

Česká zemědělská univerzita v Praze

Technická fakulta

Katedra technologických zařízení staveb

**Optimalizace internetové prezentace**

*DIPLOMOVÁ PRÁCE*

Vedoucí diplomové práce: Ing. Michal Hruška, Ph.D.

Diplomant: Bc. Petr Vosátka

Praha 2013

## **Prohlášení**

*Prohlašuji, že jsem diplomovou práci na téma „**Optimalizace internetové prezentace**“ zpracoval samostatně a použil jen literaturu uvedenou v seznamu literatury.*

*V Praze dne 9. března 2013*

.....

*Bc. Petr Vosátka*

## **Poděkování**

*Děkuji panu Ing. Michalovi Hruškovi za velmi užitečnou metodickou pomoc, kterou mi poskytl při zpracování mé diplomové práce.*

*Dále bych rád poděkoval svým rodičům a přátelům za morální i finanční pomoc při studiu.*

*Ing. Michaele Vrbové,*

*za to, že jsi.*

## **Abstrakt**

Tato diplomová práce navazuje na práci bakalářskou. Vnáší do ní nové trendy, nové styly a podstatně ji rozšiřuje. Pojednává o kaskádových stylech, jejich vývoji od CSS1, přes letitý standard CSS2.1 až po současnost v podobě verze CSS3. Práce také obsahuje různé příklady formátování HTML dokumentů, ať již klasickou cestou prostého formátování, či právě rozměrovou optimalizací pomocí kaskádových stylů. V praktické části ukazuje vývoj a optimalizaci webu občanského sdružení SK Prague United, kde působím mimo jiné i jako webmaster.

## **Klíčová slova**

optimalizace webu, prohlížeč, grafika, kaskádové styly, rozměrová optimalizace

## **Summary**

This diploma thesis substantially extends the information presented within my previous bachelor project and includes information on new trends and styles. This study relates to cascading style sheets and the development from CSS1, through the established standards of CSS2.1 to the current version of CSS3. This paper also encompasses various examples of HTML formatting, either via typical simple formatting or by spatial formatting using cascade styles. The applied section of this diploma thesis relates to the practical development and optimisation of these areas in the author's capacity as webmaster for NGO SK Prague United.

## **Key words**

web optimization, browser, graphics, cascade styles, dimensional optimization

# Obsah:

<b>1. ÚVOD – PROČ VZNIKLY KASKÁDOVÉ STYLTY .....</b>	<b>6</b>
<b>2. VYMEZENÍ POJMŮ .....</b>	<b>8</b>
2.1 DŮLEŽITÉ GRAFICKÉ FORMÁTY .....	8
2.2 ŠIRŠÍ FORMÁTOVACÍ MOŽNOSTI CSS .....	9
2.3 SNADNÁ TVORBA A ÚDRŽBA KONZISTENTNÍHO STYLU .....	9
2.4 ODDĚLENÍ STRUKTURY A STYLU .....	10
2.5 VYŠŠÍ PŘÍSTUPNOST DOKUMENTŮ .....	10
2.6 DYNAMICKÁ PRÁCE SE STYLTY .....	10
2.7 FORMÁTOVÁNÍ XML DOKUMENTŮ .....	10
<b>3. POPIS ZPŮSOBŮ OPTIMALIZACE WEBOVÝCH PREZENTACÍ A POUŽITÝCH TECHNOLOGIÍ .....</b>	<b>11</b>
3.1 OBECNÉ ROZDĚLENÍ .....	13
3.2 KASKÁDA .....	16
3.2.1 <i>Váhy tabulek stylů</i> .....	16
3.2.2 <i>Příkaz !important</i> .....	16
3.2.3 <i>Pravidlo @import</i> .....	17
3.2.4 <i>Pořadí kaskády</i> .....	17
3.3 VIZUÁLNÍ FORMÁTOVACÍ MODEL CSS .....	18
3.4 ZOBRAZOVÁNÍ DOKUMENTU .....	18
3.4.1 <i>Zobrazování prvků</i> .....	18
3.4.2 <i>Rámový model, rozměry rámu</i> .....	19
3.4.3 <i>Omezující bloky</i> .....	19
3.4.4 <i>Generování rámu</i> .....	19
3.5 PSEUDOPRVKY A PSEUDOTŘÍDY .....	20
3.6 MÉDIA .....	21
3.7 AKTUÁLNÍ ROZŠÍŘENÍ A NOVINKY CSS3 .....	23
3.7.1 <i>Zaoblené rohy</i> .....	23
3.7.2 <i>Stíny u blokových prvků</i> .....	24
3.7.3 <i>Stíny u textů</i> .....	25
3.7.4 <i>Transformace</i> .....	25
<b>4. NÁVRH OPTIMALIZACE KONKRÉTNÍ WEBOVÉ APLIKACE .....</b>	<b>27</b>
4.1 ZÁZEMÍ .....	28
4.1.1 <i>Hardwarová výbava serveru</i> .....	28
4.1.2 <i>Umístění serveru</i> .....	29
4.1.3 <i>Konektivita</i> .....	29
4.1.4 <i>Napájení</i> .....	29
4.1.5 <i>Zabezpečení přístupu</i> .....	30
4.1.6 <i>Podmínky prostředí</i> .....	30
4.2 HOSTING .....	30
4.3 PRVKY WEBOVÉ APLIKACE .....	31
4.3.1 <i>WordPress</i> .....	32
4.3.2 <i>Obecná šablona stránky</i> .....	33
4.3.2.1 <i>Zdrojový kód šablony – style.css</i> .....	34
4.3.2.2 <i>Zdrojový kód šablony – header.php</i> .....	36
4.3.2.3 <i>Zdrojový kód šablony – index.php</i> .....	36
4.3.2.4 <i>Zdrojový kód šablony – sidebar.php</i> .....	36

4.3.2.5	Zdrojový kód šablony – footer.php .....	37
4.3.2.6	Zdrojový kód šablony – single.php .....	38
4.3.2.7	Zdrojový kód šablony – page.php .....	39
4.3.2.8	Zdrojový kód šablony – archive.php .....	39
4.3.2.9	Zdrojový kód šablony – functions.php .....	40
4.3.2.10	Zdrojový kód šablony – comments.php .....	40
4.4	DŮLEŽITÉ PLUGINY .....	41
4.4.1	<i>WP HTTP Compression</i> .....	41
4.4.2	<i>W3 Total Cache</i> .....	41
4.4.3	<i>NextGEN Gallery</i> .....	42
4.4.4	<i>Float Left Right Advertising</i> .....	42
4.5	TESTOVACÍ VLASTNOSTI .....	43
4.6	PROHLÍŽEČE .....	44
4.6.1	<i>Prohlížeče postavené na jádře Trident</i> .....	46
4.6.2	<i>Prohlížeče postavené na jádře Gecko</i> .....	47
4.6.3	<i>Prohlížeče postavené na jádře WebKit</i> .....	48
4.6.4	<i>Prohlížeče postavené na jádře Presto</i> .....	49
4.6.5	<i>Ostatní grafické prohlížeče</i> .....	49
4.6.6	<i>Textové prohlížeče</i> .....	49
5.	ZHODNOCENÍ VÝSLEDKŮ .....	50
6.	POUŽITÁ LITERATURA .....	51
	<b>SEZNAM PŘÍLOH .....</b>	<b>53</b>
	<i>Příloha č.1 – header.php</i> .....	53
	<i>Příloha č.2 – index.php</i> .....	56
	<i>Příloha č.3 – archive.php</i> .....	61
	<i>Příloha č.4 – functions.php</i> .....	65
	<i>Příloha č.5 – HTML</i> .....	85
	<i>Příloha č.6 – HTML 3.2</i> .....	85
	<i>Příloha č.7 – HTML 4.01 a CSS</i> .....	86
	<i>Příloha č.8 – XHTML 1.0</i> .....	86
	<i>Příloha č.9 – XML a Stylesheet</i> .....	87
	<i>Příloha č.10 – XML a CSS</i> .....	87
	<i>Příloha č.11 – XSLT</i> .....	88
	<i>Příloha č.12 – XSL:FO</i> .....	88
	<i>Příloha č.13 – možnosti orámování CSS</i> .....	89
	<i>Příloha č.14 – struktura a hodnocení testovaného webu</i> .....	89
	<b>SEZNAM OBRÁZKŮ</b>	
	Obr. 1 – hlavní domovská stránka .....	27
	Obr. 2 – webserver v rackovém provedení, Zdroj: endora.cz .....	28
	Obr. 3 – domovská stránka administrátora .....	31
	Obr. 4 – základní šablona domovské stránky .....	33
	Obr. 5 – geografické rozdělení nejpoužívanějších prohlížečů, Zdroj: gs.statcounter.com .....	44
	Obr. 6 – ideální výsledek Acid3 testu .....	45

# 1. Úvod – proč vznikly kaskádové styly

„Původním záměrem jazyka HTML byl, kromě definování vazeb mezi dokumenty, především popis jejich struktury. Určit, která část je nadpisem, která odstavcem nebo tabulkou, co je citací z jiného zdroje, co je jménem autora, případně který úsek je seznam a který jeho položky, by bylo, mírně řečeno, problematické. To, jak se tyto jednotlivé úseky mají zobrazit, bylo definováno jinde.

Mezitím ovšem začalo vznikat množství doplňků a rozšíření jazyka HTML nad rámec standardu. Objevilo se mnoho jeho druhů a poddruhů a postupně přibýlo velké množství funkcí a značek, které slouží jen pro účely formátování dokumentů. S přibývajícimi nadstandardními rozšířeními se z dokumentů postupně začala vytrácet jejich struktura. Automatické zpracování dokumentů je stále složitější. Kvůli zmizelé struktuře se ve stránkách vedle sebe objevují úseky, které spolu vůbec nesouvisejí. Někdy je naopak souvislý text rozdělen na několika vzdálených místech dokumentu. Vyhledávače na internetu si s takovým obsahem nedokáží poradit, ztrácí se přenositelnost a kompatibilita. Stránky jsou formátovány jen pro jeden typ počítače, nebo pro jednu verzi jednoho konkrétního prohlížeče. Handicapování uživatelé, uživatelé používající jiné prohlížeče nebo jiné zařízení, nemusí stránku vůbec přečíst.

Novou technologií je tabulka kaskádových stylů (CSS), která umožňuje formátování dokumentů, definuje způsob jejich prezentace na koncových zařízeních, popisuje podobu stránek a styl jednotlivých prvků a přitom nijak neovlivňuje obsah dokumentů samotných. Při tvorbě stránek s pomocí CSS lze použít barvy a obrázky na pozadí, snadněji definovat písma, rozestupy mezi prvky, rámečky, apod.

Ještě bych se zastavil u návrhu webových stránek, kde pokud je začínáme tvořit od nuly, je dobré si promyslet, jaké aplikace k tomu využijeme, jelikož máme na výběr využít komerční produkt, open source řešení nebo čistě vlastní znalosti. Poté je třeba se zaměřit na koncové zařízení, kde bychom chtěli daný dokument prezentovat. Zde máme celkem široký výběr zahrnující osobní počítač, kapesní počítač neboli tzv. PDA, palmtop, mobilní telefon, WebTV, herní konzoli či úplně jiné alternativní zařízení jako je např. lednička.



## 2. Vymezení pojmů

Před nástupem CSS se HTML dokumenty formátovaly výhradně prostředky jazyka HTML, který k tomu nabízel poměrně široké spektrum značek a atributů.

CSS k formátovacím prostředkům představuje vhodnou alternativu, případně doplněk. Význam CSS je však širší. Jeho použití v praxi umožňuje:

- Širší formátovací možnosti
- Snadnou tvorbu a údržbu konzistentního stylu
- Oddělení struktury a stylu
- Vyšší přístupnost dokumentů
- Dynamické práce se styly
- Formátování XML dokumentů

### 2.1 Důležité grafické formáty

Pro přenos barevných fotografií a jejich následnou prezentaci používáme nejčastěji formáty JPEG (Joint Photographics Experts Group) je díky své dobré kompresi používáno ve většině digitálních fotoaparátů, PNG (Portable Graphics Network) je hlavně zaměřen na přenos obrázků v síti a je podporován konsorciem W3C, GIF (Graphics Interchange Format) je nejstarší a jeden z nejoblíbenějších formátů používaný na obrázky s paletou a jedním bytem na pixel vyvinut firmou CompuServe.

## 2.2 Širší formátovací možnosti CSS

Kaskádové styly nabízejí bezpochyby mnohem rozsáhlejší formátovací možnosti, než nám může nabídnout samotný jazyk HTML. Navíc je lze užít společně s HTML a tak je možno usnadnit přechod, než se CSS naplno osamostatní. Pro bližší pohled si zde dovolím uvést několik rozdílů mezi prostým HTML a CSS, které pomohou pochopit jejich význam.

Například `<b>` „umí“ v HTML pouze tučné písmo. CSS s pomocí vlastnosti `font-weight` umožní tučnost písma nastavit až v devíti možných stupních.

Další zajímavé srovnání atributů nabízí `<font>`. Pomocí něj lze v HTML určit velikost písma v pevně daných stupních, rodinu a barvu písma. V kaskádových stylech se používá vlastnost `font-size`, která nám umožňuje nastavit velikost v mnoha různých jednotkách, nebo relativně k rozměru jiného prvku. Navíc lze spolu s barvou písma určit barvu pozadí, čímž lze zajistit čitelnost textu za všech okolností.

## 2.3 Snadná tvorba a údržba konzistentního stylu

Grafická kvalita a ergonomie je jedna z klíčových vlastností, kterou nám konzistentní styl umožňuje. K tomu nám slouží převážně systém řízení obsahu, bez něhož se jen velice obtížně dosahuje konzistence pomocí prostého HTML. Díky systému řízení obsahu je tak v praxi možno přeformátovat prakticky celý „web“ za pár okamžiků. Pokud bychom chtěli tuto komplexní změnu udělat v prostém HTML, znamenalo by to, že bychom museli nalézt v celém kódu veškeré značky `<font>`, ručně změnit atributy mnoha tabulek a rozměry buněk.

## **2.4 Oddělení struktury a stylu**

Jazyk HTML by měl převážně sloužit k popisu struktury obsahu a formátování. Jelikož se jazyk HTML stal v poslední době dosti nepřehledným a obtížně interpretovatelným, vývojáře to přinutilo vypustit a odstranit z jazyka nesystémové věci počínaje normou HTML 4.0, avšak jsou dále po přechodnou dobu podporovány v rámci speciální normy zvané HTML Transitional.

## **2.5 Vyšší přístupnost dokumentů**

Díky značnému zjednodušení kódu formou externích stylových předpisů se stává jazyk HTML přístupnější jak osobám s určitými „hendikepy“, tak i pro zobrazování obsahu na méně běžných zařízeních (mobilní telefony, palmtopy, ...).

## **2.6 Dynamická práce se styly**

Kaskádové styly jsou vynikající prostředek při tvorbě různých dynamických efektů na webových stránkách. Nejvíce jsou rozšířeny spolu s javaskriptem v menu stránek, kde nabízí možnost dynamického ukrytí a zobrazování HTML prvků.

## **2.7 Formátování XML dokumentů**

Dále je třeba si uvědomit, že kaskádové styly slouží i k formátování obecných XML dokumentů a ne jen k formátování HTML. CSS jsou asi jedinou jednoduchou cestou, jak lze zobrazit XML dokumenty přehledným a srozumitelným způsobem.

### **3. Popis způsobů optimalizace webových prezentací a použitých technologií**

CSS nabízí rozsáhlejší formátovací možnosti než samotné HTML. Zatímco HTML umožňuje určit horizontální zarovnání, sílu a styl čáry, její barvu nelze ovlivnit. Při použití CSS lze pomocí vlastnosti border vytvořit až 8 stylů čáry o libovolné síle a barvě. Kromě horizontální čáry, lze tvořit i vertikální. V HTML lze určit jeden ze tří možných stylů odrážky, v CSS pomocí vlastnosti list-style-image lze pro odrážky použít libovolný obrázek. Co se týká písma, lze v HTML určit jeho velikost v pevně daných stupních, rodinu písma a barvu. V CSS lze velikost písma nastavit vlastností font-size v mnoha různých jednotkách, nebo relativně k jinému rozměru. Spolu s barvou písma lze určit i barvu pozadí a zajistit tak čitelnost textu za všech okolností. Ještě širší možnosti nabízí CSS při formátování bloků textu vlastnostmi padding a margin.

Konzistentní styl je jedním z klíčových faktorů grafické kvality a ergonomie webu. Na všech stránkách téhož webového místa by měly být všechny nadpisy stejné úrovně, odkazy, odstavce běžného textu stejného stylu. S kvalitně navrženým stylovým předpisem je změna typu písma, barvy nebo velikosti vybraných prvků najednou otázkou okamžiku. Jedním z nejvýznamnějších přínosů CSS je možnost důsledného oddělení struktury obsahu a formátování. Jazyk HTML slouží především k popisu struktury hypertextového obsahu. Jeho prostřednictvím lze označit strukturální význam jednotlivých částí obsahu, například hierarchických nadpisů, zdůrazněného textu, tabulek, hypertextových odkazů, zkratk, zdroje, atd.

Pokud budeme chtít formátovat celý dokument pouze s pomocí CSS včetně rozvržení stránky a polohování jednotlivých prvků, bude naším základním požadavkem mít dobře strukturovaný HTML kód. Ke zdůraznění absence

formátovacích prvků v HTML použijeme přísné normy jazyka, tj. HTML 4.0 Strict, XHTML 1.0 Strict nebo XHTML 1.1.

Dále je nutné, aby se webdesignér před návrhem stylových předpisů rozhodl, jaké prohlížeče budou podporovány. Poté stylové předpisy vytvoří a odladí je pouze pro podporované prohlížeče a ostatním uživatelům je různými technikami znepřístupní. Stránky vytvořené dle této strategie jsou nejčastěji podporovány prohlížeči Internet Explorer, Opera a v dalšími prohlížeči založenými na jádru Gecko jako je Mozilla Firefox či Netlape Navigátor, který ukončil svůj vývoj a uživatelé jsou postupně nuceni přecházet na jiné prohlížeče. Tímto způsobem vytvořené stránky je poté možno zobrazit minimálně devadesáti procentům uživatelů, zatímco ostatním se zobrazí použitelný a dostatečně přehledný obsah s implicitním HTML formátováním.

