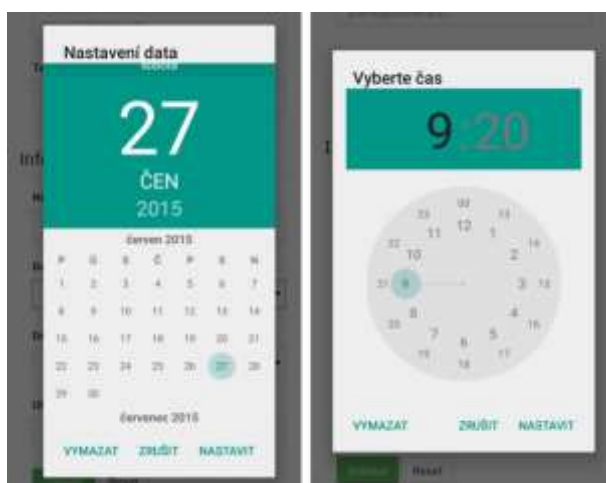
Název příběhu [vyžadován]	<input type="text" value="O šišlavém kocouru"/>
Datum vytvoření příběhu	<input type="text" value="dd.mm.rrrr"/>
Délka příběhu	<input type="text" value="--:--"/>
URL adresa videa [vyžadována]	<input type="text" value="http://"/>

! Vyplňte prosím toto pole.
Zadejte datum ve formátu
dd.mm.rrrr (den, měsíc a rok)

Obrázek 26 Formulář s využitím nových HTML5 typů elementu input [zdroj: autor]

6.9.1 Testování

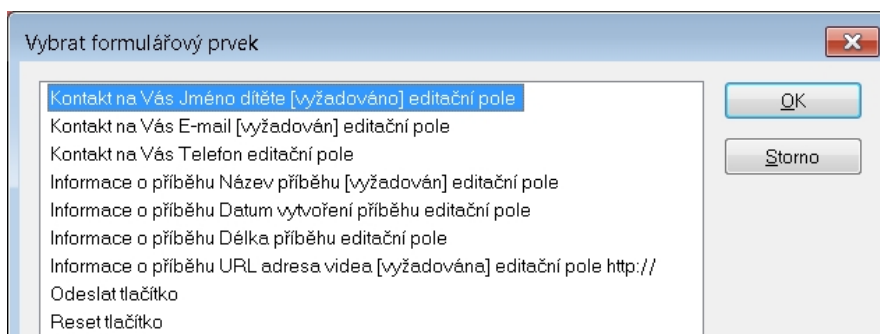
Podpora HTML typů elementu `input` je mezi prohlížeči různá. Prohlížeč Chrome nabídne pro vložení data rozhraní kalendáře a čas lze vkládat pomocí ovládacích šipek. Specifické rozhraní zobrazí také mobilní verze prohlížeče Chrome i Firefox. Tyto prohlížeče odhalí u zmíněných polí nesprávný formát zadání.



Obrázek 27 Mobilní rozhraní pro typ data (Chrome) a času (Firefox) [zdroj: autor]

Desktopové verze prohlížeče Firefox a Internet Explorer sice žádné speciální ovládací rozhraní nenabídnou, také ony ale upozorní na nevyplněné pole s atributem `required` a bez přídatné validace odhalí nesprávný formát e-mailu a URL adresy. Při vkládání telefonního čísla zobrazí mobilní prohlížeče namísto běžné klávesnice číselník.

Hlasová čtečka JAWS dokáže u takto sémanticky značeného formuláře nabídnout uživateli přehled formulářových prvků z Obrázku 28. Pro každý prvek předloží název oblasti `fieldset`, do které patří, popis, který se s ním pojí, typ prvku a případně doplní hodnotu atributu `value`.



Obrázek 28 Přehled formulářových prvků v hlasové čtečce JAWS [zdroj: autor]

Při vyplňování formuláře přečte obsah atributu `title`, dokáže rozpoznat atribut `required` a oznámí tuto skutečnost uživateli slovem „povinné“. Stejně tak upozorní na pole vyplněné v nesprávném formátu. U polí se specifickým rozhráním navíc pečlivě provádí uživatele výběrem dne, měsíce, roku a času.

7 Shrnutí výsledků

Praktická část se zabývá řešením vybraných problémů při tvorbě přístupného webového projektu. V posloupnosti byl zachycen proces tvorby počátečního návrhu webu v podobě wireframu, realizace přístupné navigace, strukturování dokumentů a obsahu, zpřístupnění tabulky a formuláře. Pro zpřístupnění multimédií byla předvedena technika tvorby legend s formátem WebVTT a interaktivní transkripce. V konečné fázi byl web validován nástrojem WAVE a byly podrobně rozebrány výsledky testování dílčích problémů v majoritních zástupcích desktopových a mobilních prohlížečů a hlasovou čtečkou JAWS 16.

Využití nových strukturálních elementů HTML5 značně snížilo tzv. předivování, často patrné v minulosti, a přidalo obsahu další sémantický význam. Z tohoto faktu dokáže těžit hlasová čtečka JAWS 16, která rozpoznává oblasti stránky a pomáhá tak uživateli v lepší orientaci na webu. Pro starší verze asistivních technologií byl web obohacen ARIA rolemi.

