

**Vysoká škola ekonomická v Praze**

**Fakulta informatiky a statistiky**

**Katedra informačních technologií**

Studijní program: Aplikovaná informatika

Obor: Informační systémy a technologie

# **Responzivní webdesign**

## **DIPLOMOVÁ PRÁCE**

Student: Bc. Jakub Hnízdil

Vedoucí: Ing. Jarmila Pavlíčková, Ph.D.

Oponent: Ing. Jiří Rosík

2015

## **Prohlášení**

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

V Praze dne 27. dubna 2015

.....  
Bc. Jakub Hnízdil

## **Poděkování**

Rád bych poděkoval vedoucí mé diplomové práce Ing. Jarmile Pavlíčkové Ph.D. za mnoho cenných rad, které jsem od ní získal při příjemných setkáních během konzultací této práce. Dále bych rád poděkoval své rodině za trpělivost v těch několika měsících psaní diplomové práce a za důležitou korekturu celého textu.

## **Abstrakt**

Tato diplomová práce se zaměřuje na responzivní webdesign. Jejím hlavním cílem je poskytnout ucelený pohled na tvorbu webových stránek právě v tzv. responzivním provedení a tím jejím čtenářům předat povědomí nejen o samotném pojmu responzivní webdesign, ale také i o jiných způsobech vytváření mobilních webů.

Dále se tato práce zabývá sestavením přehledu a porovnání vybraných technologií a frameworků s podporou tvorby responzivních webových stránek.

Každé skupině technologií či postupů je věnována samostatná kapitola tak, aby se čtenář mohl jednoduše a přehledně vracet k jednotlivým pojmům.

Po přečtení této práce by si měl čtenář odnést nejen podrobné znalosti v oblasti responzivního webdesignu, ale také rady a návody pro tvorbu webových stránek a pro rozhodování při volbě jednotlivých druhů technologií, které je možné při tvorbě webových stránek využívat.

## **Klíčová slova**

Web, webová stránka, mobilní web, responzivní webdesign, framework, media queries, mobile first, desktop first, viewport, CSS reset, CSS preprocessor, LESS, SASS, Bootstrap.

## **Abstract**

This Master thesis focuses on responsive web design. The main objective is to provide comprehensive view of the website created in so-called responsive realization, and impart awareness to readers not only about idea of responsive web design itself, but also about other ways of creating mobile webs.

Furthermore, thesis deals with constructing of overview and comparison of selected technologies and frameworks with support of creating responsive websites.

Every group of technologies or procedures is discussed in single chapter, to let reader simple and clearly return to particular concepts.

After reading this thesis the reader should acquire not only detailed knowledge about responsive web design, but also suggestions and manuals for creating websites and for decisions in choosing the type of technology, which is possible to use to creating websites.

## **Keywords**

Web, web page, mobile web, responsive webdesign, framework, flexible images, media queries, mobile first, desktop first, viewport, CSS reset, CSS preprocesor, LESS, SASS, Bootstrap.

# Obsah

1	Úvod.....	10
1.1	Cíle práce .....	10
1.2	Způsob dosažení cíle .....	10
1.3	Struktura práce .....	10
1.4	Očekávané přínosy.....	11
1.5	Rešerše .....	11
2	Responzivní webdesign .....	13
2.1	Co je to responzivní webdesign .....	13
2.2	Typy mobilních webů .....	14
2.2.1	Dva oddělené weby .....	14
2.2.2	Responzivní webdesign.....	16
2.2.3	Klasický desktop .....	17
2.2.4	Shrnutí .....	17
2.3	Statistické přehledy mobilních webů .....	17
2.3.1	Přehled využívání mobilních zařízení při přístupu na web .....	18
2.3.2	Přehled nejpoužívanějších mobilních prohlížečů .....	18
3	Přístupy k tvorbě (mobile first/desktop first).....	20
3.1	Desktop first.....	21
3.2	Mobile first.....	21
4	Rozložení layoutu.....	23
4.1	Grid .....	23
4.2	Fixní layout (fixed).....	24
4.3	Fluidní layout (fluid) .....	24
4.3.1	Plně fluidní layout .....	24
4.3.2	Fluid-fixed layout.....	25
4.3.3	Skokový layout.....	25

4.4	Ovládání pracovní plochy.....	25
5	Flexible images.....	27
5.1	HTML5 <PICTURE> .....	27
5.2	Fluidní obrázky .....	27
5.3	Flexibilní obrázky na pozadí .....	30
6	Media queries .....	32
7	Další technická řešení responzivního webdesignu .....	34
7.1	Způsoby identifikace mobilních zařízení .....	34
7.1.1	Na straně klienta .....	34
7.1.2	Na straně serveru .....	35
7.2	Viewport .....	35
7.3	CSS reset.....	36
7.4	Možnosti testování responzivního zobrazení .....	37
8	CSS preprocesory .....	40
8.1	Funkce CSS preprocesorů .....	40
8.1.1	Vnořené definice .....	40
8.1.2	Proměnné .....	41
8.1.3	Matematické výrazy.....	41
8.1.4	Mixin .....	42
8.1.5	Práce s barvami .....	42
8.1.6	Jmenné prostory .....	43
8.2	CSS preprocesory - typy.....	43
8.2.1	LESS.....	44
8.2.2	SASS.....	44
8.2.3	Stylus .....	44
8.2.4	Další CSS preprocesory .....	45
8.3	Základní porovnání syntaxe LESS a SASS (SCSS).....	45

8.3.1	Proměnné .....	45
8.3.2	Funkce.....	45
8.3.3	Mixins .....	45
8.3.4	Shrnutí .....	46
9	Definice srovnávacích kritérií .....	47
9.1	Přístup k tvorbě responzivního webdesignu .....	48
9.2	Typ layoutu .....	48
9.3	Podpora CSS preprocesorů.....	49
9.4	CSS reset.....	49
9.5	Kompatibilita s Internet Explorer .....	50
9.6	Druh licence.....	51
9.7	Nezařazená kritéria .....	52
10	Frameworky s podporou responzivního webdesignu .....	53
10.1	Bootstrap.....	53
10.2	Foundation .....	54
10.3	Gumby .....	55
10.4	Skeleton .....	55
10.5	Less Framework .....	56
11	Srovnání vybraných frameworků.....	58
11.1	Bootstrap.....	59
11.2	Less Framework .....	59
11.3	Foundation a Gumby.....	60
11.4	Skeleton .....	60
11.5	Souhrn .....	60
12	Ukázka práce s Bootstrap framework .....	61
12.1	Instalace.....	61
12.2	Ukázková šablona .....	61



12.3	Zalomování layoutu.....	62
12.4	Srovnání zobrazení.....	65
12.5	Předpřipravené styly .....	65
12.6	Souhrn .....	67
13	Závěr .....	68
	Terminologický slovník.....	69
	Seznam literatury .....	71
	Seznam obrázků .....	75
	Seznam ukázek zdrojového kódu .....	76
	Seznam tabulek.....	77
	Příloha 1 – Tabulka pro hodnocení frameworků .....	78

# **1 Úvod**

Autor si zvolil za téma své diplomové práce responzivní webdesign. Toto téma je mu blízké hned z několika důvodů. Autor se tvorbě webových stránek věnuje již více než 12 let, z toho poslední 4 roky je v tomto oboru zaměstnán. Webdesign a obzvláště responzivní webdesign je mu koníčkem, a rovněž zaměstnáním. Responzivní webdesign je dle autora zajímavé a důležité téma v oblasti tvorby webových stránek a to jak v současné době, tak také v kontextu s vývojem webů v budoucnosti.

## **1.1 Cíle práce**

Tato diplomová práce má za hlavní cíl poskytnout ucelený pohled na tvorbu webových stránek v tzv. responzivním provedení. Čtenář by si měl odnést povědomí jak o responzivním zobrazení webových stránek, tak i o jiných způsobech vytváření mobilních webů.

Dílčím cílem této práce je také sestavení přehledu a porovnání vybraných technologií a frameworků s podporou tvorby responzivních webových stránek a podrobné popsání frameworku, který v hodnocení získá nejvíce bodů a to včetně názorné ukázky jeho použití.

## **1.2 Způsob dosažení cíle**

Předání znalostí čtenářům bude dosaženo za pomoci popisu technik responzivního kódování a také samotné ukázky zdrojového kódu.

Vypracování diplomové práce předcházelo studium odporných publikací a článků s touto tematikou.

## **1.3 Struktura práce**

V kapitole 2 je popsán pojem responzivní webdesign včetně vyjmenování jeho tří základních pilířů. Dále tato kapitola představuje i další způsoby tvorby webových stránek a stručný statistický přehled několika ukazatelů, které jsou dle autora významné pro pochopení důležitosti mobilních webů.

Kapitola 3 popisuje dva základní přístupy k tvorbě responzivního webdesignu, mobile first a desktop first. Zde se již čtenář setkává s prvními ukázkami zdrojového kódu, o které jsou doplněny již všechny následující kapitoly teoretické části.

Každá z kapitol 4, 5 a 6 se věnuje jednomu z již zmíněných základních pilířů responzivního webdesignu, tedy layoutu, flexible images a media queries.

Kapitola 7 doplňuje předcházející kapitoly o ukázky zdrojových kódů dalších důležitých technologií, využívaných při tvorbě responzivních webů.

V 8. kapitole jsou představeny CSS preprocesory. Kapitola obsahuje ukázky zdrojových kódů jejich základní funkcí. Dále jsou představeny vybrané CSS preprocesory a dva nejpoužívanější CSS preprocesory jsou vzájemně porovnány.

Kapitolou 9 začíná praktická část této diplomové práce a to konkrétně definicí srovnávacích kritérií frameworků s podporou tvorby responzivních webových stránek. Autor zde uvádí jednotlivá kritéria, jejich slovní popis, bodovací stupnici a také váhy jednotlivých kritérií. Následující kapitola 10 obsahuje popis vybraných frameworků včetně jejich bodového ohodnocení dle stanovených kritérií. V kapitole 11 je poté znázorněna souhrnná tabulka s vlastnostmi jednotlivých frameworků a souhrnná tabulka jejich hodnocení, včetně slovního popisu výsledku celého srovnání.

V kapitole 12 je podrobně představen framework, který ve srovnání získal nejvíce bodů.

Závěrečná kapitola 13 shrnuje všechny předchozí kapitoly a konsoliduje naplnění stanovených cílů.

## **1.4 Očekávané přínosy**

Očekávaným přínosem této diplomové práce by mělo být přiblížení pojmu responzivní webdesign jejím čtenářům, poněvadž i když se jedná o poměrně rozšířený výraz a moderní trend v oblasti tvorby webových stránek, stále mnoho lidí dle autora netuší, co se pod tímto pojmem přesně skrývá.

Těm, kteří tento termín již dobře znají, může tato práce poskytnout návod pro samotnou tvorbu webových stránek, nebo také nápovědu pro rozhodování při volbě jednotlivých druhů technologií, které je možné při návrhu a implementaci webových stránek využívat.

## **1.5 Rešerše**

Při sestavování rešerší se autor setkal s několika bakalářskými a diplomovými pracemi na podobné téma, které byly sepsány a obhájeny na Vysoké škole ekonomické. Odborných a recenzovaných článků či publikací zabývajících se responzivním webdesignem však není mnoho. To je dle autora způsobeno tím, že se jedná o obor, kde vývojáři

programovacích jazyků poskytují pouze programové možnosti vývojářům webových stránek, ale jak s nimi každý naloží je jen a jen pouze na něm. V mnoha případech nelze nalézt jen jedno správné řešení, ale každý kodér může mít své vlastní, trochu jiné řešení. Proto se podobné informace v tomto oboru hledají spíše na fórech a diskusích vývojářů webových stránek. Tyto zdroje však není možné pro tuto práci použít.

Jako první rešeršní zdroj autor vybral diplomovou práci Ing. Petra Havelky s názvem Responzivní webdesign [1], obhájenou v roce 2014. Tento zdroj autor vybral, poněvadž se v některých bodech velmi přibližuje tématu této diplomové práce. Jedná se o nový zdroj s čerstvými informacemi, který má přehlednou strukturu a obsahuje značné množství praktických ukázek v podobě zdrojových kódů. Ing. Petr Havelka také porovnává vybrané frameworky s podporou responzivního webdesignu. Frameworky jsou však jen srovnávány. Jednotlivé vlastnosti sledovaných nástrojů nejsou nijak hodnoceny ani opatřeny vahami, jedná se tedy pouze o souhrnný přehled jejich jednotlivých funkcí. Diplomová práce je psána již pro pokročilé v oblasti kódování webdesignu a tak je i pro tuto práci vhodná. I přes drobné výhrady a neúplný terminologický slovník považuje autor tento zdroj za relevantní a prospěšný.

Hlavním rešeršním zdrojem byla pro autora kniha Ethana Marcotta s názvem Responsive web design [2]. Ethan Marcott je v kruzích webdesignu obecně uznávanou kapacitou a je mu i přisuzován „vynález“ slovního spojení responzivní webdesign. Marcott stanovil tři základní pilíře responzivního webdesignu, kterými se autor v této diplomové práci inspiroval a každému pilíři věnoval jednu samostatnou kapitolu. Tuto knihu autor považuje za ucelený souhrn myšlenek a technik responzivního webdesignu, neboť responzivní webdesign není pouze o zdrojovém kódu a technikách programování, ale také o pochopení a převzetí vize a pohledu na danou problematiku v širších souvislostech.

Dalšími, i když jen okrajovými, rešeršními zdroji byly autorovi bakalářské práce studentů Bc. Zdeňky Hatnianské [3], Bc. Davida Kolínka [4] a Bc. Milana Němce [5].

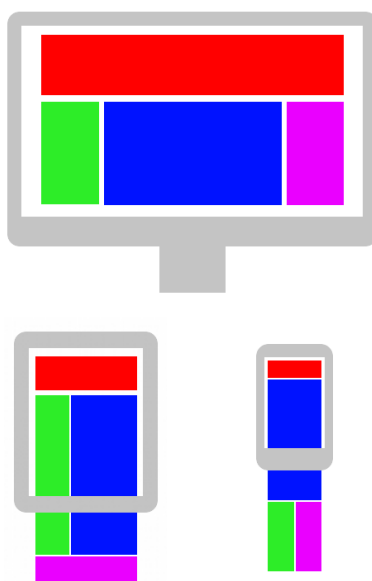
## 2 Responzivní webdesign

### 2.1 Co je to responzivní webdesign

Každá webová stránka poskytuje svým uživatelům (návštěvníkům webu) určitou formu grafického uživatelského rozhraní. Pomocí uživatelského rozhraní může návštěvník webu danou webovou aplikaci ovládat, tedy zadávat vstupy nebo získávat výstupy, případně kombinace obojího. Uživatel přistupuje na web z určitého typu zařízení. To jak se uživateli daná webová stránka vykreslí na jeho zařízení, se odvíjí od typu a rozlišení displeje tohoto zařízení.

Jak uvádí Martin Kelnar ve svém článku [6], samotné slovní spojení responzivní webdesign (angl. responsive webdesign) bylo poprvé použito ve stejnojmenném článku amerického programátora Ethana Marcotta v roce 2010. V tomto článku popisuje jeho autor i pojem responzivní architektura (angl. responsive architecture), který ho inspiroval právě k pojmenování pohyblivého a flexibilního zobrazení.

Responzivní webdesign zajišťuje, aby bylo grafické uživatelské rozhraní pro návštěvníka webu vždy automaticky přizpůsobeno jeho typu zařízení a rozlišení displeje, a to ať na web přistoupí s jakýmkoliv standardním typem zařízení, tedy desktop, tablet či mobilní telefon. Kompaktního zobrazení je dosaženo přeskupováním či zmenšováním jednotlivých prvků na webové stránce [7].



obr. 2-1 – Schéma responzivního webdesignu [Zdroj: wikipedia.org]

Tak jak uvádí Ethan Marcotte ve své knížce [2], stojí responzivní webdesign na třech základních pilířích:

- Flexibilní layout (kapitola 4)
- Flexibilní obrázky (kapitola 5)
- Media Queries (kapitola 6)

Všem těmto třem hlavním bodům responzivního webdesignu se autor věnuje v následujících kapitolách.

## **2.2 Typy mobilních webů**

Ještě před detailním popisem responzivních webů je dle autora potřeba zmínit i další přístupy k tvorbě webových stránek dostupných pro mobilní zařízení. Jedná se o tři základní typy webů [2]. Důvodů proč upřednostnit jeden před druhým je hned několik. Jedná se například o pořizovací cenu výsledného řešení, dobu, za kterou je možné webové stránky zhotovit, ale také jaké daná webová stránka potřebuje mobilní zobrazení. Všechny tři typy jsou popsány a porovnány autorem níže na základě poznatků z knihy [2] od Ethana Marcotta a autorových zkušeností z oblasti tvorby webových aplikací na zakázku.

### **2.2.1 Dva oddělené weby**

Prvním typem mobilního webu je samostatná webová stránka, nejčastěji označena subdoménou (např. m.alza.cz nebo mobil.idnes.cz). Jedná se o částečně oddělenou webovou stránku se společnou datovou základnou, která je plně přizpůsobena mobilním zařízením. Server detekuje typ zařízení (podrobněji v kapitole 7.1) a podle toho přesměruje uživatele na klasický web pro desktopová zařízení anebo na mobilní web. Většinou je takový web doplněn o možnost přepnutí z mobilního webu na klasický.

Výhodou tohoto přístupu k tvorbě webů je, že mobilní web je plně přizpůsoben mobilním zařízením, tedy i datový tok je snížen na minimum, aby se webová stránka načítala co nejrychleji. Dále odpadá řešení kompatibility mobilních zařízení například s některými přehrávači videí nebo technologií Flash.

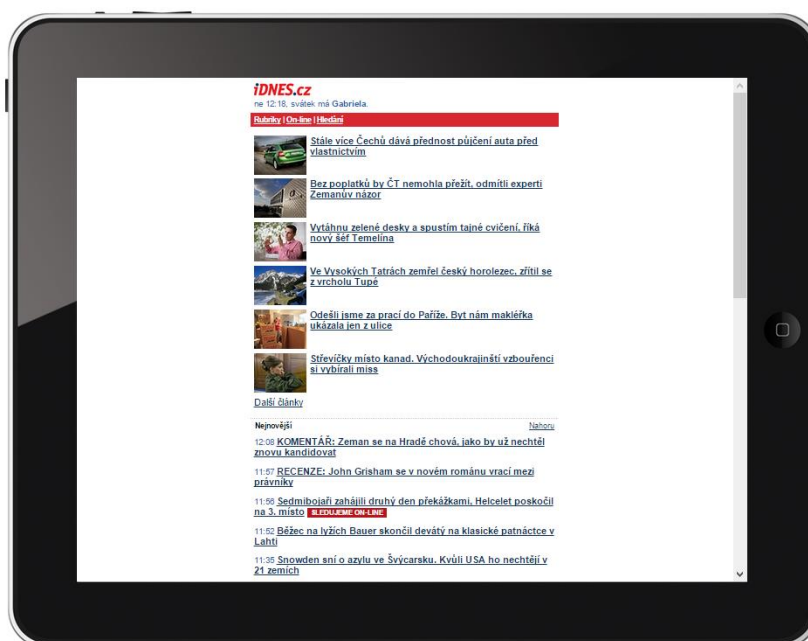
Nevýhodou je naopak pořizovací cena, která bývá až téměř dvojnásobná, poněvadž se jedná o tvorbu dvou webů. Další nevýhodou je, že pokud uživatel sdílí například na sociálních sítích odkaz přímo z mobilního zařízení, sdílí tak URL adresu mobilního webu, který se po kliknutí na odkaz zobrazí i uživatelům desktopu. Navíc při špatné detekci

mobilního zařízení může být uživatel tabletu i s velikostí displeje 10“ přesměrován na web pro mobilní telefony, který je pro takovýto displej nevhodný. Například na zpravodajském portálu [www.idnes.cz](http://www.idnes.cz) se tomu snaží předejít vytvořením třetího samostatného webu na subdoméně [t.idnes.cz](http://t.idnes.cz), který je přizpůsoben pro mobilní zařízení s větším displejem.

