ŽILINSKÁ UNIVERZITA V ŽILINE

FAKULTA RIADENIA A INFORMATIKY



BAKALÁRSKA PRÁCA

Štúdijný odbor: Informatika

Jaroslav Ondreják

**Vytvorenie podporného frameworku pre responzívny dizajn**

**Vedúci práce:** Ing. Matej Meško

Reg. Číslo: 13/2014

Žilina, 2015

Čestné prehlásenie

Čestne prehlasujem, že som bakalársku prácu vypracoval samostatne na základe vlastných teoretických a praktických poznatkov, konzultácií a štúdia odbornej literatúry.

V Žiline Dňa: ........................ Podpis: ........................

Poďakovanie

Touto cestou sa chcem poďakovať vedúcemu bakalárskej práce Ing. Matejovi Meškovi za odbornú pomoc, cenné rady a pripomienky, ktoré mi pomohli úspešne vypracovať prácu.

ABSTRAKT

ONDREJÁK, Jaroslav: Vytvorenie podporného frameworku pre responzívny dizajn. [Bakalárska práca] – Žilinská univerzita v Žiline. Fakulta riadenia a informatiky; Štúdijný program: Informatika. - Vedúci práce: Ing. Matej Meško. Stupeň odbornej kvalifikácie: Bakalár informatiky. Žilina: FRI ŽU v Žiline, 2015. Počet strán: X

Cieľom bakalárskej práce je navrhnúť a implementovať podporný framework pre responzívny dizajn. Výsledná knižnica bude použitá ako základ pre tvorbu vizuálnej časti webových stránok, ktorá bude dynamicky reagovať na zmenu veľkosti okna webového prehliadača. Výsledná knižnica dokáže správne kvalifikovať zobrazovacie zariadenie a prispôsobí webovú stránku skutočnej veľkosti displeja zariadenia.

Kľúčové slová: HTML 5, CSS3, JavaScript, responzívny webdizajn, webový prehliadač, webová stránka

ABSTRACT

ONDREJÁK, Jaroslav: Vytvorenie podporného frameworku pre responzívny dizajn. [Bakalárska práca] – Žilinská univerzita v Žiline. Fakulta riadenia a informatiky; Štúdijný program: Informatika. - Vedúci práce: Ing. Matej Meško. Stupeň odbornej kvalifikácie: Bakalár informatiky. Žilina: FRI ŽU v Žiline, 2015. Počet strán: X

Cieľom bakalárskej práce je navrhnúť a implementovať podporný framework pre responzívny design. Výsledná knižnica bude použitá ako základ pre tvorbu vizuálnej časti webových stránok, ktorá bude dynamicky reagovať na zmenu veľkosti okna webového prehliadača. Výsledná knižnica dokáže správne kvalifikovať zobrazovacie zariadenie a prispôsobí webovú stránku skutočnej veľkosti displeja zariadenia.

Kľúčové slová: HTML 5, CSS3, JavaScript, responzívny webdizajn, webový prehliadač, webová stránka

OBSAH

[ÚVOD 1](#_Toc417336643)

[1 ANALÝZA SÚČASTNÉHO STAVU 2](#_Toc417336644)

[1.1 Webová služba 2](#_Toc417336645)

[1.2 Webová stránka a webová aplikácia 2](#_Toc417336646)

[1.2.1 Frontend 3](#_Toc417336647)

[1.2.2 Backend 3](#_Toc417336648)

[1.3 W3C 3](#_Toc417336649)

[1.3.1 HTML 4](#_Toc417336650)

[1.3.2 XHTML 5](#_Toc417336651)

[1.3.3 HTML 5 5](#_Toc417336652)

[1.3.4 CSS 6](#_Toc417336653)

[1.3.5 CSS3 8](#_Toc417336654)

[1.4 JavaScript 8](#_Toc417336655)

[1.4.1 jQuery 9](#_Toc417336656)

[1.5 Webový prehliadač 9](#_Toc417336657)

[1.5.1 Podporné nástroje vykresľovania webových prehliadačov 11](#_Toc417336658)

[1.5.2 Viewport 12](#_Toc417336659)

[1.6 Nástroje na vývoj webových stránok 12](#_Toc417336660)

[1.6.1 PSPad 12](#_Toc417336661)

[1.6.2 Sublime Text 13](#_Toc417336662)

[1.7 Webový dizajn 13](#_Toc417336663)

[1.7.1 Liquid 14](#_Toc417336664)

[1.7.2 Adaptive 15](#_Toc417336665)

[1.7.3 Responsive 15](#_Toc417336666)

[2 ZBER POŽIADAVIEK PRE KNIŽNICU 15](#_Toc417336667)

[2.1 Existujúce riešenia 16](#_Toc417336668)

[2.1.1 Bootstrap 16](#_Toc417336669)

[2.1.2 Foundation 16](#_Toc417336670)

[3 ANALÝZA POŽIADAVIEK PRE KNIŽNICU 16](#_Toc417336671)

[4 IMPLEMENTÁCIA KNIŽNICE 16](#_Toc417336672)

[ZÁVER 16](#_Toc417336673)

[ZOZNAM REFERENCIÍ 17](#_Toc417336674)

[PRÍLOHY 19](#_Toc417336675)

ZOZNAM POUŽITÝCH SKRATIEK

W3C. World Wide Web Consortium

WWW. World Wide Web

# ÚVOD

Používanie internetu je v posledných rokoch veľmi rozšírené. Internet je sieť počítačových uzlov, ktoré sú poprepájané po celom svete. Táto sieť počítačov je pospájaná z miliónov serverov, poskytujúcich svoje služby miliónom používateľom. Počet serverov, zariadení ale aj používateľov neustále rastie a môžeme si dovoliť povedať, že rast sa nikdy nezastaví.

Internet zaznamenal najväčší rozmach zavedením internetovej služby WWW (World Wide Web). Tvorcom tejto služby je anglický počítačový expert, výskumník a profesor Tim Berners-Lee. Základným kameň služby WWW je webová stránka. Webová stránka môže obsahovať všetko čo dokáže počítač spracovať: text, grafika, zvuk, video.

