



Gymnázium a Jazyková škola s právem státní jazykové zkoušky Zlín
2010

Přepínání CSS stylů a správa emailových adres

The switching of CSS styles and the administration of the database

závěrečná práce ze semináře z informatiky

Jakub Macák, IV.A

Prohlášení

Prohlašuji, že jsem svou závěrečnou práci vypracoval samostatně a použil jsem pouze podklady (literaturu, projekty, SW atd.) uvedené v příloženém seznamu.

Poděkování

Rád bych touto cestou poděkoval panu učiteli Michalu Miklášovi za vstřícnost a cenné rady při tvorbě nejen této práce.

Abstrakt

Tato práce se zabývá tvorbou internetových stránek, přepínání jednotlivých CSS stylů a správu jednoduché databáze registrovaných uživatelů. Výsledná webová stránka byla navrhována pro reklamní agenturu. Web má být nejen účinný, ale měl by návštěvníka i zaujmout. Grafický návrh webu popisuje má závěrečná práce z předmětu Počítačová grafika a multimédia.

Abstract

This work deals with the website creation process, the switching of CSS styles, the simple administration and the database of registred users. The website was designed for an advertising agency. Web is not only effective but it should maintain visitors' attention. Graphic design of this website is described in the final work from the subject "The Computer Graphics and Multimedia".

Klíčová slova

CSS, webová stránka, HTML, PHP, MySQL

Obsah

1	Úvod	1
1.1	Výběr práce	1
1.2	Cíl práce	1
2	Tvorba stránky	2
2.1	Layout	2
2.2	Menu stránky	2
3	Přepínač stylů CSS	4
3.1	Přepínání v prohlížeči	4
3.2	Přepínání pomocí JavaScriptu	4
3.3	Přepínání pomocí PHP	4
3.3.1	Vytvoření formuláře	4
3.3.2	Uložení do cookie	5
3.3.3	Výběr požadovaného CSS souboru	5
3.4	Další zpracování webové stránky	5
4	Databáze e-mailových adres	6
4.1	Návrh administračního prostředí	6
4.2	Registrace zákazníka	6
4.3	Administrace adres	6
4.3.1	Přihlášení	6
4.3.2	Obsluha systému	7
4.3.3	Nápověda	8
4.4	Žádost o zaslání hesla k administraci	8
5	Instalace a zprovoznění stránek	10
6	Krátce o zpracování textu L^AT_EXem	11
6.1	Hlavička dokumentu	11
6.2	Použité prostředí	11
6.3	Struktura dokumentu, obsah a číslování	11
6.4	Další specifika dokumentu	11
7	Použité programy	12
8	Internetové zdroje	13
9	Seznam literatury	14

1 Úvod

1.1 Výběr práce

Vzhled webových stránek se od dob jejich vzniku velmi změnil. V dnešní době mají vzhled a funkčnost stejnou váhu; na rozdíl od dob minulých, kdy bylo třeba podat návštěvníkovi přesné a přehledné informace. Dnes má téměř každá společnost svou webovou stránku a s tím souvisí i potřeba odlišit se od ostatních. To byl jeden z důvodů, proč jsem si za svou závěrečnou práci zvolil právě návrh neobvyklé firemní prezentace a přepínač CSS stylů, který znamená konec stereotypu pro návštěvníky, a tím zvyšuje pravděpodobnost zájmu o služby.

Další součástí mé práce tvoří databáze registrovaných uživatelů. Pokud má uživatel zájem dostávat pravidelně nabídky dané společnosti, stačí mu pouze zaregistrovat svůj e-mail pomocí formuláře. Tím se společnosti otevírá další možnost, jak zaujmout zákazníky. Pokud by nabídky posílala pravidelně a byly by plné zajímavých informací, je vysoká pravděpodobnost, že zákazník dá přednost právě této firmě před jinou, která takovéto nadstandardní služby neposkytuje.