Obrázky obr. 2-2, obr. 2-3 a obr. 2-4 zobrazují jednotlivé tři webové stránky zpravodajského portálu [www.idnes.cz](http://www.idnes.cz). Jedná se o snímek klasického desktopového webu, webu pro mobilní zařízení s větším displejem a webu pro malá mobilní zařízení.



obr. 2-2 – [www.idnes.cz](http://www.idnes.cz) (náhled získán z [deviceponsive.com](http://deviceponsive.com))



obr. 2-3 – [t.idnes.cz](http://t.idnes.cz) (náhled získán z [deviceponsive.com](http://deviceponsive.com))



obr. 2-4 – mobil.idnes.cz (náhled získán z deviceponsive.com)

Výhodou i nevýhodou je pak dle autora i to, že web pro různá zařízení mívá často naprosto odlišné zobrazení stránky, tedy například rozdílné rozložení ovládacích prvků na webové stránce. Mobilní web je sice lépe přizpůsoben pro ovládání tahem prstu po displeji a poklepání na něj, ale uživatel klasického webu si rychle navykne, kde v desktopovém zobrazení nalezne menu, novinky, kalendář apod. Proto se uživatel v naprosto odlišném mobilním zobrazení velmi často ztrácí, poněvadž se v něm nedokáže rychle zorientovat a navíc některé prvky klasického webu ani na mobilní verzi nenalezne.

### 2.2.2 Responzivní webdesign

Responzivní webdesign je druhým typem mobilního zobrazení. Jemu je věnována celá kapitola 2, a proto zde autor zmíní pouze, tak jako u předchozího typu, klady a zápory ve smyslu porovnání jednotlivých typů mobilního zobrazení.

Výhodou responzivního zobrazení je, tak jako u předchozího typu, nižší náročnost na datový přenos, poněvadž je možné nepřenášet všechna data a v takové kvalitě jako pro klasické desktopové zobrazení. Uživateli responzivní zobrazení přináší přehledné a snadno ovladatelné prostředí, které se velmi podobá klasickému zobrazení, jen například místo tří článků vedle sebe uvidí uživatel tři články pod sebou. Mnohem snáze a hlavně levněji je možné vytvořit mobilní zobrazení pro různé typy mobilních zařízení nežli u předchozího typu, a to za pomoci media queries a několika řádků CSS.

Nevýhodou responzivního webdesignu je, že se obvykle není možné přepnout do klasického zobrazení pro desktop, i když je to v silách programovacího jazyka. Vývojáři webových stránek natolik věří svému mobilnímu zobrazení, že nepovažují za nutné tuto možnost uživateli poskytnout.



### **2.2.3 Klasický desktop**

Třetí a poslední typ mobilního zobrazení dle autora nelze považovat za mobilní, ačkoliv je mezi ně zařazen. Jedná se o webové stránky, které se nijak nepřizpůsobují velikosti displeje zařízení. Jejich obsah je totožný, ať na web uživatel přistoupí z jakéhokoliv zařízení. Design takového webu klade důraz na jednoduchost a použitelnost.

Nevýhodou je, že pokud je na webu například použita technologie Flash, pak se na mobilních zařízeních, která Flash nepodporují, Flash objekt nezobrazí. Někdy se jedná o zásadní prvky, které by uživateli s mobilním zařízením mohly znemožnit prohlížení webu (např. Flash menu). Další nevýhodou je, že mobilní zařízení, často připojené k pomalejšímu mobilnímu internetu s limitem stažených dat FUP, dostává stejné množství dat jako klasický desktop. Využívá-li uživatel mobilního zařízení webový prohlížeč, který sám dokáže snížit množství stažených dat, například načítám obrázků v nižší kvalitě, stává se tato nevýhoda podřadnou.

### **2.2.4 Shrnutí**

Autor se přiklání k názoru, že pokud si majitel webu zvolí jakýkoliv z výše uvedených typů mobilního zobrazení a programátoři odvedou kvalitní práci, je jedno, který z přístupů bude využit. Každý typ má svá pro a proti. Responzivní zobrazení je však moderní záležitostí a se stále se vyvíjejícím programovacím jazykem HTML5 a CSS3, společně s využíváním CSS preprocesorů a dalších nástrojů a technologií se stane dle autora budoucností webových stránek, neboť mobilní zařízení budou stále více vytlačovat desktopová zařízení a bude kladen větší důraz na přehledné, ale i zajímavé a originální zobrazení webové stránky.

Rozhodnutí o zvolení typu mobilního zobrazení může ulehčit například sledování procentního podílu návštěv webové stránky s mobilním zařízením.

## **2.3 Statistické přehledy mobilních webů**

Na závěr této kapitoly autor uvádí několik statistických údajů.

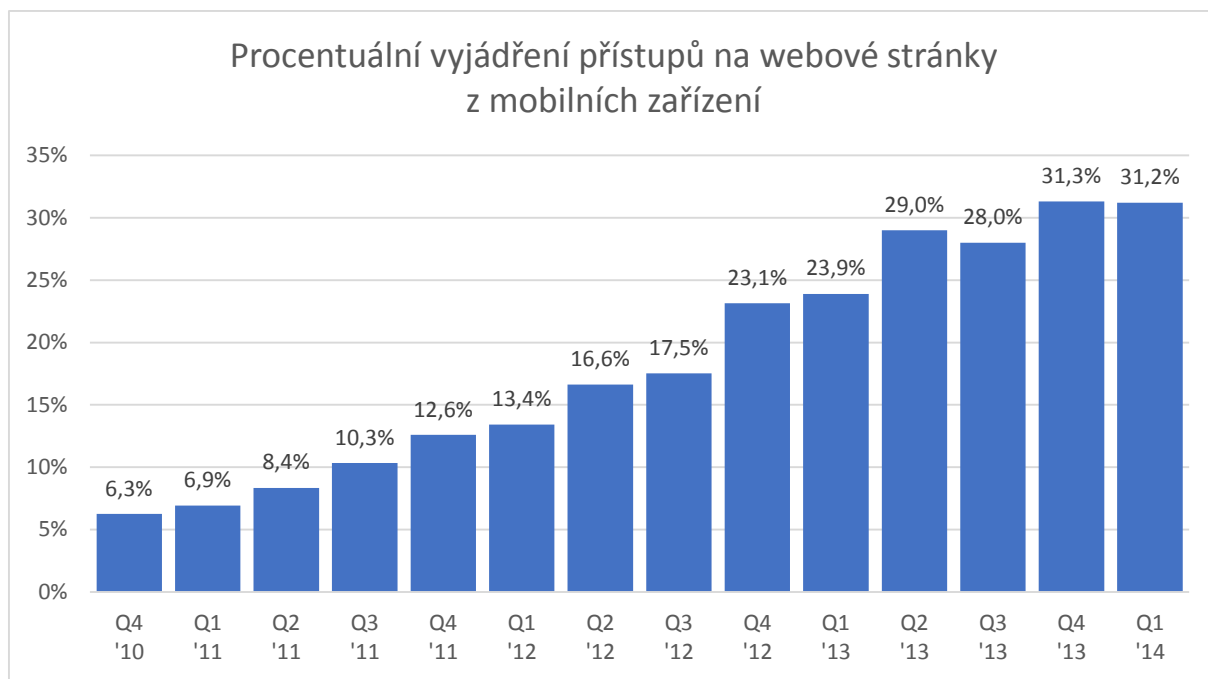
Dle autora je velmi důležité sledovat trendy, z jakých zařízení uživatelé přistupují na webové stránky a těmto trendům přizpůsobovat vzhled a „funkčnost“ webdesignu. Funkčností webdesignu autor míní schopnost designu se přizpůsobit koncovému zařízení návštěvníka webové stránky.

V následujících odstavcích jsou zmíněny konkrétní statistické přehledy, které je dle autora důležité sledovat.

### 2.3.1 Přehled využívání mobilních zařízení při přístupu na web

Následující graf na obr. 2-5 zobrazuje procentní poměr počtu přístupů na webové stránky z mobilních zařízení a to v období od čtvrtého čtvrtletí 2010 do prvního čtvrtletí 2014.

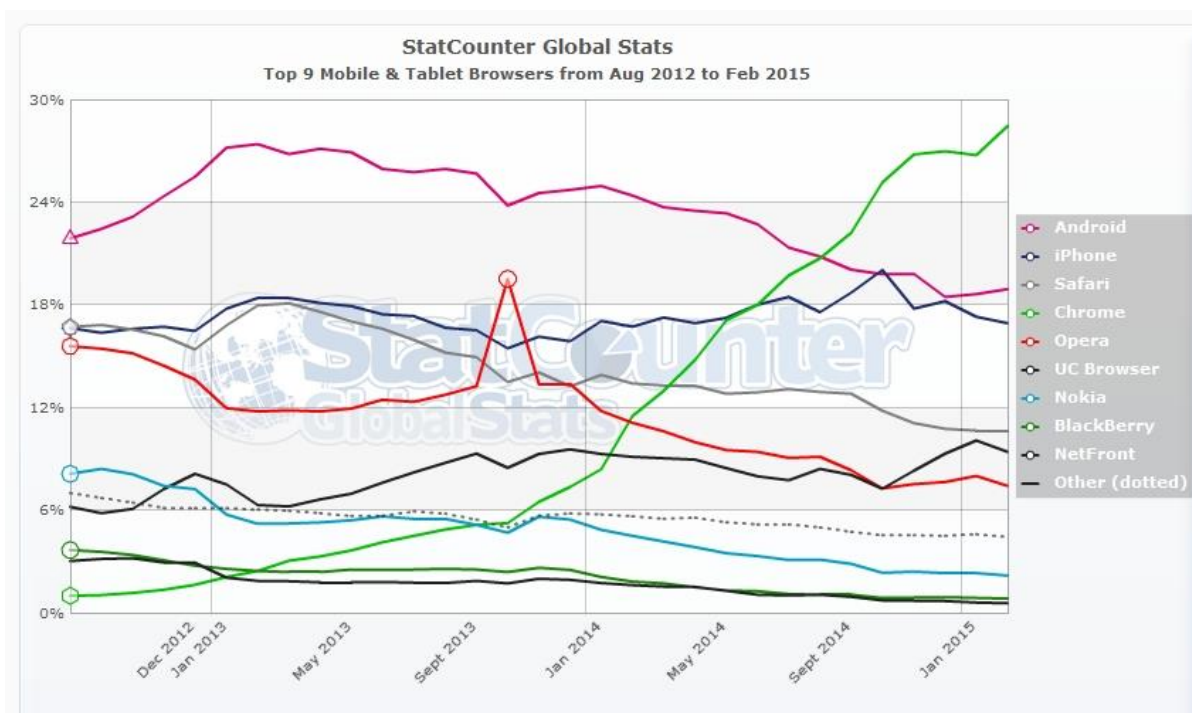
Graf znázorňuje, že trend ve využívání mobilních zařízení pro přístup na webové stránky až na velmi drobná kolísání stále roste. Dle autora je tento trend způsoben nejen rostoucí kvalitou a dostupností mobilních zařízení s přístupem na internet, ale také se na tomto vývoji významně podílí právě rozmach responzivního webdesignu, který od roku 2010 [7] uživatelům poskytuje přehledný přístup na web, který je z jejich zařízení snadno ovladatelný.



obr. 2-5 – Graf vývoje přístupu na web z mobilních zařízení [Statista 2015]

### 2.3.2 Přehled nepoužívanějších mobilních prohlížečů

Tak jako na klasickém počítači i při prohlížení webových stránek na mobilním zařízení je možné využít pro tento úkon určitý webový prohlížeč. A také jako u klasického počítače se každý webový prohlížeč „chová“ více či méně odlišně, tedy vykresluje danou webovou stránku dle svých možností a nastavení. Právě proto je dle autora taktéž důležité sledovat, které webové prohlížeče se využívají nejvíce a při optimalizaci se zaměřit na ty nejvyužívanější.



obr. 2-6 – Graf mobilních webových prohlížečů [Zdroj: gs.statcounter.com 2015]

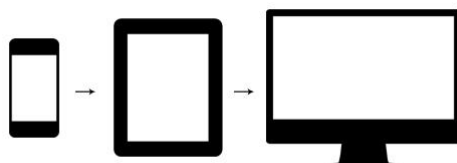
Z grafu na obr. 2-6 je jasné patrné, jak přichází do oblíbenosti webových prohlížečů Google Chrome a to hlavně od druhé poloviny roku 2013. Nyní je výrazně nejpoužívanějším mobilním webovým prohlížečem, který využívá téměř 30% všech uživatelů mobilního webu.

### 3 Přístupy k tvorbě (mobile first/desktop first)

Jak autor zmínil a odůvodnil v kapitole 2, jeho favorizovaným přístupem k mobilnímu webu je responzivní webdesign, proto se nadále bude věnovat pouze tomuto tématu.

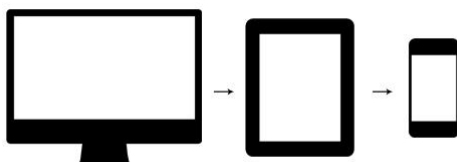
Při tvorbě mobilního webdesignu lze zaujmout jeden ze dvou základních přístupů. Mobile first nebo desktop first. Každý má své výhody, ale také své nevýhody. Podstatou jejich odlišnosti je pořadí zařízení, pro která se design webové stránky vyvíjí [8].

U mobile first jdeme tzv. cestou zdola, kdy první vytvořený design (CSS) webové stránky je optimalizován pro nejnižší (nejprimitivnější) zařízení s nejmenším displejem. Tím může být například osobní mobilní telefon, pro který je vytvořen čistě jednosloupcový layout. Poté se postupuje k sofistikovanějším zařízením jako je například smartphone, tablet, netbook nebo notebook až k osobním počítačům s monitory o vysokém rozlišení, přičemž původní jednosloupcový layout rozšiřujeme a obohacujeme o další grafické prvky a funkcionality [8].



*obr. 3-1 – Mobile first [Zdroj: designshack.net]*

Desktop first upřednostňuje opačný směr vývoje, tedy od velkých monitorů osobních počítačů až po malé displeje mobilních telefonů.



*obr. 3-2 – Desktop first [Zdroj: designshack.net]*

### 3.1 Desktop first

Programování s přístupem Desktop first tedy obnáší v první řadě vytvoření designu pro rozlišení osobních počítačů. Vytvořený CSS soubor se poté doplní o Media Queries podobně jako na ukázce kód 3-1.

```
/* CSS pro desktop */  
  
@media screen and (max-width: 480px) {  
  
    /* CSS pro mobilní zařízení */  
  
    @import 'mobile.css';  
}
```

*kód 3-1 – Media queries, desktop first*

Dochází zde tedy k „přebíjení“ původního CSS kódu pro desktopová zařízení restrikcemi pro mobilní zařízení. Sám o sobě tento způsob programování není příliš „čistý“, a takový kód se navíc velmi těžko udržuje, protože při každé změně CSS pro desktopová zařízení se musí vytvářet případné restriktce směrem dolů. Dále také takto „dodáváme“ CSS kód v plném rozsahu i starším prohlížečům, které leckteré nové prvky (například CSS3) nepodporují a nepodporují například ani samotné media queries. Mezi takovéto prohlížeče řadíme například Internet Explorer verze 8 a nižší [9].

### 3.2 Mobile first

Naopak u mobile first se nejprve vytvoří základní CSS soubor pro zvolené zařízení na nejnižší úrovni, který je dále rozšiřován pro větší rozlišení.

```
/* CSS pro mobilní zařízení  
a prohlížeče nepodporující Media Queries */  
  
@media screen and (min-width: 1024px) {  
  
    /* CSS pro zařízení s větším displejem */  
  
    @import 'mobile.css';  
}
```

*kód 3-2 – Media queries, mobile first*

Z předcházející podkapitoly vyplývá, že základní CSS soubor platí i pro staré webové prohlížeče, které nepodporují media queries a tak svůj design nerozšíří o kód pro větší rozlišení. Z toho plyne výhoda i nevýhoda. Výhodou je možnost použití nejmodernějších prvků CSS3, které by na starých prohlížečích nefungovali, protože si je ani nenačtou,

protože nepodporují media queries. Nevýhodou je naopak to, že existuje stále velké množství uživatelů, kteří používají webový prohlížeč Internet Explorer 8, který získá pouze základní vzhled z nejjednoduššího CSS, je proto potřeba zohlednit i tyto uživatele. Pro staré verze Internetu Exploreru lze snadno připravit speciální CSS a za pomoci podmíněných komentářů „vyladit“ vzhled stránek pro konkrétní verze, které jsou jinak velmi omezené co se jejich podpory CSS týče [8][10].

```
<!--[if lte IE 8]>  
Zdrojový kód určený pouze pro Internet Explorer verze 8 a nižší.  
<![endif]-->
```

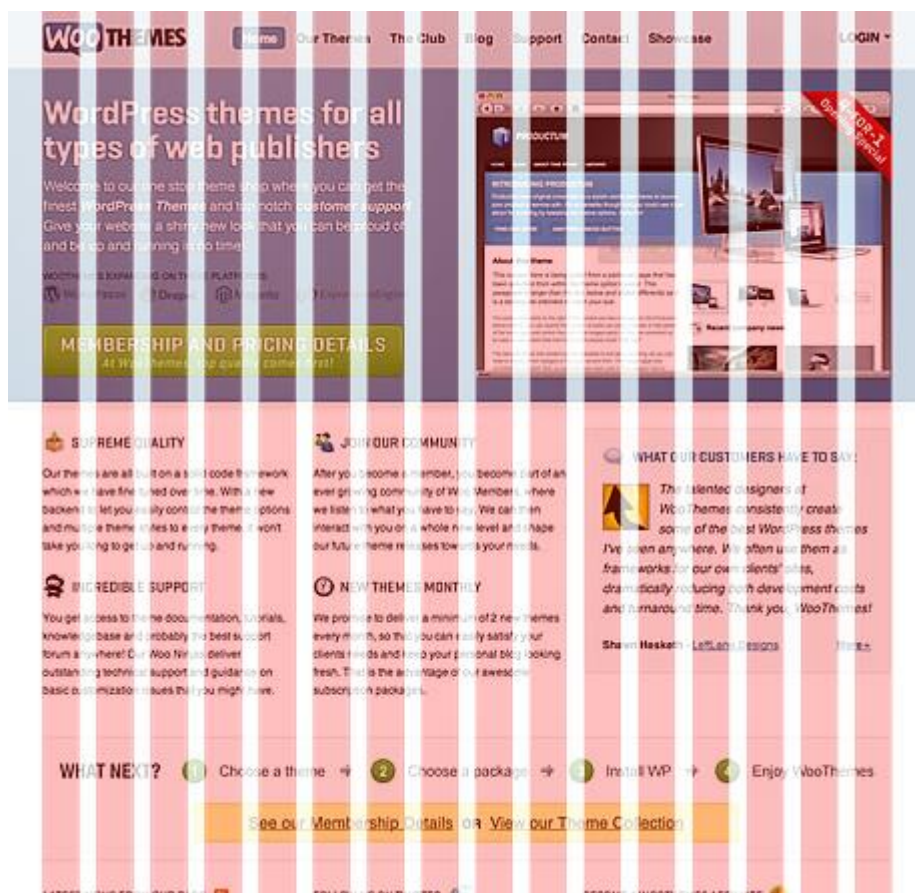
*kód 3-3 – HTML podmíněný komentář*

## 4 Rozložení layoutu

Dle autora se před samotným kódováním designu musí vývojář rozhodnout, jaký zvolí přístup k tvorbě layoutu. Jak uvádí Kayla Knight ve svém internetovém článku [11], má kodér na výběr z několika základních možností: fixní layout nebo fluidní layout. Fluidní layout se přitom ještě dále dělí na plně fluidní layout, fluidní layout s maximální šířkou (tzv. fluid-fixed) a skokový layout. Každá z těchto možností má svá pro a proti. Tak jak uvádí Ethan Marcotte ve své knize [2], rozložení layoutu je prvním pilířem responzivního webdesignu.