Pro zajímavost uvádím drobný nástin možností, co kaskádové styly dokážou:

- Nastavit libovolnou a přesnou velikost písma, p r o k l á d á n í ,  
KAPITÁLKY
- Udělat odsazení prvního řádku odstavce, zvětšit řádkování
- Zrušit nebo zvětšit prázdný prostor po odstavci
- Automaticky formátovat nadpisy (například je všechny udělat zeleně)
- Zvýrazňovat odkazy po přejetí myši
- Udělat automaticky grafické odrážky
- Určité části textu zneviditelnit, zprůhlednit nebo nezobrazit
- Předefinovat grafický význam běžných tagů (například všechno, co je kurzívou, udělat i tučně)

- Nastavit pozadí čehokoliv, stránky, tabulky ale třeba i odstavce; pozadí se nemusí opakovat a může mít přesnou pozici
- Umístit nějaký objekt (třeba část textu) kamkoliv do stránky, možnost se překrývat
- Přidat k čemukoli rolovací lišty, oříznout, orámovat, nastavit okraje
- V kombinaci se skripty je dnes CSS nejmocnější zbraň pro "rozhybání" stránek
- Hlavní význam CSS spočívá v tom, že fungují hodně automaticky, přičemž se vzhled celého webu deklaruje jedním souborem

### 3.1 Obecné rozdělení

Abychom mohli HTML nějakým způsobem zformátovat kaskádovými styly, musíme k němu nejdříve připojit stylový předpis. Pokud používáme (X)HTML dokumenty, můžeme použít několik prostředků s různými výhodami a nevýhodami s rozdílnou mírou podpory v jednotlivých prohlížečích. Obecné příklady HTML jazyka jsou vedeny v přílohách 5 až 13.

Existují čtyři základní metody, jak můžeme připojit stylový předpis, prvek `<link>`, prvek `<style>`, atribut `style` různých HTML prvků, klientské skriptování (JavaScript) a DOM (Document Object Model), které si nyní blíže popíšeme.

Prvek `<link>` používáme v případě, že chceme připojit stylový předpis, který je uložen v externím souboru. Tato varianta je většinou preferována pro normální způsob použití.

Hlavními výhodami jsou:

- Jediný soubor se stylovým předpisem pro celý web či sekci
- Předpis se „cachuje“ na straně klienta a urychlí načítání dokumentů
- Pozitivní význam z hlediska přenositelnosti a správu webového obsahu

Na straně druhé je však třeba si uvědomit, že umístění stylového předpisu do samostatného souboru může mít v určitých případech i své nevýhody. Pokud například chceme dokument zachovat jako nedělitelný celek nebo když generujeme dokument dynamicky na straně serveru.

Samotný prvek vkládáme do hlavičky dokumentu (<head>) a má následující atributy:

- Href – určuje cestu k externímu souboru se stylovým předpisem
- Type – určuje typ obsahu (content-type) odkazovaného souboru
- Rel – určuje typ vazby k externímu zdroji
- Media – nepovinný atribut určující typ media, pro který je předpis určen
- Title – přiřadí textový titulek k alternativnímu stylovému předpisu

Prvek <style> můžeme vložit přímo do (X)HTML dokumentu stejně jako v předchozím případě do hlavičky (<head>) a má následující atributy:

- Type – určuje typ obsahu (content-type) odkazovaného souboru
- Media – nepovinný atribut určující typ media, pro který je předpis určen
- Title – přiřadí textový titulek k alternativnímu stylovému předpisu

Jedná se o méně praktický způsob připojení stylového předpisu k dokumentu, který neodděluje styly od dokumentu tak zřetelně, jako kaskáda, což má za následek následující nevýhody:

- Předpis zvyšuje svůj celkový objem a „necachuje“ se, což má za následek, že se každá nova stránka musí znovu načíst
- Horší udržovatelnost dokumentu, neboť se jedna změna stylu musí provést do více dokumentů a tak hrozí, že může dojít nedopatřením k nežádoucí nekonzistenci stylu

Atribut style lze přiřadit i jednotlivým prvkům a tuto metodu poté nazýváme vřazené stylové předpisy (inline style sheets). Avšak samotné vřazování stylových předpisů do HTML dokumentu je nevhodné a od verze 1.1 zakázané. Jedinou podstatnou výhodou je snadnější testování vzhledu, ale po otestování se musí styl přenést do externího stylového předpisu.

Klientské skriptování a DOM (Document Object Model) je velmi dobrý způsob tvorby webových aplikací z hlediska přizpůsobení vzhledu dokumentu aktuálním podmínkám klienta.

Tyto výše vyjmenované metody připojení stylových předpisů k (X)HTML dokumentům se vyznačují též různou mírou kompatibility s obvykle používanými prohlížeči, mohlo by dojít k chybné interpretaci, dokonce ke zcela nečitelnému vykreslení dokumentu.



## 3.2 Kaskáda

Jedna z klíčových funkcí v CSS je kaskáda, udávající postup, kterým se prohlížeč (browser) musí řídit při zpracování definic platných stylů pro zobrazovaný dokument a použité medium. Tento postup seřadí veškeré existující tabulky stylů dle jejich váhy, bere v potaz případné použití příkazu `!important`, zařazení importovaných tabulek na dané místo apod. Cílem kaskády je tedy zjištění, zda má daný prvek pro svou konkrétní vlastnost určenou nějakou hodnotu a pokud jich má více, tak určit tu správnou.

### 3.2.1 Váhy tabulek stylů

- Od autora – vytvořené autorem stránek, který je připojí k dokumentu
- Od uživatele – uživatel může používat soubory, které obsahují jeho vlastní navržené tabulky stylů nebo je stránka vybavena rozhraním, kde je možno si zvolit jaký styl chceme používat
- Od klienta – klient musí povinně využívat vlastní výchozí tabulku stylů popisující obvyklé formátování dokumentu v daném jazyce (popř. jí alespoň emulovat)

### 3.2.2 Příkaz `!important`

Tento příkaz použijeme tehdy, pokud potřebujeme dané definici dát přednost před všemi ostatními. Tímto dává CSS určitou možnost uživateli, aby si některé vlastnosti nastavil dle vlastní potřeby. U příkazu `!important` mají nejvyšší váhu implicitní pravidla v klientovi, dále pak obvyčejná pravidla zadaná uživatelem, obvyčejná pravidla zadaná autorem stránky, důležitá pravidla zadaná autorem stránky a až v poslední řadě důležitá pravidla zadaná uživatelem. Toto se hodí hlavně pro uživatele s určitými zdravotními indispozicemi, např. člověk s horším

barevným viděním, který dává přednost zobrazení černého textu na bílém pozadí, může ve svém prohlížeči vytvořit vlastní tabulku stylů a následně pozadí změnit.

### 3.2.3 Pravidlo `@import`

Toto pravidlo nám slouží k připojení se k tabulce stylů tabulky jiné. Menší priorita definic z importovaných tabulek má menší prioritu, než definice uvedené přímo v tabulce, což je důvod, proč je nutné tato pravidla uvádět na začátku tabulky.

### 3.2.4 Pořadí kaskády

Tento postup slouží pro pořadí stanovení priority definic, které musíme dodržet.

Pro zajímavost zde uvádím tento postup:

- Najdeme všechny definice pro určitý prvek, dané pro použité medium, pokud poté vyhovují použitému selektoru, vztahují se na daný prvek kaskády
- Definice se poté setřídí podle váhy jejich původu od nejnižší po nejvyšší a to sice podle pravidla z tabulek klienta, pravidla z tabulek uživatele, pravidla z tabulek autora, příkazem `!important` z tabulek autora a nakonec pravidla `!important` z tabulek uživatele
- Následuje druhé setřídění podle specifičnosti. Pokud mají dvě definice stejnou váhu, porovnáme jejich selektory, přičemž vždy platí, že ty specifičtější mají přednost před obecnějšími. Běžné třídy jsou v tomto případě i pseudo-třídy

Jako poslední následuje setřídění dle pořadí definování, které se použije, pokud máme dvě definice se stejnou váhou původu i specifičností. Poté platí, že použijeme tu, která byla definována později.

### **3.3 Vizualní formátovací model CSS**

Klient může zpracovávat dokumenty různými způsoby dle použitého media. Mezi ty nejzákladnější patří media vizuální, zvuková a dotyková, kde dvě posledně jmenovaná se využívají jen zřídka a to pouze na specifických zařízeních.

Vizuální formátovací model nám slouží ke zpracování stromu dokumentu na vizuálních médiích. Jedná se prakticky o jednu z nejdůležitějších částí CSS technologie. Klient během vytváření stromu již přiřadí hodnoty všem vlastnostem podle pravidel kaskády a poté má již všechny prvky uspořádané ve stromu dokumentu včetně událostí popisujících jejich podobu. Následně se pak zobrazí dle postupů určených formátovacím modelem.

### **3.4 Zobrazování dokumentu**

Dokument se musí vždy vykreslit na nějaké zobrazovací ploše. V případě stránkovaných médií, jako jsou tiskárna nebo projektor, se touto plochou stává sada jednotlivých stránek. Každá tato stránka se poté jeví jako jeden celek, který je nám k dispozici pro zobrazení dokumentu. U plynulých médií, jako jsou okno prohlížeče či obrazovka, je k dispozici zobrazovací plocha neomezené délky a šířky. Klient poskytující uživateli jakýsi průzor na plochu, skrz něj uživatel vidí určitou část dokumentu. K tomu je potřeba připojit posuvný mechanismus, pomocí něhož se bude průzor nad dokumentem přesouvat.

#### **3.4.1 Zobrazování prvků**

Pro jakýkoliv prvek stromu se ve vizuálním formátovacím modelu generuje rám (box) dle rámového modelu CSS. Následně tyto rozměry ovlivňují jeho vzhled a umístění na zobrazené ploše.

### **3.4.2 Rámový model, rozměry rámu**

Rámové modely jsou generovány pro prvky dokumentu. Jejich základem je oblast obsahu. Dalšími volitelnými oblastmi mohou být oblast výplňová (padding), oblast rámečku (border) a oblast okraje (margin). Rozměr kolem všech oblastí rámu je možno určit na každé straně odlišný.

### **3.4.3 Omezující bloky**

Tento omezující blok je vytvářen hranou obsahu prvku pro všechny jeho potomky. Také na něm závisí jeho umístění a rozměry. Každý prvek tedy musí respektovat rozměry svého omezujícího bloku a také sám o sobě definuje omezující blok svým potomkům. Výchozí omezující blok je určující pro kořenový prvek dokumentu. Pokud sami neurčíme jinak, použijí se výchozí rozměry, které jsou například u stránkových médií určeny formátem výstupního media, což může být například papír.

V případě plynulých médií se rozměry výchozího omezujícího bloku nastaví podle prohlížeče (browseru). Výchozí šířka se většinou nastaví dle aktuální velikosti průzoru. Z tohoto také vyplývá, že pokud v CSS nezadáme rozměry prvků html nebo body, výchozí šířka se nastaví dle rozměru okna prohlížeče a výška bude přizpůsobena délce obsahu.

### **3.4.4 Generování rámu**

Rámy jsou vytvářeny uvnitř každého omezujícího bloku. Rozlišujeme dva druhy rámu ve vizuálním formátovacím modelu a sice blokové a řádkové, kde na hodnotě vlastnosti display závisí typ použitého rámu.

Blokové prvky se nám zobrazí jako obdélníkové bloky a platí, že pokud je prvek blokový, tak vlastnost display má hodnotu block, list-item, table či run-it.

Blokové rámy jsou vytvářeny pomocí blokových prvků, kde je v prvku generován hlavní rám tvořící omezující blok pro jeho obsah. Hlavní rám obsahuje pouze rámy, které mají ten samý typ, takže buď jen blokové nebo jen řádkové.

Když nastane situace, že je obsah bloku promíchaný, tj. máme v něm jak bloky, tak nějaký řádkový obsah, bude sjednocen, aby byl ve výsledku tvořen pouze bloky, jelikož se řádkový obsah přemění na anonymní blok.

Řádkové tzv. inline prvky nevytváří samotné bloky a jejich typ je určován hodnotami vlastností jako je inline, inline – table a nebo i run – in. Tyto prvky generují řádkové rámy.

### **3.5 Pseudoprvky a pseudotřídy**

Pokud nelze ve stromu dokumentu přiřadit prvkům v závislosti na jejich pozici styl, což se může stát například v HTML 4.0, kde neodpovídá žádný prvek prvnímu řádku odstavce. Nelze na něj tedy navázat žádný typ ze známých selektorů ve stromu dokumentu. Pro tyto případy je v CSS obsažen mocný nástroj pseudotříd a pseudoprvků, na jehož základě je možné formátovat informace, které leží mimo strom dokumentu.

Pokud potřebujeme rozšířit obsah dokumentu o prvky, které nejsou v jazyku dokumentu obsaženy, využijeme právě pseudoprvky, které jsou dokonce schopny zpřístupnit neexistující obsah v dokumentu, například pseudoprvky :before a :after, které slouží pro práci s generovaným obsahem.

Naopak pseudotřídy slouží k rozlišování prvků dle odlišných charakteristik, které nejdou odvodit ze stromu dokumentu. Třída může být i dynamická, takže může v jeden okamžik patřit či nepatřit k dané pseudotřídě, podle toho, co uživatel s dokumentem provádí.

Tyto třídy nejsou explicitně uvedeny ve zdrojovém kódu dokumentu nebo ve stromu dokumentu. V případě užití v selektorech můžeme použít pseudotřídy kdekoliv, zatímco pseudoprvky pouze za subjektem selektoru. Velikost písmen v názvech tříd je nepodstatná. Také lze někdy aplikovat jeden prvek na více tříd současně. V případě konfliktu bude rozhodovat pořadí kaskády.

### 3.6 Média

Velikým, a možná vůbec nejvýznamnějším přínosem CSS je možnost určení formy prezentace dokumentu a to hlavně pro výstupní media jako je obrazovka, papír, výstup v braillově písmu, hlasový syntetizér atd. To má tu výhodu, že některé vlastnosti jsou přístupné jen pro určitý druh media a jiné vlastnosti se zase dají použít pro více médií, pokud mají samozřejmě jiné hodnoty.

V praxi lze použít dva způsoby přiřazení stylového předpisu určitému médiu. První možnost je přímé určení media ve stylovém předpisu tzv. at-pravidly @media nebo @import. Druhá možnost je odvislá od jazyka dokumentu. Pro příklad můžeme uvést atribut media prvku <link> platný v HTML 4.

Každé médium je charakterizováno množinou CSS vlastností, které jsou pro dané médium použitelné. Jména typů médií odpovídají cílovým zařízením. Jsou to:

- All – vhodné pro všechna zařízení
- Aural – určené pro řečové syntetizéry
- Braille – určené pro zařízení, která umožňují hmatové vnímání braillova písma pro nevidomé
- Embossed – určené pro stránkovaný braillovský výstup, tj. tiskárny pro nevidomé

- **Handheld** – určené pro malá přenosná zařízení, typicky s malou, často monochromatickou obrazovkou (minulost) a nízkou rychlostí přenosu dat
- **Print** – určené pro stránkovaný výstup na nezářící materiál (nejčastěji papír) a pro tiskový náhled na obrazovce
- **Projection** – určené pro promítané prezentace, ať přímo projektorem nebo prostřednictvím tisku na transparentní materiály (fólie, filmy)
- **Screen** – určeno především pro barevné počítačové obrazovky
- **Tty** – určeno pro media používající pevnou znakovou matici, např. textové terminály, přenosná zařízení s omezenými zobrazovacími možnostmi (mobilní telefony). Protože výstup na tato media není grafický, autoři by se měli vyvarovat udávání rozměrů v pixelech (jednotka px)
- **Tv** – určeno pro televizní přijímače a obdobná zařízení charakterizovaná nízkým rozlišením, nízkou barevnou hloubkou, omezenými možnostmi rolování obrazovky, avšak s možností zvukového výstupu

### **CSS2 definuje tyto skupiny médií:**

- Průběžná nebo stránková
- Vizuální, zvuková nebo stránková
- Znaková (pro znaková zařízení) nebo grafická (pro grafická zařízení)
- Interaktivní (pro zařízení umožňující interakci s uživatelem) nebo statická (neumožňující interakci s uživatelem)
- Všechna (zahrnuje všechna média)<sup>[19]</sup>

## 3.7 Aktuální rozšíření a novinky CSS3

Tento styl je již třetí verzí od roku 2005, kdy byl vývoj této technologie zahájen konsorciem W3C. Očekává se, že konečný vývoj této technologie bude ukončen během roku 2015. K dnešnímu dni je již ovšem k dispozici většina nových vlastností, které dnes běžné webové prohlížeče podporují.

