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](#_Toc417745846)

[1 ANALÝZA SÚČASTNÉHO STAVU 2](#_Toc417745847)

[1.1 Webová služba 2](#_Toc417745848)

[1.2 Webová stránka a webová aplikácia 3](#_Toc417745849)

[1.3 W3C 4](#_Toc417745850)

[1.3.1 HTML 5](#_Toc417745851)

[1.3.2 XHTML 6](#_Toc417745852)

[1.3.3 HTML 5 6](#_Toc417745853)

[1.3.4 CSS 7](#_Toc417745854)

[1.3.5 CSS3 8](#_Toc417745855)

[1.4 JavaScript 9](#_Toc417745856)

[1.4.1 jQuery 10](#_Toc417745857)

[1.5 Webový prehliadač 10](#_Toc417745858)

[1.5.1 Podporné nástroje vykresľovania webových prehliadačov 12](#_Toc417745859)

[1.5.2 Viewport 13](#_Toc417745860)

[1.6 Nástroje na vývoj webových stránok 13](#_Toc417745861)

[1.6.1 PSPad 13](#_Toc417745862)

[1.6.2 Sublime Text 14](#_Toc417745863)

[1.7 Webový dizajn 14](#_Toc417745864)

[1.7.1 Liquid 15](#_Toc417745865)

[1.7.2 Adaptive 16](#_Toc417745866)

[1.7.3 Responsive 16](#_Toc417745867)

[2 ZBER POŽIADAVIEK PRE KNIŽNICU 17](#_Toc417745868)

[2.1 Existujúce riešenia 17](#_Toc417745869)

[2.1.1 Bootstrap 17](#_Toc417745870)

[2.1.2 Foundation 17](#_Toc417745871)

[3 ANALÝZA POŽIADAVIEK PRE KNIŽNICU 17](#_Toc417745872)

[4 IMPLEMENTÁCIA KNIŽNICE 18](#_Toc417745873)

[4.1 Štruktúra frameworku 18](#_Toc417745874)

[4.2 Jadro frameworku 18](#_Toc417745875)

[4.2.1 Preddefinované štýly 19](#_Toc417745876)

[4.2.2 Rozloženie 20](#_Toc417745877)

[4.2.3 Skrývajúce elementy 20](#_Toc417745878)

[4.2.4 Elementy s nastavenou maximálnou šírkou 20](#_Toc417745879)

[4.3 Plugin pre zobrazenie menu 20](#_Toc417745880)

[4.4 Plugin pre zobrazenie slideshow 20](#_Toc417745881)

[ZÁVER 20](#_Toc417745882)

[ZOZNAM REFERENCIÍ 21](#_Toc417745883)

[PRÍLOHY 23](#_Toc417745884)

ZOZNAM POUŽITÝCH SKRATIEK

HTTP. *Protokol*

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 je pospájaná z miliónov serverov, poskytujúcich svoje služby miliónom používateľom. Počet serverov a zariadení z ktorých je možné pristupovať na webové stránky 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 text, grafiku, zvuk a video.

Používatelia sa pripájajú na webové stránky 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.

Čo vytvára otázku ako by sme mali stránku správne formátovať a členiť aby sa správne zobrazila 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 displeja zariadenia. Knižnica bude vytvorená tak, aby správne zobrazovala stránky na 4 najpoužívanejších prehliadačoch.

# ANALÝZA SÚČASTNÉHO STAVU

Prenos a zobrazovanie webových stránok v internete je založená na architektúre klient-server. Webová stránka je uložená na webovom serveri, ku ktorému pristupujeme prostredníctvom HTTP protokolu v sieti internet. Webová stránka sa zvyčajne zobrazuje pomocou webového prehliadača (Obrázok 1). Webový prehliadač odošle požiadavku na webový server a odpoveďou na ňu sú dáta prezentujúci stránku samotnú. Tieto dáta webový prehliadač zvizualizuje do finálnej podoby zobrazenej stránky.



