MASARYKOVA UNIVERZITA
FAKULTA INFORMATIKY

HTML5, CSS3 téma pro WordPress

BAKALÁŘSKÁ PRÁCE

JIŘÍ BARTOŠÍK

Vedoucí bakalářské práce:
Mgr. Jonáš Ševčík

Vypracoval:
Jiří Bartošík

Brno, jaro 2013

Poděkování:

Děkuji Jonáši Ševčíkovi za spolupráci, odborné vedení a konzultace při zpracování této bakalářské práce.

Prohlašuji, že tato práce je mým původním autorským dílem, které jsem vypracoval samostatně. Všechny zdroje, prameny a literaturu, které jsem při vypracování používal nebo z nich čerpal, v práci řádně cituji s uvedením úplného odkazu na příslušný zdroj.

….....

Shrnutí

Cílem práce je seznámení uživatelů i začínajících programátorů s tvorbou šablony pro redakční systém WordPress. Šablona podporuje novou verzi jazyka HTML, tedy HTML5 a vlastnosti CSS3. Při vytváření návodu a samotné šablony je použit nejnovější WordPress a pomocné moduly pro galerii obrázků a kontaktní formulář. Jednotlivé kroky jsou doprovázeny použitým kódem. V práci jsou řešeny problémy, na které uživatel může při tvorbě narazit. Jako praktická ukázka je vytvořena responsivní šablona splňující všechny předepsané požadavky a mající standardní vlastnosti při tvorbě současného webu.

Klíčová slova

Redakční systém, WordPress, HTML5, CSS3, šablona, kaskádové styly, resposnivní web

Obsah

Motto 1
Úvod 2
1 WordPress 4
1.1 Historie 4
1.2 Šablony 5
1.3 Soubory šablon 5
1.4 Důležité příkazy 6
2 HTML5 8
2.1 Zjednodušený zápis 8
2.2 Podpora 9
2.3 Nové vlastnosti CSS3 9
2.4 Značky <audio> a <video> 13
3 Tvorba šablony 15
3.1 Pravidla šablony 15
3.2 Návrh webu 16
3.3 Struktura šablony stránek 17
3.4 CSS vlastnosti 20
3.5 Úprava NextGEN Gallery 24
3.6 Mobilní šablona 25
3.7 Responsivní web 25
Závěr 28
Literatura 29
Přílohy 31
Obsah elektronických příloh 39

Motto

„Můžete ho milovat nebo nenávidět, ale nemůžete popřít, že HTML4 je nejúspěšnější značkovací jazyk všech dob. HTML5 staví na tomto úspěchu. Nemusíte zahazovat své existující značky. Nemusíte se nic znovu učit. Pokud váš web včera pracoval v HTML4, bude pracovat i dnes v HTML5. Tečka.“[1]

(vlastní překlad, autor Mark Pilgrim) [1]

1 „Love it or hate it, you can’t deny that HTML 4 is the most successful markup format ever. HTML5 builds on that success. You don’t need to throw away your existing markup. You don’t need to relearn things you already know. If your web application worked yesterday in HTML 4, it will still work today in HTML5. Period.“ (Mark Pilgrim) [1]

Úvod

Již dlouhou dobu se zajímám o redakční systémy. Je všeobecně známo, že mezi nejznámější systémy patří WordPress, je uživatelsky přívětivý a jednoduchý na správu obsahu. Zaujaly mě také nové možnosti jazyka HTML5[2] a CSS3[3]. Původně se předpokládalo, že XHTML[4] nahradí HTML, jehož vývoj byl verzí 4.01 ukončen, ale v roce 2007 došlo k založení nové skupiny, která má vytvořit verzi HTML5. Je velmi pravděpodobné, že weboví vývojáři postupně přejdou k vývoji webu na HTML5. Jedná se o dynamicky se rozvíjející sektor. Nyní dochází k tvorbě tohoto standardu a každý týden, měsíc vývojáři přichází s novými funkcemi a vlastnostmi.

Často se dostávám do situace, že potřebuji naprogramovat vlastní šablonu i s podporou současného HTML5 a CSS3. Nejaktuálnější nalezený návod v českém jazyce pochází od Ladislava Janečka. Tento článek se zaměřuje na tvorbu šablony na WordPress, má prozatím dva díly [2, 3]. Článku schází podrobnější kroky, není ještě plně dokončen a pracuje s podporou XHTML. V anglickém jazyce jsem pracoval s návodem [4, 5]. Oba zmíněné články sice postačí při vytváření jednodušší šablony, ale vůbec neřeší podporu HTML5 nebo CSS3. Tuto problematiku jsem nalezl v [6] a v nemalém množství také na internetu. Základy, ale i pokročilé problematiky kaskádových stylů můžeme nalézt v [7], kde bohužel schází aktuální problematika CSS3. Poněvadž jsem nenalezl ucelený český návod na tvorbu šablony pro redakční systém WordPress s podporou HTML5 a CSS3, rozhodl jsem se věnovat tvorbě návodu této šablony.

Cílem práce je vytvořit návod či tutoriál, seznámit uživatele a začínající programátory s tvorbou, postupy při vytváření, problémy, na které mohou narazit při vytváření šablony. Snažím se uvést uživatele do nových metod tvorby webu. Materiálů k této problematice je dostatek, ovšem většinou v anglickém jazyce a bez podpory HTML5 a CSS3. Jako praktickou ukázku vytvářím šablonu

2 HTML5 není prozatím standardizováno, standardizace je plánovaná na rok 2022.
3 CSS3 – kaskádové styly verze 3, jazyk pro popis způsobu zobrazování stránek HTML.
4 XHTML – Extensible Hypertext Markup Language
(rozšiřitelný hypertextový značkovací jazyk).

splňující nová pravidla HTML5 a CSS3. Šablona je podporována nejrozšířenějšími prohlížeči (IE[5] 8+, Mozilla Firefox, Opera a Chrome) a je upravená pro pohodlnější přístup z mobilních telefonů s operačním systémem Android a iOS. V práci chci tedy vypracovat příručku pro českého uživatele, abych pomohl budoucím začátečníkům v této rychle se rozvíjející oblasti.

V první kapitole je uživatel seznámen s redakčním systémem WordPress. Ukáže mu jeho snadnou instalaci a nejdůležitější nastavení. Naučí se jeho správu a nejdůležitější funkce, jako přidávání nových stránek, editaci a další nastavení. Pozornost je věnována především seznámení se s šablonou, její aktivací a nastavením.

HTML5 a nové kaskádové styly CSS3 jsou probrány v druhé kapitole. Uživateli jsou představeny nové vlastnosti CSS3, nové příkazy a elementy, které jsou použity při samotné tvorbě šablony. Představeny jsou nejpoužívanější a nejnovější vlastnosti kaskádových stylů verze 3. Využity jsou také značky <audio> a <video>, které musí splňovat kodeky a zápis podle pravidel HTML5.

Třetí kapitola představuje stěžejní část práce, jelikož je zaměřena na tvorbu šablony. Jsou zde probrána pravidla tvorby šablon pro WordPress. Při tvorbě již pracuji s podporou HTML5 a CSS3 tak, aby byly stránky validní[6]. Jsou zde zmíněny postupy, kroky tvorby a úskalí, s nimiž se může uživatel setkat.

Část třetí kapitoly je zaměřena také na úpravu šablony pro mobilní telefony, podporu systému Android a iOS, ale i na časté problémy s prohlížeči na těchto mobilních zařízeních. Ukazuje se však, že tvorba mobilních šablon je na ústupu a je možné vytvářet web-design tzv. responsivně[7]. Je pravděpodobný odklon od tvorby jednotlivých mobilních šablon k responsivnímu web-designu. Na ukázku je vytvořena responsivní šablona webu pro WordPress s podporou HTML5. V této podkapitole se předpokládá již základní znalost tvorby šablony, proto jsou zde probírány jen hlavní části tvorby responsivního web-designu.

5 IE – Internet Explorer – internetový prohlížeč od společnosti Microsoft.
6 Splňuje požadavky HTML5 a CSS3.
7 Responsivní – přizpůsobení šířky stránky zařízení, na kterém jsou stránky zobrazeny (tzn. proměnlivá šířka stránky).

1 WordPress

WordPress je svobodný open source[8] redakční publikační systém. Jedná se o multiplatformní software. Pracuje tedy na operačním systému, který podporuje PHP a databázi MySQL. Software je napsán v programovacím jazyce PHP a používá databáze MySQL. Tento open source je vyvíjen pod GNU GPL[9]. Momentálně je systém podporován společností Automattic. V roce 2010 byla založena nezisková organizace *The WordPress Foundation*. Práva na WordPress byla předána této organizaci, pro případ, kdyby společnost Automattic přestala Wordpress podporovat či vyvíjet.

1.1 Historie

Redakční systém WordPress vznik roku 2003. Jeho rozšíření nastalo až v roce 2004, kdy došlo ke změně licence. Jednotlivé verze tohoto publikačního systému mají, mimo čísla verze, také kódová jména (podle jazzových hudebníků). V roce 2005 byla vydána verze 1.5 (*Strayhorn*). Přinesla možnost vytvářet statické stránky, šablony, umožnila zapínat a vypínat uživatelské moduly podle potřeby. Také byla opatřena novým tématem vzhledu. Tyto skutečnosti vedly WordPress k jeho rozšíření. Rok 2008 přinesl verzi 2.6 s opravou chyb. Přidáno bylo sledování revizí příspěvků a mnoho dalších funkcí. S mnohem zásadnější změnou přichází pak verze 2.7. Instalace aktualizací a modulů přes webové rozhraní (nikoliv pouze FTP), komentáře ve vláknech a úplně nový vzhled uživatelského panelu v administraci. Verze 2.8 a 2.9 opravují chyby, přidávají funkci koše, hromadné aktualizace, automatické opravy databáze a nebo jednoduchý editor obrázků. Wordpress 3.0 slučuje dvě větve WordPressu, do té doby vyvíjené zvlášť, Wordpress a Wordpress MU (Multi User). Následují verze 3.1, 3.2, 3.3, které přináší opravy a další úpravy. *Gershwin* to je kódové označení verze 3.2. Tato verze přináší optimalizaci funkčnosti, vylepšení aktualizací, nově upravený vzhled administrátorského panelu a zcela novou výchozí šablonu *Twenty Eleven*. V předešlém vydání se jednalo o šablonu *Twenty Ten*. Došlo také k omezení

8 Open source - software s veřejným zdrojovým kódem.
9 GNU GPL (General Public License) - Licence pro svobodný software.

podpory starších verzí jazyka PHP a databáze MySQL. WordPressu 3.2 pracuje minimálně s MySQL 5.0 a PHP 5.2.4. Dne 13. prosince 2011 vyšel Wordpress 3.3 *Sonny* přináší například přepracovanou nápovědu nebo zlepšení pro nahrávání souborů. Návod byl vytvářen pro aktuální WordPress verze 3.4 *Green* vydané 13. června 2012. Responsivní verze šablony je vytvořena po 11. prosinci 2012. Je proto vystavěna již na novější verzi 3.5 *Elvin* [8, 9].

