**Vysoká škola ekonomická v Praze**

**Fakulta informatiky a statistiky**

**Katedra informačního a znalostního inženýrství**

Studijní program: Aplikovaná informatika

Obor: Aplikovaná informatika

**CSS/JS knihovna pro zjednodušení vývoje grafického rozhraní webových stránek**

**BAKALÁŘSKÁ práce**

Student : Martin Škára

Vedoucí : Ing. et Ing. Stanislav Vojíř, Ph.D.

**2018**

Prohlášení:

Prohlašuji, že jsem bakalářskou práci zpracoval samostatně a že jsem uvedl všechny použité prameny a literaturu, ze které jsem čerpal.

V Praze dne TODO . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .   
 Martin Škára

Poděkování

TODO

Abstrakt

TODO

Klíčová slova

TODO

Abstract

TODO

Keywords

TODO

Obsah

TODO

Úvod

TODO

1. Existující knihovny pro řešení vizuální stránky webů

Vytvořit knihovnu pro zjednodušení řešení vizuální stránky webů (resp. celého kódingu) napadlo již bezpočet vývojářů. S rozmachem responzivního webdesignu se základ většiny těchto knihoven skládá z vytvoření responsivního grid systému, nejčastěji využívaných komponent na webových stránkách (např. menu, tlačítka, popup a další) a případně dalších pomocných tříd či nástrojů zjednodušujících samotný vývoj. Pro tvorbu takových knihoven jsou nejčastěji využívány jazyky CSS a JavaScript (a ve většině případů také preprocesory CSS a další javascriptové knihovny). Více o těchto technologiích lze nalézt v části 2.. Mezi jedny z nejznámějších knihoven patří Bootstrap a Foundation, které jsou popsány v následujících kapitolách. [1][2][3]

* 1. Bootstrap

Bootstrap byl původně vytvořen v roce 2010 vývojáři a designery ve společnosti Twitter, nicméně již od začátku byl vyvíjen jako open-source software. Z tohoto důvodu byl dříve nazýván také jako *Twitter Bootstrap*. Jeho první verze vyšla 19.8.2011. S druhou verzí byli přidány styly pro responzivitu, se třetí bylo k celému kódu přistupováno jako mobile-first. Aktuální (leden 2018) verze je již čtvrtá – na rozdíl od starších verzích nestojí na preprocesoru LESS, ale SASS a využívá flexbox (viz 2.1.3.).

Základ Bootstrapu tvoří responzivní grid systém, kterým lze obsah kontejneru stránky dělit do dvanácti sloupců. Kontejner obsahu stránky má danou maximální šířku na 1170 pixelů, případně lze využít též nastavení, aby zabíral 100 procent šířky okna. Mezi sloupci se defaultně udržuje mezera 30 pixelů. Všechny rozměry jsou ale deklarovány proměnnými, a tak lze celý Bootstrap snadno kustomizovat. Typický HTML kód, kde je použitý Bootstrap grid může pak vypadat nějak takto:

<div class="container">

<div class="row">

<div class="col-sm">

Obsah prvního sloupce

</div>

<div class="col-sm">

Obsah druhého sloupce

</div>

<div class="col-sm">

Obsah třetího sloupce

</div>

</div>

</div>

Další součástí jsou pak styly resetující některé defaultní chování prvků na stránce či třídy pro správné zobrazení takových prvků. Lze zde najít styly pro textové prvky, obrázky, tabulky apod. Ve většině případů mají již základně předpřipravený vzhled.

Největší částí Bootstrapu pak tvoří předpřipravené komponenty. Mezi ty patří následující:

* Upozornění
* Štítky
* Drobečková navigace
* Tlačítka a jejich skupiny
* Karty s obsahem
* Carousel
* Rozklikatelné záložky
* Dropdown menu
* Formulářové prvky a jejich skupiny
* Jumbotrony
* Seznamy
* Modální okna
* Navigace
* Stránkování
* Popover
* Progress bary
* Tooltipy

Některé z komponent využívají i JavaScript a je tak nutné i s Bootstrapem použít knihovnu jQuery (viz TODO) a v některých případech i další pluginy či knihovny třetích stran.