Nové vlastnosti v CSS3, které si dále podrobněji popíšeme, jsou následující:

- Zaoblené rohy – vlastnost, pomocí které můžeme zaoblit rohy u HTML prvků
- Stíny u blokových prvků – vlastnost nastavující stín u blokového prvku
- Stíny u textů – vlastnost nastavující stín u prostého textu
- Transformace – vlastnost umožňující transformaci prvku, dělí se na translate, rotate a scale
- Průhlednost, barevné modely a další – jsou ještě stále ve vývoji či se právě dotvářejí, takže je nebudu blíže specifikovat <sup>[20]</sup>

### 3.7.1 Zaoblené rohy

#### **border-radius s jednou hodnotou**

Touto hodnotou ovlivníme všechny čtyři rohy daného prvku.

```
div {  
-webkit-border-radius: 25px;  
-moz-border-radius: 25px;  
border-radius: 25px;  
}
```

**Podpora:** Firefox, Opera, Chrome, Safari, Explorer



### **border-radius se dvěma hodnotami**

První hodnota ovlivňuje *levý horní* a *pravý dolní* roh, druhá *levý dolní* a *pravý horní* roh.

```
div {  
-webkit-border-radius: 10px 25px;  
-moz-border-radius: 10px 25px;  
border-radius: 10px 25px;  
}
```

**Podpora:** Firefox, Opera, Chrome, Safari, Explorer

### **border-radius se čtyřmi hodnotami**

První hodnotou ovlivní *levý horní* roh, druhá *pravý horní* roh, třetí *pravý dolní* roh a ta čtvrtá ovlivní *levý dolní* roh.

```
div {  
-webkit-border-radius: 10px 20px 30px 40px;  
-moz-border-radius: 10px 20px 30px 40px;  
border-radius: 10px 20px 30px 40px;  
}
```

**Podpora:** Firefox, Opera, Chrome, Safari, Explorer

## **3.7.2 Stíny u blokových prvků**

### **box-shadow**

Vlastnost, která má čtyři hodnoty. První nastavujeme horizontální posun stínu od daného objektu. Druhá nastaví vertikální posun stínu od objektu. Třetí hodnota slouží k nastavení okraje stínu, kde poté stín přechází do ztracena a poslední hodnota je určena pro barvu stínu.

```
div {  
-webkit-box-shadow: 15px 10px 20px #000;  
-moz-box-shadow: 15px 10px 20px #000;  
box-shadow: 15px 10px 20px #000;  
}
```

**Podpora:** Firefox, Opera, Chrome, Safari, Explorer<sup>[20]</sup>

### 3.7.3 Stíny u textů

#### text-shadow

Tato vlastnost je úplně stejná jako u předchozího objektu, má celkem čtyři hodnoty. První z nich nastavujeme horizontální posun stínu od textu. Druhá naopak slouží k nastavení vertikálního posunu stínu od textu. Třetí se nastavuje hodnota okraje stínu, kde poté stín přechází až do ztracena a čtvrtá hodnota je určena pro barvu stínu.

```
div {  
-webkit-text-shadow: 15px 10px 20px #000;  
-moz-text-shadow: 15px 10px 20px #000;  
text-shadow: 15px 10px 20px #000;  
}
```

**Podpora:** Firefox, Opera, Chrome, Safari, Explorer

### 3.7.4 Transformace

#### transform: translate

Díky této vlastnosti se může prvek pohybovat po ose x a y. Podhodnotou této vlastnosti je v tomto případě translate, ta nabývá dvou hodnot a to osy x a y, kam se má daný prvek posunout. Když je x kladné, tak se daný prvek posune doprava, při záporné hodnotě doleva. U osy y je to stejné, když je hodnota kladná, tak se prvek posune dolů, při záporné hodnotě se posune směrem nahoru.

```
div {  
-ms-transform: translate(2em,1em);  
-webkit-transform: translate(2em,1em);  
-moz-transform: translate(2em,1em);  
transform: translate(2em,1em);  
}
```

**Podpora:** Firefox, Opera, Chrome, Safari, Explorer<sup>[20]</sup>

## **transform: rotate**

Vlastnost umožňující rotaci daného prvku. Podhodnotou této vlastnosti je v tomto případě rotace, která má pouze jedinou hodnotu a to pootočení. Prvek se poté pootočí kolem své osy a to ve směru hodinových ručiček o danou hodnotu.

```
div {
-ms-transform: rotate(50deg);
-webkit-transform: rotate(50deg);
-moz-transform: rotate(50deg);
transform: rotate(50deg);
}
```

**Podpora:** Firefox, Opera, Chrome, Safari, Explorer

## **transform: scale**

Díky této vlastnosti umožníme prvku změnu své velikosti. Podhodnotou této vlastnosti je v tomto případě scale, která má jen jednu hodnotu a to hodnotu zvětšení. Poté se prvek zvětší o danou hodnotu. Když je jeho hodnota 1, tak se jedná o standardní velikost.

```
div {
-ms-transform: scale(2,4);
-webkit-transform: scale(2,4);
-moz-transform: scale(2,4);
transform: scale(2,4);
}
```

**Podpora:** Firefox, Opera, Chrome, Safari, Explorer<sup>[20]</sup>

## 4. Návrh optimalizace konkrétní webové aplikace

Nejdříve je třeba uvést stručný úvod do problému, se kterým jsem se při vytváření a správě webových stránek setkal. Následně budou vysvětleny různé postupy jejich rozměrové, datové, rychlostní, popřípadě další optimalizace. Na obr.1 je uvedeno torzo webových stránek. Ty je třeba optimalizovat z hlediska zobrazení v různých prohlížečích, tak z pohledu zařízení, ve kterých budou zobrazována, například mobilní telefon, tablet, laptop či desktop a v neposlední řadě rychlosti jejich zobrazování ve zmíněných zařízeních a prohlížečích.

Obr. 1 – hlavní domovská stránka

**PRAGUE UNITED** akce co děláme rozvrh a rezervace o nás sparing partneři

**Velikonoční squashová bitva na Haštali – 30.3.2013**  
04.03.2013

Je čas po zimě opět změnit svoje síly a pořádně se zničit! Kdy? 30.3. 13.30 – 18.00 Kde? Squash centrum Haštál Při naplnění kapacity 32 hráčů (min. 30-ti) Vás čeká systém turnaje – pavouk s dvojitou eliminací, který je vyjímečný tím, že i když vyhr...

more info

**Půldenní squashový kemp**  
12.3.2013

Další z řady populárních squashových kempů, se opět vyvedl. Na hráče čekal bohatý a různorodý program, který vedli naši zkušení trenéři Karolína Uhrinová a Petr Vosátka. Cena kempu byla stanovena na 400,-. Rámcový harmonogram kempu: 16.00 – zahřátí, rozcvičení, protažení 16.30 – fyzický trénink 17.00 – squashový trénink 19.00 – workshop na dané téma 19.30 [...]

**Squashový turnaj ve čtyřhře – výsledky**  
12.3.2013

Praha , Squash centrum Haštál 8.2.2013 Osm nohou na kurtu, osm nohou na kurtu = turnaj smíšených čtyřher. V pátek 8.2.2013 večer nás sešlo 8 smíšených squashových párů na dolních kurtech na Haštali. Některé dvojice se znaly, a ti co ne, tak se hodně rychle skamarádily. Jelikož čtyřhra není úplně běžná soutěž a je zde [...]

**Pravidelné Akce**  
Prague United

- Trénink s Karolínou
- Trénink s Petrem
- Trénink s Adámem
- Fyzický trénink s Davidem
- Anкета o 50% slevu

© COPYRIGHT BY SK PRAGUE UNITED | PETR VOSÁTKA & ADAM KILIÁN & TEREZA GRIGAROVÁ napište nám

## 4.1 Zázemí

Hostingové služby nám poskytuje společnost Stable s.r.o., kde v současné době hostuje více jak 53 000 domén. Tato společnost se již přes deset let úspěšně pohybuje na českém webhostingovém trhu. Je akreditovaným registrátorem .cz domén a poskytovatelem webhostingu. Společnost se dlouhodobě specializuje na webhosting vysokozátěžových projektů. Z toho vznikla i spolupráce se společnostmi Seznam.cz, T-Mobile Czech Republic a.s., Centrum Holdings a mnoha dalšími, také tedy s námi.

### 4.1.1 Hardwarová výbava serveru

Webserver s názvem medvídek (srv7.endora.cz) je řízen procesorem 2x QuadCore Operon 2.2 Ghz, 24 GB velkou RAM paměť, diskové pole 4x300 GB SATA2 10k zapojené v RAID10, dále používá flashcache 60 GB SSD.

**Obr. 2 – webserver v rackovém provedení, Zdroj: endora.cz**



#### **4.1.2 Umístění serveru**

Server je umístěn v moderním datovém centru, které využívá připojení společnosti SuperNetwork. Centrum je velmi dobře zabezpečeno a je vybaveno veškerými technologiemi potřebnými pro profesionální telekomunikační a webhostingové služby. Samozřejmostí je zálohované napájení, stabilní samozhášecí systém, uzamykatelné stojany různých velikostí, výkonná a zálohovaná klimatizace. Ve stejném centru se nachází i servery společnosti Seznam.cz, Nova.cz, Libimseti.cz, Toplist.cz a mnoho dalších.

#### **4.1.3 Konektivita**

- Zálohované připojení k NIX.CZ 2 x 10 Gbps (český internet).
- Zahraniční konektivita pomocí 3 nezávislých poskytovatelů o celkové kapacitě 10 Gbps.
- Samostatná linka do SIX.SK 1x 1 Gbps (slovenský internet).
- Každý ze serverů je připojen do dvou nezávislých switchů.

#### **4.1.4 Napájení**

- Záložní zdroje napájení na jednotlivé servery, diesel agregát, UPS, kogenerace
- Každý stojan má připojení ze 3 nezávislých fází elektrické rozvodné sítě 230/400 V

#### **4.1.5 Zabezpečení přístupu**

- Nepřetržitá recepce a přístup (nepřetržitý režim 24 hodin denně / 7 dní v týdnu / 365 dnů v roce)
- Centrum je vybaveno uzamykatelnými stojany se zabezpečením proti vylomení a bezpečnostním kamerovým systémem

#### **4.1.6 Podmínky prostředí**

- Dvoustupňové klimatizační jednotky udržují stálou teplotu a vlhkost
- Možnost odvětrání každého jednotlivého stojanu
- Rackový stojan plné výšky, šíře 19", výška 42U
- Snadný příjezd nákladního vozu pro vykládání a nakládání

### **4.2 Hosting**

Hostingový server je pro naši potřebu opatřen aplikací WordPress ve verzi 3.4.2, dále je k dispozici PHP verze 5.4.11, MySQL 5.1.67, podpora .htaccess, 5 GB prostoru pro webovské stránky, 30 GB FUP na měsíční přenos dat a maximální počet 50 000 souborů ve WWW aplikaci.

## 4.3 Prvky webové aplikace

Veškeré grafické rozhraní a jeho editace je přístupná z WordPressu přes tzv. dashboard, kde je možnost editovat prakticky veškerá nastavení, na které si uživatel vzpomene. Přes toto rozhraní lze vkládat již předem definované stránky nebo obrázky. Hlavně však v našem případě editace kaskádového stylu. Jeho funkčnost se dá dále upravit pomocí různých pluginů, které slouží k doplnění vizuální bohatosti stránek nebo poskytují zpětnou vazbu webmasterovi o náročnosti určité skriptovací operace na paměť, předpokládanou dobu odezvy dané instance. Základní grafické rozhraní uvádím na obr.3.

Obr. 3 – domovská stránka administrátora

The screenshot displays the WordPress administrator dashboard. At the top, there is a navigation bar with the site name 'SK Prague United', a user profile 'vosatka', and a notification about a new WordPress 3.5.1 update. The main content area is divided into several sections:

- Úvodní stránka Aktualizace:** A notification icon with the number 2.
- Nástěnka (Dashboard):** The main area containing several widgets:
  - Aktuální přehled:** A summary table showing 5 posts, 27 pages, 5 categories, and 0 tags. It also indicates 0 comments, 3 approved comments, 0 pending comments, and 0 spam.
  - Rychlý příspěvek:** A form for quickly creating a new post, including fields for title, content, and tags, with buttons for 'Uložit koncept', 'Smazat', and 'Publikovat'.
  - Pluginy:** A list of installed plugins, including 'Ultimate TinyMCE' and 'BuddyPress XML-RPC Receiver'.
  - Nejnovější koncepty:** A section indicating that there are currently no drafts.
  - Nejnovější komentáře:** A section indicating that there are currently no comments.
  - Další užitečné novinky a informace o WordPressu:** A list of links to various WordPress resources and news.
- Left Sidebar:** A vertical menu with icons and labels for various dashboard functions: Příspěvky, Portfolio, Média, Odkazy, Stránky, Profily, Komentáře, Quform, Vzhled, Pluginy, Uživatelé, Nástroje, Nastavení, SEO, Facebook, Manage Menus, Floating Social Media Settings, Gallery, SH Slideshow, Performance, WP Translate, and Zmenšit menu.



### 4.3.1 WordPress

„WordPress je svobodný open source redakční publikační systém napsaný v PHP a MySQL. Je vyvíjen pod licencí GNU GPL a je silně podporován společností Automattic, jejíž vývojáři se WordPressu věnují. WordPress je oficiálním nástupcem systému b2/cafelog a má širokou uživatelskou a vývojářskou komunitu, kdy počet stažení samotné verze 3.0 dosahuje téměř 10 miliónů.

WordPress je používán dle oficiálních statistik jako CMS na více než patnácti procentech webových stránek na světě. Překonává tak jednoznačně OpenSource CMS jako Joomla či Drupal, které se drží pod deseti procenty.

WordPress od verze 3.0 umožňuje správu více webů pod jednou instalací, čímž zároveň nahrazuje WordPress MU. WordPress je schopen také vytvořit více webů na jednom serveru a databázi – pouze uložením souborů do různých podadresářů.

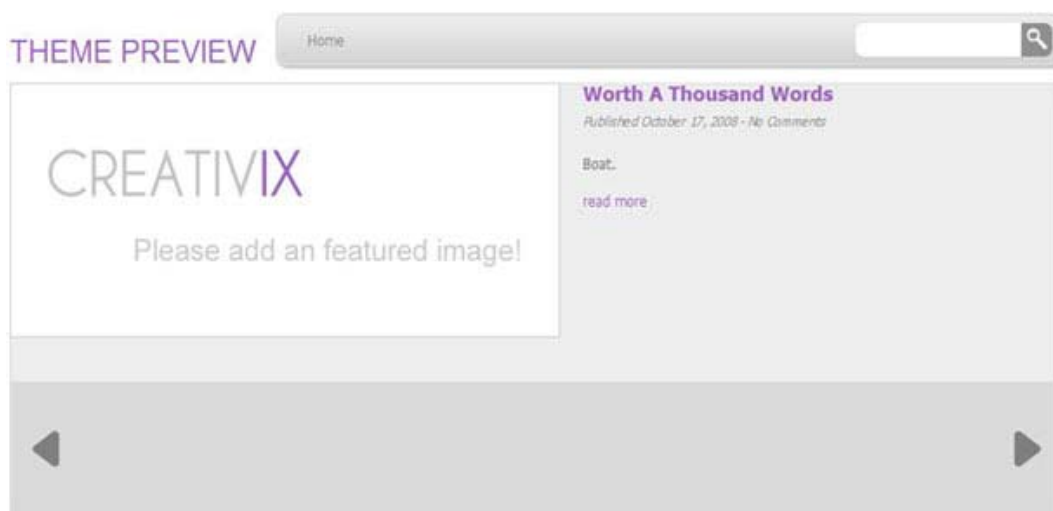
Základní vlastnosti:

- Dodržuje standardy XML, XHTML a CSS
- Integrovaný správce odkazů
- Struktura trvalých odkazů přátelská k internetovým vyhledávačům
- Podpora pluginů
- Podpora témat vzhledu
- Podkategorie a více kategorií pro příspěvek (tzv. štítky)
- Podpora pro trackback a pingback
- Typografický filtr pro formátování a styl textu<sup>[22]</sup>

### 4.3.2 Obecná šablona stránky

Struktura WordPressu šablony je tvořena různými php soubory, kdy každý z těchto souborů ovládá jinou oblast šablony. Některé soubory se mohou ve svých funkcích částečně prolínat. Přesto většinou platí, že při potřebě změny umístění určitého elementu šablony je nutné zasáhnout pouze do jednoho souboru. Znalost těchto zákonitostí je nutná např. pro překlad šablony do českého jazyka (v případě, že se v adresáři nenachází soubor pot). Pokud tedy např. potřebujeme umístit reklamu adsense na vstupní hlavní stránku, pak je potřeba zasáhnout do souboru index.php. Pokud ji však chcete mít umístěnou pouze na stránce jednotlivého příspěvku, pak je potřeba sáhnout po souboru single.php. Níže uvádím přehled hlavních mnou používaných souborů ve Word Pressu a jejich zdrojových kódů. Na obr.4 je ukázka základního zobrazení čisté šablony, která byla mnou posléze dále editována až do současné podoby, viz obr.1.

Obr. 4 – základní šablona domovské stránky



### 4.3.2.1 Zdrojový kód šablony – style.css

Pro fungování šablony je tento soubor naprosto nezbytný. Obsahuje definice veškerých nastavení rozměrů, fontů a stylů šablony. Pomocí tohoto kódu mohou opravovat veškeré vlastnosti zobrazení včetně nastavení okrajů, barev, rozměrů jednotlivého textu. V tomto dokumentu se mohou užít testovací vlastnosti, čímž bude docíleno toho, že i starší prohlížeče, které ještě podporovaly tyto testovací vlastnosti budou schopny alternativně zobrazit určité nastavení.

```
.alignnone {
margin: 5px 20px 20px 0;
}
.aligncenter, div.aligncenter {
display:block;
margin: 5px auto 5px auto;
}
.alignright {
float:right;
margin: 5px 0 20px 20px;
}
.alignleft {
float:left;
display:inline;
margin: 5px 20px 20px 0;
}
.aligncenter {
display: block;
margin: 5px auto 5px auto;
}
a img.alignright {
float:right;
margin: 5px 0 20px 20px;
}
a img.alignnone {
margin: 5px 20px 20px 0;
}
a img.alignleft {
float:left;
margin: 5px 20px 20px 0;
}
a img.aligncenter {
display: block;
margin-left: auto;
margin-right: auto
}
.wp-caption {
background: #fff;
border: 1px solid #EEE;
```

```

max-width: 96%; /* obrázek nepřekročí povolenou oblast */
padding: 5px !important;
padding-bottom: 25px !important;
text-align: center;
}
div.aligncenter {
padding-bottom: 0px !important;
}
div.alignnone {
margin: 0px;
padding: 0px;
display: block;
margin: 5px 20px 20px 0;
padding-bottom: 0px !important;
}
div.alignleft {
margin: 5px 20px 20px 0;
padding-bottom: 0px !important;
}
div.alignright {
margin: 5px 0 20px 20px;
clear: both;
padding-bottom: 0px !important;
}
.wp-caption img {
padding: 0px !important;
border: none !important;
height: auto;
margin:0;
max-width: 98.5%;
padding:0;
width: auto;
}
.wp-caption p.wp-caption-text {
font-size:11px;
line-height:17px;
margin:0;
margin-top: 5px;
width: 100%;
text-align: center;
padding:0 4px 5px;
}
.sticky p {
background: #f2f7fc;
padding: 10px;
}
.gallery .gallery-caption {
color: #888;
font-size: 12px;
margin: 0 0 12px;
}
.commentlist .bypostauthor {
}

```

### 4.3.2.2 Zdrojový kód šablony – header.php

Jedná se o soubor, ve kterém se nachází veškeré informace o nadpisu stránky. Dále pak se v něm nachází informace o navigaci a veškeré rozložení vrchní části šablony. Jelikož je tento kód značně obsáhlý, je v práci uveden jako příloha č.1.