### 4.1 Grid

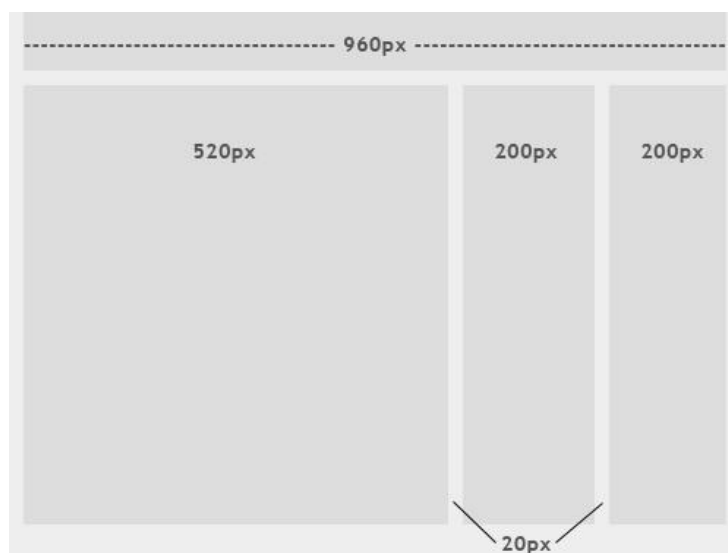
Rozvržení prvků na webové stránce se u responzivních webů velmi často řídí tzv. gridem. Grid je hrubá struktura, podle které jsou umísťovány jednotlivé prvky webdesignu [12]. Takovýto grid se skládá nejčastěji z 10 nebo 12 stejně širokých sloupců, jejichž šířka je v případě responzivní webové stránky udána v procentech. Šířka jednotlivých prvků jako jsou bloky textu, bannery apod. se poté řídí počtem sloupců, které zaujímají, viz obr. 4-1.



obr. 4-1 – Schéma gridu [12]

## 4.2 Fixní layout (fixed)

Fixní layout je snem každého kodéra webového designu. Všechny prvky mají předem stanovou pevnou šířku. Jedná se o nejjednodušší přístup. Kódování takovéto stránky je velmi jednoduché a rychlé. Pevná šířka prvků je vždy stejná neheldě na prohlížeč viz schéma na obr. 4-2.



obr. 4-2 – Schéma fixního layoutu [Zdroj: smashingmagazine.com]

Nevýhodou fixního designu je, že pokud si uživatel zobrazí web na displeji s nižším rozlišením, než pro jaký byl web optimalizován, v prohlížeči neuvidí celý web najednou, ale bude muset za pomoci posuvníku zobrazit přesahující část webu. Naopak pro uživatele s vysokým rozlišením displeje se web zobrazí příliš malý a není tak využit potenciál jejich rozlišení [11].

## 4.3 Fluidní layout (fluid)

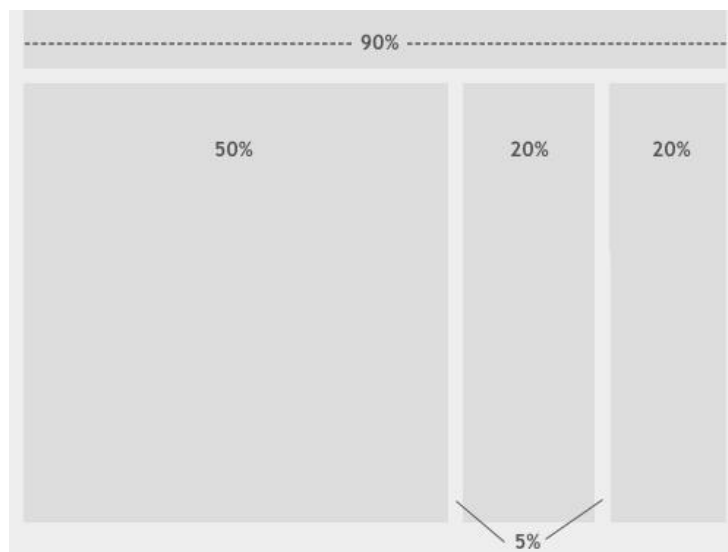
### 4.3.1 Plně fluidní layout

Fluidní (flexibilní) layout, viz schéma na obr. 4-3, na rozdíl od fixního nemá pevně definovanou šířku, nýbrž je šířka udána v procentech. Layout se tak do jisté míry přizpůsobuje šířce displeje uživatele. Kodér však nemá plnou kontrolu nad šíří jednotlivých prvků na webu. Toto je problém hlavně u obrázků a videí, která často musí mít stanovenou pevnou velikost nebo alespoň minimální velikost.

Plně fluidní layout je postaven na relativních jednotkách *em* nebo *rem*.



Jednotka *em* vyjadřuje relativní velikost fontu daného elementu. Např. 2em udává dvojnásobnou velikost fontu daného elementu. Jednotka *rem* patří mezi ty novější relativní jednotky a její podpora je dostupná až od Internet Explorer 9 [13].



obr. 4-3 – Schéma fluidního layoutu [Zdroj: smashingmagazine.com]

#### 4.3.2 Fluid-fixed layout

Při velmi malém rozlišení displeje může dojít ke „kolizi“ v zobrazení jednotlivých prvků a naopak na displeji s vysokým rozlišením může dojít k nadměrnému roztažení layoutu.

Takovýto layout pak může být doplněn o svou minimální a maximální šířku, kterou nepřekročí. O takovém layoutu poté hovoříme jako o fluid-fixed. To však nemusí vždy zamezit výše zmíněným problémům, protože vlastnost *min-width* a *max-width* je podporována u Internetu Exploreru až od verze 6 [11][13][14].

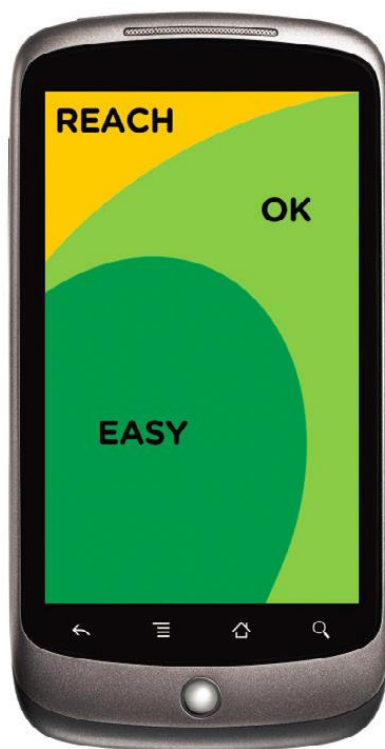
#### 4.3.3 Skokový layout

Skokový layout využívá pevné šířky, avšak za pomoci media queries stanoví body zlomu, tedy šířky displeje, při kterých si webová stránka přepne jiný zdroj kaskádových stylů, který přizpůsobí pevné šířky nové šířce displeje. Při zmenšování okna prohlížeče tak nedochází plně ke kontinuálnímu zmenšování prvků na webové stránce, ale při dosažení určité šířky okna prohlížeče se skokově prvky jeho obsahu zmenší.

#### 4.4 Ovládání pracovní plochy

Ať již kodér zvolí jakýkoliv typ layoutu, musí mít vždy dle autora na paměti, aby se jednotlivé prvky rozvržené v layoutu dali jednoduše ovládat. Luke Wroblewski ve své knize [15] uvádí dotykové schéma displeje pro uživatele, kteří drží mobilní telefon v pravé

ruce. Na obr. 4-4 je toto schéma znázorněno. Schéma zobrazuje oblasti displeje, které jsou s různým stupněm námahy dosažitelné palcem pravé ruky, je-li mobilní telefon uchopen právě v pravé ruce. Pro dosažení některých míst (oblast REACH), je u zařízení s větším displejem zapotřebí přehmatu a změny uchopení mobilního telefonu, proto by v těchto místech dle autora neměli být důležité a často používané prvky, hlavně pak ovládací prvky webové stránky.



*obr. 4-4 – Dotykové schéma [15] str. 73*

## 5 Flexible images

Dalším pilířem responzivního webdesignu [2] jsou flexibilní obrázky (flexible images). Jak zacházet s obrázky na webové stránce pro mobilní zařízení je několik způsobů.

### 5.1 HTML5 <PICTURE>

Jednou z možností, i když zatím nedostupnou, jak optimalizovat velikost obrázků je využití HTML5 a tagu <PICTURE> (kód 5-1). Použití tohoto tagu vyžaduje uložení obrázku v několika velikostech pro každou zvolenou velikost displeje [2]. Tento způsob s sebou přináší několik výhod i nevýhod.

```
<picture>
  <source srcset="img/large.jpg" media="(min-width: 800px)">
  <source srcset="img/default.jpg">
  <img srcset="img/default.jpg" alt="A gray cat looks
    on with anticipation">
</picture>
```

*kód 5-1 – HTML5 <PICTURE> [2] str. 64*

Výhodou je v tomto případě to, že dané zařízení, přes které uživatel přistupuje na web, získá obrázek v optimalizované velikosti. To také znamená úsporu v přenosu dat, což je obzvláště v případě datového tarifu výhodou. Obrázek optimalizovaný pro mobilní zařízení je tedy méně náročný na datový přenos a tím se také zvýší rychlost jeho stažení a načtení.

Nevýhodou je ukládání každého obrázku v několika velikostech. V případě použití redakčního systému tento úkon nijak administrátora webové stránky nezatíží, poněvadž každý lepší redakční systém umí vložený obrázek automaticky uložit v požadovaných velikostech. Další a již klasickou nevýhodou HTML5 je podpora ve webových prohlížečích.

Tag <PICTURE> je zatím pouze ve fázi přípravy a v současné době ještě není v HTML5 dostupný [2].

### 5.2 Fluidní obrázky

Snadným, avšak ne zcela kompatibilním, způsobem jak docílit responzivních obrázků je použití vlastnosti elementů *max-width*. Tato vlastnost byla zmíněna již v souvislosti s fluidním layoutem. U obrázků vykonává kodérům podobnou službu.

Ethan Marcott ve své knize [2] uvádí následující příklad (kód 5-2 a kód 5-3). Na ukázce kód 5-3 si lze navíc povšimnout jeho techniky výpočtu velikostí elementů, a to absolutních

i relativních, kdy velikost získanou podílem dvou jiných hodnot nikterak nezaokrouhluje a do kaskádových stylů zapisuje celou její hodnotu. K tomuto způsobu zápisu neceločíselných hodnot se přiklání i sám autor.

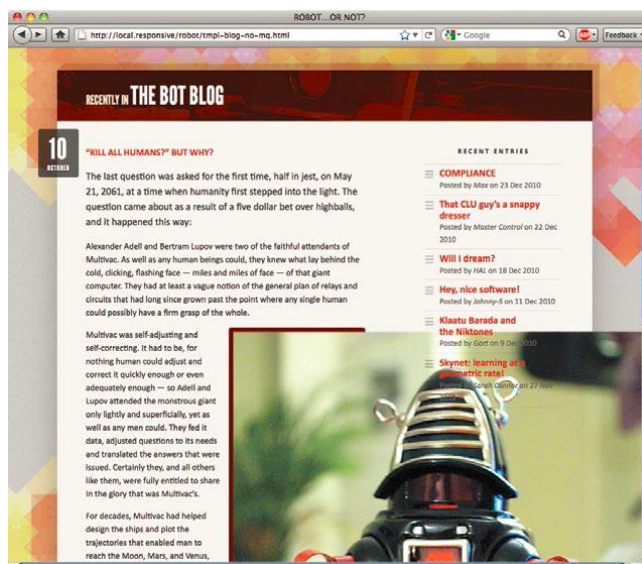
```
<div class="figure">
  <p>
    
    <b class="figcaption">Lo, the robot walks</b>
  </p>
</div>
```

kód 5-2 – HTML, fluidní obrázek [2] str. 46

```
.figure {
  float: right;
  margin-bottom: 0.5em;
  margin-left: 2.55319149%; /* 12px / 470px */
  width: 49.14893617%; /* 231px / 470px */
}
```

kód 5-3 – CSS, fluidní obrázek [2] str. 46

Obrázek je zabalen do elementu *figure*, kterému je nastaveno zarovnání, odsazení a flexibilní šířka. Element je tak definován a zarovnán v prostoru. Šířka elementu *figure* ale nijak obrázek neovlivňuje.



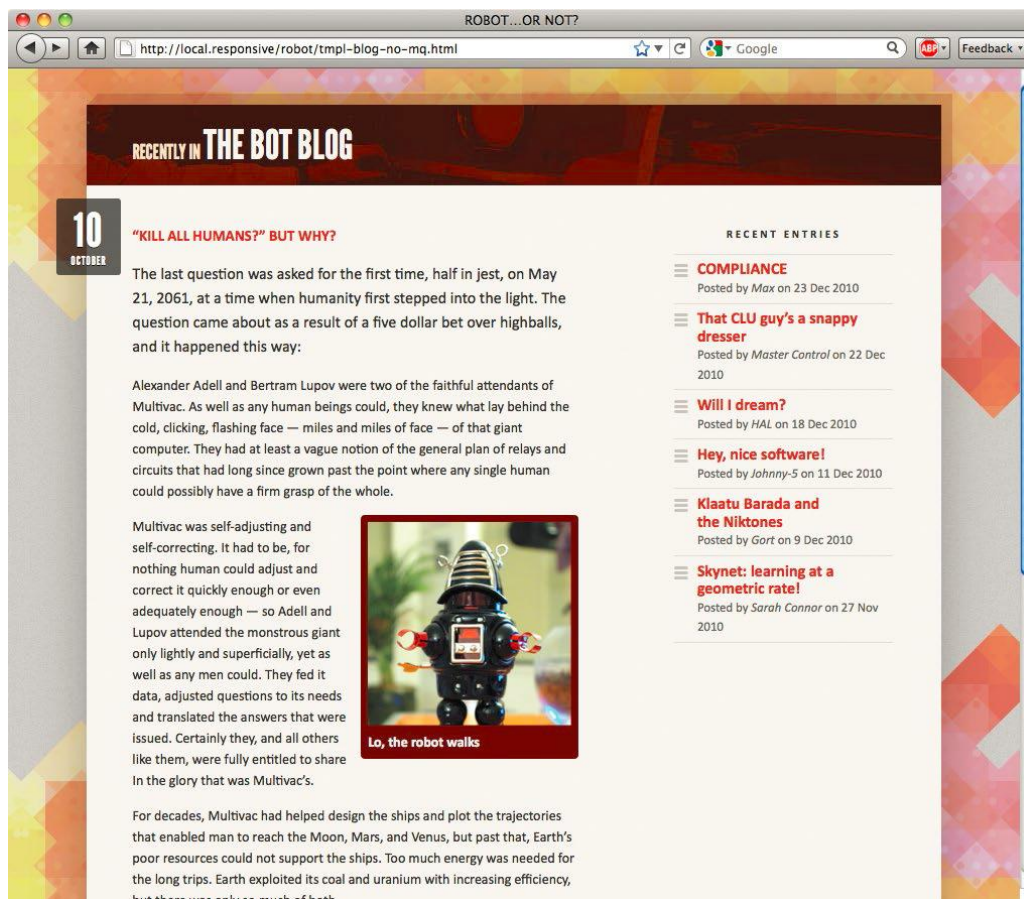
obr. 5-1 – Fluidní obrázek 1 [2], str. 47

CSS kód je ještě neúplný. Velikost obalového elementu neurčuje velikost obrázku, který je v elementu zanořen. Obalový element se zmenšuje a zvětšuje na základě šířky jeho rodičovského elementu, obrázek ale zůstává stále stejně velký. Dochází k jeho „přetečení“ přes okraj obalového elementu.

Chybí zde klíčová vlastnost obrázku, která obrázku v elementu *figure* nastaví maximální šířku *max-width* na hodnotu *100%* (kód 5-5). Obrázek se zmenší na šířku obalového elementu (obr. 5-2).

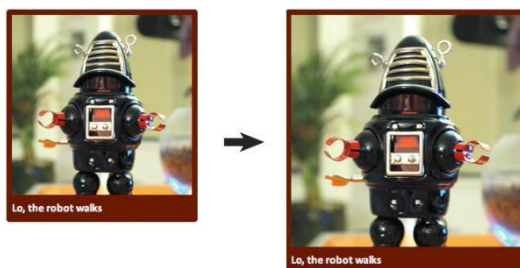
```
.figure img {  
  max-width: 100%;  
}
```

kód 5-4 – CSS, fluidní obrázek, *max-width* [2] str. 48



obr. 5-2 – Fluidní obrázek 2 [2] str. 49

Velkou výhodou vlastnosti *max-width* je, že se obrázek zmenší se zachováním poměru stran a nedochází tak k deformaci obrázku (obr. 5-3).



obr. 5-3 – Fluidní obrázek, zachování poměru stran [2] str. 50

Vlastnost *max-width* se dá použít nejen pro obrázky, ale také pro videa a další objekty.

I zde se kodéři setkávají s problémem kompatibility vlastnosti *max-width* u prohlížečů Internet Explorer. U této vlastnosti se však jedná pouze o problém u verze prohlížeče 6 a starší [2]. Autor obecně nepovažuje zajištění kompatibilního zobrazení pro zmíněnou verzi prohlížeče za důležitou, a proto tento problém neshledává nikterak závažným. Bude-li vývojář chtít alespoň trochu ochránit uživatele Internetu Exploreru 6 a starších verzí, je možné za pomoci podmíněného komentáře (viz kód 3-3) nastavit pro tyto verze maximální šířku obrázku na hodnotu 100%. Obrázek sice obalový element nepřeteče, ale nezachová si poměr stran. Jedná se tedy o řešení, které umožní uživateli web prohlížet, ale nedopřává mu příliš kvalitní obsah, který však uživatelé tohoto webové prohlížeče nemohou, dle autora, příliš očekávat.

### 5.3 Flexibilní obrázky na pozadí

Ethan Marcott dále v knize [2] popisuje jak dosáhnout flexibilních obrázků na pozadí webové stránky. Příklad opět demonstruje na své cvičné webové stránce blogu, která ilustruje jeho ukázky zdrojových kódů v celé knize.

U flexibilních obrázků na pozadí se zabývá problémem barevného pozadí se vzorem. Webová stránka má dva základní bloky. Blok s obsahem a blok s výpisem dalších příspěvků. Bloky jsou na sobě výškově nezávislé, každý blok může mít různou výšku v závislosti na svém obsahu, není tedy možné nastavit pozadí každému bloku zvlášť, poněvadž by ve většině případů pozadí jednoho bloku nedosahovalo na konec webové stránky. Poněvadž mají bloky flexibilní šířku, je potřeba počítat s tím, že se jejich šířka v závislosti na šířce webového prohlížeče mění.



obr. 5-4 – Základní grafický návrh [2] str. 57





## 6 Media queries

Třetím a posledním pilířem responzivního webdesignu [2] jsou media queries, které umožňují využívat různé části CCS stylů nebo celé různé CSS soubory například podle detekce velikosti, orientace či typu displeje. Níže autor uvádí ukázky CSS3 media queries.

Syntaxe zápisu media queries je dvojího typu. Podmínka se zapisuje do kaskádových stylů (kód 6-1) nebo přímo do HTML (kód 6-2). Autor ze svých zkušeností upřednostňuje zápis do CSS souborů. Poté jsou kaskádové styly zcela odděleny od zdrojového kódu šablony, která je těmito kaskádovými styly graficky formátována. Takto je dodržen obecný formát, kdy je sada CSS stylů oddělena od implementace zdrojového kódu webové stránky.

```
@media ( podmínky ) {  
    /* CSS kód, který je využíván pouze při splnění podmínek. */  
}
```

kód 6-1 – CSS, media queries

```
<link rel="stylesheet" href="mobile.css" media=" podmínky ">
```

kód 6-2 – HTML, media queries

Základním způsobem jak detekovat mobilní zařízení je dle šířky viewportu *max-width* (kapitola 7.2), nebo šířky displeje obrazovky *max-device-width*.

