

Vysoká škola ekonomická v Praze
Fakulta informatiky a statistiky
Vyšší odborná škola informačních služeb v Praze

Jaroslav Mika

Proces tvorby webových prezentací

Bakalářská práce

< 2006 >

Prohlašuji, že jsem bakalářskou práci na téma Proces tvorby webových prezentací zpracoval samostatně a použil pouze zdrojů, které cituji a uvádím v seznamu použité literatury.

V Praze dne 20. 8. 2006

Obsah:

1. Internet a služba WWW	7
2. Funkční návrh stránek	9
2.1 Struktura stránek	9
2.1.1 Hierarchická struktura	9
2.1.2. Globální struktura	9
2.1.3 Lokální struktura	9
2.2 Navigace mezi stránkami	10
2.2.1 Horizontální systémy	10
2.2.2 Vertikální systémy	10
2.2.3 Stahovací systémy	10
2.3 Rozvržení (layout) stránek	10
2.3.1 Jednosloupcová struktura	11
2.3.2 Vícesloupcová struktura	11
2.4 Umístění stránek a volba domény	12
3. Grafická podoba stránky	14
3.1 Písmo	14
3.2 Barvy	15
3.3 Obrázky a fotografie	16
3.3.1 Velikost obrázků	16
3.3.2 Grafické formáty	17
3.3.3 Animace na stránkách	18
4. Zdrojový kód	19
4.1 Kaskádové styly	20
4.1.1 Definice kaskádových stylů	20
4.1.2 Selektory	21
4.1.3 Pseudoprvky a pseudotřídy	23
4.1.4 Slučování stylů	24
4.1.5 Vložení stylu do stránky	24
4.1.6 Pravidla pro používání stylů	25

4.2 Rozvržení stránky (<i>layout stránky</i>)	26
4.2.1 Pomocí kaskádových stylů	26
4.2.2 Využití tabulek	27
4.2.3 Využití rámců (<i>frames</i>)	28
4.3 Rozdílné prohlížeče	28
4.4 Zdrojový kód v HTML souboru	30
4.5 Validace stránek	32
5. Optimalizace stránek	33
5.1 SEM (<i>Search Engine Marketing</i>)	33
5.2 SEO (<i>Search Engine Optimization</i>)	33
6. Vytváření stránek Prague Golf Tours	35
6.1 Funkční model stránek	35
6.1.1 Struktura stránek	35
6.1.2 Navigační systémy	36
6.1.3 Rozvržení stránek	36
6.2 Vytváření stránek	37
6.2.1 Element <code><body></code>	37
6.2.2 Layout úvodní stránky	38
6.2.3 Přepínání mezi aktivními sekcemi	47
6.2.4 Vytváření podstránek	47
6.2.5 Problémy se stránkami	50
6.3 Optimalizace stránek	51
6.4 Závěrečná korektura	51
7. Použité zdroje:	52

Anotace

Jak již napovídá název, chci ve své práci ukázat postup při vytváření webových stránek. Cílem je ukázat posloupnost prací, které vedli k vytvoření jedné, konkrétné webové stránky.

Prvním záměrem je představit možné metody a postupy, které se dají v různých fázích tvorby webových stránek použít. Ať už při vytváření struktury stránek, jejich grafické podoby nebo samotném vytváření stránek v jazyku HTML. Zejména tomuto poslednímu tématu je věnována velká část práce. Důraz jsem v této části kladl zejména na využívání kaskádových stylů.

Část práce je věnována tématu optimalizace stránek. Jde o vytvoření (upravení) stránky tak, aby co nejlépe vyhovovala fulltextovým vyhledávačům a byli jimi co nejlépe indexována.

V závěrečné části je ukázán postup práce na konkrétní webové prezentaci, právě s využitím některých z již dříve popsaných metod a postupů, které se jeví pro danou stránku jako nejvýhodnější i se zdůvodněním. Důraz je i v této části práce kladen zejména na práci v jazyku HTML a kaskádových stylech.

1. Internet a služba WWW

Co je internet? Pokud bude odpověď znít, že internet je celosvětová síť, tak odpověď nebude zcela správná. Přesnější by bylo říci, že internet je decentralizovaná soustava navzájem propojených počítačových sítí, které jsou založeny na společném základu. To co dílčí sítě spojuje je soustava protokolů označovaná jako TCP/IP. Propojení jednotlivých sítí je takové, že uživatel může výsledný efekt považovat za jednu jedinou síť, ačkoliv jednotlivé počítače či sítě tvořící internet mohou být různých typů.

Původně byl internet založen pro podporu a urychlení výzkumných a vývojových projektů. Jednotlivá vědecká pracoviště mohla sdílet své výpočetní prostředky, vyměňovat si pracovní zprávy nebo informace. Postupem času se ukázalo, že právě výměna informací je tím nejpoužívanějším co internet přinesl – zrodila se elektronická pošta – e-mail.

Tak jak se internet postupem doby rozvíjel, přibývalo i služeb, které mohou uživatele využívat. Jednou z nich je i služba WWW.

WWW (World Wide Web)

Společně s elektronickou poštou nejvyužívanější služba internetu. Jedná se o síť navzájem propojených hypertextových dokumentů.

Samotná služba www stojí na třech základních kamenech. Prvním je jazyk HTML (HyperText Markup Language), který slouží k popisu obsahu a podoby webové stránky.

Druhým kamenem je protokol http (či šifrovaný https), což je protokol používaný při komunikaci mezi prohlížečem a webovým serverem. Definuje podobu přenášených informací, možnosti a náležitosti požadavku i možné odpovědi (viz tabulka).

Jedná se o bezstavový protokol fungující na principu požadavek – odpověď. Klient (internetový prohlížeč) si nejprve přes DNS zjistí IP adresu dané domény a naváže komunikaci. Následně zašle požadavek, na který daný server odpoví. Komunikace se po té uzavírá. Pro další požadavek je třeba otevřít komunikaci novou.

Tabulka 1 – Kategorie odpovědí serveru

Kategorie stavových kódů	Čísla stavových kódů	Popis
Informační	100 - 199	Zprávy definované konkrétní aplikací
Úspěch	200 - 299	Požadavek byl úspěšně zpracován
Přesměrování	300 - 399	Klient musí pro konečné zpracování požadavku vykonat určitou další činnost
Chyba klienta	400 - 499	Problémy na straně klienta
Chyba serveru	500 - 599	Problémy na straně serveru

(čerpáno z [11])

Třetím jsou již zmíněné odkazy. Ty se zapisují ve standardizované podobě pomocí URL (Uniform Resource Locator). Obecný tvar URL je následující:

schéma://uživatel:heslo@počítač:port/cesta;parametry?dotaz#fragment

Schéma určuje typ služby, na který adresa ukazuje (obvykle http, nebo FTP). Uživatel a heslo udávají „jméno“ uživatele a heslo, pod kterým se k dané službě uživatel přihlašuje.

Tabulka 2 – Typy služeb

Schéma	Popis
file	soubor na lokálním disku
FTP	soubor přístupný přes službu FTP
gopher	Gopher protokol
http	webové stránky
mailto	adresa elektronické pošty
telnet	terminálový přístup ke vzdálenému počítači

(čerpáno z [5]; str. 24)

Počítač udává adresu počítače, buď formou domény, nebo IP adresy. V případě, že není uveden uživatel, tak se vynechává „zavináč“ před adresou. Cesta udává jméno a cestu vedoucí k požadovanému souboru. Port udává číslo portu, na kterém je služba dostupná, a píše se v případě, že se nepoužívá standardní port (pro protokol http se standardně používá port 80).

Parametry a dotazy se využívají jen někdy, např. u dynamicky generovaných webových stránek, popřípadě při předávání dotazů vyhledávacím službám. Fragment se používá při odkazování na určitou část zdroje (např. jeden obrázek na webové stránce).

Pokud se obecný tvar URL převede na adresu pro www, dojde se k následující podobě:

http://počítač:port/cesta?dotaz#fragment

2. Funkční návrh stránek

Na začátku, ještě předtím než se začne na počítači vytvářet webová stránka, je třeba mít jasnou představu o tom, jak bude tato stránka vypadat a fungovat. Tedy mít jasno v následujících bodech:

2.1 Struktura stránek

Zde je cílem vymyslet systém uspořádání stránek. Nesmí se stát, že výsledkem bude zamotaná soustava chaoticky propojených stránek, ve které se návštěvník nebude schopen orientovat. Stránky musí být jasným způsobem uspořádány.

2.1.1 Hierarchická struktura

V této struktuře jsou stránky členěny do sekcí, sekce do podsekcí, ty pak do dalších podsekcí nižších úrovní. Na úvodní stránce je jen seznam hlavních sekcí, kliknutím na jednu z nich se návštěvník dostane na seznam podsekcí spadajících do dané sekce.

2.1.2. Globální struktura

Veškeré sekce či stránky webu jsou přístupné v jednom menu, které je stále viditelné. Obvykle je místo, kde se návštěvník nachází vyznačeno vůči ostatním položkám odlišnou barvou, nebo daná položka není v menu odkazem.