Používatelia sa pripájajú pomocou koncových zariadení, ako sú napríklad počítače, notebooky a v  neposlednej rade mobilnými zariadeniami, ktoré sa stávajú „lídrom“ v prístupe na internet. Každé z týchto zariadení môže mať rôzne rozmery a veľkosť displeja.

Týmto sa dostávame k problému, ako by sme mali stránku formátovať a členiť aby sa dokázala správne zobrazovať na všetkých zariadeniach?

Základom mojej práce je vývoj knižnice, pomocou ktorej programátor s dodržaním jednoduchej štruktúry a pravidiel, zabezpečí správne zobrazenie webovej stránky bez ohľadu na rozmery zariadenia. Knižnica bude vytvorená tak, aby správne zobrazovala stránky na 4 najpoužívanejších prehliadačoch.

# ANALÝZA SÚČASTNÉHO STAVU

Prenos a zobrazovanie webových stránok v službe WWW je založená na architektúre klient-server. Webová stránka je uložená na webovom serveri, ku ktorému pristupujeme prostredníctvom tejto služby v sieti internet. Webová stránka sa zvyčajne zobrazuje pomocou webového prehliadača. Pre zobrazenie webovej stránky je potrebný prenos stránky do webového prehliadača.

## Webová služba

Webová služba je softvér, ktorý poskytuje spôsob kooperácie medzi softvérovými aplikáciami, bežiacimi na rozdielnych platformách[[1]](#footnote-2)alebo frameworkoch[[2]](#footnote-3). S webovou službou zariadenia komunikujú spôsobom, ktorý je predpísaný v popise služby pomocou protokolu SOAP *(Simple Object Access Protocol)*. Rozhranie služby je popísané vo formáte WSDL *(Web Services Description Language).* (1)

## Webová stránka a webová aplikácia

Webová stránka je dokument napísaný v takzvanom značkovacom jazyku. Webová stránka je webový dokument určený pre službu WWW. Webová stránka bola pôvodom určená pre zobrazenie textov, obrázkov, tabuliek, zoznamov a iných primitívnych prvkov. Dnešné web stránky obsahujú rôzne komplikovanejšie prvky. Môžu obsahovať navigačnú lištu pre navigovanie na iné stránky alebo časti webu, slideshow[[3]](#footnote-4) na zobrazenie obrázkov alebo fotografií, rôzne typografické prvky, nadpisy, logá, timeline.

Pre navigovanie na iné stránky sa používajú hypertextové odkazy. Hypertextové odkazy sú prepojením medzi webovými stránkami. Môžu byť napríklad vo formáte textu, obrázku alebo tlačidla.

Webové stránky boli spočiatku písané manuálne ako dokumenty v jazyku HTML, takzvané statické stránky. Stránky je možné generovať pomocou skriptovacích jazykov ako je napríklad PHP alebo ASP, ktoré bežia na strane servera. Tieto skriptovacie jazyky dokážu vložiť rôzny obsah z rôznych externých aj interných zdrojov do webovej stránky, napríklad dáta z relačnej databázy. Keďže takéto stránky sú generované dynamicky, nazývame ich dynamické stránky.

Vlastnosti dynamických stránok sa dajú veľmi dobre využiť ako náhrada pre klasické desktopové aplikácie. Celá logika aplikácie beží na serveri a môžeme ju označiť ako dátovú vrstvu. Používateľské rozhranie (GUI[[4]](#footnote-5)) beží vo webovom prehliadači a ten sám o sebe logiku aplikácie nepozná, označíme ju ako prezentačná vrstva. Aplikácie implementované takýmto spôsobom nazývame webové aplikácie.

Vo vývoji webových aplikácií sa používajú termíny frontend a backend pre oddelenie prezentačnej a dátovej vrstvy.

### Frontend

V oblasti vývoja webových aplikácií termín frontend predstavuje rozhranie užívateľa s ktorým interaktívne pracuje. Frontend je všetko, čo používateľ vidí na svojom zobrazovacom zariadení. Časť frontend beží na klientskej strane vo webovom prehliadači. Medzi hlavné technológie spojené s vývojom frontendu sú HTML, CSS, JavaScript, jQuery.

### Backend

Termín backend je používaný na označenie celej logiky webovej aplikácie, všetko to, čo beží na pozadí a čo používateľ nevidí. Sú to rôzne výpočty, získavanie dát z relačnej databázy, ale aj samotný životný cyklus od zadania URL adresy do prehliadača až po zobrazenie obsahu používateľovi. Pre vývoj backendu sa používa obrovské množstvo technológií, ktoré môžu byť použité samostatne alebo ich kombináciou.

## W3C

http://www.w3.org/standards/

World Wide Web Consortium je medzinárodná komunita nazývaná konzorcium, produkujúca slobodné štandardy, ktoré vyvíjajú a publikujú členské organizácie. Hlavnou úlohou konzorcia je zjednotenie rôznych technológií, ktoré sú používané na internete.

Konzorcium vyvíja technické špecifikácie a postupy na zabezpečenie nezávislosti na prehliadačoch a iných softvérových alebo hardvérových požiadavkách. Zameriava sa na technológie, s ktorými môžeme pristupovať k webu z rôznych miest, kedykoľvek a za použitia rôznych zariadení.

Medzi známe štandardy ktoré zahŕňajú vytvorenie a správne zobrazenie webových stránok patria napríklad: HTML5, XHTML, XML, CSS, DOM, SVG a iné.

Používanie a dodržiavanie týchto štandardov je dobrovoľné.

### HTML

http://www.w3schools.com/html

http://www.w3.org/standards/webdesign/htmlcss

HyperText Markup Language je základný značkovací jazyk, popisujúci webové stránky, respektíve webové dokumenty. Pomocou html popisujeme štruktúru stránky. Značkovací jazyk je súbor značkovacích tagov, respektíve značiek. Každý html tag popisuje odlišnú časť obsahu webového dokumentu. Použitým týchto tagov môže webový prehliadač rozoznať a zobraziť správne časti dokumentu. Každý tag je určený svojím preddefinovaným názvom a je ohraničený lomenými zátvorkami. Tagy môžu byť párové alebo nepárové.