1.2 Šablony

Na stránkách WordPressu nalezneme velké množství šablon. Šablonu nejprve stáhneme a vložíme do adresáře <?php echo home_url(); ?>/wp-content/themes/. Přidanou šablonu nám WordPress automaticky zobrazí v administraci a my ji aktivujeme. Vzhled stránek se přizpůsobí této šabloně, kterou můžeme dále upravovat.

V verzi WordPressu 3.4 jsou nahrány šablony *Twenty Ten* a *Twenty Eleven*. Verze 3.5 obsahuje šablonu *Twenty Twelve*. Na stránkách WordPressu nalezneme více než 1600 dalších vzhledů (http://wordpress.org/extend/themes/).

1.3 Soubory šablon

Každá šablona by měla obsahovat základní soubory. Mezi ně patří header.php, kde nalezneme hlavičku dokumentu, typicky zde začíná HTML dokument a obsahuje nadpis stránek. Stránky jsou zakončeny souborem footer.php a jsou zde zaznamenáni autoři, copyright apod. Index.php udává vzhled úvodní stránky, page.php vzhled ostatních stránek a single.php vzhled příspěvku (obr. 1.3.1).

header.php	hlavička
index.php | page.php | single.php	tělo
footer.php	patička

Obr. 1.3.1: Struktura šablony.

Pokud požadujeme, aby každá stránka obsahovala navigační panel nebo menu přidáváme i soubor slidebar.php, který vkládáme do těla stránek. Obdobně

postupujeme u komentářů (comments.php), archivu (archive.php) nebo vyhledávače (search.php). Každá šablona musí obsahovat soubor functions.php, kde jsou nadefinovány funkce použité v šabloně a samozřejmě soubor definující kaskádové styly (style.css) [10].

Přehled všech souborů, které obsahuje šablona, nalezneme například v CheatSheet od Ekin Ertaç [11]. Seznam zaznamenává všechny soubory šablon, ale také všechny základní funkce a příkazy.

1.4 Důležité příkazy

Později v práci budeme potřebovat znát některé příkazy PHP na vkládání obsahu. Pro vkládání obsahu z WordPressu používáme následující kód (obr. 1.4.1).

```
<?php if (have_posts()) : while (have_posts()) : the_post(); ?>
<?php the_title(); ?>
<?php the_content(); ?>
<?php endwhile; else: ?>
<?php _e('Sorry, no posts matched your criteria.'); ?>
<?php endif; ?>
```

Obr. 1.4.1: Kód pro vkládání obsahu.

Titulek přidáme pomocí <?php the_title(); ?> a obsah pomocí <?php the_content(); ?>. Typicky se používá celý výše zmíněný kód. Pro výpis názvu stránek z WordPressu se používá tag <?php bloginfo('name'); ?>, podobně adresa má příkaz <?php bloginfo('url'); ?>. Hlavička a patička se vkládá pomocí <?php get_header(); ?> a <?php get_footer(); ?>.

Využívat budeme také navigační menu. Kódem (obr. 1.4.2) si nadeklarujeme menu nejprve ve funkcích (functions.php) a poté vložíme do šablony (např. index.php) pomocí kódu (obr. 1.4.3).

```
add_action( 'init', 'register_my_menus' );
function register_my_menus() {register_nav_menus(array(
'galerie' => __( 'Galerie menu' )));}
```

Obr. 1.4.2: Kód vkládaný do function.php.

```
<?php wp_nav_menu(array('menu_id'=>'nav',
'theme_location'=>'galerie'));?>
```

Obr. 1.4.3: Kód navigačního menu.

Tvorbu navigačního menu dokončíme ve WordPressu. V sekci Apperance (vzhled) > Menu vytvoříme nové menu, můžeme jej upravovat, ale také přidávat nové odkazy.

2 HTML5

Původně se předpokládalo, že XHTML nahradí HTML, jehož vývoj byl verzí 4.01 ukončen. V březnu roku 2007 však došlo k založení nové skupiny, která obnovila vývoj jazyka HTML a zahájila práce na verzi 5. HTML5.0 by měla být schválena po připomínkách v roce 2014. Vývojový tým již pracuje také na verzi HTML5.1, kde by měla probíhat standardizace kolem roku 2016. HTML5 by pak mělo být dokončeno kolem roku 2022.

2.1 Zjednodušený zápis

Jazyk HTML5 nejenom nijak nenarušuje předchozí verzi, navíc dává větší svobodu při psaní. Kód, který odpovídal předchozí verzi jazyka, odpovídá také verzi 5. Do nové verze se také zavádí dodatečná syntaxe. Weboví vývojáři si častokrát zjednodušovali práci a nepsali kód podle specifikací HTML. Prohlížeče přesto dokázaly obsah zobrazit správně. Nejčastěji používané části kódu na stránkách, i když původně porušujícího pravidla, našly cestu do verze 5.

Specifikace jazyka zaznamenala značné zjednodušení. Není potřeba zdlouhavý zápis o verzi jazyka a DTD specifikaci dokumentu (začátek dokumentu: *<!DOCTYPE html>*) (obr. 2.1.1). Verze přidává nové elementy pro zpřehlednění zdrojového kódu (*<header>*, *<section>*, *<article>*, *<nav>*, *<footer>*, …) [12].

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN">          <!DOCTYPE html>
```

Obr. 2.1.1: Zjednodušení deklarace HTML4.1 a HTML5.

V nové specifikaci se také nenachází mnoho starých zavedených značek. Mezi zavržené značky pro HTML5 patří například *basefont*, *big*, *center*, *font*, *s*, *strike*, *tt* nebo například *u*. Zrušena byla podpora rámců *frame* a některých atributů.

2.2 Podpora

Prohlížeče ještě plně nepodporují HTML5 a nové vlastnosti kaskádových stylů CSS3. Tyto nedostatky můžeme řešit pomocí prefixů (kapitola 3.6). Další možností je použití polyfillů. Jedná se obvykle o JS[10] knihovnu, jenž přidá podporu nových vlastností. Právě technologie progressive enhancement (postupné vrstvení technologie) často využíváme, převážně při tvorbě webu pro starší prohlížeče. Prohlížeč pak spustí první vlastnost, které rozumí a umí ji přeložit. Pokud prohlížeč novou vlastnost neumí interpretovat, může do stránek vložit polyfill na podporu této vlastnosti. Typicky používané pro spuštění nových vlastností CSS3. V případě, kdy prohlížeč naopak vlastnost správně interpretuje, polyfill se již nepoužije.

2.3 Nové vlastnosti CSS3

Třetí verze CSS přináší mnoho nových vlastností, ale také problémů. Podpora vlastností není stálá a mezi prohlížeči jsou velké výjimky. Rozdílná podpora je zobrazena na některých stránkách [13, 14]. Při psaní nových vlastností CSS proto píšeme kódy pro každý prefix[11]. Prohlížeče potom hledají podporované prefixy a ostatní ignorují. Jedna z vlastností HTML a kaskádových stylů je přeskakování kódu, který neumí interpretovat. Příkladem prefixů může být -webkit-, -moz- nebo -o-. Vývojáři se snaží postupně tato pravidla sjednocovat (standardizovat). Uvedeme a představíme několik novějších vlastností, které se dají používat i na naší šabloně. V komentářích se uvádí podpora ze strany prohlížečů.

Nové použitelné vlastnosti CSS3 [15]:

- ***border-radius*** - zakulacené hrany objektu

```
-webkit-border-radius: 12px; /* Android ≤ 1.6, iOS 1-3.2, Safari 3-4 */
border-radius: 12px;     /* Android 2.1+, Chrome, Firefox 4+, IE 9+,
                                iOS 4+, Opera 10.50+, Safari 5+ */
```

Obr. 2.3.1: Vlastnost border-radius.

10 JS – JavaScript – interpretovaný programovací jazyk, vkládaný přímo do HTML.
11 Prefix – speciálně dodefinovaná vlastnost pro různé prohlížeče.

Border-radius patří mezi nejvíce využívané vlastnosti CSS3. Stačí nastavit velikost zakulacené hrany objektu, například v pixelech.

- ***box-shadow*** – stínování elementu (objektu)

```
.box_shadow {
  -webkit-box-shadow: 0px 0px 4px 0px #ffffff; /* Safari 3-4, iOS 4.0.2 - 4.2, Android 2.3+ */
    box-shadow: 0px 0px 4px 0px #ffffff; /* Opera 10.5, IE9+, Firefox 4+, Chrome 6+, iOS 5 */ }
```

Obr. 2.3.2: Vlastnost shadow.

První hodnota je posun souřadnic podle osy x, druhá hodnota posun od osy y, třetí položkou je „The blur distance“ vzdálenost rozostření, další hodnota udává velikost stínu a poslední barvu. Barvu vybíráme standardně z hodnot barev pomocí šestnáctkového RGB zápisu nebo uvádíme anglický název. Za barvou můžeme také dopsat *inset*, který utvoří stín uvnitř objektu. Třetí a čtvrtou položku můžeme vynechat. Vlastnost *box-shadow* nebude fungovat pro IE9[12] a starší verze prohlížeče.

- ***text-shadow*** - text se stínem

```
text-shadow: 1px 1px 3px #888;        /* Chrome, Firefox 3.5+, IE 10+, Opera 9+, Safari 1+ */
```

Obr. 2.3.3: Vlastnost text-shadow.

- ***rotate*** – rotace elementu div

```
.box_rotate {
  -webkit-transform: rotate(7.5deg);  /* Safari 3.1+, Chrome */
  -moz-transform: rotate(7.5deg);  /* Firefox 3.5-15 */
  -ms-transform: rotate(7.5deg);  /* IE9+ */
  -o-transform: rotate(7.5deg);  /* Opera 10.5-12.00 */
   transform: rotate(7.5deg);  /* Firefox 16+, Opera 12.50+  */ }
```

Obr. 2.3.4: Vlastnost rotate.

Položka *rotate* udává počet stupňů náklonu objektu. Kladné hodnoty udávají náklon doprava, záporné náklon doleva. Vlastnost je problematická z hlediska překrývání objektů.

12 IE – Internet Explorer od společnosti Microsoft, IE9 je značení pro verzi 9.

- ***opacity*** – průhlednost, průsvitnost

```
opacity: 0.5; /* Chrome 4+, Firefox 2+, Safari 3.1+, Opera 9+, IE9+, iOS 3.2+, Android 2.1+ */
/* filter:Alpha(opacity=50);        IE8 and earlier */
```

Obr. 2.3.5: Vlastnost opacity.

Průhlednost objektů je podporována všemi moderními prohlížeči. *Opacity* bez použití je s hodnotou 1. Pro průhlednost pak udáváme hodnotu v rozmezí <0, 1>. IE8 a nižší verze vlastnost *opacity* nepodporují. Pro starší prohlížeče proto používáme *filter:Alpha(opacity=50)*. Tento filtr není validní s HTML5, tento kód CSS proto píšeme do podmíněných komentářů[13].

- ***column-count*** – rozdělení do sloupců