2.1.3 Lokální struktura

Krátký seznam odkazů umístěný obvykle pod hlavním textem stránky.

Obecně platí, že nejdůležitější informace - to hlavní co chtějí stránky prezentovat – by měly být maximální dvě kliknutí od titulní strany.

2.2 Navigace mezi stránkami

Jedním z klíčových prvků na webových stránkách je jednoduchá, přehledná a účelná navigace. Musí jednoznačně návštěvníkovi říci na kterém webu a v jaké jeho části se v danou chvíli nachází, kde již byl a musí ukazovat další možnosti postupu.

Každá stránka by měla obsahovat odkaz na stránku v struktuře nadřizenou či na stránku logicky předcházející a následující. Součástí stránky by měl být i odkaz na homepage (obvykle logo v horní levé části). U rozsáhlých webů je vhodné na domovskou stránku umístit odkaz na mapu stránek. Mapa zobrazuje celou strukturu webu a obsahuje odkazy na jednotlivé sekce a podsekce.

Pro umístění odkazů se obvykle používá jeden z těchto systémů:

2.2.1 Horizontální systémy

Navigační prvky (odkazy) jsou umístěny ve vodorovném směru v jednom či několika řádcích v horní části stránky. Výhodou tohoto systému je, že odkazy se zobrazí jako první, a uživatel tak může okamžitě pokračovat dál - nemusí čekat na načtení celé stránky.

2.2.2 Vertikální systémy

Navigační odkazy jsou umístěny v postraním (obvykle levém) sloupci. Výhodou je, že toto uspořádání umožňuje pojmout větší počet odkazů než vodorovné systémy, aniž by to působilo přeplněným dojmem.

2.2.3 Stahovací systémy

Navigační lišta je pevně ukotvena na určitém místě a nemění svou pozici. Jediné co se mění, je obsah jednotlivých sekcí.

2.3 Rozvržení (layout) stránek

Layout je rozmístění jednotlivých prvků (text, obrázek, tabulka, oddělovač atd.) na stránce. Pokud se zvolí určitý layout stránky, tak by se měl používat na všech stránkách daného webu. Samozřejmě, že v různých sekcích mohou být drobné rozdíly, ale základní styl a základní prvky by se neměly výrazně lišit.

2.3.1 Jednosloupcová struktura

Nejjednodušší způsob rozvržení webové stránky, kdy jednotlivé prvky obsahu stránky postupně plynou jeden za druhým – nadpis, podnadpisy, texty, grafika – vše je umístěno pod sebou v jednom sloupci.

Obrázek 1 – Jednosloupcová struktura



Tento design je dnes zastaralý. Ale právě jeho jednoduchost zajišťuje bezproblémové zobrazení stránky všemi druhy prohlížečů a ve všech možných velikostech oken prohlížeče.

2.3.2 Vícesloupcová struktura

V současnosti nejvíce používaný způsob rozvržení stránky. Použití více sloupců umožňuje do omezeného prostoru umístit více informací, aniž by došlo ke snížení přehlednosti.

Nejčastěji se používá dvou či tří sloupců. V levém sloupci bývají obvykle umístěny navigační prvky a text je v sloupci pravém (resp. v prostředím při použití tří sloupců). Při třech sloupcích bývá pravý použit pro podrobnější navigaci v dané sekci, nebo je na něm umístěna reklama, buď v podobě banneru či odkazů typu Pay-per-click.

Obrázek 2 – Vícesloupcová struktura



Musí se vzít v úvahu i rozměry stránky. Je třeba hlídat, aby stránka nebyla příliš dlouhá. Stránka s odkazy by se měla vejít celá na jednu obrazovku monitoru tak, aby uživatel viděl pohromadě všechny možnosti dalšího postupu. V případě stránek obsahující pouze text by neměla být stránka delší než několik obrazovek (5 – 10 obrazovek)

Na stránce by se neměl vyskytovat horizontální scrollbar. Takže je třeba hlídat i šířku stránky. Vzhledem k v dnešní době nejrozšířenějšímu rozlišení 1024x 768px by stránka neměla být širší než 1000px.

2.4 Umístění stránek a volba domény

Problém při volbě doménového jména je s tím, že množství domén je již zaregistrováno. Ověření, zda je určité jméno volné je ale jednoduché. Stačí kontaktovat libovolnou agenturu poskytující domény a ověřit si to. Samotná registrace je v současnosti zdarma. Platí se jen poplatek za vedení domény.

Pokud je zvolena a zaregistrována vhodná doména, je čas na rozhodnutí, kam stránky umístit. Stačí si vybrat webhostingovou agenturu, která poskytuje právě ty služby (PHP, databáze, mailové schránky apod.) a parametry (počet souborů, jejich maximální velikost, max. velikost uploadu atd.) které jsou pro danou stránku potřeba. Samozřejmě, že je tu i hledisko ceny, která se výrazně liší.

Jinou možností je využití tzv. freehostingových agentur (např. www.webzdarma.cz), které poskytují prostor na umístění stránek zdarma. Ale za cenu několika nevýhod :

na stránkách musí být umístěna reklama (obvykle v podobě banneru)

validace – i když se vytvoří validní stránka, tak není zaručeno, že bude validní i vložená reklama

Poslední možnost je využít služeb velkých portálů typu Seznam.cz, které poskytují prostor zdarma a bez reklamy. Podmínkou je, že u nich musí být vedena e-mailová schránka. Navíc je prostor omezený (u Seznamu.cz 1MB). Prostor se dá zvětšit, ale to je již za poplatek.

3. Grafická podoba stránky

3.1 Písmo

HTML umožňuje použít teoreticky libovolný font. Text na stránce se ale tímto fontem naformátuje jen v tom případě, že tento font má na svém počítači nainstalovaný i návštěvník. HTML (nebo CSS) umožňuje nadefinovat více fontů najednou. Následující fonty se použijí v tom případě, že font před nimi není v počítači návštěvníka nainstalovaný.

Podle společných znaků jsou fonty rozděleny do několika kategorií (rodin):

písma Serif – patková písma (Times Roman)

písma Sans Serif – bezpatková písma (Arial, Tahoma, Verdana)

písma Courier - neproporcionální písmo (Courier) aj.

Jedno ze základních pravidel pro práci s písmem mluví o minimalizaci počtu typů písma na stránce. Běžná webová stránka si musí vystačit s maximálně dvěma druhy písma (v případě odlišného písma pro nadpisy, třemi). Obvykle si ale na jedné stránce používá jeden font.

Někdy je třeba určitou část textu odlišit či zvýraznit. Zvýrazněné nápisy slouží jako opěrné body a usnadňují orientaci. Nedoporučuje se v textu měnit velikost písma. Vnáší se tím nepravidelnost do řádkové struktury a působí to rušivě. Také se nedoporučuje text podtrhávat. Podtržení je bráno jako znak ukazující na aktivní odkaz.

Nejčastěji se důležitý text zvýrazňuje změnou řezu písma – tučné písmo či kurziva (elementy `` nebo `` v jazyku HTML). Velmi výrazná je i změna barvy písma, nebo barvy podkladu pod písmem.

Je zde i možnost vložit text (zejména různé nadpisy) na stránku pomocí obrázku. Tím se zajistí, že se text zobrazí přesně tak jak má. Používání tohoto postupu není ale doporučováno organizací W3C.

3.2 Barvy

Aby se stránka líbila, musí na ní být všechny barvy navzájem vhodně sladěny. Bude-li na stránce barev málo, nebo mezi nimi budou malé rozdíly, bude stránka nezajímavá. Bude-li na stránce barev příliš, bude výsledkem chaos snižující vnímání obsahu stránky.

Existují tři principy jak kombinovat barvy:

1. **monochromatické schéma** – používá se bílá a černá barva v kombinaci s jednou další barvou (může se jednat o více variant téže barvy s různou sytostí a světlostí)
2. **analogické schéma** - používají se barvy, které spolu na barevném kruhu sousedí. Vybraná barva se doplní jejími sousedy zleva i zprava. Jedna barva je na stránce dominantní.
3. **komplementární schéma**– barvy stojící v kruhu přímo proti sobě. Při správném použití mohou vytvářet jak rovnováhu, tak kontrast.

Obrázek 3 – Kruh barev



čerpáno z: [18]

Většina zdrojů stále doporučuje používat na webových stránkách paletu 216 tzv. bezpečných barev.

Obrázek 4 – Paleta barev



čerpáno z: [18]

Jedná se o barvy, u nichž je zaručeno, že se zobrazí správně i při použití 8b barevné hloubky – 256 barev.

Studené barvy (modrá, zelená) se hodí na barvu pozadí, protože zdánlivě ustupují. Naopak teplé barvy (červená, žlutá) vystupují, takže se hodí na barvu textu.

3.3 Obrázky a fotografie

Použití grafiky na webových stránkách je dnes již jejich standardní součástí. Ale platí, že jakékoli obrázky (fotografie, loga, oddělovací prvky aj.) umístěné na stránce zpomalují její načtení. Čím více je na stránce obrázků, tím je stránka objemnější, a tím je její načtení pomalejší. Vzhledem k tomu, že cílem je vytvářet stránky, které se zobrazují co možná nejrychleji je třeba si dávat pozor na velikost vkládaných obrázků.