Pre deklaráciu typu dokumentu html sa používa tag <!DOCTYPE>, so špecifickým označením typu dokumentu, ktoré je odlišné v závislosti od štandardu. Typ webového dokumentu býva umiestnený na začiatku. Definíciou tagu <html> začína html dokument a končí tagom </html>. Zobraziteľná časť dokumentu sa nachádza medzi tagmi <body> a </body>.

Obsah ohraničený párovými tagmi rovnakého typu, spolu s týmito tagmi sa nazývajú elementy. Zjednodušene element je všetko od štartovacieho tagu po ukončovací tag. Elementy sa môžu do seba vnárať, respektíve dediť. Znamená to že elementy môžu obsahovať iné elementy. Html elementy neobsahujúce žiadny obsah sa nazývajú prázdne elementy a nemajú ani ukončovací tag. Elementy môžeme rozdeliť do dvoch základných skupín:

* Blokové elementy – zobrazujú sa ako blok, to znamená, že zaberajú celú šírku nadradeného elementu. Vložením viacerých blokových elementov sa zobrazia pod sebou.
* Inline elementy – sú elementy, ktoré zaberajú šírku svojho obsahu. Ak vložíme viac takýchto elementov a ich spoločná šírka sa zmestí do jedného riadku, aj sa tak zobrazia.

Html elementy môžu obsahovať atribúty, ktoré poskytujú rozširujúce informácie o elemente. Atribúty vždy obsahujú iba začiatočné tagy a majú tvar nazov\_atributu=“hotnota\_atributu“ . Každé elementy, majú vlastné atribúty a majú špecifický význam.

### XHTML

http://www.w3schools.com/html/html\_xhtml.asp

Extensible HyperText Markou Language je skoro identický ako HTML. Syntax XHTML zavádza prísnejšie pravidlá pre štruktúru webového dokumentu a je založená na syntaxi XML. Takýto dokument môžeme jednoducho rozšíriť o novú značku alebo atribút. Pre deklaráciu XHTML je povinné uviesť tag <!DOCTYPE> v prvom riadku webového dokumentu. Používanie jednotlivých tagov je prísne definované nasledujúcimi predpismi:

* tagy <html>, <head>, <title> a <body>, sú povinné a musí ich obsahovať každý webový dokument.
* všetky tagy musia byť párové, to znamená že musia mať začiatočný a ukončovací tag.
* žiadny tag nemôže byť ukončený skôr ako jeho detský tag.
* všetky tagy a ich atribúty musia byť písané malým písmom.
* všetky atribúty musia byť v úvodzovkách.

### HTML 5

http://www.w3.org/TR/html5/

Táto špecifikácia je piatou verziu základnej revízie jadra jazyka htmlslužby www. V tejto verzii sú zavedené nové vymoženosti na pomoc vývojárom webových aplikácií. Sú zavedené nové elementy na základe výskumu praktík, ktoré používajú vývojári. Vo verzii HTML5 sú upravené niektoré zo základných tagov a elementov pre jednoduchšie používanie. Nová verzia sa taktiež nevyhla odstránením zastaraných elementov.

Prvotné zmeny sa týkajú tagu, slúžiaceho na deklaráciu samotného štandardu HTML5. Touto zmenou je jeho najjednoduchšia možná forma, <! DOCTYPE html>. Tag na definovanie znakového kódovania je tiež veľmi jednoduchý, píšeme ho v tvare <meta charset=”UTF-8”>.

Niektoré najzaujímavejšie pridané elementy v tejto verzii html sú nasledovné:

* Elementy <header>, <section>, <footer>, <article>, <nav>, a <main>. Všetky tieto elementy sú blokové elementy.
* Atribúty pre ovládanie formulárových prvkov number, date, time, calendar, a range.
* Grafické elementy <svg> a <canvas>.
* Elementy pre podporu mutimédií <audio> a <video>.

Všetky moderné webové prehliadače podporujú verziu HTML5. Každý webový prehliadač, či už starý alebo nový, spracúva neznáme elementy ako inline elementy.

### CSS

http://www.w3.org/Style/CSS/

http://www.w3schools.com/css/

Cascading Style Sheets (CSS) je jednoduchý mechanizmus pridávania štýlov webovým dokumentom. CSS spolu s html tvoria základný kameň tvorby webových stránok. Zatiaľ čo html definuje základnú štruktúru webového dokumentu, CSS definuje ako budú elementy zobrazené. Napríklad na definovanie fontov, farieb, odsadení, okrajov a podobne. CSS šetrí mnoho času pri navrhovaní webových stránok. Jeho syntax a používanie je veľmi jednoduché a intuitívne. Hlavným dôvodom vzniku CSS bolo, že dokumenty html neboli nikdy určené na to, aby obsahovali tagy pre formátovanie dokumentu.

Syntax CSS je založená na písaní pravidiel pre html elementy. Pravidlo je množina obsahujúca selector a blok deklarácie vlastností. Selector ukazuje na html element, na ktorý chceme aplikovať štýl. Blok deklarácie vlastností, môže obsahovať neurčitý počet vlastností oddelených bodkočiarkou, a je ohraničený zloženými zátvorkami. Každá vlastnosť musí obsahovať hodnotu. Hodnoty vlastností môžu byť číselné alebo preddefinované pre danú vlastnosť.

Príklad CSS pravidla:

p { color: blue; }

Selektory umožňujú vyhľadanie html elementov na základe výrazov. Výraz môže definovať výber podľa ich názvu, atribútu id, atribútu class, a mnoho ďalších. Selektory môžu byť poskladané z viacerých poprepájaných výrazov, alebo ich môžeme zgrupovať.