```
.box_columns {
  -webkit-column-count: 2;  -webkit-column-gap: 15px; /* Safari 3, Chrome */
  -moz-column-count: 2;     -moz-column-gap: 15px; /* Firefox 3.5+ */
   column-count: 2;         column-gap: 15px; /* Opera 11+ */  }
```

Obr. 2.3.6: Vlastnost column-count a column-gap.

Vlastnost počet sloupců, která rozděluje například element na více sektorů, není ještě plně podporována. Jako alternativy, pokud potřebujeme podporu starších prohlížečů, se využívá plovoucích elementů (obr.) s velikostí 50% (dva sloupce), 33% (tři sloupce) apod. nebo javascript *columnizer*.

- ***columns*** – rozdělení do sloupců (další řešení)

```
.box_columns2 {
  columns:100px 3;
  -webkit-columns:100px 3;  /* Safari and Chrome */
  -moz-columns:100px 3;     /* Firefox */ }
```

Obr. 2.3.7: Vlastnost columns.

13 Podmíněné komentáře – Komentáře pouze pro některé prohlížeče, typicky IE.
Příklad pro IE8- <!--[if lt IE 8]> kód například CSS vlastnosti <![endif]-->.
(lt = nižší než, lte = nižší nebo rovno, gt = vyšší než)

Pro vlastnost rozdělení do sloupců můžeme využít také zjednodušený zápis *columns*. První místo udává mezeru, druhé počet sloupců. Pro další zjednodušení můžeme vynechat mezeru a psát pouze počet sloupců. Je zřejmé, že zjednodušený způsob je mezi uživateli nejoblíbenější.

- ***background-size*** – velikost obrázku na pozadí s procentuálními hodnotami

```
.box_bgsize {
  -webkit-background-size: 100% 100%; /* Safari 3-4 */
    background-size: 100% 100%; /* Opera, IE9+, Safari 5, Chrome, Firefox 4+ */ }
```

Obr. 2.3.8: Vlastnost background-size.

Využívá se při opakování obrázků na pozadí elementů apod. První položka udává šířku a druhá výšku obrázku na pozadí.

- ***linear-gradient*** – vytváření pozadí objektu pomocí gradientu

```
.box_gradient {
  background-color: #444444;
  background-image:  -webkit-gradient(linear, left top, left bottom, from(#444444), to(#999999));
                                                                    /* Safari 4+, Chrome */
  background-image:  -webkit-linear-gradient(top, #444444, #999999);
                                                  /* Chrome 10+, Safari 5.1+, iOS 5+ */
  background-image:  -moz-linear-gradient(top, #444444, #999999);      /* Firefox 3.6-15 */
  background-image:  -o-linear-gradient(top, #444444, #999999);    /* Opera 11.10-12.00 */
  background-image:   linear-gradient(to bottom, #444444, #999999);
                                                /* Firefox 16+, IE10,     Opera 12.50+ */ }
```

Obr. 2.3.9: Vlastnost linear-gradient.

Na této vlastnosti je vidět ukázka postupné standardizace. Opera i Firefox postupně podporují v novějších verzích přímo CSS vlastnost *linear-gradient*. Vkládají se dvě barevné hodnoty, které se pak mísí a vytvářejí obraz pozadí objektu.

- multiple ***background*** – pozadí z několika obrázků [16]

```
.box_multiple_background {
  background-image: url('mouse-start-left.png'), url('mouse-start-right.png'), url('mouse-end-right.png'),  url('mouse-end-left.png'), url('mouse-top.png'), url('mouse-bottom.png'), url('mouse-left.png'), url('mouse-right.png');
  background-position:left top, right top, right bottom, left bottom, left top, left bottom, left top, right top;
  background-repeat:no-repeat, no-repeat, no-repeat, no-repeat, repeat-x, repeat-x, repeat-y, repeat-y;
```

Obr. 2.3.10: Vlastnost multiple background.

Novou vlastností CSS3 je také pozadí z několika obrázků, viz ukázka (obr. 2.3.10). To využijeme například při vytváření resposivního web-designu, kdy potřebujeme měnit šířku a výšku obrázku.

2.4 Značky <audio> a <video>

První zmínka o elementech pochází z roku 2005. V únoru 2007 dochází k první zkušební implementaci. První implementace proběhla 28. února v prohlížeči Opera, inspirace značkou *<img>*. První prohlížeč s podporou značek *<audio>* a *<video>* byl Safari. Prohlížeč IE přidává podporu jako poslední v březnu 2010. Dnes jsou značky *<audio>* a *<video>* podporované ve všechny nejznámějších prohlížečích.

V dřívější specifikaci HTML se používaly značky *<object>* a *<embed>*. Tyto značky využívaly nejčastěji zásuvných modulů Flash playeru. Po vydání Flash playeru 8 v květnu 2005 se objevil server YouTube. Od roku 2007 se využívají značky *<embed>*. Jejich používání je však komplikované, proto proběhl přesun ke specifikaci HTML5 a novým značkám *<audio>* a *<video>*.

V roce 2007 byl formátem pro video kandidát kodek *Ogg Theora*. Firma Apple formát kritizovala za podřadnou kvalitu a podpořila kodek *H.264*. Alternativou ke kodeku *H.264*, podporující Apple a Microsoft, se stal projekt *WebM* ohlášený 19. května 2010. Výsledkem je podpora tří kodeků: *Ogg Theora*, *WebM* a *H.264*. Díky technologii progressive enhancement (kapitola 2.2) při psaní kódu používáme vždy jeden svobodný kodek *Ogg Theora* nebo *WebM* (podporující prohlížeče FF[14], Opera, Chrome) a kodek *H.264* (podporující prohlížeče IE, Safari a Chrome).

14 FF – zkratka prohlížeče Mozilla Firefox

```
<video width="320" height="240" controls>
  <source src=“zkouska.mp4“>
  <source src=“zkouska.webm“>
  <source src=“zkouska.ogv“>
</video>
```

Obr. 2.4.1: Příklad značky <video>.

U formátu pro audio je situace velmi podobná. Existují tři podporované kodeky. Kodek *Ogg Vorbis* (podporující prohlížeče FF, Opera a Chrome), kodek *WAV* (podporující FF, Safari, Opera a Chrome) a kodek *MP3* (Safari, Chrome a IE). [17]

```
<audio controls>
  <source src=“zkouska.mp3“>
  <source src=“zkouska.ogg“>
  <source src=“zkouska.wav“>
</audio>
```

Obr. 2.4.2: Příklad značky <audio>.

3 Tvorba šablony

Při tvorbě vlastní šablony musíme dodržovat základní pravidla, která pomáhají při pozdějších úpravách nejenom nám, ale i ostatním uživatelům.

Každá šablona musí v hlavičce obsahovat kód na podporu modulů. Abychom předešli problémům, kdy šablona nepodporuje některé moduly, využívá se úpravy některé již vydané šablony, například šablony výchozí. Získáme tím fungující stránky mnohem dříve a pohodlněji. Stránky přitom obsahují všechny náležitosti.

Při vytváření šablony budeme upravovat momentálně výchozí *Twenty Eleven*. Složka šablon se ve WordPressu nachází v adresáři /wp-content/themes/. Nainstalovaný systém obsahuje v základu dvě šablony *Twenty Eleven* a *Twenty Ten*. *Twenty Eleven* si nakopírujeme do adresáře /wp-content/themes/ a přejmenujeme jej, například na *html5*. Od tohoto okamžiku budeme upravovat položky vždy v této složce.

3.1 Pravidla šablony

Informace o šabloně jsou uloženy na začátku souboru kaskádových stylů „style.css“. Upravíme soubor podle naší potřeby, například podle (obr. 3.1.1).

```
/*
Theme Name: HTML5 theme
Theme URI:
Author: student
Author URI: www.neco.cz
Description: HTML5 a CSS3 theme
Version: 1.00
License: GNU General Public License
License URI: license.txt
Tags: html5, css3,
*/
```

Obr. 3.1.1: Informace o šabloně.

Všechny kaskádové styly se načítají ze souboru style.css. Soubor budeme v průběhu tvorby neustále upravovat. V dokumentu ponecháme informace

o šabloně a za tento komentář vložíme doporučený kód CSS tzv. Reset CSS [18]. Kód sjednotí zobrazení v jednotlivých prohlížečích.

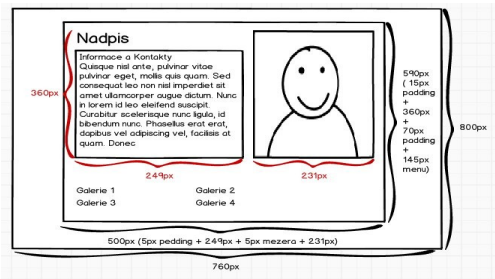
Každá stránka webu[15] obsahuje hlavičku, tělo a patičku dokumentu. V hlavičce webu (header.php) je začátek těla HTML. Do kódu mezi značky *<head>* nebudeme zasahovat. Kód podporuje HTML5 a nastavuje funkce z Wordpressu přímo do stránek. Důležitým souborem je také functions.php, který spolupracuje s moduly a zajišťuje jejich funkčnost. Na konci souboru header.php začíná tělo *<body>*, které není v hlavičce zakončeno. Kód, který se nachází za značkou *<body <?php body_class(); ?>>* smažeme. Odstraníme tak původní hlavičku výchozí šablony. Později se na toto místo vrátíme.

Patička (footer.php) naopak zakončuje tělo *</body>* a celý dokument HTML. Zde se obvykle nachází copyright, práva, rok vydání a další údaje webu. V dokumentu ponecháme pouze komentář a zakončovací značky *</body>* a *</html>*.

3.2 Návrh webu

Před začátkem psaní webu je vhodné si vytvořit návrh našeho webu, prvotní nákres (layout). Vytváření nákresu nám může usnadnit následnou práci. Při psaní máme navíc přehled o jednotlivých částech stránky a můžeme pak případný problém, například s překrýváním, dříve odhalit.

Strukturu si můžeme načrtnout na papír. Daleko vhodnějším způsobem prezentace je využití některého z návrhářských programů. Příkladem může být program Balsamiq, kde rychle a bez předchozích znalostí dokážeme strukturu během několika minut zachytit. (obr. 3.2.1). Návrháři často používají také placené programy. Ukazuje se, že oblíbeným programem je například *PhotoShop*.



Obr. 3.2.1: Návrh v programu Balsamiq [19].

15 Web – Vlastní prezentace Wordpressu.

V návrhu nesmíme zapomínat počítat s okraji (padding a margin). Internetové prohlížeče nepočítají s vnějšími a vnitřními okraji shodně a interpretují je různě.

Naše stránka bude prezentovat obrázky a díla umělce. Stejným způsobem si proto vytvoříme nákres ještě pro galerii a video galerii. Stránky budou prezentovat obrazy a obsahovat videoukázky. V galerii obrázků budeme používat modul[16] *NextGEN Gallery* (verze 1.9.7) spolu s modulem *Fancybox* (verze 1.0.6). *NextGEN Gallery* si později upravíme, aby se shodovala s grafickou podobou webu. Tento postup nám ušetří mnoho času.

Při vytváření dalších návrhů vycházíme z prvního, již vytvořeného nákresu. Hlavní velkosti neměníme, pouze vnitřní rozdělení. Dva sloupce již nepotřebujeme a tak je v našem návrhu můžeme vypustit. Menu pod nimi vynecháme také. Výsledkem je návrh bez vnitřní struktury, kterou již nepotřebujeme, protože *NextGEN Gallery* nám zajistí ostatní formátování. Při vkládání příspěvků pak jednoduše vybereme šablonu pro galerii obrázků nebo videogalerii. Rozdíl je nepatrný, jde pouze o zarovnání. Tato šablona tak zajišťuje konzistenci náhledu obou šablon stránek.

3.3 Struktura šablony stránek

Každá šablona stránky má na začátku souboru definovaný svůj název. Využíváme komentáře v php (obr. 3.3.1). Podle *template name* se nám šablona zobrazí v nabídce šablon ve WordPressu. Při psaní stránky pak jen vybereme z nabídky.