3.3.1 Velikost obrázků

Velikost obrázků je ovlivněna především dvěma faktory:

Rozlišením (rozměry) obrázku

Rozlišení se udává jako součin jeho šířky a výšky v pixelech (např. 200x300 px). Jeden pixel prvku zabere na obrazovce právě jeden pixel plochy. Z toho plyne i rozměrové omezení. Obrázek by neměl být větší než je rozlišení monitoru. Doporučuje se používat obrázky o rozměrech maximálně 600x600px.

V případě kdy je na stránce větší počet větších obrázků se vyplatí používat tzv. náhledy – zmenšené podoby obrázků umístěné na stránce, které odkazují na obrázek ve skutečné velikosti.

Barevnou hloubkou

Tj. počtem bitů potřebných pro uchování informace o barvě jednoho pixelu. Čím větší je, tím více barev se dokáže zobrazit:

8bitová hloubka – 256 barev

24 bitová hloubka - již přes 16 milionů barev.

Na druhou stranu, čím větší hloubka je, tím více místa pak obrázek zabírá, a tím pomalejší je načítání stránky (24 bitový obrázek je asi trojnásobně větší než 8 bitový o stejném rozlišení).

Zmenšení obrázku (a tedy jejich rychlejšího načtení) lze dosáhnout tzv. komprimací. Komprimace spočívá ve zmenšení počtu barev v obrázku - podobné barevné odstíny barev se slíjí do jedné barvy. Dojde tím k poškození kvality obrázku, ale správnou mírou komprimace se dá dosáhnou solidní kombinace přijatelné kvality a velikosti. Nelze komprimovat grafiku ve formátech GIF a PNG.

Dalším způsobem jak urychlit načítání je tzv. prokládání obrázků. Při načtení tohoto typu obrázku se nejprve zobrazí hrubý náhled, který se postupně vyhlazuje a zaostřuje (zatímco text je již čitelný). Použití prokládání je použitelné jen u velkých či velmi kvalitních (velký počet DPI) obrázků. Jinak je zbytečné.

3.3.2 Grafické formáty

Dalším důležitým rozhodnutím při umístění obrázku na stránku je výběr vhodného formátu. V současnosti jsou široce podporovány dva formáty:

GIF (Graphic Image Format)

Vzhledem k době svého vzniku (80. léta 20. stol) podporuje jen 256 barev (8 bitovou barevnou hloubku). Umožňuje vytvářet prokládané či transparentní (tj. s jednou barvou průhlednou) obrázky. Ve specifikaci GIF 89a také animované GIF, což jsou jednoduché animace. Umožňuje tzv. bezztrátovou kompresi, tzn. že zkomprimovaný a následně dekomprimovaný obrázek je shodný s původním originálem.

Tento formát se hodí pro jednoduché obrázky, kde je použito málo barev – ikony, loga a obrázky nakreslené v počítači. Umožňuje i snížit počet použitých barev (např. na černou a bílou – 2 barvy)

JPEG (Point Photographic Experts Group)

Na rozdíl od GIFu používá tzv. ztrátovou kompresi (= výsledný obrázek je méně kvalitní a nelze z něj dekompresí získat původní obrázek).

Tento formát není vhodný pro grafiku vytvořenou na počítači. Ale znamenitě se hodí pro uchovávání naskenovaných fotografií. Díky tomuto svému zaměření podporuje pouze 24 bitovou barevnou hloubku.

Tabulka 3 – Srovnání grafických formátů GIF a JPEG

	GIF	JPEG
Barevná hloubka	1 - 8	24
Max. počet barev	258	16 milionů
Podpora prokládaných obrázků	ano	ano
Transparentní obrázky	ano	ne
Animované obrázky	ano	ne
Komprese	bezztrátová	ztrátová

(čerpáno z [5]; s. 76)

Dalším možným formátem je formát PNG, který má nahradit formát GIF. Problémem je, že není příliš podporován (např. Internet Explorer jej zobrazí pouze do 24 bitové hloubky, i když je schopen hloubky až 48 bitové) ani rozšířen.

3.3.3 Animace na stránkách

Použití animací je dvousečné. Na jedné straně stránku oživuje a dodává jí na zajímavosti. Ale na druhé straně animace odvádějí návštěvníkovu pozornost od zobrazených informací a tím prodlužují dobu potřebnou ke vstřebání informací obsažených na stránce. Proto se doporučuje používat animace s rozmyslem a ne v příliš velkém rozsahu.

4. Zdrojový kód

Původně byl jazyk HTML zamýšlen jako jazyk pro popis struktury dokumentu. Veškeré stylování mělo být řízeno stylovým předpisem. Vývoj šel ale jiným směrem. Nejprve došlo k omezení stylování na minimum. Dalším krokem bylo zabudování formátovacích prostředků přímo do jazyka HTML.

Tento způsob řešení vedl k tomu, že se původní myšlenka HTML, jako jazyka pro popis struktury, vytratila. Jazyk HTML se stal nepřehledným, zbytečně objemným a těžko interpretovatelným. Tento směr vývoje se ukázal jako nevhodný.

Došlo tedy k oprášení myšlenky na stylový předpis a v roce 1996 vznikla Specifikace kaskádových stylů 1. Také norma HTML 4.0 zakázala používat množství formátovacích značek umístěných přímo v jazyku HTML. (mimo speciální normy HTML 4.0 Transitional). Zakázané značky jsou nahrazeny právě kaskádovými styly .

Další vývojový stupeň – XHTML (verze 1.0 a 1.1) – je v používání těchto značek ještě striktnější, ve verzi 1.1 již není používání těchto značek povoleno vůbec. Veškeré formátování je řízeno v kaskádových stylech..

Dochází tedy k návratu k původní myšlence oddělení struktury webového dokumentu (jazyk HTML) a formátovacích prvků (CSS styly).

Používání CSS stylů přináší, ve srovnání s formátovacími prvky umístěnými přímo v jazyku HTML, množství výhod:

- širší formátovací možnosti
- snadnější tvorbu a údržbu stránek
- oddělení struktur a formátování
- vyšší přístupnost stránek
- dynamickou práci se styly
- formátování XML dokumentů aj.

V závislosti na použité verzi normy HTML (XHTML) lze i kaskádové styly použít v různém rozsahu:

- pro obohacení formátovacích možností
- pro veškeré formátování mimo rozmístnění (layout) prvků
- pro veškeré formátování, včetně rozmístnění prvků.

S ohlednutím k doporučením organizace W3C by právě poslední varianta, společně s normou XHTML měla být tou jedinou alternativou.

4.1 Kaskádové styly

4.1.1 Definice kaskádových stylů

Způsob zápisu definice je jednoduchý. Například pro odstavec s velikostí písma 11 pixelů je následující:

```
p {font-size: 11px;}
```

Definice se skládá ze dvou částí – selektoru (v tomto případě p) a deklarace (font-size: 11px). Deklarace se skládá z vlastnosti, která se definuje (font-size) a hodnoty kterou má nabývat (11px).

4.1.2 Selektory

Tabulka 4 - Selektory

Vzorec	Název	Význam
*	Univerzální selektor	Váže se ke všem prvkům
E	Typový selektor	Váže se k libovolnému prvku E (tj. k prvku typu E)
E F	Selektor následníka	Váže se k libovolnému prvku F, který je následníkem prvku E
E > F	Selektor potomka	Váže se k libovolnému prvku F, který je potomkem (tj. přímým následníkem prvku E)
E:first-child	Pseudotřída :first-child	Váže se k prvku E, je-li E prvním potomkem svého rodiče
E:link	Pseudotřída :link	Váže se k prvku E, je-li E kotvou hypertextového odkazu, jehož cíl ještě nebyl navštíven
E:visited	Pseudotřída :visited	Váže se k prvku E, je-li kotvou hypertextového odkazu, jehož cíl již byl navštíven
E:active	Dynamické pseudotřída	Váže se k prvku E v aktivním stavu(tj. obvykle je-li vybrán myší či klávesnicí
E:hover	Dynamické pseudotřída	Váže se k prvku E v okamžiku, kdy je nad ním kurzor myši
E:focus	Dynamické pseudotřída	Váže se k prvku E (zpravidla prvek formuláře či odkaz) v okamžiku, kdy je na něm kurzor
E:lang(c)	Pseudotřída :lang()	Váže se k prvku typu E, je-li obsah prvku v jazyce (c). Na jazyku dokumentu závisí, jak se jazyk c určí
E + F	Sousední selektor	Váže se k libovolnému prvku F, která bezprostředně předchází prvek E
E[xyz]	Selektor atributu	Váže se k libovolnému prvku E, který má nastaven atribut xyz (libovolnou hodnotou)
E[xyz =“abc“]	Selektor atributu	Váže se k libovolnému prvku E, který má nastaven atribut xyz na hodnotu abc
E[xyz ~ = “abc“]	Selektor atributu	Váže se k libovolnému prvku E, který má nastaven atribut xyz na seznam hodnot oddělených mezerou a jedna z těchto hodnot je rovna abc
E[xyz =“abc“]	Selektor atributu	Váže se k libovolnému prvku E, jehož atribut xyz obsahuje seznam hodnot oddělených pomlčkami a jedna z těchto hodnot je rovna abc
E.abc	Selektor třídy	Váže se k libovolnému prvku E, jehož třída je abc.
E#id	Selektor ID	Váže se k libovolnému prvku E, jehož Id je id