### 4.3.2.3 Zdrojový kód šablony – index.php

Tento soubor obsahuje informace o těle hlavní stránky. Nachází se zde rozmístění jednotlivých panelů a lze tam přidat vše, co chceme mít na hlavní stránce, například reklamu. Většinou zobrazuje krátké úryvky posledně napsaných příspěvků, náhledů, rozložení stránky a sidebarů. Ve spodní části se nachází stránkování, které může být v základní formě „poslední příspěvek“, „další příspěvek“ nebo lze pluginem WordPress-pagenavi zobrazit počet stránek a navigaci. Jelikož je tento kód značně obsáhlý, je v práci uveden jako příloha č.2.

### 4.3.2.4 Zdrojový kód šablony – sidebar.php

V tomto souboru jsou obsaženy všechny informace o postranních panelech. Vícenásobné sidebary jsou pak definovány souborem functions.php. Obsah widgetů je spravován v administračním rozhraní WordPressu.

```
<div class="sidebar">
<ul id="sidebar">
<?php if ( !function_exists('dynamic_sidebar')
    || !dynamic_sidebar() ) : ?>
    <?php endif; ?>
</ul>
</div>
</div>
```

### 4.3.2.5 Zdrojový kód šablony – footer.php

Informace o rozmístění různých elementů ve spodní části šablony WordPressu. Nachází se zde informace o tvůrci šablony, případně další oblast pro widgety.

```
<div id="footer">

    <div class="wordpress-icon">
        <a href="http://www.wordpress.org/"></a>
    </div>

    <p>
        COPYRIGHT BY <a href="<?php echo home_url();?>" title="
<?php bloginfo('name'); ?>" target="_blank"><?php
bloginfo('name'); ?></a> | <span class="copyright"><i>Petr
Vosátka & Adam Kilián & Tereza Grigarová</i></span>
    </p>

    <div class="footer_right">
        <ul>
            <?php wp_nav_menu(array('theme_location' =>
'footer_menu', 'container' => false, 'depth' => 0, 'fallback_cb'
=> 'wpcx_default_menu')); ?>
        </ul>
    </div>
</div>

<p class="tagline"><?php bloginfo('description');?></p>

</div>
</div>

<?php wp_footer(); ?>

</body>
</html>
```

### 4.3.2.6 Zdrojový kód šablony – single.php

Zobrazení jednotlivých příspěvků se řídí jednoduchým souborem single.php. Soubor obsahuje pravidla vždy jen pro stránku s jedním příspěvkem. Může se upravit, pokud se má na stránce zobrazovat sidebar nebo vzhled jednotlivých stránek s příspěvkem.

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

<div id="sub-column">
  <div id="sub-top">
    <?php get_template_part('scripts/breadcrumb'); ?>
  </div>
  <div id="sub-content">
    <div class="content">
      <?php if (have_posts()) : ?>
      <?php while (have_posts()) : the_post(); ?>
        <div class="post">
          <h1 class="first"><?php the_title();?></h1>
          <div class="post_meta">
            <span class="category"><?php
the_category(' ') ?></span>
            <span class="date"><?php
the_time(get_option('date_format')); ?></span>
          </div>
          <?php the_content();?>
          <?php the_tags( '<p class="meta">Tags: ', ' ',
', '</p>'); ?>
        </div>
        <?php wp_link_pages(array('before' => '<div
class="wp-pagenavi">Pages: ', 'after' => '</div>')); ?>
        <?php edit_post_link('Edit Post'); ?>
        <div class="post_nav">
          <div class="nav-previous left"><?php
previous_post_link('%link', '<span class="meta-nav">&laquo;</span>
%title') ?></div>
          <div class="nav-next right"><?php
next_post_link('%link', '%title <span class="meta-
nav">&raquo;</span>') ?></div>
        </div>
        <?php endwhile; endif;?>
      <div id="comment-wrap">
        <?php comments_template(); ?>
      </div>
    </div>
  </div>

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

### 4.3.2.7 Zdrojový kód šablony – page.php

Soubor page.php obsahuje informace o zobrazení stránky jako takové, tzn. stránek, které se vytvoří v menu WordPressu v kolonce „stránky“ – „Přidat stránku“. Může se zde nastavit, zda bude na stránce sidebar, případně můžeme vytvořit libovolný počet samostatných php souborů a přiřadit je k jednotlivým stránkám při jejich vytváření.

```
<?php get_header(); ?>
<div id="sub-column">
    <div id="sub-top">
        <?php get_template_part('scripts/breadcrumb'); ?>
    </div>
    <div id="sub-content">

        <div class="content">
            <?php if (have_posts()) : ?>
            <?php while (have_posts()) : the_post();
?>

            <div class="post">
                <h1 class="first"><?php
the_title();?></h1>

                <?php the_content();?>

            </div>
            <div class="wp-pagenavi"><?php
wp_link_pages('before=Pages: &pagelink=<span>%</span>'); ?></div>
            <?php endwhile; endif;?>
            <?php comments_template(); ?>
            <?php edit_post_link('Edit Post'); ?>
        </div>
    <?php get_sidebar(); ?>
    <?php get_footer(); ?>
```

### 4.3.2.8 Zdrojový kód šablony – archive.php

Díky tomuto souboru se může upravovat vzhled jednotlivých archivů. Pokud složka šablony neobsahuje soubor archive.php, pak se zobrazí archivy dle souboru index.php, tedy jako hlavní stránka. Pokud soubor archive.php vytvoříme, určíme tím stránce nová pravidla. Pokud se následně vytvoří soubor category.php, pak se určí pravidla pro archivy dle kategorií. Jelikož je tento kód značně obsáhlý, je v práci uveden jako příloha č.3.



### 4.3.2.9 Zdrojový kód šablony – functions.php

Tento soubor dovolí upravovat a vkládat rozličné php kódy, kterými se bude chtít přepsat základní funkce šablony. Využívá se hlavně pro specifikaci počtu sidebarů, při změně počtu znaků ve výpisech, nebo třeba pro přidání vlastního odkazu pro možnost úpravy v administrátorském rozhraní WordPressu. Jelikož je tento kód značně obsáhlý, je v práci uveden jako příloha č.4.

### 4.3.2.10 Zdrojový kód šablony – comments.php

Tento soubor upravuje vzhled a funkce komentářů. Pokud se však v nastavení WordPressu povolí.

```
<?php
    if ( post_password_required() ) { ?>
        <p class="nocomments">This post is password protected.
Enter the password to view comments.</p>
        <?php
            return;
        }
    ?>
<!--Editovatelné pole. -->
<?php // Začátek komentářů & funkce Trackbacks ?>
<?php if ( have_comments() ) : ?>
<h3 id="comment-wrap"><?php comments_number('No Comments', 'One
Comment', '% Comments' );?> to "<?php the_title(); ?>"</h3>
<div class="pagination">
    <?php paginate_comments_links( array('prev_text' =>
'&laquo;', 'next_text' => '&raquo;') ) ?>
</div>
<ol class="commentlist">
    <?php wp_list_comments('avatar_size=68'); ?>
</ol>
<?php // Konec komentářů ?>
    <?php else : // zobrazuje se, pokud ještě nebyl udělen komentář?>
        <?php if ('open' == $post->comment_status) : // komentáře jsou
uzavřené ?>
            <?php elseif($post->post_type != "page") : ?>
                <p></p>
            <?php endif; ?>
        <?php endif; ?>
    <?php if ('open' == $post->comment_status) : ?>
    <?php comment_form(); ?>
    <?php endif; ?>
```

## **4.4 Důležité pluginy**

Tyto doplňky, které na mnou vytvořeném webu používám, slouží ke správnému zobrazování a optimalizace kódu pro zobrazení v různorodých zařízeních.

### **4.4.1 WP HTTP Compression**

Díky tomuto doplňku můžeme o 60-80 % zmenšit velikost načítané stránky a díky tomu i objem přenesených dat. Toto řešení najde široké uplatnění především v datových mobilních sítích GPRS, EDGE či UMTS, kde se povětšinou uplatňují zásady férového užívání služby, neboli FUP. V praxi to znamená, že má uživatel stanoven předem daný objem dat za určitou časovou jednotku. Pokud objem překročí, tak se mu výrazně omezí rychlost datového připojení. WordPress podporuje komprimovaný i nekomprimovaný přenos. Ke komprimaci se používá Gzip. Jedná se o aplikační software určený k bezztrátové kompresi dat pomocí algoritmů LZ77 (slovníkový algoritmus) a Huffmanova kódování (konvertuje znaky vstupního souboru do bitových řetězců různé délky).

### **4.4.2 W3 Total Cache**

Tento plugin nám snižuje dobu, po kterou se daná webová stránka načítá díky optimalizovanému progresivnímu vykreslování. Jeho hlavním posláním je však myšlenka, proč stahovat znovu něco, co už mám stažené? Princip řešení je prostý. Prohlížeč se při analýze stránky u objektu napřed „podívá“ do vlastní cache paměti, jestli tam daný objekt již není. Například se může jednat o pozadí nebo logo firmy. Při stahování první strany ze serveru se obrázek stáhne. Při dalším procházení se již obrázky načítají pomocí cache paměti. Důležité je, aby se stejné obrázky nedávaly vícekrát na různá místa. Prohlížeč nezjistí, že se jedná o tentýž obrázek. Můžeme dokonce dopředu načítat úplné stránky díky neviditelnému

elementu <iframe>. Je však potřeba podotknout, že toto neviditelné načítání dat může pozorného uživatele zmást při čekání na dokončení načítání.

### **4.4.3 NextGEN Gallery**

Tento plugin spravuje veškeré obrázky na webu a můžeme jím nastavit rozměr, efekty (vodoznak, Web 2.0) a obtékání. Galerii fotografií můžeme zobrazit jako seznam, prezentaci či prohlížeč. Dále můžeme tvořit samostatná alba, u kterých můžeme volit mezi rozšířenou nebo kompaktní verzí. K dispozici je také velké množství rozšíření (efekty, hlasování, slideshow) a možnost úpravy vzhledu pomocí CSS stylů. Tento plugin také umí automaticky upravovat velikost souboru, který na web nahráváme. Díky tomu šetří datový tok mezi uživatelem a serverem. Plugin navíc podporuje funkci pro zmenšení obrázku v případě, že na straně uživatele je detekováno mobilní zařízení. Díky tomu i uživatel s mobilním telefonem vidí obrázky celé a nemusí se snažit rolovat či zoomovat.

### **4.4.4 Float Left Right Advertising**

Jedná se o doplněk pro weby, které potřebují zobrazovat reklamu v postranních lištách, aniž by to narušovalo celistvost webu jako takového. Z hlediska nastavení je možnost udělat statické nebo dynamické banery, které jsou neustále na stejném místě z hlediska ukotvení stránky. Pokud potřebujeme webové stránky optimalizovat pro různá zobrazovací zařízení, nabízí se nám varianta detekovat používané rozlišení na straně uživatele a možnost reklamy vypnout. Můžeme nadefinovat tuto velikost pro vypnutí reklamy. Díky tomu se celková přehlednost webových stránek výrazně zlepší. Tuto funkci doporučuji používat hlavně v mobilních zařízeních, kde se čitelnost a efektivita zobrazovaných informací projeví nejvíce.

## 4.5 Testovací vlastnosti

„Ačkoliv většina novinek jazyka CSS3 se ještě nedostala do fáze kandidáta na doporučení W3C (kompletní verze specifikace), spoustu z nich už podporují prohlížeče Firefox, Internet Explorer, Chrome, Safari a Opera. Jako obranu před tím, že některé nedokončené nové vlastnosti jazyka CSS se mohou ještě změnit, implementovali výrobci prohlížečů testovací vlastnosti, tzv. vendor prefix. To umožňuje jednotlivým prohlížečům zavést vlastní podporu těchto vlastností, které nekolidují s finální specifikací, ani se staršími prohlížeči. Testovací vlastnosti také zajišťují, že jakmile se specifikace dostane do konečné fáze, tak existují webové stránky, které při použití experimentální funkce, budou nadále správně fungovat (odpovídající testovací vlastnosti k vlastnosti box-shadow je kupříkladu vlastnost `-webkit-box-shadow`). Každý rozšířený prohlížeč používá svou vlastní předponu testovacích vlastností:

-webkit- (prohlížeče s jádrem WebKit, což jsou například Safari a Opera),

-moz- (Firefox),

-ms- (Internet Explorer),

-o- (Opera),

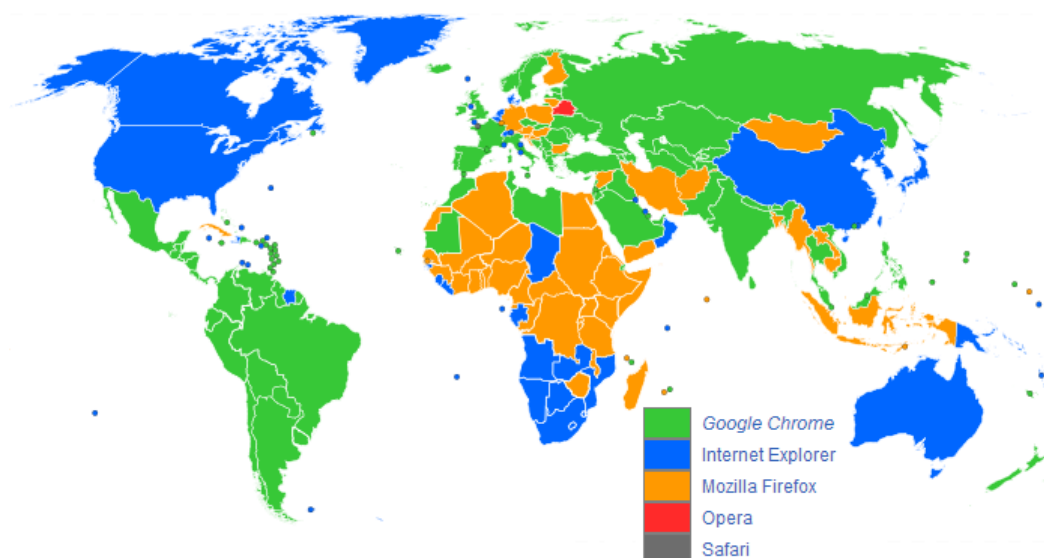
-khtml- (Konqueror).

Dále je možné testovat přítomnost mnoha funkcí knihovnamí JavaScriptu.<sup>[[17]]</sup>

## 4.6 Prohlížeče

Dalším problémem z hlediska optimalizace webových stránek je prohlížeč. Volba výběru správného browseru bývá někdy stěžejním problémem. Prohlížeče se dělí dle svého jádra, které rovněž ovlivňuje jejich následné chování z hlediska optimalizace stránek. WordPress umožňuje nainstalovat plugin detekující typ prohlížeče, který používá konkrétní uživatel a případně mu oznámí formou vyskakovacího okna, že se jedná o nepodporovanou verzi. Optimalizace webu pro veškeré prohlížeče se z mé vlastní zkušenosti rovná „alchymii“, kterou lze jen stěží dovést k naprosté dokonalosti. Při návrhu webových stránek je proto třeba se rozhodnout, kterou cestou půjdeme a jaké jádro či jádra budeme preferovat. Toto lze samozřejmě omezit díky užití testovacích vlastností. Používat vlastnosti u každé proměnné či deklarace je ale nemožné. Níže uvádím jejich základní rozdělení a stručnou historii jednotlivých jader a možnosti testování prohlížeců.

Obr. 5 – geografické rozdělení nejpoužívanějších prohlížečů, Zdroj: [gs.statcounter.com](http://gs.statcounter.com)



Testování internetových prohlížečů na správnost implementace zobrazované informace je jedna z nejdůležitějších věcí. K tomu nám poslouží tzv. Acid3 test.

„Acid3 je v pořadí již třetí test, který poukazuje na chyby v implementacích webových standardů internetových prohlížečů. Test v letech 2007–2008 za pomoci dobrovolníků vytvořil Ian Hickson a oficiálně byl publikován začátkem roku 2008 skupinou Web Standards Project.

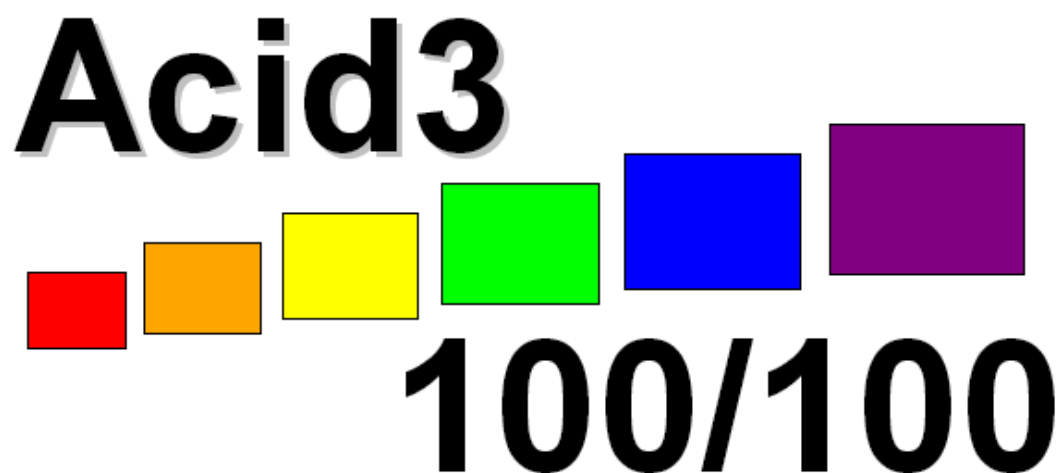
Oproti svým předchůdcům je Acid3 mnohem rozsáhlejší a soustředí se na testování několika specifikací, mj. DOM, JavaScript, SVG a CSS3.

