

**Univerzita Palackého v Olomouci**  
**Pedagogická fakulta**  
**Katedra technické a informační výchovy**



**Obor:**

Přírodopis se zaměřením na vzdělávání a informační výchova se zaměřením na vzdělávání

**Optimalizace webové stránky se zaměřením na  
mobilní zařízení**

**BAKALÁŘSKÁ PRÁCE**

Autor: Marek Straka

Vedoucí: Mgr. Jan Kubrický, Ph.D.

Olomouc 2015

## **PROHLÁŠENÍ:**

Prohlašuji, že jsem bakalářskou práci vypracoval zcela samostatně a že jsem uvedl veškeré použité zdroje a literaturu, ze kterých jsem čerpal.

V Olomouci dne 20. duben 2015

.....

Marek Straka

## **PODĚKOVÁNÍ**

Chtěl bych poděkovat vedoucímu mé bakalářské práce Mgr. Janu Kubrickému, Ph.D. za odborné vedení, cenné rady a vstřícnost při konzultacích. Dále děkuji všem, kteří mě při psaní bakalářské práce podporovali.

## **ABSTRAKT**

Práce se zaměřuje na problém optimalizace webu se zaměřením na mobilní zařízení a na problém nedostatku odborné literatury v českém jazyce vztahující se k tématu. Cílem je vysvětlit, co je optimalizace webu a přiblížit nástroje a metody jak jí efektivně dosáhnout. Stěžejní metodou je responzivní web design, jeho vývoj a základní pilíře pro tvorbu. Přínosem této práce je prezentace dvou nejběžnějších přístupů optimalizace včetně vytvoření příkladových webů a obohacení české literatury zabývající se optimalizací webu pro mobilní zařízení.

### **Klíčová slova:**

webová stránka, mobilní zařízení, responzivní web design, HTML5, CSS3, media queries

## **ABSTRACT**

This thesis focuses on the issue of web optimization for mobile phones and on the lack of technical literature in Czech which deals with this theme. The aim is to explain, what the optimization is and to give an idea of the tools and the methods for achieving optimization. The crucial method is the responsive web design, its development and foundations for the creation. The contribution of this thesis is a presentation about the two most common accesses of optimization including the creation of exemplary webs and the enrichment of the Czech literature dealing with the optimization of web for the mobile devices.

### **Keywords:**

website, mobile devices, responsive webdesign, HTML5, CSS3, media queries

# OBSAH

<b>ÚVOD.....</b>	<b>8</b>
<b>1 CO JE TO OPTIMALIZACE WEBOVÉ STRÁNKY – RESPONZIVITA .....</b>	<b>9</b>
1.1    Responzivní web design.....	9
1.2    Trend moderní doby .....	10
1.3    Mobilní zařízení .....	11
1.4    Historie.....	12
<b>2 ZÁKLADNÍ PILÍŘE RESPONZIVNÍHO WEB DESIGNU.....</b>	<b>15</b>
2.1    Plovoucí mřížka .....	15
2.2    Flexibilní obrázky .....	16
2.3    Media queries .....	16
2.4    Dynamický obsah.....	17
<b>3 NÁSTROJE PRO TVORBU RESPONZIVNÍ WEBOVÉ STRÁNKY .....</b>	<b>19</b>
3.1    HTML5 .....	19
3.2    CSS3.....	20
3.2.1    Media queries.....	21
3.3    JavaScript .....	21
3.4    Frontend framework.....	22
3.4.1    Frontend versus backend .....	22
3.4.2    Framework pro tvorbu webových stránek .....	22
<b>4 METODY TVORBY RESPOZIVNÍ WEBOVÉ STRÁNKY .....</b>	<b>26</b>
4.1    Detekce zařízení – přesměrování .....	26
4.1.1    Přesměrování na straně klienta .....	26
4.1.2    Přesměrování na straně serveru .....	27
4.2    Detekce šířky.....	28
4.2.1    Desktop-first .....	29

4.2.2	Mobile-first .....	30
4.2.3	Viewport .....	32
4.2.4	Breakpointy.....	33
4.3	Responzivní front-end framework .....	34
4.3.1	Proč použít framework.....	34
<b>5</b>	<b>TVORBA WEBOVÉ STRÁNKY OPTIMALIZOVANÉ PRO MOBILNÍ ZARÍZENÍ.....</b>	<b>36</b>
5.1	První model - jednoúčelový přístup .....	37
5.1.1	Desktopová verze.....	37
5.1.2	Mobilní verze.....	40
5.2	Druhý model - responzivní přístup .....	42
<b>6</b>	<b>ZÁVĚR.....</b>	<b>46</b>
	<b>SEZNAM ZDROJŮ .....</b>	<b>47</b>
	Literární zdroje .....	47
	Elektronické zdroje .....	49
	<b>SEZNAM OBRÁZKŮ .....</b>	<b>51</b>

## ÚVOD

V posledních pár letech dochází k rozvoji a velkému rozšíření mobilních zařízení a takzvaných chytrých zařízení s přístupem k internetu. Od mobilních telefonů s malými displeji a nízkým zobrazovacím rozlišením, přes výkonné chytré telefony (smartphony) až po multimediální tablety s vysokým HD rozlišením. Z tohoto důvodu, je v současné době pro tvůrce webových stránek téma číslo jedna optimalizace se zaměřením na mobilní zařízení. Je nezbytné vytvářet webové stránky moderní, přizpůsobivé (responzivní), čili takové webové stránky, které jsou univerzální k různým požadavkům. Zároveň je třeba definitivně upustit od vytváření statických webových stránek, dělaných na míru určitým zobrazovacím zařízením (Kadlec 2014).

Hlavním důvodem proč je v dnešní době tak důležité, aby byly webové stránky optimalizované pro mobilní zařízení, je různá úhlopříčka a zobrazovací rozlišení jejich displejů. Dalším důvodem je bezpochyby také odlišný výkon, kterým mobilní zařízení disponují, a tudíž je třeba tomu přizpůsobit i náročnost webové stránky na přenos a zpracování dat. V neposlední řadě se webová stránka zobrazená na mobilních zařízeních může lišit také svými vlastnostmi a svou podstatou. Také samotní návštěvníci mají odlišná očekávání od odlišných přístupů k jedné a té stejné webové stránce (Jahoda 2013).

Metod jak vytvořit webovou stránku optimalizovanou pro mobilní zařízení je více. V současné době nejmodernější a nejuniverzálnější metodou je responzivní web design. Ten zabezpečí správné zobrazení webové stránky nejen na mobilních zařízeních ale také na velkých obrazovkách smart TV či reklamních panelů. Jelikož se jedná o moderní způsob tvorby webových stránek, je třeba také používat moderní vývojářské nástroje. Různé metody využívají různé nástroje a vznikají tak různé přístupy jak docílit optimalizace webových stránek. Realizaci ovlivňují nejrůznější faktory, jako například: funkční podstata a s ní související složitost a náročnost, cílová skupina uživatelů a tak dále. Právě odlišné přístupy nám dávají možnost volby. Pokud se někdo rozhodne vytvořit si webovou stránku optimalizovanou pro mobilní zařízení, nebo již je majitelem neoptimalizované webové stránky a chtěl by ji rozšířit či nahradit webovou stránkou plně optimalizovanou pro veškerá zobrazovací zařízení, je pro něj důležité znát jak metody, jejich výhody a nevýhody, tak vhodné nástroje k dosažení tohoto cíle.



# 1 CO JE TO OPTIMALIZACE WEBOVÉ STRÁNKY – RESPONZIVITA

Optimalizace webové stránky pro mobilní zařízení je souhrn specifických přístupů, metod a postupů, pomocí nichž lze vytvořit takovou webovou stránku, která se optimálně zobrazí nejen na desktopových počítačích, ale také na mobilních zařízeních. Optimalizace v praxi znamená přizpůsobení se, v tomto případě se jedná o přizpůsobení webové stránky mobilním zařízením a jejich požadavkům. Nejvyšším stupněm optimalizace zobrazení webových stránek je v současné době responzivní web design.

## 1.1 Responzivní web design

Responzivní (adaptabilní) design je odborný termín, který vznikl odvozením od speciálních vlastností a reakcí grafického prostředí webových prohlížečů. Responzivní design je soubor metod a nástrojů, pomocí nichž se vytváří webová stránka. Takto vytvořená webová stránka nabývá vlastností, díky nimž se návštěvníkům zobrazí v plné kvalitě a správném layoutu<sup>1</sup> a to v jakémkoliv zařízení s různými webovými prohlížeči a také bez ohledu na velikost a rozlišení displeje.

Pojem responzivní design poprvé použil Ethan Marcotte roku 2010. Zmínil se o něm ve svém příspěvku, publikovaném magazínem A List Apart. Od té doby se responzivní design stával čím dál tím více nepostradatelnou technikou při vytváření webových stránek. Díky tomuto novému responzivní pohledu ve světě webdesignu se webové prostředí hodně změnilo a přizpůsobilo. Přizpůsobující se layout responzivní webové stránky je zajišťován takzvanou plovoucí mřížkou s plovoucím obsahem (základ tvoří text a obrázky) a modulem media queries<sup>2</sup>, který spadá pod kaskádové styly CSS3. Responzivní design je charakteristický tím, že šířky jednotlivých elementů nastavuje plovoucí, neboli pohyblivé a upouští od nastavování přesných rozměrů. V praxi to znamená, že se upouští od nastavování šířky v pixelech a stále více se používají procentuální poměry viewportu<sup>3</sup> webového prohlížeče (Fielding 2014).

---

<sup>1</sup> Prostorové schéma estetického rozložení prvků návrhu (Ambrose 2011).

<sup>2</sup> Modul povolující aplikovat různé styly na základě vlastností zařízení (Kadlec 2014).

<sup>3</sup> Okno webového prohlížeče, ve kterém se zobrazuje webová stránka (Kadlec 2014).

## 1.2 Trend moderní doby

Koncem minulého století pronikl internet do mobilních telefonů. Zatím však jen jako prostý text.

Velkým mezníkem ve světě web designu bylo v roce 2007 uvedení prvního iPhoneu na trh. Byl to jeden z prvních smartphonů a pro svět internetu a web designu to bylo nové místo kam proniknout. Do té doby byly webové stránky určeny jen pro desktopové monitory, které disponovaly jen malou škálou velikostí a zobrazovacích rozlišení. S iPhoneem vznikla nová potřeba web designu, vytvářet webové stránky pro zcela nový typ displeje. Prvním řešením tohoto problému bylo vytváření webových stránek odlišných od těch pro desktopové monitory. Avšak s postupem doby se začaly objevovat další typy smartphonů, později tablety a jiná zařízení, s nespočtem variací velikostí, což způsobilo značný problém. Nebylo možné vytvářet odlišné webové stránky pro všechny typy displejů. Novým řešením bylo vytvořit takový web design, který by se správně zobrazoval na všech typech displejů a přizpůsoboval se automaticky i těm nově vzniklým. Tímto řešením je designerská metoda responzivní web design, kterou prezentoval Ethan Marcotte v roce 2010 (Peterson 2014).

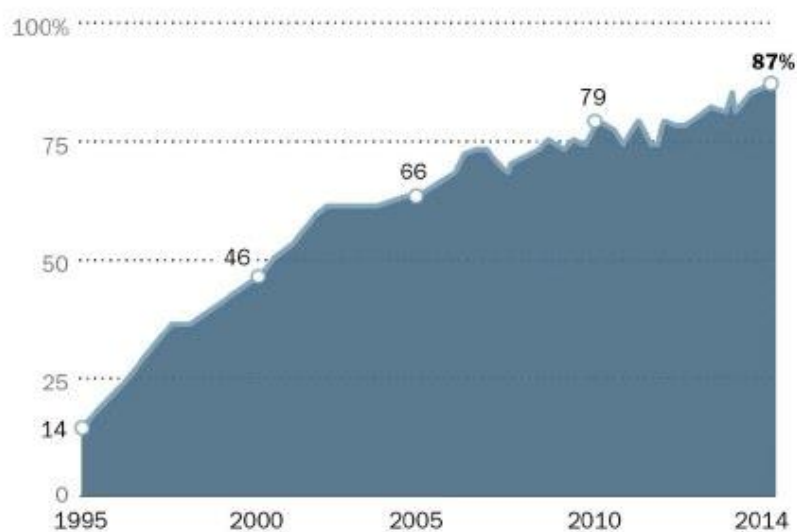
Responzivní web design se pořád zdokonaluje. Se vznikem nových technologií se přizpůsobuje nejen velikosti a rozlišení displejů, ale také dotykovým displejům a pracovnímu výkonu celého zařízení (Peterson 2014).

Nejen že trend pokračuje stejným směrem, ale dokonce se vyvíjí a je neustále rychlejší. Před dvaceti lety to začalo prudkým rozšířením přístupu uživatelů na Internet (Obrázek 1). V posledních deseti letech se trend hodně měnil a vyvíjel a více než samotný přístup na Internet se začaly řešit možnosti, jak se k Internetu připojit. V roce 2014 pak průzkumy ukázaly, že v Americe vlastní smartphon umožňující se připojit k internetu 58 % dospělých lidí. Tablet pak vlastní 35 % Američanů (Peterson 2014).

---

## Internet use, 1995-2014

% of American adults who use the internet, over time



Source: Pew Research Center surveys, 1995-2014.