Poslední části Bootstrapu jsou pomocné CSS třídy. Ty obsahují vlastnosti týkající se zobrazení prvků, barev a textových vlastností, pozicování a rozměrů.

Vývojář si může zvolit, které části knihovny využije.[2]

* 1. Foundation

V roce 2008 vytvořila společnost ZURB, která se zabývá vývojem softwaru nástroj (kolekci kódu) pro zjednodušení své práce v jazycích HTML, CSS a JavaScript. Nazvala jej ZURB Style Guide. Tento nástroj byl dále rozvíjen a v roce 2010 byl přejmenován na Foundation. Jeho druhá verze vyšla v roce 2011 a od té doby je vyvíjen jako open-souce. Od třetí verze (2012) stojí na preprocesoru SASS. Čtvrtá verze přinesla, stejně jako Bootstrap 3 s sebou přinesla mobile-first koncept. Obecně lze říci, že obě knihovny, které v této práci zmiňuji, se vyvíjely podobným směrem a sledují tak většinou trendy – i když většinou s jistým zpožděním, zřejmě kvůli kompatibilitě napříč prohlížeči.

Na rozdíl od Bootstrapu 4, poslední verze Foundation, tj. 6.4.3, nenutí vývojáře používat grid systém postavený na flexboxu. Celkově si lze vybrat ze tří grid systémů, které Foundation nabízí.

Prvním z nich je *Float Grid* – obsah v rámci něj lze dělit až do dvanácti sloupců, jednotlivým sloupcům se dle nastavení určuje procentuální šířka a využívají CSS vlastností float. HTML kód postavený na takovém grid systému může vypadat takto:

<div class="row">

<div class="small-2 large-4 columns">

Obsah prvního sloupce

</div>

<div class="small-4 large-4 columns">

Obsah druhého sloupce

</div>

<div class="small-6 large-4 columns">

Obsah třetího sloupce

</div>

</div>

Pokud kód srovnáme s kódem Bootstrapu, tak až na názvy tříd není vidět velký rozdíl, až na nepotřebnost třídy container – ve Foundation šířku obsahu stránky obstarává rovnou třída row.

Druhým z grid systémů Foundation je *Flex Grid* – jeho zápis se od Float Gridu neliší, nicméně v pozadí již nevyužívá CSS vlastnosti float a procentuální šířku sloupců, ale vlastnost CSS flex. Pro sloupec, který zabírá polovinu (50%) řádku, tak může tato vlastnost vypadat takto:

flex: 0 0 50%;

Posledním a zároveň nejnovějším grid systémem ve Foundation je *XY Grid*. Ten, stejně jako Flex Grid využívá také flexbox, nicméně nabízí daleko více možností včetně vertikálního grid systému. Ten umožňuje manipulovat s obsahem nejen v rámci sloupců, ale i v rámci řádků. Proto u většiny tříd tohoto grid systému najdeme v názvu za pomlčkou zmíněno, v jaké ose právě pracujeme (x pro horizontální a y pro vertikální grid). Podobně jako u Bootstrapu se zde vyskytuje i třída grid-container, která určuje šířku obsahu stránky (defaultně nastavená na 1200px). Ukázka horizontálního a vertikálního gridu:

<!—- horizontální grid -->

<div class="grid-container">

<div class="grid-x grid-margin-x">

<div class="cell small-4">

Obsah prvního sloupce

</div>

<div class="cell small-4">

Obsah druhého sloupce

</div>

<div class="cell small-4">

Obsah třetího sloupce

</div>

</div>

</div>

<!—- vertikální grid -->

<div class="grid-y" style="height: 500px;">

<div class="cell small-6 medium-8 large-2">

Obsah prvního řádku

</div>

<div class="cell small-6 medium-4 large-10">

Obsah druhého řádku

</div>

</div>

Lze zmínit, že u vertikálního gridu není důležitá třída grid-container, ale je důležitá výška obsahu.

Stejně jako Bootstrap je obsahuje Foundation pomocné třídy týkající se především vlastností flexboxu, pozicování, zarovnávání a zobrazování prvků na stránce.