Aby bylo možné lépe strukturovat CSS a lépe od sebe odlišit jednotlivá zařízení například dle viewportu, je možné při sestavování podmínek media queries použít logický operátor *AND* (kód 6-3).

```
@media ( podmínka ) AND ( podmínka ) {  
    /* CSS kód, který je využíván pouze při splnění všech podmínek. */  
}
```

kód 6-3 – CSS, media queries, logický operátor

Kromě šířky viewportu (viz kapitola 7.2) a displeje zařízení, je také možné detekovat orientaci daného zařízení, tedy zdali je displej orientován na šířku nebo na výšku, dále detekovat poměru stran obrazovky a detekovat vysokokapacitní displeje typu Retina. Způsob detekce je v ukázce kód 6-4 [16].



```
Detekce orientace zařízení:  
@media (orientation:portrait) { ... } - na výšku  
@media (orientation:landscape) { ... } - na šířku  
  
Detekce poměru stran obrazovky:  
@media screen and (device-aspect-ratio: 16/9) { ... }  
  
Detekce vysokokapacitních displejů typu Retina:  
@media only screen and (-webkit-min-device-pixel-ratio : 1.5),  
only screen and (min-device-pixel-ratio : 1.5) { ... }
```

*kód 6-4 – CSS, media queries, detekce vlastností displeje*

Ani u media queries se vývojáři nevyhnou problému s kompatibilitou u prohlížečů Internet Explorer. Uživatelům Internetu Exploreru verze 8 a starší jsou media queries odepřeny.

## 7 Další technická řešení responzivního webdesignu

### 7.1 Způsoby identifikace mobilních zařízení

Základní myšlenkou responzivního webdesignu je přizpůsobení vzhledu webové stránky pro dané zařízení. Aby bylo možné přizpůsobit vzhled webové stránky, je zapotřebí koncové zařízení rozpoznat, tedy určit, zdali se jedná o klasické PC nebo o mobilní zařízení. K rozpoznání zařízení, které návštěvník webu používá, slouží dvě základní metody. Na straně klienta a na straně serveru [17].

#### 7.1.1 Na straně klienta

Při detekci zařízení na straně klienta je serverem vždy dodán na koncové zařízení stejný výstup. Způsob vykreslení se určuje až ve webovém prohlížeči. Pomocí CSS3 je možné detekovat nejen, zdali se jedná o mobilní webový prohlížeč, ale také je možno detekovat například šířku okna prohlížeče nebo přímo displeje zařízení a tím dokonale přizpůsobit vzhled webové stránky. Ve většině případů se k detekci využívá media queries, ale je možné využít i JavaScript [17].

```
/* široká obrazovka */

@media screen and (max-width: 1100px) {
  body {
    font-size: 100%;
  }
}

/* iPad / střední šířka */

@media screen and (max-width: 768px) {
  body {
    font-size: 95%;
  }

  #main-wrapper aside {
    display: none;
  }
}

/* chytré telefony */

@media screen and (max-width: 420px) {
  body,
  #comments {
    font-size: 85%;
  }
}
```

kód 7-1 – CSS, detekce zařízení na straně klienta [Zdroj: igloonet.cz]

Nevýhodou tohoto postupu je, že některé starší webové prohlížeče nepodporují tento způsob detekce [17].

### 7.1.2 Na straně serveru

Druhou možností jak detekovat zařízení je na straně serveru, kdy je typ zařízení rozpoznán pomocí parametru User-Agent v hlavičce HTTP. Detekovat je možné platformu zařízení, která velmi napoví, o jaké zařízení se jedná. V parametru User-Agent se také přenáší informace o prohlížeči [17][18].

```
User-Agent Google Chrome:  
Mozilla/5.0 (X11; Linux i686) AppleWebKit/534.30 (KHTML, like Gecko)  
Chrome/12.0.742.12 Safari/534.30  
  
User-Agent Mozilla Firefox:  
Mozilla/5.0 (X11; Linux i686; rv:6.0.2) Gecko/20100101 Firefox/6.0.2  
  
User-Agent iPhone:  
Mozilla/5.0 (iPhone; U; CPU iPhone OS 3_0 like Mac OS X; en-us)  
AppleWebKit/528.18 (KHTML, like Gecko) Version/4.0 Mobile/7A341 Safari/528.16
```

*kód 7-2 – User-Agent, detekce zařízení na straně serveru [Zdroj: igloonet.cz]*

Nevýhodou tohoto způsobu detekce zařízení je, že každý webový prohlížeč a každá verze webového prohlížeče má vlastní parametr User-Agent. S rostoucím množstvím webových prohlížečů a jejich verzí je téměř neudržitelné pokrýt všechny možnosti, které se mohou v parametru User-Agent objevit. K odstranění tohoto problému slouží veřejné databáze verzí mobilních zařízení, které po předání obsahu parametru User-Agent vrací požadované informace. Takovouto databází je například WURFL [17].

## 7.2 Viewport

Viewport označuje šířku okna prohlížeče, tedy prostor pro vykreslení webové stránky. U webových stránek, které jsou optimalizované pro mobilní zařízení, je důležité pomocí META tagu VIEWPORT a jeho parametru CONTENT, nastavit správnou šířku webové stránky.

```
<meta name="viewport" content="width=device-width">
```

*kód 7-3 – HTML, META VIEWPORT*

Zdrojový kód 7-3 zajistí, aby webový prohlížeč nastavil viewport na šířku displeje zařízení. Prohlížeč poté ví, jak má rozmístit obsah na dostupném prostoru [19].

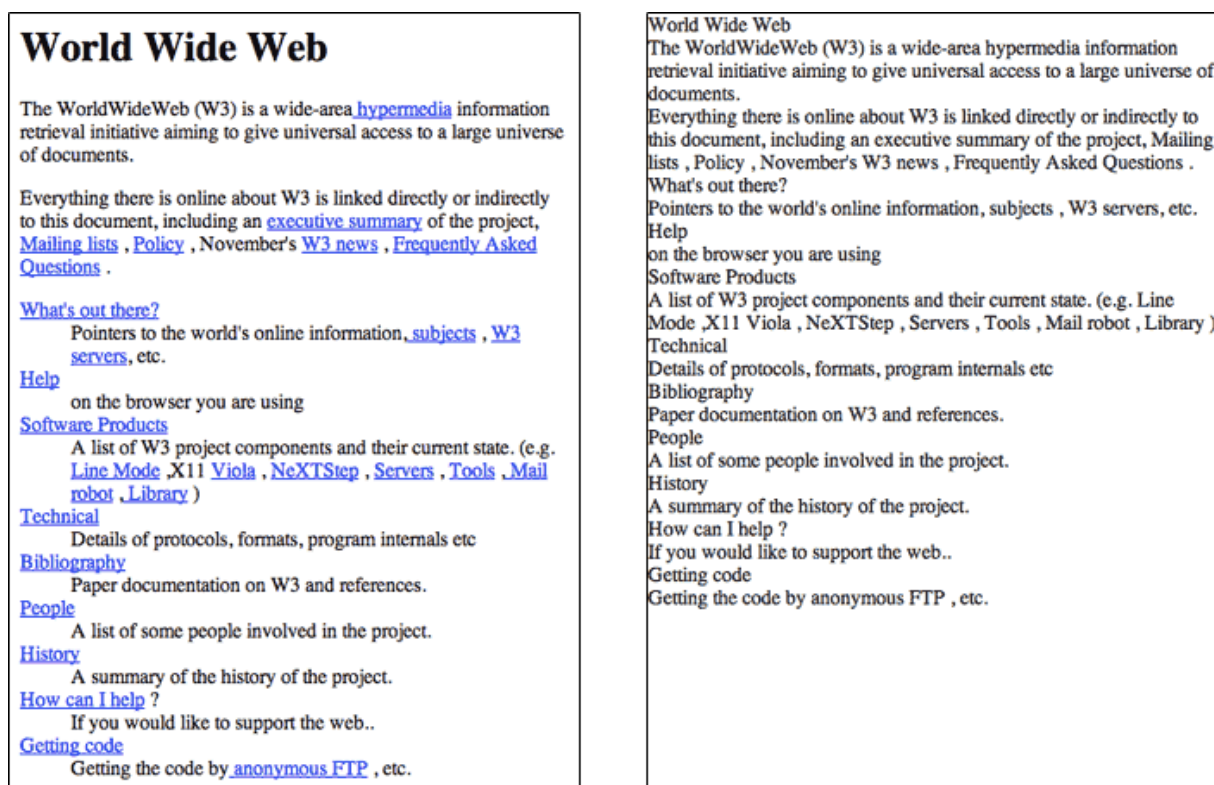
Dále je možné pomocí hodnoty *user-scalable* parametru CONTENT prohlížeči říci, zdali uživatel může obsah webové stránky zoomovat, tedy přibližovat a oddalovat a tím zvětšovat a zmenšovat její obsah.

### 7.3 CSS reset

CSS reset je soubor pravidel zapsaných v jazyce CSS, který resetuje výchozí styly některých HTML elementů, které jsou přednastavené ve webových prohlížečích. Patří mezi ně například zbarvení a podržení odkazů, odsazení odstavců a podobně.

Výhodou využívání CSS reset je ten, že programátora nezaskočí některé výchozí nastavení webového prohlížeče a s kódováním začíná opravdu na „zelené louce“ [20].

Na obr. 7-1 je vidět webová stránka bez a s využitím CSS reset.



obr. 7-1 – CSS reset [Zdroj: cssreset.com]

Základní CSS reset může vypadat například takto, viz kód 7-4, kdy resetuje výchozí nastavení vnějšího a vnitřního odsazení a vlastnosti rámečku u základních HTML elementů, jako jsou HTML, BODY, DIV, SPAN, nadpisy a odstavec.

```
html, body, div, span, h1, h2, h3, h4, h5, h6, p {
    margin: 0;
    padding: 0;
    border: 0;
}
```

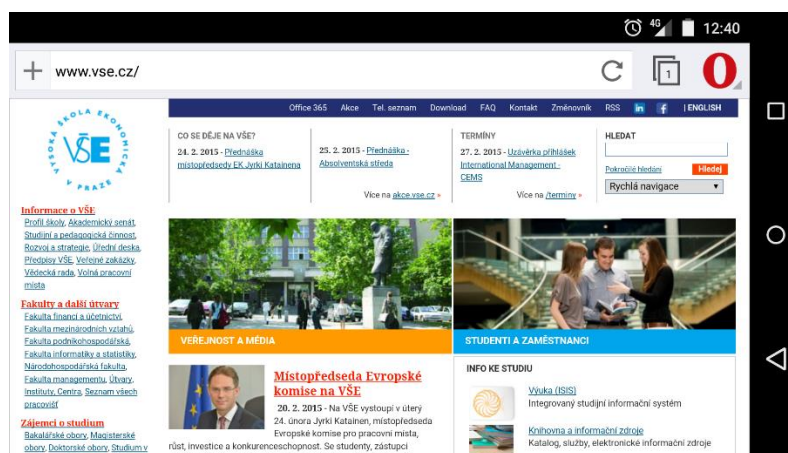
kód 7-4 – CSS reset

Mezi nejznámější a nepoužívanější CSS resety patří tyto [20]:

- Eric Meyer's "Reset CSS" 2.0
- HTML5 Doctor CSS Reset
- Yahoo! (YUI 3) Reset CSS
- Universal Selector '\*' Reset
- Normalize.css 1.0

## 7.4 Možnosti testování responzivního zobrazení

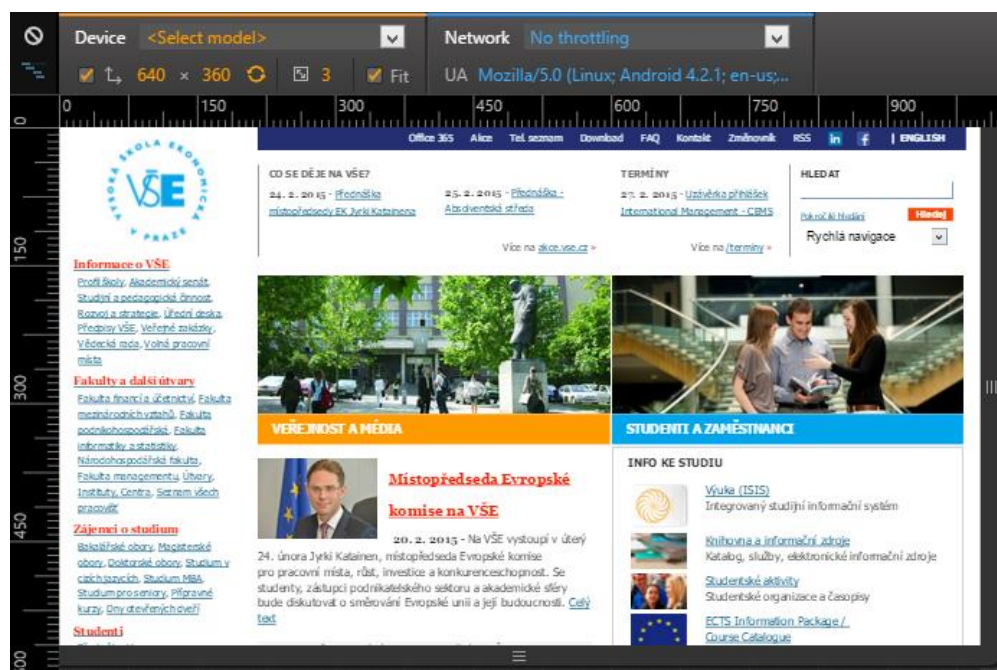
Vývojář má stoprocentní jistotu, že se jeho webová stránka zobrazí správně na všech mobilních zařízeních, pouze pokud svůj web otestuje na všech dostupných zařízeních všech značek se všemi typy prohlížečů. Autor upozorňuje, že je to úkol nejen nelehký, ale přímo nemožný. Není v silách vývojáře vzít do ruky každé mobilní zařízení a otestovat v něm své webové stránky, tak jako na obr. 7-2.



obr. 7-2 – Snímek obrazovky webu vse.cz na zařízení LG Nexus 5

K usnadnění této činnosti slouží různá rozšíření do webových prohlížečů, která simulují použití například chytrého telefonu nebo tabletu. Webový prohlížeč Google Chrome a Opera poskytují již v základu široké možnosti pro vývojáře. Opera navíc vytvořila celou verzi svého prohlížeče přizpůsobeného pro vývojáře pod názvem Opera developer. Mezi oblíbená rozšíření patří Firebug nebo například Web Developer, viz obr. 7-3, který je

dostupný pro prohlížeče Google Chrome, Opera a Mozilla Firefox. Microsoft poskytuje ve svém prohlížeči Internet Explorer verze 10 a vyšší také vývojářské nástroje, včetně kompatibilního zobrazení na starších verzích tohoto prohlížeče.



obr. 7-3 – Náhled webu vse.cz ve vývojářském nástroji Opera Developer

Další možností jsou on-line webové nástroje, které po zadání URL adresy webové stránky zobrazí náhled webu v základních rozlišeních a orientacích displeje.

Mezi takovéto on-line webové nástroje patří například: ami.responsivedesign.is (ukázka viz obr. 7-4), deviceponsive.com, responsivetest.net, responsive.is, screenqueri.es nebo quirkttools.com [21]. Tyto nástroje sice dokáží vykreslit webovou stránku a simulovat tak mobilní zobrazení, není však vždy jisté, že web bude na skutečném zařízení takto vypadat. Slouží spíše pro rámcový náhled, zdali je webová stránka na malém displeji přehledná, zdali se dobře ovládá tahem prstu apod. Nabízí-li nástroj využití proxy serveru je možné za pomoci této funkce simulovat, jak se webová stránka zachová, přistoupíte-li na web například z jiného státu nebo s jinak jazyčným prohlížečem. Náhled webových stránek se dá také využít pro prezentaci webu například klientovi či sponzorovi.





obr. 7-4 – Náhled webu vse.cz [náhled vytvořen na [ami.responsivedesign.is](http://ami.responsivedesign.is)]

Nástroje, které poskytují simulaci mobilního prostředí, jsou velmi užitečné a vývojářům usnadní práci. Vždy by však měl dle autora vývojář své dílo otestovat alespoň na několika málo skutečných mobilních zařízeních s těmito dnes hojně využívanými rozlišeními displejů.

Klasické PC – 1600 x 992 px

Notebook – 1280 x 802 px

Tablet – 768 x 1024 px

Mobilní telefon – 320 x 480 px

## 8 CSS preprocessory

CSS preprocesor je nástroj (programovací jazyk), který umožňuje rozšířit základní syntaxi programovacího jazyka CSS, nebo spíše umožňuje vytvořit si „vlastní“ syntaxi pro zápis kaskádových stylů a poté vygenerovat výsledné CSS [22][23][24].

CSS preprocessory pracují na dvou základních principech, a to buď na straně klienta nebo na straně serveru nebo případně umožňují obojí [22].

Použití na straně klienta obnáší překlad „vlastní“ syntaxe až přímo ve webovém prohlížeči. Tento úkon je náročný na výkon a tedy prodlužuje dobu načítání webové stránky. Proto existují v CSS preprocesorech tzv. hlídači, kteří kontrolují, zdali se daný soubor, pojmenovaný například *style.less*, změnil a pouze v takovém případě provedou znovu jeho překlad [23].

Při využívání principu na straně serveru je potřeba doinstalovat danou knihovnu s kompilátorem přímo na server, kde se při změně souboru, opět například *style.less*, automaticky tento soubor přeloží a vygeneruje se soubor s klasickým CSS. Tímto způsobem je méně zatěžován webový prohlížeč uživatele, ale využívání CSS preprocesorů vyžaduje kompatibilní instalaci kompilátoru [24].

CSS preprocessory obecně usnadňují zápis složitých a rozsáhlých kaskádových stylů, které se při vyšší složitosti stávají nepřehlednými. Dále umožňují využívat například matematické výrazy, proměnné, tzv. mixiny a jmenné prostory [23][24][25][26][27].

Jednotlivé CSS preprocessory se od sebe liší většinou jen syntaxí programovacího jazyka [6].

### 8.1 Funkce CSS preprocesorů

#### 8.1.1 Vnořené definice

Vnořené definice umožňují „pročištění“ kódů CSS od duplikace názvů nadřazených selektorů a tříd v kaskádě a tím CSS kód zjednodušit a zpřehlednit [22].

Na níže uvedeném příkladu je vidět tentýž kód zapsaný bez a s CSS preprocesorem [22][23][24][25].



<pre>/* klasické CSS */  .box {   width: 50%; } .box .title {   font-size: 2em; } .box .text {   font-size: 1em; } .box .text p {   margin: 0.5em 0; }</pre>	<pre>/* CSS s vnořenými definicemi */  .box {   width: 50%;   .title {     font-size: 2em;   }   .text {     font-size: 1em;     p {       margin: 0.5em 0;     }   } }</pre>
--	---

kód 8-1 – Porovnání vnořené definice CSS a LESS

### 8.1.2 Proměnné

Velkou výhodou oproti běžnému CSS je možnost definování proměnných. Vhodně zvolené proměnné pak jsou definovány na začátku CSS dokumentu a změnou jejich hodnoty se provede změna všude tam, kde je daná proměnná použita. Pomocí proměnných se mohou například jednoduše globálně měnit barvy, písma, velikosti a podobně [22][23].