```
<?php /**
 * @package WordPress
 * @subpackage HTML5
 * Template Name: Index
*/; ?>
```

Obr. 3.3.1: Hlavička šablony stránek.

16 Modul – doplňkový (zásuvný) software doinstalovaný do WordPressu, který rozšiřující jeho funkce. Zásuvný modul se anglicky označuje jako Plugin (od slova plug in - zasunout).

Hlavičku a patičku vložíme přímo do šablony (obr. 3.3.2). Mezi tyto příkazy píšeme kód šablony stránky. Tradičně se první stránka webu pojmenovává jako index, proto první kroky úpravy směřují do tohoto dokumentu.

```
<!-- Informace o šabloně strnky -->
<?php get_header(); ?>

<!-- Obsah dokumentu -->

<?php get_footer(); ?>
```

Obr. 3.3.2: Ukázka souboru index.

Předpokládáme základní znalost jazyka HTML a CSS. Vycházíme z návrhu webu, který byl vytvořen v předešlé podkapitole. Základní kostra se bude skládat ze dvou do sebe zanořených značek div pojmenovaných *container* a *container_page*. Začátek těchto kontejnerů (obr. 3.3.3) umístíme do header.php (hlavičky) a ukončíme v souboru footer.php (patičce).

```
<div class="container">
<div class="container_page">
```

Obr. 3.3.3: Vložený kód do header.php

Hlavičku dokumentu máme dokončenou, doplníme ještě patičku, kam vložíme za dvě značky * * speciální značku *<footer>*. V něm se nachází údaje o webu (obr. 3.3.4). Následuje ukončení těla *</body>* a celého dokumentu *</html>*.

```
</div>
</div>
  <footer class="footer">
    Umělecko řemeslné práce a restaurování, Jan Ševčík
  </footer>
</body>
</html>
```

Obr. 3.3.4: Kompletní obsah patičky.

Elementy *<footer>*, *<header>*, *<nav>*, *<section>* a *<article>* se upřednostňují před dříve používanými značkami * *. Section se jako obalová značka nepoužívá, proto zvolíme typ div. Element section označuje sémantický oddíl v obsahu webu.

Podle předlohy je index (úvodní stránka) rozdělen na tři části. Na pravý sloupec s informacemi, levý sloupec s profilovou fotografií a na navigační menu. Pro levou část využijeme element *<article>*, pro pravý sloupec *<section>* a na navigační menu využije elementu *<nav>* (obr. 3.3.5).

```
<?php if (have_posts()) : while
(have_posts()) : the_post(); ?>
  <?php the_title(); ?>
  <?php the_content(); ?>
  <?php endwhile; else: ?>
  <?php _e('Sorry, no posts matched your
criteria.'); ?>
  <?php endif; ?>



  <?php wp_nav_menu(array('menu_id'=>'nav',
'theme_location'=>'galerie'));?>
```

```
<article class="col_left">
 <?php if (have_posts()) : while
(have_posts()) : the_post(); ?>
  <header class="headh1">
    <?php the_title(); ?>
  </header>
  <?php the_content(); ?>
  <?php endwhile; else: ?>
  <?php _e('Sorry, no posts matched your
criteria.'); ?>
  <?php endif; ?>
</article>
<section class="col_right box_round">
</section>

<nav class="nav_menu">
  <?php wp_nav_menu(array('menu_id'=>'nav',
'theme_location'=>'galerie'));?>
</nav>
```

Obr. 3.3.5: Kód šablony index.

Vlastnosti CSS (značené kurzívou) musíme ještě dopsat do style.css. Vlastnosti CSS stylů pojmenováváme tak, aby název vystihoval jejich použití. Zjednodušuje to práci nejenom nám, ale i ostatním.

Obdobně vytvoříme page.php, šablonu výchozí pro všechny ostatní stránky, používaná také modulem *NextGEN Galery*. Kód bude tentokrát složen pouze z jedné části *article* (obr. 3.3.6).

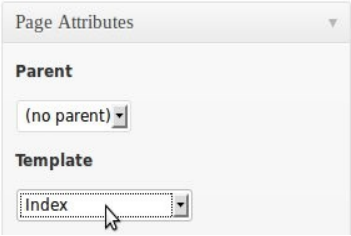
```
<article class="article">
  <?php if (have_posts()) : while (have_posts()) : the_post(); ?>
  <header class="articleh2">
    <?php the_title(); ?>
  </header>
  <?php the_content(); ?>
  <?php endwhile; else: ?>
  <?php _e('Sorry, no posts matched your criteria.'); ?>
  <?php endif; ?>
</article>
   <a class="article_back" href="<?php bloginfo('url'); ?>"> Zpět  </a>
```

Obr. 3.3.6: Kód šablony page.php.

Třetí šablona pro vkládání videí, bude shodná s předešlou šablonou, jen navíc bude obsahovat ještě jednu vlastnost *<article class="article* ***article_video****">* pro přizpůsobení vzhledu u videí. Uložíme ji například jako video.php a v informací o šabloně pojmenujeme *Video Galerie*.

3.4 CSS vlastnosti

Před nastavením CSS vlastností, vytvoříme novou stránku s informacemi ve WordPressu a jako šablonu stránky zvolíme *index* (obr. 3.4.1).



Obr. 3.4.1: CSS pro tělo stránek.

Nastavíme web jako statickou stránku a vybereme ji jako Front page (Úvodní stránku). Komentáře v CSS se píší pomocí zanačky */* komentář */*. Pro rychlé vyhledávání se vlastnosti stylů často rozdělují do částí. Do komentářů se vkládá *=název*. Znak = se v CSS vlastností nepoužívá a proto umožňuje rychlé vyhledávání, například *=Structure*, *=Container* apod. Pro tělo *<body>* a odkazy *<a>* nastavíme velikost, vlastnosti písma a jeho barvu v celém dokumentu (obr. 3.4.2).

```
/* =Structure */
body {
   color: black !important;
   font: 1em Arial, Helvetica, sans-serif !important;
   font-weight: 300;
}
```

```
a{
    padding: 0px;
    text-decoration: none;
    color: black !important;
}
```

Obr. 3.4.2: CSS pro tělo stránek.

Znak *!important* definuje důležitost vlastnosti. Nemusíme tedy stále upravovat předchozí styly v šabloně, ale stačí je přepsat nebo přidat k vlastnostem značku !*important*. Podobně budeme postupovat v dalších elementech.

Strukturu webu tvoříme pomocí návrhu v kapitole 3.2 a nastavíme

zarovnání webu (obr. 3.4.3). Na pozadí umístíme požadovaný obrázek. Obrázky vkládáme do adresáře /wp-content/themes/html5/images/. Barvu pozadí (černou) nastavíme přímo ve vlastnostech šablony přes WordPress.

```
/* =Container */
.container{
        margin: 0 auto;
        position: relative;
        padding: 0px;
        width: 760px;
        height: 800px;
        background-image:url('images/background.gif');
}
```

```
.container_page{
        margin: 0 auto;
        top: 100px;
        position: relative;
        width: 500px;
        height: 590px;
}
```

Obr. 3.4.3: CSS vlastnosti stránky.

Levý a pravý sloupec bude shora odsazen o 15px[17], nastavíme velikosti a na závěr vložíme profilovou fotografii (obr. 3.4.4). V levém sloupci změníme (zmenšíme) písmo na 0.8em[18].

```
.col_right{
        margin-top: 15px;
        width:231px;
        height: 360px;
        float: right;
        background-image:url('images/main.gif');
}
```

```
.col_left{
        margin-top: 15px;
        width: 249px;
        height: 360px;
        float: left;
        padding-left: 5px;
        font-size: 0.8em;
}
```

Obr. 3.4.4: CSS vlastnosti pravého a levého sloupce.

V levém sloupci se nachází vložená stránka, abychom oddělili text od nadpisu přidáme vlastnosti pro nadpis .headh1 (obr. 3.4.5). Pro nadpis používáme novou funkci CSS3 *text-shadow*.

```
/* =Header */
header{
        color: black !important;
}
```

```
.headh1{
        margin:0 auto;
        padding: 0px;
        padding-bottom: 0.4em;
        font-size: 1.8em;
        text-align:left;
        text-shadow: 1px 1px 3px #888;
}
```

Obr. 3.4.5: CSS vlastnosti nadpisu

17 Px – počet pixelů, počet bodů na obrazovce počítače.
18 Em – relativní hodnota, velikost odpovídající šířce velkého písmena „M“.

Obrázku nastavíme zakulacené rohy pomocí efektu *radius* (kap. 2.3), vložit ho můžeme, například mezi *Efekty* (obr. 3.4.6).

```
/* =Effekts */
.box_round {
        -webkit-border-radius: 8px;
        border-radius: 8px;
}
```

Obr. 3.4.6: Efekt border radius.

Nyní přidáme vlastnosti pro naši navigaci. Příkazem *clear* ukončíme *float* (plovoucí sloupce). Navigaci vypisuje WordPress jako seznam. Požadujeme menu bez značek (*list-style*) a zarovnané ve dvou sloupcích. Na dva sloupce využijeme vlastnost *float* a velikostí 50% (obr. 3.4.7). Vlastnost CSS3 *box_columns* není podporována Internet Exlorerem. Tento postup je proto vhodnější.