1.2 Cíl práce

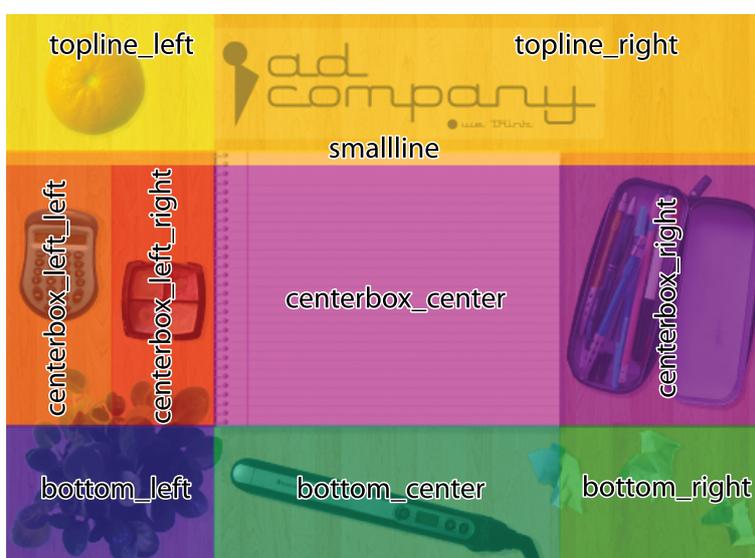
Hlavním cílem této práce je poukázat na možnosti přepínání vzhledů internetové stránky, jelikož mnoho webů tuto možnost svým návštěvníkům nenabízí, a ukázat odlišné technologie a postupy při tvorbě takové stránky a databáze.

2 Tvorba stránky

2.1 Layout

Pro prostředí svých webových stránek jsem využil různé vzhledy stolů – konkrétně stůl teenagera, stůl kutila a stůl počítačového nadšence. Jednotlivé předměty na stolech představují položky menu, pomocí nichž se uživatel na stránce pohybuje a které mají určitou spojitost s oborem, jakým se společnost zabývá.

Při navrhování layoutu¹ bylo potřeba dodržet několik pravidel. Rozložení stránky jsem koncipoval tak, aby každá část byla v různých vzhledech stejně vysoká a široká. Proto jsem volil jednotné rozložení divů², které jsou popsány na obrázku níže. Celkové rozměry webové stránky jsou 655 px (výška) na 900 px (šířka).



Obrázek 1: Layout s popisem divů

2.2 Menu stránky

Jednotlivé předměty položené na stole představují položky menu. Co daný předmět představuje, uživatel nezjistí hned při prvním pohledu na stránku, ale až tehdy, jakmile přejede přes objekt, jak lze vidět na dalším obrázku.

Tohoto efektu je docíleno pomocí CSS³ a vlastnosti `hover`. Takto vypadá kód pro jednu položku menu ve stylu CSS.

```
#topline_left {background-image: url('images/topleft.jpg');
                height: 163px;
                width: 245px;
                float: left;}
```

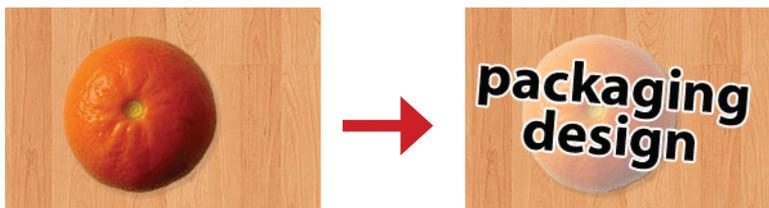
¹rozložení stránky

²oddíl stránky

³Cascading Style Sheets – tabulky kaskádových stylů

```
#topline_left a {height: 163px;
                  width: 245px;
                  float: left;}

#topline_left a:hover {background-image: url('images/topleft_hover.jpg');
                       height: 163px;
                       width: 245px;
                       float: left;}
```