<pre>/* Bez proměnných */  h1 {   color: #ffcc00; }</pre>	<pre>/* S proměnnými */  @base-font-color: #ffcc00;  h1 {   color: @base-font-color; }</pre>
---	--

kód 8-2 – Porovnání proměnných CSS a LESS

### 8.1.3 Matematické výrazy

Na rozdíl od samotného CSS, s využitím preprocesorů odpadá manuální dopočítání šířek, délek a dalších podobných vlastností. Nyní je možné jednoduše sčítat, odečítat, násobit a dělit pixely či jiné jednotky [22][23].

```
@base: 5%;
@filler: @base * 2;
@other: @base + @filler;

color: #888 / 4;
background-color: @base-color + #111;
height: 100% / 2 + @filler;
```

kód 8-3 – LESS, matematické výrazy [Zdroj: lesscss.cz]

Pokud je potřeba například rozdělit *DIV* o šířce 500 px na třetiny, stačí jeho šířku vydělit třemi. Sčítat se dají například i barvy. Chceme-li základní barvu zesvětlit nebo ztmavit, stačí připsat následující řádek kódu.

#### 8.1.4 Mixin

Pomocí mixinu, viz kód 8-4, je možné předávat vlastnosti jedné třídy třídě jiné. Třída může být doplněna o parametry. V praxi to oproti klasickému CSS vypadá tak, že se nadefinuje například třída *.h1* pro nadpisy které mají vypadat jako nadpis H1. Třída bude obsahovat informace o barvě, velikosti a fontu písma, které budou nadefinovány v parametrech. Při použití mixinu získá daná třída vlastnosti třídy *.h1* s předem nadefinovanými defaultními hodnotami anebo hodnotami nově nadefinovanými [22][23].

<pre>/* LESS */  .rounded-corners (@radius: 5px) {   border-radius: @radius;   -webkit-border-radius: @radius;   -moz-border-radius: @radius; }  .header {   .rounded-corners; }  .footer {   .rounded-corners(10px); }</pre>	<pre>/* Výsledné CSS */  .header {   border-radius: 5px;   -webkit-border-radius: 5px;   -moz-border-radius: 5px; }  .footer {   border-radius: 10px;   -webkit-border-radius: 10px;   -moz-border-radius: 10px; }</pre>
---	--

kód 8-4 – LESS, mixins v porovnání s CSS

#### 8.1.5 Práce s barvami

CSS preprocesory umožňují pracovat s barvami za pomoci přednastavených funkcí. Mezi tyto funkce u CSS preprocesoru LESS patří například *lighten()* nebo *darken()*, které mají dva parametry. Prvním parametrem je vstupující barva a druhým parametrem je změna odstínu vyjádřená procentem. Autor uvádí názorný příklad použití funkce *darken(#0000FF, 20%)*. Funkce má na vstupu modrou barvu v hexadecimálním zápisu, kterou funkce *darken()* ztmavuje o 20%. Výstupní hodnota funkce s těmito parametry je *#000099*. Porovnání odstínu vstupní a výstupní barvy je znázorněno na obr. 8-1.



obr. 8-1 – LESS, *darken()*, porovnání výstupní barvy

Ukázka kód 8-5 zobrazuje další funkce pro práci s barvami, které jsou definovány v CSS preprocesoru LESS.

```
lighten(@color, 10%); // vrátí barvu, která je o 10% světlejší než @color
darken(@color, 10%); // vrátí barvu, která je o 10% tmavší než @color

saturate(@color, 10%); // vrátí barvu o 10% více sytou než @color
desaturate(@color, 10%); // vrátí barvu o 10% méně sytou než @color

fadein(@color, 10%); // vrátí barvu o 10% méně průhlednou než @color
fadeout(@color, 10%); // vrátí barvu o 10% více průhlednou než @color
fade(@color, 50%); // vrátí barvu @color s 50% průhledností

spin(@color, 10); // vrátí barvu s o 10 stupňů teplejším odstínem než @color
spin(@color, -10); // vrátí barvu o 10 chladnější než @color

mix(@color1, @color2); // vrátí kombinaci barev @color1 a @color2
```

kód 8-5 – LESS, práce s barvami [Zdroj: <http://www.lesscss.cz/>]

### 8.1.6 Jmenné prostory

Jmenné prostory umožňují stejně tak jako v ostatních programovacích jazycích zabalit více definovaných pravidel do jednoho „balíku“, který je možné využít později ve zdrojovém kódu [22][23]. Na kód 8-6 je ukázáno, jak lze odkazu ve třídě *header* přiřadit vlastnosti z přednastavené třídy *bundle button*.

```
.bundle {
  .button() {
    display: block;
    border: 1px solid black;
    background-color: grey;
    &:hover { background-color: white; }
  }
  .tab { ... }
  .citation { ... }
}

.header a {
  color: orange;
  .bundle > .button;
}
```

kód 8-6 – LESS, jmenné prostory [Zdroj: [lesscss.cz](http://lesscss.cz/)]

## 8.2 CSS preprocesory - typy

CSS preprocesorů je celá řada, liší se však většinou jen svou syntaxí. Mezi dva nejrozšířenější patří LESS a SASS. Autor tak usuzuje dle četnosti výskytu těchto dvou CSS preprocesorů v množství článků, které při zpracování diplomové práce přečetl a také dle podpory překladu kódu CSS například u PHP frameworků.

Pro úplnost autor uvádí i několik dalších CSS preprocesorů. Nejsou u nich uvedeny ukázky zdrojových kódů, ale o to více se autor zaměřuje na rozdíly v syntaxi mezi LESS a SASS.

### **8.2.1 LESS**

Jedním z nejpoužívanějších CSS preprocesorů je LESS. Umožňuje kompilaci jak na straně klienta, tak na straně serveru [22][23][24].

Pro veškeré předchozí ukázky zdrojových kódů preprocesoru v kapitole 8, byl použit právě CSS preprocesor LESS.

LESS je deklarativní, to znamená, že zachovává deklarativní povahu CSS. Zápis je snadno čitelný a pochopitelný i pro začátečníky. Není tedy příliš složité přejít z klasického CSS do zápisu pomocí LESS [25].

Domovskou stránkou CSS preprocesoru LESS je web <http://lesscss.org>, který se dočkal i českého překladu na webu <http://www.lesscss.cz/>. Je tedy velmi jednoduše dostupný pro českou skupinu kodérů.

### **8.2.2 SASS**

Zkratka SASS znamená Syntactically Awesome Style Sheets. K využívání SASS je zapotřebí nainstalovat si skriptovací programovací jazyk Ruby [27].

SASS je na rozdíl od LESS imperativní, což znamená, že syntaxe SASS více připomíná jiné programovací jazyky jako je například PHP nebo Ruby. Tento styl zápisu více vyhovuje zkušenějším programátorům [25].

SASS podporuje dvě různé syntaxe. Původní SASS syntaxi a novu SCSS (Sassy CSS), která je přehlednější. Autor při ukázkách kódu SASS preprocesoru preferuje SCSS syntaxi.

Domovskou stránkou s popisem instalace, dokumentací a ukázkami použití je web <http://sass-lang.com/>.

### **8.2.3 Stylus**

Stylu je dalším z řady CSS preprocesorů. Stejně tak jako SASS se jedná o imperativní CSS preprocesor [25].

Jeho domovskou stránkou je web <http://learnboost.github.io/stylus/>, který je v porovnání se stránkami CSS preprocesorů LESS a SASS velmi stručný.

### 8.2.4 Další CSS preprocesory

- **CSS-Crush** (<http://the-echoplex.net/csscrush/>)
- **Myth** (<http://www.myth.io/>)
- **Rework** (<https://github.com/reworkcss/rework>)
- **Clay** (<http://fvisser.nl/clay/>)
- **DtCSS** (<https://code.google.com/p/dtcss/>)

## 8.3 Základní porovnání syntaxe LESS a SASS (SCSS)

Jak již bylo v této kapitole zmíněno, zápis LESS a SASS jsou různého typu. LESS používá deklarativní zápis kódu a SASS imperativní. To znamená, že v některých případech se syntaxe zápisu velmi liší. Ve většině případů si jsou však oba způsoby zápisu velmi podobné a mnohdy se liší pouze v použití zástupného znaku.

### 8.3.1 Proměnné

LESS používá na začátku proměnných symbol `@`. SASS ve své syntaxi používá na začátku proměnných symbol `$`. Další manipulace s proměnnými se již nikterak neliší.

<pre>/* LESS */  @base-font-color: #ffcc00;  h1 {     color: @base-font-color; }</pre>	<pre>/* SASS (SCSS) */  \$base-font-color: #ffcc00;  h1 {     color: \$base-font-color; }</pre>
--	---

*kód 8-7 – Porovnání proměnných LESS a SASS (SCSS)*

### 8.3.2 Funkce

I CSS preprocesor SASS umí pracovat s barvami pomocí předdefinovaných funkcí. Většina funkcí se jmenuje stejně. Jediným rozdílem je tedy syntaxe zápisu proměnných. Pokud jsou parametry funkcí předávány v proměnné, LESS používá na začátku proměnné `@`, SASS používá `$`.

### 8.3.3 Mixins

V případě mixinů je syntaxe SASS více obsáhlá. Deklarace mixinu se uvozuje pomocí `@mixin` a při jeho využití je obsah mixinu vkládán pomocí `@include`.

<pre> /* LESS */  .rounded-corners (@radius) {   border-radius: @radius;   -webkit-border-radius: @radius;   -moz-border-radius: @radius; }  .header {   .rounded-corners(10px); } </pre>	<pre> /* SASS (SCSS) */  @mixin rounded-corners (\$radius) {   border-radius: \$radius;   -webkit-border-radius: \$radius;   -moz-border-radius: \$radius; }  .header {   @include rounded-corners(10px); } </pre>
---	--

*kód 8-8 – Porovnání mixins LESS a SASS (SCSS)*

### 8.3.4 Shrnutí

Jak již autor zmínil, obecně platí, že deklarativní zápis využívající LESS je bližší začátečníkům nebo například grafickým designerům, kteří s kaskádovými styly taktéž někdy přijdou do styku. Programátorům zvyklým na imperativní zápis zdrojových kódů, může dle autora více vyhovovat syntaxe preprocesoru SASS.

Rozdíly v zápisu nejsou však nijak velké a naučí-li se někdo jedné syntaxi, může se snadno naučit i té druhé. Pokud začátečník nejprve začne například s preprocesorem LESS a později přejde k SASS, může si své předchozí práce překonvertovat do SASS pomocí různých online konvertorů dostupných zdarma.

## 9 Definice srovnávacích kritérií

Tato kapitola se zabývá popisem a definicí kritérií, podle kterých autor srovnal vybrané frameworky s podporou responzivního webdesignu. Srovnávací kritéria sestavil autor sám metodou expertního odhadu na základě svých zkušeností programátora s využíváním frameworků. Kritérií je celkem šest. Každé kritérium hodnotí daný framework na bodové škále od 0 do 10 bodů, kdy 10 bodů symbolizuje maximální pozitivní počet bodů, které je možné získat. Dále má každé kritérium přiřazenou váhu. Váhy jsou normované, čili součet všech vah přes všechna kritéria dává jedničku. Maximální počet získaných bodů je tedy 60 a maximální počet získaných vážených bodů je 10. Každý ze sledovaných frameworků má svou tabulku, do které jsou vyplňovány hodnoty jednotlivých kritérií a získané body. Vzor této tabulky je uveden v příloze 1.

Některá kritéria nejsou přímo srovnávací, poněvadž dle autora není možné v některých případech přímo rozhodnout, která z možností, kterou daný framework podporuje, je správná nebo lepší. U takových kritérií vždy autor daný důvod slovně vysvětluje. V takových případech se hodnotí například pouze fakt, zdali daný framework podporuje alespoň jednu z výčtu možností, případně zdali jich podporuje více.

Po sestavení všech kritérií a jejich doplnění pro jednotlivé frameworky se ukázalo, že mohou kritéria nabývat v některých případech stejné hodnoty u všech sledovaných frameworků. Ve výsledné tabulce tab. 11-1 autor však tato kritéria ponechává, i když nemají na výsledné hodnocení žádný vliv. Je tedy nasnadě dát takovýmto kritériím váhu nula, poněvadž nemají na celkové hodnocení žádný vliv a ušetřenou váhu pak rozmělnit mezi ostatní kritéria, které by bylo možné více od sebe odlišit a zpřesnit tak výsledné hodnocení. Autor tak však neučinil z jednoho pragmatického důvodu, a to proto, aby bylo možné v budoucnu vložit do tohoto způsobu srovnání a hodnocení frameworků jiný libovolný framework, a porovnat ho tak s již vybranými frameworky. Bude-li v budoucnu potřeba vložit do současného porovnání další framework, je pravděpodobné, že bude mít hodnotu některých kritérií odlišnou od současných frameworků, u kterých se daná hodnota neliší. Tato kritéria jsou také v tabulce zobrazena, aby tak ucelila celkový přehled vybraných frameworků, který poskytuje komplexní informace o jejich funkcionalitě a vlastnostech.

## 9.1 Přístup k tvorbě responzivního webdesignu

Jak již bylo zmíněno v kapitole 3, existují dva základní přístupy k tvorbě responzivního webdesignu. Mobile first (dále „MF“) a desktop first (dále „DF“). V rámci srovnání nelze hodnotit dle toho, který z přístupů daný framework podporuje, poněvadž není obecně jeden přístup uznávaný lepším nežli druhý a oba přístupy mají ve světě své skálopevné zastánce. Hodnotit lze pouze podle toho, kolik těchto přístupů framework podporuje.

**Výčet hodnocených možností kritéria:** žádný, MF, DF.

**Způsob hodnocení:** hodnotí se dle počtu podporovaných přístupů k tvorbě responzivního webdesignu (MF, DF), které daný framework podporuje, a to následovně:

- není podporován žádný: 0 bodů
- je podporován právě jeden: 7 bodů
- jsou podporovány oba dva: 10 bodů

Poněvadž autor nastavil bodování pouze v kladných hodnotách, je rozdíl mezi žádnou podporou a podporou jednoho z typů přístupu celých 7 bodů. Dle autora je podpora alespoň jednoho typu přístupu k tvorbě responzivního webdesignu důležitá. Naopak rozdíl mezi jedním a dvěma podporovanými přístupy již není tak důležitý a bodový rozdíl je zde vyjádřen pouze 3 body.

**Váha kritéria:** 0,2

## 9.2 Typ layoutu

Dalším kritériem srovnávání frameworků je typ layoutu, viz kapitola 4. Autor původně zamýšlel toto kritérium nehodnotit, poněvadž neexistuje obecné mínění, který typ layoutu je lepší. Poněvadž se však jedná o porovnání frameworků s podporou responzivního webdesignu, je u tohoto kritéria hodnoceno, zdali daný framework podporuje tvorbu responzivního (fluid) layoutu. Zdali se jedná o plně fluidní, fluid-fixed nebo skokový již nijak bodové hodnocení neovlivňuje, tato informace je však uvedena ve srovnávací tabulce pro přesnější a podrobnější přehled.

**Výčet hodnocených možností kritéria:** fixed, fluid, fluid-fixed, skokový



**Způsob hodnocení:**

- není podporován fluidní layout: 0 bodů
- je podporován fluidní layout: 10 bodů

**Váha kritéria:** 0,1

### 9.3 Podpora CSS preprocesorů

CSS preprocesorům je věnována celá kapitola 8. Zde taktéž není možné jednoznačně určit, který z preprocesorů je lepší.

Autor zde ve výčtu hodnocených možností kritéria uvádí pouze dva z celé řady CSS preprocesorů. Jedná se o obecně nejrozšířenější a nejvyspělejší preprocesory a jak srovnání ukázalo, také jediné dva, které vybrané frameworky podporují.

**Výčet hodnocených možností kritéria:** žádný, LESS, SASS

**Způsob hodnocení:**

- není podporován žádný: 0 bodů
- podporován právě jeden: 7 bodů
- jsou podporovány oba dva: 10 bodů

Stejně tak jako u přístupu k tvorbě responzivního webdesignu, je i u tohoto kritéria dle autora velmi důležitá podpora alespoň jednoho z CSS preprocesorů. Podpora obou preprocesorů je již jen malé plus s přídatkem 3 bodů.

**Váha kritéria:** 0,25

### 9.4 CSS reset

Funkce CSS resetu je blíže popsána v kapitole 7.3. Jak je v této kapitole zmíněno, existuje několik typů CSS resetů. Dle autora není možné v případě tohoto kritéria určit, který typ CSS resetu je lepší, poněvadž každý slouží pro jiný způsob programování. Například je-li webová stránka programována v HTML5, je vhodné využít CSS reset pro HTML5.

Avšak využití jakéhokoliv CSS resetu zajistí do jisté míry bezpečnější programování. Proto je pouze hodnoceno, zdali daný framework obsahuje některý z těchto typů CSS resetu. Pokud framework používá vlastní zestručnělý CSS reset, je tato možnost dle autora ne zcela vhodná a tudíž i bodově znevýhodněna, poněvadž takový reset nemusí pokrývat všechny přednastavené hodnoty, které pokrývají již zhotovené a otestované CSS resety.

### **Výčet hodnocených možností kritéria:**

- Eric Meyer's "Reset CSS" 2.0 (dále „EM“)
- HTML5 Doctor CSS Reset (dále „D“)
- Yahoo! (YUI 3) Reset CSS (dále „Y“)
- Universal Selector '\*' Reset (dále „U“)
- Normalize.css 1.0 (dále „N“)
- vlastní CSS reset (dále „vlastní“)
- žádný

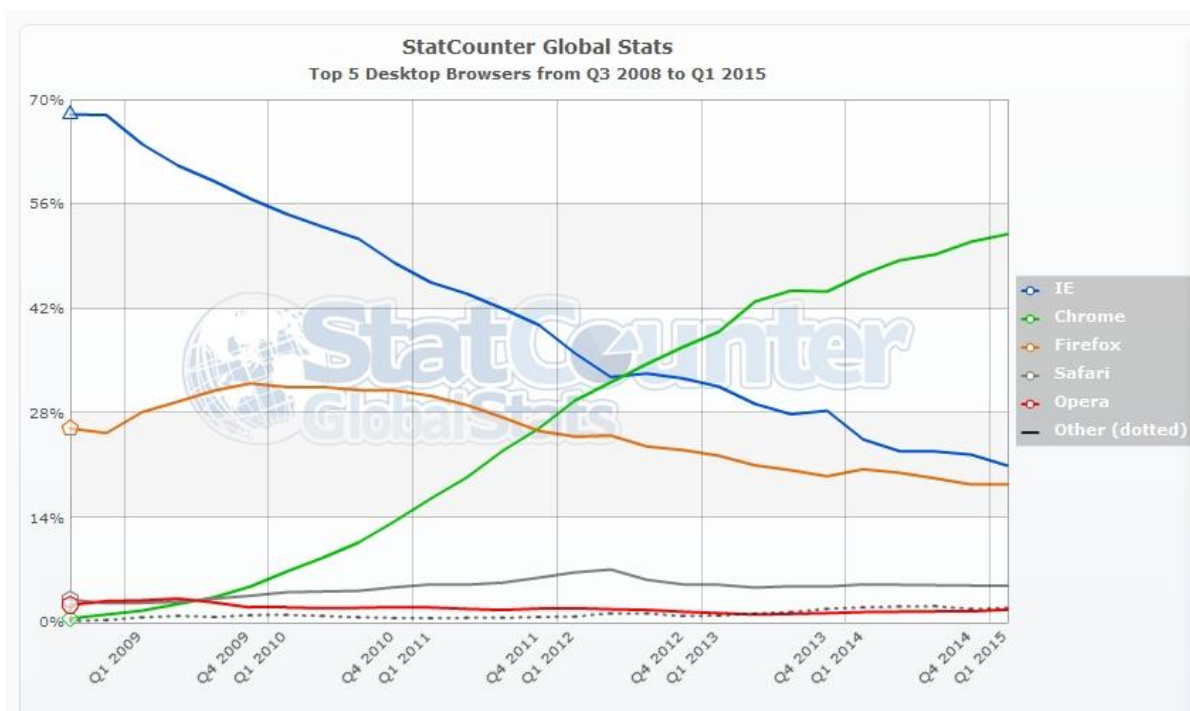
### **Způsob hodnocení:**

- neobsahuje CSS reset: 0 bodů
- obsahuje vlastní zjednodušení CSS reset: 5 bodů
- obsahuje CSS reset: 10 bodů

**Váha kritéria:** 0,1

## **9.5 Kompatibilita s Internet Explorer**

Kompatibilita s webovým prohlížečem Internet Explorer (dále „IE“) je dle autora stále důležitým parametrem srovnání, protože i když za posledních několik let jeho obliba strmě klesá (obr. 9-1), jedná se stále o jeden z nepoužívanějších webových prohlížečů mezi uživateli internetu, a i když je již jen malé procento uživatelů využívající jeho starší verze, je potřeba poskytnout všem návštěvníkům webových stránek kvalitní zobrazení a to bez výjimky. Autor uvažuje zpětnou kompatibilitu jen do verze 7. Od verze 10 jsou již prohlížečem ve značné míře podporovány nejmodernější prvky HTML5 a CSS3 a prohlížení webových stránek se tak stalo pro uživatele Internetu Exploreru kompatibilním [9].



obr. 9-1 – Graf desktopových webových prohlížečů [Zdroj: gs.statcounter.com]

**Výčet hodnocených možností kritéria:** žádná, 7+, 8+, 9+

**Způsob hodnocení:**

- bez zpětné podpory IE: 0 bodů
- podpora 9+: 6 bodů
- podpora 8+: 8 bodů
- podpora 7+: 10 bodů

Hodnotí se způsobem čím více podporovaných verzí prohlížeče Internet Explorer, tím více bodů. Například kompatibilita s prohlížečem verze 9 je hodnocena pouze 5 body, poněvadž se jedná o prohlížeč vydaný v roce 2011, je využíván mnoha uživateli a jeho podpora by dle autora měla být samozřejmostí.