Obrázok 1 – zobrazenie stránky vo webovom prehliadači

## Webová služba

Webová služba je komunikačný interfejs, respektíve API, ktorému sa posiela požiadavka cez HTTP protokol. Poskytuje spôsob kooperácie medzi softvérovými aplikáciami,pri čom nezáleží aká technológia je použítá na klientskej alebo serverovej strane. Komunikácie pomocou webových služieb je realizovaná cez sieť, nevyžaduje pripojenie do internetu.S webovou službou zariadenia komunikujú spôsobom, ktorý je predpísaný v popise služby. Na popis API pre volanie procedúry a jej dát slúži protokol SOAP *(Simple Object Access Protocol)*. Existujú služby ako RPC, REST a iné, ktoré využívajú protokol SOAP. 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, určený pre službu WWW. Pôvodom určená pre zobrazenie textov, obrázkov, tabuliek, zoznamov a iných primitívnych prvkov. Dnešné web stránky obsahujú rôzne komplikovanejšie prvky. Môžu obsahovať napríklad navigačnú lištu pre navigovanie na iné stránky alebo časti webu, slideshow[[1]](#footnote-2) na zobrazenie obrázkov alebo fotografií, rôzne typografické prvky, nadpisy, logá, timeline. V skutočnosti nezáleží na tom, čo webová stránka obsahuje, pretože HTML definuje iba štruktúru dát.

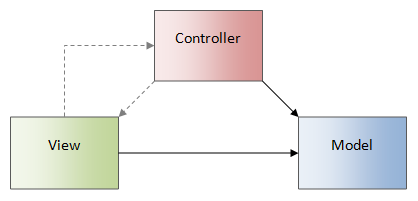
Pre odkazovanie na iné stránky sa používajú hypertextové odkazy, čím vytvárajú medzi jednotlivými stránkami prepojenie.

Webové stránky boli spočiatku písané manuálne ako dokumenty v jazyku HTML, takzvané statické stránky. Zobrazená stránka je výsledkom odpovede na HTTP požiadavku, ktorá môže byť generovaná rôznymi technológiami na strane serveru. Tým pádom je možné do stránky vložiť rôzny obsah z rôznych externých aj interných zdrojov, napríklad: dáta z relačnej databázy, iného serveru, atď. Keďže takéto stránky sú generované dynamicky, nazývame ich dynamické stránky.

Vlastnosti dynamických webových stránok sa dajú veľmi dobre využiť ako náhrada pre klasické *desktopové* aplikácie. Dáta a celá logika aplikácie je umiestnená na serveri a nesie názov **serverová** časť. Používateľské rozhranie (GUI[[2]](#footnote-3)) beží vo webovom prehliadači, takzvaná **klientská** časť, a ten sám o sebe logiku aplikácie nepozná a iba posúva vstupné dáta a akcie logike na serveri. Aplikácie implementované takýmto spôsobom nazývame webové aplikácie a najlepšie ich reprezentuje MVC architektúra.

Architektúra MVC delí aplikáciu na 3 logické časti tak, aby bolo možné ich upravovať samostatne a celkový vplyv zmien na ostatné časti bol čo najmenší. Vravíme o častiach *Model*, *View* a *Controller*. Model reprezentuje dáta a logiku aplikácie, View zobrazuje používateľské rozhranie a Controller je zodpovedný za riadenie udalostí v aplikácií. Pre ilustráciu návrhu slúži Obrázok 2.

Vo vývoji webových aplikácií sa používajú termíny frontend a backend pre oddelenie klientskej a serverovej časti .



Obrázok 2 – diagram MVC architektúry[[3]](#footnote-4)

## 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>. Meta informácie o dokumente, ako nadpis stránky, znaková sada použítá na kódovanie dokumentu a pripojené externé zdroje sa nachádzajú medzi párovími tagmi <head> a </head>. Zobraziteľná časť dokumentu sa nachádza medzi tagmi <body> a </body>.

Obsah ohraničený párovými tagmi rovnakého typu, spolu s týmito tagmi sa nazýva element. 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 syntaxe 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, ktorý definuje ako budú elementy zobrazené. Definuje zobrazenie 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 oddelenie formátovanie vzhľadu dokumentu od jeho štruktúry a dát.

Syntax CSS je založený na písaní pravidiel pre html elementy. Pravidlo je množina obsahujúca selector a blok deklarácie vlastností. Selector definuje HTML element, na ktorý chceme aplikovať daný štýl. Blok deklarácie vlastností, môže obsahovať viacero 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 zoskupiť.

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. Hlavnou výhodou použitia externého súboru je zníženie prenášaného objemu dát, keďže je pravdepodobné, že štýl sa nebude veľmi často meniť.