Pro úspěšné splnění testu musí webový prohlížeč zobrazit Acid3 na pixel stejně se srovnávacím zobrazením a animace celého testu musí být plynulá.

Testem už úspěšně prošly poslední oficiální verze prohlížečů:

Google Chrome, Mozilla Firefox, Internet Explorer, Apple Safari, Opera<sup>«</sup><sup>[24]</sup>

Obr. 6 – ideální výsledek Acid3 testu



To pass the test, a browser must use its default settings, the animation has to be smooth, the score has to end on 100/100, and the final page has to look exactly, pixel for pixel, like [this reference rendering](#).

#### **4.6.1 Prohlížeče postavené na jádře Trident**

„Trident (též známé jako MSHTML) je renderovací jádro používané v prohlížeči Internet Explorer pro Microsoft Windows. Jeho první verze se objevila v říjnu 1997 v Internet Exploreru 4.0 a byla postupně vylepšována. Aktuální verze 5.0 je součástí Internet Exploreru 9.0. Starší verze byly známé svou slabší podporou webových standardů, některých nestandardních rozšíření a slabší bezpečností. Díky své integraci do operačního systému Windows je renderovací jádro využíváno v řadě komponent systému a využívá ho i řada instalovaných aplikací.

- Internet Explorer
- Netscape Browser 8.0 (umožňuje používat i jádro Gecko)
- 32 bit Web Browser
- AmiWeb
- Fast Browser
- NeoPlanet
- Smart Explorer
- AOL Explorer
- Avant Browser

## 4.6.2 Prohlížeče postavené na jádře Gecko

Gecko je open source renderovací jádro používané produkty Mozilla pro vykreslování webových stránek. Je napsáno v programovacím jazyce C++ a licencováno pod trojlíčením MPL (licence pro svobodný software) / GPL (všeobecná veřejná licence) / LGPL (licence pro svobodný software). Díky licenci a podpoře webových standardů je renderovací jádro používáno v řadě jiných prohlížečů. Jádro bylo původně vytvořeno Netscape Communications Corporation, ale nyní je vyvíjeno v Mozilla Corporation. Gecko díky svému bohatému API (rozhraní programování aplikací) nenabízí pouze možnost renderování webových stránek, ale slouží též k vykreslování grafického rozhraní, které využívá prohlížeč Firefox či Thunderbird. Jedná se o multiplatformní jádro, takže je k dispozici pro řadu platform jako Microsoft Windows, Linux, Mac OS X a další.

- Mozilla Firefox (umožňuje používat i jádro Trident)
- Mozilla Suite
- SeaMonkey
- Epiphany
- Galeon
- Netscape Navigator
- Camino
- K-Meleon
- Flock
- SkipStone



### 4.6.3 Prohlížeče postavené na jádře WebKit

WebKit je název renderovacího jádra prohlížeče a současně na tomto jádře postaveném frameworku, který využívají aplikace operačního systému Mac OS X společnosti Apple Inc. Část kódu jádra WebKit, která má na starost interpretaci jazyků HTML a JavaScript. Vychází zcela z knihoven KHTML (renderovací jádro) a KJS (javascriptový engine) vyvíjených KDE (desktopové prostředí pro linux) od roku 1998. WebKit byl původně určen jen pro operační systém Mac OS X, ale v dnešní době jej najdeme jako základ pro webové prohlížeče ostatních operačních systémů nebo jiných zařízení, např. čtečky e-knih.

- Konqueror
- Safari
- OmniWeb
- Google Chrome
- Comodo Dragon
- Maxthon (umožňuje používat i jádro Trident)
- Arora
- Epiphany
- Midori

#### **4.6.4 Prohlížeče postavené na jádře Presto**

Presto je proprietární renderovací jádro používané pro vykreslování stránek webovým prohlížečem Opera od firmy Opera Software. Poprvé se objevilo 28. ledna 2003 v Opeře 7.0 pro Windows a nahradilo tak starší jádro, které neslo označení Elektra.

- Opera 7 a výše (bude přecházet na jádro WebKit)

#### **4.6.5 Ostatní grafické prohlížeče**

- Amaya
- Arachne
- Dillo
- Links2
- Lobo
- NetSurf

#### **4.6.6 Textové prohlížeče**

- w3m
- Elinks
- Links
- Lynx<sup>[23]</sup>

## 5. Zhodnocení výsledků

Je více než zřejmé, že nelze s určitostí vypracovat nějaký univerzální návod, který nám poslouží jako „mustr“ při snaze o optimalizace internetových stránek. Aspektů je mnoho a navzájem se mohou i vylučovat. Obecně lze říci, že čím méně rozsáhlý kód, tím stránky v případě dobrého hostingu poběží rychleji. Avšak, aby se jednalo o jasné pravidlo, museli bychom si odmyslet celou řadu dalších zrádností jako jsou různé webové prohlížeče, užití většího množství obrázků, snaha o vyčištění kódu od zbytečností, kterých jsem díky tvorbě stránek ve WordPressu nebyl ušetřen, nebo různé pluginy, které rovněž mohou zatěžovat stranu klienta či stranu serveru. Výsledky testování a struktury obsahu webu SK Prague United jsou uvedeny v příloze č.14.

Norma CSS3 přinesla další a modernější technologie, které při správném využití umožní webovým stránkám vypadat uživatelsky přívětivěji a v mnoha případech též zrychlí celý web či ulehčí práci grafikům. Je však třeba používat tyto nové funkce s rozvahou, neboť řada z nich nebyla ještě plně implementována. A to, že se může například v prohlížeči Mozilla Firefox jevit určité řešení jako jednoduše geniální, neznamená, že stejnou radost budou sdílet uživatelé Internet Exploreru.

Tato práce mi ukázala možnosti, jakými lze webové stránky optimalizovat. Nelituji hodin strávených nad testováním odezvy webu v odlišných prohlížečích, operačních systémech, světových lokalitách. Dále všech jeho pluginů, php kódu, javastriptu a dalších věcí patřících k běžnému životu webmastera. Věřím, že nabyté poznatky z této práce v budoucnu plně využijí v pracovním procesu, ať již jako správce webu občanského sdružení SK Prague United, nebo v úplně odlišné organizaci a svými radami či postřehy pomohu vylepšit jejich webovou prezentaci. Doba statických webových stránek je již nenávratně pryč a lidé, kteří co nejdříve přijdou na chuť kaskádovým stylům, budou hrát prim v tvorbě webových stránek, jejich optimalizaci a možná jednou budou i navrhovateli nových standardů v kaskádových stylech. Nezbyvá než doufat, a já v to věřím.

## 6. Použitá literatura

- [1] Marek Prokop: CSS, kaskádové styly pro webdesignéry, CP Books, a.s., Brno 2005, s. 288
- [2] Jiří Žára a kol.: Moderní počítačová grafika, Computer Press, Brno 2004, s.609
- [3] Petr Staníček: CSS, kaskádové styly, Computer Press, Brno 2003, s. 178
- [4] URL: <http://www.w3.org/TR/css3-background/>
- [5] URL: <http://interval.cz/clanky/css3-kaskadove-styly-budoucnosti/>
- [6] URL: <http://www.jakpsatweb.cz/css/css-uvod.html>
- [7] URL: <http://www.ics.muni.cz/~makub/xml/html2xml.html>
- [8] URL: <http://www.w3.org/MarkUp/draft-ietf-iiir-html-01.txt>
- [9] URL: <http://www.w3.org/TR/REC-html32.html>
- [10] URL: <http://www.w3.org/TR/html401/>
- [11] URL: <http://www.w3.org/TR/xhtml1/>
- [12] URL: <http://www.w3.org/TR/REC-xml>
- [13] URL: <http://www.w3.org/TR/REC-CSS2/>
- [14] URL: <http://www.w3.org/TR/xslt>
- [15] URL: <http://xmlgraphics.apache.org/fop>
- [16] URL: <http://interval.cz/>

[17] Elizabeth Castro, Bruce Hyslop: Názorný průvodce tvorbou WWW stránek , Computer Press, Brno 2012, s. 439

[18] Brian P. Hogan: HTML 5 a CSS 3, Computer Press, Brno 2012, s. 272

[19] Vosátka, Petr. *Optimalizace internetové prezentace pro různá rozlišení a rozměry zobrazovacích zařízení*: bakalářská práce. Praha, Česká Zemědělská Univerzita, Fakulta Technická, 2009. 37 l., 5 l. příl. Vedoucí bakalářské práce Michal Hruška.

[20] URL: <http://cs.wikipedia.org/wiki/CSS3>

[21] URL: [http://w3techs.com/technologies/overview/content\\_management/all](http://w3techs.com/technologies/overview/content_management/all)

[22] URL: <http://cs.wikipedia.org/wiki/WordPress>

[23] URL: [http://cs.wikipedia.org/wiki/Webový\\_prohlížeč](http://cs.wikipedia.org/wiki/Webový_prohlížeč)

[24] URL: <http://www.root.cz/clanky/acid3-bitva-prohlizecu-vrcholi-prave-dnes/>

# Přílohy

## Příloha č.1 – header.php

```
<?php $wpcx_cxOptions = get_option('cxOptions'); ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" <?php
language_attributes(); ?>
<head profile="http://gmpg.org/xfn/11">
<meta http-equiv="Content-Type" content="<?php
bloginfo('html_type'); ?>; charset=<?php bloginfo('charset'); ?>"
/>

<title><?php wp_title('&laquo;', true, 'right'); ?> <?php
bloginfo('name'); ?></title>

<link rel="pingback" href="<?php bloginfo('pingback_url'); ?>" />

<?php if ( is_singular() ) wp_enqueue_script( 'comment-reply' );
?>

<?php wp_head(); ?>

</head>

<body <?php body_class(); ?>>
<?php DISPLAY_ACURAX_ICONS(); ?>
<div id="wrapper">

    <div id="header_top">

        <div id="logoname">

            <?php

                $wpcx_header = get_header_image();

                if(!empty($wpcx_header)) {

                    echo "<a href='" . home_url() . "' title='" .
get_bloggerinfo('name') . "'><img src='" . $wpcx_header . "' alt='" .
get_bloggerinfo('name') . "' /></a>";

                } else {

                    if(!empty($wpcx_cxOptions["logo_name"])) {

                        $wpcx_logo_name =
$wpcx_cxOptions["logo_name"];

                    } else{
```

```

        $wpcx_logo_name = "<div class='logo_text'" .
wpcx_cut_text(get_bloginfo('name'), 13, "") . "</div>";

    }

    echo "<a href='" . home_url() . "' title='" .
get_bloginfo('name') . "'>" . $wpcx_logo_name . "</a>";

    }

?>

</div>

<div id="topmenu">

    <?php

        if(!empty($wpcx_cxOptions["social_facebook"])) {

            ?>

            <a href="<?php echo
$wpccx_cxOptions["social_facebook"];?>" title="facebook"
target="_blank"></a>

            <?php

            }

            if(!empty($wpcx_cxOptions["social_linkedin"])) {

                ?>

                <a href="https://www.linkedin.com/e/fpf/<?php echo
$wpccx_cxOptions["social_linkedin"]; ?>" title="linkedin"
target="_blank"></a>

                <?php

                }

                if(!empty($wpcx_cxOptions["social_twitter"])) {

                    ?>

                    <a href="http://www.twitter.com/#!/<?php echo
$wpccx_cxOptions["social_twitter"]; ?>" title="twitter"

```

```

target="_blank"></a>

    <?php
    }

    if(!empty($wpcx_cxOptions["social_rss"])) {

        if($wpcx_cxOptions["social_rss"] == "yes") {

            ?>

            <a href="<?php bloginfo('rss2_url'); ?>"
title="rss" target="_blank"></a>

            <?php
            }
        }
    }
?>
</div>
</div>
<div id="header">
    <div class="navigation">
        <ul id="navbar">
            <?php wp_nav_menu(array('theme_location' =>
'nav_menu', 'container' => false, 'fallback_cb' =>
'wpcx_default_menu')); ?>
        </ul>

        <div id="search">
            <form id="searchform" method="get" action="/">
                <input type="text" value="" name="s"
id="searchbox" />
                <input type="submit" id="searchbutton" value="" />
            </form>
        </div>
    </div>
</div>

```



## Příloha č.2 – index.php

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

<?php

global $post;

if(!empty($wpcx_cxOptions["slide_max"])) {

    $wpcx_slide_max = $wpcx_cxOptions["slide_max"];

} else {

    $wpcx_slide_max = 0;

}

if(!empty($wpcx_cxOptions["slide_sort"])) {

    $wpcx_slide_sort = $wpcx_cxOptions["slide_sort"];

} else {

    $wpcx_slide_sort = "date";

}

if(!empty($wpcx_cxOptions["slide_order"])) {

    $wpcx_slide_order = $wpcx_cxOptions["slide_order"];

} else {

    $wpcx_slide_order = "DESC";

}

if(!empty($wpcx_cxOptions["slide_show"])) {

    $wpcx_slide_show = $wpcx_cxOptions["slide_show"];

} else {

    $wpcx_slide_show = "yes";

}

if($wpcx_slide_show != "no") {
```

```

        $wpcx_args = array( 'meta_key' => 'feat_slideshow',
        'meta_value'=> '1', 'suppress_filters' => 0, 'post_type' =>
array('post', 'page'), 'post_status' => 'publish', 'numberposts'
=> $wpcx_slide_max, 'orderby' => $wpcx_slide_sort, 'order' =>
$wpcx_slide_order);

        $wpcx_myposts = get_posts($wpcx_args);

        if(!$wpcx_myposts) {

                $wpcx_args = array('suppress_filters' => 0,
        'post_type' => array('post', 'page'), 'post_status' => 'publish',
        'numberposts' => $wpcx_slide_max, 'orderby' => $wpcx_slide_sort,
        'order' => $wpcx_slide_order);

                $wpcx_myposts = get_posts($wpcx_args);

        }

        if($wpcx_myposts) {

                ?>

                <div id="slide_wrapper">

                        <ul id="slideshow">

                                <?php

                                        foreach( $wpcx_myposts as $post ) :
setup_postdata($post);

                                                $wpcx_slideshow_title =
get_the_title();

                                                $wpcx_slideshow_text =
wpcx_cut_text(get_the_excerpt(), 290);

                                                $wpcx_thumb_big =
wpcx_get_wp_generated_thumb("slideshow");

                                                if(empty($wpcx_thumb_big)) {

                                                        $wpcx_thumb_big =
get_template_directory_uri()."/images/default_slideshow.jpg";

                                                }

                                                ?>

                                <li>" /><div
class="slide_text"><h1><a href="<?php the_permalink();?>"><?php
the_title();?></a></h1><span class="date"><?php the_time('d.m.Y');

```

```

?></span><p><?php echo $wpcx_slideshow_text;?></p><a href="<?php
the_permalink();?>">more info</a></div></li>

        <?php

        endforeach;

    ?>

</ul>

<div class="slide_nav">

    <ul id="slide_navigation">

        <?php

                                foreach( $wpcx_myposts as $post ) :
setup_postdata($post);

                                $wpcx_thumb =
wpcx_get_wp_generated_thumb("slideshow_thumb");

                                if(empty($wpcx_thumb)) {

                                    $wpcx_thumb =
get_template_directory_uri()."/images/default_slideshow_small.jpg"
;

                                    }

                                ?>

                                <li><a href="#">"
/></a></li>

                                <?php

                                endforeach;

                                ?>

        </ul>

        <a id="arrowleft" href=""></a>
        <a id="arrowright" href=""></a>

    </div>

```

```

        </div>

        <?php
        }
    }

    ?>

<div id="big-column">

    <div id="column-top">
    </div>

    <div id="column-content">

    <?php

    global $post;

    if(!empty($cxOptions["featured_sort"])) {

        $feat_sort = $cxOptions["featured_sort"];

    } else {

        $feat_sort = "date";

    }

    if(!empty($cxOptions["featured_order"])) {

        $feat_order = $cxOptions["featured_order"];

    } else {

        $feat_order = "DESC";

    }

    $args = array( 'meta_key' => 'feat_front', 'meta_value'=>
'1', 'suppress_filters' => 0, 'post_type' => array('post',
'page'), 'post_status' => 'publish', 'numberposts' => 2, 'orderby'
=> $feat_sort, 'order' => $feat_order);

    $myposts = get_posts( $args );

    if(!$myposts) {

        $args = array('suppress_filters' => 0, 'post_type' =>
array('post'), 'post_status' => 'publish', 'numberposts' => 2,
'orderby' => $feat_sort, 'order' => $feat_order);

```

```

        $myposts = get_posts( $args );
    }

    foreach( $myposts as $post ) : setup_postdata($post);

        $thumb = wpcx_get_wp_generated_thumb("feat_thumb");

        ?>
        <div class="feat-post" id="feat-post-<?php
the_ID();?>">

            <h2><a href="<?php the_permalink();?>"
title="<?php the_title();?>"><?php the_title();?></a></h2>
            <h3><?php the_time(get_option('date_format')));
?></h3>

            <?php the_excerpt();?>
            <a href="<?php the_permalink();?>"><?php
if($thumb) {?><?php }
?></a>

        </div>

        <?php endforeach; ?>

    <div class="latest-posts">
        <h2>Pravidelná kafe</h2>
        <h3>Prague United</h3>
        <ul class="latest-posts">
            <?php

                wp_reset_query();

query_posts("posts_per_page=8&orderby=date&order=DESC");

                while (have_posts() ) : the_post();