Obrázek 2: Položka menu před a po přejetí myší

Pomocí téměř stejného kódu jsou udělány i další položky menu. Aby ale byla stránka zajímavější, jsou předměty a odkazy v dalších vzhledech různě rozházeny. Aby bylo možné toto udělat, je potřeba použít PHP⁴. Pro všechny tři styly existuje pouze jeden soubor `index.php` a pomocí PHP můžeme vytvářet stránku podle potřeb uživatele. V proměnné `$styl` je uložen název aktivního stylu, a tak je možné vyrobit jednoduchou podmínku, která nám právě pomocí této proměnné vytvoří různé odkazy.

```
<?php if ($styl=="styl_teenager/styl_teenager"){
print ("

```

V této podmínce se nám vyskytuje i další proměnná `$page`. Ta slouží pro další část stránky – `div` s hlavním obsahem. Obsah této proměnné získáme pomocí funkce `$_GET` následujícím způsobem:

```
<?php $page = $_GET["page"];?>
```

Poté stačí opět vytvořit jednoduchou podmínku, která nám podle zadaných parametrů vytiskne požadovaný obsah.

```
if ($page=="") {
print("<h1>Welcome</h1>
<p>AdCompany </p>");
}
if ($page=="packing") {
print ("

# 


```

⁴Hypertext Preprocessor – Hypertextový preprocesor

3 Přepínač stylů CSS

Jak jsem zmiňoval výše, přepínač CSS stylů slouží hlavně pro zpestření uživatelské návštěvy na stránce. V současné době můžeme tuto funkci najít například na webových stránkách měst a obcí, kde slouží pro zpřístupnění stránek všem obyvatelům. Nejedná se o přehlídku grafických návrhů a novinek, ale hlavně o zvětšování nebo zmenšování písma, přepínání pouze na textovou verzi apod. Jako příklad lze uvést webové stránky obce Tlumačov http://www.tlumacov.cz/servis/zmena_stylu.php a webové stránky města Otrokovice <http://www.otrokovice.cz> (odkaz vpravo nahoře – přístupný web). Existuje několik možností, jak lze přepínání realizovat. Rád bych jednotlivé možnosti alespoň stručně popsal a zaměřil se hlavně na využití PHP, pomocí kterého funguje přepínání i na mých webových stránkách.

3.1 Přepínání v prohlížeči

První možností je přepínání vzhledu přímo pomocí webového prohlížeče. Tato možnost je dle mého názoru pro uživatele nejméně příjemná a navíc ne moc lidí o této volbě ví. Navíc ne všechny prohlížeče (Internet Explorer) tuto funkci uživateli nabízí. Pokud ale chceme toto přepínání použít, musíme jej zaznamenat do hlavičky webové stránky. V místě, kde připojujeme šablonu CSS, změním atribut `rel="stylesheet"` na `rel="alternate stylesheet"`. Musíme použít také atribut `title`, ve kterém bude napsán název daného stylu. Ten se bude také zobrazovat při výběru v prohlížeči. Finální zdrojový kód by mohl vypadat asi takto:

```
<link rel="stylesheet" type="text/css" href="1.css" title="1" />
<link rel="alternate stylesheet" type="text/css" href="2.css" title="2" />
...
```

3.2 Přepínání pomocí JavaScriptu

Další možnost jak zrealizovat přepínání stylů je pomocí JavaScriptu⁵. Zde by bylo na místě vysvětlit rozdíl mezi PHP a JavaScriptem. Program v JavaScriptu se spouští až po stažení WWW stránky z internetu – na straně klienta, kdežto PHP pracuje na straně serveru a uživateli je přenášén až konečný výsledek.

V JavaScriptu bohužel programovat neumím, chtěl jsem pouze na tuto možnost upozornit. Na internetu existuje mnoho hotových příkladů, které lze použít přímo na svých webových stránkách. Jeden takový postup vyšel na serveru Interval, na adrese <http://interval.cz/clanky/alternativni-styly-zmena-stylu-pomoci-javascriptu/>.

3.3 Přepínání pomocí PHP

Přepínání stylů pomocí PHP jsem použil pro své stránky. Ve své práci využívám cookie⁶ a jednoduchý formulář pro výběr stylu.