PEW RESEARCH CENTER

---

Obrázek 1: Výsledky průzkumu vývoje počtu uživatelů Internetu 1995-2014

### 1.3 Mobilní zařízení

Mobilní nebo také přenosné zařízení je elektronický přístroj charakteristický tím, že jeho zdrojem energie je nějaký druh baterie. Velmi často obsahuje nějaký typ přijímací, či vysílací antény, pomocí které přístroj komunikuje s ostatními zařízeními (wifi, rádio, mobilní síť, satelit a jiné). Různé přístroje bývají osazeny různými komponenty podle toho, jaké funkce se od nich očekávají. Pro ovládání přístroje se dříve používaly číslíkové klávesnice, dnes se používají QWERTY klávesnice, intuitivní tlačítka, dotykové displeje a kamerové senzory pro snímání pohybu a gest. Dříve mobilní zařízení disponovaly jednoduchými, pouze řádkovými, monochromatickými displeji, dnes jsou běžné LCD displeje s HD rozlišením a nespočtem typů a velikostí. Některé jsou zaměřené na určitou činnost, jiné jsou multifunkční a dokáží plně nahradit desktopovou počítačovou stránici (Questia 2008, Questia 2011).

Typy mobilních zařízení:

- mobilní telefon (smartphone)
- čtečky elektronických knih
- PDA (personal digital assistant)
- tablet
- notebook, netbook, smartbook
- GPS navigace
- multimediální přehrávače
- herní konzole

## 1.4 Historie

### Design s pevnou šířkou

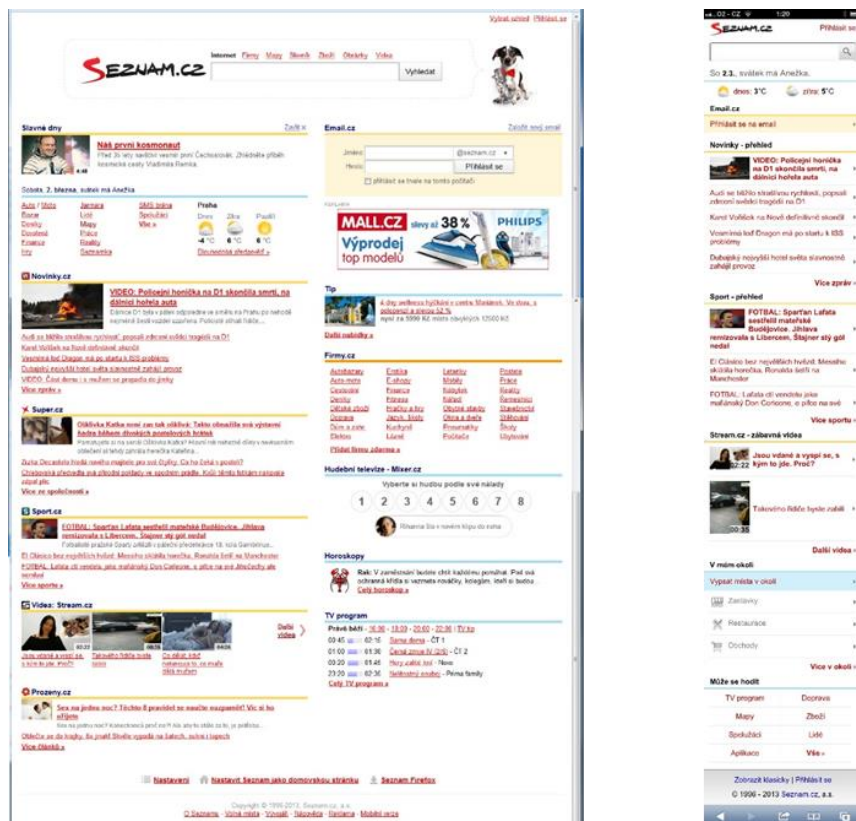
Při tvorbě layoutu tímto stylem se počítá s určitou velikostí displeje, předpokládá se jen velmi malá variabilita. Jednotlivým částem layoutu se nastavuje pevná šířka v pixelech. Je to úplně první způsob designování webových stránek a používá se až do dnes. Na přelomu roku 2000 se stalo nejrozšířenějším rozlišení monitorů s šířkou 1024 pixelů. Pořád však existovaly užší nebo širší monitory, s nimiž bylo při designování potřeba počítat. Řešením bylo nastavit webové stránce o trochu menší šířku, tak aby se zobrazovala i v užších monitorech. V širších monitorech se pak stránka zobrazovala uprostřed a po stranách byly dva prázdné pásy (Peterson 2014).

### Éra mobilního web designu

V polovině devadesátých let minulého století se objevily první telefony s internetem. Do roku 2000 dokázaly zpracovávat jen základní HTML kód a zobrazovat jen jednoduchý text v odstínech šedi. Na přelomu tisíciletí došlo k velkým změnám, smartphony dostaly první barevné displeje (průkopníkem na trhu byla finská Noika) a mobilní prohlížeče dokázaly pomocí CSS2 a JavaScriptu zobrazovat běžné webové stránky (Peterson 2014, Vokáč 2015).

Velký průlom přinesl iPhone od americké společnosti Apple. Jeho inovace odstartovaly novou éru mobilního webového designu. iPhone jako první dokázal zobrazit webové stránky plnohodnotně a stejně jako desktopové počítače. Vzhledem k malému rozlišení displeje se stránky zobrazovaly ve zmenšené podobě a uživatel si mohl přiblížit jejich část. Novinkou byl také multitouchový displej, díky němuž bylo

přiblížení pomocí gesta možné. Avšak zmenšené zobrazení nebylo optimální, a tak webový vývojáři a designeři přišli s nápadem vytvořit dvě oddělené webové stránky. Jednu budou zobrazovat uživatelé desktopových počítačů a druhá, přizpůsobená menšímu rozlišení, se bude zobrazovat na iPhoneu po přesměrování na „mobilní“ m subdoménu (desktop: <http://seznam.cz>; mobile: <http://m.seznam.cz>), (Obrázek 2) (Peterson 2014).



Obrázek 2: Desktopová verze (vlevo) a mobilní verze webu (vpravo) (<http://seznam.cz>)

Těmto mobilní zástupným webům se říkalo iPhone weby, protože vznikly díky iPhone a pro iPhone, který byl v té době jediným takovým zařízením na trhu. To se ale co nevidět změnilo a začaly se objevovat smartphony jiných výrobců, které však neměly stejně velký displej jako iPhone. Nově vzniklé přístroje konkurence měly displeje buď užší, nebo širší než iPhone a díky tomu vznikl problém s optimalizací mobilního zástupného webu, který nevyhovoval svou velikostí a nezobrazoval se správně. Navíc v roce 2010 společnost Apple přišla se svým prvním tabletem s názvem iPad. Po této události se opět přidává další typ zařízení s úplně jinou velikostí displeje a potřebou zobrazovat webové stránky. Metoda zástupných mobilních webů je již nedostačující (Peterson 2014).

Weboví designeři se již v roce 2000 zabývali konceptem plovoucího layoutu, tehdy však z důvodů obav nad špatnou kontrolou šířky jednotlivých prvků a ve výsledku celého layoutu od tohoto konceptu upustili. V roce 2010 se k dříve zavržené myšlence vrátili a začali se užitím procentuálních poměrů opět zabývat. Samotná změna nastavení šířky v procentech na místo pixelů nebyla však dostačující. Bylo potřeba upravit i celé grafické schéma, zredukovat některé nadbytečné prvky, aby zbyl dostatek prostoru pro důležité části. O tyto úpravy layoutu se postará CSS modul zvaný media queries. Media queries byl již ve verzi CSS2, avšak jen s velmi omezenými možnostmi, až ve verzi CSS3 se media queries těší velkému využití v oblasti úpravy webových stránek. Velmi důležitou částí plovoucího layoutu je také přizpůsobitelnost velikosti obrázků a flexibilní velikost písma. Všechny tyto vývojářské prvky, každý sám o sobě nejsou ničím převratným, jsou buď v omezené míře, nebo samostatně používány už mnoho let. Až americký programátor, web designer a řečník Ethan Marcotte v roce 2010 přišel s myšlenkou spojit tyto prvky a využít jejich vlastností k vytvoření responzivního webového designu. V současné době se responzivní web design neustále mění a vyvíjí, protože se mění a vyvíjí i jeho nástroje (Sklar 2011, Peterson 2014).

## 2 ZÁKLADNÍ PILÍŘE RESPONZIVNÍHO WEB DESIGNU

Ethan Marcotte otec Responzivního web designu je také tvůrcem základních pilířů responzivního designu. Účelem těchto pilířů je jednoduše vytvořit takový obsah, který bude spustitelný na všech zařízeních. Cílem není přizpůsobit se zařízením, cílem již od počátku bylo přizpůsobit se uživateli, umožnit přístup bez jakýchkoli překážek, překonat rozdíly a vytvořit takové podmínky, které sjednotí veškerá zařízení a poskytnou všem uživatelům stejně podmínky a možnosti (Sharkie 2013).

Základní pilíře responzivního web designu jsou celkem čtyři:

### 2.1 Plovoucí mřížka

Plovoucí mřížka je prvním pilířem responzivního web designu. Myšlenka plovoucí mřížky vznikla v návaznosti na koncept plovoucího layoutu, který byl velmi aktuálním tématem v roce 2006, kdy se hodně diskutovalo o optimální šířce layoutu pro displeje s rozlišením  $1024 \times n$  pixelů (Sharkie 2013).

Mřížka jako taková je schematické rozložení jednotlivých částí webové stránky, je to v podstatě způsob, jak uspořádat layout webu. Základním částem mřížky se říká sloupce. Jednoduchost či složitost layoutu se odvíjí od počtu sloupců, na které je rozdělený. Nejjednodušší layouty jsou postaveny z jednoho, dvou nebo tří sloupců, které mohou být postaveny v jedné rovině. Složitější layouty mohou obsahovat devět, dvanáct nebo více sloupců. Rozvržení rozsáhlejších layoutů je složité. Sloupce zde mohou být situovány jak svisle, tak i vodorovně (Marcotte 2011, Sharkie 2013).

V minulosti bylo běžné vytvářet mřížku s fixními rozměry určenými hodnotami v pixelech. Podstata plovoucí mřížky spočívá v tom, že jednotlivým sloupcům se nastavuje relativní šířka. Podle počtu sloupců a jejich rozložení v mřížce celého layoutu se vypočítá poměrná šířka, která se pak jednotlivým sloupcům udává v procentech. Se změnou velikosti viewportu, která je ovlivněna velikostí a rozlišením zařízení se mění i poměrná velikost (šířka) jednotlivých sloupců. Dochází tak k žádoucí deformaci celého layoutu, což je cílem plovoucí mřížky (Marcotte 2011, Sharkie 2013).

## 2.2 Flexibilní obrázky

Druhým pilířem responzivního web designu jsou flexibilní obrázky. Základními vlastnostmi obrázku, v kontextu bitmapové grafiky, jsou rozměry, čili velikost v pixelech a kvalita, čili hustota pixelů nejčastěji udávaná na palec čtvereční. Obě vlastnosti představují pro použití obrázků v plovoucí mřížce velký problém (Sharkie 2013).

Jak již bylo řečeno, koncept základních pilířů responzivního web designu je zaměřen na uživatele a pro uživatele je grafická stránka webu, hned po funkčním obsahu, tím nejdůležitějším. Prostřednictvím grafiky s uživatelem komunikujeme, již volbou barvy pozadí či písma na uživatele působíme, a to buď pozitivně, nebo negativně. Flexibilní obrázek se musí zobrazit v potřebné velikosti, tak aby se přizpůsobil plovoucí mřížce a v potřebné kvalitě, aby byl pro uživatele dobře čitelný. Problém nastává při volbě velikosti a kvality obrázku. Obrázek velkých rozměrů s velmi dobrou kvalitou se bude dobře zobrazovat na velkých displejích s velkým rozlišením, ale na malých displejích mobilních zařízení bude jeho zobrazení dělat problém a v potaz také musíme vzít jeho datovou náročnost, která by měla být také optimální k možnostem uživatele. Pokud zvolíme obrázek menších rozměrů a kvality, bude ideální pro mobilní zařízení, problém však nastane u velkých displejů, například u desktopových monitorů nebo smart TV s HD rozlišením. Ideální řešení nabízí HTML5 a CSS3, jejichž prostřednictvím lze načítat různé zdroje obrázků pro displeje různých velikostí a rozlišení (Marcotte 2011, Sharkie 2013).

## 2.3 Media queries

V původním konceptu Ethana Marcotta stály pouze tři pilíře a posledním z nich je modul kaskádových stylů media queries. Ve spolupráci s nejnovější verzí HTML, již páté generace, je tento pilíř tím nejsilnějším a nejstabilnějším. Media queries ve velmi volném doslovném překladu znamená dotazy na médium, což velmi dobře vystihuje podstatu a funkci. Media queries pracuje na bázi jednoduchých podmínek a dotazů. Zjišťuje potřebné informace o webovém prohlížeči, či o samotném zařízení, ze kterého uživatel přistupuje na webovou stránku. Na základě těchto informací mění, vkládá nebo odebírá části nebo dokonce celé zdroje kaskádových stylů a mění tak layout webové stránky, tak aby bylo pro uživatele neustále optimální (Marcotte 2011, Sharkie 2013).