V praxi se lze nejčastěji setkat s následujícími selektory:

Typový selektor

Tento typ selektoru ovlivňuje vlastnosti u již existujících prvků jazyka HTML (nadpisy h1 – h6, table, img, body apod.). Ovlivňuje všechny výskyty daného prvku na stránce. Zápis definice tohoto typu může být např. :

```
h1 {color: red}
```

Selektor následníka (kontextový selektor)

Tento typ selektoru se váže k elementům, které jsou umístěny uvnitř jiného elementu. Tyto vnořené elementy pak přebírají vlastnosti elementu který je obsahuje (dědičnost)

Např.

```
p.middle {font-size: 11px;
          color: black;
          text-decoration: underline;}
p.middle ul {font-weight: bold;}
```

První definice nastaví velikost a barvu písma, a to, že písmo bude podtržené. Druhá definice říká, že každý nesetříděný seznam obsažený v elementu `<p class="middle">` bude napsaný tučně. Ale zároveň si uchová černou barvu, velikost 11px a bude podtržený.

Selektor třídy

Způsob zápisu je podobný typovému selektoru, jen s tím rozdílem, že se přidá název třídy.

```
např.: p.tucne {font-weight: bold;}
```

Tato definice se již netýká všech elementů `<p>`, ale jen těch, kterým je přidělena třída `tucne`.

Ta se přiděluje pomocí atributu `CLASS`:

```
<p class="tucne"> ... text ... </p>
```

Je možné definovat i obecnou třídu, která není navázána na žádný element.

```
např.: .tucne {font-weight: bold;}
```

Takto nadefinovanou třídu je možné vložit do HTML dokumentu dvěma způsoby. Buď pomocí elementu `<div>` nebo elementu `` spolu s příslušnou třídou. Rozdíl mezi těmito dvěma elementy je v tom, že element `<div>` je elementem blokovým, kdežto element `` je elementem řádkovým (inline).

Jsou přípustné i tzv. vícenásobné třídy, kdy se jednotlivé třídy oddělují tečkou:

```
p.small.horni {color:red}
```

Selektor ID

Způsob zápisu i volání je téměř totožný se selektorem třídy. Také umožňuje složený zápis a také se dá volat pomocí elementů <div> a nebo pomocí zápisu

```
<p id="tucne"> ... text ... </p>
```

Na rozdíl od třídy se místo tečky používá doublehash (#)

```
př. # 102 {font-size: 11px;}
```

Hlavní rozdíl je ve způsobu použití. ID slouží k definici jedinečného jména elementu na jednom dokumentu – tzn. že na jedné webové stránce může být maximálně jeden element s id="102". Naproti tomu použití třídy není takto omezeno.

4.1.3 Pseudoprvky a pseudotřídy

Pseudoprvky ani pseudotřídy nejsou ve zdrojovém kódu explicitně uvedeny. Zapisují se přímo za selektor od kterého se oddělují dvojtečkou.

Pseudoprvky

Pseudoprvky rozšiřují možnosti formátování dokumentu o prvky, které v jazyku HTML neexistují. Problém je ale často s kompatibilitou, protože ne všechny prohlížeče pseudoprvky podporují.

:first-line – umožňuje formátování prvního řádku. Používá se u blokových elementů a s omezeným počtem vlastností.

:first-letter – ovlivňuje formátování prvního písmene (iniciály) elementu. Opět jen s omezeným počtem vlastností.

:before, :after - umožňují vložit generovaný text před (za) běžný obsah elementu.

Pseudotřídy

Rozlišují prvky dle jiných charakteristik než jsou jména, atributy či jejich obsah.

:link, :visited – umožňují odlišit navštívené a nenavštívené odkazy

:hover, :active, :focus – tyto pseudotřídy mění vzhled některých prvků dle přání uživatele

:hover – změny vyvolané umístěním kurzoru na odkaz, ale bez jeho aktivace (neklikne se na něj)

:active – působí na prvek v době mezi stiskem a uvolněním tlačítka

:focus – působí na element, kdy je připraven reagovat na události z klávesnice

Pseudotřídy :lang a :first-child jsou zatím podporovány jen v několika málo prohlížečích. Takže nejsou příliš použitelné.

4.1.4 Slučování stylů

Definice stylů se dají slučovat.. Je možné pro jeden selektor nastavit více deklarácí najednou, nebo použít jednu definici pro více selektorů.

1. přiřazení více deklarácí jednomu selektoru

```
p {font-size: 11px;
  color: #ffffff;
  font-family: verdana;}
```

2. přiřazení deklarace více selektorům

```
h1, h2, h3 {color: white;
            font-family:arial;}
```

4.1.5 Vložení stylu do stránky

Způsobu, jak připojit kaskádové styly na HTML dokument je několik:

Přímé vložení stylu do hlavičky stránky pomocí elementu <style >

```
<style type="text/css">
  - jednotlivé definice-
</style>
```

Takto nadefinované styly se používají jen na té stránce, v jejíž hlavičce jsou napsány. Pokud se mají použít na více stránkách, je třeba je napsat (či rozkopírovat) na každou jednotlivou stránku. Tento způsob se často využívá v průběhu vývoje stránky, kdy se testuje výsledný vzhled (po skončení testování se obvykle přepíše do externího souboru).

Externí soubor se styly

Tento způsob spočívá ve využití externího souboru (s koncovkou .css), který neobsahuje nic jiného než deklarace použitých stylů. Takto vytvořené styly se dají připojit k jakémukoliv HTML souboru. Připojují se přes element `<link>`.

```
<link rel="stylesheet" type="text/css" href="soubor.css" />
```

Import stylů

Způsob velmi podobný způsobu předchozímu. CSS styly jsou opět uloženy externím souboru. Rozdíl je ve způsobu volání externího souboru, kdy se místo elementu `<link>` používá zápis:

```
<style type="text/css">
    @import "soubor.css"
</style>
```

Rozdíl vůči předchozímu způsobu je v tom, že tento způsob není podporován staršími prohlížeči. Ty takto připojené styly vůbec nezaregistrují

Přímý zápis do zdrojového kódu HTML dokumentu

Styly se dají vepsat přímo do kódu stránky pomocí atributu `style`. Např. `<p style="color:red">` či `<h2 style="color:red">`. Dá se volat i pomocí tříd nebo elementů `<div>` a ``.

Není vhodné tento styl příliš používat. Popírá totiž myšlenku kaskádových stylů, oddělení vzhledu od obsahu a vrací se k původní podobě obsáhlého kódu.

4.1.6 Pravidla pro používání stylů

Platí, že styly nadefinované v elementu `<style>` překryjí styly nadefinované v externím souboru. A styly nadefinované atributem `style` překryjí všechny ostatní.

Pokud se na jeden dokument používá více stylů, skládají se tyto styly podle určitých (a celkem složitých) pravidel. Obecně platí, že vyšší váhu mají později uvedené styly. V případě, že je nutné určit určitou deklaraci za důležitou, dá se využít zápis se slovem `!important`.

```
př. h6 { font-size: 8px !important; }
```

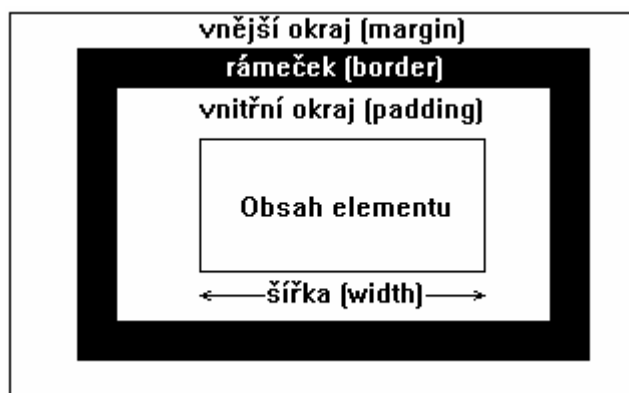

4.2 Rozvržení stránky (layout stránky)

4.2.1 Pomocí kaskádových stylů

Jedním z již uvedených způsobů se vytvoří potřebný styl. Základní vlastnosti, které by měl mít nadefinované jsou výška (height) a šířka (width), které určují jak velký prostor, bude element zabírat. Pokud nebudou nadefinovány, přizpůsobí se velikost prostoru tomu co element obsahuje. Nadefinovaný styl se následně volá pomocí elementu <div>.

Ke správnému pochopení vlastností, které mají dopad na formátování stránky stačí nahlédnout na následující obrázek.

Obrázek 5 – Boxová koncepce



Každý element je tvořen jedním nebo několika obdélníkovými boxy, které se skládají ze samotného obsahu elementu, a dále mohou obsahovat vnitřní okraj (vlastnost padding), rámeček (border) a venkovní okraje (margin).