3.3.1 Vytvoření formuláře

Aby si uživatel mohl požadovaný styl vybrat, je potřeba vytvořit jednoduchý formulář. Pro výběr stylu jsem zvolil položky typu `radio`. Formulář jsem umístil do `divu` v pravém horním

⁵multiplatformní, objektově orientovaný skriptovací jazyk

⁶malé množství dat, která server pošle prohlížeči, který je uloží na počítači uživatele

rohu stránky. Takto vypadá jedna položka formuláře.

```
<input name="styl" type="radio" value="styl_it/styl_it" />IT style<br />
```

Dále je třeba vytvořit `input` typu `submit`. Ten slouží pro potvrzení vybrané informace. Důležitý je také atribut `name`, který je třeba pro další zpracování formuláře. Nyní formulář zpracujeme pomocí PHP. Metodou `POST` získáme informace o tom, co se stane po kliku na tlačítko `submit`. Pokud se tedy hodnota proměnné `$odeslatformular` rovná řetězci `submit`, pak uložíme do proměnné `$styl` název stylu, který si uživatel zvolil. Pokud uživatel žádný styl nevybere, uloží se do této proměnné `styl_it`.

3.3.2 Uložení do cookie

Aby tedy bylo možné uchovat název vybraného stylu i pro další prohlížení stránky nebo pro další návštěvu uživatele, je potřeba tuto informaci uložit do cookie; ta se skládá z několika částí.

```
setCookie ("název", "hodnota", doba trvání, "cesta", "server");
```

V mé práci má cookie následující podobu:

```
setCookie("csscookie", $styl, time()+3600*24*365, "/", "", 0);
```

Ze zadaných hodnot tedy vyplývá, že se do této cookie uloží proměnná `$styl` na dobu jednoho roku. Tím pádem může uživatel ze stránky odejít a při jeho další návštěvě se vybere poslední zvolený styl. Na začátku PHP skriptu probíhá kontrola, která ověřuje, zda taková informace existuje.

3.3.3 Výběr požadovaného CSS souboru

Výběr požadovaného CSS souboru už tedy nebude problém. Do atributu `href` se vypíše proměnná `$styl`, která odkazuje na složku s daným CSS předpisem a obrázky.

3.4 Další zpracování webové stránky

Fungující webová stránka je téměř hotová. Problém by mohl nastat v případě, že by bylo vloženo příliš mnoho textu. V úvahu by tedy přicházela dvě řešení. První, které dle mého názoru není příliš šťastné ani dobře vypadající je použití CSS předpisu `overflow: auto`. V tomto případě by se na pravý okraj stránky umístila rolovací lišta, která by nám umožňovala se v textu pohybovat. Druhé, asi lepší řešení by bylo použití stránkování. Text bychom si mohli rozdělit na určité úseky a ty poté pomocí PHP tisknout do tohoto `divu`. Vespod by se pak nacházelo stránkování, kterým by uživatel přepínal stránky.

4 Databáze e-mailových adres

Další součástí mé závěrečné práce bylo vytvořit systém, kam by se mohli registrovat uživatelé, jenž mají zájem dostávat aktuální zprávy e-mailem a administrace této databáze.

4.1 Návrh administračního prostředí

Pro administrační a registrační prostředí jsem využil podobného layoutu jako na hlavních stránkách webu. Toto prostředí jsem pouze barevně odlišil a vytvořil nový banner, který nás informuje o tom, že se nacházíme v jiné sekci webu.

Pro správné fungování systému bylo třeba vytvořit několik speciálních stránek. První je pro uživatele – registrace. Další jsou pak pro obsluhu systému – tedy přihlašovací stránka, samotná databáze, nápověda a formulář pro zaměstnance, kteří se nemohou do systému přihlásit, jelikož neznají heslo. V každé stránce se nachází i jiné menu s položkami, které daný uživatel potřebuje.

4.2 Registrace zákazníka

Nejdříve bych se chtěl zaměřit na formulář, kterým se potenciální zákazník zaregistruje do systému. Na tuto stránku se dostane po kliknutí na odkaz Registration v pravém horním rohu hlavní stránky, hned nad přepínačem CSS stylů.