## 2.4 Dynamický obsah

Všeobecně stanovený poslední pilíř responzivního web designu je dynamický obsah. Je ze všech pilířů nejmladší a neustále se mění. Sdělení čili obsah webové stránky je tím nejdůležitějším. Ať už se jedná o webové prezentace s informacemi, multimediální portály nebo internetové obchody nabízející zboží, vždy je podstatou obsah (Sharkie 2013).

*“Content” is everything that a user of the site consumes when they interact with your service.*

*(Sharkie 2013)*

Ta úplně nejdůležitější část obsahu webové stránky je text. Při adaptaci webové stránky na mešní displeje s menším rozlišením zpravidla dochází ke změně nebo odebrání některých přebytečných částí. Většinou jde o zjednodušení grafických prvků a o redukci strukturálních částí. Co však vždy musí být zachováno, je text. Když opět do centra našeho uvažování postavíme uživatele, tak víme, že naší prioritou je, aby text webové stránky byl vždy dobře čitelný a přehledný. Velikost textu je proto potřeba nastavovat také relativně stejně jako je relativní velikost jeho obalu, čili sloupce v plovoucí mřížce. Relativní velikost textu se nastavuje pomocí hodnoty *em*, vůči ní stojí nejběžnější absolutní hodnota *px*. Defaultně je nastaveno, že  $1em = 16px$ . Výpočet správné relativní hodnoty velikosti textu pro určitý prvek zjistíme pomocí vzorce, který publikoval Ethan Marcotte v článku o plovoucí mřížce (Marcotte 2011, Sharkie 2013).

```
target ÷ context = result
```

```
24 ÷ 16 = 1.5
```

*(Marcotte 2011)*

*Target* vyjadřuje hodnotu velikosti textu určitého prvku, se kterým chceme pracovat a chceme u něj tuto hodnotu zjistit, je tedy naším cílem. Proměnná *context* nabývá hodnoty nadřazeného (rodičovského) prvku. Zpravidla to bývá hodnota velikosti textu nastavené v prvku *container*<sup>4</sup> nebo *body*. V praktické ukázce příkladu, která se nachází

---

<sup>4</sup> Základní prostorový prvek obalující celou webovou stránku (Powers 2012).

pod všeobecným vzorcem, potřebujeme nastavit pro cílový prvek velikost textu 24px. Nadřazený prvek (*container*) nemá velikost textu specifikovanou, počítáme tedy s výchozí hodnotou 16px. Matematickou operací podělíme hodnotu cílového prvku s hodnotou nadřazeného prvku a výsledkem je relativní hodnota velikosti textu pro cílový prvek v jednotkách *em* (Marcotte 2011, Sharkie 2013)

Mezi jednotlivými pilíři je velká provázanost. Dynamický obsah se částečně pojí i s flexibilními obrázky protože velká část obsahu webových stránek je proložena obrázky, například ilustrační foto produktu nabízeného v internetovém obchodu. Dynamický obsah je společně s flexibilními obrázky vázaný na plovoucí mřížku, která udává tvar layoutu. S layoutem tedy s plovoucí mřížkou je také propojena ta část flexibilních obrázků, která tvoří grafiku webové stránky. Důležité je, aby se všechny části sobě navzájem přizpůsobovaly. Přizpůsobování a správné zpracování celého systému za všech okolností zajišťuje třetí pilíř media queries (Sharkie 2013).

## 3 NÁSTROJE PRO TVORBU RESPONZIVNÍ WEBOVÉ STRÁNKY

### 3.1 HTML5

**HyperText Markup Language** je v překladu do češtiny hypertextový značkovací jazyk. Je základním stavebním kamenem pro tvorbu webových stránek. Poslední pátá specifikace HTML byla po deseti letech vývoje koncem roku 2014 konečně dokončena. Oproti předchozím verzím obsahuje více inovací získaných na základě výzkumů vývojářských praktik. Charakteristika inovací je zaměřená na uživatele a na efektivnější spolupráci a komunikaci mezi systémy (Duckett 2008, W3C 2015).

Velké změny proběhly na základě vývojářské platformy, byly definovány nové tagy a atributy k popisu obsahu a také struktury. Zápisy tagů se zkracují pro rychlejší použití a pro přehlednost zdrojového kódu. Rozsáhle inovace proběhly i v oblasti multimédií. V předchozí verzi byla tato oblast velkou slabinou. Problém s multimédií byl řešen zásuvnými moduly<sup>5</sup>, které však v mobilních prohlížečích a zvláště v těch od společnosti Apple nepracovaly správně. S novou specifikací lze multimediální soubory do webových stránek vkládat přímo a uživatel by již neměl mít problém s jejich zobrazením. V rámci kompatibility mezi různými zařízeními je to velký krok kupředu. Novinkou v nové verzi HTML je možnost vytvářet aplikace, které lze spouštět a pracovat s nimi také v režimu offline. Dalším inovativním prvkem je canvas, jehož funkcí je interpretovat vektorovou grafiku. Tato funkce se využívá při vykreslování grafů, animací a her a také k úpravě obrázků. Největším přínosem v oblasti responzivity je bezpochyby inovace prostředí. HTML5 zahrnuje spoustu nových prvků, které zobrazují a upravují obsah webových stránek a oprostřují tak webové stránky od nutnosti využití zásuvných modulů. Toto je velká výhoda při optimalizaci webových stránek pro mobilní zařízení a pro speciální multimediální zařízení s možností připojení k internetu, jako jsou smart TV či digitální informační panely (Hogan 2011, MacDonald 2013).

---

<sup>5</sup> Zásuvný modul (anglicky plugin) je sekundární software, který rozšiřuje primární software, většinou se jedná o multimediální podporu

## 3.2 CSS3

**Cascading Style Sheets** v českém překladu kaskádové styly, je v informatice označení pro jazyk popisující, jak se zobrazí elementy značkovacích jazyků HTML, XHTML a XML. Poslední třetí verze kaskádových stylů je ve vývoji již od roku 2005 a její dokončení se odhaduje v průběhu roku 2015. Verze CSS3 byla vyvíjena společně s poslední verzí HTML5, jsou spolu propojeny a společně přináší do prostředí vývoje webových stránek spoustu inovací (Duckett 2008, Powers 2012).

Třetí verze kaskádových stylů je stěžejní pro spolupráci s jazykem HTML5, je podstatou pro budování hlubšího a bohatšího prostředí. Nové prvky, které CSS3 přináší, velmi usnadňují práci webovým designérům. Grafické prvky nezbytné pro design moderní webové stránky jako jsou například stínování či zaoblování blokových prvků nebo stínování textu. Velkou novinkou je nastavování barvy u prvků, které bylo obohaceno o vlastnost *A* (*alfa kanál*<sup>6</sup>) ke stávající definici základních barevných kanálů RGB. Výsledný zápis se skládá ze čtyř hodnot RGBA, přičemž poslední hodnota udává průhlednost prvku. Další podstatnou inovací v oblasti grafiky je nová vlastnost transformace prvku. Transformace se rozlišuje na tři typy, prvním typem je prostý posun po ose x a y, dalším typem je rotace a poslední typ umožňuje úpravu velikosti prvku. Všechny zmíněné grafické úpravy bylo dříve nutné provádět vkládáním vhodných obrázků, které je imitovaly. Toto designerské řešení se však vždy potýkalo s velkými problémy kompatibility s mobilními zařízeními a se zobrazováním na menších displejích. Inovace CSS3 jsou základním pilířem pro responzivní web design. Abychom docílili dobré optimalizace pro veškerá zařízení, je velmi důležité mít optimální zdrojový kód i pro všechny typy prohlížečů. Ačkoliv je vývoj CSS3 téměř u konce, je na něj stále pohlíženo, jako na pracovní návrh, a tak je třeba některé jeho funkce označit specifickými prefixy<sup>7</sup> pro určení kompatibility s daným prohlížečem (*-moz-* Firefox, *-o-* Opera, *-ms-* Internet Explorer, *-webkit-* Chrome a Safari). Následující obrázek prezentuje podporu CSS3 v některých verzích vybraných webových prohlížečů (Obrázek 3) (Powers 2012, Fielding 2014).

---

<sup>6</sup> Vlastnost pixelů bitmapy vyjádřit průhlednost, to znamená zobrazit vrstvu pod bitmapou (Snider 2013)

<sup>7</sup> Předpona umožňující použití prvku, který nebyl implementován do webového prohlížeče (Fielding 2014)



Obrázek 3: Vyjádření podpory CSS3 ve webových prohlížečích

### 3.2.1 Media queries

Modul media queries je zahrnutý v poslední, třetí verzi kaskádových stylů. Media queries byly známé již od verze CSS2 ale jejich funkce byla omezena jen na funkci print. Tato funkce upravuje vzhled stránky, tak aby byl vhodný pro tisk. V CSS3 je modul media queries nejvíce revolučním prvkem, obsahuje celou řadu nových funkcí. Pomocí media queries lze určit vlastnosti zařízení, jako jsou: výška, šířka nebo rozlišení obrazovky, výška a šířka okna webového prohlížeče nebo orientace zařízení (portrét nebo krajina). Na základě získaných informací o zařízení se aplikují odpovídající kaskádové styly, které přizpůsobí layout webové stránky. Nové možnosti dělají z media queries nepostradatelný nástroj pro tvorbu responzivních webů. Umožňují vytvořit webové stránky nezávislé na typu zařízení. Takové stránky pak může uživatel navštívit bez toho, aby byl omezen možnostmi svého zařízení, a to jeho typem, velikostí, rozlišením ani výkonem (Gasston 2011, Hogan 2011, Lunn 2013).

### 3.3 JavaScript

Responzivní webové stránky se vytváří převážně za pomoci nástrojů HTML a CSS, JavaScript se používá jen výjimečně. Do HTML kódu webové stránky se přidávají kusy kódu JavaScriptu za účelem udržení optimalizace i ve starších webových prohlížečích. Pro testování optimalizace a podpory HTML5 a CSS3 pro nejrůznější prohlížeče existuje JavaScriptová knihovna Modernizr. Modernizr funguje tak, že na základě podmínky otestuje, zda webový prohlížeč podporuje například canvas. Pokud ano,

prvek spustí, pokud však zjistí, že ne, nahradí jej buď jiným prvkem, nebo zobrazí hlášku, že prvek není podporován tímto prohlížečem. Je možné tímto způsobem nahrazovat jak HTML kód, tak i CSS. Dalším využitím JavaScriptu je detekce mobilních zařízení přes webový prohlížeč. Detekce zařízení se používají převážně u tvorby responzivního webu metodou zástupných mobilních webů (Watson 2012, Peterson 2014).

## 3.4 Frontend framework

### 3.4.1 Frontend versus backend

Strukturu webových stránek a všeobecně jakéhokoli softwaru můžeme rozdělit, na základě úhlu pohledu, na dvě části: frontend a backend. Jedná se o pojmy užívané v oblasti softwarového inženýrství. Obě části softwaru jsou propojeny a jsou na sobě závislé (Treehouse 2012).

Z pohledu uživatele, který navštívil webovou stránku, je frontend to, co vidí v okně webového prohlížeče. Frontendem na příklad internetového obchodu je seznam nabízených produktů, nákupní košík, případně účet uživatele nebo objednávkový formulář a celkový layout. Souhrnně a jednoduše, **frontend je uživatelské prostředí** webové stránky (Treehouse 2012, Adaptic 2015).

Opakem frontendu je backend. Jedná se tedy o tu část webové stránky, která stojí v pozadí. Jsou to zdroje a data načítaná například z databází ale také samotný CMS<sup>8</sup> (Content Management Systém), který může upravovat jak data, tak i strukturu celé webové stránky. Backend internetového obchodu tedy bude databáze produktů a CMS systém sloužící k úpravě frontendu. Stručně **backend je technická podpora a datový zdroj** webové stránky (Treehouse 2012, Adaptic 2015).

Koncept backend – frontend je v současné době nejčastějším schématem užívaným při vývoji webových aplikací s dynamickým obsahem (Treehouse 2012, Adaptic 2015).

### 3.4.2 Framework pro tvorbu webových stránek

Ve webovém vývojářství je framework termín, který označuje balíček složek a souborů s určitou strukturou, které obsahují standardizované kódy HTML, CSS a JavaScriptu a mohou být využity jako podpora při vytváření webů nebo pomocí nich lze vytvořit

---

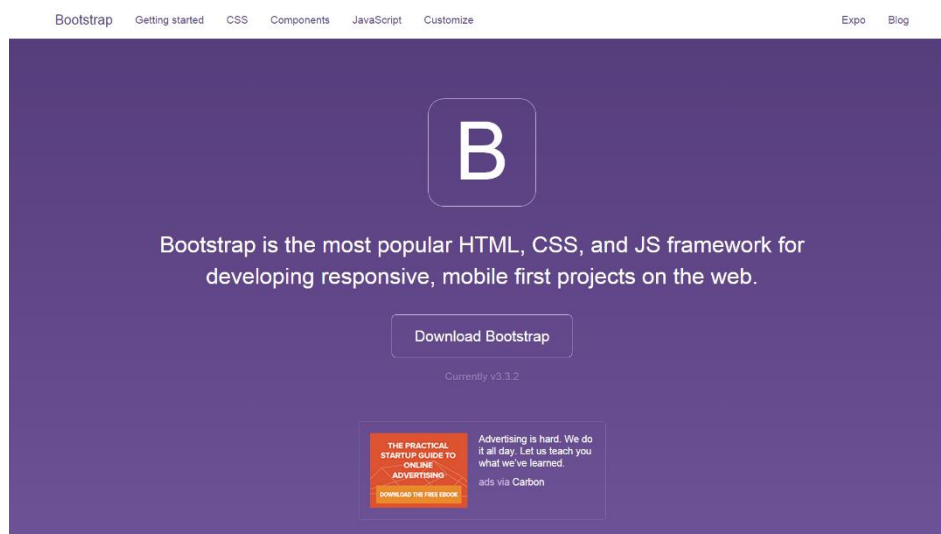
<sup>8</sup> CMS - Systém pro správu obsahu webové stránky

celý web. Z širšího hlediska je framework soubor pojmů, postupů a kritérií, které pomáhají řešit určité problémy (Awwwards 2015).