Zbytek obsahu knihovny, který však vývojář může či nemusí využít lze shrnout v tomto seznamu:

* Textové a resetující styly
* Styly tlačítek
* Slidery a přepínače
* Menu
* Stránkování
* Drobečková navigace
* Rozbalující se záložky
* Upozornění
* Karty
* Popup okna
* Styly pro tabulky
* Štítky
* Styly pro responzivní zobrazení iframe
* Progress bary

Některé z komponent využívají i JavaScript, konkrétně knihovnu jQuery a případné další pluginy třetích stran.

Pokud srovnáme Foundation a Bootstrap, lze říci, že nám dávají skoro stejné možnosti – kolem Bootstrapu se však vytvořila větší komunita, a o Foundation lze říci, že je daleko více kustomizovatelné. [2][3]

1. Použité technologie

V této kapitole budou popsány technologie, které budou použity při vytváření samotné knihovny. Mezi ně patří CSS a jeho preprocesor SASS, JavaScript a jeho knihovna jQuery a nástroje pro zkompilování knihovny – Gulp či Webpack.

* 1. CSS

CSS (Cascading Style Sheets) je jazyk, který ovlivňuje vlastnosti zobrazení prvků v souborech vytvořených pomocí dalších jazyků jako HTML, XHTML či XML. Jeho hlavním smyslem je oddělení vzhledových vlastností od obsahu jednotlivých prvků. Jeho současná verze je CSS3, starší verze nabízeli daleko méně možností a vlastností.

Již od první verze CSS byla zavedena syntaxe vybírání prvků pomocí selektorů a nastavování jejich hodnot. Nastavovat šly vlastnosti písma a textu, barvy textu i pozadí, rozměry elementů a jejich pozice apod. Zápis CSS může vypadat například takto:[6]

.text { /\*selektor třídy\*/

color: black;

font-size: 12px;

font-weight: bold;

}

* + 1. Historické přístupy ke tvorbě layoutu webových stránek

S vývojem CSS a dalšími verzemi bylo postupem času samozřejmě možné využívat nové selektory i vlastnosti. Cílem této kapitoly však není vyjmenování všech vlastností dle jednotlivých verzí, ale především naznačit, jak se díky vývoji CSS vyvinul i layout (rozvržení) stránky a tím pádem i přístup k jejich tvorbě.

S omezenými možnostmi starších verzí CSS a větší rozdílnosti prohlížečů nebyly možnosti, jak vytvořit sloupcový layout rozsáhlé. Zřejmě jedno z nejstarších řešení je *tvorba layoutu pomocí rámů (frames)*. To spočívalo v rozdělení stránky do několika rámů, ve kterých se zobrazuje obsah dalších HTML stránek. Například v jednom z rámců mohlo být menu stránky a ve druhém se načítal obsah jednotlivých sekcí. Příklad:

<frameset cols=“25%,\*“>

<frame src=“menu.html“>

<frame src=“sekce.html“>

</frameset>

Rámy však neměli tolik možností (i když jejich zanořováním mohly vznikat komplexnější layouty) a vznikaly kvůli nim i potíže s tiskem, přidávání a ukládání stránek atd. – to bylo zapříčiněno tím, že uživatel v rámci rámů načítal de facto více stránek. Dnes se již rámy nepoužívají, nejsou ani součástí HTML5.

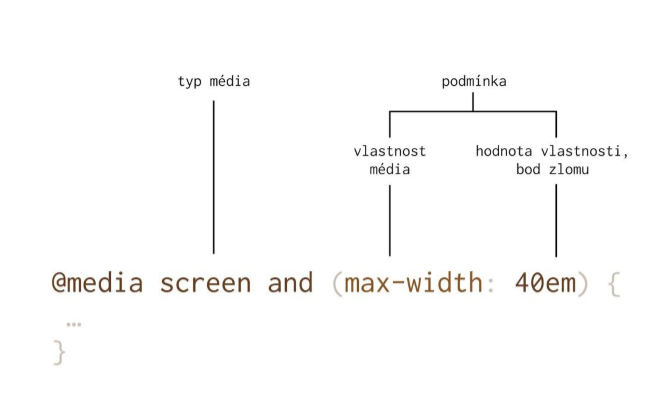
Další ze starších řešení je tzv. *tabulkový layout*. Ten spočíval v použití HTML tabulek, díky nimž lze obsah dělit a zarovnávat horizontálně i vertikálně. Tento layout se používal zejména proto, že v minulosti neexistovalo spolehlivé řešení, jak obsah dělit do flexibilních sloupců, zarovnávat vertikálně, přizpůsobovat jeho rozměry apod. Dnes tabulky jako nástroje pro zarovnání obsahu stále používají při vytváření e-mailových šablon – e-mailové klienty podporují daleko méně vlastností CSS než klasické webové prohlížeče. [7][9]