Na této stránce se nachází formulář, pomocí kterého se zákazník zaregistruje. Vyžadován je po něm jen e-mail. Po zadání položek do formuláře a kliknutí na tlačítko `submit` se informace odešlou. Vše je řešeno pomocí metody `POST`. Jednotlivé položky se uloží do proměnných a dále se kontrolují. Je také zapotřebí zmínit, že ve chvíli, kdy uživatel stiskne tlačítko `submit`, proběhne připojení k databázi pomocí funkce `connect_to_dtb ($link)` s parametrem `$link`. Tato funkce se nachází ve speciálním souboru `utils.php`, kde jsou základní funkce uloženy. Tato stránka je pak zavolána pomocí funkce `require("utils.php");`. Nyní ale zpět ke zpracování formuláře. Nejprve jsou všechny hodnoty očištěny o zbytečné mezery z obou stran pomocí funkce `trim`. Jelikož vyžadujeme pouze e-mailovou adresu, budeme kontrolovat jen ji, a to postupně pomocí několika podmínek. Samozřejmě nesmí být adresa prázdná. Pokud se tak stane, vypíšeme uživateli varovnou zprávu a data neodešleme. Další podmínkou je správný tvar adresy. To řešíme funkcí `check_email`, opět uloženou v souboru `utils.php`. Tuto funkci jsem nevymyslel sám, ale použil jsem hotový příklad[1]. Pokud proměnná projde i přes tuto podmínku, zbývá nám zkontrolovat, jestli zadaný e-mail již není v databázi. Spočítáme tedy všechny uložené adresy a pokud se žádná taková v databázi nevyskytuje, můžeme data uložit a to pomocí PHP a MySQL⁷ příkazu `mysql_query` a `INSERT INTO`. Uživateli nakonec vypíšeme informaci o úspěšném uložení a ukončíme spojení s databází.

4.3 Administrace adres

4.3.1 Přihlášení

Aby se uživatel mohl dostat k administraci adres, je po něm vyžadováno heslo. Celý postup je realizován skriptem Page Password Protect 2.13[2]. V PHP poli je uloženo heslo, pomocí kterého se uživatel přihlásí. Na mých stránkách je třeba zadat heslo 123456. Pokud zadá správné údaje, tato informace se uloží do cookie, vše je však zašifrováno funkcí `md5`. Dále

⁷databázový systém

se pak přesměruje stránka přímo na tabulku, kde se vypíše všichni registrovaní uživatelé, což provede funkce `header('Location: ');`.

4.3.2 Obsluha systému

Po přihlášení se tedy uživatel dostává do samotné databáze. Zde má několik možností, co bude s tabulkou dělat. Data může seřazovat podle ID, měnit informaci o zásílání e-mailu nebo uživatele úplně smazat. Adresy si může také otevřít v novém okně, aby je mohl jednodušeji vložit do poštovního klienta a rozeslat zprávy. Rozvržení tabulky znázorňuje následující obrázek:

User ID	Name	Surname	Email	Adress	Company	Sending emails	Delete
1	Jakub	Macák	macak.kuba@seznam.cz	Hložkova 358 Otrokovice		✉	✖
2	Lukáš	Mikeska	mikeska@gorkon.cz	Pozlovice		✉	✖
3	Jan	Holý	holy@it.cz	Napjedla 302	Klávesnice s.r.o.	✉	✖

Obrázek 3: Rozvržení tabulky

Celá tabulka představuje právě jednu tabulku databáze, která se jmenuje `registred_users_email`. Zde se nacházejí stejné atributy, konkrétně `id`, `name`, `surname`, `email`, `adress`, `company`, `sending_emails` a `del`. Tabulka se vypisuje postupně, a to díky proměnné `$html`, která se vypíše až ke konci celého skriptu.

Pokud chce uživatel některou hodnotu měnit, stačí na ni kliknout. Všechny obrázky jsou proto odkazy. Každý odkaz si v URL nese potřebné informace.