Najpoužívanejšie CSS výrazy selektorov sú:

* Názov elementu – ako selektor použijeme názov elementu, napríklad p { … }.
* Atribút id – ako selektor použijeme názov atribútu id, pred ktorým napíšeme znak „#”, napríklad #hlavny { … }.
* Atribút class – ako selektor použijeme názov atribútu class, pred ktorým napíšeme znak „bodky“, napríklad .hlavny { … }.

Keď webový prehliadač číta CSS, aplikuje štýly na html elementy, čím formátuje webovú stránku. Sú tri hlavné možnosti kde môžu byť štýly uložené:

* Externý súbor.
* Interne v dokumente html.
* Priamo v elemente na ktorý chceme štýl aplikovať.

Najviac používaná je prvá možnosť uloženia CSS v externom súbore. Ak máme externý súbor, jednoducho môžeme zmeniť vzhľad celej webovej stránky zmenením len jedného súboru. Každá stránka, ktorá chce používať dané štýly, musí obsahovať referenciu na externý súbor. Pre tento účel slúži <link> element, umiestnený v sekcii head html stránky. Externý súbor s CSS musí byť ukončený koncovkou .css.

Štýl, ktorý chceme aplikovať len na jeden element nazývame inline štýl. Musí byť umiestnený v atribúte style daného elementu a definujeme iba vlastnosti pre daný štýl. Takouto formou definovania štýlov strácame mnoho výhod, preto by sme inline štýly mali používať šetrne.

Pri kombinácií viacerých druhov uloženia CSS štýlov musíme brať do úvahy v akom poradí sú jednotlivé štýly aplikované. Najvyššiu prioritu majú inline štýly, potom externé a interné štýly a posledné v poradí štýly, ktoré sú preddefinované webovým prehliadačom.

### CSS3

http://www.w3schools.com/css/css3\_intro.asp

http://www.w3.org/TR/2001/WD-css3-roadmap-20010523/

CSS3 je posledným štandardom pre CSS. Je kompletne spätne kompatibilný so skoršími verziami CSS. Ako rástla popularita CSS, bol veľký záujem rozšíriť jeho špecifikáciu o nové vlastnosti. Skôr ako boli pridané nové vlastnosti, tvorcovia sa rozhodli zjednotiť mnoho monolytických a individuálnych vlastností do celkov.

CSS3 sa rozčleňuje do takzvaných „modulov“, ktoré obsahujú celú „staršiu“ CSS špecifikáciu. Navyše sú pridané nové moduly, ktoré poskytujú rozsiahlejšiu možnosť formátovania webovej stránky. Niektoré najdôležitejšie CSS3 moduly sú:

* Selektory
* Pozadia a okraje
* Textové efekty
* 2D/3D transformácie
* Animácie

Väčšina modulov a väčšina nových vlastností sú implementované vo všetkých moderných prehliadačoch.

## JavaScript

JavaScript je základný programovací jazyk pre html a celkovo pre web. Pôvodne bol vyvýjaný Brendanom Eichom zo spoločnosti Netscape Communications pod názvom „Mocha“, neskôr pod menom „LiveScript“. Pred uvedením verejnosti bol pomenovaný ako „JavaSript“, najmä vďaka popularite jazyka Java. Reálne neexistuje žiadny významný vzťah medzi jazykmi Java a JavaSript a ani výrazná podoba ich syntaxi. Najväčšia podobnosť je, že sú založené na spoločnom predkovi, jazyku C. JavaScript dokáže mnoho vecí, ktoré sú bežne potrebné pri tvorbe webových stránok.

Spomeniem základné, popredné možnosti JavaSriptu:

* Môže meniť obsah html stránky
* Môže meniť html atribúty
* Môže meniť štýly html elementov
* Môže validovať vstupné dáta od používateľa

Nad programovacím jazykom JavaScript existuje naprogramovaných množstvo knižníc. Najdôležitejšie je však spomenúť knižnicu jQuery.

### jQuery

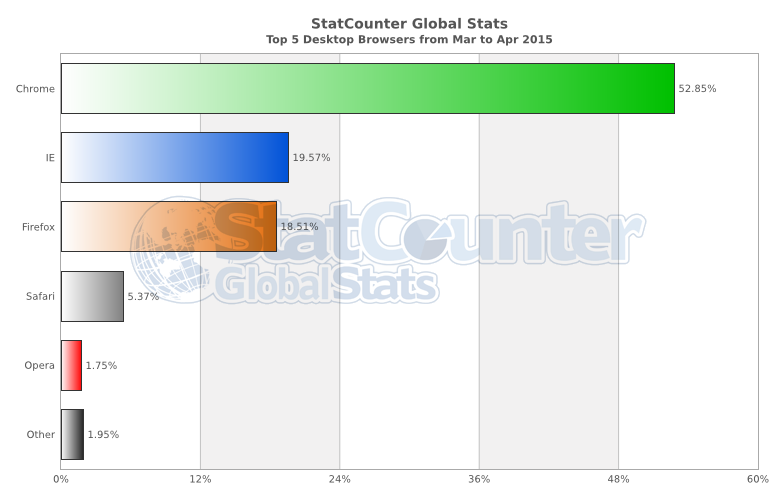
jQuery je odľahčená JavaScriptová knižnica. Rieši veľa spoločných úloh, ktoré si vyžadujú mnoho riadkov kódu napísaných v JavaScripte a obaľuje ich do metód, ktoré môžu volať jediným riadkom kódu. To je pre programátora veľkou výhodou, vzhľadom na rýchlosť písania aj prehľadnosti kódu.