```
/* =Navigation */  .nav_menu{
        color: black !important;
        clear: both;
        padding-top: 70px;
        padding-left: 5px;
        width: 500px;
        height: 145px;
        text-indent:1.0em;
        line-height: 1.4em;
        list-style: none;  }
.nav_menu ul{
        display: list-item;
        margin:0px 0;
        padding: 0px;
        list-style-type:none;
        list-style: none;
        text-align:left;
        font-weight:bold;
        text-indent: 0em;
}
```

```
.nav_menu ul li {
        float: left;
        width: 50%;
        display: list-item;
        margin:0px 0;
        padding: 0px;
        list-style-type:none;
        list-style: none;
        text-align:left;
        font-weight:bold;  }
.nav_menu ul li a{
        color: black;
}
.nav_menu ul a:hover{
        text-decoration:underline;
        color: black;
}
```

Obr. 3.4.7: CSS vlastnosti navigace.

Nakonec přidáme vlastnosti pro element *footer*, který umístíme pod obrázek. Barva písma bude například stříbrná, aby text nebyl tak výrazný (obr. 3.4.8).

```
/* =Footer */
.footer{
        margin: 0 auto;
        padding: 0px;
        position: relative;
        width: 760px;
        font-size: 0.6em;
        color: silver;
        right: 6em;
        top: -0.6em;
        height: 25px;
        text-align: right;
}
```

Obr. 3.4.8: CSS vlastnosti elementu footer.

Jako druhý soubor upravíme page.php. K této šabloně stránek přidáme CSS vlastnosti *.article*, *.article* p a *.articleh1* pro nadpis (obr. 3.4.9).

```
.article{
        margin: 0 auto;
        width: 470px;
        height: 580px;
        text-align:center;
}
.article p {
        text-indent:1.0em;
        line-height: 2em;
}
.articleh1{
        padding-top: 0.5em;
        padding-left: 0.1em;
        font-size: 1.8em !important;
        margin:0 auto;
        text-align:left;
        font: inherit;
        text-shadow: 1px 1px 3px #888;
}
```

Obr. 3.4.9: CSS vlastnosti pro article.

Šablona bude sloužit pro modul *NextGEN Gallery*. Modul pomáhá spravovat galerii a generuje hotové stránky s náhledy obrázků (viz kapitola 3.5). K tlačítku *Zpět* přidáme vlastnosti. Pomocí CSS umístěné pod galerii (obr. 3.4.10). Doplníme k vlastnostem efekt *hover*, který slouží pro spuštění vlastností po najetí kurzorem na odkaz. Ve formě odkazu může být text, obrázek a nově podle pravidel HTML5 a CSS3 také element či blok.

```
.article_back{
        position:absolute;
        padding: 3px;
        margin: 0 auto;
        float: right;
        right: 1.6em;
        bottom: 0em;
        text-align:center;
        border:1px solid #DDDDDD;
}
.article_back:hover{
        color: white !important;
        background-color: black;
        padding: 3px;
}
.article_back a:hover{
        color: white !important;
}
```

Obr. 3.4.10: CSS vlastnosti pro tlačítko zpět.

Závěrem doplníme vlastnost pro šablonu videí (obr. 3.4.11). Šablona nám bude sloužit pro specifické nastavení i zarovnání pro text a nadpis. Později může šablona stránek také posloužit pro snadnější úpravy.

.article_video articleh1{
padding-bottom: 0.6em;
}
	.article_video articleh1 {
	text-align: left;
	text-indent:0em;
	line-height:2em;
	}

Obr. 3.4.11: CSS vlastnosti pro tlačítko zpět.

3.5 Úprava NextGEN Gallery

K naší práci budeme potřebovat doinstalovat modul *NextGEN Gallery* (aktuálně ve verzi 1.9.7.). Bohužel modul není validní HTML5 dokument a budeme muset upravit jeden soubor CSS. Nalezneme shutter-reloaded.css v adresáři /wp-content/plugins/nextgen-gallery/shutter/ a upravíme. Stačí zakomentovat dva řádky (obr. 3.5.1). Nyní je kód validní pro CSS3. Filtr alpha není podporován v HTML5, byl využíván ve starších verzích (kap. 2.3). *Opacity=50* odpovídá CSS3 vlastnosti *opacity: 0.5*.

```
/* filter:alpha(opacity=80); */
/* filter: alpha(opacity=100); */
```

Obr. 3.5.1: Zakomentované řádky.

Ostatní nastavení modulu jsou pro naši konkrétní šablonu, pokud nepotřebujete upravovat nastavení *NextGEN Gallery* nemusíme v modulu nic dalšího modifikovat. Ostatní nastavení provedeme pak přímo přes administraci.

Pro šablonu provedeme globální nastavení našeho modulu přes WordPress. Velikost Thumbnail (náhledu) bude 135 x 80 px a počet obrázků na stránku nastavíme například na 15 do 3 sloupců. *Slideshow* nastavíme na rozlišení 485 x 450 px a rychlost změny na 5 sekund. Odstraníme efekty, neboť používáme plugin *Fancybox*, který automaticky pracuje se všemi obrázky.

Upravit navigace je možné více postupy. Ukazujeme pouze jednu možnost. Nejprve upravíme soubor gallery.php v adresáři /wp-content/plugins /nextgen-gallery/view/. Přesuneme *Slideshow link* a *Piclense link* dolů pod galerii. Pro zanoření navigačního menu použijeme * * (příloha 3.5.1).

Nakonec opravíme CSS vlastnosti a sladíme tak vzhled s webem. Upravujeme soubor /wp-content/plugins/nextgen-gallery/css/nggallery.css. Části *Gallery navigation*, *Slideshow* a *JS Slideshow* smažeme. Do vlastností CSS šablony vložíme kód přílohy 3.5.2. Důvod vkládání do style.css je mobilní šablona, která bude modifikovat vlastnosti jiným způsobem.

Ukázku námi vytvořené šablony naleznete v příloze 3.5.3, 3.5.4 a 3.5.5.

3.6 Mobilní šablona

Pro vytváření mobilní šablony existuje více způsobů. Jednou z možností je vygenerovaní šablony přímo přes moduly. Takto vytvořené šablony nejsou vždy vyhovující. Vývojáři proto upravují existující šablony a vytváří speciální modifikace pro mobilní zařízení. Můžeme využívat například modul *WordPress mobile pack 1.2.5*, v sekci *Mobile Switcher* nastavíme přepínání mezi šablonami. Modul detekuje zařízení a pokud se jedná o mobilní přístup zobrazí se mobilní modifikace stránek. Modul tak přesměruje Wordpress na druhou šablonu.

Při tvorbě mobilní verze postupujeme obdobně, jako při vytváření verze na desktopy a notebooky. Nejčastější rozdíly jsou velikost písma, odkazů a uspořádání. Navigace webu by se ale neměla znatelně lišit.

Příklad šablony pro náš web nalezneme v souboru přiloženému k práci. Tato mobilní verze je modifikovaná verze předchozí šablony.

3.7 Responsivní web

Nyní již dovedeme navrhnout a naprogramovat jednoduchou šablonu. Je zřejmé, že mobilní verze webu je na ústupu a nahrazuje ho tvorba responsivního web-designu. Responsivní web znamená tvorba šablony webu, která se přizpůsobí velikosti obrazovky tzn. web s proměnlivou šířkou. Nemusíme vytvářet mobilní šablonu, protože máme jednu šablonu pro všechna zařízení.

Budeme již předpokládat základní znalosti tvorby šablon pro WordPress. Web „postavíme“ na nejnovější šabloně *Twenty Twelve*, která byla vydána na začátku prosince roku 2012 spolu s verzí 3.5.1. Obsahuje základní prvky

responsivního web-desingu, který nám ušetří spoustu práce. K tomu nám dopomáhá například cílení CSS stylů pomocí *@media screen* nebo pomocí dalších technik. Používá se často technika „mobile-first“, kdy vytváříme šablonu tzv. „zdola nahoru“. Nejdříve vytváříme mobilní verzi, následně se zaměřujeme na zařízení s větší obrazovkou, jako jsou tablety, notebooky a na závěr desktopy. Technika „mobile-first“ přináší zrychlení načítání stránek na mobilních zařízeních. Ukázku responsivní šablony naleznete v přiloženém souboru, ukážeme si zde jen nejdůležitější úpravy.

Používat budeme také *NextGEN Gallery*, upravíme podle kapitoly 3.5 (viz obr. 3.5.1), tak abychom dosáhli validního modulu s HTML5 a CSS3. Nebudeme používat předem navolené sloupce. Galerie se musí přizpůsobovat velikosti rozlišení. V nastavení galerie změníme počet sloupců a obrázků na „0“. Vypneme také používání *Slideshow*. Na závěr v */wp-content/plugins/nextgen-gallery/css/nggalery.css* upravíme *.ngg-gallery-thumbnail-box* a *.ngg-gallery-thumbnail*, kde zakomentujeme „*float: left;*“.

Pozadí jako orámování stránek můžeme vytvořit při tvorbě responsivního web-designu dvěma způsoby. Klasický způsob znamená obalit stránky znaky * * a u každého zvolit *background-image*. Rohy rámečku se zobrazí vždy. Strany s různou šířkou vytvoříme s pomocí *background-repeat:repeat-x* nebo *repeat-y*. Druhý způsob, novější, využívající zjednodušení zápisu podle CSS3, vytvoříme pomocí tzv. multiple *background*. Tento způsob značně zjednodušuje tvorbu takového rámečku. CSS3 dovoluje použít více pozadí v elementu a příkazy tak řetězit za sebe (kapitola 2.3). Vzhled webu s orámováním (příloha 3.7.1, 3.7.2, 3.7.3 a 3.7.4).

V Galerii používáme novou vlastnost CSS3 *columns*. S cílením přes *@media* pak dosáhneme různého počtu sloupců (obr. 3.7.1). Různý počet sloupců se na závěr upraví automaticky podle šířky internetového prohlížeče (příloha 3.7.5, 3.7.6, 3.7.7 a 3.7.8).