V dnešní době se pro tvorbu layoutu se využívají vlastnosti CSS, HTML slouží pouze k znázornění struktury obsahu. Postupem času se využívalo zejména proměnlivých rozměrů prvků, float a display inline-block, absolutního pozicování či display table. Většina těchto vlastností však nebyla defaultně určena pro tvorbu layoutu (např. float byl určen pro obtékání textu). V moderní tvorbě webových stránek se pro tvorbu layoutu používají zejména moduly CSS Flexbox a CSS Grid, pro tvorbu responzivního layoutu jsou pak také důležitá pravidla Media Queries. Více v následujících kapitolách. [11]

* + 1. Media Queries

Specifikace Media Queries udává v rámci jazyka podmínky, které umožní aplikovat určité CSS vlastnosti v určitých situacích. Dnes slouží zejména pro definování CSS dle velikosti okna prohlížeče a orientace zařízení. Jsou jedním z nejdůležitějších nástrojů pro tvorbu responzivního layotu. Celkově lze nimi ale definovat i styly pro monitory s určitým rozlišením a poměrem, styly pro Retina displeje, pro tisk a pro prohlížeče se specifickými vlastnostmi.

Typickou zjednodušenou strukturu Media Query lze vidět na obrázku níže. [10]

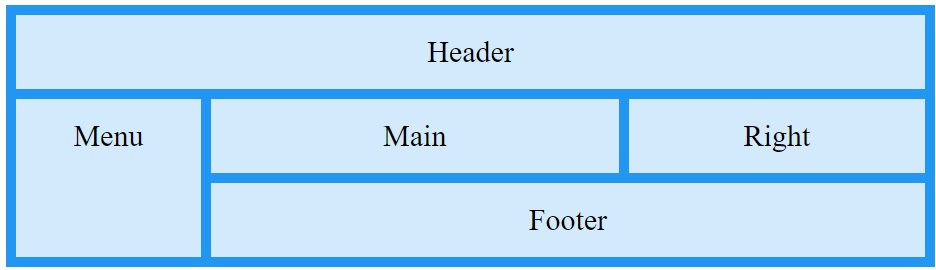
* + 1. Flexible Box Layout

Obrázek 1 - Zjednodušená struktura Media Query zápisu¨[10]

Flexible Box Layout (také flexbox) určuje v rámci kontejneru (tj. prvku obsahující dalších prvky) jak se budou chovat elementy v rámci něj. V defaultním nastavení manipuluje s prvky tak, že se snaží přizpůsobit jejich vzájemné rozměry jejich potřebnému obsahu. Prvky jsou tak flexibilní. Jednotlivým prvkům lze nastavit jako část rodičovského prvku mají zabírat (například půlku, či 2x větší část než ostatní prvky). Tím lze de facto vytvořit grid systém. Krom toho ale flexbox nabízí možnost takto s prvky pracovat nejen horizontálně, ale i vertikálně, měnit jejich pořadí nezávisle na HTML kódu (pořadí tak určuje CSS), zarovnávat elementy v obou směrech apod. [1][11]

* + 1. CSS Grid

CSS Grid je jeden z nejnovějších a nejmodernějších přístupů k tvorbě layoutu webových stránek. Na rozdíl od flexboxu se jedná o nástroj, kterým lze vytvořit kompletní layout stránky – tj. nestará se pouze o rozměry a pozicování prvků. Pro vysvětlení si představme layout stránky ukázaný dále na obrázku:



Obrázek 2 - Návrh běžného layoutu webové stránky[12]