Realizovaný responzivní web se vhodně rozloží na obrazovkách různých druhů zařízení a přispívá tak k lepší použitelnosti především na telefonech. Jeho filosofii ocení všichni ti, kteří přibližují webové stránky, ať už nativně v prohlížeči nebo softwarovou lupou. Přidanou hodnotou k použitelnosti je také mřížkový systém CSS frameworku Bootstrap 3.3.5. Ten zaručuje vyvážené rozestupy mezi prvky stránky. Komplikace s responzivním návrhem a využitím HTML5 elementů nastanou ve verzích prohlížeče Internet Explorer nižších než 9. Řešením bylo využití polyfillů HTML5 Shiv a Respond.js, které tyto nedostatky napravují.

Začlenění responzivní navigace dokazuje, že také sofistikovaná řešení CSS frameworků mohou přispět k lepší přístupnosti. Standardní horizontální navigace mění na mobilních zařízeních a při velkém zvětšení rozložení na vertikální vysunovací navigaci. Přesto je stále zaručena její přístupnost pomocí informací určených přímo hlasovým čtečkám.

Uživatel bez výrazných zrakových vad nemá při pohledu na přehledně uspořádanou tabulku potíže asociovat datové buňky s odpovídajícím záhlavím a pochopit význam dat. Nevidomý se však může při procházení buněk tabulky

snadno ztrácet. Také pro tabulky proto existují zásady správné tvorby. Realizovaný příklad byl obohacen o titulek, souhrn významu dat a datové buňky byly programově asociovány s odpovídajícím záhlavím. Hlasová čtečka JAWS poté dokázala předkládat informace ve formě vhodné pro nevidomé.

Komplikovanější výsledky odhalilo testování zpřístupnění videí. Všechny testované prohlížeče podporují zobrazení legend ve formátu WebVTT, jejich konkrétní reprezentace je však značně nejednotná. Zatímco prohlížeče Chrome a Opera i v mobilních verzích aplikují na popisky veškeré kaskádové styly, Firefox podporuje pouze elementy pro úpravu písma a Internet Explorer jen popisky zobrazí. Nativní přehrávač prohlížeče Internet Explorer navíc popisky nepodbarvuje.

Pro tvorbu interaktivních transkripcí žádnou přímočarou cestu HTML5 nenabízí. S využitím ARIA atributů a JavaScriptu byla v praktické části implementována jednoduchá ukázka interaktivního přepisu videa s nativně vestavěnými anglickými titulky. Při vypnutém JavaScriptu je přepis stále přístupný formou prosté transkripce.

Posledním testovaným řešením bylo využití HTML5 elementů při tvorbě přístupného formuláře. V tomto případě podal nejlepší výsledky prohlížeč Chrome. Ten reaguje na nové typy elementu input poskytnutím specifických rozhraní a ve spolupráci s hlasovou čtečkou provede jejich ovládním i nevidomého uživatele. Obdobně reagují také mobilní verze prohlížečů Chrome, Opera a Firefox. Hlasová čtečka JAWS i všechny testované prohlížeče byly bez přídavného JavaScriptu schopny reagovat na nevyplněné povinné pole a oznámit tuto skutečnost uživateli.

8 Závěry a doporučení

Téma přístupnosti webu je velmi komplexní a zahrnuje jak znalosti technické, tak znalosti netechnické. Webový kodér musí často zvažovat volbu vhodného postupu a striktní dodržování pravidel validátorů a metodik nemusí v konečném důsledku představovat pro uživatele optimální řešení. Ke správnému rozhodnutí mu může napomoci povědomí o reálných potřebách handicapovaných uživatelů a fungování asistivních technologií, které při práci s webem používají.

Lidem s postižením zlepšil přístupný Internet značně kvalitu života. Nevidomí jsou nyní schopni provést mnohé úřední úkony bez nutnosti podstupovat riziko pohybu v neznámém prostředí, fyzicky handicapovaným rozšířil možnosti pracovního uplatnění, ale z benefitů přístupnosti těží všichni uživatelé Internetu. Důležité je si uvědomit, že uživatelů se specifickými potřebami přístupnosti je mnoho a žádný webový obsah by jim neměl být odepřen.

V práci byly představeny zásady tvorby přístupného webu dle příslušných metodik a zbořeny byly mýty, které často vedou webové kodéry k zanedbání přístupnosti. Prakticky byly předvedeny vybrané novinky, které do oblasti přístupnosti přinášejí moderní technologie, a otestována byla jejich reálná použitelnost. Ukázány byly také možnosti využití řady nástrojů specializovaných na podporu přístupnosti, CSS frameworků, polyfillů HTML5 Shiv s Respond.js a vyvráceno bylo přesvědčení o nepřístupnosti JavaScriptu. Tím byl dosažen hlavní cíl práce.

Ačkoli je specifikace HTML5 považována za dokončenou, podpora novinek, které přináší, je v prohlížečích mnohdy nejednotná. O to komplikovanější je situace asistivních technologií, které vychází z podkladů dodaných prohlížečem, a jsou tak vždy o krok pozadu. V konečném důsledku se musí kodér sám přesvědčit o funkčnosti použitého řešení. Výsledky testování jsou však převážně pozitivní.

Na realizovaném webovém projektu byla podrobně zkoumána spolupráce hlasové čtečky JAWS 16 s HTML5 a specifikací WAI-ARIA. Jejich kombinace poskytuje nevidomým dostatečné podklady pro práci s webem. Zpřístupněna byla hlavní struktura stránky, grafický i textový obsah, tabulka i formulář. Ambice zpřístupnit multimediální obsah naplňuje HTML5 ve spolupráci s formátem WebVTT.