Na internetu existuje velká spousta různých frameworků, které nabízí, že nám ušetří čas i práci při tvorbě webových stránek. Některé jsou známé, jiné naopak méně, některé jsou aktualizovány a vylepšovány již roky, jiné jsou ještě v uvozovkách v plenkách. Jednotlivé projekty se od sebe odlišují technickými vlastnostmi, vývojovými koncepty a samozřejmě paletou nabízených prvků s různou funkčností a grafickým zpracováním (Sharkie 2013, Fielding 2014, Peterson 2014).

Mezi nejznámější a co do kvality nejvíce propracované patří: (Peterson 2014)

- a) **Bootstrap** pocházející z vývojářských dílen společnosti Twitter. Bootstrap je nejpopulárnější open source HTML, CSS, a JavaScript responzivní framework. Za jeho vznikem stojí velká skupina zkušených lidí a k jeho zlepšování se zapojuje také velké množství nezávislých přispěvatelů. Projekt frontend frameworku Bootstrap byl spuštěn v polovině roku 2010, z toho vyplývá, že se jedná o projekt starší a tudíž by měl být vyzrálý. K lednu roku 2015 Bootstrap vydal již 31 verzí a poslední aktualizací je verze 3.3.2. Bootstrap je založen na konceptu mobile-first<sup>9</sup> a využívá principů plovoucí mřížky, je tedy plně adaptabilní. U jako jediného projektu z uvedených je použitý CSS procesor LESS<sup>10</sup> i SASS<sup>11</sup>. Díky širokému rozšíření má velmi dobrou podporu (Sharkie 2013, Fielding 2014, Peterson 2014, Bootstrap 2015).



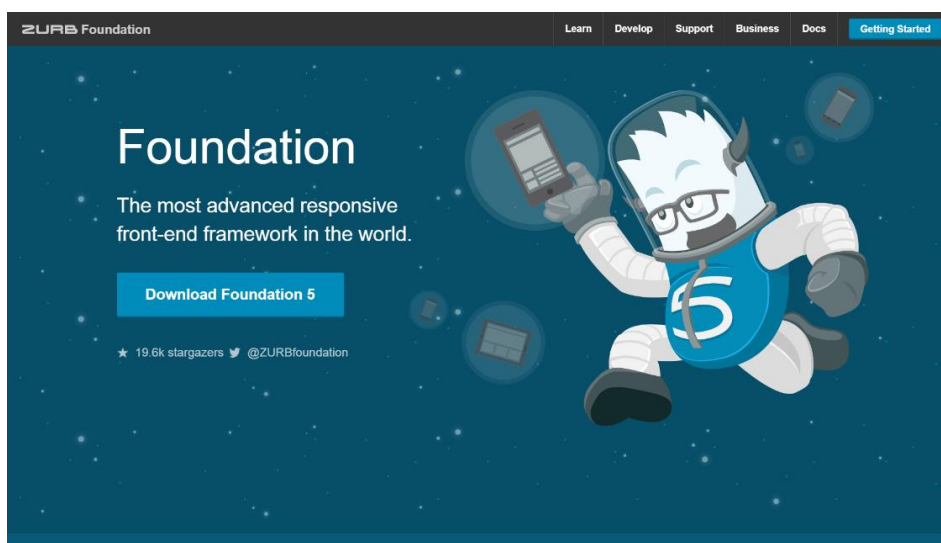
Obrázek 4: Twitter Bootstrap (<http://getbootstrap.com>)

<sup>9</sup> Metoda tvorby responzivního webu zaměřená na mobilní zařízení

<sup>10</sup> Kaskádové styly spolupracující s dynamickými prvky jako funkce, výpočty a proměnné (Gasston 2015)

<sup>11</sup> Kaskádové styly využívající prvků dědičnosti, variability a jednoduchosti (Gasston 2015)

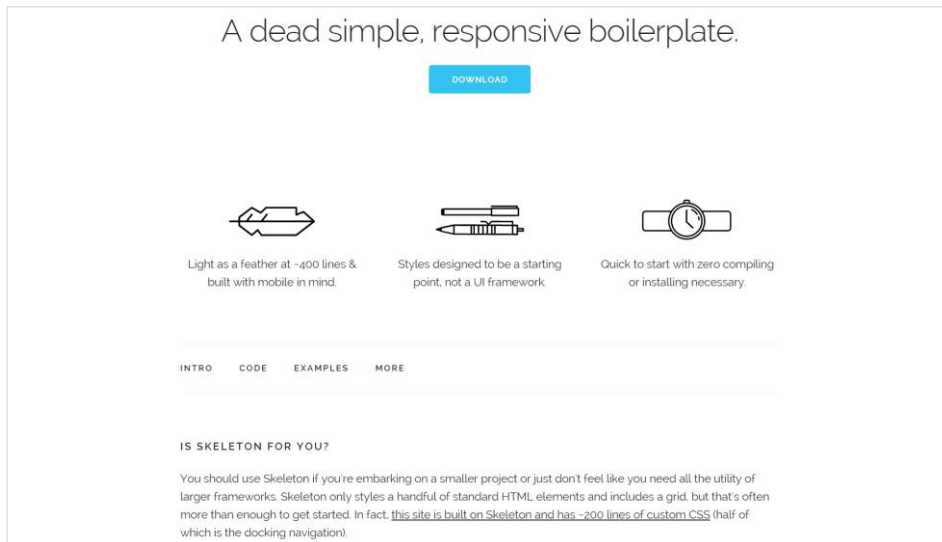
b) **Foundation** vytvořen web designerskou společností ZURB. Foundation je jeden z nejvíce rozšířených responzivních frontend frameworků. Co se týká dostupnosti zdrojových kódů, licence open source byla vydána až s verzí 2.0 v roce 2011. Poslední aktualizace k lednu 2015 byla vydána ve verzi 5.5.1. Jedinečnou vlastností, kterou Foundation vyčnává nad ostatní responzivní frameworky je *Interchange Resposive Content* dále jen IRC. IRC umožňuje pomocí dotazů media queries vkládat nebo nahrazovat části HTML kódu ve webové stránce. V praxi to znamená, že Foundation založený na konceptu mobile-first, přidává s různými šířkami viewportu kusy HTML kódu, popřípadě dokáže měnit velikosti obrázků, tak aby kvalita odpovídala zobrazení. I přes specifické vlastnosti, které každý projekt má, jsou si oba dost podobné, jsou podobně staré a mají tudíž i velké zkušenosti a za vývojem obou projektů stálo mnoho lidí. Je tedy jen otázka názoru, který ze dvou rozsáhlých frontend frameworků získá na světové úrovni více hodnocení v superlativech. Foundation používá CSS procesor SASS (Sharkie 2013, Fielding 2014, Peterson 2014, Foundation 2015).



Obrázek 5: Zurb Foundation (<http://foundation.zurb.com>)



- c) **Skeleton**, je třetím příkladem frontend frameworku, není tak známý a ani se nepyšní žádnými prvenstvími ve světě webového inženýrství jako dva již zmíněné příklady. Odlišuje se tím, že je postaven a konceptu desktop-first<sup>12</sup>. Skeleton je dobrou volbou při tvorbě mobilně orientovaných webových stránek. Pracuje tak, že při snižující se velikosti viewportu odebírá nebo mění jednotlivé prvky webové stránky (Firdaus 2013, Sharkie 2013).



Obrázek 6: Skeleton (<http://getskeleton.com>)

<sup>12</sup> Metoda tvorby responzivního webu zaměřená na desktopové počítače

## 4 METODY TVORBY RESPOZIVNÍ WEBOVÉ STRÁNKY

### 4.1 Detekce zařízení – přesměrování

Metoda přesměrování webové stránky je založena na principu detekce typu zařízení pomocí zjišťování informací z webového prohlížeče. Existují dva typy přesměrování, které se provádí různými skriptovacími jazyky. Mezi nejběžnější a nejvíce používané patří JavaScript a PHP (Zdroják 2011).

Princip detekce je založen na porovnávání informací získaných s user-agent webového prohlížeče. User-agent je textový řetězec, který vystupuje jako uživatel v interakci se serverem. Webový prohlížeč se tímto způsobem, při komunikaci se serverem, prokazuje a poskytuje informace o sobě, případně o zařízení. Skript pro detekci typu prohlížeče získá informace z user-agent a porovná je s klíčovými slovy, které má zadané a při shodě provede akci. Například pokud se jedná o detekci mobilního zařízení, skript najde shodu ve slově *mobile* a vykoná akci přesměrování webové stránky na mobilní verzi (Sood 2014).

#### 4.1.1 Přesměrování na straně klienta

Přesměrování na straně klienta se provádí za pomoci jazyku JavaScript. Do zdrojového kódu webové stránky je implementován skript s dotazem na user-agent. Při přečtení stránky se přečte i skript a po jeho vyhodnocení se buď provede, nebo neprovede přesměrování. Problém nastává v případě, pokud zařízení nemá JavaScript ve svém webovém prohlížeči povolen. Další nepříjemností při použití tohoto typu přesměrování je, že se webová stránka i s obsahem již načítá a až v průběhu této činnosti se spustí skript. Toto může být velkou nevýhodou právě pro mobilní zařízení, protože mobilní internet je podmíněn určitým objemem dat. Návštěvníci responzivního webu tak očekávají, že bude web datově co nejméně náročný. Vhodnější by tedy bylo, kdyby se skript spustil ještě před samotným načítáním stránky ve webovém prohlížeči uživatele. Posledním negativem přesměrování na straně klienta je, že není ideální z pohledu SEO (Pasqua 2013).

## Skript pro detekci typu prohlížeče na straně klienta – JavaScript

```
<script>
    Var isMobile = navigator.userAgent.match
    (/ (iPhone|iPod|iPad|Android|BlackBerry) /);

    if(isMobile)
    {
        window.location =
"http://marekstraka.php5.cz/mobile";
    }
</script>
```

(PC-HELP 2014)

### 4.1.2 Přesměrování na straně serveru

Druhým typem přesměrování webové stránky je přesměrování na straně serveru. Na rozdíl od přesměrování na straně klienta se v tomto případě používá skriptovací programovací jazyk PHP, který je zpracováván na straně serveru. Dochází tak ke spuštění a zpracování skriptu ještě na straně serveru a na základě výsledků skriptu se provede příslušná akce. Velkou výhodou je větší bezpečnost kódu. Jelikož se PHP zpracovává na serveru, tak k uživateli připutuje již konečné řešení a samotný skript nikoliv. Je tak znemožněno jeho zkopírování nebo zneužití (Pasqua 2013).

## Skript pro detekci typu prohlížeče na straně serveru – PHP

```
<?php
    $useragent=$_SERVER['HTTP_USER_AGENT'];
    if(preg_match('',$useragent)||preg_match('',$substr
    ($useragent,0,4)))
    header('Location: http://marekstraka.php5.cz/mobile');
?>
```

(Detect Mobile Browsers 2014)

Přesměrování je akce, při níž je změněna původní URL adresa zadaná uživatelem, na adresu definovanou skriptem. Přesměrování nejčastěji probíhá na úrovni domén, nebo adresářů. V rámci jedné domény druhého řádu například *www.seznam.cz* lze vytvořit podadresář *www.seznam.cz/mobile* odkud bude mobilní webová stránka načítána. Druhým způsobem řešení je, k doméně druhého řádu vytvořit doménu třetího řádu, zpravidla to bývá v tomto provedení *www.m.seznam.cz* a zde si uložit verzi mobilní webové stránky. V případě vlastnictví domény prvního řádu, jsou obě řešení srovnatelná, nic to nestojí a provedení je rychlé a jednoduché. U velkých webových projektů se však setkáváme spíše s provedením přesměrování na mobilní subdoménu. V případě freehostingů, kde získáme zdarma pouze doménu třetího řádu, nemáme jinou možnost, než použít metodu s vytvořením podadresáře, protože další subdoménu již vytvořit nelze (Pasqua 2013).