HTML kód takto navrženého layoutu pomocí CSS Grid bude pak vypadat takto:[12]

<div class="grid-container">

<div class="item1">Header</div>

<div class="item2">Menu</div>

<div class="item3">Main</div>

<div class="item4">Right</div>

<div class="item5">Footer</div>

</div>

CSS:

.item1 { grid-area: header; }

.item2 { grid-area: menu; }

.item3 { grid-area: main; }

.item4 { grid-area: right; }

.item5 { grid-area: footer; }

.grid-container {

display: grid;

grid-template-areas:

'header header header header header header'

'menu main main main right right'

'menu footer footer footer footer footer';

grid-gap: 10px;

}

Jak jde vidět ze zápisu, tvorba layoutu pomocí CSS Gridu může být velmi jednoduchá. I přesto se však tento postup autor práce nakonec nerozhodl použít, protože ještě dnes (únor 2018) může způsobovat problémy na některých webových prohlížečů a vzhledem ke struktuře knihovny (viz. kapitola 3.), která se skládá zejména z jednotlivých komponent považuje za vhodnější použít flexbox.

* 1. Preprocesory CSS

Preprocesory jsou jazyky postavené nad CSS, které je následně nutné zkompilovat právě do CSS souborů. Využívány jsou proto, že na rozdíl od jazyka CSS nabízejí více funkcí a zjednodušují díky tomu práci. Mezi nejznámější preprocesory patří LESS, SASS a Stylus. I přesto, že se jednotlivé preprocesory mohou lišit syntaxí a odlišností některých funkcí, lze říci, že téměř každý nabízí výsledné funkce:

* Ukládání hodnot do proměnných
* Zanořování selektorů
* Skládání více souborů
* Mixiny (skupiny vlastností s parametry, které lze následně používat v rámci celého kódu)
* Dědění
* Další funkce (práce s barvami, operátory atd.)

Lze namítnout, že některé z těchto funkcí má již implementováno i samotné CSS, a tak by nebylo nutné používat preprocesor (například proměnné, skládání souborů, práci s operátory v rámci calc funkcí), avšak s funkcemi preprocesorů lze daleko lépe pracovat a v některých případech (jako právě u již zmiňovaných proměnných v CSS, které například nefungují v prohlížeči Internet Explorer ad.) fungují na všech prohlížečích. [13][1]

Jelikož ve firmě Appio Digital, již je autor práce zaměstnancem a v rámci jejich některých projektů může být případně knihovna využita, se již od června 2017 používá ve vývoji preprocesor SASS (do té doby se používalo LESS), a protože i SASS patří k preprocesoru, ke kterému se uchyluje většina vývojářů (toho si lze například všimnout i u Bootstrapu, který přešel taktéž z LESS na SASS), rozhodl se i autor práce použít pro vývoj knihovny preprocesor SASS. [2]

* + 1. Syntactically awesome style sheets

Syntactically awesome style sheets (zkráceně a dále SASS) vzniklo již roku 2006. Navrženo bylo Hamptonem Lintornem Catlinem a vyvinuto Natalií Weizenbaum. Postupem času se přidali další vývojáři – SASS je totiž vyvíjeno jako open-source. Jeho součástí jsou všechny funkce jmenované v kapitole 2.2.. SASS se samozřejmě neustále vyvíjí, největší změnou a novinkou, kterou však lze zmínit je vyvinutí jeho syntaxe. Na počátku mělo SASS imperativní (tj. odpovídající více běžným programovacím jazykům) syntaxi. Ta je samotná dnes nazývaná jako SASS (stejně jako celý preprocesor). Stále ji lze používat, avšak primárně doporučovaná již není a většina moderních knihoven v ní již není dále psána. Syntaxe SASS vypadá takto:

.trida

color: #fff

width: 200px

height: 200px

S verzí 3 pak nabídlo SASS (preprocesor) i syntaxi zvanou SCSS. Pokud, jako ve výše uvedené ukázce zápisu nepoužijeme žádné funkce preprocesorů, vypadá pak jako normální CSS zápis, což je i jedna z jejích výhod.