**Váha kritéria:** 0,15

## 9.6 Druh licence

Posledním parametrem, který bude napříč frameworky srovnáván je druh licence. U typu licence se hodnotí její „volnost“, tedy zdali je možné daný framework volně využívat pro nekomerční i komerční účely.

Frameworky poskytují nejčastěji tento druh licence:

- MIT – tato licence umožňuje se software nakládat téměř libovolně. Je možné software používat, kopírovat, upravovat, distribuovat a dokonce i prodávat. Jedinou podmínkou této licence je, že musí být vždy uveden text licence MIT u každé kopie a odvozenin původního software [28].

#### **Výčet hodnocených možností kritéria: MIT**

##### **Způsob hodnocení:**

- placená licence pro nekomerční využití: 0 bodů
- placená licence pouze pro komerční využití: 3 bodů
- volně šiřitelná licence pro komerční i nekomerční využití: 10 bodů

Jak již bylo nastíněno na začátku této kapitoly o hodnotících kritériích, toto kritérium nabývá u všech pozorovaných frameworků stejné hodnoty. Nově pozorovaný framework však může nabízet zcela odlišný typ licence nežli tyto současné a proto u tohoto kritéria zůstává nenulová váha.

**Váha kritéria: 0,2**

### **9.7 Nezařazená kritéria**

Autor zamýšlel zařadit mezi kritéria i následující vlastnosti frameworků. Neučinil tak, ale protože se povětšinou jedná o vlastnosti, které mají všechny vybrané frameworky stejné, jejich zařazením mezi kritéria by nedošlo ke změně pořadí. Jedná se o následující vlastnosti: základní šířka gridu, počet sloupců gridu a JavaScriptové knihovny.

Dle autora je v dnešní době již neodmyslitelnou součástí webových stránek využívání JavaScriptu, i když jeho použitím často dochází ke ztrátě kompatibility, poněvadž ne všechny webové prohlížeče JavaScript a jeho nejnovější knihovny podporují. JavaScript dodává webové stránce patřičnou dynamiku a uživateli zpřijemňuje, je-li ho použito vkusně a s mírou, ovládání webu. Výběr knihoven je však velmi široký a s přihlédnutím k tomu, že neexistuje žádné relevantní pořadí nebo hodnocení jednotlivých knihoven, je tato vlastnost také vynechána, poněvadž hodnotit pouze počet podporovaných knihoven není vypovídající.

## 10 Frameworky s podporou responzivního webdesignu

Před samotným výčtem všech sledovaných frameworků je potřeba zmínit, co to vlastně framework je. Jiří Knesl [29] uvádí, že framework je obecně skupina knihoven, která usnadňuje řešení dané problematiky.

Framework s podporou responzivního webdesignu pak dle autora usnadňuje práci právě v oblasti responzivního webdesignu, protože obsahuje základní responzivní řešení, které je možné dále rozvíjet a dále obsahuje knihovny pro podporu responzivního zobrazení.

Důležité je také upřesnit, že se u frameworků pro responzivní webdesign jedná o tzv. *Front-end* frameworky, tedy nástroje pro podporu vývoje té části webové stránky, kterou vidí návštěvník webu, tedy interaktivní interface [30].

Jako zdroje informací o jednotlivých frameworkcích sloužily jejich domovské webové stránky, které jsou vždy uvedeny v popisu každého frameworku, a dále zdroje [31][32] a web github.io.

### 10.1 Bootstrap

Bootstrap je jedním z nejrozšířenějších a nejkomplexnějších frameworků pro tvorbu webových stránek s responzivním zobrazením. Framework podporuje přístup mobile first se skokovým layoutem optimalizovaným pro několik základních šířek okna webového prohlížeče. Framework podporuje jak CSS preprocesor LESS tak i SASS.

Hodnocená verze: 3.2

Autor: Twitter Inc.

Domovská webová stránka: <http://getbootstrap.com/>

tab. 10-1 – Hodnocení Bootstrap

Kritérium	Hodnota	Body	Váha kritéria	Ohodnocení
Přístup k tvorbě responzivního webdesignu	MF	7	0,20	<b>1,40</b>
Typ layoutu	Skokový	10	0,10	<b>1,00</b>
Podpora CSS preprocesorů	LESS, SASS	10	0,25	<b>2,50</b>
CSS reset	N	10	0,10	<b>1,00</b>

Kritérium	Hodnota	Body	Váha kritéria	Ohodnocení
Základní šířka gridu	750 px, 970 px, 1 170 px			
Počet sloupců gridu	12			
Kompatibilita s IE	8+	8	0,15	<b>1,20</b>
Druh licence	MIT	10	0,20	<b>2,00</b>
<b>Součet</b>	-	<b>55</b>	-	<b>9,10</b>

## 10.2 Foundation

Foundation je dalším frameworkem, který podporuje přístup mobile first. Svým uživatelům nabízí CSS preprocesor SASS, čímž klade na vývojáře vyšší nároky na softwarovou výbavu, aby mohli s tímto CSS preprocesorem pracovat (kapitola 8.2.2). Kompatibilita s prohlížečem Internet Explorer nejde příliš dozadu a plně funkční web je dopřán uživatelům tohoto prohlížeče až od verze 9.

Hodnocená verze: 5.5

Autor: ZURB, Inc.

Domovská webová stránka: <http://foundation.zurb.com/>

tab. 10-2 – Hodnocení Foundation

Kritérium	Hodnota	Body	Váha kritéria	Ohodnocení
Přístup k tvorbě responzivního webdesignu	MF	7	0,20	<b>1,40</b>
Typ layoutu	Fluid-fixed	10	0,10	<b>1,00</b>
Podpora CSS preprocesorů	SASS	7	0,25	<b>1,75</b>
CSS reset	N	10	0,10	<b>1,00</b>
Základní šířka gridu	max-width 62.5 em			
Počet sloupců gridu	12			
Kompatibilita s IE	9+	6	0,15	<b>0,90</b>
Druh licence	MIT	10	0,20	<b>2,00</b>
<b>Součet</b>	-	<b>50</b>	-	<b>8,05</b>

### 10.3 Gumby

Gumby je první z vybraných frameworků, který se řídí přístupem desktop first. Má fluid-fixed layout s maximální šířkou 940 px se základním gridem o počtu 12 sloupců. Stejně jako Foundation podporuje CSS preprocesor SASS.

Hodnocená verze: 2.6

Autor: Primavision Inc. or Prema Toy Inc.

Domovská webová stránka: <http://gumbyframework.com/>

tab. 10-3 – Hodnocení Gumby

Kritérium	Hodnota	Body	Váha kritéria	Ohodnocení
Přístup k tvorbě responzivního webdesignu	DF	7	0,20	<b>1,40</b>
Typ layoutu	Fluid-fixed	10	0,10	<b>1,00</b>
Podpora CSS preprocesorů	SASS	7	0,25	<b>1,75</b>
CSS reset	EM	10	0,10	<b>1,00</b>
Základní šířka gridu	max-width 940 px	-	-	-
Počet sloupců gridu	12	-	-	-
Kompatibilita s IE	9+	6	0,15	<b>0,90</b>
Druh licence	MIT	10	0,20	<b>2,00</b>
<b>Součet</b>	-	<b>50</b>	-	<b>8,05</b>

### 10.4 Skeleton

Skeleton je neúspornějším frameworkem ze všech sledovaných. Jeho tvůrci se pyšní tím, že jeho zdrojový kód se rozkládá na pouhých 400 řádcích. To se však odráží na jeho knihovnách, které oproti ostatním frameworkům nejsou příliš rozsáhlé. Využívá přístup mobile first a má fluidní layout s fixní maximální šířkou 960 px. Jako jediný ze sledovaných frameworků nepodporuje žádný CSS preprocesor.

Hodnocená verze: 2.0.1

Autor: Dave Gamache

Domovská webová stránka: <http://getskeleton.com/>

tab. 10-4 – Hodnocení Skeleton

Kritérium	Hodnota	Body	Váha kritéria	Ohodnocení
Přístup k tvorbě responzivního webdesignu	MF	7	0,20	<b>1,40</b>
Typ layoutu	Fluid-fixed	10	0,10	<b>1,00</b>
Podpora CSS preprocesorů	žádný	0	0,25	<b>0,00</b>
CSS reset	N	10	0,10	<b>1,00</b>
Základní šířka gridu	max-width 960 px	-	-	-
Počet sloupců gridu	12	-	-	-
Kompatibilita s IE	7+	10	0,15	<b>1,50</b>
Druh licence	MIT	10	0,20	<b>2,00</b>
<b>Součet</b>	-	<b>47</b>	-	<b>6,90</b>

## 10.5 Less Framework

Less framework je posledním ze sledovaných. Jak již jeho název napovídá, svým uživatelům umožňuje využívat CSS preprocesor LESS. Je to také druhý zástupce desktop first. Jedná se podobně jako u Skeletonu o jednodušší a méně obsáhlý framework. Za jeho vývojem stojí Finský vývojář Joni Korpi.

tab. 10-5 – Hodnocení Less Framework

Kritérium	Hodnota	Body	Váha kritéria	Ohodnocení
Přístup k tvorbě responzivního webdesignu	DF	7	0,20	<b>1,40</b>
Typ layoutu	Fluid-fixed	10	0,10	<b>1,00</b>
Podpora CSS preprocesorů	LESS	7	0,25	<b>1,75</b>
CSS reset	vlastní	5	0,10	<b>0,50</b>
Základní šířka gridu	max-width 992 px	-	-	-
Počet sloupců gridu	10	-	-	-
Kompatibilita s IE	7+	10	0,15	<b>1,50</b>
Druh licence	MIT	10	0,20	<b>2,00</b>
<b>Součet</b>	-	<b>49</b>	-	<b>8,15</b>



Hodnocená verze: 4

Autor: Joni Korpi

Domovská webová stránka: <http://lessframework.com/>

## 11 Srovnání vybraných frameworků

Samotné srovnání frameworků obsahuje dvě tabulky. První tabulka tab. 11-1 obsahuje souhrn hodnot sledovaných frameworků v rámci jednotlivých kritérií. Tato tabulka slouží jako stručný přehled funkcí a vlastností vybraných frameworků.

Druhá tabulka tab. 11-2 zobrazuje bodové ohodnocení jednotlivých frameworků. Poslední řádek tabulky je součtový a udává celkový součet bodů každého frameworku.

tab. 11-1 – Srovnání vybraných frameworků (hodnoty)

	<b>Bootstrap</b>	<b>Foundation</b>	<b>Gumby</b>	<b>Skeleton</b>	<b>Less Framework</b>
Verze	3.2	5.5	2.6	2.0.1	4
Přístup k tvorbě RW	MF	MF	DF	MF	DF
Typ layoutu	Skokový	Fluid-fixed	Fluid-fixed	Fluid-fixed	Fluid-fixed
CSS preprocesory	LESS, SASS	SASS	SASS	žádný	LESS
CSS reset	N	N	EM	N	vlastní
Základní šířka gridu	750 px, 970 px, 1 170 px	max-width 62.5 em	max-width 940 px	max-width 960 px	max-width 992 px
Počet sloupců gridu	12	12	12	12	10
Kompatibilita IE	8+	9+	9+	7+	7+
Druh licence	MIT	MIT	MIT	MIT	MIT

tab. 11-2 – Srovnání vybraných frameworků (body)

	Váha	Bootstrap		Foundation		Gumby		Skeleton		Less Framework	
	-	Body	Vážené body	Body	Vážené body	Body	Vážené body	Body	Vážené body	Body	Vážené body
Přístup k tvorbě RW	0,2	7	1,40	7	1,40	7	1,40	7	1,40	7	1,40
Typ layoutu	0,1	10	1,00	10	1,00	10	1,00	10	1,00	10	1,00
CSS preprocesory	0,25	10	2,50	7	1,75	7	1,75	0	0,00	7	1,75
CSS reset	0,1	10	1,00	10	1,00	10	1,00	10	1,00	5	0,50
Základní šířka gridu	-	-	-	-	-	-	-	-	-	-	-
Počet sloupců gridu	-	-	-	-	-	-	-	-	-	-	-
Kompatibilita IE	0,15	8	1,20	6	0,90	6	0,90	10	1,50	10	1,50
Druh licence	0,2	10	2,00	10	2,00	10	2,00	10	2,00	10	2,00
<b>Součet</b>	-	<b>55</b>	<b>9,10</b>	<b>50</b>	<b>8,05</b>	<b>50</b>	<b>8,05</b>	<b>47</b>	<b>6,90</b>	<b>49</b>	<b>8,15</b>

## 11.1 Bootstrap

Framework Bootstrap má nejvyšší součet bodů i vážených bodů. Je tedy z vybraných frameworků nejlepší v rámci zadaných kritérií a jejich hodnocení. Bodový náskok získal hlavně v oblasti CSS preprocesorů, poněvadž umožňuje pracovat jak s LESS, tak i SASS. Z tohoto hlediska je dostupný jak začátečníkům, tak i pokročilým a profesionálům.

## 11.2 Less Framework

Na druhém místě je s náskokem 0,10 bodů Less Framework. I když oproti svým konkurentům ztrácí v oblasti CSS resetu, kdy nepoužívá žádný ze standardních souborů, ale používá svůj vlastní. Naopak získává díky své zpětné kompatibilitě s prohlížečem Internet Explorer, která sahá až k verzi 7.

### 11.3 Foundation a Gumby

Oba frameworky se umístily těsně za Less Frameworkem. Oba mají zcela totožné hodnocení. Jejich ztráta 0,10 bodu je téměř zanedbatelná, ale i tak v tomto srovnání zaujímají třetí pozici a to i přesto, že mají v součtu bodů o jeden více. Stanovené váhy však výsledný vážený součet bodů rozdělily v jejich neprospěch. Oproti Less Frameworku mají navrch co se CSS resetu týká, ale oba frameworky ztrácejí v kritériu kompatibility s prohlížečem Internet Explorer, kde nabízejí kompatibilní zobrazení pouze do verze 9.

### 11.4 Skeleton

Důraz na jednoduchost a úspornost zdrojového kódu vynesla Skeletonu poslední místo v tomto srovnání. Framework ztratil hodně bodů v kritériu CSS preprocesorů, protože nepodporuje žádný. A i když získal ještě několik bodů navíc díky kompatibilitě s Internetem Explorerem 7+, dělí ho od třetího frameworku 1,15 bodů a od prvního dokonce 2,20 bodů.

### 11.5 Souhrn

Z tabulek tab. 11-1 a tab. 11-2 je patrné, že vybrané frameworky se od sebe příliš neliší. Největší bodový rozdíl má kritérium CSS preprocesory, kde je rozdíl mezi nejvyšším a nejnižším počtem získaných bodů 2,50. Výsledné pořadí je tedy následující:

1. Bootstrap (9,10 bodů)
2. Less Framework (8,15 bodů)
3. Foundation, Gumby (8,05 bodů)
4. Skeleton (6,90 bodů)

## 12 Ukázka práce s Bootstrap framework

### 12.1 Instalace

Na domovské webové stránce frameworku <http://getbootstrap.com> jsou ke stažení základní tři balíčky.

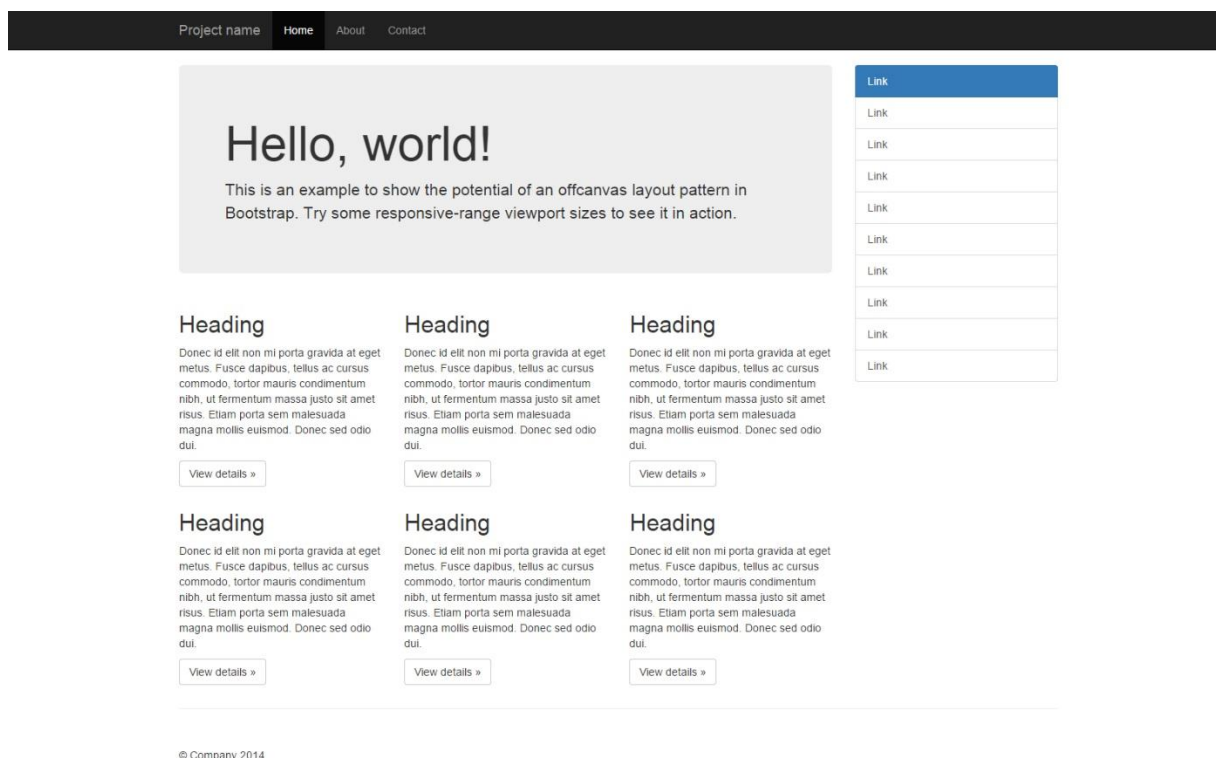
První balíček **Bootstrap** obsahuje pouze kompilované a minimalizované soubory s CSS, JavaScript a sadu písma.