## 4.2 Detekce šířky

Metoda detekování šířky displeje různých zařízení je v rámci responzivního web designu prováděna pomocí modulu kaskádových stylů media queries. Na základě breakpointů jsou v kaskádových stylech vytvářeny jednoduché podmínky. Jako kritérium podmínky slouží různé vlastnosti displeje zařízení, například šířka a výška displeje nebo jen viewportu, dále pak rozlišení, poměr stran nebo orientace displeje a tak dále. Podmínka je splněna hodnotou buď nižší než je zadáno nebo naopak vyšší. Po splnění podmínky je aplikován alternativní kaskádový styl. Na ukázce předvedu formu zápisu a dva základní typy podmínky media queries pro šířku viewportu (McFarland 2013, Fielding 2014, Peterson 2014, Gasston 2015).

```
@media (max-width:768px) { /*platnost do maximální hodnoty*/
  #container {
    width: 100%; /*css kód aplikován po splnění podmínky*/
  }
}
@media (min-width:768px) { /*platnost nad maximální hodnotu*/
  #container {
    width: 1000px; /*css kód aplikován po splnění podmínky*/
    margin: 10px auto;
  }
}
```

Podmínka je vždy uvozena výrazem *@media*, který vyjadřuje, že se jedná o dotaz na media queries. Poté následuje v kulatých závorkách samotná podmínka, která se skládá z ověřované vlastnosti a ze zlomové hodnoty. Uvnitř svorkových závorek se nachází tělo podmínky, což je kaskádový styl, který bude aplikován v případě splnění podmínky (McFarland 2013, Fielding 2014, Peterson 2014, Gasston 2015).

První ze dvou příkladů podmiňuje, že `#container` bude defaultně zarovnán do levého horního rohu viewportu a bude přizpůsoben jeho stoprocentní šířce. Podmínka *max-width:768px* udává, že tyto vlastnosti budou prvku nastaveny při šířce viewportu maximální tedy rovné hodnotě 768 pixelů nebo libovolně menší.

Naopak druhá podmínka *min-width:768px* udává minimální rovnou hodnotu platnosti. Což znamená, že při šířce viewportu rovné 768 pixelů a vyšší bude prvku `#container` nastavena šířka 1000 pixelů a bude z deseti pixelovým odsazením zarovnán na střed viewportu.

#### **4.2.1 Desktop-first**

Dříve než se začaly používat mobilní zařízení jako například smartphony a tablety, vývojáři a designeři vytvářeli své webové stránky na míru desktopovým webovým prohlížečům. Webová stránka měla fixní univerzální šířku a cílem při tvorbě takových webových stránek bylo, aby se dobře zobrazovaly na velkých displejích. Tehdy však ještě nešlo o desktop-first přístup, ten vznikl až s nástupem éry responzivního web designu. Myšlenka desktop-first je ovlivněna tradicí, vychází se z původního zaměření na desktopové počítače a druhotně je třeba přizpůsobit webovou stránku mobilním webovým prohlížečům (Clark 2012, McFarland 2013, Fielding 2014).

V praxi vypadá vytváření webové stránky podle přístupu desktop-first následovně. Weboví vývojáři a designeři vytvoří webovou stránku s pevnou šířkou mřížky a s layoutem, který se bude optimálně zobrazovat na obrazovkách desktopových počítačů. Poté se zvolí určité breakpointy pro menší, různě velká mobilní zařízení, například jeden breakpoint pro tablety v módu portrét a větší smartphony v módu krajina a druhý pro smartphony v módu portrét a ostatní menší zařízení. Na základě breakpointů za pomoci media queries se pak vytváří odlišné kaskádové styly, které webovou stránku upravují, mění rozložení a šířku mřížky, mění velikost obrázků a celou grafiku a velmi často jsou některé nadbytečné prvky úplně skrývány. Přístup

desktop-first je také označován jako degradační, protože z úplné webové stránky jsou odebírány její části a stránka je pak v uvozovkách degradována na úroveň mobilního prohlížeče (Clark 2012, McFarland 2013, Fielding 2014).

Výhodou přístupu desktop-first je, že webová stránka se na desktopových webových prohlížečích zobrazuje v základních kaskádových stylech bez použití media queries. Problémem je totiž podpora media queries, jedná se o poměrně mladý prvek kaskádových stylů, což znamená, že dobře pracuje až v posledních verzích moderních webových prohlížečů. Ve starších verzích je podpora buď jen částečná, nebo není podporován vůbec. Oproti tomu většina mobilních webových prohlížečů media queries podporuje a je tak zajištěna dobrá optimalizace zobrazení webové stránky jak pro desktopové počítače, tak pro mobilní zařízení (Clark 2012, McFarland 2013, Fielding 2014).

Nevýhodou je náročnost změn jednotlivých kaskádových stylů pro různá zařízení z pohledu rozsahu kódu. Při každé změně breakpointu je většinou potřeba upravit značnou část kódu a některé prvky se upravují i vícekrát. Často tak může dojít k nekorektnímu zobrazování. Duhou nepřijemností je složitější optimalizace pro mobilní zařízení. Opět je třeba změnit celou podstatu webové stránky a odebrat a nahradit spoustu prvků. I přesto nebude mobilní verze úplně dokonalá (Clark 2012, McFarland 2013, Fielding 2014).

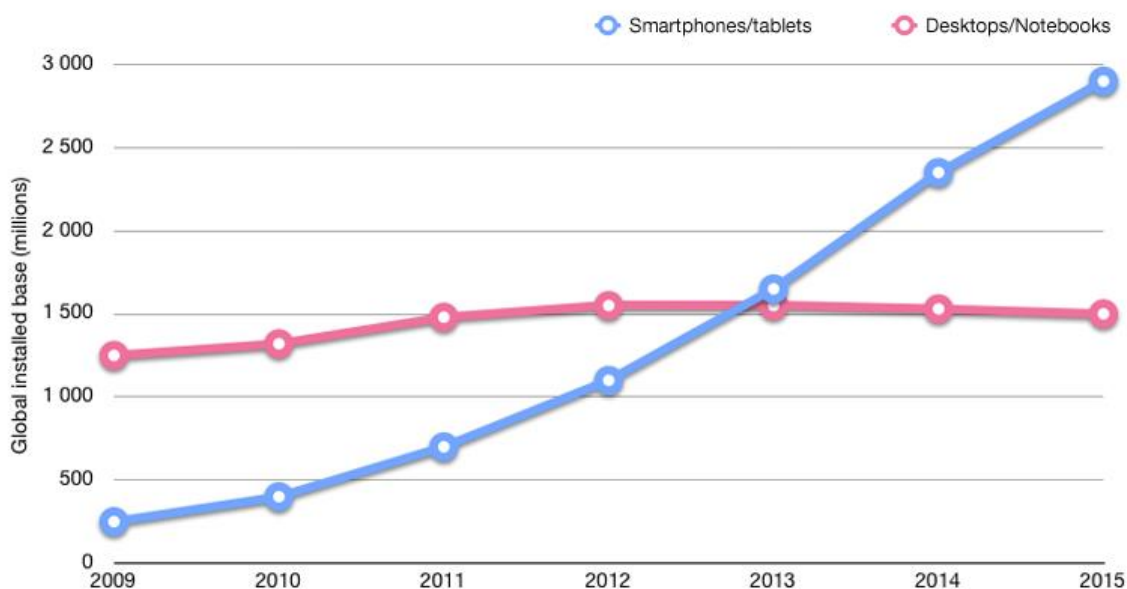
#### **4.2.2 Mobile-first**

Mobile-first je velmi zjednodušeně řečeno opačný pohled vůči desktop-first. Mobile-first je nový přístup měnící zažité postupy při vytváření webových stránek. Přenáší pozornost od desktopových počítačů, které jsou v současné době spíše na ústupu, k mobilním zařízením, které se dostávají do popředí a jsou stále více oblíbené. Webové stránky vytvářené tímto přístupem jsou primárně optimalizovány pro mobilní zařízení a poté postupně rozšiřovány pro širší obrazovky. Přístup mobile-first si získal pozornost tvůrců webových stránek díky knize Luka Wroblewského *Mobile First, 2011* a to, pouhé dva roky po vzniku responzivního přístupu. Zde je vidět platnost neochvějného tvrzení „co bylo včera moderní, dnes je již zastaralé“ (Wroblewski 2011, Clark 2012, Lunn 2013, McFarland 2013, Sharkie 2013, Fielding 2014, Peterson 2014).

Při vytváření mobile-first webové stránky je nejdůležitější moderní uživatel a jeho potřeba být mobilní. Weboví vývojáři a designeři vytvoří webovou stránku s plovoucí

mřížkou a s obsahem optimalizovaným pro mobilní zařízení, zvláště pro smartphony. Pro zařízení s větší obrazovkou a s vyšším rozlišením jsou stanoveny breakpointy tak, aby pokryly celou škálu všech typů. K jednotlivým breakpointům prostřednictvím media queries jsou vytvářeny další kaskádové styly, které základní styly rozšiřují. Pro jednotlivé vyšší stupně velikosti zobrazení se zvětšuje šířka mřížky, navyšuje se počet sloupců, přidávají se prvky, přizpůsobuje se grafika a velikost textu a obrázků (Wroblewski 2011, Clark 2012, Lunn 2013, McFarland 2013, Sharkie 2013, Fielding 2014, Peterson 2014).

Výhod je hned několik. V první řadě hlavně to, že přístup vychází ze současnosti a je zaměřen na budoucnost. Počítá se s mobilitou. Již v současné době má společnost k dispozici velké množství různých typů mobilních zařízení. Tento současný trend ukazuje, že budoucnost se bude ubírat právě tímto směrem (Obrázek 7).



Obrázek 7: Porovnání růstu uživatelů přistupujících k internetu z desktopových počítačů a notebooků oproti smartphonům a tabletům za období 2007 - 2015

Dalším pozitivem je na pohled jednodušší a přehlednější kód. Základní kaskádové styly pro smartphony se na základě breakpointů pouze rozšiřují o další prvky a grafiku. Nemusí docházet k rozsáhlému skrývání prvků. S tím souvisí další výhoda a tou je menší časová náročnost tvorby webové stránky. S přístupem mobile-first zdrojový kód není tak rozsáhlý a vytvoření webové stránky tudíž zabere méně času (Wroblewski 2011, Clark 2012, Lunn 2013, McFarland 2013, Sharkie 2013, Fielding 2014, Peterson 2014).

Nevýhoda přístupu mobile-first je jen jedna jediná a tou je podpora. Starší desktopové prohlížeče, zvláště pak Internet Explorer verze 8 a nižší nepodporují media queries a způsob jak vyřešit tento problém je značně obtížný. Existuje ještě jedna nepříjemná záležitost, která bývá také považována za nevýhodu mobile-first přístupu a tou je odlišné a netradiční uvažování při tvorbě mobile-first webové stránky. Mobile-first přístup vyžaduje odlišnou zkušenost a k jeho zvládnutí je třeba naučit se nové postupy a pravidla. Potřebu učit se něco nového, když to dovedu vytvořit jinak, může někdo považovat za nevýhodu (Wroblewski 2011, Clark 2012, Lunn 2013, McFarland 2013, Sharkie 2013, Fielding 2014, Peterson 2014).

### 4.2.3 Viewport

Viewport je důležitý prvek v oblasti responzivního web designu. Jako meta element byl používán již dříve a to u neresponzivních webů. Viewport je okno webového prohlížeče, ve kterém se zobrazuje webová stránka (Obrázek 8). Po otevření okna webového prohlížeče a spuštění webové stránky se v prostoru viewportu zobrazí menu, hlavička, tělo a pozadí webové stránky. Pokud má webová stránka pevnou šířku, zbylý prostor viewportu se vykreslí nadefinovaným pozadím nebo jen bílými pruhy po stranách, pokud má však nastavenou stoprocentní šířku, vyplní celý viewport. Prostor viewportu zpravidla není stejně velký jako obrazovka zařízení ani jako okno webového prohlížeče, od jeho šířky a výšky je třeba odečíst navigační panel, posuvník, stavový řádek a případné okraje aplikace. Většina moderních webových prohlížečů však má již funkci fullscreen, která umožňuje viewport roztáhnout přes celou obrazovku. U mobilních webových prohlížečů je viewport chápán stejně, jen s tím rozdílem, že jeho šířka již odpovídá šířce obrazovky, protože zde chybí postranní posuvník (Fielding 2014, Peterson 2014).



Obrázek 8: Znárodnění plochy viewportu v okně webového prohlížeče



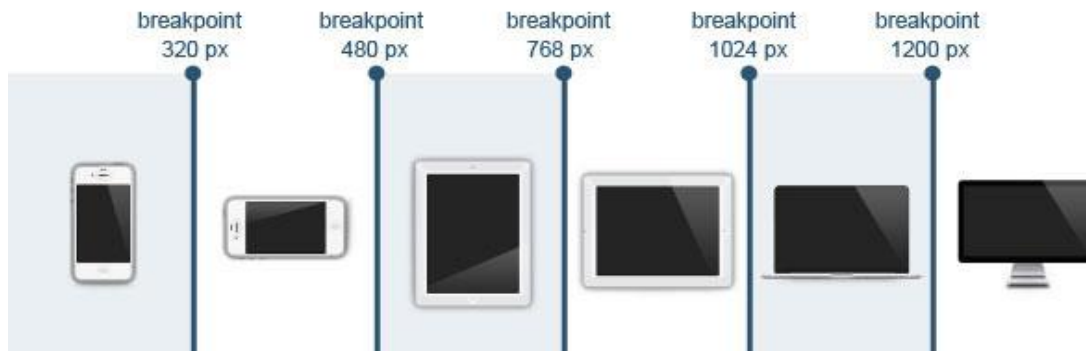
#### 4.2.4 Breakpointy

Pro responzivní web design je právě breakpoint tím nejpodstatnějším a nedůležitějším pojmem. Abychom dobře pochopili co to breakpoint je, potřebujeme nejprve porozumět pojmům viewport a media queries (McFarland 2013, Fielding 2014, Peterson 2014).