Nejdříve bych popsal skript obsluhující seřazování podle ID. Pomocí funkce `$_GET` získám proměnnou `$sort`. Pokud se zde taková proměnná nevskytuje, nastavím výchozí hodnotu proměnné `$razeni` na `ASC`⁸. Pokud uživatel klikne na jednu z ikon pro seřezování, pošle se hodnota přes URL nastavená v odkazu obrázku a aktualizují databázový příkaz `ORDER BY`, podle kterého se upřesňuje výběr dat z databáze. Je nutné posílat tuto proměnnou i v situacích, kdy měním informace o posílání e-mailu nebo mažu záznam, protože chci, aby řazení zůstalo stejné jako před vykonáním tohoto požadavku.

Další ovladatelnou položkou je změna informace o posílání e-mailu. Zde je princip podobný.

```
if (!MyIsNull($sending_emails) and !MyIsNull($id)) {
if ($sending_emails==0) {
$sending_emails=1;
}else{
$sending_emails=0;
}
$sqry=("UPDATE registred_users_email SET
sending_emails=".$sending_emails." WHERE id=".$id." LIMIT 1");
$vysledek=mysql_query($sqry);
MySQL_Select_DB($maindatabase);
sqlr=MySQL_Query($sqry);
```

⁸ascending – vzestupně

```

if ($sqlr) {
print ("
```

Takto vypadá celá podmínka pro požadovanou změnu. Funkce `MyIsNull` kontroluje, je-li daná proměnná prázdná. Pokud mám tedy proměnné `$sending_emails` a `$id`, mohu měnit hodnotu na opačnou, než jaká byla před kliknutím. Poté databázovým příkazem `UPDATE` změním hodnotu, a to právě u jednoho konkrétního řádku. Odkaz pro obrázek pak vypadá následovně:

```

<a href=\"admin_table.php?id=".$zaznam["id"]."&sending_emails=
".$zaznam["sending_emails"]."&sort=".$sort.">");
if ($zaznam["sending_emails"]=="1"){
$html = $html.("<img src=\"images/edit/mail.jpg\" alt=\"mail\"
border=\"0\" title=\"send\" ".$MY_DELCONFIRM_CHANGE_EMAIL."/>");
}else{
$html = $html.("<img src=\"images/edit/nomail.jpg\" alt=\"nomail\"
border=\"0\" title=\"nosend\" ".$MY_DELCONFIRM_CHANGE_EMAIL." />");}

```

V podmínce `if` akorát měním obrázky podle toho, zda se uživateli mají zasílat zprávy či nikoliv, a proměnná `$MY_DELCONFIRM_CHANGE_EMAIL` zobrazuje vyskakovací okno pro potvrzení o vykonání příkazu. Opět se nachází v souboru `utils.php`. Nakonec uživateli vypíše hlášku, zda se vše podařilo. Téměř totožně je pak zajištěno i mazání řádků z tabulky. Nejedná se však o smazání konečné, v databázi řádek zůstane, jen jej uživatel neuvidí.

Další nedílnou součástí je řádkování. Použil jsem hotový a fungující skript z internetu^[3]. Princip je takový, že se pomocí funkce `mysql_num_rows` spočítá počet řádků v databázi, určí se počet záznamů na stránku. Využije se i funkce `ceil`, která vrací nejbližší celé vyšší číslo. Přes URL si pak přepínám jednotlivé stránky a cyklem `for` vypíše jednotlivé stránky dolů pod tabulku jako odkazy.

Poslední částí této stránky je výpis všech platných adres do nového okna stránky, kterou takto otevírám pomocí JavaScriptu. V samotné stránce není pak nic jiného, než jednoduchý výpis položek, aby uživatel mohl adresy použít a zaslat na ně nabídky.

Rád bych zmínil, že přístup do stránky s tabulkou i s výpisem adres je kontrolován a pokud uživatel nemá přihlašovací cookie a zadá absolutní adresu do prohlížeče, skript jej vrátí na přihlašovací stránku.