Ačkoli tvorba stylovaných titulků není dosud příliš rozšířená, jejich podpora ze strany prohlížečů Chrome a Opera byla bezproblémová.

Pro zlepšení použitelnosti webu byl návrh postaven na filozofii responzivního webdesignu s využitím přelomové vlastnosti CSS3, media queries. Web se proto ideálně rozloží na obrazovkách různých zařízení, včetně mobilních telefonů. Vyváženého rozmístění prvků na webu bylo dosaženo na základě systému mřížky frameworku Bootstrap 3.3.5. Jeho přínos pro handicapované uživatele byl prokázán také využitím responzivní přístupné navigace.

Téma práce bylo velmi sevřené, stanovené jako průnik přínosů moderních technologií do oblasti přístupnosti webu. Z toho důvodu by bylo možné pokračovat zkoumáním řady souvisejících oblastí. Aktuálně je hojně diskutovaným tématem přístupnost dynamických webových aplikací, kde komplikovaná uživatelská rozhraní a aktualizace obsahu činí handicapovaným značné potíže.

Webový frontend se neustále vyvíjí a kodérům se budou nabízet stále nové možnosti pro obohacení webů. Před jejich použitím by však měli být obezřetní a vždy mít na paměti potřeby handicapovaných uživatelů.

9 Seznam použité literatury

9.1 Literární zdroje

- [1] BROWN, Tiffany B, Kerry BUTTERS a Sandeep PANDA. HTML5 okamžitě. 1. vyd. Brno: Computer Press, 2014, 256 s. ISBN 978-80-251-4296-7.
- [2] 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.
- [3] CONNOR, Joshue O. Pro HTML5 accessibility: building an inclusive web. New York: Distributed to the book trade worldwide by Springer Science Business Media, c2012, xix, 365 p. ISBN 80-86815-11-0.
- [4] CUNNINGHAM, Katie. Accessibility Handbook: Making 508 Websites for Everyone. Sebastopol: O'Reilly Media, 2012, xiii, 80 p. ISBN 14-493-2285-9.
- [5] HOGAN, Brian P. HTML5 a CSS3: výukový kurz webového vývojáře. Vyd. 1. Brno: Computer Press, 2011, 272 s. ISBN 978-80-251-3576-1.
- [6] KADLEC, Tim. Responzivní design profesionálně. Vyd. 1. Brno: Zoner Press, 2014, 246 s. Encyklopedie Zoner Press. ISBN 978-80-7413-280-3.
- [7] KRUG, Steve. Web design - nenuťte uživatele přemýšlet!. 2., aktualiz. vyd. Brno: Computer Press, 2006, 167 s. ISBN 80-251-1291-8.
- [8] LAZARIS, Louis. CSS okamžitě. Brno: Computer Press, 2014, 168 s. ISBN 978-80-251-4176-2.
- [9] MACDONALD, Matthew. HTML5: The missing manual: The book that should have been in the box. 2nd edition. Sebastopol: O'Reilly Media, 2013, 518 p. ISBN 978-1-44936-326-0.
- [10] PFEIFFER, Silvia. HTML5 audio a video: kompletní průvodce. Vyd. 1. Brno: Zoner Press, 2011, 350 s. Encyklopedie webdesignera. ISBN 978-80-7413-147-9.
- [11] SIKOS, Leslie. Web Standards: Mastering HTML5, CSS3, and XML. 2nd. New York: Apress, c2014, 524 p. ISBN 978-1-484208-84-7.
- [12] STANÍČEK, Petr. CSS Kaskádové styly: Kompletní průvodce. Praha: Computer Press, 2003, xi, 178 s. ISBN 80-722-6872-4.
- [13] ŠPINAR, David. Tvoříme přístupné webové stránky: připraveno s ohledem na novelu Zákona č. 365/2000 Sb., o informačních systémech veřejné správy. Vyd. 1. Brno: Zoner Press, 2004, 360 s. ISBN 80-868-1511-0.