Breakpoint je bod, který je nastavený na určitou šířku, po jehož prolomení dochází ke změně layoutu webové stránky. Změny jsou prováděny pomocí media queries. Jako jednotlivé breakpointy se používají šířky displejů odlišných typů zařízení, tak aby byla obsažena celá škála (Obrázek 9). Avšak šířka displeje slouží jen jako orientační měřítko, ve skutečnosti se breakpointy nastavují na šířku viewportu, je tak možné testovat veškeré typy zařízení pouze změnou šířky okna prohlížeče. U responzivní webové stránky se používají dva, nejčastěji však tři nebo i více breakpointů. Někdy jsou změny jen drobného charakteru, jako například změna velikosti písma nebo změna počtu či prostorová orientace položek v menu. Většinou však dochází k na první pohled viditelným změnám celého schématu webové stránky. Důležitou změnou je úprava mřížky, to je změna počtu sloupců, u tabletů se nastavují většinou dva sloupce a u smartphonů pak jen jeden sloupec. Dále se zjednodušuje grafika a některé nadbytečné prvky se skrývají, kvůli úspoře místa. Základní používané breakpointy podle šířky rozlišení se dělí na tyto čtyři skupiny (McFarland 2013, Fielding 2014, Peterson 2014):

- 1) Velmi malá zařízení (<768px) – například smartphony, PDA
- 2) Malá zařízení (768px – 992px) – například tablety, netbooky
- 3) Středně velká zařízení (992px – 1200px) – notebooky, monitory počítačů
- 4) Velká zařízení (>1200px) – velké monitory počítačů, HD displeje, smart TV

Koncept breakpointů vznikl a používá se kvůli nárůstu velkého množství typů zobrazovacích zařízení s přístupem k internetu a s tím vzniklo i velké množství různých rozlišení displejů (McFarland 2013, Fielding 2014, Peterson 2014).



Obrázek 9: Základní používané breakpointy s příkladem zařízení

### 4.3 Responzivní front-end framework

V praxi je hlavním cílem frontend frameworků ušetřit čas programátorům při vytváření webových stránek. Výroba webové stránky je vlastně velmi podobná výrobě osobního automobilu. Struktura webových stránek je v podstatě téměř stejná. Hlavní rozložení jednotlivých částí webu má již své určité schéma, které se dodržuje a moc se nemění stejně tak, jako se téměř nemění struktura podvozku u různých aut. Také jednotlivé prvky jako například *menu*, *button*, *label* nebo *form* jsou v základu stejné, odlišit je od ostatních můžeme jen jedinečným designem. I auta mají kola, světla či dveře a záleží pak jen, jakou mají barvu, velikost nebo tvar, aby nebyla všechna stejná. Frontend framework je jako velký sklad externího dodavatele, který nám dává k dispozici hotové, funkční prvky, jako jsou kola, světla nebo dveře. Při výrobě pak máme možnost využít některé z hotových prvků a vhodně je zakomponovat. Toto řešení přináší velkou úsporu času, protože nemusíme pořád dokola vytvářet stejnou věc od začátku, ale použijeme již existující řešení v různých případech. Responzivní frameworky jsou ve své podstatě stejné, jako základní frameworky pro vytváření webových stránek. Mají však jednu důležitou vlastnost a to, že předdefinované prvky a schémata již pracují z responzivním webovým designem. Zdrojové kódy prvků se přizpůsobují tak, aby bylo možné je zobrazovat na celé škále typů současných zařízení, s různou velikostí a rozlišením displeje (Peterson 2014, Awwwards 2015).

#### 4.3.1 Proč použít framework

Důvodů proč použít při tvorbě webové stránky responzivní frontend framework je několik. Některé zde vyjmenuji a přiblížím jejich podstatu.

- Použitím frameworku **ušetříme čas**. Podmínkou je, že se však v dané problematice orientujeme a pracujeme s frameworkem správně. Frameworky obsahují

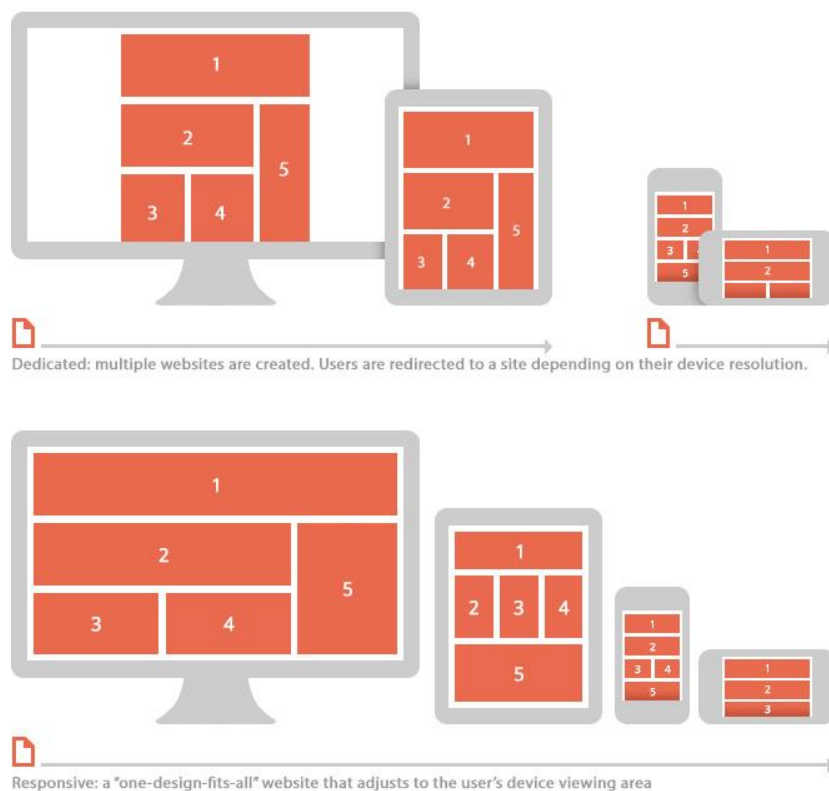
předdefinovaná pravidla a styly pro tlačítka, fonty, formuláře a další. Šablony frameworků mají také přednastavené mřížky s určitými počty sloupců s danými šířkami. Při tvorbě webové stránky už jen vkládáte HTML kód jednotlivých prvků a krok po kroku si poskládáte přesně to, co potřebujete. Kaskádové styly prvků jsou již nadefinovány a tak již nemusíte každý prvek definovat od začátku (Firdaus 2013).

- Kolem velkých a rozšířených frameworků stojí vždy **velká komunita**, která s největší pravděpodobností stála u zrodu celého projektu, udržuje jeho aktuálnost, spravuje a rozšiřuje ho. Díky skupinám lidí, které framework neustále zlepšují, se zvyšuje i jeho **popularita** a z kvalitnější se podpora (Firdaus 2013).
- **Kompatibilita mezi webovými prohlížeči** je vlastnost webové stránky, která zabezpečuje, aby se stránka zobrazovala stejně ve všech webových prohlížečích (všech značek, typů a s co největší možnou podporou starších verzí). Frameworky nám tuto práci velmi usnadňují a případné problémy s nekompatibilitou snižují téměř na minimum. Je tomu tak, protože velké projekty mají dostatek lidí, kteří se problémem mezi-prohlížečové kompatibility zabývají ještě dlouhou dobu před tím, než se framework dostane na veřejnost (Firdaus 2013).
- Dobrý framework také obsahuje dobrou **dokumentaci**. Dokumentace je velmi užitečný text doplňující zdrojový kód o informace: co který kus kódu vyjadřuje, jak pracuje a jak je třeba s ním zacházet. Dobrá dokumentace je třeba, pokud s kódem pracují i jiní lidé než jen jeho tvůrce. Díky dokumentaci je kód přehlednější a srozumitelnější (Firdaus 2013).

## 5 TVORBA WEBOVÉ STRÁNKY OPTIMALIZOVANÉ PRO MOBILNÍ ZAŘÍZENÍ

Vytváření statických webových stránek s pevnou šířkou je v dnešní moderní době až zpátečnické jednání. Současnost a budoucnost se ubírají směrem mobility. Netvrdím, že velkoplošné obrazovky existovat nebudou, právě naopak, ale je třeba počítat s tím, že uživatel bude chtít jedním tlačítkem či pouze gestem přenést obraz do svého mobilního zařízení. Také laptopy se pomalu stávají pouze přenosnými počítačovými stanicemi. Novým vývojovým stupněm jsou mobilní dokovací stanice s odnímatelnými displeji. Displej samostatně funguje jako tablet a spolu s dokem je rozšířen o další hardwarové komponenty a tvoří plnohodnotný laptop. A to je mobilní éra teprve na vzestupu, novější technologie doslova vytlačují své předchůdce přes noc a k pokroku tak dochází téměř nepřetržitě. Proto je třeba jít s dobou a přizpůsobit se těmto změnám od základu. Co je větším základem než webová stránka, která tvoří, jako dílek skládačky, celý Internet, jenž je největším nástrojem globalizace a kýžené mobility.

Při vytváření webových stránek optimalizovaných pro dnešní moderní, mobilní dobu jsem postupoval podle dvou přístupů, které prezentuje následující obrázek (Obrázek 10).



Obrázek 10: Schématické znázornění dvou přístupů tvorby webových stránek optimalizovaných pro mobilní zařízení

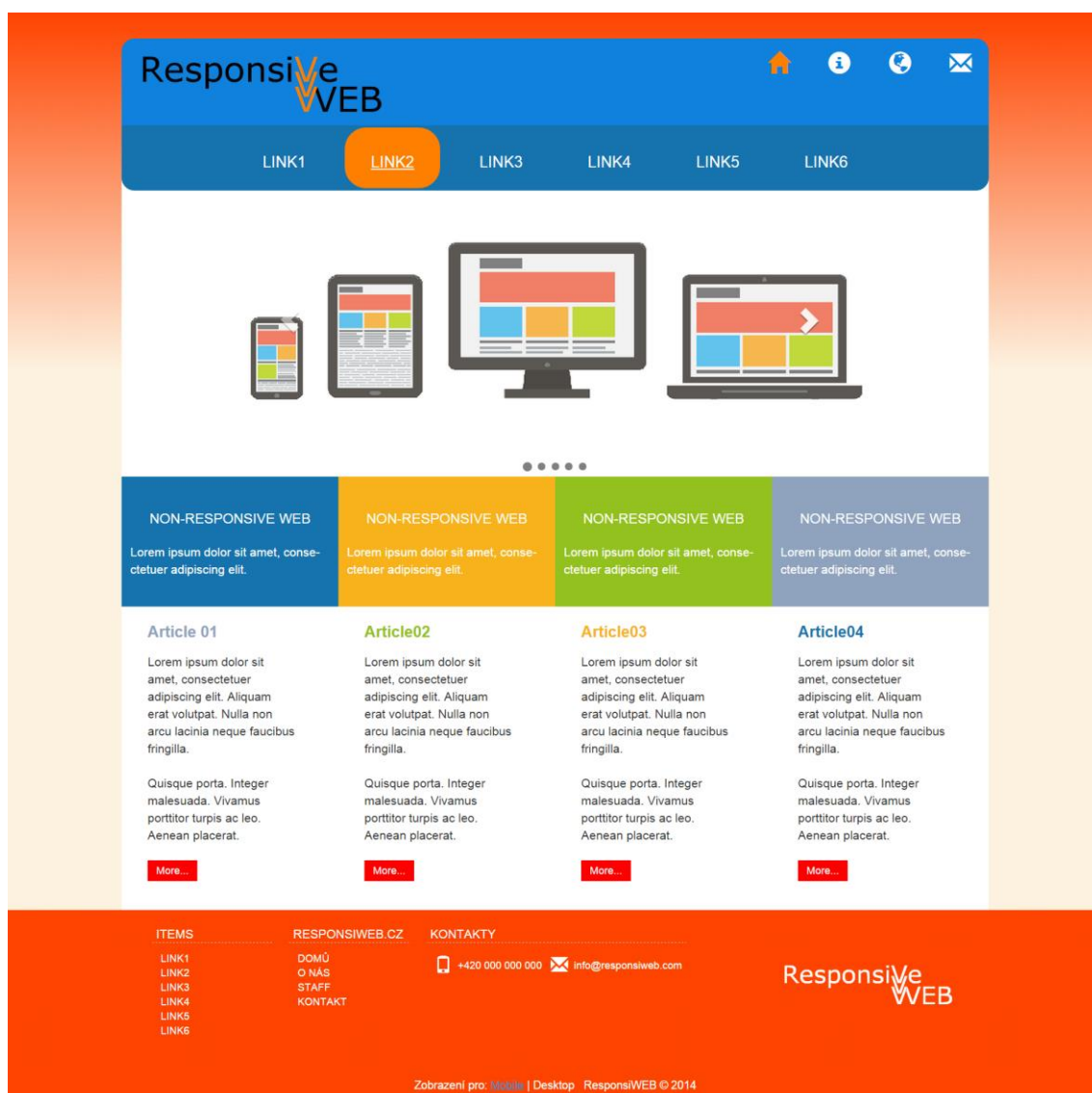
Znalosti, které jsem získal v předešlých letech studiem internetu a praktickými zkušenostmi v oblasti tvorby webových stránek jsem doplnil o nové informace převážně z webových stránek, ale také z odborné literatury. Největším přínosem pro mne bylo studium CSS3 verze kaskádových stylů a speciálně modulu media queries. Své dosavadní znalosti jsem také musel obohatit o novinky jazyka HTML5. Pomocí nově získaných informací společně s vlastními zkušenostmi jsem vytvořil dva modely webových stránek s různými přístupy k optimalizaci pro mobilní zařízení. První model jsem vypracoval podle jednoúčelového přístupu, který řeší dodatečnou optimalizaci již existujících webových stránek. Druhý model webové stránky je nejmodernějším a nejuniverzálnějším řešením optimalizace a vytvářel jsem ho na základě responzivního přístupu.