Druhý balíček s názvem **Source code** obsahuje předchozí balíček Bootstrap a dále zdrojové soubory s LESS, dokumentaci, sadu ukázek a šablon a Grunt. Bootstrap používá Grunt (<http://gruntjs.com/>) ke kompilaci LESS souborů do formátu CSS. Aby bylo možné používat Grunt, je nutné nejdříve nainstalovat na vývojářský počítač Node.js (<https://nodejs.org/>), který obsahuje npm, neboli node packaged modules (<https://www.npmjs.com/>). Npm poté zajistí instalaci Grunt přes příkazovou řádku.

Třetí a poslední balíček s názvem **Sass** obsahuje podporu SASS.

### 12.2 Ukázková šablona

Pro ukázkou frameworku autor vybral Bootstrap šablonu s názvem Off-canvas (obr. 12-1).



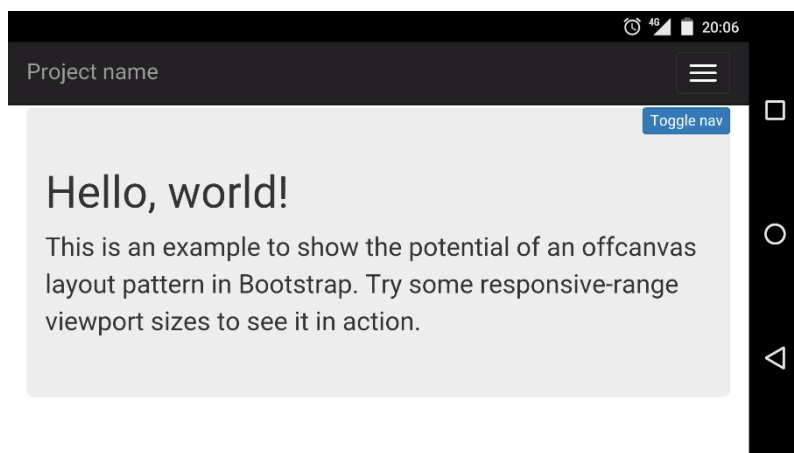
obr. 12-1 – Ukázková šablona Off-canvas

Autor tuto šablonu vybral, protože obsahuje dva druhy menu a texty v obsahu jsou rozčleněny do několika bloků. Na šabloně jsou proto názorně vidět změny, které přicházejí s různými šířkami okna prohlížeče.

### 12.3 Zalomování layoutu

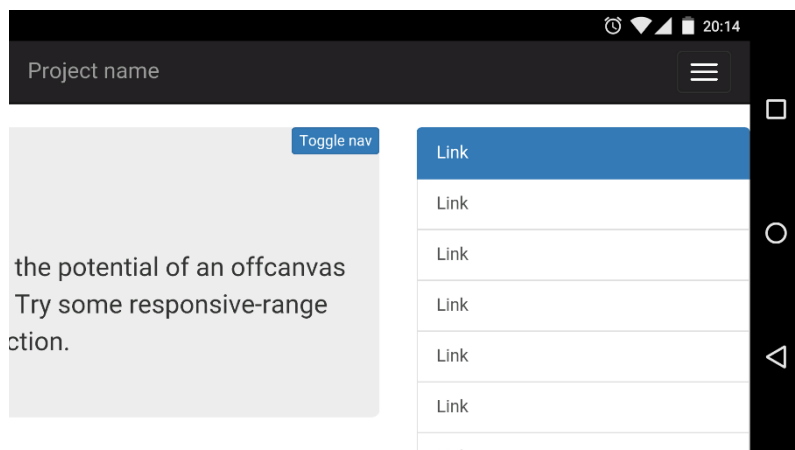
Layout ukázkové šablony je skokový. Základní grid je rozdělen do 12 stejně širokých sloupců. Každý sloupec má šířku  $8\frac{33}{100}\%$  a odsazení zprava a zleva 15 px.

Poněvadž se jedná o mobile first, je šířka gridu nastavována od nejnižší po nejvyšší šířku okna prohlížeče. První skok nastává, pokud okno prohlížeče dosáhne minimální šířky 768 px. Při šířce nižší než je 768 px je web plně optimalizován pro mobilní zařízení, viz snímek obrazovky mobilního zařízení na šířku na obr. 12-2. Na tomto obrázku je vidět klasické mobilní menu symbolizované třemi vodorovnými čárkami. Navigační lišta s nápisem „Project name“ a ikonka menu má fixní pozici, je tedy vždy vidět na vrchu stránky a to i při posunu po stránce směrem dolů.



obr. 12-2 – Off-canvas (mobilní zařízení)

Pravé menu, které je na obr. 12-1 zobrazeno vedle obsahu po pravé straně a obsahuje seznam odkazů (Link) je zde ukryto pod tlačítkem „Toggle nav“. Po kliknutí na toto tlačítko se obsah webové stránky posune částečně mimo obrazovku a zobrazí se zmíněné menu tak, jak je vidět na obr. 12-3. Tímto způsobem je dle autora velmi šikovně a hospodárně nakládáno s místem na malé ploše mobilních zařízení.



obr. 12-3 – Off-canvas, Toggle menu (mobilní zařízení)

Další skoky nastávají při šířce 992 px a 1 200 px. Skoku neboli změny šířky gridu je v Bootstrapu docíleno pomocí media queries tak, jak je znázorněno na ukázce kód 12-1, na které je zobrazena část zdrojového kódu ze souboru bootstrap.css, konkrétně řádky 1 585 až 1 599, které na základě šířky okna prohlížeče mění šířku gridu, tedy šířku třídy *container*.

```
@media (min-width: 768px) {
  .container {
    width: 750px;
  }
}

@media (min-width: 992px) {
  .container {
    width: 970px;
  }
}

@media (min-width: 1200px) {
  .container {
    width: 1170px;
  }
}
```

kód 12-1 – Bootstrap, media queries, šířka gridu

V Bootstrapu se pomocí media queries zdaleka nemění pouze šířka gridu, ale také například rámeček nebo odsazení navigační lišty viz kód 12-2.

```

@media (min-width: 768px) {
  .navbar {
    border-radius: 4px;
  }
}

@media (min-width: 768px) {
  .navbar-header {
    float: left;
  }
}

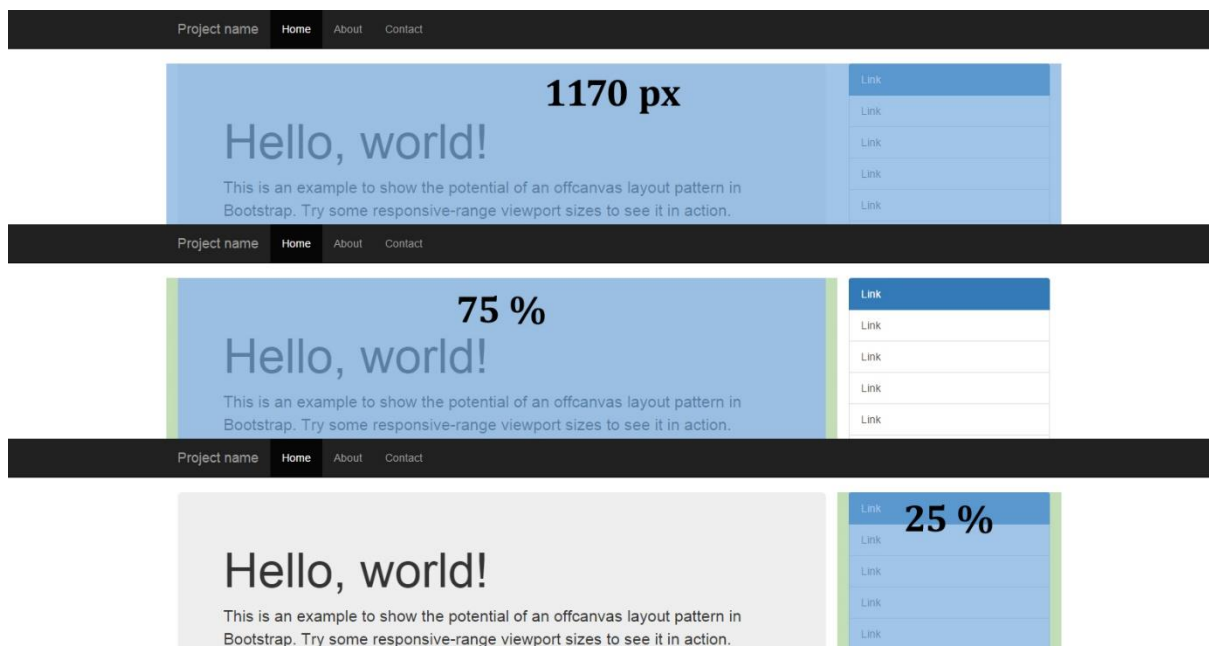
```

kód 12-2 – Bootstrap, media queries, navbar

Na obr. 12-4 je znázorněna šířka gridu a jednotlivých bloků obsahu (text a menu) při šířce okna prohlížeče vyšší než je 1 200 px, kdy je šířka gridu nastavena na 1 170 px (horní modrá plocha na obrázku).

Obsah na levé straně, který obsahuje text „Hello, world!“ má šířku 9 sloupců, tedy 75 % (prostřední modrá plocha na obrázku) a má odsazení zprava i zleva 15 px (prostřední zelené plochy na obrázku).

Pravé menu má šířku 3 sloupců, tedy 25 % (spodní modrá plocha na obrázku) a také odsazení zprava i zleva 15 px (spodní zelené plochy na obrázku). Tento model šířky obsahu je použit pro minimální šířky okna prohlížeče 1 170 px, 992 px a 768 px. Při skoku se mění pouze základní šířka gridu, ale poměr šířky obsahu zůstává stejný.



obr. 12-4 – Off-canvas, rozložení sloupců (desktop zařízení)

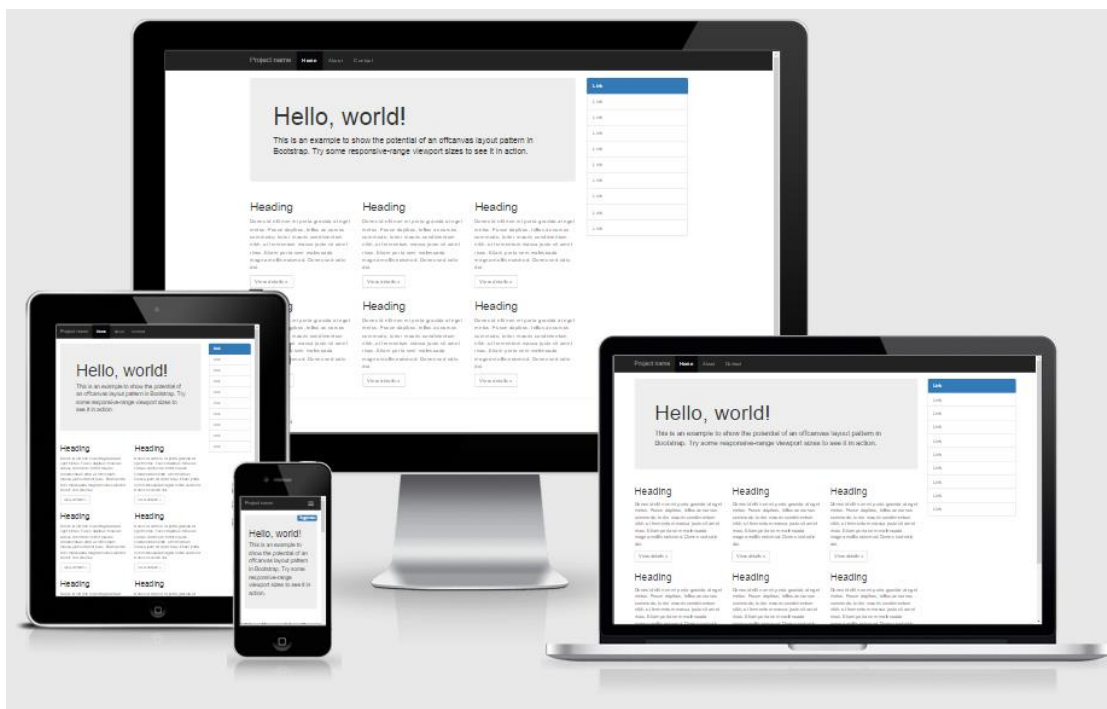


## 12.4 Srovnání zobrazení

Komplexně porovnat jednotlivá zobrazení je možné na obr. 12-5. Na něm je pomocí souhrnného výstupu z webové stránky [ami.responsivedesign.is](http://ami.responsivedesign.is) znázorněno zobrazení šablony Off-canvas ve všech třech základních šířkách gridu a při šířce nižší než 768 px.

Při minimální šířce okna prohlížeče 1 200 px a 922 px jsou textové bloky s nadpisem „Heading“ zobrazeny vždy tři vedle sebe. Při minimální šířce 768 px a nižší, tedy zobrazení určené pro zařízení typu tablet a mobilní telefon, jsou tyto textové bloky vedle sebe již pouze dva.

Kromě zobrazení s menší šířkou než 768 px se stále zachovává poměr šířky obsahu a menu 75 % a 25 %. Zmenšuje se pouze základní šířka gridu a ubývá místa po stranách webové stránky.



obr. 12-5 – Off-canvas, různá zobrazení [náhled vytvořen na [ami.responsivedesign.is](http://ami.responsivedesign.is)]

## 12.5 Předpřipravené styly

Velkou výhodou takovýchto frameworků jsou právě předpřipravené kaskádové styly CSS. Přednastavené styly velmi urychlují zprovoznění webové stránky, poněvadž stačí využívat předdefinované třídy kaskádových stylů na rozdíl od psaní vlastních a testování jejich kompatibility v prohlížečích.

Pro ukázkou předdefinovaných kaskádových stylů autor vybral třídy *btn* a *btn-default*. Obě třídy slouží ke stylování odkazů. Rozdíl mezi odkazem bez využití a s využitím těchto dvou tříd je znázorněn na obr. 12-6. Na levé straně je odkaz s textem „View details »“, který nevyužívá žádný styl, pouze dědí základní atributy jako barvu, písmo a velikost od elementů *A* a *BODY*. Na pravé straně je stejný odkaz, který ale používá třídy *btn* a *btn-default*.

## Heading

Donec id elit non mi porta gravida at eget metus. Fusce dapibus, tellus ac cursus commodo, tortor mauris condimentum nibh, ut fermentum massa justo sit amet risus. Etiam porta sem malesuada magna mollis euismod. Donec sed odio dui.

[View details »](#)

## Heading

Donec id elit non mi porta gravida at eget metus. Fusce dapibus, tellus ac cursus commodo, tortor mauris condimentum nibh, ut fermentum massa justo sit amet risus. Etiam porta sem malesuada magna mollis euismod. Donec sed odio dui.

View details »

obr. 12-6 – Off-canvas, styl odkazu

CSS kód vybrané třídy *btn-default* je znázorněn na ukázkce kód 12-3. Tato třída dodává odkazu barvu pozadí, ohraničení, hover efekt a další vlastnosti. Autor pro ilustraci znázorňuje na téže ukázkce zdrojového kódu zápis stejné třídy *btn-default* v CSS a také v LESS, který Bootstrap obsahuje v balíčku Source code. LESS zápis využívá hned několik svých výhod. Ve výňatku zdrojového kódu jsou použity vnořené definice, mixins *.button-variant()*, funkce *darken()* a proměnné *@btn-default-color*, *@btn-default-bg*, *@btn-default-border*, *@color*, *@background* a *@border*.

Zápis se syntaxí LESS je dle autora jednak mnohem přehlednější, ale také jeho znovupoužitelnost je na vysoké úrovni. Využití mixins umožňuje konfigurovat barvu textu, barvu pozadí a barvu rámečku tlačítka, přičemž se nijak neduplikuje zdrojový kód třídy *.button-variant*, která tyto vlastnosti přiděluje.

Zápis je sice složitější z pohledu množství použitých znaků, jako jsou závorky, *&* nebo *@*, to však vyvažuje „funkčnost“ těchto zástupných symbolů. Na levé straně ukázkky je klasický zápis v CSS, na pravé straně pak zápis v LESS.

```

.btn-default {
  color: #333;
  background-color: #fff;
  border-color: #ccc;
}
.btn-default:hover,
.btn-default:focus,
.btn-default.focus,
.btn-default.active,
.btn-default.activated,
.open > .dropdown-toggle.btn-default {
  color: #333;
  background-color: #e6e6e6;
  border-color: #adadad;
}
.btn-default.active,
.btn-default.activated,
.open > .dropdown-toggle.btn-default {
  background-image: none;
}
.btn-default.disabled,
.btn-default[disabled],
fieldset[disabled] .btn-default,
.btn-default.disabled:hover,
.btn-default[disabled]:hover,
fieldset[disabled] .btn-default:hover,
.btn-default.disabled:focus,
.btn-default[disabled]:focus,
fieldset[disabled] .btn-default:focus,
.btn-default.disabled.focus,
.btn-default[disabled].focus,
fieldset[disabled] .btn-default.focus,
.btn-default.disabled:active,
.btn-default[disabled]:active,
fieldset[disabled] .btn-default:active,
.btn-default.disabled.activated,
.btn-default[disabled].activated,
fieldset[disabled] .btn-default.activated {
  background-color: #fff;
  border-color: #ccc;
}
.btn-default .badge {
  color: #fff;
  background-color: #333;
}

```

```

.btn-default {
  .button-variant(@btn-default-color;
    @btn-default-bg;
    @btn-default-border);
}
.button-variant(@color; @background; @border) {
  color: @color;
  background-color: @background;
  border-color: @border;

  &:hover,
  &:focus,
  &.focus,
  &:active,
  &.activated,
  .open > .dropdown-toggle& {
    color: @color;
    background-color: darken(@background, 10%);
    border-color: darken(@border, 12%);
  }
  &:active,
  &.activated,
  .open > .dropdown-toggle& {
    background-image: none;
  }
  &.disabled,
  &[disabled],
  fieldset[disabled] & {
    &,
    &:hover,
    &:focus,
    &.focus,
    &:active,
    &.activated {
      background-color: @background;
      border-color: @border;
    }
  }
}
.badge {
  color: @background;
  background-color: @color;
}

```

kód 12-3 – Bootstrap, .btn-default, porovnání CSS a LESS

## 12.6 Souhrn

Frameworky, jakým je například Bootstrap, jsou dle autora vhodné zejména, pokud potřebujete rychle a s nízkými náklady zprovoznit webové stránky, u kterých není kladen vysoký nárok na originalitu designu a jejich design není příliš komplikovaný. Zdarma získáte rozšiřitelný základ designu webové stránky, která je navíc responzivní. Autor osobně využívá takovýchto frameworků v rámci školních projektů nebo v případě tzv. proof of concept, tedy ověření určitého předpokladu v začátcích projektu.

Takovéto frameworky mohou ale také sloužit jako předloha a inspirace při návrhu a kódování designu, neboť jejich zdrojový kód je propracovaný a prověřený tisíci uživateli.

## 13 Závěr

Autor v této diplomové práci konsoliduje znalosti nabyté z knih a článků, které zpracoval pro účely tohoto textu. Získané poznatky doplňuje o své osobní názory a zkušenosti získané praxí v tomto oboru.

Dle autora bylo dosaženo všech cílů definovaných v úvodní kapitole. Hlavním cílem bylo poskytnutí uceleného pohledu na tvorbu responzivního web designu. Toho bylo dosaženo definováním pojmu responzivní webdesign v kapitole 2, dále popisem způsobů dosažení mobilního webu včetně kladů a záporů jednotlivých způsobů v kapitole 3. Kapitoly 4, 5 a 6 se věnují třem základním pilířům responzivního webdesignu a postupně tak popisují druhy layoutu, možnosti flexible imagase a media queries. Kapitoly 7 a 8 jsou věnovány dalším technickým řešením, která se využívají při tvorbě responzivního webdesignu, přičemž kapitola 8 je celá věnována CSS preprocesorů, včetně srovnání dvou nejpoužívanějších preprocesorů.