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. Toto štýlovanie poväčšine znamená prepisovanie aplikačného kódu. Cieľom je oddeliť štýl od kódu aplikácie, tak dizajnér nemusí zasahovať do kódu aplikácie, 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 štýly definované v atribúte, 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:

* Možná manipulácia s DOM
* Môže validovať vstupné dáta od používateľa

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

### jQuery

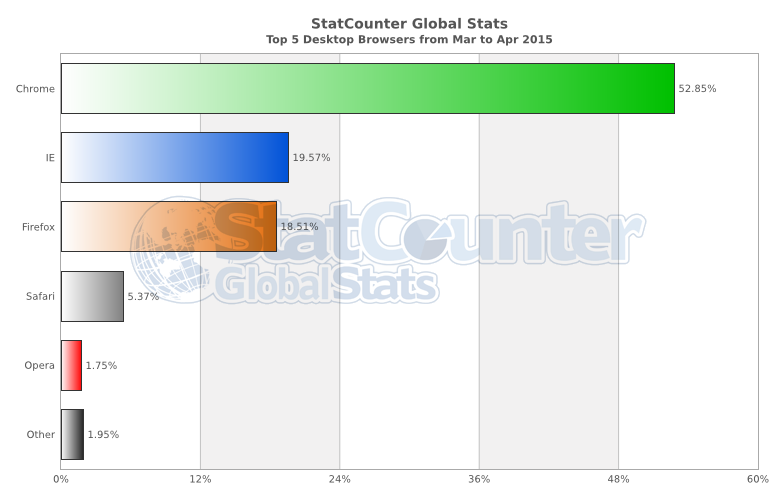
Je framework pre jazyk JavaScript. Rieši veľa spoločných úloh, ktoré si vyžadujú mnoho riadkov kódu napísaných v JavaScripte a obaľuje ich do metód, ktoré môžu volať jediným riadkom kódu. To je pre programátora veľkou výhodou, vzhľadom na rýchlosť písania aj prehľadnosti kódu.

Knižnica jQuery zjednodušuje nasledovné funkcie:

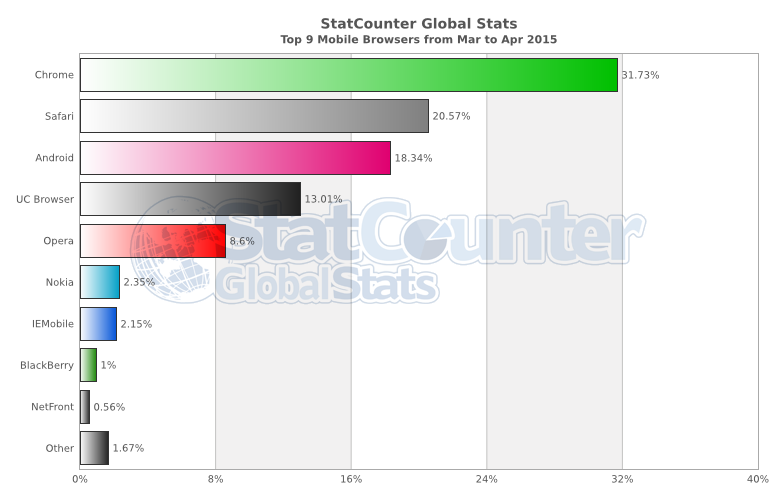
* HTML/DOM manipuláciu
* CSS manipuláciu
* HTML event metódy
* Efekty a animácie
* AJAX
* Utility

## Webový prehliadač

Webový prehliadač je program nainštalovaný v počítači alebo inom zariadení, ktorého hlavnou úlohou je zobrazenie webovej stránky na zobrazovacom zariadení. 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 3 a Obrázok 4.



Obrázok 3 – najpoužívanejšie prehliadače pre počítače[[4]](#footnote-5)



Obrázok 4 – najpoužívanejšie prehliadače pre mobilne zariadenia[[5]](#footnote-6)