Knižnica jQuery obaľuje nasledovné funkcie:

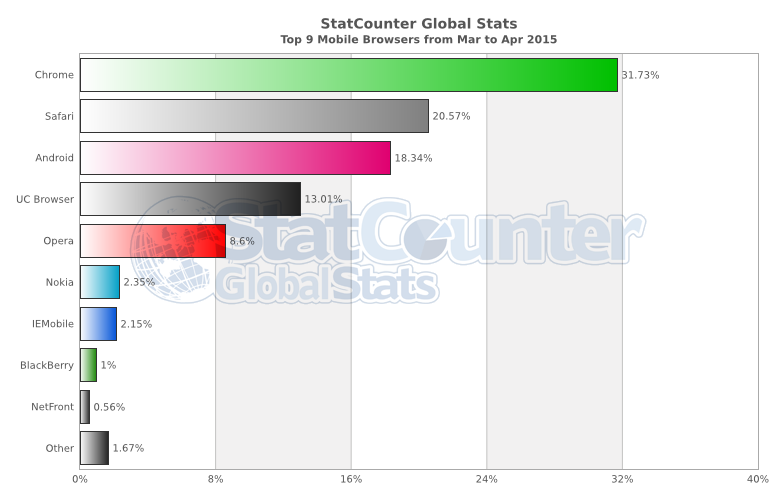
* HTML/DOM manipuláciu
* CSS manipuláciu
* HTML event metódy
* Efekty a animácie
* AJAX
* Utility

## Webový prehliadač

Webový prehliadač je program nainštalovaný v počítači alebo inom zariadení, ktorého hlavnou úlohou je zobrazenie webovej stránky na zobrazovacom zariadení. Webový prehliadač ma väčšinou GUI pre zobrazenie obsahu webovej stránky. Na počítačoch sú v súčasnosti najpoužívanejšie 4 webové prehliadače Google Chrome, Mozzila Firefox, Safari, Internet Explorer. Na mobilných zariadeniach prevažne Google Chrome a Safari optimalizované pre mobilné zariadenia. Na presnejšie rozdelenie desktopových a mobilných prehliadačov podľa používania v Marci 2015 slúži Obrázok 1 a Obrázok 2.



Obrázok 1 - Najpoužívanejšie prehlaidače pre počítače[[5]](#footnote-6)



Obrázok 2 - Najpoužívanejšie prehliadače pre mobilne zariadenia[[6]](#footnote-7)

### Podporné nástroje vykresľovania webových prehliadačov

Renderovacie jadro webového prehliadača je softvér, ktorý načíta zdrojový kód webovej stránky skladajúci sa zo štrukturálnych a obsahových informácií a štýlov spolu s obrázkami a zobrazí naformátovaný obsah do webového prehliadača.

Renderovacie jadro je najčastejšie používané ako súčasť webových prehliadačov alebo emailových klientov. Podla štatistík nástroja StatCounter je v marci 2015 používateľmi používané jadro WebKit, ktoré sú súčasťou webových prehliadačov Google Chrome a Safari.

V produkte Mozilla je použité renderovacie jadro Gecko. Pôvodne bolo vyvíjané firmou Netscape Communication Corporation, ale momentálne je vyvíjané firmou Mozilla Corporation.

Internet Explorer obsahuje renderovacie jadro Trident už od verzie 4.0 a je postupne vylepšované. Toto jadro je využívané v mnohých komponentoch operačného systému Windows.

Štandard CSS obsahuje vlastnosti na podporu jednotlivých renderovacích jadier. Používa špeciálne predpony pre dané renderovacie jadro, ktoré sú implementované v najpoužívanejších prehliadačoch. Ako príklad uvádzam vlastnosť CSS pre nastavenie veľkosti zaoblenia rohov na ohraničení elementu *element*.

.element{

-moz-border-radius: 2em;

-ms-border-radius: 2em;

-o-border-radius: 2em;

-webkit-border-radius: 2em;

border-radius: 2em;

}

### Viewport

Tag <meta name=viewport> dáva zmysel pri použití návrhu webu s responzívnym dizajnom, pretože prehliadače v mobilných zariadeniach vykresľujú web zmenšený, aby ho bolo celý vidieť a používateľ si musí sám priblížiť web do čitateľnej veľkosti.

Napríklad zariadenie môže mať skutočnú šírku displeja 360 pixlov, ale rozlíšenie displeja môže mať väčšie napríklad šírka 720 pixlov. Prehliadač roztiahne *okno zobrazenia* na 720 pixlov pre vykreslenie webovej stránky. S použitím meta tagu viewport môžeme definovať odlišné správanie okna zadávaním odlišných parametrov atribútu *content*. Môžeme nastaviť šírku okna zobrazenia v prehliadači hodnotou width=360. Alebo hodnotou width=device-width. Táto hodnota nastaví veľkosť okna rovnú skutočnej šírke zobrazovacieho zariadenia.

Ďalším atribútom, ktorý nám viewport ponúka je initial-scale=1, ktorý vyjadruje v akom pomere sa web zobrazí vzhľadom na skutočnú veľkosť obrazovky. Hodnota 1 definuje, že veľkosť okna zobrazenie a skutočná veľkosť obrazovky je v pomere 1:1. Menšie hodnoty znamenajú zmenšenie a vyššie hodnoty zväčšenie. Stránku nie je možné zmenšiť pod stanovenú hodnotu.

## Nástroje na vývoj webových stránok

Na vývoj webových stránok nie sú potrebné žiadne špeciálne rozhrania. Ako bolo spomenuté v kapitole 1.2 webová stránka je štandardný textový dokument zo špecifickou príponou. Na tvorbu a editáciu dokumentu postačí aj jednoduchý textový editor. Existujú však pokročilejšie textové editory, ktoré ponúkajú napríklad nastavenie zvýraznenia syntaxe podľa zvoleného typu dokumentu, automatické návrhy a dopĺňanie kódu, generovanie kódu.