Jednotlivé kapitoly obsahují pro ilustraci ukázky zdrojových kódů a ukázky v podobě snímků webových stránek, na které jsou tyto zdrojové kódy aplikovány.

Díličního cíle této diplomové práce bylo dosaženo v kapitolách 9, 10, 11 a 12, kde jsou nejdříve definována jednotlivá kritéria pro hodnocení frameworků s podporou responzivního webdesignu, poté jsou představeny vybrané frameworky, které byly hodnoceny dle stanovených kritérií a následně jsou jejich vlastnosti porovnány, bodově ohodnoceny a frameworky jsou seřazeny dle nejvyššího získaného počtu bodů. Framework, který získal nejvyšší počet bodů je podrobně představen v kapitole 12, která obsahuje i ukázky práce s tímto frameworkem.

Dle autora může tato diplomová práce poskytnout cenné informace a rady nejen začátečníkům a laikům v tomto oboru, ale také pokročilým webovým vývojářům, kteří zmíněné přístupy a technologie v současné době ještě neznají nebo nevyužívají, ale rádi by se přiučili novým trendům v oblasti webového designu a tvorby webových stránek obecně.

## Terminologický slovník

Termín	Zk.	Význam [zdroj]
Canvas		Plocha ohraňující pracovní plochu. [2]
Default		Výchozí, též standardní. [33]
Element		Elementy jsou různě velké kousky dokumentu psaného v jazyce HTML, XHTML, CSS apod. Jednotlivé elementy se skládají z tzv. tagů a v některých případech i z doplňujících atributů a svého obsahu. [34]
Desktop		Obecné označení PC - jedná se o osobní počítač, ne o server, mainframe nebo konzoli. [33]
Fair User Policy	FUP	Snížení propustnosti sítě po překročení objemu přenesených dat. [35]
Flash		Flash je technologie využívající se při tvorbě bannerů, menu, celých prezentací, vytváření on-line her či interaktivních aplikací. Technologie vyžaduje instalaci tzv. Flash plugin. [34]
Framework		Skupina knihoven, která usnadňuje řešení dané problematiky [29]
Front-end		Pojem frontend pochází z oblasti programování webových aplikací, kde slouží k označení části webu viditelné běžným návštěvníkům. Např. u internetového obchodu slouží jako frontend katalog zboží, nákupní košík a objednávkový formulář. [34]
Grid		Hrubá struktura, podle které jsou umístovány jednotlivé prvky webdesignu. [12]
Hypertext Transfer Protocol	HTTP	Protokol používaný při komunikaci mezi prohlížeči a webovskými servery. [18]
JavaScript	JS	JavaScript je objektově orientovaný programovací jazyk, využívaný při tvorbě webových stránek. [34]

Kaskádové styly	CSS	Kaskádové styly (z anglického Cascading Style Sheets) jsou moderním jazykem umožňujícím účinné formátování stránek psaných v jazycích HTML, XHTML či XML. [34]
Layout		Pojmem layout označují webdesignéři rozmístění základních prvků na stránce. Layout je tedy jakési schéma, jež říká, kde bude umístěn logotyp, hlavní navigace, drobečková navigace, formulář pro fulltextové vyhledávání a další obvyklé součásti stránky. [34]
Media queries		Modul CSS, který zjišťuje vlastnosti zařízení. [9]
Netbook		Označení Netbook zavedl Intel v souvislosti s uvedením své nové platformy Atom (kódové jméno Diamondville). Termín označuje malé, lehké (váha okolo 1 kg) a levné notebooky, určené primárně pro nenáročné použití - internet a jednodušší kancelářské aplikace. [33]
Retina displej		Retina displej je takový displej, u kterého nejsou uživatelé schopni pouhým okem rozlišit jednotlivé pixely. [36]
Tablet		Tablet je zkrácené označení pro "Tablet PC". Jedná se většinou o lehčí notebook s dotykovým displejem. Lze ovládat pomocí stylusu a používat jej např. jako poznámkový blok. Nejlehčí variantou jsou tablety, které neobsahují klávesnici, ale pouze několik funkčních kláves kolem displeje, ovládají se kompletně pomocí stylusu a veškerý HW tabletu je zabudován přímo v těle dotykové LCD obrazovky. [33]
Tag		Tag (anglicky štítek, cedulka) označuje části značek (elementů) v kódu HTML jazyka. Tag může být párový i nepárový. [34]
User-Agent	UA	Parametr hlavičky http [18]

## Seznam literatury

- [1] HAVELKA, Petr. *Responsivní webdesign*. Praha, 2013. Diplomová práce. Vysoká škola ekonomická v Praze.
- [2] MARCOTTE, Ethan. *Responsive web design*. New York: A Book Apart, 2011. ISBN 978-098-4442-577.
- [3] HATNIANKOVÁ, Zdeňka. *Flexibilní design webových stránek pro různá zobrazovací zařízení*. Praha, 2014. Bakalářská práce. Vysoká škola ekonomická v Praze.
- [4] KOLÍNEK, David. *Využití responzivního designu pro optimalizaci webových stránek pro mobilní zařízení*. Praha, 2014. Bakalářská práce. Vysoká škola ekonomická v Praze.
- [5] NĚMEC, Milan. *Responzivní webdesign*. Praha, 2014. Bakalářská práce. Vysoká škola ekonomická v Praze.
- [6] KELNAR, Martin. Responsivní Webdesign Komplexně. In: Responsivní Webdesign Komplexně [online]. 2012 [cit. 2015-02-11]. Dostupné z: <http://blog.martinkelmar.cz/responsivni-webdesign-komplexne/>
- [7] MARCOTTE, Ethan. Responsive Web Design. In: *Responsive Web Design* [online]. 2010 [cit. 2015-01-14]. Dostupné z: <http://alistapart.com/article/responsive-web-design>
- [8] MICHÁLEK, Martin. Mobile First v CSS. In: *Vzhůru dolů - Mobile First v CSS* [online]. 2013 [cit. 2015-02-11]. Dostupné z: <http://kratce.vzhurudolu.cz/post/42187934506/mobile-first-css>
- [9] W3SCHOOLS. CSS3 @media Rule. In: *CSS3 @media Rule* [online]. 2015 [cit. 2015-02-11]. Dostupné z: [http://www.w3schools.com/cssref/css3\\_pr\\_mediaquery.asp](http://www.w3schools.com/cssref/css3_pr_mediaquery.asp)
- [10] BÍLEK, Jan. Mobile-friendly: průvodce optimalizací mobilního webu. In: *Mobile-friendly: průvodce optimalizací mobilního webu* [online]. 2011 [cit. 2015-02-11]. Dostupné z: <http://www.zdrojak.cz/clanky/mobile-friendly-pruvodce-optimalizaci-mobilniho-webu/>
- [11] KNIGHT, Kayla. Fixed vs. Fluid vs. Elastic Layout: What's The Right One For You?. In: *Fixed vs. Fluid vs. Elastic Layout: What's The Right One For You?* [online]. 2009 [cit. 2015-02-22]. Dostupné z: <http://www.smashingmagazine.com/2009/06/02/fixed-vs-fluid-vs-elastic-layout-whats-the-right-one-for-you/>

- [12] AMRAN, Adam. Úvod do grid systémů. In: *Úvod do grid systémů - Meebio Blog* [online]. 2011 [cit. 2015-04-23]. Dostupné z: <http://blog.meebio.cz/clanek/158/uvod-do-grid-systemu/>
- [13] W3SCHOOLS. CSS Units. In: *CSS Units* [online]. 2015 [cit. 2015-02-11]. Dostupné z: [http://www.w3schools.com/cssref/css\\_units.asp](http://www.w3schools.com/cssref/css_units.asp)
- [14] KADLEC, Tim. *Responzivní design – profesionálně*. Brno: Zoner Press, 2014. ISBN 978-80-7413-280-3.
- [15] WROBLEWSKI, Luke. *Mobile first*. New York: A Book Apart, 2011. ISBN 978-193-7557-027.
- [16] MICHÁLEK, Martin. CSS3 Media Queries – podmíněné zobrazení pro média. In: *CSS3 Media Queries – podmíněné zobrazení pro média* [online]. 2013 [cit. 2015-03-21]. Dostupné z: <http://www.vzhurudolu.cz/prirucka/css3-media-queries>
- [17] CHEETAH. Detekce mobilních zařízení. In: *Detekce mobilních zařízení* [online]. 2011 [cit. 2015-02-21]. Dostupné z: <https://igloonet.cz/blog/detekce-mobilnich-zarizeni/>
- [18] KOSEK, Jiří. Základy protokolu HTTP. In: *Aplikace na Webu: 5. Základy protokolu HTTP* [online]. 2013 [cit. 2015-02-21]. Dostupné z: <http://www.kosek.cz/clanky/iweb/05.html>
- [19] MALÝ, Martin. Mobilizujeme web v HTML5. In: *Mobilizujeme web v HTML5* [online]. 2011 [cit. 2015-03-21]. Dostupné z: <http://www.zdrojak.cz/clanky/mobilizujeme-web-v-html5/>
- [20] CSS Reset. *CSS Reset* [online]. 2015 [cit. 2015-02-28]. Dostupné z: <http://www.cssreset.com>
- [21] RALSTON, Steve a Miluše POKORNÁ. Responzivní design: jak otestovat jeho funkčnost. In: *Responzivní design: jak otestovat jeho funkčnost* [online]. 2013 [cit. 2015-02-22]. Dostupné z: <http://interval.cz/clanky/responzivni-design-jak-otestovat-jeho-funkcnost/>
- [22] LESS: stejné CSS za méně peněz. In: MALÝ, Martin. *LESS: stejné CSS za méně peněz* [online]. 2010 [cit. 2015-02-11]. Dostupné z: <http://www.zdrojak.cz/clanky/less-stejne-css-za-mene-penez/>
- [23] LESS: Dynamický jazyk pro tvorbu stylesheetů. *LESS: Dynamický jazyk pro tvorbu stylesheetů* [online]. 2015 [cit. 2015-02-11]. Dostupné z: <http://www.lesscss.cz/>



- [24] SASS, LESS, STYLUS NEBO ČISTÉ CSS? (1). In: *SASS, LESS, STYLUS NEBO ČISTÉ CSS? (1)* [online]. 2012 [cit. 2015-02-11]. Dostupné z: <http://phpfashion.com/sass-less-stylus-nebo-ciste-css-1>
- [25] MICHÁLEK, Martin. Průvodce CSS preprocesory: který vybrat?. In: *Průvodce CSS preprocesory: který vybrat?* [online]. 2014 [cit. 2015-03-21]. Dostupné z: <http://www.vzhurudolu.cz/blog/15-css-preprocesory-4>
- [26] MICHÁLEK, Milan. CSS preprocesory: Nejspíš nějaký potřebujete a nejspíš je to LESS. In: *Vzhůru dolů - CSS preprocesory: Nejspíš nějaký potřebujete a...* [online]. 2013 [cit. 2015-02-11]. Dostupné z: <http://kratce.vzhurudolu.cz/post/56084086629/css-preprocesory>
- [27] Sass: Syntactically Awesome Style Sheets. *Sass: Syntactically Awesome Style Sheets* [online]. 2015 [cit. 2015-02-21]. Dostupné z: [sass-lang.com](http://sass-lang.com)
- [28] Softwarové licence - Slunečnice.cz - programy rychle a zadarmo. *Softwarové licence - Slunečnice.cz - programy rychle a zadarmo* [online]. 2015 [cit. 2015-02-22]. Dostupné z: <http://www.slunecnice.cz/licence/>
- [29] KNEŠL, Jiří. Frameworky vs DevStacky. In: *Frameworky vs DevStacky* [online]. 2015 [cit. 2015-04-05]. Dostupné z: <http://www.zdrojak.cz/clanky/frameworky-vs-devstacky/>
- [30] KOPERSKI, Adrian. Difference between Front-End Development and Back-End Development. In: *Difference between Front-End Development and Back-End Development* [online]. 2014 [cit. 2015-02-28]. Dostupné z: <http://mannigital.com/blog/2014/01/23/difference-between-front-end-and-back-end-development>
- [31] CSS Front-end Frameworks with comparison - By usabli.ca. *CSS Front-end Frameworks with comparison - By usabli.ca* [online]. 2015 [cit. 2015-02-22]. Dostupné z: <http://usabli.ca/github.io/front-end-frameworks/compare.html>
- [32] VERMILION. Responsive CSS Framework Comparison: Bootstrap, Foundation, Skeleton. *Vermilion* [online]. 2014 [cit. 2015-02-28]. Dostupné z: <http://responsive.vermilion.com>
- [33] Svět hardware. *Slovník základních pojmů / Svět hardware* [online]. 2015 [cit. 2015-04-23]. Dostupné z: <http://www.svethardware.cz/slovník/>
- [34] ADAPTIC, S. R. O. Internetový slovníček. *Internetový slovníček / Adaptic* [online]. 2015 [cit. 2015-04-23]. Dostupné z: <http://www.adaptic.cz/znalosti/slovnicek/>

- [35] VYLEŤAL, Martin. Není FUP jako FUP. In: *Není FUP jako FUP - Lupa.cz* [online]. 2006 [cit. 2015-04-23]. Dostupné z: <http://www.lupa.cz/clanky/neni-fup-jako-fup/>
- [36] HRUŠKA, Daniel. Co to vlastně je Retina displej. In: *Co to vlastně je Retina displej* [online]. 2012 [cit. 2015-04-23]. Dostupné z: <http://jablickar.cz/co-to-vlastne-je-retina-displej/>

## Seznam obrázků

obr. 2-1 – Schéma responzivního webdesignu [Zdroj: wikipedia.org] .....	13
obr. 2-2 – www.idnes.cz (náhled získán z deviceponsive.com).....	15
obr. 2-3 – t.idnes.cz (náhled získán z deviceponsive.com) .....	15
obr. 2-4 – mobil.idnes.cz (náhled získán z deviceponsive.com) .....	16
obr. 2-5 – Graf vývoje přístupu na web z mobilních zařízení [Statista 2015] .....	18
obr. 2-6 – Graf mobilních webových prohlížečů [Zdroj: gs.statcounter.com 2015] .....	19
obr. 3-1 – Mobile first [Zdroj: designshack.net].....	20
obr. 3-2 – Desktop first [Zdroj: designshack.net] .....	20
obr. 4-1 – Schéma gridu [12] .....	23
obr. 4-2 – Schéma fixního layoutu [Zdroj: smashingmagazine.com] .....	24
obr. 4-3 – Schéma fluidního layoutu [Zdroj: smashingmagazine.com].....	25
obr. 4-4 – Dotykové schéma [15] str. 73.....	26
obr. 5-1 – Fluidní obrázek 1 [2], str. 47 .....	28
obr. 5-2 – Fluidní obrázek 2 [2] str. 49.....	29
obr. 5-3 – Fluidní obrázek, zachování poměru stran [2] str. 50 .....	29
obr. 5-4 – Základní grafický návrh [2] str. 57 .....	30
obr. 5-5 – Relativní a absolutní šířka [2] str. 59 .....	31
obr. 7-1 – CSS reset [Zdroj: cssreset.com].....	36
obr. 7-2 – Snímek obrazovky webu vse.cz na zařízení LG Nexus 5 .....	37
obr. 7-3 – Náhled webu vse.cz ve vývojářském nástroji Opera Developer .....	38
obr. 7-4 – Náhled webu vse.cz [náhled vytvořen na ami.responsivedesign.is] .....	39
obr. 8-1 – LESS, darken(), porovnání výstupní barvy.....	42
obr. 9-1 – Graf desktopových webových prohlížečů [Zdroj: gs.statcounter.com].....	51
obr. 12-1 – Ukázková šablona Off-canvas.....	61
obr. 12-2 – Off-canvas (mobilní zařízení).....	62
obr. 12-3 – Off-canvas, Toggle menu (mobilní zařízení) .....	63
obr. 12-4 – Off-canvas, rozložení sloupců (desktop zařízení) .....	64
obr. 12-5 – Off-canvas, různá zobrazení [náhled vytvořen na ami.responsivedesign.is] ..	65
obr. 12-6 – Off-canvas, styl odkazu .....	66

## Seznam ukázek zdrojového kódu

kód 3-1 – Media queries, desktop first .....	21
kód 3-2 – Media queries, mobile first .....	21
kód 3-3 – HTML podmíněný komentář .....	22
kód 5-1 – HTML5 <PICTURE> [2] str. 64.....	27
kód 5-2 – HTML, fluidní obrázek [2] str. 46 .....	28
kód 5-3 – CSS, fluidní obrázek [2] str. 46 .....	28
kód 5-4 – CSS, fluidní obrázek, max-width [2] str. 48 .....	29
kód 5-5 – CSS, relativní šířka obrázku na pozadí [2] str. 60 .....	31
kód 6-1 – CSS, media queries .....	32
kód 6-2 – HTML, media queries .....	32
kód 6-3 – CSS, media queries, logický operátor .....	32
kód 6-4 – CSS, media queries, detekce vlastností displeje.....	33
kód 7-1 – CSS, detekce zařízení na straně klienta [Zdroj: igloonet.cz].....	34
kód 7-2 – User-Agent, detekce zařízení na straně serveru [Zdroj: igloonet.cz] .....	35
kód 7-3 – HTML, META VIEWPORT .....	35
kód 7-4 – CSS reset.....	37
kód 8-1 – Porovnání vnořené definice CSS a LESS.....	41
kód 8-2 – Porovnání proměnných CSS a LESS.....	41
kód 8-3 – LESS, matematické výrazy [Zdroj: lesscss.cz] .....	41
kód 8-4 – LESS, mixins v porovnání s CSS.....	42
kód 8-5 – LESS, práce s barvami [Zdroj: <a href="http://www.lesscss.cz/">http://www.lesscss.cz/</a> ].....	43
kód 8-6 – LESS, jmenné prostory [Zdroj: lesscss.cz] .....	43
kód 8-7 – Porovnání proměnných LESS a SASS (SCSS).....	45
kód 8-8 – Porovnání mixins LESS a SASS (SCSS) .....	46
kód 12-1 – Bootstrap, media queries, šířka gridu.....	63
kód 12-2 – Bootstrap, media queries, navbar.....	64
kód 12-3 – Bootstrap, .btn-default, porovnání CSS a LESS .....	67

## Seznam tabulek

tab. 10-1 – Hodnocení Bootstrap .....	53
tab. 10-2 – Hodnocení Foundation.....	54
tab. 10-3 – Hodnocení Gumby.....	55
tab. 10-4 – Hodnocení Skeleton.....	56
tab. 10-5 – Hodnocení Less Framework .....	56
tab. 11-1 – Srovnání vybraných frameworků (hodnoty) .....	58
tab. 11-2 – Srovnání vybraných frameworků (body) .....	59

## Příloha 1 – Tabulka pro hodnocení frameworků

Kritérium	Hodnota	Body	Váha kritéria	Ohodnocení
Přístup k tvorbě responzivního webdesignu		0 až 10	0,20	<b>0 až 2,00</b>
Typ layoutu		0 až 10	0,10	<b>0 až 1,00</b>
Podpora CSS preprocesorů		0 až 10	0,25	<b>0 až 2,50</b>
CSS reset		0 až 10	0,10	<b>0 až 1,00</b>
Základní šířka gridu		-	-	-
Počet sloupců gridu		-	-	-
Kompatibilita s IE		0 až 10	0,15	<b>0 až 1,50</b>
Druh licence		0 až 10	0,20	<b>0 až 2,00</b>
<b>Součet</b>	-	<b>0 až 60</b>	<b>1,00</b>	<b>0 až 10,00</b>