## **5.1 První model - jednoúčelový přístup**

Princip tohoto přístupu je schematicky znázorněn v horní polovině Obrázku 10. Hlavním prvkem optimalizace pro mobilní zařízení je v tomto případě mobilní verze webové stránky, která se zobrazí po přesměrování z desktopové verze při přístupu na webovou stránku z mobilního zařízení. Je tedy charakteristické pro tento způsob optimalizace, že je třeba vytvořit dvě webové stránky. Jednu, která se bude zobrazovat na smart TV obrazovkách, desktopových monitorech nebo na různých typech laptopů a větších tabletech. Druhá webová stránka bude určena pro menší mobilní zařízení, jako jsou smartphony, PDA nebo menší displeje dalších chytrých řízení. Název přístupu jednoúčelový má tedy své odůvodnění v tom, že každá ze dvou webových stránek má svůj jediný účel, pouze jedno zaměření.

### **5.1.1 Desktopová verze**

Jak jsem již zmínil, bylo třeba, abych vytvořil dvě webové stránky. Jako první jsem vytvořil běžnou webovou stránku s pevnou šířkou. Cílem je zobrazení na velkých a středně velkých displejích různých zařízení. Převážně však jde o desktopové počítače, laptopy ale i tablety, rozhodující je velké zobrazovací rozlišení. V minulosti tvořily největší cílovou skupinu desktopové počítače, a proto se dnes tomuto typu webové stránky říká desktopová ve srovnání s mobilní verzí. Na následujícím obrázku (Obrázek 11) prezentuji náhled desktopové verze webové stránky, kterou jsem vytvořil jako ukázkou.



Obrázek 11: Desktopová verze webu (rozlišení 1920 x n)

Postup není ničím specifický, jako první jsem vytvořil layout a z něj jsem pak navrhnul rozložení hlavních prvků na stránce. Vzhledem k tomu, že webová stránka má statickou šířku, zvolil jsem zarovnání na střed viewportu. V závislosti na zobrazovacím rozlišení displeje se zbylá šířka viewportu vykresluje nastaveným pozadím. Pozadí za samotnou stránkou jsem vyřešil novinkou CSS3 definováním barev *linear-gradient*.

```
background: linear-gradient(#FE4400, #FDF2DF 33%);
```

Jedná se o plynulý přechod dvou barev. Procentuální hodnota v závorce vyjadřuje polohu mísení barev z celkové délky prvku, na který je tato vlastnost aplikována. Další

inovací kaskádových stylů verze 3 je bez-obrázkové řešení grafiky celého webu. Jedinými použitými obrázky je logo v hlavičce a v patičce webové stránky a obrázky v grafickém prvku carousel. Jako ikony v hlavičce a patičce byly použity vektorové ikony (glyphicons). Významnou a novou vlastností, kterou jsem použil na místo obrázkové grafiky je zaoblování rohů prostorových prvků *border-radius*.

```
border-radius: 15px 15px 0px 0px;
```

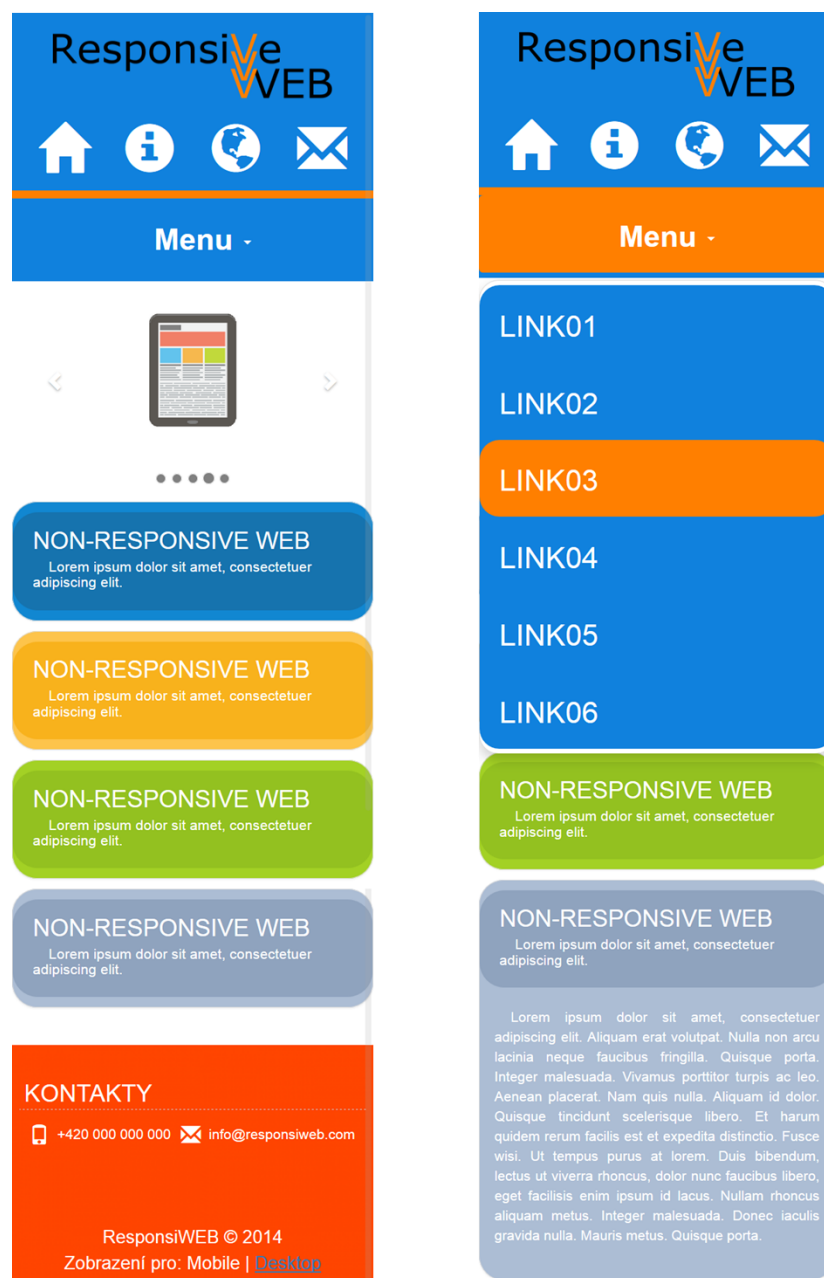
Příklad použití vlastnosti *border-radius* nastavuje zaoblení levého a pravého horního rohu na poloměr 15 pixelů. Tento řádek kódu je použit u prvku *#header*.

Tím nejpodstatnějším co jsem do webové stránky přidal, co je pro optimalizace na mobilní zařízení v tomto přístupu nezbytné, je skript, sloužící k přesměrování uživatele na mobilní verzi webové stránky. Použil jsem přesměrování na straně serveru pomocí PHP skriptu. Oproti přesměrování na straně klienta, pomocí JavaScriptu, je toto řešení rychlejší, nezatěžuje datový tok uživatele před-načítanými daty a hlavně není závislé na webovém prohlížeči uživatele. Problém s přesměrováním by totiž mohl nastat v případě, kdyby uživatel neměl ve svém webovém prohlížeči JavaScript povolen.

```
<?php
    $useragent=$_SERVER['HTTP_USER_AGENT'];
    if(preg_match('',$useragent)||preg_match('',substr
        ($useragent,0,4)))
        header('Location: mobile/index.html');
?>
```

## 5.1.2 Mobilní verze

Mobilní verze webové stránky je stěžejním prvkem optimalizace pro mobilní zařízení v konceptu jednoúčelové webové stránky. Bez mobilní verze by zůstala jen samotná desktopová verze se statickou šířkou, což by vyhovovalo možná tak před více než deseti lety, v současné době však nikoliv. Vytvořením mobilní verze rozšiřujeme desktopovou webovou stránku o ideální zobrazení pro menší mobilní zařízení, převážně pro smartphony. Toto řešení se velmi často používá v případě, že desktopová stránka již existuje a je třeba ji optimalizovat neinvazivním způsobem. Následující obrázek (Obrázek 12) prezentuje mobilní verzi webové stránky ve dvou zobrazeních.



Obrázek 12: Mobilní verze webu (rozlišení 360 x n)



Vlevo na obrázku je webová stránka v základním zobrazení. Webová stránka v pravé části obrázku ukazuje skryté prvky v aktivní fázi po rozkliknutí. V horní části se nachází rozbalovací menu s položkami a pod ním v obsahové části webové stránky jsou položky s možností po kliknutí rozbalit další obsah.

Mobilní verze webové stránky cílená na smartphony a podobná mobilní zařízení se vytváří zpravidla s jednosloupcovou mřížkou. Jednotlivé části desktopové webové stránky, které jsem se rozhodl přenést do mobilní verze, se skládají pod sebe v jednom sloupci. Pro velkou rozmanitost různých velikostí a zobrazovacích rozlišení u smartphonů je lepší při vytváření mobilní verze používat relativní šířky. Šířku celé webové stránky i šířky jednotlivých částí jsem proto navrhoval tak, aby se přizpůsobovaly 100% šířce viewportu. Eliminoval jsem tak možnost odlišného zobrazování webové stránky na různě velkých displejích.

Při vytváření mobilní verze webové stránky jsem použil některé prvky z responzivního frontend frameworku Bootstrap, které jsem po stránce designu, vhodně přizpůsobil. Jedná se o rozbalovací menu a obsahové položky. Design mobilní verze je sladěn s designem desktopové verze a použil jsem i stejné prvky CSS3. V patičce obou verzí webové stránky se nachází odkaz pro možnost přepínání mezi mobilní a desktopovou verzí, pro případ potřeby, a abych vyšel vstříc i konzervativním uživatelům.

## 5.2 Druhý model - responzivní přístup

Responzivní přístup znázorňuje schéma na spodní část Obrázku 10. Tento přístup je plně optimální pro veškerá zobrazovací zařízení sám o sobě. Není již třeba vytvářet žádnou zástupnou webovou stránku, ani řešit přesměrování. Je jen jedna webová stránka pro veškerá zařízení. Hlavním prvkem optimalizace zobrazení v rámci tohoto přístupu je responzivní web design. Podstatou responzivního web designu jsou základní pilíře: plovoucí mřížka, flexibilní obrázky, media queries a dynamický obsah, hlouběji se těmto pojmům věnuji v kapitole 2 *Základní pilíře responzivního web designu*. Na následujícím obrázku je náhled responzivní webové stránky při rozlišení 1366 x n (Obrázek 13), které je v současné době nejběžnějším rozlišením (Obrázek 14).



Obrázek 13: ResponsiWeb (rozlišení 1366 x n) – breakpoint >1200px

Screen Resolutions		
1	1366x768	20.56%
2	1920x1080	7.79%
3	1024x768	6.02%
4	1280x800	5.50%
5	360x640	5.37%
6	768x1024	5.35%
7	1280x1024	4.87%
8	1600x900	4.40%
9	1440x900	4.35%
10	320x568	4.20%

Obrázek 14: Výsledky statistiky nejběžnějších rozlišení k březnu 2015

Responzivní webovou stránku ResponsiWeb jsem vytvářel s přístupem desktop-first. Začal jsem tedy s velkými obrazovkami s velkým zobrazovacím rozlišením a postupně jsem přecházel k menším.

Při vytváření webové stránky responzivním přístupem jsem věnoval pozornost především názorné demonstraci funkcí responzivního web designu. Pro základní layout jsem navrhl plovoucí mřížku se čtyřmi sloupci. Co se týká media queries stěžejní bylo stanovit breakpointy. Základní výchozí breakpoint jsem stanovil na velké a extra velké obrazovky s rozlišením nad 1200 pixelů šířky (Obrázek 13). Pro druhý breakpoint jsem zvolil hodnotu rozlišení do 850 pixelů šířky, ta obsahuje jen nepatrné změny a je určena pro některé laptopy a tablety s větším rozlišením v režimu krajina. Další breakpoint jsem vytvořil s cílem na tablety v režimu portrét a smartphony v režimu krajina s hodnotou rozlišení do 590 pixelů šířky (Obrázek 14). Poslední breakpoint pokrývá šířku rozlišení nižší než 590 pixelů a je určen pro smartphony v režimu portrét (Obrázek 15). Flexibilním obrázkem je logo webové stránky v grafickém formátu PNG s použitím alfa kanálu. Velikost obrázku se automaticky přizpůsobuje potřebám layoutu ve všech breakpointech. Co se týká dynamického obsahu, ačkoliv je pouze ilustrativního charakteru, i přesto má vlastnosti responzivního web designu. V rámci obsahu ale také grafiky jsem do layoutu vložil některé inovativní prvky HTML5 a CSS3.