                    ?>

                    <li id="latest-post-<?php the_ID();?>"><a
href="<?php the_permalink();?>" title="<?php the_title();?>"><?php
the_title();?></a>

                        <?php endwhile;?>
                    </li>
                </ul>
            </div>

</div>

<?php get_footer(); ?>

```

## Příloha č.3 – archive.php

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

<div id="sub-column">

    <div id="sub-top">
        <?php get_template_part('scripts/breadcrumb'); ?>
    </div>

    <div id="sub-content">

        <div class="content">

            <?php if (have_posts()) :
            echo "<h1 class='first'>"; ?>

            <?php if (is_day()) { ?>
            Archive <?php
the_time(get_option('date_format')); ?>

            <?php } elseif (is_month()) { ?>
            Archive <?php the_time('n/Y'); ?>

            <?php } elseif (is_category()) { ?>
            Archive <?php echo single_cat_title(); ?>

            <?php } elseif (is_author()) { ?>
            Author Archive

            <?php } elseif (is_year()) { ?>
            Archive <?php the_time('Y'); ?>

            <?php } elseif (is_search()) { ?>
            Search Results

            <?php } elseif (isset($_GET['paged']) &&
!empty($_GET['paged'])) { ?>
            Blog Archives
            <?php }
            echo "</h1>";
            while (have_posts()) : the_post();
            $wpcx_thumb =
wpcx_get_wp_generated_thumb("reference");
            ?>

            <div id="post-<?php the_ID(); ?>"

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

```

the_permalink();?>" title="<?php the_title();?>"><?php
the_title();?></a></h1>
<div class="post_meta">
    <span
class="category">
        <?php

$wpccx_category = get_the_category();
$wpccx_name = $wpccx_category[0]->cat_name;
$wpccx_id = $wpccx_category[0]->cat_ID;
$wpccx_link = get_category_link($wpccx_id);
echo "<a href='\$wpccx_link' title='\$wpccx_name'>\$wpccx_name</a>";
        ?>
        </span>
    <span
class="date"><?php the_time(get_option('date_format')); ?></span>
    <span
class="social">
        <?php

if(!empty($wpccx_cxOptions["social_fb_like"])) {

if($wpccx_cxOptions["social_fb_like"] == "yes") {
        ?>
            <iframe
src="http://www.facebook.com/plugins/like.php?href=<?php echo
urlencode(get_permalink($post->ID));
?>&layout=button_count&show_faces=false&width=75&a
ction=like&font&colorscheme=light&height=21"
scrolling="no" frameborder="0" style="border:none;
overflow:hidden; width:130px; height:21px;"
allowTransparency="true"></iframe>
            <?php
        }
    }
    ?>
    <?php

```

```

if(!empty($wpcx_cxOptions["social_google_like"])) {

if($wpcx_cxOptions["social_google_like"] == "yes") {

    ?>

        <span
class="google"><g:plusone size="medium"></g:plusone></span>

        <?php
        }
    }
    ?>
</span>

    </div>

    <?php
    if(!empty($wpcx_thumb)) {
        ?>
        <div

class="with_image">

        <?php

        } else {
            echo "<div

class='no_image'>";

        }
        ?>

        <?php the_excerpt();?>

        </div>

    </div>

    <?php endwhile; else: ?>

    <div class="post">
        <h1>Nothing found</h1>

```



```
your criteria.</p>
                                <p>Sorry, no posts matched
                                </div>
                                <?php endif; ?>
                                <?php
                                get_template_part('scripts/wp-pagenavi');
                                if(function_exists('wp_pagenavi')) {
                                    wp_pagenavi();
                                } else {
                                    wp_link_pages();
                                }
                                ?>
                                </div>
                                <?php get_sidebar(); ?>
                                <?php get_footer(); ?>
```

## Příloha č.4 – functions.php

```
<?php

$wpcx_cxOptions = get_option('cxOptions');

// Set Content Width

$content_width = 600;

//Load Styles

add_action('wp_print_styles', 'wpcx_add_my_stylesheet');

function wpcx_add_my_stylesheet() {

    if ( !is_admin() ) {
        global $wpcx_cxOptions;

        if(!empty($wpcx_cxOptions['colorpicker'])) {
            $wpcx_colour = $wpcx_cxOptions['colorpicker'];
        } else {
            $wpcx_colour = false;
        }

        $wpcx_main_style = get_template_directory_uri() .
'/style.css';
        wp_register_style('mainStyle', $wpcx_main_style);
        wp_enqueue_style('mainStyle');

        if($wpcx_colour) {
            $wpcx_sub_style = get_template_directory_uri() .
'/style.php?color=' . $wpcx_colour;
        } else {
            $wpcx_sub_style = get_template_directory_uri() .
'/style.php';
        }
        wp_register_style('wpcx_sub_style', $wpcx_sub_style);
        wp_enqueue_style('wpcx_sub_style');
    }

}

add_action('admin_init', 'wpcx_add_to_admin_style');

function wpcx_add_to_admin_style() {
```

```

    $wpcx_admin_style = get_template_directory_uri() .
'/admin.css';
    wp_register_style('wpcx_admin_style', $wpcx_admin_style);
    wp_enqueue_style('wpcx_admin_style');

}

function wpcx_add_margin_header() {

    global $wpcx_cxOptions;

    if(!empty($wpcx_cxOptions["logo_margin"])) {

        echo "<style type='text/css'>#header { margin-top: " .
$wpcx_cxOptions["logo_margin"] . "px;}</style>";

    }

}

add_action('wp_head', 'wpcx_add_margin_header');

//Load JS

add_action('init', 'wpcx_add_my_scripts');

function wpcx_add_my_scripts() {

    if ( !is_admin() ) {

        wp_register_script('wpcx_jquery.cycle',
get_template_directory_uri() . '/scripts/jquery.cycle.all.js',
array('jquery'), '1.0' );
        wp_enqueue_script('wpcx_jquery.cycle');
        wp_register_script('wpcx_jquery.superfish',
get_template_directory_uri() . '/scripts/jquery.superfish.js',
array('jquery'), '1.0' );
        wp_enqueue_script('wpcx_jquery.superfish');
        wp_register_script('wpcx_jquery.fredsel',
get_template_directory_uri() . '/scripts/jquery.carouFredSel-
2.3.1.js', array('jquery'), '1.0' );
        wp_enqueue_script('wpcx_jquery.fredsel');
        wp_register_script('wpcx_jquery.fancybox',
get_template_directory_uri() . '/scripts/jquery.fancybox-
1.3.4.pack.js', array('jquery'), '1.0' );
        wp_enqueue_script('wpcx_jquery.fancybox');
        wp_register_script('wpcx_custom',
get_template_directory_uri() . '/scripts/custom.js',
array('jquery'), '1.0' );
        wp_enqueue_script('wpcx_custom');

    } else {

```

```

        wp_register_script('wpcx_jquery.picker',
get_template_directory_uri() . '/scripts/jquery.picker.js',
array('jquery'), '1.0' );
        wp_enqueue_script('wpcx_jquery.picker');
        wp_register_script('wpcx_admin_custom',
get_template_directory_uri() . '/scripts/admin_custom.js',
array('jquery'), '1.0' );
        wp_enqueue_script('wpcx_admin_custom');

    }
}

// Custom Logo Support

define( 'HEADER_IMAGE', '%s/images/logo.png' );

define( 'HEADER_IMAGE_WIDTH', apply_filters( '', 520 ) );

define( 'HEADER_IMAGE_HEIGHT', apply_filters( '', 155 ) );

define( 'HEADER_TEXTCOLOR', '' );

define( 'NO_HEADER_TEXT', true );

add_custom_image_header( '', 'wpcx_admin_header_style' );

//Add custom post type "Portfolio"

function wpcx_post_type_myportfolio() {

register_post_type(
    'myportfolio',
    array('label' => 'Portfolio',
        'singular_label' => 'Portfolio',
        'public' => true,
        'show_ui' => true,
        'capability_type' => 'post',
        'hierarchical' => false,
        'rewrite' => array("slug" => "portfolio-item",
"with_front" => false),
        'supports' => array('title', 'editor', 'custom-
fields', 'revisions', 'thumbnail'),
        'menu_position' => 4
    )
);

    register_taxonomy( 'portfolio', 'myportfolio', array(
'hierarchical' => true, 'label' => 'Portfolio Categories' ) );

}

add_action('init','wpcx_post_type_myportfolio');

if (!function_exists( 'wpcx_admin_header_style' )) {

```

```

function wpcx_admin_header_style() { ?>

    <style type="text/css">

        #heading {
            height: <?php echo HEADER_IMAGE_HEIGHT; ?>px;
            width: <?php echo HEADER_IMAGE_WIDTH; ?>px;
        }

        #heading h1, #heading #desc {
            display: none;
        }

    </style>
    <?php }
}

//WP 3.0 Menu

add_action('init', 'wpcx_register_custom_menu');

function wpcx_register_custom_menu() {
    register_nav_menu('footer_menu', 'Footer Menu');
    register_nav_menu('nav_menu', 'Navigation Menu');
}

//Remove wp_nav wrapping

function wpcx_my_nav_unlister( $menu ){
    return preg_replace( array( '#^<ul[^\>]*>#', '#</ul>$#' ), '',
    $menu );
}

add_filter( 'wp_nav_menu', 'wpcx_my_nav_unlister' );

// Fallback Nav Menus

function wpcx_default_menu() {

    echo "<li><a href='" . home_url() . "'>Home</a></li>";

}

// Cut some text

function wpcx_cut_text($text, $chars, $points = "...") {
    $length = strlen($text);
    if($length <= $chars) {
        return $text;
    } else {
        return substr($text, 0, $chars)." ".$points;
    }
}
}

```

```

// Get WP Generated thumb

function wpcx_get_wp_generated_thumb($position) {
    global $post_id;
    $thumb = get_the_post_thumbnail($post_id, $position);
    $thumb = explode("\", $thumb);
    if(!empty($thumb[5])) {
        return $thumb[5];
    }
}

if (function_exists('add_image_size')) {
    add_image_size( 'slideshow', 480, 210, true );
    add_image_size( 'slideshow_thumb', 100, 75, true );
    add_image_size( 'reference', 250, 130, true );
    add_image_size( 'feat_thumb', 295, 150, true );
}

// Styling & Script Functions

function wpcx_script() {

    global $wpcx_cxOptions;

    if(!empty($wpcx_cxOptions["slide_effect"])) {

        $wpcx_slide_effect = $wpcx_cxOptions["slide_effect"];

    } else {

        $wpcx_slide_effect = "scrollRight";

    }

    if(!empty($wpcx_cxOptions["slide_speed"])) {

        $wpcx_slide_speed = $wpcx_cxOptions["slide_speed"];

    } else {

        $wpcx_slide_speed = 500;

    }

    if(!empty($wpcx_cxOptions["slide_duration"])) {

        $wpcx_slide_duration = $wpcx_cxOptions["slide_duration"];

    } else {

        $wpcx_slide_duration = 3000;

    }
}

```

```

    $wpcx_script = '<script type="text/javascript"
src="https://apis.google.com/js/plusone.js"></script>
<script type="text/javascript">
    jQuery(document).ready(function($) {
        jQuery("ul#slideshow").cycle({
            fx:      "'. $wpcx_slide_effect.'",
            speed:   "'. $wpcx_slide_speed.'',
            timeout: "'. $wpcx_slide_duration.'',
            pager:   "ul#slide_navigation",
            pagerAnchorBuilder: function(idx, slide) {
                // return selector string for existing anchor
                return "#slide_navigation li:eq(" + idx + ") a";
            }
        });
    });
</script>';

    echo $wpcx_script;
}

add_action('wp_head', 'wpcx_script');

function wpcx_style() {

    $wpcx_style = "";

    global $wpcx_cxOptions;

    if(!empty($wpcx_cxOptions["logo_size"])) {

        $wpcx_logo_size = $wpcx_cxOptions["logo_size"];

        $style = '<style type="text/css">
#logoname a {
font-size: "'. $wpcx_logo_size.'px !important;
}
</style>
';

    }

    if(empty($wpcx_cxOptions["logo_file"])) {

        $wpcx_style .= '<style type="text/css">
#logoname {
margin-top: 12px;
}
</style>
';

    }
}

```

```

        ';
    }

    if(!empty($wpcx_style)) {
        echo $wpcx_style;
    }
}

add_action('wp_head', 'wpcx_style');

function wpcx_list_pages() {
    echo "<li><a href='".get_bloginfo('url')."'">Home</a>";
    wp_list_pages('title_li=');
}

// Featured Options

add_action("admin_init", "wpcx_admin_init");
add_action('save_post', 'wpcx_save_feat');

function wpcx_admin_init(){
    add_meta_box("page_feat", "Creativix Options",
"wpcx_feat_options", "page", "normal", "high");
    add_meta_box("post_feat", "Creativix Options",
"wpcx_feat_options", "post", "normal", "high");
}

function wpcx_feat_options(){

    global $post;
    $wpcx_custom = get_post_custom($post->ID);
    if(!empty($wpcx_custom)) {
        if(!empty($wpcx_custom["feat_slideshow"][0])) {
            $wpcx_featured = $wpcx_custom["feat_slideshow"][0];
        }
        if(!empty($wpcx_custom["feat_front"][0])) {
            $wpcx_feat_front = $wpcx_custom["feat_front"][0];
        }
    }
}

?>
<div class="inside">
    <table class="form-table">
        <tr>
            <th><label for="featured">Feature in
Slideshow?</label></th>
            <td><input type="checkbox" name="featured"
value="1" <?php if(isset($wpcx_featured) && $wpcx_featured == 1) {
echo "checked='checked'";} ?></td>

```



```

        </tr>
        <tr>
            <th><label for="featured">Feature on
frontpage?</label></th>
            <td><input type="checkbox" name="feat_front"
value="1" <?php if(isset($wpcx_feat_front) && $wpcx_feat_front ==
1) { echo "checked='checked'";} ?></td>
        </tr>
    </table>
</div>
<?php
}

function wpcx_save_feat(){

    global $post;

    if($post->post_type == "post" OR $post->post_type == "page") {

        if ( defined('DOING_AUTOSAVE') && DOING_AUTOSAVE )
            return $post_id;

        if(isset($_POST["featured"])) {

            update_post_meta($post->ID, "feat_slideshow",
$_POST["featured"]);

        } else {

            delete_post_meta($post->ID, "feat_slideshow", '');

        }

        if(isset($_POST["feat_front"])) {

            update_post_meta($post->ID, "feat_front",
$_POST["feat_front"]);

        } else {

            delete_post_meta($post->ID, "feat_front", '');

        }

    }

}

//Check for Post Thumbnail Support

if ( function_exists( 'add_theme_support' ) )
add_theme_support( 'post-thumbnails' );

```

```

// Register Sidebar

if ( function_exists('register_sidebar') )
    register_sidebar();

// Add RSS

add_theme_support('automatic-feed-links');

// Custom BG

add_custom_background();

// Editor Style

add_editor_style();

//Loading Theme Options Framework

$wpcx_themename = "WP-Creativix";
$wpcx_shortcode = "Creativix";

$wpcx_options = array (

array( "type" => "open",
"title" => "Frontpage Settings"
),

array( "name" => "Activate Slideshow on Frontpage?",
"desc" => "Decide if you want to show the big Slideshow on
Frontpage. By default the Slideshow will be shown.",
"id" => "slide_show",
"type" => "selectnormal",
"options" => array("yes", "no"),
"std" => "yes"),

array( "name" => "Limit Posts/Pages in Slideshow",
"desc" => "Set maximum amount of posts/pages to be shown in
the slideshow. Leave empty if you don't want to limit them.",
"id" => "slide_max",
"type" => "text",
"std" => ""),

array( "name" => "Sort Posts/Pages in Slideshow",
"desc" => "Choose how you want to sort your posts/pages in
the Slideshow. Default sort order is by date",
"id" => "slide_sort",
"type" => "selectnormal",
"options" => array("post_date", "rand", "title"),
"std" => "post_date"),

array( "name" => "Order Posts/Pages in Slideshow",

```

```

        "desc" => "Choose how you want to order your posts/pages in
the Slideshow. Default order is DESC",
        "id" => "slide_order",
        "type" => "selectnormal",
        "options" => array("DESC", "ASC"),
        "std" => "DESC"),

array( "name" => "Choose Sliding Effect",
        "desc" => "Choose an effect that will be applied for the
Slideshow. Default ist scrollRight.",
        "id" => "slide_effect",
        "type" => "selectnormal",
        "options" => array("scrollRight", "fade", 'scrollUp',
'scrollDown', 'blindZ', 'fadeZoom'),
        "std" => "scrollRight"),

array( "name" => "Slideshow Transition Speed",
        "desc" => "Set the Speed for the slideshow's transition.
Default ist set to 500 (ms)",
        "id" => "slide_speed",
        "type" => "text",
        "std" => "500"),

array( "name" => "Slideshow Duration",
        "desc" => "Set duration for each single Slide Item. Default
is set to 3000 (ms)",
        "id" => "slide_duration",
        "type" => "text",
        "std" => "3000"),

array( "name" => "Sort Posts/Pages for Featured Posts on
Frontage",
        "desc" => "Choose how you want to sort your posts/pages in
the Featured Area on Frontage. Default sort order is by date",
        "id" => "featured_sort",
        "type" => "selectnormal",
        "options" => array("post_date", "rand", "title"),
        "std" => "post_date"),

array( "name" => "Order Posts/Pages for Featured Posts on
Frontpage",
        "desc" => "Choose how you want to order your posts/pages in
the Featured Area. Default order is DESC",
        "id" => "featured_order",
        "type" => "selectnormal",
        "options" => array("DESC", "ASC"),
        "std" => "DESC"),

array( "name" => "Sort Posts for Latest Articles",
        "desc" => "Choose how you want to sort your latest Articles
displayed next to featured Articles. Default sort order is by
date",
        "id" => "articles_sort",
        "type" => "selectnormal",

```