Pro vytváření layoutu stránky pomocí kaskádových stylů existuje několik způsobů, které se dají kombinovat..

Využití elementu <div>

Někdy stačí využít jedné vlastnosti elementů <div>. Při použití několika po sobě jdoucích elementů <div>, dojde k tomu, že se elementy jakoby „přilepí“ pod sebe. Je to dáno tím, že <div> je blokovým elementem, a tedy v případě, kdy nemá nadefinovanou šířku, zabírá celou šířku nadřazeného elementu. Už tato vlastnost je využitelná např. při vytváření navigačních menu.

Určení pozice

V tomto případě se nadefinují přesně rozměry (výška a šířka) boxu a pomocí vlastností `top` a `left` se určí pozice levého horního rohu. To je výchozí bod, od kterého se počítá výška a šířka elementu. Vlastností `position` se určuje způsob umístění daného elementu – absolutně (tj. přesně zadaná pozice) či relativně (tj. o kolik se posune vůči své umístění).

Plovoucí objekty

Pomocí vlastnosti `float` lze zajistit, že se jednotlivé elementy `<div>` nebudou řadit pod sebe, ale vedle sebe. `Float` může dosahovat dvou hodnot – `left` a `right`. Pokud bude mít několik za sebou jdoucích `divů` nastavenou vlastnost `float` na `left`, dojde k tomu že se přilepí vedle sebe zleva (v případě `float:right` – zprava).

4.2.2 Využití tabulek

Myšlenka je jednoduchá. Celá stránka se seveře do elementu `<table>`. Pro umístění prvku na stránku (text, obrázek) pak stačí vyplnit jen příslušnou buňku. V případě složitějšího designu se používají vnořené tabulky, což jsou tabulky vložené do jedné buňky jiné tabulky (takto to může jít i přes několik úrovní).

Je pravda, že při využití tabulek lze dosáhnout velmi přesného rozvržení stránky, ale za cenu řady nevýhod.

při použití např. textového prohlížeče či čtecího zařízení neuvidí uživatel nic jiného než nesmyslnou změť slov, vět, obrázků apod.

aktualizace takto vytvořených stránek je velmi obtížná a případné změny velikostí jednotlivých prvků na stránce je téměř nemožné bez přepracování celé stránky.

takto vytvořené stránky jsou špatně indexovány vyhledávacími roboty.

zbytečně rozsáhlý kód (objemově), kdy formátovací značky tvoří někdy i více než polovinu objemu kódu a kód samotný je velmi nepřehledný

4.2.3 Využití rámu (frames)

Rámy vlastně nejsou nic jiného než boxy, do kterých se vkládají webové stránky. Rámů může být na stránce libovolný počet, ale nedoporučuje se používat více než 3 až 4 rámy na jednu stránku.

Výhoda použití rámu je jasná - snadné rozvržení obsahu na stránce. Nad ní ale převládají nevýhody:

- nemožnost se zvenčí odkazovat na stránky uvnitř webu tvořeného rámy

- rámy nejsou podporovány všemi typy přehrávacích zařízení

- takto vytvořené stránky jsou velmi špatně indexovány pomocí vyhledávacích strojů.

Používat tabulky či rámy pro layout stránek je dnes přežitek a ani jeden z nich by se již neměl používat. Dle doporučení organizace W3C je jediným správným způsobem pro vytváření layoutu stránky používat výhradně kaskádové styly.

4.3 Rozdílné prohlížeče

To že se stránka zobrazí správně v Internet Exploreru neznamena, že se zobrazí správně i v prohlížečích ostatních. Je sice pravda, tak by tomu tak být mělo, ale bohužel tomu tak není. Implementace kaskádových stylů je u různých prohlížečů různá, popřípadě jsou styly naimplementovány s chybami.

Vyplatí se vytvořit stránku tak, aby se zobrazovala správně co největšímu počtu návštěvníků. Ideálním způsobem by bylo odladit stránku na všechny prohlížeče. To ale v reálu není příliš uskutečnitelné. Proto by měla být správně zobrazena alespoň na nejvíce používaných prohlížečích – což jsou v současnosti Internet Explorer (asi 80% uživatelů) a Mozilla Firefox (15%). Popřípadě ještě na prohlížeč Opera.

Na co si dát pozor při ladění stránek pro IE a Firefox?

Šířka objektů

Jedním z problémů spojených s vytvářením layoutu pomocí CSS je výpočet šířky. Internet Explorer (IE) a Firefox (tedy dva nejrozšířenější prohlížeče) jí vypočítávají každý jinak.

Např. Je nadefinovaná šířka elementu na 200px. K tomu vnitřní okraj 10px, rámeček 5px, a vnější okraj 20px.

V tomto případě dospěje Firefox k tomu, že celková šířka je 230px. Což je správně - obsah, plus vnitřní okraj a rámeček. Ale u IE bude celková výsledná šířka stále jen 200px! Vnitřní okraj a rámeček se odčítají od celkové šířky.

Řešení tohoto problému je složitější. Všude kde je to jen možné je vhodné používat vlastnost margin místo vlastnosti padding, a u těch elementů, které okraje mít nemají se vyplatí okraje vynulovat (nastavit jim margin i padding roven nule)

Definování velikosti písma.

Problém je, pokud se v CSS použije hodnota typu large, small, big atd. Každý prohlížeč má nastavenou pro danou hodnotu jinou velikost písma. Řešení je v tomto případě jednoduché. Stačí přestat používat tyto hodnoty a místo nich deklarovat velikost písma napevno, pomocí pixelů.

Barva pozadí elementu

Pokud má mít objekt barvu na pozadí, je třeba definovat jeho výšku. Jinak se v IE barva nezobrazí, resp. se podbarví pouze vyplněný obsah a ne celý element.

Mezery u elementů

Rozdíl v mezerách před a za blokovými elementy. Opět každý z prohlížečů má nastavenou jinou velikost. Řešení je stejné jako v bodě jedna - vynulovat hodnoty margin a padding či jim deklarovat pevnou velikost v pixelech.

Nadpisy h1, h2, h3 až... h6

Jak IE tak i Firefox mají pro ně nastavenou jiné vlastnosti písma (velikost, tučnost, průřez aj.) a mezery před a za elementem. Opět platí že je vhodné si tyto vlastnosti nadefinovat ve stylech napevno. Nebo je vynulovat.

4.4 Zdrojový kód v HTML souboru.

Nechci se zde podrobně rozepisovat o tom jak má vypadat zdrojový kód HTML souboru nebo jaké tagy se dají v HTML používat. Rád bych zmínil jen několik bodů, které si zaslouží pozornost.

Nadpisy

Pro nadpisy je výhodné používat k tomu určené elementy <h1> – <h6>. Jednak mají předdefinované vlastnosti (velikost písma, řez písma, podržení apod.). A jednak zejména elementy <h1> a <h2> jsou brány vyhledávacími stroji jako jeden z hlavních aspektů při indexaci stránek.

Pokud zvolená úroveň nadpisu vzhledově nevyhovuje potřebám, není problém si jí přizpůsobit pomocí kaskádových stylů.

Zvýraznění části textu

V případě, že je třeba zdůraznit určitou část textu na stránce, tak je výhodné použít jeden z předdefinovaných elementů či , které nahradili elementy typu a <i>. Důvod je jednoduchý. Zatímco tyto značky jen zvýrazňovaly části textu, tak značky a dávají na zvýrazněný text i důraz.

„Prvky frází dodávají částem textu informace o struktuře. Obvyklý význam prvků frází je následující: značí důraz, značí větší důraz.“

(citováno z [2]; s. 98)

Při použití elementu se text zobrazí napsaný kurzívou, a v případě elementu bude text zobrazen tučně. Opět platí, že pokud tento vzhled nevyhovuje, není problém jej změnit pomocí kaskádových stylů.

Obrázky

U všech obrázků na stránce je třeba uvést atribut ALT. Dle specifikace CSS je atributem povinným. Jeho obsah by měl co nejlépe vyjadřovat obsah a smysl obrázku. Používá se jako zástupný text za obrázek v případě, kdy návštěvník používá textového prohlížeče, nebo pokud má vypnuté zobrazování obrázků. Stejnou funkci zastává atribut TITLE pro prohlížeč Firefox.

Všechny obrázky by měly mít uvedeny i rozměry a to pomocí atributů HEIGHT a WIDTH (musí odpovídat skutečným velikostem, nesmí se použít na změnu rozměrů). Pokud jsou uvedeny, tak si prohlížeč dokáže vyhradit pro obrázky nedefinovaný prostor. Stránka potom při načítání „neposkakuje“ vlivem načítání dalších a dalších obrázků.

Metadata

Neboli informace o informacích, která se nezobrazují, nýbrž slouží k popisu stránky. Na stránce nejsou metadata povinné, ale je vhodné je používat. Metadata je velký počet, ale naprosto dostatečné je používat metadata sepsaná v Dublin Core. Což je

„soubor metadataových prvků, jehož záměrem je usnadnit vyhledávání elektronických zdrojů“
(citováno z [19])

V současnosti Standard Dublin Core obsahuje 16 prvků.