### 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

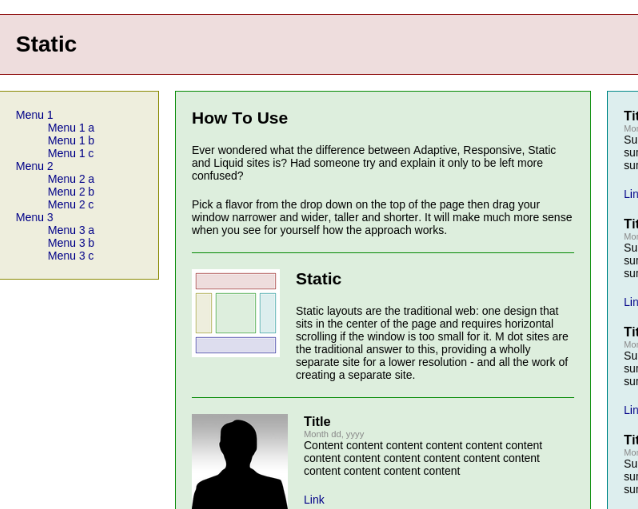
http://www.sublimetext.com/

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

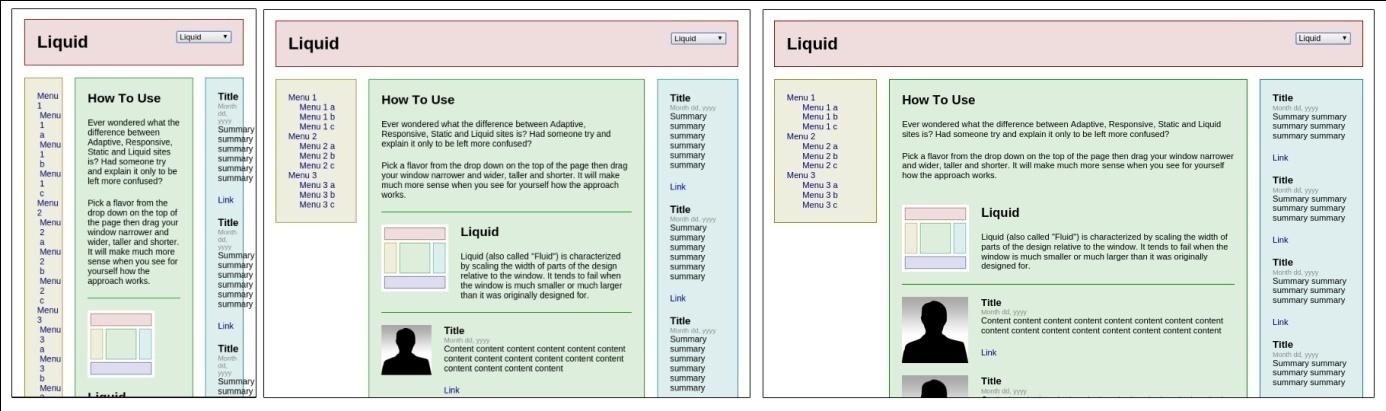
Webový dizajn (webdizajn) je funkčné riešenie a vzhľad webovej aplikácie. Pod pojmom funkčné myslíme zobrazenie a interpretáciu obsahu hlavne webovým prehliadačom. Webové stránky môžu byť z pohľadu návrhu dizajnu **statické** alebo **dynamicky reagujúce na zmenu šírky okna**. Statické návrhy dizajnu nijakým spôsobom nereagujú na zmenu šírky okna a ostávajú nezmenené, pozri Obrázok 5. Automaticky na webových prehliadačoch naskočí horizontálny scrollbar[[6]](#footnote-7). V súčasnosti sa najviac využíva návrh **responzívny** webdizajn, ktorý je obsahuje návrhy **liquid** webdizajn a **adaptive** webdizajn.



Obrázok 5 – statický webdizajn[[7]](#footnote-8)

### Liquid

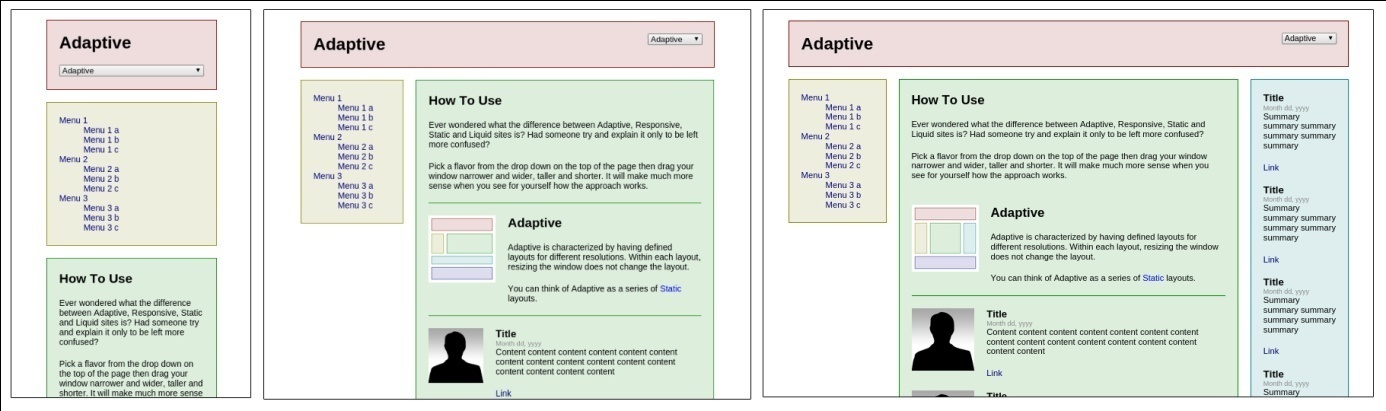
Liquid (premenlivý) alebo tiež nazývaný fluid (tečúci) je charakterizovaný naťahovaním častí webdizajnu proporcionálne šírke oknu webového prehliadača (Obrázok 6).