```

        "options" => array("date", "rand", "title"),
        "std" => "date"),

array( "name" => "Order Posts for Latest Articles",
      "desc" => "Choose how you want to order your latest
Articles displayed next to featured Articles. Default order is
DESC",
      "id" => "articles_order",
      "type" => "selectnormal",
      "options" => array("DESC", "ASC"),
      "std" => "DESC"),

array("type" => "close"),

array( "type" => "open",
      "title" => "Styling Settings"
      ),

array( "name" => "Logo Font Size",
      "desc" => "Choose a font size for the Logo. Default font
size is set to 26px.",
      "id" => "logo_size",
      "type" => "text",
      "std" => "26"),

array( "name" => "Logo Text",
      "desc" => "Choose text for the logo. Leave this empty to
use your Wordpress Blog Title instead.",
      "id" => "logo_name",
      "type" => "text",
      "std" => ""),

array( "name" => "Logo/Nav Margin",
      "desc" => "Choose how many space should be between Social
Menu and Navigation/Logo in px.",
      "id" => "logo_margin",
      "type" => "text",
      "std" => ""),

array( "name" => "Logo Upload",
      "desc" => "Upload your own Logo. The Logo will not be
cropped/resized. You must apply correct width/height before
uploading. (Folder wp-creativix/images/logos/ must be writeable)",
      "id" => "logo_file",
      "type" => "upload",
      "std" => ""),

array( "name" => "Choose an highlight Colour",
      "desc" => "Click on the input field and choose a highlight
colour with the help of the colour-picker.",
      "id" => "colorpicker",
      "type" => "text",
      "std" => "939393"),

```

```

array("type" => "close"),

array( "type" => "open",
      "title" => "Social Media Settings"
    ),

array( "name" => "Facebook Profile URL",
      "desc" => "If you want to show the icon, enter the URL
(http://..) that points to your Facebook profile. (Leave empty to
not show the icon)",
      "id" => "social_facebook",
      "type" => "text",
      "std" => ""),

array( "name" => "Linkedin Profile ID",
      "desc" => "Insert your Linkedin Profile ID to show the
Linkedin Icon in the Header. (Leave empty to not show the icon)",
      "id" => "social_linkedin",
      "type" => "text",
      "std" => ""),

array( "name" => "Twitter Username",
      "desc" => "Insert your Twitter Username to show the Twitter
Icon in the Header. (Leave empty to not show the icon)",
      "id" => "social_twitter",
      "type" => "text",
      "std" => ""),

array( "name" => "Show RSS Icon in Header?",
      "desc" => "Choose if you want to display the RSS Icon in
the Header. Default is set to yes.",
      "id" => "social_rss",
      "type" => "selectnormal",
      "options" => array("yes", "no"),
      "std" => "yes"),

array( "name" => "Let Users like posts on Facebook",
      "desc" => "Choose if you want to display the facebook-like-
button next to your Post. Default is set to yes.",
      "id" => "social_fb_like",
      "type" => "selectnormal",
      "options" => array("yes", "no"),
      "std" => "yes"),

array( "name" => "Let Users +1 your Posts",
      "desc" => "Choose if you want to display the Google +1
Button next to your Post. Default is set to yes.",
      "id" => "social_google_like",
      "type" => "selectnormal",
      "options" => array("yes", "no"),
      "std" => "yes"),

array("type" => "close"),

```

```

array( "type" => "open",
      "title" => "Portfolio Settings"
    ),

array( "name" => "Sort Portfolio Items by",
      "desc" => "Choose how you want to sort your Portfolio items
on your Portfolio Page. Default sort order is by date",
      "id" => "portfolio_sort",
      "type" => "selectnormal",
      "options" => array("date", "rand", "title"),
      "std" => "date"),

array( "name" => "Order Portfolio Items",
      "desc" => "Choose how you want to order your Portfolio
items on your Portfolio Page. Default order is DESC",
      "id" => "portfolio_order",
      "type" => "selectnormal",
      "options" => array("DESC", "ASC"),
      "std" => "DESC"),

array( "name" => "Items per Page?",
      "desc" => "How many Portfolio Items would you like to
display per Page? Default is set to 9",
      "id" => "portfolio_page",
      "type" => "text",
      "std" => "9"),

array("type" => "close"),

array( "type" => "open",
      "title" => "Blog Settings"
    ),

array( "name" => "Sort Blog Items by",
      "desc" => "Choose how you want to sort your Blog items on
your Blog Page. Default sort order is by date",
      "id" => "blog_sort",
      "type" => "selectnormal",
      "options" => array("date", "rand", "title"),
      "std" => "date"),

array( "name" => "Order Blog Items",
      "desc" => "Choose how you want to order your Blog items on
your Blog Page. Default order is DESC",
      "id" => "blog_order",
      "type" => "selectnormal",
      "options" => array("DESC", "ASC"),
      "std" => "DESC"),

array( "name" => "Items per Page?",
      "desc" => "How many Blog Items would you like to display
per Page? Default is set to 10",
      "id" => "blog_page",
      "type" => "text",

```

```

        "std" => "10"),
array("type" => "close")
);

// create the Options page on the admin side

function wpcx_add_admin() {

    global $wpcx_themename, $wpcx_shortcode, $wpcx_options;

    // Saving and Updating the options

    if (!empty($_GET['page']) && $_GET['page'] ==
basename(__FILE__) && !empty($_POST) &&
check_admin_referer('wpcx_save_theme_options','wpcx_options_nonce'
)) {

        if (!empty($_REQUEST['action']) && 'save' ==
$_REQUEST['action']) {

            $wpcx_cxOptions = array();

            // print_r($_REQUEST);

            foreach ($wpcx_options as $wpcx_value) {

                if (!empty($wpcx_value['id'])) {

                    if(isset($_REQUEST[$wpcx_value['id']])) {

                        $wpcx_cxOptions[$wpcx_value['id']] =
esc_attr($_REQUEST[ $wpcx_value['id'] ]);

                        if(strpos($wpcx_cxOptions[$wpcx_value['id']], "http") !== false) {

                            $wpcx_cxOptions[$wpcx_value['id']] =
esc_url($_REQUEST[ $wpcx_value['id'] ]);

                                }

                            }

                        }

                    }

                }

            // Logo Upload

            if(!empty($_FILES["logo_file"])) {

                // Upload Logo

```

```

$wpcx_dir = TEMPLATEPATH . "/images/logos/";

if (is_writable($wpcx_dir)) {

    if ((($_FILES["logo_file"]["type"] ==
"image/gif") || ($_FILES["logo_file"]["type"] == "image/jpeg") ||
($_FILES["logo_file"]["type"] == "image/png") ||
($_FILES["logo_file"]["type"] == "image/pjpeg")) &&
($_FILES["logo_file"]["size"] < 1048576)) {

        if ($_FILES["logo_file"]["error"] > 0){
            echo "Return Code: " .
$_FILES["logo_file"]["error"] . "<br />";
        } else {
            $_FILES["logo_file"]["name"] =
str_replace(' ', '_', $_FILES["logo_file"]["name"]);
            if (file_exists($wpcx_dir .
$_FILES["logo_file"]["name"])) {
                echo $_FILES["logo_file"]["name"]
. " already exists. ";
            } else {

switch($_FILES["logo_file"]["type"]) {
                case "image/jpeg" : $wpcx_end
= ".jpg";
                    break;
                case "image/png" : $wpcx_end =
".png";
                    break;
                case "image/gif" : $wpcx_end =
".gif";
                    break;
            }
            $wpcx_newname = time().$wpcx_end;

if(move_uploaded_file($_FILES["logo_file"]["tmp_name"], $wpcx_dir
. $wpcx_newname)) {

                $wpcx_cxOptions['logo_file'] =
$wpcx_newname;

            }

        }
    }
}

if(!empty($_REQUEST['del_pic'])) {
    $wpcx_cxOptions['logo_file'] = "";
}

```



```

    }

    update_option('cxOptions', $wpcx_cxOptions);

    if (!empty($wpcx_value['id']) && isset($_REQUEST[
    $wpcx_value['id']])) {

        update_option('cxOptions', $wpcx_cxOptions);

    } elseif(!empty($wpcx_value['id'])) {

        delete_option( $wpcx_value['id'] );

    }

    header("Location:
themes.php?page=functions.php&saved=true");

    } elseif(!empty($_REQUEST['action']) && 'reset' ==
$_REQUEST['action']) {

        delete_option('cxOptions');

        header("Location:
themes.php?page=functions.php&reset=true");

    }

}

// Add Options page to the admin menu

add_theme_page($wpcx_themename." Options", "$wpcx_themename
Options", 'edit_theme_options', basename(__FILE__), 'wpcx_admin');

}

function wpcx_admin() {

    global $wpcx_themename, $wpcx_shortcode, $wpcx_options;

    ?>

    <div class="wrap">
        <div style="float: left; margin-top: 50px;">
            <h1></h1>
        </div>

        <div style="float: left; margin-left: 150px;">
            <form action="https://www.paypal.com/cgi-bin/webscr"
method="post">

```

```

        <input type="hidden" name="cmd"
value="_donations">
        <input type="hidden" name="business"
value="dennis.nissle@iwebix.de">
        <input type="hidden" name="lc" value="US">
        <input type="hidden" name="item_name" value="WP-
Creativix Theme Donation">
        <input type="hidden" name="currency_code"
value="USD">
        <input type="hidden" name="bn" value="PP-
DonationsBF:btn_donateCC_LG.gif:NonHostedGuest">
        <input type="image" src="<?php echo
get_template_directory_uri(); ?>/images/donate.png" border="0"
name="submit" alt="PayPal - The safer, easier way to pay online!">
        
    </form>
</div>

<h2 style="clear: both; margin-left: 5px; margin-bottom:
20px;">WP-Creativix Options Page</h2>

<?php
    if ( !empty($_REQUEST['saved']) && $_REQUEST['saved']
) echo '<div id="message" style="float: left; width: 655px;
margin-left: 10px;" class="updated
fade"><p><strong>'. $wpcx_themename.' settings
saved.</strong></p></div>';
    if ( !empty($_REQUEST['reset']) && $_REQUEST['reset']
) echo '<div id="message" style="float: left; width: 655px;
margin-left: 10px;" class="updated
fade"><p><strong>'. $wpcx_themename.' settings
reset.</strong></p></div>';
    ?>

<form method="post" enctype="multipart/form-data">

<?php

//print_r($options);

foreach ($wpcx_options as $wpcx_value) {

    $wpcx_cxOptions = get_option('cxOptions');

    switch ($wpcx_value['type']) {

        case "open": ?>
            <div class="container" style="clear:
both;background-color: #e8e8e8; border: 1px solid #CCC; padding:
10px; font-size: 11px; width: 650px; margin: 10px; float: left;
color: #3b3b3b;">
                <h3><?php echo $wpcx_value['title']; ?></h3>

```

```

        <?php break;

        case "close": ?>
            </div>
        <?php break;

        case 'text': ?>
            <div style="margin-top:15px; float: left;
clear: both;">
                <?php echo $wpcx_value['name']; ?><br/>
                <input name="<?php echo $wpcx_value['id'];
?>" id="<?php echo $wpcx_value['id']; ?>" type="<?php echo
$wpcx_value['type']; ?>" value="<?php if (
!empty($wpcx_cxOptions[$wpcx_value['id']])) { echo
esc_attr($wpcx_cxOptions[$wpcx_value['id']]); } else { echo
$wpcx_value['std']; } ?>" />
            </div>
            <div style="width: 350px; float: right;
margin-right: 20px; margin-top: 15px; background-color: #fff0b5;
border: 4px solid #FFF; padding: 5px; font-size: 10px;">
                <?php echo $wpcx_value['desc']; ?>
            </div>
        <?php break;

        case 'textarea': ?>

            <div style="margin-top:15px;padding:0; float:
left; clear: both;">
                <?php echo $wpcx_value['name']; ?><br/>
                <textarea style="width: 200px;
height:70px; font-size: 10px; border: 1px solid #b6b6b6;"
name="<?php echo $wpcx_value['id']; ?>" id="<?php echo
$wpcx_value['id']; ?>" type="<?php echo $wpcx_value['type']; ?>"
cols="" rows=""><?php if (
!empty($wpcx_cxOptions[$wpcx_value['id']])) { echo
esc_attr($wpcx_cxOptions[$wpcx_value['id']]); } else { echo
$wpcx_value['std']; } ?></textarea>
            </div>
            <div style="width: 350px; float: right;
margin-right: 20px; margin-top: 25px; background-color: #fff0b5;
border: 4px solid #FFF; padding: 5px; font-size: 10px;">
                <?php echo $wpcx_value['desc']; ?>
            </div>
        <?php break;

        case 'upload': ?>

            <div style="margin-top:15px;padding:0; float:
left; clear: both;">
                <?php echo $wpcx_value['name']; ?><br/>
                <input type="file" name="<?php echo
$wpcx_value['id'];?>" id="<?php echo $wpcx_value['id'];?>" />
            </div>

```

```

        <div style="width: 350px; float: right;
margin-right: 20px; margin-top: 25px; background-color: #fff0b5;
border: 4px solid #FFF; padding: 5px; font-size: 10px;">
            <?php echo $wpcx_value['desc']; ?>
        </div>
        <?php if (
!empty($wpcx_cxOptions[$wpcx_value['id']])) {
            echo "<div class='logo' style='float:
left; clear: both;'><img style='float: left; clear: both;'
src='".get_template_directory_uri()."/images/logos/".$wpcx_cxOptio
ns[$wpcx_value['id']]."'/><p style='float: left; clear:
both;'><input type='checkbox' name='del_pic' value='del' />Delete
Logo</p><input type='hidden' name='logo_file'
value='". $wpcx_cxOptions[$wpcx_value['id']]."' /></div>";
        }
        ?>
        <?php break;

        case 'selectnormal': ?>
            <div style="margin-top:15px; padding:0; float:
left; clear: both;">
                <?php echo $wpcx_value['name']; ?><br/>
                <select name="<?php echo
$wpcx_value['id']; ?>" id="<?php echo $wpcx_value['id']; ?>"><?php
foreach ($wpcx_value['options'] as $wpcx_option) { ?><option<?php
if(!empty($wpcx_cxOptions[$wpcx_value['id']])
&&$wpcx_cxOptions[$wpcx_value['id']] == $wpcx_option) { echo '
selected="selected"'; } elseif ($wpcx_option ==
$wpcx_value['std']) { echo ' selected="selected"'; } ?>><?php echo
$wpcx_option; ?></option><?php } ?></select>
            </div>
            <div style="width: 350px; float: right;
margin-right: 20px; margin-top: 15px; background-color: #fff0b5;
border: 4px solid #FFF; padding: 5px; font-size: 10px;">
                <?php echo $wpcx_value['desc']; ?>
            </div>
            <?php break;

        case "checkbox": ?>
            <div style="margin-top:15px; padding:0; float:
left; clear: both;">
                <?php echo $wpcx_value['name']; ?><br/>
                <?php if(get_option($wpcx_value['id'])) {
$wpcx_checked = "checked=\"checked\""; }else{ $wpcx_checked = ""; }
                ?>
                <input type="checkbox" name="<?php echo
$wpcx_value['id']; ?>" id="<?php echo $wpcx_value['id']; ?>"
value="true" <?php echo $wpcx_checked; ?> />
            </div>
            <div style="width: 350px; float: right;
margin-right: 20px; margin-top: 15px; background-color: #fff0b5;
border: 4px solid #FFF; padding: 5px; font-size: 10px;">
                <?php echo $wpcx_value['desc']; ?>
            </div>

```

```

        <?php break;
    }
}
?>

<div class="container" style="clear: both; background-
color: #e8e8e8; border: 1px solid #CCC; padding: 10px; font-size:
11px; width: 650px; margin: 10px; float: left; color: #3b3b3b;">
    <p class="submit" style="float: left; margin-right:
20px;">
        <input name="save" type="submit" value="Save
changes" />
        <input type="hidden" name="action" value="save" />
    </p>
    <?php
wp_nonce_field('wpcx_save_theme_options','wpcx_options_nonce'); ?>
    </form>
    <form method="post">
        <p class="submit">
            <input name="reset" type="submit" value="Reset" />
            <input type="hidden" name="action" value="reset"
/>
        </p>
    </form>
</div>

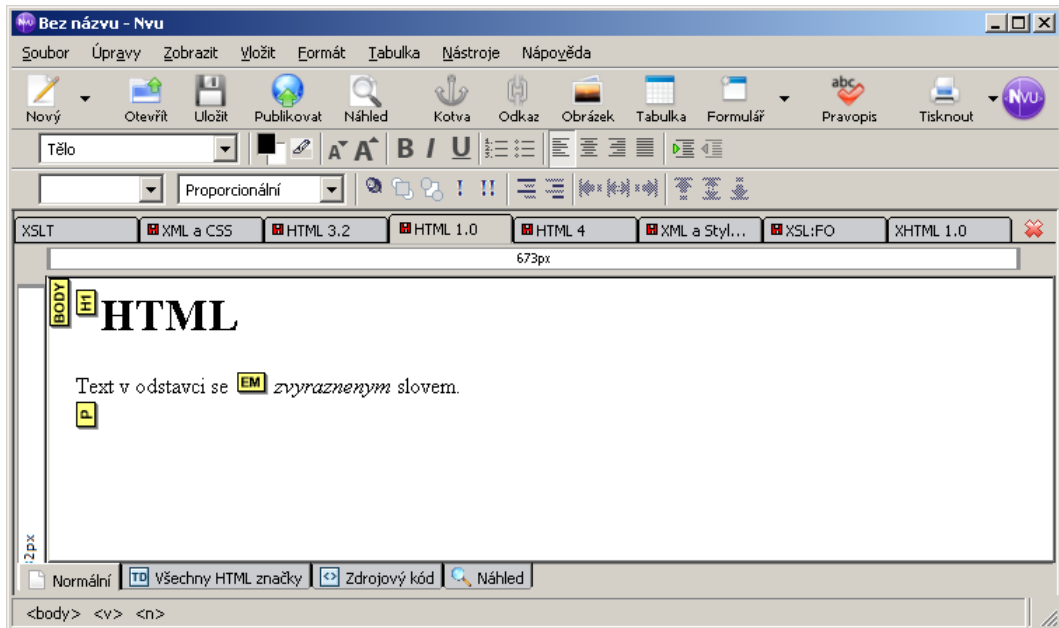
<?php
}

add_action('admin_menu', 'wpcx_add_admin');