4.3.3 Nápověda

Součástí celého systému je i jednoduchá nápověda. Tabulka popisuje uživatelům, co která položka znamená a co se stane po kliku na ni. Nápověda je jistě potřeba, zvláště začátečníci by mohli mít problémy s obsluhou systému a tato stránka by jim mohla práci usnadnit.

4.4 Žádost o zaslání hesla k administraci

Poslední stránkou celého systému je formulář, který umožňuje podat žádost o zaslání hesla k administraci. Je koncipován stejně jako předchozí, data se však neuloží do databáze, ale

pošlou se e-mailem. Povinné jsou všechny položky; opět se kontroluje správnost zadané adresy a zda telefon obsahuje jen čísla. K posílání zpráv slouží v PHP funkce `mail` a celá vypadá takto:

```
mail("admin@admin.com", $email, $text, $headers);
```

Zde musí být zadána adresa, na kterou bude zpráva zaslána, proměnná `$email` nám udává odesílatele, `$text` je samotný text zprávy a v `$headers` jsou uložena všechna data potřebná k poslání zprávy – kódování, typ zprávy, předmět atd.

5 Instalace a zprovoznění stránek

Pro správné fungování mé práce je třeba nainstalovat webový server (např. WampServer) a webový prohlížeč, nejlépe Mozilla Firefox. Celou složku je poté nutné uložit do adresáře `www` ve složce `wamp` na disku. Dále je zapotřebí importovat databázi do stránky phpMyAdmin. Databáze se nachází v textovém souboru ve složce `projekt_macak` a její název je `projekt_macak_dtb.txt`. Vytvoří se tak databáze s názvem `projekt_macak` a tabulka s názvem `registred_users_email`.

6 Krátce o zpracování textu L^AT_EXem

Pro zpracování tohoto dokumentu jsem zvolil sazbu systémem L^AT_EX, jelikož nabízí mnoho možností pro psaní nejen matematických textů, ale například i textů právě do informatiky.

6.1 Hlavička dokumentu

Aby byl dokument správně zpracován, je třeba hned v úvodu dokumentu nastavit potřebné parametry. Nejdůležitějšími jsou právě `\documentclass` nebo `\usepackage`, díky kterému se správně vysází české znaky.

6.2 Použité prostředí

V této práci je použito i několik prostředí. Nejvíce jsem využíval prostředí `verbatim`, které velmi jednoduše a přehledně odliší normální text od zdrojových kódů. Pro sazbu zdrojového kódu na řádek slouží příkaz `\verb`. V práci je použito i několik obrázků. K jejich vkládání je použito příkazu `\begin{figure}` a `\end{figure}`. Díky programu T_EXnicCenter bylo však jejich vložení do textu velmi jednoduché, a to díky položce v menu.

6.3 Struktura dokumentu, obsah a číslování

Velkou výhodou sazby L^AT_EXem je strukturování dokumentu a následné generování obsahu. Pro rozlišení nadpisů jsem využil příkazů `\section` pro hlavní nadpis, `\subsection` pro nadpis druhé úrovně a `\subsubsection` pro nadpis třetí úrovně. Generování obsahu je pak velmi snadné, jelikož jsem nemusel řešit žádné číslování nadpisů, L^AT_EXse o vše postará sám. Obsah jsem pak vygeneroval příkazem `\tableofcontents`. Číslování stránek je také velmi jednoduché, jen v úvodu je použito římských číslic a pro další text číslic arabských.