Obrázok 6 – liquid webdizajn, zobrazenie pre mobil, tablet, počítač7

### Adaptive

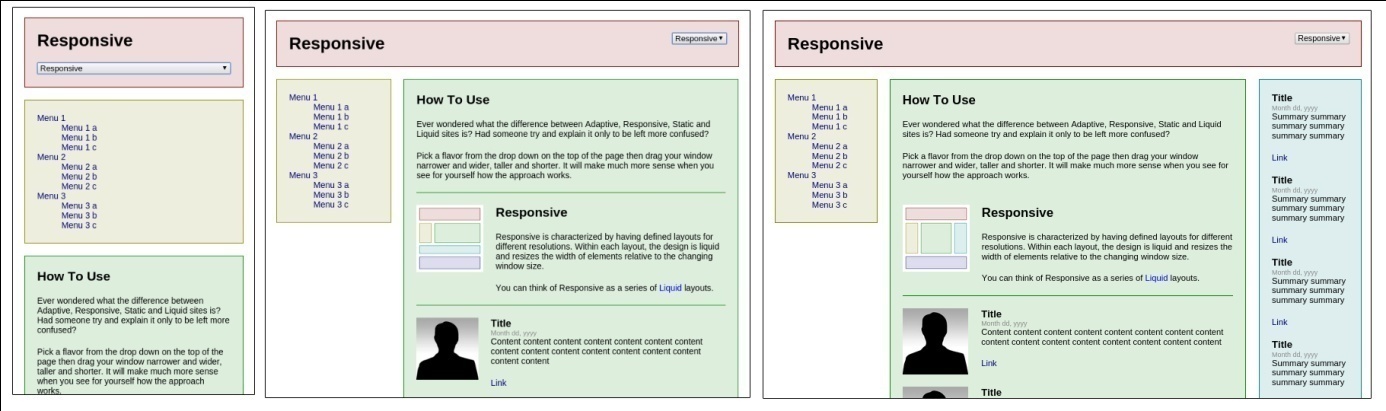
Adaptívny webdizajn má definované rozličné rozloženia (layouty) pre rozličné rozlíšenia. Môžeme si to predstaviť ako množinu statických rozložení. (Obrázok 7).



Obrázok 7 – 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 8). Cieľ je aby jeden dizajn bol zobraziteľný na všetkých zariadeniach a nebolo potrebné vytvárať osobitné zobrazenie webu pre každé zariadenie.



Obrázok 8 – 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 3).

# 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 – zobrazenie stránky vo webovom prehliadači 2

Obrázok 2 – diagram MVC architektúry 4

Obrázok 3 – najpoužívanejšie prehliadače pre počítače 11

Obrázok 4 – najpoužívanejšie prehliadače pre mobilne zariadenia 11

Obrázok 5 – statický webdizajn 15

Obrázok 6 – liquid webdizajn, zobrazenie pre mobil, tablet, počítač 15

Obrázok 7 – adaptívny webdizajn, zobrazenie pre mobil, tablet, počítač7 16

Obrázok 8 – 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. **Slideshow** je funkčný prvok webu, ktorý slúži na prezeranie položiek v množine. [↑](#footnote-ref-2)
2. GUI je grafické používateľské rozhranie. [↑](#footnote-ref-3)
3. [online] dostupné na: http://www.zdrojak.cz/wp-content/uploads/2009/05/02-MVC-zakladni-vztahy-124164090266884.png [↑](#footnote-ref-4)
4. [online] dostupné na: http://gs.statcounter.com/#desktop-browser-ww-monthly-201503-201504-bar [↑](#footnote-ref-5)
5. [online] dostupné na: http://gs.statcounter.com/#mobile\_browser-ww-monthly-201503-201504-bar [↑](#footnote-ref-6)
6. **Scrollbar** slúži pre posun zobrazenia webovej stránky vzhľadom na okno prehliadača po horizontálne osi. [↑](#footnote-ref-7)
7. [online] Ukážka reálneho správania sa v prehliadači dostupné na: http://www.liquidapsive.com/ [↑](#footnote-ref-8)