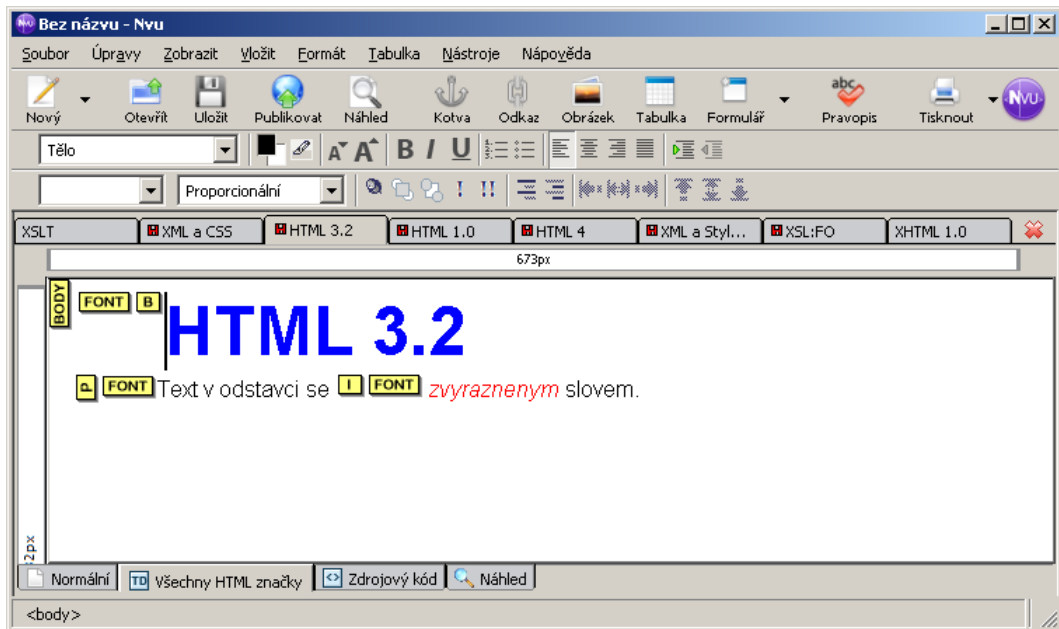
?>

```

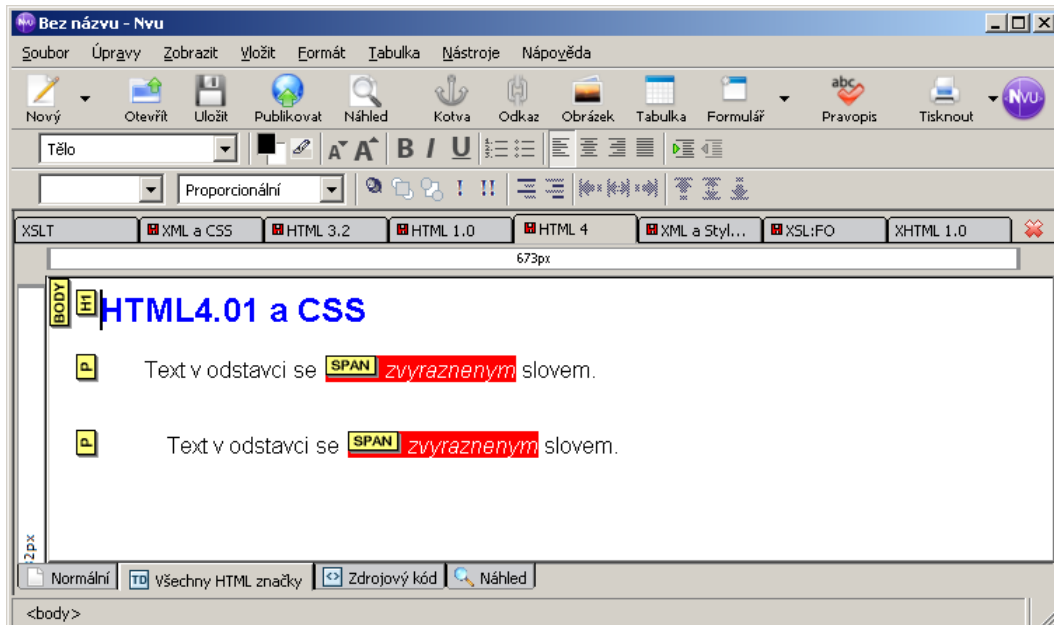
## Příloha č.5 – HTML



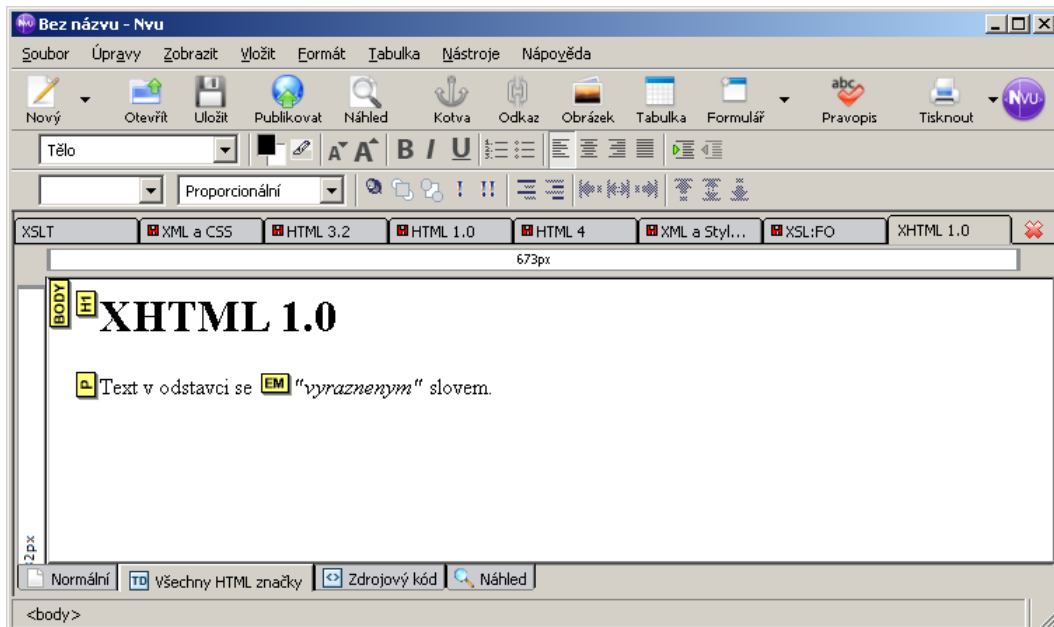
## Příloha č.6 – HTML 3.2



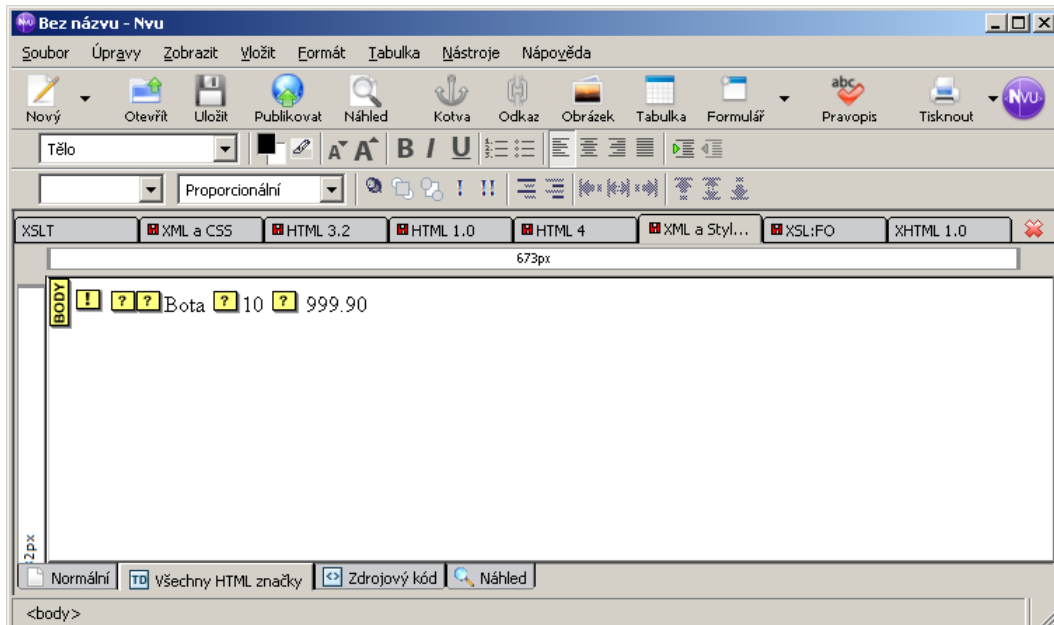
## Příloha č.7 – HTML 4.01 a CSS



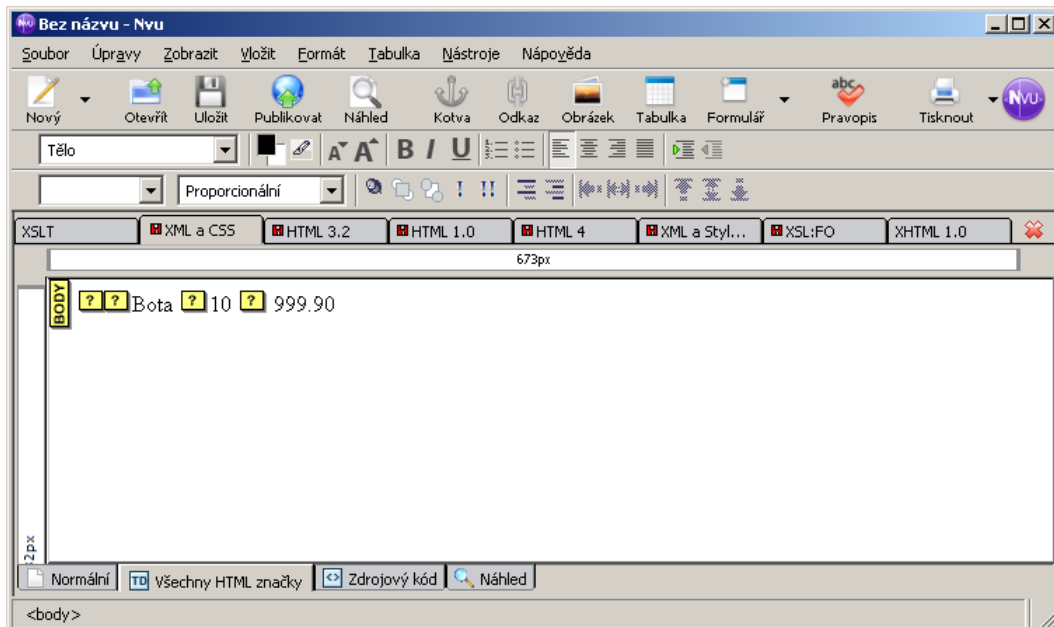
## Příloha č.8 – XHTML 1.0



## Příloha č.9 – XML a Stylesheet

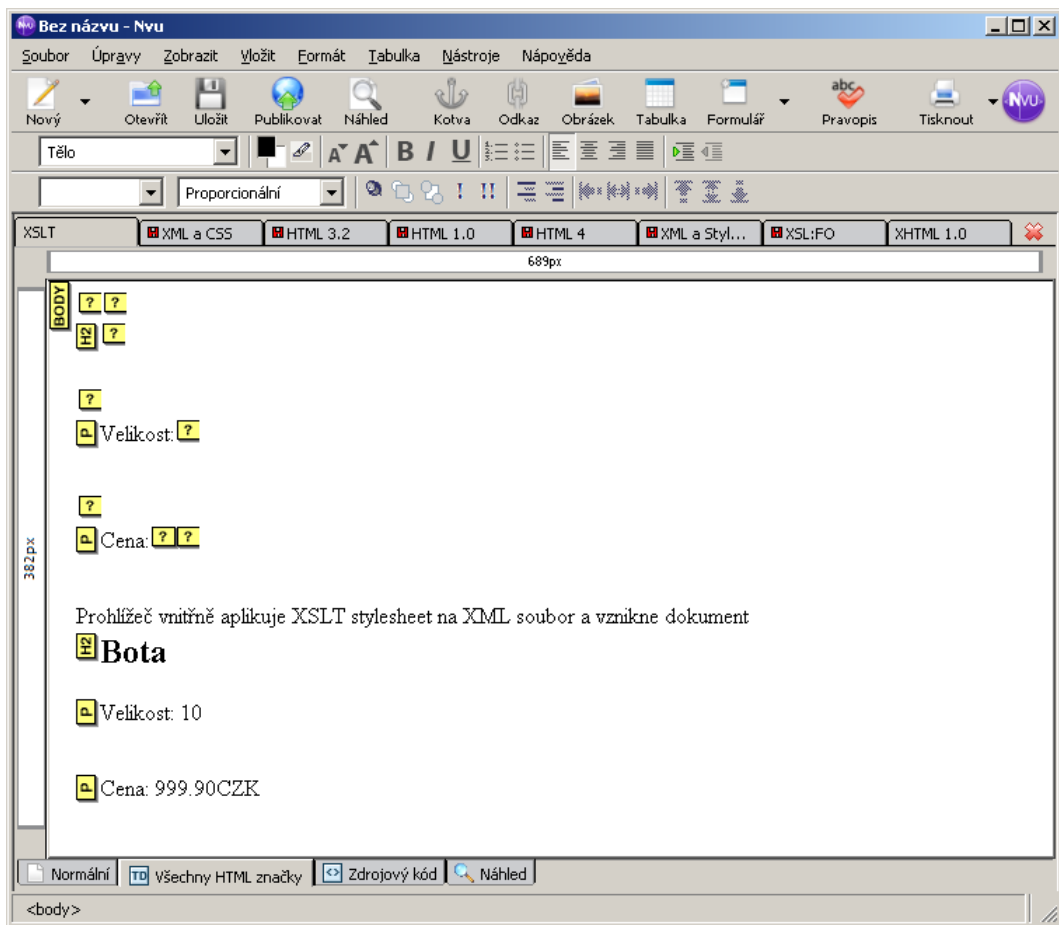


## Příloha č.10 – XML a CSS

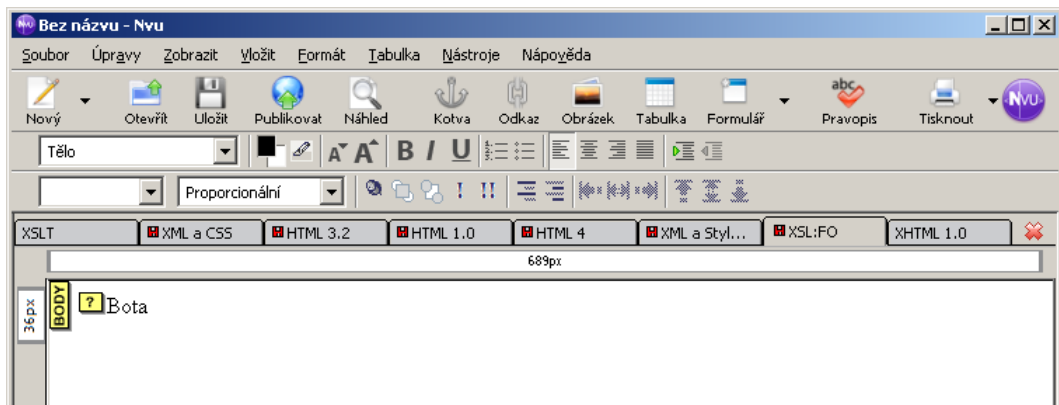




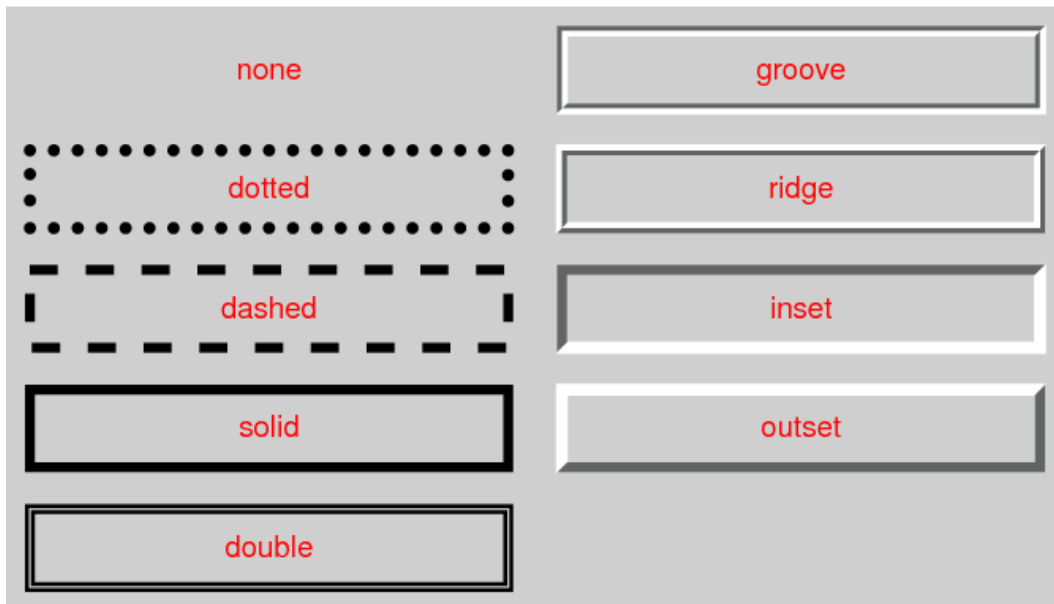
## Příloha č.11 - XSLT



## Příloha č.12 – XSL:FO



### Příloha č.13 – možnosti orámování CSS



### Příloha č.14 – struktura a hodnocení testovaného webu