```
@media screen and (min-width: 960px) {
    .ngg-galleryoverview{
        columns:4 ;
        -o-columns: 4; /* Opera */
        -webkit-columns:4 ; /* Safari and Chrome */
        -moz-columns:4; /* Firefox */
```

```
        -moz-column-gap:2px; /* Firefox */
        -webkit-column-gap:2px; /* Safari and Chrome */
        column-gap:2px;
    }
    .site-content {
        width: 960px;
    }
}
```

Obr. 3.7.1: Příklad cílené vlastnosti columns na 4 sloupce.

Při přístupu přes zařízení s menším rozlišením obrazovky dojde ke změně navigačního menu na mobilní menu. Tuto část kódu jsme nemuseli upravovat, šablona ji obsahuje ve výchozím nastavení a v šabloně *Twenty Twelve*.

Průhlednost objektů při najetí myši řešíme pomocí vlastnosti *opacity*. Tu nastavíme na 0.5. Pozadí objektu nastavíme na bílou barvu, aby neprosvítalo pozadí. Výjimky u starších verzí IE řešíme podle kapitoly 2.3 za pomoci podmíněných komentářů.

Video v HMTL5 bez rozměrů velikosti se u responsivního webu automaticky upravuje. Přidáme video za pomoci značek HTML5. Pomocí atributů si nastavíme, jak chceme video zobrazit, popřípadě můžeme přidat více zdrojů videa. Můžeme nechat zobrazit také ovládací prvky.

Závěr

Cílem práce bylo vytvořit návod na tvorbu šablon pro WordPress s podporou současného HTML5 a CSS3. Šablona měla podporovat nejpoužívanější prohlížeče (IE, Mozilla Firefox, Chrome a Safari), a být přístupná z mobilních zařízení s operačním systémem Android a iOS.

Prvním krokem byla tvorba šablony pro desktopy a druhé šablony pro mobilní zařízení. Přepínání bylo řešeno pomocí pluginu (*Mobile Switcher)* na detekci zařízení. Při tvorbě šablony byly používány nové možnosti jazyka HTML5 a nové vlastnosti CSS3. Šablona je vytvářena na redakčním systému WordPressu (verze 3.4.2, u responsivní šablony 3.5.1). Využívány jsou také pluginy *NextGEN Gallery 1.9.12*, *Contact Form Plugin 3.43* a *Captcha 3.3*.

Bakalářská práce je zakončena vytvořením responsivní šablony, kde jsou uživateli představeny nové funkce a možnosti při tvorbě současného moderního webu. Tato šablona upravuje obsah podle šířky monitoru zařízení, které na ni přistupuje. Velká část webů, již dnes díky responsivnímu web-designu, mobilní šablonu nepoužívá. Ukazuje se, že HTML5 je nová rychle se rozšiřující technologie. HTML5 spolu s CSS3 přináší uživateli mnoho nových funkcí a zjednodušuje tvorbu webu i šablon nejenom pro WordPress. Je pravděpodobné, že stále více webů a uživatelů bude používat HTML5. Výhodou je také to, že se uživatelé nemusí učit práci s dalším jazykem. Technologie jako například Java či Flash budou na ústupu, protože HTML5 nabízí podobné vlastnosti a funkce s daleko jednodušší implementací.

Práce nabízí prostor pro budoucí rozšíření v oblasti Media API[19], které přidává další možnosti úpravy audia a videa. Pomocí vlastnosti CSS3 *@font-face* můžeme vytvořit vlastní, například umělecké, písmo pro naši šablonu nebo stáhnout a použít písmo již vytvořené. Můžeme také provést vylepšení responsivního web-designu, například přizpůsobit u menších rozlišení obrazovky tlačítka pro dotykové ovládání nebo více rozčlenit jednotlivé responsivní verze. Užitečným by mohl být také plugin *Lazy Load*[20], který načte v galerii pouze obsah viditelné plochy a ostatní obrázky z galerie stáhne až po srolování stránky.

19 API – Application Programming Interface – rozhraní pro programování aplikací.
20 Lazy Load – Plugin for jQuerty (http://www.appelsiini.net/projects/lazyload).

Literatura

[1] PILGRIM, Mark. Dive Into HTML5. *Dive Into HTML5* [online]. 2009 [cit. 2013-04-16]. Dostupné z: http://diveintohtml5.info

[2] JANEČEK, Ladislav. WordPress – tvorba vlastní šablony. *Programujeme.com* [online]. 4. 11. 2011 [cit. 2012-11-18]. Dostupné z: <http://programujte.com/clanek/2011070500-wordpress-tvorba-vlastni-sablony/>

[3] JANEČEK, Ladislav. WordPress – tvorba vlastní šablony podruhé. *Programujeme.com* [online]. 25. 7. 2012 [cit. 2012-11-18]. Dostupné z: <http://programujte.com/clanek/2012040800-wordpress-tvorba-vlastni-sablony-podruhe/>

[4] NEGOIT, Cosmin. How to Create Your first WordPress Theme: Part 1. *1stwebdesigner.com* [online]. 7. 5. 2010 [cit. 2012-11-18]. Dostupné z: <http://www.1stwebdesigner.com/tutorials/how-to-create-your-first-wordpress-theme-part-1/>

[5] NEGOIT, Cosmin. How to Create Your first WordPress Theme: Part 2. *1stwebdesigner.com* [online]. 21. 5. 2010 [cit. 2012-11-18]. Dostupné z: <http://www.1stwebdesigner.com/tutorials/how-to-create-your-first-wordpress-theme-part-2-2/>

[6] CROFT, Jeff, R LLOYD a Dan RUBIN. *Mistrovství v CSS: pokročilé techniky pro webové designéry a vývojáře*. Vyd. 1. Brno: Computer Press, 2007, 409 s. ISBN 978-80-251-1705-7.

[7] HOGAN, Brian P. *HTML5 a CSS3: výukový kurz webového vývojáře*. Vyd. 1. Brno: Computer Press, 2011, 272 s. ISBN 978-80-251-3576-1.

[8] WordPress. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2012-11-20]. Dostupné z: <http://cs.wikipedia.org/wiki/WordPress>

[9] WordPress. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2012-11-20]. Dostupné z:

<http://en.wikipedia.org/wiki/WordPress>

[10] WORDPRESS. *Stepping Into Templates* [online]. 2012 [cit. 2012-12-07]. Dostupné z: <http://codex.wordpress.org/Stepping_Into_Templates>

[11] WordPress CheatSheet. ERTAÇ, Ekin. *Ekin Ertaç* [online]. 2012 [cit. 2012-12-07]. Dostupné z: <http://ekinertac.com/>

[12] W3C School: HTML5 Introduction. *W3cschool.com* [online]. 2013 [cit. 2013-03-04]. Dostupné z: <http://www.w3schools.com/html/html5_intro.asp>

[13] CSS3 Browser Support Reference. *W3schools.com* [online]. 2012 [cit. 2012-11-27]. Dostupné z: <http://www.w3schools.com/cssref/css3_browsersupport.asp>

[14] HTML5 & CSS3 Support. *Findmebyip.com* [online]. 2012 [cit. 2012-11-27]. Dostupné z: <http://www.findmebyip.com/litmus/>

[15] CSS3, Please!. *Css3please.com* [online]. 2012 [cit. 2012-12-02]. Dostupné z: <http://css3please.com/>

[16] SMOLA, Martin. Novinky v CSS3: Multiple Background. *Root.cz* [online]. 12. 9. 2012. 2012 [cit. 2013-03-27]. Dostupné z: http://www.root.cz/clanky/novinky-v-css3-multiple-background/

[17] PFEIFFER, Silvia. *HTML5 - audio a video: kompletní průvodce*. Vyd. 1. Brno: Zoner Press, 2011, 350 s. ISBN 978-80-7413-147-9.

[18] MEYERA, Erica. CSS Tools: Reset CSS. *Meyerweb.com* [online]. 2011 [cit. 2012-12-12]. Dostupné z: <http://meyerweb.com/eric/tools/css/reset/index.html>

[19] Balsamiq. *Balsamiq* [online]. 2012 [cit. 2012-11-23]. Dostupné z: <http://www.balsamiq.com/>

[20] MICHÁLEK, Martin. *Bez HTML5/CSS3 to nejde*. 2012. Dostupné z: <http://www.slideshare.net/machal/muni-2012-12publicpdf>

[21] DOMES, Martin. *333 tipů a triků pro CSS: [sbírka nejužitečnějších návodů pro váš web]*. Vyd. 1. Brno: Computer Press, 2009, 272 s. ISBN 978-80-251-2360-7.

Přílohy

```
<?php if (!defined ('ABSPATH')) die ('No direct access allowed'); ?><?php if (!empty ($gallery))
: ?>

<div class="ngg-galleryoverview" id="<?php echo $gallery->anchor ?>">


	<!-- Thumbnails -->
	<?php foreach ( $images as $image ) : ?>

	<div id="ngg-image-<?php echo $image->pid ?>" class="ngg-gallery-thumbnail-box"
<?php echo $image->style ?> >
		<div class="ngg-gallery-thumbnail" >
			<a href="<?php echo $image->imageURL ?>" title="<?php echo
$image->description ?>" <?php echo $image->thumbcode ?> >
				<?php if ( !$image->hidden ) { ?>
				<img title="<?php echo $image->alttext ?>" alt="<?php echo
$image->alttext ?>" src="<?php echo $image->thumbnailURL ?>" <?php echo $image->size ?
> />
				<?php } ?>
			</a>
		</div>
	</div>

	<?php if ( $image->hidden ) continue; ?>
	<?php if ( $gallery->columns > 0 && ++$i % $gallery->columns == 0 ) { ?>
		<br style="clear: both" />
	<?php } ?>

	<?php endforeach; ?>

<?php if ($gallery->show_piclens) { ?>
	<!-- Piclense link -->
	<div class="piclenselink">
		<a class="piclenselink" href="<?php echo $gallery->piclens_link ?>">
			<?php _e('[View with PicLens]','nggallery'); ?>
		</a>
	</div>
<?php } ?><div class="location">

<?php if ($gallery->show_slideshow) { ?>
	<!-- Slideshow link -->
	<div class="slideshowlink">
		<a class="slideshowlink slideshowlink_a" href="<?php echo $gallery-
>slideshow_link ?>">
			<?php echo $gallery->slideshow_link_text ?>
		</a>
	<!-- Pagination -->
	<?php echo $pagination ?>
	</div>
<?php } ?>
</div>
</div>
<?php endif; ?>
```

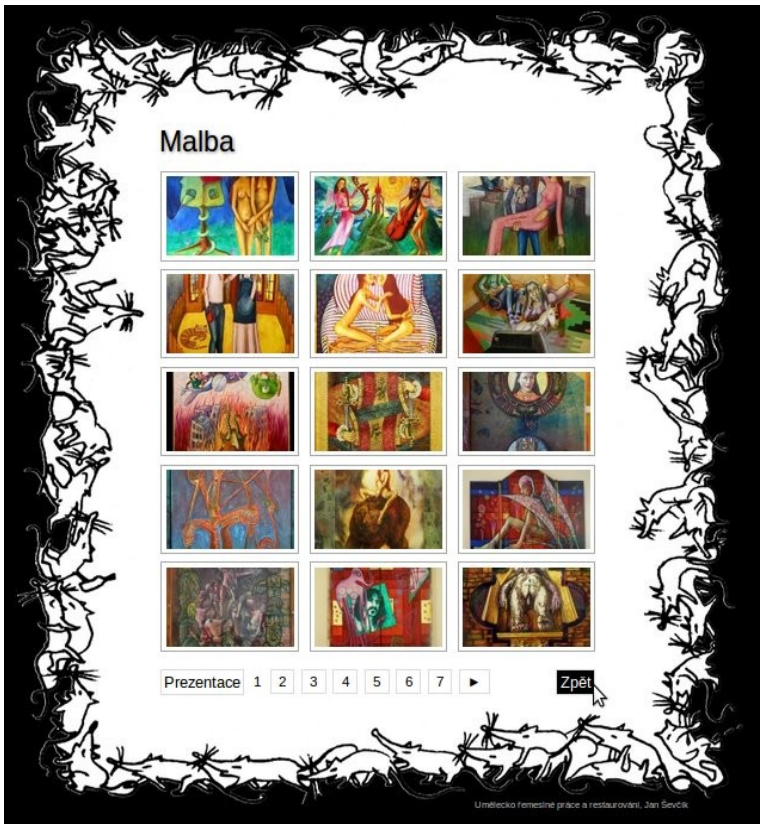
Příloha 3.5.1: Úprava souboru gallery.php.