Tabulka 5 – Standard Dublin Core

Title	<i>Název</i>	Subject	<i>Předmět</i>
Creator	<i>Tvůrce</i>	Description	<i>Popis</i>
Publisher	<i>Vydavatel</i>	Contributor	<i>Příspěvatel</i>
Date	<i>Datum</i>	Type	<i>Typ</i>
Format	<i>Format</i>	Identifier	<i>Identifikátor</i>
Source	<i>Zdroj</i>	Language	<i>Jazyk</i>
Relation	<i>Vztah</i>	Coverage	<i>Pokrytí</i>
Rights	<i>Práva</i>	Audience	<i>Publikum</i>

(čerpáno z [20])

Čitelný a dobře strukturovaný kód

Rád bych zdůraznil, že je důležité psát zdrojový kód tak, aby se v něm vyznal i někdo jiný než autor. To platí jak pro kód v HTML souboru tak i v kaskádových stylech. Vhodné je kód rozčlenit do určitých logických bloků a ty od sebe oddělit, třeba mezerou (která se ve výsledku nezobrazí) a jednotliví části uvnitř bloku oddělit odsazením.

Doporučuje se i využití komentářů (opět se nezobrazují) pro označení začátku i konce logického bloku. Či pro vepsání poznámky pro další programátory.

Pokud se začne kód nějakým způsobem členit a odsazovat, tak by zvolený způsob měl být použit v celém souboru.

4.5 *Validace stránek*

K ověření, zda je vytvořená stránka validní (tj. že vyhovuje standardům) stačí navštívit webové stránky konsorcia W3C, konkrétně jejich online webový validátor na adrese <http://validator.w3.org/>.

Nejenže dokáže zjistit, zda je stránka validní či nikoliv, ale v případě odhalení nedostatků chyby i vypíše. Občas je ale trochu problém zjistit, co vlastně chybu vyvolalo.

Často bývají validátory i součástí běžně používaných editorů pro vytváření webových aplikací (PSPad, HTML-Kit aj.).

5. Optimalizace stránek

Vzhledem k tomu, že existuje něco kolem miliardy webových stránek, je důležité nejen stránky vytvořit a umístit na web, ale také zajistit, aby lidé o stránkách věděli a navštěvovali je. K tomu slouží právě optimalizace stránek. Jedná se sadu nástrojů, které mají zajistit, aby byla návštěvnost stránek co nejvyšší.

5.1 SEM (Search Engine Marketing)

Nejedná se ani tak o optimalizaci, jako spíš o využití reklamních a prezentačních možností, které internet nabízí:

Tématické katalogy a seznamy

Umístění stránky do katalogů či seznamů velkých portálů typu Seznam, Centrum, Atlas aj.

Reklamní bannery

Využití reklamních bannerů (reklamních proužků) – jedná se zpravidla o obdélníkový obrázek, animaci, nebo interaktivní grafiku umístěnou na webových stránkách. Bannery, i přes jejich malou účinnosti, tvoří na internetu stále jednu z nejčastějších forem reklamy .

Odkazy typu pay-per-click (platba za klik)

Oproti klasickým bannerům je zde výhoda v tom, že inzerent platí až ve chvíli, kdy si na odkaz potenciální klient klikl. To znamená, že inzerent ví přesně kolik a kdy platí.

5.2 SEO (Search Engine Optimization)

V tomto případě jde o optimalizaci webových stránek pro fulltextové vyhledávače tak, aby ji zobrazovali na předních pozicích při zadání určitých klíčových slov.

Cílem je naoptimalizovat stránku tak, aby se umístila ve výsledcích na první stránce, tj. mezi 1 – 10 místem na zvolená klíčová slova. Nižší umístění je bez efektu, protože lidé obvykle prolétnou očima jen úvodní stránku a pak místo dalšího hledání raději zadají nová vyhledávací slova.

„V oblasti SEO neexistuje a nikdy nebude existovat dogma, které by zaručovalo lepší pozice ve vyhledávačích. Nikdo nezná přesná pravidla jak vyhledávače řadí odkazy. Všechny znalosti, které se dnes dají považovat na platné a ověřené, jsou založené na vlastním zkoušení, pozorování a odhadech. Fulltextové vyhledávače tyto pravidla neposkytují.“

(citováno [16])

Je to tak. Principy a pravidla podle kterých jednotlivé vyhledávací stroje stránky indexují je tajný. Nikdo, mimo samotných tvůrců a majitelů, neví jak přesně pracují. Přesto existuje několik praxí ověřených kroků, o kterých se dá předpokládat, že na ně vyhledávací roboti při indexaci stránek dávají určitý důraz:

Validní stránka

První a nejdůležitější předpoklad. Stránka může být naoptimalizována sebelépe, ale pokud není validní, tak ji roboti zaindexují hůře.

Správná struktura URL adres

Jako nejlépe indexovatelné se jeví stránky jejíž adresa je v podobě `http://www.doména/soubor.html`. V adrese by se neměli vyskytovat žádná `id = určitá hodnota` apod. Při převodu na požadovanou podobu se dá použít modul serveru Apache `Mod_rewrite`, který umožňuje vytvořit téměř libovolnou podobu adresy.

Použití vhodných klíčových slov

Klíčová slova, na které se stránky optimalizují se uvádějí v metadatech `keywords`, `description`, `title`. Dále je vhodné je umístit do nadpisů, zejména úrovně `<h1>` a `<h2>`, popřípadě do zvýrazněných částí textu pomocí elementů `` a ``.

Zpětné odkazy na stránku

Je vhodné stránky provázat odkazy nejen v menu, ale i v obsahu stránky. Pokud jsou jednotlivé stránky vhodně pojmenovány klíčovými slovy odpovídajícími obsahu stránky, tak se zvýší počet výskytů těchto slov, a tím se zvýší šance na jejich lepší zaindexování.

6. Vytváření stránek Prague Golf Tours

Na těchto stránkách jsem spolupracoval v průběhu své školní praxe. Mým úkolem bylo převést grafický návrh do HTML podoby a vytvoření všech statických sekcí. Na stránkách se mnou dále spolupracovali grafik (grafický návrh, výběr vhodných fotek apod.) a programátor (dynamicky generová část stránek, komunikace s databází) .

Jak již napovídá název, tyto stránky si nechala vyrobit společnost pořádající zájezdy golfistů, zejména z Británie, na česká golfová hřiště (Karlštejn, Konopiště, Poděbrady aj.). Stránky byly vytvořeny kompletně v angličtině. Jsou dostupné z adresy: www.praguegolftours.cz.

Obrázek 6 – Náhled stránky www.praguegolftours.com



6.1 Funkční model stránek

6.1.1 Struktura stránek

Strukturu stránek jsme vytvořili po dohodě se zadavatelem. Vzhledem k tomu, že stránky nejsou nijak rozsáhlé, je i struktura stránek celkem jednoduchá. Zvolili jsme globální strukturu, kdy jsou odkazy na všechny sekce přístupné z úvodní strany. Několik sekcí je navíc provázáno odkazy umístěnými v textu stránek.

Většina sekcí má pouze jednu úroveň. Pouze sekce Kurzy (Courses) a Speciální nabídky (Special and fixed price offers) jsou víceúrovňové. V první úrovni obsahují seznam kurzů (či speciálních nabídek) a v druhé pak detailní informace o jednotlivých kurzech (speciálních nabídkách). Každá speciální nabídka ještě navíc obsahuje odkaz na rezervační formulář.

6.1.2 Navigační systémy

Jako vhodný navigační systém jsme zvolili systém vertikální. Odkazy na sekce a konkrétní stránky jednotlivých hřišť jsou umístěny pod sebou v levé části stránek.

Rozhodli jsme se pro toto řešení z více důvodů. Prvním bylo to, že počet odkazů je značný a v případě horizontálního systému by menu působilo přeplněně a nepřehledně. Druhým důvodem bylo i to, že nebylo jisté kolik odkazů vlastně bude (v průběhu práce se jejich počet stále měnil).

6.1.3 Rozvržení stránek

Použita je zde vícesloupcová struktura. Konkrétně dvousloupcová. Na každé stránce je vlevo umístěno navigační menu. V pravém pak obsah jednotlivých sekcí.

Levé navigační menu a horní část obsahového sloupce (horní obrázek s flashem a zelený pruh s mottem) jsou společné pro všechny sekce. Nijak se nemění – ani vzhledově, ani obsahově.

Layout stránek jsem vytvořil pomocí kaskádových stylů umístěných v externím souboru a volaných pomocí elementu <link>.

U rozměrů stránek byla explicitně stanovena pouze šířka a to na 990 px. To z toho důvodu, aby se stránka vešla celá na obrazovku monitoru při jeho rozlišení 1024x768px. Délka stránek explicitně zadána nebyla. Přizpůsobuje se délce obsahu stránky.