### PSPad

http://www.pspad.com/

Editor PSPad je voľne šíriteľný univerzálny editor. Editor je užitočný pre všetkých, ktorí pracujú v operačnom systéme Windows, pracujú s obyčajným textom, tvoria webové stránky, programujú. PSPad ponúka široké spektrum funkcií, ako napríklad:

* Práca s projektami
* Práca s viacerými dokumentmi súčasne
* Porovnávanie textov s farebným zvýraznením rozdielov
* Zvýraznenie syntaxe

### Sublime Text

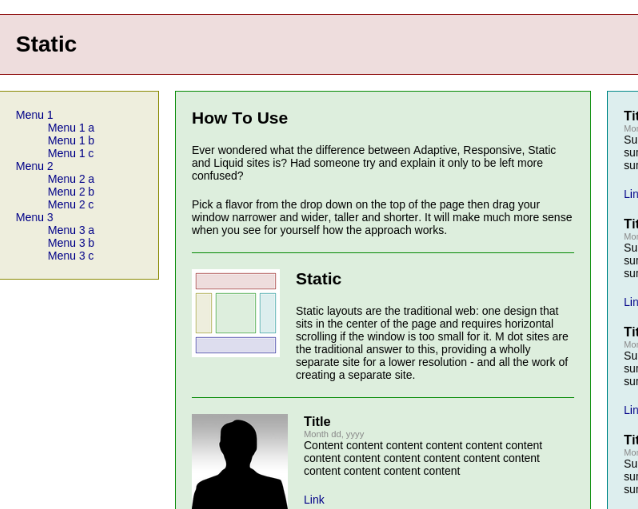
http://www.sublimetext.com/

Sublime Text je sofistikovaný textový editor pre písanie rozličného kódu. Obsahuje šikovné používateľské rozhranie, s pridanými funkciami a obrovskou výkonnosťou. Medzi popredné funkcie môžeme zaradiť:

* Reagovanie na udalosti pri používaní programu alebo manipulácií zo súborom v reálnom čase.
* Viacnásobný výber - umožňuje robiť zmeny v rovnakom čase na viacerých miestach v texte.
* Pridávanie rôznych pluginov.
* Nastavovanie editora pomocou zmeny preddefinovaných JSON objektov.
* Nastavenie zvýrazňovania syntaxe pre otvorený dokument.
* Automatické generovanie kódu alebo častí obsahu.
* Podpora konzoly.

## Webový dizajn

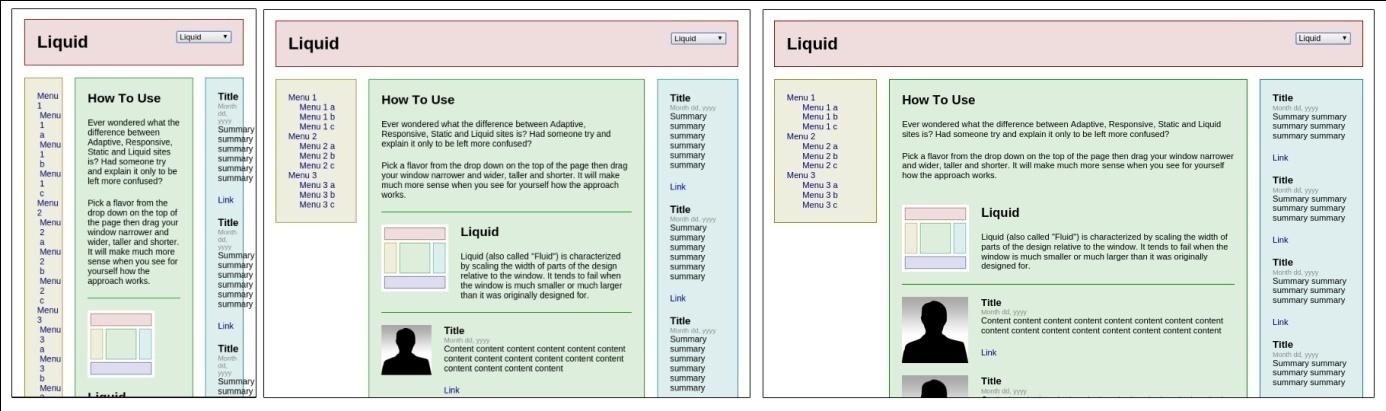
Webový dizajn (webdizajn) je funkčné riešenie a vzhľad webovej aplikácie. Pod pojmom funkčné myslíme, že jeho úlohou je zvládnuť zobrazenie a interpretáciu obsahu hlavne webovým prehliadačom. Webové stránky môžu byť z pohľadu návrhu dizajnu **statické** alebo **dynamicky reagujúce na zmenu šírky okna**. Statické návrhy dizajnu nijakým spôsobom nereagujú na zmenu šírky okna a ostávajú nezmenené, pozri Obrázok 3. Automaticky na webových prehliadačoch naskočí horizontálny scrollbar[[7]](#footnote-8). Dynamické pružne reagujú na veľkosť okna a snažia sa prispôsobiť zobrazenie šírky zariadenia. V súčasnosti sa najviac využíva návrh **responzívny** webdizajn, ktorý je obsahuje návrhy **liquid** webdizajn a **adaptive** webdizajn.



Obrázok – statický webdizajn[[8]](#footnote-9)

### Liquid

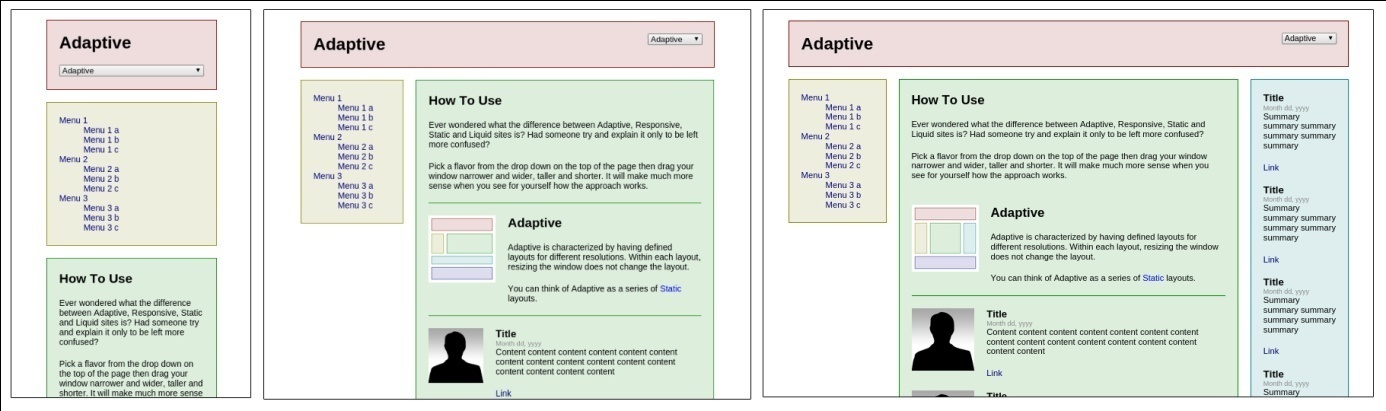
Liquid (premenlivý) alebo tiež nazývaný fluid (tečúci) je charakterizovaný naťahovaním častí webdizajnu proporcionálne šírke oknu webového prehliadača (Obrázok 4).