```
/* ----------- Gallery navigation -------------*/
.ngg-navigation {
	font: 1em Arial, sans-serif !important;
	font-size:0.9em !important;
	text-align:left;
	margin: 0px auto;
	margin-left: 5px;

}
.location{
	clear:both !important;
	width: 490px;
	position: absolute;
	bottom: 0px;

}
.current, .more{
	color: black !important;
}
.ngg-navigation span {
	font: 1em Arial, sans-serif !important;
	font-size:1em !important;
	float: left;
	font-weight:bold;
	margin: 0px auto;
	margin-left: 5px;
	margin-top: 3px;
	padding: 5px;
}

.ngg-navigation a.page-numbers,
.ngg-navigation a.next,
.ngg-navigation a.prev,
.ngg-navigation span.page-numbers,
.ngg-navigation span.next,
.ngg-navigation span.prev {
	border:1px solid #DDDDDD;
	margin-right:3px;
	padding:3px 8px;
   text-decoration: none;
}
.ngg-navigation a.page-numbers:hover,
.ngg-navigation a.next:hover,
.ngg-navigation a.prev:hover,
.ngg-navigation span.page-numbers:hover,
.ngg-navigation span.next:hover,
.ngg-navigation span.prev:hover {
	background-color: black;
	color: #FFFFFF !important;
	text-decoration: none !important;
}
```

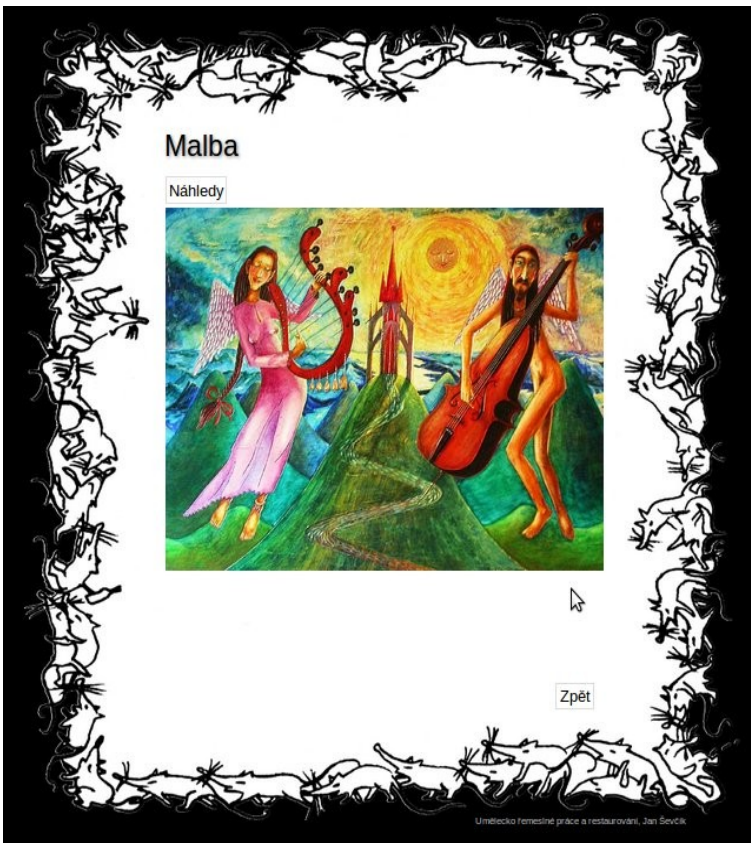
```
/* ----------- Slideshow -------------*/
.slideshow {
	margin-left: auto;
	margin-right: auto;
	outline: none;
}

.slideshowlink {
	clear:both !important;
	padding: 0px;
}

.slideshowlink a{
	border:1px solid #DDDDDD;
	margin: 0px auto;
	margin-left: 4px;
	margin-top: 4px;
	padding: 3px;
	text-align:left;
	float: left;
}

.slideshowlink_a:hover {
	background-color: black;
	color: #FFFFFF !important;
	text-decoration: none !important;

}

/* ----------- JS Slideshow -------------*/
.ngg-slideshow {
   overflow:hidden;
   position: relative;
	margin: 0px auto;
	margin-left: 5px;
	margin-top: 4px;
	padding: 5px;
	text-align:left;
	float: left;
}

.ngg-slideshow * {
   vertical-align:middle;
}

/* See also :
http://www.brunildo.org/test/img_center.html */
.ngg-slideshow-loader{
   display: table-cell;
   text-align: center;
   vertical-align:middle;
}
.ngg-slideshow-loader img{
   background: none !important;
   border: 0 none !important;
   margin:auto !important; }
```

Příloha 3.5.2: Upravený CSS kód pro nggallery.css.

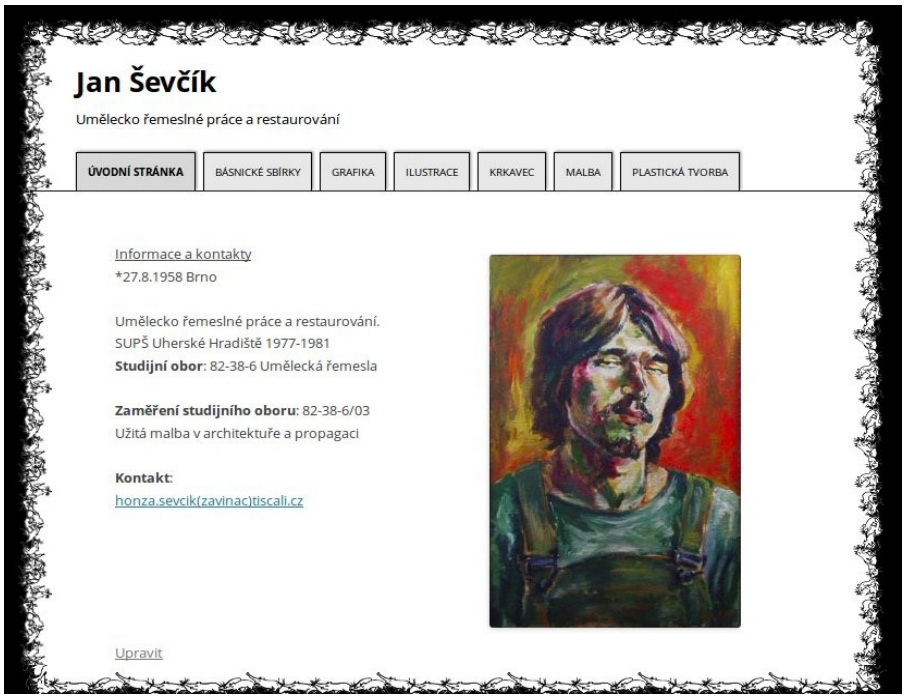
Příloha 3.5.3: Ukázka úvodu stránek.



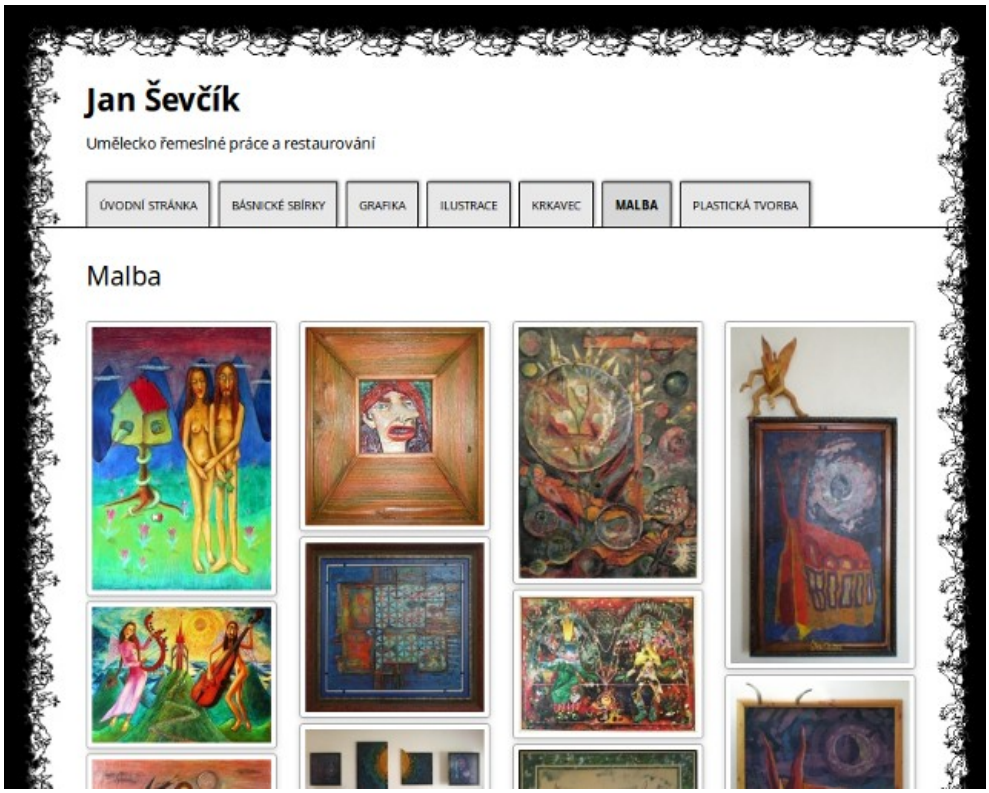
Příloha 3.5.4: Ukázka galerie stránek.



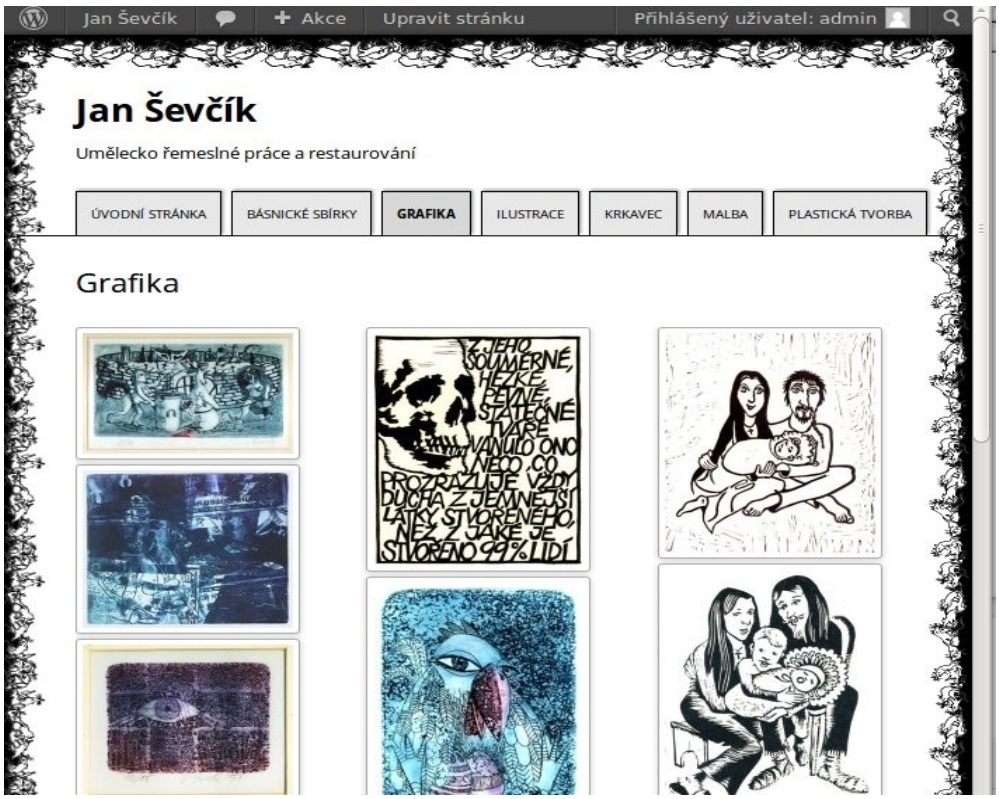
Příloha 3.5.5: Ukázka Slideshow (prezentace) obrázků.



Příloha 3.7.1: Ukázka úvodu stránek responsivního webu.



Příloha 3.7.2: Ukázka galerie stránek responsivního webu.

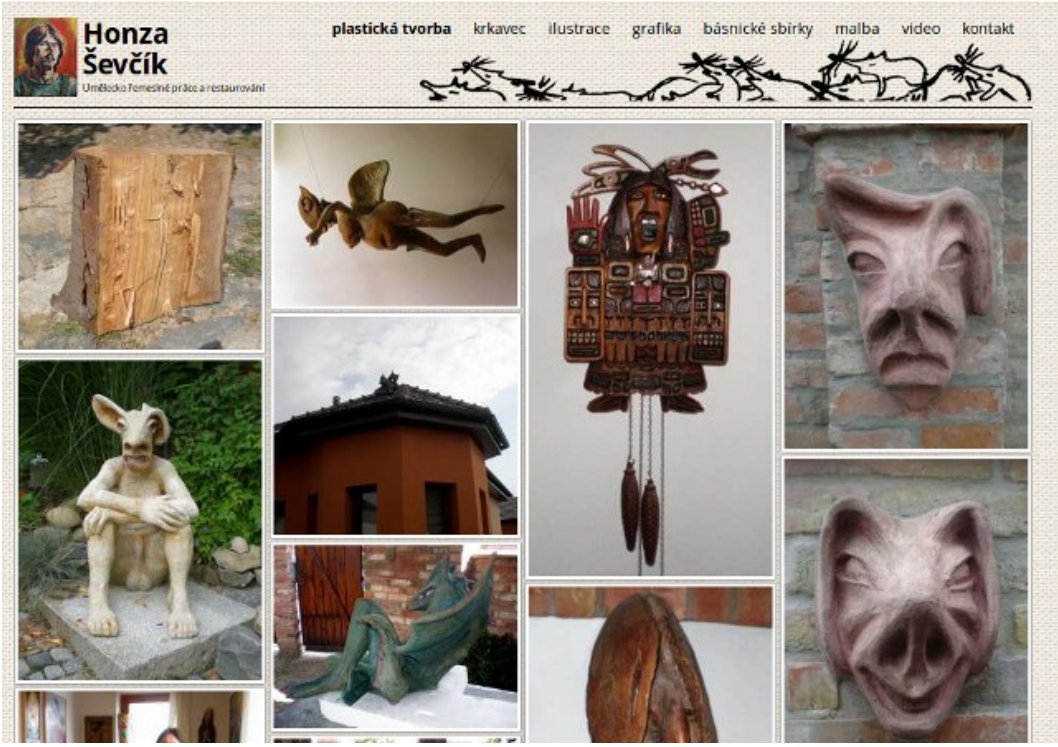


Příloha 3.7.3: Ukázka galerie stránek responsivního webu.