6.2 Vytváření stránek

První co bylo třeba udělat bylo „zalomit“ (tj. převést grafický návrh do HTML podoby) úvodní stránku. K tomu jsem dostal návrh stránky ve formátu PNG, podklady od zákazníka a několik informací od grafika:

- font písma, jeho velikost a barva, možné způsoby zvýraznění části textu
- který text bude vložen obrázkem (kvůli použití neobvyklého fontu) a který v HTML
- obrázky s požadovanými přechody barev
- informace o grafické podobě odkazů aj.

Použitím programu Macromedia Fireworks jsem si z grafického návrh vyexportoval potřebné obrázky ve správném rozlišení a potřebné kvalitě, oddělovací grafické prvky a získal kódy použitých barev.

6.2.1 Element `<body>`

Nejprve jsem u elementu `<body>` nastavil všechny vlastnosti, které jsou společné pro všechny stránky. V tomto případě se jednalo jen o barvu pozadí. Zápis v CSS stylu je následující:

```
body {background-color: #ffffff;
      background-image: url(images/pozadi.jpg);
      background-repeat: repeat-x;
      padding: 0;
      margin: 0;}
```

Pozadí stránek je tvořeno 1px obrázkem vytvořeným tak, aby obsahoval barvy v daném pořadí. Horní část obrázku (173px) tvoří přechod od tmavě modré barvy do světle modré. Jde o pozadí pro horní obrázek s golfistou. Dále následuje část obsahující zelenou barvu (52px) tvořící oddělovací zelený pruh. Ve spodní části je přechod ze zelené barvy do bíle. Ten tvoří pozadí jednotlivých sekcí.

Obrázek jsem pomocí vlastnosti `background-repeat` s hodnotou `repeat-x` roztáhl na celou šířku stránky. Protože přechod má jen omezenou délku a nebylo ještě jasné jak budou dlouhé jednotlivé sekce, tak jsem zároveň nastavil barvu pozadí na bílou. Tím se zajistí, že stránka bude mít zvolené pozadí, i když přesáhne délku přechodu.

Zároveň jsem vynuloval vlastnosti margin a padding (vnitřní a vnější okraj). To z toho důvodu, že v případě nevynulování Firefox zobrazí stránku zleva a svrchu odsazenou o několik pixelů.

Dnes bych ještě do definice přidal deklarace týkající se písma (font, barvu a velikost). Tedy vlastnosti společné pro veškeré texty na stránce. V tomto případě jsem tak neučinil a musel je definovat v nižších úrovních vícekrát. Přidělal jsem si tím jen práci, a zvýšil objem kódu.

6.2.2 *Layout úvodní stránky*

První způsob, kterým jsem řešit layout stránek bylo absolutní pozicování v CSS stylech. Výsledkem byla sice vyhovující podoba stránky, ale za cenu zbytečně rozsáhlého zdrojového kódu. Stačí se podívat na levé menu. To obsahuje dvanáct položek. A pro každou z nich jsem zvlášť vytvořil absolutně určenou pozici, jejíž zápis byl následující:

```
.jedna{position: absolute;  
    left: 0px;  
    top: 27px;  
    width: 226px;  
    height: 27px;}
```

Výsledkem bylo, že jsem tento způsob opustil a přešel na, v tomto okamžiku, výhodnější způsob. Ten spočíval ve využití vlastnosti float. Obsah stránky jsem rozdělil do tří samostatných sekcí:

```
navigační menu  
hlavička  
střed
```

Navigační menu

Obrázek 7 – Navigační menu



Celé navigační menu je sevřeno do jednoho elementu `<div class="lm">`, kterému jsem nastavil šířku, výšku a barvu pozadí (jeden z odstínů zelené). Dále jsem využil vlastnost `float` s hodnotou `left`, aby se menu zarovnal do levého horního rohu.

Horní část menu (Welcome to Prague) je tvořena obrázkem vložením na stránku elementem ``. Zbytek textu (... golf tours) je součástí velkého obrázku s golfistou. Obrázek zároveň slouží jako odkaz na úvodní stránku. Po kliknutí na něj se na ní dá takto dostat z jakékoliv podstránky. Je nepsanou dohodou, že logo firmy či jiný objekt (obvykle právě v levém horním rohu) je využíván právě k tomuto účelu.

Uvnitř elementu `lm` jsem umístil jednotlivé položky menu. Pro ně jsem vytvořil následující kód:

```
.lm_polozka {width: 226px;
             height: 22px;
             background-image: url(images/pozadi_menu_buton.jpg);
             background-repeat: repeat-x;
             font-family: Tahoma;
             color: white;
             font-size: 12px;
             font-weight: bold;
             vertical-align: middle;
             padding-top: 5px; }
```

Položce jsem nastavil pouze výšku a šířku (šířka je shodná s šířkou elementu `lm`). Nic víc pro umístění nebylo třeba. `padding-top` způsobil odsunutí obsahu položky (textu v položce) o 5px od horního okraje.

Položky jsem volal pomocí elementu `<div>`. Využil jsem zde toho, že `<div>` je blokový element. Tzn. že zabírá celou šířku nadřazeného elementu, v tomto případě elementu `<div class="lm">`. To způsobilo, že se jednotlivé položky menu umístili přímo pod sebe, nevznikla mezi nimi žádná mezera, a nebylo třeba jejich umístění jinak korigovat.

Barva pozadí byla vytvořena opět přechodem. Opět měl šířku 1px a výškou odpovídal výšce položky (27px). Vlastností `background-repeat` s hodnotou `repeat-x` jsem zajistil, že se roztáhne přes celou šířku položky.

Dále jsem položce nastavil vlastnosti pro písmo

- font Tahoma,
- velikost písma na 12px,
- bílou barvu
- tučnost.

Obrázek 8 – Položka navigačního menu



Takto vytvořená deklarace se dala jednoduše použít pro všech dvanáct položek menu, bez jakýchkoliv dalších úprav. Jednoduše jsem ji dvanáctkrát volal pomocí elementu `<div class="lm_polozka">`.

Aktivní sekce

Ta sekce, která je aktivní je označena golfovým míčkem (v levé části každé položky). To, že se zobrazuje tam kde má, je zajištěno pomocí skriptu napsaného v PHP, který vytvořil programátor.

Pro jeho zobrazení bylo třeba doplnit kód CSS stylu o následující položky :

```
.lm_polozka a { text-decoration: none;
                color: #FFFFFF;
                padding-left: 25px;
                display: block; }
.lm_polozka a:hover { text-decoration: underline;}
```

Tento kontextový selektor říká, že veškeré odkazy umístěné v elementu `lm_polozka` budou psány bíle, nebudou podtrženy a posunou se o 25px doprava. To proto, aby se udělalo místo pro golfový míček. Po přejetí kurzoru přes odkaz se navíc pro zvýraznění podtrhne (to zajišťuje druhá deklarace)

Poslední součástí menu je spodní část obsahující odkazy na jednotlivá golfová hřiště. Pro ně jsem vytvořil následující deklarace:

```
.leve_spodni { width: 201px;
               background-color:#315E25 ;
               color:white;
               font-size: 11px;
               font-weight: bold;
               padding-left: 20px;
               line-height: 2.2;
               font-family: Tahoma;}

.vlajka{background-image: url(images/ico_jamka.gif);
        background-repeat: no-repeat;
        margin-left: 5px;
        padding-left: 11px;}
```

V první deklaraci mimo vlastností ovlivňujících text, barvy pozadí a šířky je také určena výška řádku pomocí vlastnosti `line-height`. Udává se jako násobek velikosti použitého písma. Dále pomocí `padding-left` jsem odsunul text o 20 px od levého okraje.

Druhou definici jsem musel přidat později. Původně byly odkazy na hřiště umístěny v netříděném seznamu, bez odrážek. Na přání klienta se jako odrážka přidala vlaječka. Protože jsem si neuvědomil, že v CSS stylech existuje vlastnost `list-style-image`, která umožňuje přidat libovolný obrázek jako odrážku k seznamu, použil jsem tento způsob, kdy je obrázek vložen na pozadí daného prvku, a obsah je posunut o 11px od levého okraje. Tím jsem dosáhl také požadovaného efektu, i když složitějším způsobem.

Pomocí kontextového selektorů `.leve_spodni` a `.leve_spodni a: hover` jsem nastavil vlastnosti pro odkazy v této sekci – po přejetí kurzoru se odkaz podtrhne, jinak zůstane neměnný.

Hlavička

Obrázek 9 – Podoba hlavičky



Celá hlavička je sevřena do jednoho elementu `<div>`, kterému jsem přímo v HTML souboru pomocí atributu `style` nastavil tyto vlastnosti:

```
<div style="width: 774px; overflow: hidden; margin: 0px; float: left;">
```

Šířka je jasná. Vlastnost `float` s hodnotou `left` jsem použil proto, aby se tento element zarovnal zleva k levému menu. `Overflow: hidden` způsobí v případě, kdy obsah elementu přesahuje jeho velikost, oříznutí této přesahující části. Je to zde spíše jako pojistka pro případné budoucí změny.