Obrázek 15: ResponsiWeb (rozlišení 768 x n) – breakpoint < 850px



Obrázek 16: ResponsiWeb (rozlišení 360 x n) – breakpoint < 590

Do HTML kódu jsem vložil SVG pro vektorovou grafiku. Jedná se o speciální typ kódu, kterým lze na základě souřadnic vykreslovat objekty přímo do webové stránky. SVG jsem použil při vykreslování kruhů s číslem uprostřed, jež mají za úkol označovat jednotlivé prvky na webové stránce. Následující kód jsem použil pro vykreslení kruhu s číslem jedna v hlavičce webové stránky.

```
<svg class="header-number">
  <circle cx="50%" cy="50%" r="50" fill="#FFFFFF" />
  <text x="27px" y="75px" font-weight="bold"
    font-size="75px" fill="#ED7020">1</text>
</svg>
```

V rámci CSS3 jsem použil grafickou novinku a to stínování prvků. Pomocí vlastnosti `box-shadow` lze kaskádovými styly vykreslit stín blokovému prvku. Tuto vlastnost jsem využil v designování prvků obsahové části. Vytvořil jsem decentní podsvícení prvku a při pohybu kurzorem myši na prvek se intenzita podsvícení lehce zvýší, záměrem je indikovat aktivitu (Obrázek 17).

```
box-shadow: inset 0px 0px 10px 0px white;
```



Obrázek 17: Prvek bez stínování (vlevo), prvek se stínováním - neaktivní (uprostřed), prvek se stínováním - aktivní (vpravo)

## 6 ZÁVĚR

Optimalizace webových stránek pro mobilní zařízení by měla být v dnešní moderní době, kdy smartphony a tablety pomalu nahrazují desktopové počítače, samozřejmostí. Uživatelé Internetu chtějí být mobilní a provozovatelé webových stránek by se měli ve svém vlastním zájmu tomuto trendu přizpůsobit. V této oblasti došlo k velkému průlomů v roce 2010, kdy vznikl responzivní web design. Tento nový návrh řešení problému optimalizace webových stránek představil Ethan Marcotte nejprve v magazínu A List Apart a později také ve své knize Responsive Web Design, kde se věnoval základním pravidlům responzivity, které nazval základními pilíři. Responzivní web design je univerzální metoda pro vytváření webových stránek plně optimalizovaných pro veškerá zobrazovací zařízení.

Řešením problému optimalizace webové stránky pro mobilní zařízení vycházející z možných situací se zabývá praktická část mé bakalářské práce. Existují dva typy situací, ze kterých se vychází při řešení problému optimalizace webových stránek pro mobilní zařízení. První velmi častou situací je, že webová stránka již existuje a je třeba její optimalizaci dodatečně dořešit. V praxi to znamená, že webová stránka je v provozu již delší dobu a výhodnějším řešením optimalizace je připojit k ní další webovou stránku určenou pro mobilní zařízení. Druhá situace nastává v okamžiku plánování tvorby nové webové stránky. V tomto případě je nejlepší volbou metoda responzivního web designu.

Každá ze situací vyžaduje odlišný přístup, metody a myšlení, proto jsem vytvořil praktické řešení, jak docílit optimalizace pro každou situaci zvlášť. Pro situaci, kde je třeba dodatečně optimalizace jsem vytvořil webovou stránku určenou pro desktopové počítače a doplnil ji o mobilní verzi. Obě webové stránky jsem propojil skriptem, který provádí přesměrování na základě detekce mobilního zařízení. V druhé situaci jsem navrhl plně responzivní webovou stránku, při jejíž tvorbě jsem dodržel všechny základní pilíře responzivního web designu.

# SEZNAM ZDROJŮ

## Literární zdroje

- 1 **AMBROSE, Gavin, Paul Harris.** *Basics Design 02: Layout: Second Edition*, Switzerland: AVA Publishing SA 2011. ISBN 978-2-940411-49-8
- 2 **CLARK, Richard, Oli Studholme, Christopher Murphy, Divya Manian.** *Beginning HTML5 and CSS3*, New York: Apress, 2012. ISBN 978-1-4302-2875-2
- 3 **DUCKETT, Jon.** *Beginning Web Programming with HTML, XHTML, and CSS Second Edition*, Indianapolis: Wiley Publishing, Inc., 2008. ISBN 978-0-470-25931-3
- 4 **FIELDING, Jonathan.** *Beginning Responsive Web Design with HTML5 and CSS3*, New York: Apress, 2014. ISBN 978-1-4302-6694-5
- 5 **FIRDAUS, Thoriq.** *Responsive Web Design by Example Beginner's Guide*, Birmingham: Packt Publishing, 2013. ISBN 978-1-84969-542-8
- 6 **GASSTON, Peter.** *The Book of CSS3 - A Developer's Guide to the Future of Web Design, 2nd Edition*, San Francisco: No Starch Press, Inc., 2015. ISBN 978-1-59327-580-8
- 7 **HOGAN, Brian P.** *Html5 And Css3 - Develop with Tomorrow's Standards Today*, 2011. ISBN 978-1-934356-68-5
- 8 **KADLEC, Tim.** *Responzivní design – profesionálně*. Brno: Zoner Press, 2014. ISBN 978-80-7413-280-3
- 9 **LUNN, Ian.** *CSS3 Foundations*, Chichester: John Wiley & Sons Ltd, 2013. ISBN 978-1-118-35654-8
- 10 **MACDONALD, Matthew.** *HTML5: The Missing Manual, 2nd Edition*, Sebastopol: O'Reilly Media, 2013. ISBN 978-1-44936-326-0
- 11 **MARCOTTE, Ethan.** *Responsive Web Design (Brief Books for People Who Make Websites, No. 4)*, New York: A Book Apart, 2011. ISBN 978-0-9844425-7-7
- 12 **MCFARLAND, David Sawyer.** *CSS3: The Missing Manual, Third Edition*, Sebastopol: O'Reilly Media, 2013. ISBN 978-1-449-32594-7
- 13 **PASQUA, Rachel, Noah Elkin.** *Mobile Marketing - An Hour a Day*, Chichester: Wiley, 2013. ISBN 978-1-118-38844-0

- 14 **PETERSON, Clarissa.** *Learning Responsive Web Design - A Beginner's Guide*, Sebastopol: O'Reilly Media, 2014. ISBN 978-1-44936-294-2
- 15 **POWERS, David.** *Beginning CSS3*, New York: Apress, 2012. ISBN 978-1-4302-4473-8
- 16 **SHARKIE, Craig, Andrew Fisher.** *Jump Start Responsive Web Design*, Cambridge: SitePoint, 2013. ISBN 978-0-9873321-7-2
- 17 **SKLAR, Joel.** *Principles of Web Design: The Web Technologies Series 5th edition*, Boston: Cengage Learning, 2011. ISBN 978-1111528706
- 18 **SNIDER, Lesa.** *Photoshop CC: The Missing Manual, First Edition*, Sebastopol: O'Reilly Media, 2013. ISBN 978-1-449-34241-8
- 19 **SOOD, Aditya K, Richard Enbody.** *Targeted Cyber Attacks, 1st Edition - Multi-staged Attacks Driven by Exploits and Malware*, Waltham: Syngress, 2014. ISBN 978-0-12-800604-7
- 20 **WATSON, Adam.** *Learning Modernizr*, Birmingham: Packt Publishing, 2012. ISBN 978-1-78216-022-9
- 21 **WROBLEWSKI, Luke.** *Mobile First*, New York: A Book Apart, 2011. ISBN 978-1-937557-02-7



## Elektronické zdroje

- 1 **ADAPTIC** [online]. 2012 [cit. 2015-03-05] Dostupné z:  
*<http://www.adaptic.cz/znalosti/slovnicek/backend/>*
- 2 **ADAPTIC** [online]. 2012 [cit. 2015-03-05] Dostupné z:  
*<http://www.adaptic.cz/znalosti/slovnicek/frontend/>*
- 3 **AWWWARDS** [online]. 2015 [cit. 2015-03-05] Dostupné z:  
*<http://www.awwwards.com/what-are-frameworks-22-best-responsive-css-frameworks-for-web-design.html>*
- 4 **BÍLEK, Jan.** Mobile-friendly: průvodce optimalizací mobilního webu [online]. 2011 [cit. 2015-02-27] Dostupné z: *<http://www.zdrojak.cz/clanky/mobile-friendly-pruvodce-optimalizaci-mobilniho-webu/>*
- 5 **BOOTSTRAP** [online]. 2015 [cit. 2015-03-11] Dostupné z:  
*<http://getbootstrap.com>*
- 6 **DETECT MOBILE BROWSERS** [online]. 2014 [cit. 2015-02-27] Dostupné z:  
*<http://detectmobilebrowsers.com>*
- 7 **FOUNDATION** [online]. 2015 [cit. 2015-03-11] Dostupné z:  
*<http://foundation.zurb.com>*
- 8 **JAHODA, Bohumil.** Jak na mobilní (responsivní) web [online]. 2013 [cit. 2015-02-14], Dostupné z: *<http://jecas.cz/mobilni-web#media-queries>*
- 9 **LONG, Josh.** I Don't Speak Your Language: Frontend vs. Backend [online]. 2012 [cit. 2015-03-05] Dostupné z: *<http://blog.teamtreehouse.com/i-dont-speak-your-language-frontend-vs-backend>*
- 10 **PC-HELP** [online]. 2014 [cit. 2015-02-27] Dostupné z: *<http://www.pc-help.cz/viewtopic.php?f=61&t=124452>*
- 11 **QUESTIA** [online]. 2008 [cit. 2015-02-20], Dostupné z:  
*<https://www.questia.com/read/1G1-191015232/mobile-devices>*
- 12 **QUESTIA** [online]. 2011 [cit. 2015-02-20], Dostupné z:  
*<https://www.questia.com/read/1G1-252847618/mobile-devices-in-2011>*
- 13 **VOKÁČ, Luděk.** Smartphonům je 20 let. Projděte si jejich historii [online]. 2012 [cit. 2015-02-20], Dostupné z: *[http://mobil.idnes.cz/smartphonum-je-20-let-projdete-si-jejich-historii-fus-/mob\\_tech.aspx?c=A121028\\_220246\\_mob\\_tech\\_vok](http://mobil.idnes.cz/smartphonum-je-20-let-projdete-si-jejich-historii-fus-/mob_tech.aspx?c=A121028_220246_mob_tech_vok)*

- 14 **W3C** [online]. 2015 [cit. 2015-02-22], Dostupné z:  
*<http://www.w3.org/TR/html5/>*

## SEZNAM OBRÁZKŮ

- **Obrázek 1:** Průzkum vývoje počtu uživatelů internetu 1995-2014. Dostupné z: <http://www.pewinternet.org/files/2014/02/01-internet-use-over-time.jpg>. Strana: 11
- **Obrázek 2:** Desktopová verze (vlevo) a mobilní verze webu (vpravo) (<http://seznam.cz>). Dostupné z: <http://www.artweby.cz/blog/mobilni-stranky-nebo-responzivni-web>. Strana: 13
- **Obrázek 3:** Vyjádření podpory CSS3 ve webových prohlížečích. Dostupné z: <http://caniuse.com/#search=CSS3>. Strana: 21
- **Obrázek 4:** Twitter Bootstrap (<http://getbootstrap.com>). Dostupné z: <http://getbootstrap.com>. Strana: 23
- **Obrázek 5:** Zurb Foundation (<http://foundation.zurb.com>). Dostupné z: <http://foundation.zurb.com>. Strana: 24
- **Obrázek 6:** Skeleton (<http://getskeleton.com>). Dostupné z: <http://getskeleton.com>. Strana: 25
- **Obrázek 7:** Porovnání růstu uživatelů přistupujících k internetu z desktopových počítačů a notebooků oproti smartphonům a tabletům za období 2007 - 2015. Dostupné z: <http://digitics.in/2014/05/>. Strana: 31
- **Obrázek 8:** Znázornění plochy viewportu v okně webového prohlížeče. Autor: Marek Straka. Strana: 32
- **Obrázek 9:** Základní používané breakpointy s příkladem zařízení. Dostupné z: <http://www.sonataweb.co.uk/essay/breakpoints.png>. Strana: 33
- **Obrázek 10:** Schématické znázornění dvou přístupů tvorby webové stránky optimalizované pro mobilní zařízení. Dostupné z: <http://www.customfitonline.com/media/106086/responsive-vs-dedicated.jpg>. Strana: 36
- **Obrázek 11:** Desktopová verze webu (rozlišení 1920 x n). Autor: Marek Straka. Strana: 38
- **Obrázek 12:** Mobilní verze webu (rozlišení 360 x n). Autor: Marek Straka. Strana: 40
- **Obrázek 18:** ResponsiWeb (rozlišení 1366 x n) – breakpoint >1200px. Autor: Marek Straka. Strana: 42

- **Obrázek 19:** Výsledky statistiky nejběžnějších rozlišení k březnu 2015.  
Dostupné z: <http://www.w3counter.com/globalstats.php>. Strana: 43
- **Obrázek 20:** ResponsiWeb (rozlišení 768 x n) – breakpoint < 850px. Autor: Marek Straka. Strana: 44
- **Obrázek 21:** ResponsiWeb (rozlišení 360 x n) – breakpoint < 590. Autor: Marek Straka. Strana: 44
- **Obrázek 17:** Prvek bez stínování (vlevo), prvek se stínováním - neaktivní (uprostřed), prvek se stínováním - aktivní (vpravo). Autor: Marek Straka. Strana: 45