Obrázok – liquid webdizajn, zobrazenie pre mobil, tablet, počítač7

### Adaptive

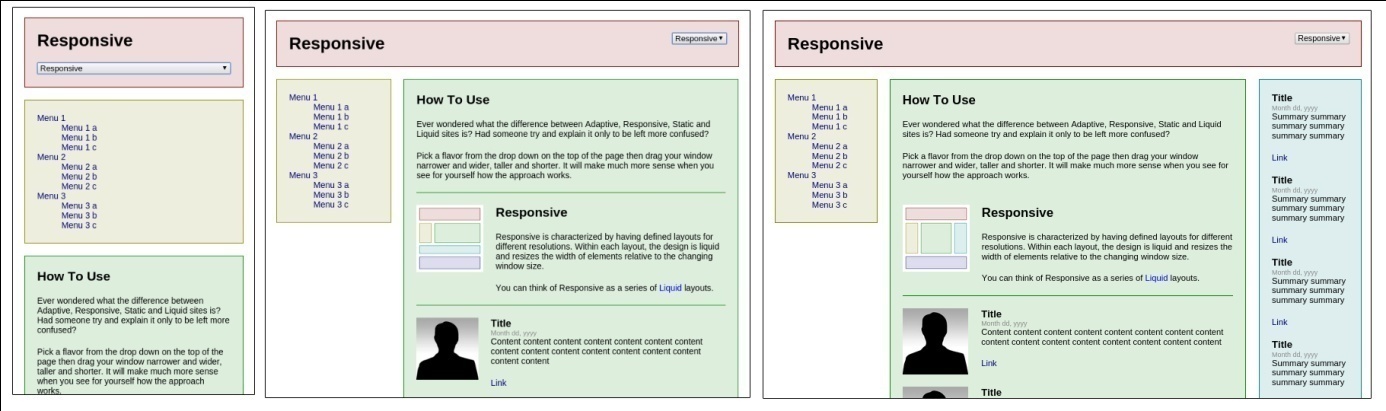
Adaptívny webdizajn má definované rozličné rozloženia (layouty) pre rozličné rozlíšenia. Môžeme si to predstaviť ako množinu statických rozložení. (Obrázok 5).



Obrázok – adaptívny webdizajn, zobrazenie pre mobil, tablet, počítač7

### Responsive

Responzívny webdizajn je v súčasnosti najpoužívanejší návrh webového dizajnu. Definuje rozličné rozloženia pre rozličné rozlíšenia a zároveň každý z týchto rozložení sa prispôsobuje šírke okna webového prehliadača. Responzívny webdizajn obsahuje základne vlastnosti liquid aj adaptive webdizajnu (Obrázok 6).



Obrázok – responzívny webdizajn, zobrazenie pre mobil, tablet, počítač7

# ZBER POŽIADAVIEK PRE KNIŽNICU

Nástupom smartfónov, tabletov a celkovo mobilných zariadení, začal sa prudko zvyšovať počet používateľov, ktorí cez ne pristupujú k webu. Tieto zariadenia majú rádovo nižšiu veľkosť displeja ako desktopové počítače.

Rovnako je dôležitá správna kvalifikácia zariadenia, či je to smartfón, tablet alebo desktopový počítač vzhľadom k tomu, že smartfón môže mať rovnaké rozlíšenie ako desktopový počítač. Treba brať ohľad na to, že používateľ na smartfóne pracuje s dotykovou obrazovkou. Preto by mali byť mala byť knižnica „mobile-friendly“.

Knižnica by mala zabezpečiť správne zobrazenie minimálne 95% používateľom a generovať kód podľa štandardov W3C, ktorá zabezpečuje správne zobrazenie webovej stránky.

Vzhľadom na to, že knižnica bude tvoriť časť webovej stránky, nie je rýchlosť načítania a zobrazenia webovej stránky závislá od výkonu počítača na ktorom beží prehliadač. Závislá je od pripojenia k internetu a veľkosti stránky, ktorú musí stiahnúť do prehliadača.

## Existujúce riešenia

V súčastnosti existujú riešenia, ktoré dokážu zabezpečiť požiadavky pre výslednú webovú stránku. Najviac používaný je framework Bootstrap, za ním druhý v poradí framework Foundation.

### Bootstrap

### Foundation

# ANALÝZA POŽIADAVIEK PRE KNIŽNICU

Je potrebné aby bola stránka zobrazená správne na celú šírku okna bez ohľadu na veľkosť zariadenia. Ako najvýhodnejšia voľba, sa zdá byť zvolenie responzívneho dizajnu webu. Jeho výhody sú bližšie rozobraté v Kapitole 1.7.

Vývoj pri responzívnom dizajne komplikuje fakt, že ak chceme mať odlišné zobrazenie elementov na odlišnom rozlíšení, musí to programátor zabezpečiť a vytvoriť predlohu správania sa na každom rozlíšení.

Funkcie knižnice by mali správne pracovať na 4 najpoužívanejších prehliadačoch, vzhľadom na to, že tieto prehliadače pokrývajú 96,3% používateľov pristupujúcich na internet v mesiaci marec 2015 (Obrázok 1).