9.2 Internetové zdroje

- [14] ABZ.cz: slovník cizích slov [online]. 2014 [cit. 2014-11-6]. Hendikep, handicap. Dostupné z: <<http://slovník-cizich-slov.abz.cz/web.php/slovo/hendikep-handicap-hendykep>>.
- [15] Accessible Culture. JAWS, IE and Headings in HTML5 [online]. c2015, poslední revize 17. 10. 2011 [cit. 2014-7-26]. Dostupné z: <<http://accessibleculture.org/articles/2011/10/jaws-ie-and-headings-in-html5>>.
- [16] Balsamiq. Balsamiq Mockups [online]. 2015 [cit. 2014-8-1]. Dostupné z: <<https://balsamiq.com/products/mockups>>.
- [17] Bootstrap. Getting started. [online]. 2015 [cit. 2014-8-1]. Dostupné z: <<http://getbootstrap.com/getting-started/#download>>.
- [18] CZ.NIC. Služba CAPTCHA Help. Jak to funguje [online]. 2013, poslední revize 5. 10. 2012 [cit. 2014-8-3]. Dostupné z: <https://www.captchahelp.cz/jak_to_funguje>.
- [19] Český statistický úřad. Demografické charakteristiky seniorů [online]. 2015, poslední revize 31. 12. 2014 [cit. 2014-11-6]. Dostupné z: <[http://www.czso.cz/csu/2014edicniplan.nsf/t/B700265830/\\$File/310035142d.pdf](http://www.czso.cz/csu/2014edicniplan.nsf/t/B700265830/$File/310035142d.pdf)>.
- [20] DEVERIA, Alexis. Caniuse [online]. c2015, poslední revize 12. 5. 2015 [cit. 2015-7-22]. Dostupné z: <<https://www.w3.org/wiki/HTML/Elements/video>>.
- [21] Freedom Scientific. JAWS ARIA Support. [online]. c2015, poslední revize 28. 10. 2014 [cit. 2014-7-28]. Dostupné z: <<http://www.freedomscientific.com/Content/Documents/Other/JAWS-ARIA-Support.doc>>.
- [22] Freedom Scientific. MAGic Screen Magnification Software. [online]. 2015, [cit. 2014-8-10]. Dostupné z: <<http://www.freedomscientific.com/Products/LowVision/MAGic>>.
- [23] GALOP. Odečítací program JAWS. [online]. 2015 [cit. 2014-2-27]. Dostupné z: <<http://www.galop.cz/jaws>>.
- [24] Google (společnost). Jak funguje vyhledávání: Procházení a indexování [online]. 2015, poslední revize 28. 10. 2014 [cit. 2014-11-4]. Dostupné z: <<http://www.google.com/insidesearch/howsearchworks/crawling-indexing.html>>.
- [25] jQuery. Downloading jQuery. [online]. 2015, [cit. 2014-8-1]. Dostupné z: <<https://jquery.com/download>>.

- [26] KABELKA, Roman. Blind Friendly Web. Testování přístupnosti webových stránek se screenreaderem JAWS. [online]. 2015, poslední revize 28. 8. 2013 [cit. 2014-3-22]. Dostupné z: <<http://ewj.blindfriendly.cz>>.
- [27] KESTEREN, Anne. Live WebVTT Validator. [online]. 2015, poslední revize 26. 6. 2015 [cit. 2014-8-1]. Dostupné z: <<https://quuz.org/webvtt>>.
- [28] LHOTSKÁ, Lenka, Jakub KUŽÍLEK a Olga ŠTĚPÁNKOVÁ. ATIS4all. Asistivní technologie [online]. 2015 [cit. 2014-11-6]. Dostupné z: <<http://atisforum.site44.com/dokumenty/atis4all-studie-definiceateorieat.pdf>>
- [29] Lighthouse international. The Vision Simulator. [online]. 2015 [cit. 2014-11-6]. Dostupné z: <<http://www.lighthouse.org/about-low-vision-blindness/vision-simulator>>.
- [30] MARIOTTI, Silvio. World Health Organization. Global data on visual impairment 2010 [online]. 2012 [cit. 2014-11-6]. 20 Avenue Appia, 1211 Geneva 27, Switzerland. Dostupné z: <<http://www.who.int/entity/blindness/GLOBALDATAFINALforweb.pdf>>.
- [31] Ministerstvo vnitra České republiky. Vyhláška č. 64/2008 Sb [online]. 2015, poslední revize 28. 8. 2013 [cit. 2014-3-22]. Dostupné z: <<http://www.mvcr.cz/clanek/vyhlaska-c-64-2008-sb-o-forme-uverejnovani-informaci-souvisejicich-s-vykonem-verejne-spravy-prostrednictvim-webovych-stranek-pro-osoby-se-zdravotnim-postizenim-vyhlaska-o-pristupnosti-10.aspx>>.
- [32] Niksedk. GitHub. Subtitle Edit 3.4.7. [online]. 2015, poslední revize 22. 7. 2015 [cit. 2014-8-1]. Dostupné z: <<https://github.com/SubtitleEdit/subtitleedit/releases>>.
- [33] PAVLÍČEK, Radek. Co se mi líbí na cajtydne.cz z hlediska přístupnosti. [online]. 2015, poslední revize 28. 7. 2014 [cit. 2014-7-26]. Dostupné z: <<http://poslepu.cz/co-se-mi-libi-na-cajtydne-cz-z-hlediska-pristupnosti>>.
- [34] PAVLÍČEK, Radek. Přednáška o přístupnosti HTML5 v kombinaci s WAI-ARIA [online]. 2012, poslední revize 10. 11. 2012 [cit. 2014-11-4]. Dostupné z: <<http://www.slideshare.net/radlicek/pristupnost-html5vkombinaciswaiaria>>.
- [35] PAVLÍČEK, Radek. Přístupnost HTML5 (seriál) [online]. 2015, poslední revize 12. 11. 2012 [cit. 2015-4-18]. Dostupné z: <<http://www.zdrojak.cz/clanky/pristupnost-html5-strukturovani-obsahu>>.
- [36] PAVLÍČEK, Radek. Přístupnost webových stránek orgánů veřejné správy – otázky a odpovědi [online]. 2015, poslední revize 11. 3. 2015 [cit. 2014-7-26]. Dostupné z: <<http://poslepu.cz/pristupnost-webovych-stranek-organu-verejne-spravy-otazky-a-odpovedi>>.
- [37] PAVLÍČEK, Radek. Přístupný web a jak se vyvarovat chyb. Ministerstvo vnitra České republiky [online]. 2015, poslední revize 2014 [cit. 2014-11-6].