```
/* =CLASSIC BACKGROUND */
/* top mouse */
.row-top{
        width: 100%;
        height: 28px;
        background-image:url('mouse-top.png');
        background-repeat:repeat-x; }
.row-top-left{
        float: left;
        background-image:url('mouse-top-
left.png');
        background-repeat: no-repeat;
        background-color: #000000;
        width: 28px;
        height: 28px; }
.row-top-right{
        float: right;
        background-position: right;
        background-image:url('mouse-top-
right.png');
        background-repeat: no-repeat;
        background-color: #000000;
        width: 28px;
        height: 28px; }
/* left mouse */
.mouse-left {
        background-image:url('mouse-left.png');
        background-repeat:repeat-y; }
/* right mouse */
.mouse-right {
        background-position: right;
        background-image:url('mouse-right.png');
        background-repeat:repeat-y; }
/* bottom mouse */
.row-bottom{
        width: 100%;
        height: 28px;
        background-image:url('mouse-bottom.png');
        background-repeat:repeat-x; }
.row-bottom-left{
        float: left;
        background-image:url('mouse-bottom-
left.png');
        background-repeat: no-repeat;
        background-color: #000000;
        width: 28px;
        height: 28px; }
.row-bottom-right{
        float: right;
        background-position: right;
        background-image:url('mouse-bottom-
right.png');
        background-repeat: no-repeat;
        background-color: #000000;
        width: 28px;
        height: 28px; }
```

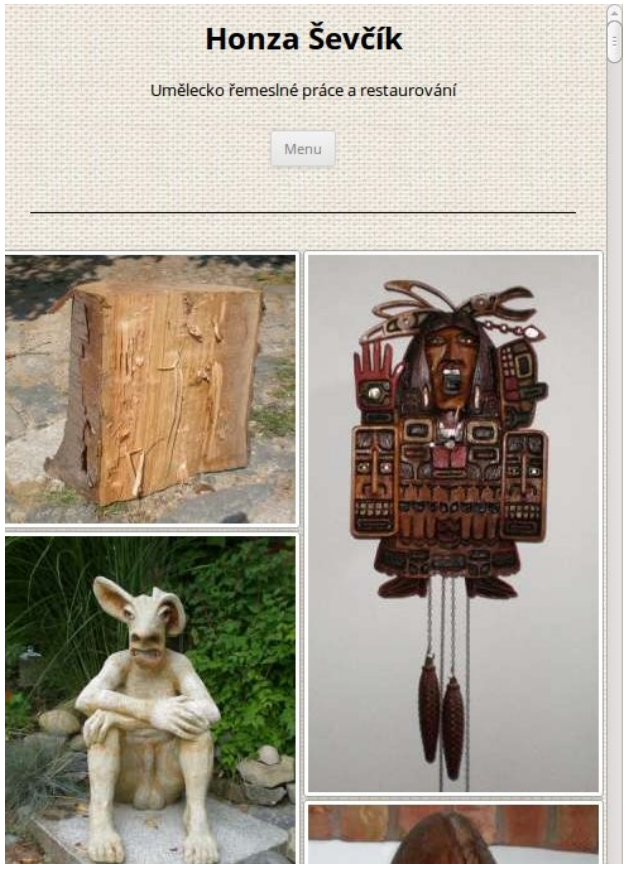
```
/* =MULTIPLE BACKGROUND */

.site{

background-image: url('mouse-start-
left.png'), url('mouse-start-right.png'),
url('mouse-end-right.png'),  url('mouse-
end-left.png'), url('mouse-top.png'),
url('mouse-bottom.png'), url('mouse-
left.png'), url('mouse-right.png');

background-position:left top, right top,
right bottom, left bottom, left top, left
bottom, left top, right top;

background-repeat:no-repeat, no-repeat,
no-repeat, no-repeat, repeat-x, repeat-x,
repeat-y, repeat-y;

}
```

Příloha 3.7.4: Ohraničení stránek, použití klasického modelu a multiple background.

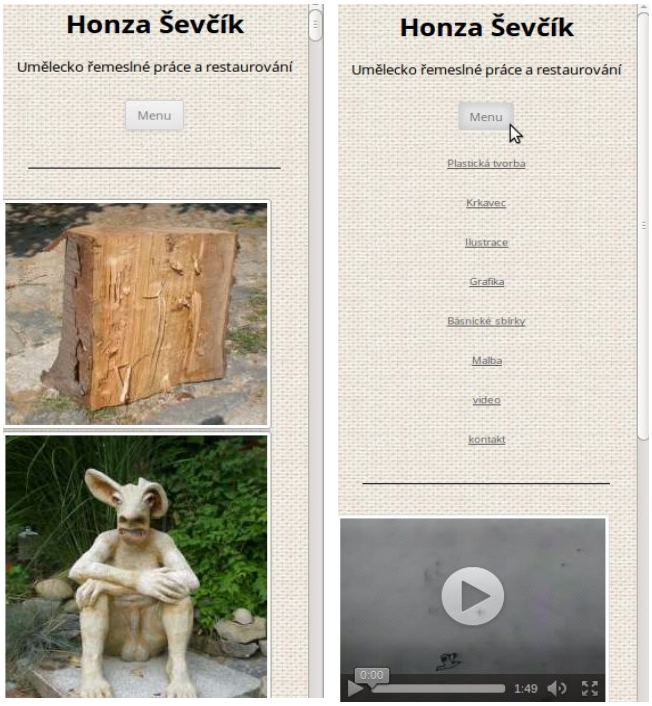


Příloha 3.7.5: Ukázka vytvořené responsivní šablony.

Příloha 3.7.6: Ukázka vytvořené responsivní šablony.



Příloha 3.7.7: Ukázka vytvořené responsivní šablony.



Příloha 3.7.8: Ukázka vytvořené responsivní šablony.

Obsah elektronických příloh

Součástí práce jsou tyto elektronické přílohy:

- Práce ve formátu pdf.
- Šablona 1 (šablona z kapitol 3.3, 3.4, 3.5 a 3.6).
- Šablona 2 (šablona s ukázkou multiple background, kapitola 3.7).
- Šablona 3 (finální responsivní šablona, kapitola 3.7).