# IMPLEMENTÁCIA KNIŽNICE

Z analýzy požiadaviek vyplýva, že knižnicu budeme tvoriť pre responzívny dizajn. Uľahčíme vývoj webových stránok programátorom tým, že preddefinujeme správanie elementov pri zmene rozlíšenia. Toto správanie však má vývojár pod kontrolou, správnym použitím implementovaných tried v knižnici. Všetky procesy zabezpečujúce správanie sa daného elementu sú viazané na atribút class v html elementoch. Vývojár jednoducho pri písaní HTML dokumentu dopísaním kľúčových názvov tried aplikuje na daný element definovanú funkčnosť.

## Štruktúra frameworku

Hlavnou požiadavkou na implementáciu, je použitie modulárneho návrhu knižnice, vzhľadom na optimalizáciu rýchlosti načítania webovej stránky. Programátor webových stránok si môže sám zvoliť, ktoré vlastnosti od knižnice vyžaduje a tie následne pripojiť do frameworku. Týmto sa zníži veľkosť dát, ktoré treba načítať pre zobrazenie webovej stránky a stránka sa načíta rýchlejšie. Knižnica bude pozostávať z **jadra** a **prídavných pluginov**.

## Jadro frameworku

Jadro tvorí základ knižnice a celý kód jadra knižnice je napísaný len v CSS a CSS3. Obsahuje

* automatické resetovanie preddefinovaných štýlov z prehliadača,
* obaľovacie kontajnery,
* základné rozloženie pomocou stĺpcov s definovanými odsadeniami,
* základné rozloženie pomocou stĺpcov s nulovými odsadeniami,
* fixovanie definovaného rozloženia bez ohľadu na veľkosť šírky webového prehliadača,
* skrývajúce elementy pri určitých hraničných rozlíšeniach
* elementy, ktoré majú definovanú maximálnu šírku natiahnutia

### Preddefinované štýly

Jadro frameworku v sebe obsahuje základné CSS pravidlá na nastavenie spoločného základného nastavenia dokumentu. Týmto krokom predefinujeme CSS štýly, ktoré sú integrované v prehliadači. Každý prehliadač môže mať rôzne nastavené štýly pre základné HTML elementy.

Prvé pravidlo definuje veľkosť vnútorných a vonkajších okrajov každého elementu na hodnotu 0.

\* {

margin: 0;

padding: 0;

}

Pravidlo, ktoré nastaví aby element html mal nastavenú vlastnosť box-sizing na hodnotu border-box. Toto zaručí, že skutočná veľkosť elementu (boxu) sa bude skladať zo súčtov veľkosti elementu, veľkosti orámovania a veľkosti vnútorného okraju. Vzhľadom na to že element html je predkom skoro všetkých elementov môžeme nastaviť pravidlo, aby všetky elementy mali nastavenú vlastnosť box-sizing na hodnotu inherit, čo znamená zdedená.

html {

box-sizing: border-box;

}

\* {

box-sizing: inherit;

}

Všetkým netriedeným zoznamom vypneme zobrazovanie zarážiek jednotlivých položiek zoznamu, nastavením vlastnosti list-style-type na hodnotu none.

### Rozloženie

Každé základné rozloženie potrebuje mať nejaké elementy, ktoré slúžia na obalenie iných elementov. Pre takéto základne obalenie obsahu sme definovali triedu container. Element s touto triedou si udržiava veľkosť závislú od veľkosti obsahu, bez ohľadu či je obsah v toku napravo alebo naľavo vlastnosťou float.

### Skrývajúce elementy

Elementy, ktoré necheme zobrazovať pri určitých rozlíšeniach môžeme skryť pridaním triedy hidden-small, hidden-medium alebo hidden-large.

### Elementy s nastavenou maximálnou šírkou

## Plugin pre zobrazenie menu

## Plugin pre zobrazenie slideshow

# ZÁVER

# ZOZNAM REFERENCIÍ

1. **W3C.** The World Wide Web Consortium (W3C). [Online] 11. Februar 2004. http://www.w3.org/TR/ws-arch/.

ZOZNAM ILUSTRÁCIÍ

Obrázok 1 - Najpoužívanejšie prehlaidače pre počítače 10

Obrázok 2 - Najpoužáívanejšie prehlaidaše pre mobilne zariadenia 10

Obrázok 3 – statický webdizajn 14

Obrázok 4 – liquid webdizajn, zobrazenie pre mobil, tablet, počítač 15

Obrázok 5 – adaptívny webdizajn, zobrazenie pre mobil, tablet, počítač7 15

Obrázok 6 – responzívny webdizajn, zobrazenie pre mobil, tablet, počítač7 16

# PRÍLOHY

K práci prikladám CD médium, na ktorom je uložená vypracovávaná knižnica, spolu s príkladovým webom využívajúci túto knižnicu pre návrh dizajnu.

1. **Platforma** je softvérové vybavenie zariadenia. [↑](#footnote-ref-2)
2. **Framework** je knižnica, ktorá ponúka jednoduché API na riešenie špecifického problému. [↑](#footnote-ref-3)
3. **Slideshow** je funkčný prvok webu, ktorý slúži na prezeranie položiek v množine. [↑](#footnote-ref-4)
4. GUI je grafické používateľské rozhranie. [↑](#footnote-ref-5)
5. [online] dostupné na: http://gs.statcounter.com/#desktop-browser-ww-monthly-201503-201504-bar [↑](#footnote-ref-6)
6. [online] dostupné na: http://gs.statcounter.com/#mobile\_browser-ww-monthly-201503-201504-bar [↑](#footnote-ref-7)
7. **Scrollbar** slúži pre posun zobrazenia webovej stránky vzhľadom na okno prehliadača po horizontálne osi. [↑](#footnote-ref-8)
8. [online] Ukážka reálneho správania sa v prehliadači dostupné na: http://www.liquidapsive.com/ [↑](#footnote-ref-9)