- Dostupné z: <<http://www.mvcr.cz/clanek/pristupny-web-a-jak-se-vyvarovat-chyb.aspx>>.
- [38] PAVLÍČEK, Radek. WCAG 2.0 (seriál) [online]. 2015, poslední revize 13. 5. 2010 [cit. 2014 3 22]. Dostupné z: <<http://www.zdrojak.cz/clanky/wcag-2-0-zaciname/>>.
- [39] Speciální základní škola Poděbrady. Oficiální stránky [online]. 2015 [cit. 2015-6-2]. Dostupné z: <<http://www.spec-skola.cz>>.
- [40] ŠPINAR, David. Charakteristika a výhody přístupnosti: Hendikepovaný uživatel Internetu [online]. 2015, poslední revize 28. 10. 2014 [cit. 2014-11-4]. Dostupné z: <<http://pristupnost.nawebu.cz/texty/charakteristika-vyhody.php>>.
- [41] The Paciello Group. Colour Contrast Analyser. [online]. 2015, poslední revize 7. 8. 2015 [cit. 2014-8-1]. Dostupné z: <<http://www.paciellogroup.com/resources/contrastanalyser>>.
- [42] Web Accessibility Initiative (WAI). Financial Factors in Developing a Web Accessibility Business Case for Your Organization [online]. 2012 [cit. 2014-11-5]. Dostupné z: <<http://www.w3.org/WAI/bcase/fin.html#seo>>.
- [43] Web Accessibility Initiative (WAI). WAI-ARIA Overview [online]. 2015, poslední revize 12. 6. 2014 [cit. 2014-7-26]. Dostupné z: <<http://www.w3.org/WAI/intro/aria.php>>.
- [44] Web Accessibility in mind (WebAIM). Accessible CSS. [online]. 2015, poslední revize 30. 8. 2013 [cit. 2014-8-4]. Dostupné z: <<http://webaim.org/techniques/css>>.
- [45] Web Accessibility in mind (WebAIM). Screen Reader User Survey #5 Results [online]. 2015, poslední revize 26. 2. 2014 [cit. 2014-7-26]. Dostupné z: <<http://webaim.org/projects/screenreadersurvey5>>
- [46] Web Accessibility in mind (WebAIM). WAVE – web accessibility evaluation tool. Checklist [online]. 2015 [cit. 2014-3-22]. Dostupné z: <<http://wave.webaim.org>>.
- [47] Web Accessibility in mind (WebAIM). WebAIM's WCAG 2.0 Checklist [online]. 2015, poslední revize 28. 8. 2013 [cit. 2014-3-22]. Dostupné z: <<http://webaim.org/standards/wcag/checklist>>.
- [48] World Wide Web Consortium (W3C). HTML5: A vocabulary and associated APIs for HTML and XHTML [online]. 2015, poslední revize 28. 10. 2014 [cit. 2014-7-24]. Dostupné z: <<http://www.w3.org/TR/html5/dom.html>>.
- [49] World Wide Web Consortium (W3C). HTML5: Techniques for providing useful text alternatives. [online]. 2015, poslední revize 5. 10. 2012 [cit. 2014-7-16]. Dostupné z: <<http://dev.w3.org/html5/alt-techniques>>.

- [50] World Wide Web Consortium (W3C). Inaccessibility of CAPTCHA. Alternatives to Visual Turing Tests on the Web [online]. 2015, poslední revize 23. 11. 2005 [cit. 2014-8-3]. Dostupné z: <<http://www.w3.org/TR/turingtest>>.
- [51] World Wide Web Consortium (W3C). Supported States and Properties [online]. 2015, poslední revize 28. 10. 2014 [cit. 2014-7-26]. Dostupné z: <http://www.w3.org/TR/wai-aria/states_and_properties>.
- [52] World Wide Web Consortium (W3C). Understanding the Four Principles of Accessibility [online]. 2015, poslední revize 11. 12. 2008 [cit. 2014-3-22]. Dostupné z: <<http://www.w3.org/TR/WCAG20>>.
- [53] World Wide Web Consortium (W3C). WAI-ARIA 1.0 Authoring Practices. Implementing Live Regions [online]. 2014, poslední revize 8. 4. 2014 [cit. 2014-7-26]. Dostupné z: <<http://www.w3.org/WAI/PF/aria-practices>>.
- [54] World Wide Web Consortium (W3C). Web Content Accessibility Guidelines (WCAG) 2.0 [online]. 2015, poslední revize 11. 12. 2008 [cit. 2014-3-22]. Dostupné z: <<http://www.w3.org/TR/UNDERSTANDING-WCAG20/intro.html#introduction-fourprincs-head>>.
- [55] World Wide Web Consortium (W3C). WebVTT: The Web Video Text Tracks Format [online]. 2015, poslední revize 22. 7. 2015 [cit. 2014-8-3]. Dostupné z: <<http://dev.w3.org/html5/webvtt>>.
- [56] World Wide Web Consortium (W3C) Wiki. HTML/Elements/video [online]. 2015, poslední revize 3. 11. 2014 [cit. 2014-7-22]. Dostupné z: <<https://www.w3.org/wiki/HTML/Elements/video>>.