.trida {

color: #fff;

width: 200px;

height: 200px;

}

V zásadě je ale jedno, jaká syntax je použita, jelikož se tím nijak nemění možnost funkcí. [13]

Pro tvorbu knihovny však autor použil syntaxi SCSS.

* 1. JavaScript

JavaScript je jednoduchý skriptovací jazyk, jehož syntaxe vychází z jazyků C a Java. Jeho kód se spouští v prohlížeči uživatele a umožňuje s webovou stránkou manipulovat a reagovat na akce uživatele. V rámci vytvořené knihovny je použit zejména kvůli manipulaci se třídami elementů a pro vytvoření dynamických komponent knihovny, např. carouselu. Pro zjednodušení je použita knihovna jQuery.

* + 1. jQuery

jQuery je javascriptová knihovna, která usnadňuje práci s HTML dokumentem. S její pomocí lze mnoha věcí, které by normálně psal vývojář v čistém JavaScriptu, dosáhnout daleko jednodušeji a přehledněji. [14]

* 1. Nástroje pro zkompilování knihovny

Jelikož knihovna má být k dispozici k použití i v rámci jednoho CSS a jednoho JS souboru (pokud vývojář použije některé z komponent využívající JavaScript/jQuery), je nutné celou knihovnu zkompilovat. Mimo utvoření a sestavení těchto souborů proběhne během kompilace ještě několik operací:

* Kompilace SASS (resp. SCSS) souborů do validního CSS
* K potřebným vlastnostem CSS se přidají potřebné prefixy pomocí pluginu Autoprefixer, což tak zlepší kompatibilitu napříč prohlížeči
* Výsledné CSS a JS soubory se minifikují

Kompilace knihovny by mohla být provedena mnoha způsoby, autor v rámci knihovny uvede příklady kompilace v nástrojích Gulp a Webpack -> potřebné nástroje a pluginy budou nainstalovány pomocí balíčkovacího systému Yarn.

* + 1. Yarn

Yarn je balíčkovací systém Node.js, který stejně jako známější a starší NPM čerpá balíčky a moduly z rozhraní dostupného na adrese [www.npmjs.org](http://www.npmjs.org). Balíčky se instalují pomocí příkazů v příkazovém řádků a jejich struktura je také zapsána v souboru package.json. Yarn autor použil namísto NPM zejména proto, že instalace balíčků pomocí něj je rychlejší (ale nabízí i další výhody). [15]

* + 1. Gulp

Gulp je nástroj napsaný v JavaScriptu sloužící pro automatizaci úloh. Je dostupný i jako balíček, takže lze nainstalovat pomocí systému Yarn. Úlohy, které chce vývojář automatizovat se zapisují di soubor gulpfile.js, samotný Gulp se následně spouští z příkazové řádky.

* + 1. Webpack

Webpack je balíčkovací systém, díky kterému lze pracovat převážně s JavaScriptem, ale i dalšími soubory (CSS, obrázky apod.). Mezi navzájem provázanými soubory zachovává vztah a díky možnostem nastavení a pluginům s nimi dokáže během procesu manipulovat .