6.4 Další specifika dokumentu

Pro sazbu textu jsem využil i mnoho dalších příkazů, které jsou zde k dispozici. Jako příklad bych uvedl příkaz `\footnote`. Ten usnadní vypsání poznámek pod čarou, zároveň se u slov označených tímto příkazem udělají odkazy. K sazbě na další stránku slouží příkaz `\newpage` a k vytvoření vertikálních mezer příkaz `\vspace{}`. K nahrazení mezer u jednopísmených předložek jsem použil program OpenOffice.org Writer. Text stačí vložit do tohoto programu, a poté pomocí regulárních výrazů tyto mezery naharadit. Příkaz pro nalezení vypadá následovně

```
([ ]{1}[:alpha:]{1}) ([:alpha:]*\>).
```

Příkaz `$1~$2` slouží pro nahrazení těchto mezer nezalomitelnou mezerou. Návod je možné najít na stránkách Gymnázia a Jazykové školy Zlín.^[4]

7 Použité programy

PSPad, verze 4.5.3 (2291) PSPad je univerzální freeware editor, určený nejen pro psaní webových stránek. Slouží také jako jednoduchý textový editor. Při tvorbě stránek jsem ocenil barevné zvýraňování kódu, označování chybějících závorek či nahrazování v textu.

Adobe Photoshop CS4 Extended, verze 11.0 Adobe Photoshop je grafický editor s velkým množstvím funkcí. Program jsem využil při grafickém návrhu stránek. Pro tvorbu internetových stránek program nabízí skvělou funkci – uložení pro web. Jednoduché nařezání všech částí ušetří mnoho času.

WampServer, version 2.0 WampServer je program pro tvorbu webových aplikací pomocí Apache, PHP a MySQL databáze. Program je velmi jednoduchý na ovládání a zaručuje bezproblémový chod a test webových aplikací.

Mozilla Firefox, verze 3.6.3 Mozilla Firefox je webový prohlížeč vyvíjený společností Mozilla Corporation. Při práci na projektu jsem jej využil na prohlížení webových stránek nebo pro zobrazení výsledného zdrojového kódu.

T_EXnicCenter, 1.0 Stable Release Candidate 1 T_EXnicCenter je editor sloužící pro sazbu textu v L^AT_EXu. Pro usnadnění práce je zde mnoho funkcí jako vkládání symbolů nebo obrázků. Při psaní tohoto textu jsem využil distribuci MikT_EX2.7.

8 Internetové zdroje

Reference

- [1] **VRÁNA, Jakub:** *Kontrola e-mailové adresy*
Přístup z internetu:
<http://php.vrana.cz/kontrola-e-mailove-adresy.php> 4.2
- [2] **Zubrag.com:** *Web Page Password Protect*
Přístup z internetu:
<http://www.zubrag.com/scripts/password-protect.php> 4.3.1
- [3] **BRONISLAV, Filip:** *Nekonečné stránkování v PHP*
Přístup z internetu:
<http://crazydog.cz/pro-web/php-scripty/nekonecne-strankovani-v-php/> 4.3.2
- [4] **Webové stránky Gymnázia a Jazykové školy s právem státní jazykové zkoušky Zlín:** *Základní typografická pravidla při tvorbě dokumentů*
Přístup z internetu:
<http://www.gjszlin.cz/gztgm/vyuka/ivt/esf/ostatni-gdm/zakladni-typograficka-pravidla-esf.pdf> 6.4
- [5] **ŠVAMBERG, Michal:** *Seriál Jak na LaTeX*
Přístup z internetu:
<http://www.root.cz/serialy/jak-na-latex/>
- [6] **STANÍČEK, Petr:** *PHP přepínač stylů*
Přístup z internetu:
http://www.pixy.cz/pixylophone/2003_11_archiv.html
- [7] **Jak psát web:** *Jak psát web, o tvorbě internetových stránek*
Přístup z internetu:
<http://www.jakpsatweb.cz/>
- [8] **Php.net:** *Php.net*
Přístup z internetu:
<http://php.net/>

9 Seznam literatury

Reference

- [1] **DOMES, Martin:** *333 tipů a triků pro CSS*.
1. vyd. 2009 : Computer Press, a.s., ISBN 978-80-251-2360-7
- [2] **Kolektiv autorů:** *PHP Programujeme profesionálně*.
1. vyd. 2001 : Computer Press, a.s., ISBN 807-226-3102
- [3] **CASTRO Elizabeth:** *HTML, XHTML a CSS*.
1. vyd. 2007 : Computer Press, a.s., ISBN 978-80-251-1531-2