10 Seznam obrázků

Obrázek 1 Plocha, pozorovaná uživatelem s relativně dobrým zrakem [3]	16
Obrázek 2 Plocha, pozorovaná uživatelem trpícím zeleným zákalem (simulace) [3]	16
Obrázek 3 Plocha, pozorovaná uživatelem trpícím retinopatií (simulace) [3].....	17
Obrázek 4 Popisek v elementu figcaption svázan s obrázkem elementem figure [zdroj: autor]	40
Obrázek 5 Porovnání originálu s náhradou loga atributem alt [zdroj: autor]	41
Obrázek 6 Ilustrační ukázka ikon v menu rezervačního systému fitness centra [zdroj: autor]	42
Obrázek 7 Příklady CAPTCHA obrázků [4]	43
Obrázek 8 Ukázka interaktivní transkripce [zdroj: autor]	46
Obrázek 9 Formátované WebVTT titulky [zdroj: autor]	51
Obrázek 10 Tabulka lekcí v rezervačním systému pro fitness centrum [zdroj: autor]	53
Obrázek 11 Reprezentace atributu placeholder v prohlížeči Google Chrome [zdroj: autor]	58
Obrázek 12 Výsledný wireframe [zdroj: autor]	61
Obrázek 13 Základní HTML5 struktura [zdroj: autor].....	62
Obrázek 14 Rozvržení ARIA rolí [zdroj: autor].....	62
Obrázek 15 Standardní rozložení horizontální navigace [zdroj: autor].....	63
Obrázek 16 Vertikální rozložení navigace na malých úhlopříčkách [zdroj: autor]..	64
Obrázek 17 Podoba navigace se zakázanými CSS styly [zdroj: autor]	65
Obrázek 18 Použití nástroje Colour Contrast Analyser [zdroj: autor]	67
Obrázek 19 Seznam nadpisů v hlasové čtečce JAWS [zdroj: autor]	68
Obrázek 20 Tvorba legend v editoru Subtitle Edit [zdroj: autor]	69

Obrázek 21 Validace kódu WebVTT [zdroj: autor].....	70
Obrázek 22 Zobrazení legend v nativním přehrávači prohlížeče Opera [zdroj: autor].....	70
Obrázek 23 Zobrazení legend v prohlížeči Firefox (vlevo) a Internet Explorer [zdroj: autor]	71
Obrázek 24 Tabulka přehledu tříd s odpovědnými pedagogy [zdroj: autor].....	72
Obrázek 25 Seznam tabulek na stránce v hlasové čtečce JAWS [zdroj: autor].....	73
Obrázek 26 Formulář s využitím nových HTML5 typů elementu input [zdroj: autor]	74
Obrázek 27 Mobilní rozhraní pro typ data (Chrome) a času (Firefox) [zdroj: autor]	74
Obrázek 28 Přehled formulářových prvků v hlasové čtečce JAWS [zdroj: autor]....	75

11 Seznam ukázek kódů

Ukázka kódu 1 Deklarace cesty k polyfillům HTML5 Shiv a Respond [zdroj: autor] 8	
Ukázka kódu 2 Zápis záhlaví s navigačním panelem ve starších verzích HTML [zdroj: autor]	26
Ukázka kódu 3 Zápis záhlaví s navigačním panelem v HTML5 [zdroj: autor]	27
Ukázka kódu 4 Zápis zápatí ve starších verzích HTML [zdroj: autor].....	30
Ukázka kódu 5 Zápis zápatí v HTML5 [zdroj: autor]	31
Ukázka kódu 6 Zápis s použitím pouze nadpisů úrovně 1 [15]	32
Ukázka kódu 7 Zápis s použitím nadpisů různé úrovně [15]	32
Ukázka kódu 8 Tvorba zástupného obsahu animace v elementu canvas [zdroj: autor]	33
Ukázka kódu 9 Využití CSS3 vlastnosti media queries pro rozdělení úhlopříček [zdroj: autor]	37

Ukázka kódu 10 Nastavení viewportu pro účely responzivního designu [8]	38
Ukázka kódu 11 Použití elementů figure a figcaption [zdroj: autor].....	39
Ukázka kódu 12 Zjednodušený úryvek kódu ikon v menu [zdroj: autor]	43
Ukázka kódu 13 Kód pro vyvolání nativního přehrávače [zdroj: autor].....	47
Ukázka kódu 14 Připojení alternativních textových souborů [zdroj: autor].....	49
Ukázka kódu 15 Začátek souboru WebVTT [zdroj: autor]	50
Ukázka kódu 16 Formátování textů v souboru WebVTT [zdroj: autor].....	50
Ukázka kódu 17 Úryvek CSS pro formátování WebVTT titulků [zdroj: autor].....	52
Ukázka kódu 18 Zjednodušený úryvek zápisu caption [zdroj: autor]	54
Ukázka kódu 19 Zjednodušený úryvek použití atributu aria-describedby [zdroj: autor]	54
Ukázka kódu 20 Úryvek kódu použití kombinace headers/ID [zdroj: autor].....	55
Ukázka kódu 21 Zápis nových HTML5 typů elementu input [zdroj: autor]	57
Ukázka kódu 22 Zápis elementu input typu password [zdroj: autor].....	57
Ukázka kódu 23 Použití atributu placeholder [zdroj: autor].....	58
Ukázka kódu 24 Kombinace HTML5 a ARIA rolí pro základní strukturu webu [zdroj: autor]	63
Ukázka kódu 25 Zápis navigace s využitím základu frameworku Bootstrap [zdroj: autor]	64
Ukázka kódu 26 Strukturování úryvků aktualit do elementů article [zdroj: autor].....	66
Ukázka kódu 27 Strukturování celých článků [zdroj: autor].....	66
Ukázka kódu 28 Připojení souboru legend k videu [zdroj: autor].....	70
Ukázka kódu 29 Obsah CSS souboru pro stylování legend [zdroj: autor]	70

12 Přílohy

1) Výsledný kód WebVTT souboru legend

WEBVTT

00:00:08.100 --> 00:00:13.380

Bylo jednou jedno autíčko
a to se strašně moc nudilo.