Návrh knihovny

TODO

Používatelnost, struktura, plány

Realizace knihovny

TODO

realizace, kompilace, použití + dokumentace

Vytvoření modelových stránek pro demonstraci možností implementované knihovny

TODO

TODO seznamy obrázků, tabulek, příloh

Terminologický slovník

|  |  |
| --- | --- |
| Termín | Význam (zdroj) |
| Grid | Grid systémy můžete vnímat jako jakési vodítko při navrhování. Jedná se v zásadě o hrubou strukturu, podle které jsou do návrhu umisťovány a pozicovány jednotlivé prvky. Dávají návrhu řád a pomáhají docílit již zmíněné přehlednosti a čitelnosti, ať už díky tomu, že se díky nim snáze udržuje rovnoměrný spacing mezi prvky, anebo díky přesnějším a “hezčím” rozměrům, kterých se s grid systémem lépe dosahuje.[4] |
| HTML | HyperText Markup Language – značkovací jazyk využívaný pro tvorbu webových stránek. Udává webovým prohlížečům, jak mají daný obsah zobrazit – udává strukturu dokumentu. Lze s ním dále manipulovat pomocí JavaScriptu či prvkům udávat vzhled pomocí jazyka CSS. [6] |
| Mobile-first přístup | Mobile-first je metoda tvorby (či návrhu) webových stránek, kde se primárně stránka vytvoří pro mobilní zařízení a následně se upraví pro uživatele zařízení s větší obrazovkou a desktopů. Přesně opačný přístup je pak nazýván *desktop-first*. |
| Node.js | Serverový framework, který využívá Javascript na straně serveru, na rozdíl od původního použití na straně klienta. |
| Open-Source software | Open-source software je software jehož zdrojový kód mlže kdokoliv používat, upravovat či vylepšovat.[5] |
| Responzivita | Responzivní web je takový, který se správně zobrazuje a přizpůsobuje zařízením, které uživatel používá. |
| XHTML | Extensible HyperText Markup Language – jazyk vyvíjený jako předpokládaný nástupce HTML, je téměř identický ale vyžaduje více striktní zápis. [8] |
| XML | Extensible Markup Language – značkovací jazyk určený pro serializaci dat, výměnu dat mezi aplikacemi a publikování dokumentů. |

Seznam citací

1. Vzhůru do CSS3. Martin Michálek, 2015. ISBN 978-80-260-8440-2.
2. Bootstrap · The most popular HTML, CSS, and JS library in the world. [online]. Mark Otto, Jacob Thornton, and Bootstrap contributors [cit. 2018-01-26]. Dostupné z: <https://getbootstrap.com>
3. Foundation [online]. Campbell, California: ZURB [cit. 2018-01-26]. Dostupné z: <https://foundation.zurb.com/>
4. AMRAN, Adam. Úvod do grid systémů. In: Meebio [online]. Meebio, 01.11.2011 [cit. 2018-01-26]. Dostupné z: <http://blog.meebio.cz/clanek/158/uvod-do-grid-systemu/>
5. What is open source? In: *Opensource.com*[online]. Red Hat [cit. 2018-01-27]. Dostupné z: <https://opensource.com/resources/what-open-source>
6. SANDEEP, Panda, Tiffany B. BROWN a Kerry BUTTERS. *HTML5 Okamžitě*. Computer Press, 2015. ISBN 978-80-251-4355-1.
7. УСАЧЕВ, Максим a fantasai. Evolution of CSS Layout: 1990s to the Future. In: *Fantasai: home* [online]. 2012 [cit. 2018-01-31]. Dostupné z: <http://fantasai.inkedblade.net/weblog/2012/css-layout-evolution/>
8. SCHAFER, Steven M. *HTML, XHTML a CSS: bible [pro tvorbu WWW stránek] : 4. vydání*. Praha: Grada, 2009. Průvodce (Grada). ISBN 978-80-247-2850-6.
9. BŘÍZA, Petr. Tvorba layoutu webu – teoretický úvod. In: *Interval.cz* [online]. 2004 [cit. 2018-02-01]. Dostupné z: <https://www.interval.cz/clanky/tvorba-layoutu-webu-teoreticky-uvod/>
10. MICHÁLEK, Martin. *Vzhůru do (responzivního) webdesignu*. Verze 1.1. Praha: vlastním nákladem autora, 2017. ISBN 978-80-88253-00-6.
11. ANDREW, Rachel. *CSS3 Layout Modules* [online]. 2nd edition. United Kingdom: edgeofmyseat.com, 2014 [cit. 2018-02-07].
12. CSS Grid Layout. *W3schools.com* [online]. [cit. 2018-02-07]. Dostupné z: <https://www.w3schools.com/css/css_grid.asp>
13. *Sass: Syntactically Awesome Style Sheets* [online]. Hampton Catlin, Natalie Weizenbaum, Chris Eppstein, and numerous contributors, 2017 [cit. 2018-02-07]. Dostupné z: <https://sass-lang.com/>
14. *JQuery* [online]. The jQuery Foundation, 2018 [cit. 2018-02-08]. Dostupné z: <http://jquery.com/>
15. *Yarn* [online]. [cit. 2018-02-09]. Dostupné z: <https://yarnpkg.com>

TODO přílohy?