00:00:16.146 --> 00:00:22.035

A jednou si autíčko vymyslelo,
že pojede na výlet <u>do lesa</u>.

00:00:24.000 --> 00:00:30.004

Autíčko vědělo, že se do lesa nesmí.
Ale přesto do něj jelo.

00:00:33.000 --> 00:00:40.020

Cesta byla moc kamenitá.
Autíčko vjelo na kámen a rozbilo si kolo.

00:00:42.000 --> 00:00:49.000

Červené autíčko bylo smutné
a křičelo: <v.hlasite Auticko><i>Pomoc! Pomoc! Pomoc!</i></v>

00:00:50.000 --> 00:00:52.012

Volání zaslechl houbař.

00:00:52.100 --> 00:00:55.964

<v Houbar><i>Proč jsi tak smutné, autíčko?</i></v>

00:00:57.000 --> 00:01:01.838

<v Auticko><i>Vyjelo jsem do lesa
a rozbilo jsem si kolo o kámen.</i></v>

00:01:03.000 --> 00:01:07.001

<v Houbar><i>A ty nevíš,
že autíčka do lesa nesmí?</i></v>

00:01:08.000 --> 00:01:11.224

<v Auticko><i>Vím, ale les byl moc krásný.</i></v>

00:01:13.000 --> 00:01:16.938

Houbař opravil červenému autíčku kolo.

00:01:18.000 --> 00:01:21.089

Autíčko mu poděkovalo.

00:01:22.000 --> 00:01:25.042

<v Auticko><i>Jak se ti odvděčím houbaři?</i></v>

00:01:27.000 --> 00:01:30.015

<v Houbar><i>Slib mi, že už do lesa nikdy nepojedeš.</i></v>

00:01:32.000 --> 00:01:36.080
<v Autíčko><i>Slibuji! A jestli chceš,
tak tě odvezu domů.</i></v>

00:01:38.000 --> 00:01:44.192
Červené autíčko odvezlo houbaře domů
a už nikdy do lesa nejelo.

00:01:48.000 --> 00:01:52.000
Zazvonil zvonec
a pohádky je konec!

2) Zápis sémantiky přístupné tabulky

```

<table class="table table-hover table-condensed table-boxed" aria-
describedby="summary">
  <caption>Přehled tříd s odpovědnými pedagogy</caption>
  <thead>
    <tr>
      <th id="trida">Třída</th>
      <th id="program">Program</th>
      <th id="ucitelka">Třídní učitelka</th>
      <th id="dalsi">Další pedagog ve třídě</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td headers="trida">1.A přípravný stupeň</td>
      <td headers="program">speciální škola</td>
      <td headers="ucitelka">Šárka Kolářová</td>
      <td headers="dalsi">Iva Jelínková, Štěpánka
Matoušková</td>
    </tr>
    <tr>
      <td headers="trida">2.A přípravný stupeň</td>
      <td headers="program">speciální škola</td>
      <td headers="ucitelka">Gabriela Bečvářová</td>
      <td headers="dalsi">Zuzana Nykodýmová, Jana Férová</td>
    </tr>
    <tr>
      <td headers="trida">3.A přípravný stupeň</td>
      <td headers="program">speciální škola</td>
      <td headers="ucitelka">Andrea Havlinová</td>
      <td headers="dalsi">Martina Nováková, Dana Borová</td>
    </tr>
    <tr>
      <td headers="trida">4.A rehabilitační</td>
      <td headers="program">speciální škola</td>
      <td headers="ucitelka">Lucie Nováková</td>
      <td headers="dalsi">Žaneta Martínková, Šárka Fialová</td>
    </tr>
    <tr>
      <td headers="trida">5.A kombinované postižení 1</td>
      <td headers="program">speciální škola</td>
      <td headers="ucitelka">Vendula Šukalová</td>
      <td headers="dalsi">Monika Fingrová, Lenka Černá</td>
    </tr>
    <tr>
      <td headers="trida">6.A kombinované postižení 2</td>
      <td headers="program">speciální škola</td>
      <td headers="ucitelka">Martina Běhalíková</td>
      <td headers="dalsi">Květa Soukalová, Vladka
Zmeškalová</td>
    </tr>
    <tr>
      <td headers="trida">7.A auti 1</td>
      <td headers="program">speciální škola</td>
      <td headers="ucitelka">Jitka Šípková</td>
      <td headers="dalsi">Míla Holá, Pavlína Reissigová</td>
    </tr>
  </tbody>
</table>

```

```

</tr>
<tr>
  <td headers="trida">8.A auti 2</td>
  <td headers="program">speciální škola</td>
  <td headers="ucitelka">Andrea Marešová</td>
  <td headers="dalsi">Jitka Stesková, Petra Lochmanová</td>
</tr>
<tr>
  <td headers="trida">9.A auti 3</td>
  <td headers="program">speciální škola</td>
  <td headers="ucitelka">Jitka Teslíková</td>
  <td headers="dalsi">Marika Bendová</td>
</tr>
<tr>
  <td headers="trida">1.B (0-1. ročník)</td>
  <td headers="program">praktická škola</td>
  <td headers="ucitelka">Kristina Benešová</td>
  <td headers="dalsi">Hanka Čápková, Jan Suchánek</td>
</tr>
<tr>
  <td headers="trida">2.B (2-3. ročník)</td>
  <td headers="program">praktická škola</td>
  <td headers="ucitelka">Blanka Kvízová</td>
  <td headers="dalsi">Libuše Pátová</td>
</tr>
<tr>
  <td headers="trida">3.B (4-5. ročník)</td>
  <td headers="program">praktická škola</td>
  <td headers="ucitelka">Lenka Čvančarová</td>
  <td headers="dalsi">Kateřina Jirmanová</td>
</tr>
</tbody>
</table>
<p id="summary">
  Tabulka poskytuje přehled aktuálně provozovaných ročníků a druhů
  tříd s odpovědnými pedagogy.
  V prvním sloupci je zaznamenán ročník a zaměření třídy. Druhý
  sloupec rozlišuje, pod který studijní
  programu třída spadá. Ve třetím sloupci naleznete jméno třídní
  učitelky a ve čtvrtém jména
  pomocných pedagogů v dané třídě.
</p>

```

3) Zdrojový kód přístupného formuláře

```

<form>
  <fieldset>
    <legend>Kontakt na Vás</legend>
    <div class="form-group">
      <label class="col-sm-3" for="fl_name_surname">Jméno dítěte
[vyžadováno]</label>
      <div class="col-sm-9">
        <input id="fl_name_surname" type="text" class="form-
control" required />
      </div>
    </div>
    <div class="form-group">
      <label class="col-sm-3" for="fl_email">E-mail
[vyžadován]</label>
      <div class="col-sm-9">
        <input id="fl_email" type="email" class="form-control"
placeholder="jmeno@domena.cz" required />
      </div>
    </div>
    <div class="form-group">
      <label class="col-sm-3" for="fl_telefon">Telefon</label>
      <div class="col-sm-9">
        <input id="fl_telefon" type="tel" class="form-control"
title="Zadejte telefon ve formátu +420 xxx xxx xxx" placeholder="+420
xxx xxx xxx" />
      </div>
    </div>
  </fieldset>
  <fieldset>
    <legend>Informace o příběhu</legend>
    <div class="form-group">
      <label class="col-sm-3" for="fl_story_name">Název příběhu
[vyžadován]</label>
      <div class="col-sm-9">
        <input id="fl_story_name" type="text" class="form-
control" required aria-required="true" />
      </div>
    </div>
    <div class="form-group">
      <label class="col-sm-3" for="fl_cas">Datum vytvoření
příběhu</label>
      <div class="col-sm-9">
        <input id="fl_cas" type="date" class="form-control"
title="Zadejte datum ve formátu dd.mm.rrrr (den, měsíc a rok)"
required aria-required="true" />
      </div>
    </div>
    <div class="form-group">
      <label class="col-sm-3" for="fl_cas">Délka příběhu</label>
      <div class="col-sm-9">
        <input id="fl_cas" type="time" class="form-control"
title="Zadejte čas ve formátu MM.ss (minuty, sekundy)" required aria-
required="true" />
      </div>
    </div>
  </fieldset>

```

```
</div>
<div class="form-group">
  <label class="col-sm-3" for="f1_url">URL adresa videa
[vyžadována]</label>
  <div class="col-sm-9">
    <input id="f1_url" type="url" class="form-control"
value="http://" required aria-required="true" />
  </div>
</div>
<div class="form-group">
  <div class="col-sm-offset-3 col-sm-9">
    <button type="submit" class="btn btn-success pull-
right">Odeslat</button>
    <button type="reset" class="btn btn-default pull-
right">Reset</button>
  </div>
</div>
</fieldset>
</form>
```



UNIVERZITA HRADEC KRÁLOVÉ
Fakulta informatiky a managementu
Rokitanského 62, 500 03 Hradec Králové, tel: 493 331 111, fax: 493 332 235

Zadání k závěrečné práci

Jméno a příjmení studenta:

Tomáš Skořepa

Obor studia:

Aplikovaná informatika

Jméno a příjmení vedoucího práce:

Monika Borkovcová

Název práce:

Využití HTML5 a CSS3 při tvorbě přístupného webu

Název práce v AJ:

Utilisation of HTML5 and CSS3 technologies for development of accessible web sites

Podtitul práce:

Podtitul práce v AJ:

Cíl práce: Cílem práce je provést čtenáře zásadami tvorby přístupného webu s využitím moderních technologií.

Osnova práce:

1. Úvod
2. Cíl a metodika práce
3. Teoretická východiska
4. Přístupnost a použitelnost
5. Optimalizace HTML5 a CSS3 pro prohlížeče
6. Návrh přístupných webových stránek
7. Výsledky
8. Závěry a doporučení
9. Literární zdroje

Projednáno dne: *13.10.2014*

Podpis studenta

Podpis vedoucího práce