Tento element obsahuje další dva elementy `<div>`, jejichž vlastnosti jsem již nastavoval v CSS souboru. První má tuto deklaraci:

```
.hlavicka {height: 173px;
width: 100%;
background-color: #1462AC;
background-image: URL(images/image_2.jpg);
float:left;
margin: 0px;
padding: 0px; }
```

Tento element má pevně danou výšku (dle velikosti obrázku), šířka je nastavena na 100%, tzn. že přejímá šířku nadřazeného elementu. Opět jsem vynulovat jak vnější, tak vnitřní okraj a zarovnal element vlevo. Obrázek je zde vložen na pozadí elementu vlastností background-image.

Obsahem tohoto elementu je jen flashové motto umístěné nalevo od golfisty. Samotný flashový soubor (swf) se na stránku umísťuje elementem <object>. Tomuto objektu jsem nastavil odsazení shora pomocí margin-top.

Druhý element <div> má tuto deklaraci:

```
.hlavicka2 {width: 749px;
height: 52px;
padding-left: 15px;
background-color: #315E25;
color: white;
font-family: Tahoma;
font-size: 12px;
float:left;
overflow: hidden;}
```

Opět má nastavenou výšku, šířku a zarovnání vlevo. Pozadí není tvořeno obrázkem, ale odstínem zelené barvy. Levý vnitřní okraj je nastaven na 15px a vlastnost overflow zajišťuje ochranu proti přetečení obsahu přes zadanou velikost elementu.

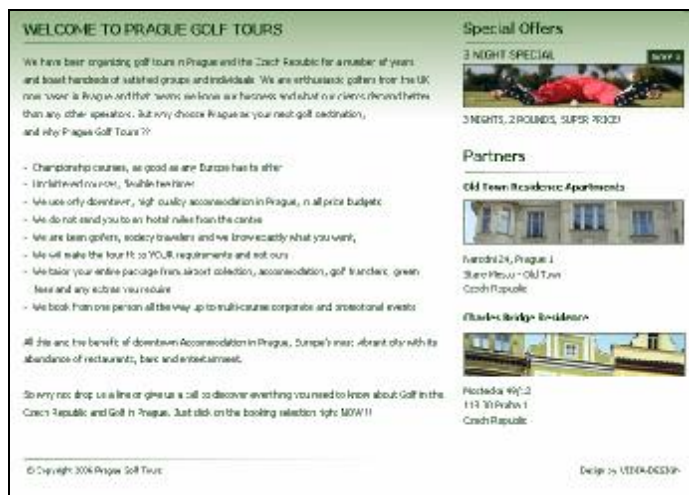
Součástí tohoto elementu je vpravo umístěné logo magazínu Today's Golfers. Jedná se o obrázek ve formátu gif, který je do kódu vložen elementem ``. V třídě je nadefinováno zarovnání vpravo a šířka obrázku.

Text nalevo od loga je vložen pomocí samostatného elementu `<div class="hl_text">` (v tomto případě by bylo možná výhodnější použít kontextový selektor). Opět jsem u něj musel nadefinovat vlastnosti pro text (velikost, font aj.). Umístění tohoto textu jsem vytvořil kombinací relativního pozicování, kdy jsem obrázek posunul o 32px níž, a vlastnosti float, která ho zarovnala vpravo, těsně k logu. Pro správné zobrazení má nastavenou i šířku.

Poslední součástí tohoto elementu je text v jeho levé části. Atributem STYLE jsem mu nastavil odsazení shora (`margin-top`) tak aby byl umístěn uprostřed (vertikálně). Zároveň je zarovnán vlevo vlastnosti float nastavenou na hodnotu left. Odsazení zleva o 15px je způsobeno dědičností (je nastaveno u nadřazeného elementu `<div="hlavicka2">`).

Střední část

Obrázek 10 – Obsah úvodní strany



Střední blok je na úvodní stránce tvořen dvěma samostatnými elementy `<div>`.

Střední sloupec

Element `<div class="stred">`, slouží pro vložení střední textové části. Nadpis je vložen elementem `<h1>`, který jsem pomocí CSS stylu upravil:

```
h1 {font-family: Tahoma;
    font-size: 16px;
    color:#003300;
    margin-bottom: 10px;
    margin-top: 0px; }
```

Nadpis je navíc napsán velkými písmeny (to jsem nadefinoval ve třídě `.velka` pomocí deklarace `text-transform: uppercase`). Pod nadpisem je jako oddělovací znak linka. Ta je vložena elementem ``.

Vlastnosti textu jsou nadefinovány v třídě `text`:

```
.text {font-family: Tahoma;
    font-size: 11px;
    color:#003300;
    line-height: 1.8;
    text-align: justify; }
```

Jednotlivé odrážky v textu jsem původně vyřešil pomocí neřazeného seznamu. Pak si ale zákazník přál je změnit na pomlčky. Proto jsem seznam smazal a nahradil ho řádkováním pomocí elementu `
` a pomlčkami umístěnými přímo v textu.

Nyní bych to řešil jinak. Nechal bych seznam seznamem, nastavil bych mu vlastnost `list-style-type` na hodnotu `none`, čímž bych vymazal odrážku. Zároveň bych v CSS stylu nastavil pro elementu `` na pozadí obrázek obsahující pomlčku. Nebo bych obrázek s pomlčkou vložil přímo jako odrážku seznamu již zmíněnou vlastností `list-style-image`

Pravý sloupec

Tento div obsahuje kód pro sekce `Special offers` a `Partners` (`<div="prave_menu">`). Ve CSS stylu jsem mu nastavil pevnou šířku a výšku v pixelech. Dále průhledné pozadí a zarovnání pomocí vlastnosti `float` vpravo.

Poslední co jsem nastavil bylo odsazení elementu zprava a shora pomocí vlastnosti `margin`. Odsazení shora jsem nastavil tak, aby byl nadpis `Special Offers` zarovnán ve stejné výšce jako nadpis uvedený v elementu `<div class="stred">`,

Special Offers

Vzhledová podoba je vytvořena pomocí CSS stylů a HTML značek. Ale obsah jednotlivých částí (nadpis, text pod obrázkem, cena) je získáván z databáze speciálních nabídek pomocí PHP skriptu (stejně, tedy kombinací CSS, HTML a databáze je tvořena celá sekce Special and fixed price offers, která obsahuje veškeré speciální nabídky).

Obrázek 11 – Detail části speciální nabídky



Partners

Sekce partnerů je již vytvořena čistě v HTML a CSS stylech. Spíše jen modifikacemi již existujících stylů pro text – jednou je přidána vlastnost font-weight (pro zvýraznění názvů jednotlivých partnerů), podruhé jsou nastaveny mezery před a za textem (jiné než je obecné nastavení pro text)

Obrázek 12 – Detail části partnerů



Obrázky jsou na stránku vloženy pomocí elementu . Mají v kaskádovém stylu nastavený rámeček o šířce jednoho pixelu a barvě odpovídající barvě nadpisu.

Patička

Poslední částí která je na stránce umístěna je patička. Ta obsahuje informace o copyrightu a odkaz na stránky VIDIA-DESIGN. Je od ostatního textu oddělena linkou, která je opět vložena elementem ``.

Copyright i odkaz mají každý vytvořenou vlastní třídu. Liší se tím, že jeden je pomocí vlastnosti float zarovnán vlevo a druhý vpravo. Liší se i velikostí mezery před elementem. Ta je zastavena tak, aby byli oba elementy horizontálně zarovnáni.

6.2.3 Přepínání mezi aktivními sekcemi

Způsob přepínání mezi jednotlivými sekcemi je řešen pomocí PHP skriptu, který vytvořil programátor, se kterým jsem při práci na těchto stránkách spolupracoval.

Jeho podstatou je, že hlavička a levé menu zůstávají stejné a jediné co se mění je obsah části, kterou jsem označil jako střed. Do ní se vkládají obsahy jednotlivých sekcí. PHP skript zajišťuje, že se při kliknutí na odkaz v levém menu zobrazí správná sekce (tj. správný Huml soubor).

Obsah jednotlivých sekcí je uložen v samostatných HTML souborech. Nejedná se o klasický HTML soubor ale spíše o „šablonu“. Kód těchto souborů neobsahuje elementy typu `<html>`, `<body>`, `<head>` aj. Není jich potřeba, protože soubor se vkládá do již existující stránky, kde je struktura stránky již vytvořena.

Všechny tyto soubory jsou uloženy v jednom adresáři (templates). Jinak by vzhledem k relativnímu adresování nebyla stránky správně zobrazovány.

6.2.4 Vytváření podstránek

Jak jsem již napsal, jednotlivé podstránky nejsou HTML souborem s klasickou strukturou. Existují dva typy podstránek. Jeden má v HTML souboru následující zápis:

```

<div class="siroky_stred">
  <h1><span class="velka"> Nadpis podstránky </span></h1>
  
  <div class="text">
    ....obsah elementu...
  </div>
</div>

```

Element `<div class="siroky_stred">` je povinný. Bez něj se stránka nezobrazí správně. V CSS stylu má nadeřinovanou požadovanou šířku, odsazení shora a zleva, plus zarovnání zleva pomocí vlastnosti `float`. To proto, aby se připojil bez odsazení k levému menu.

Dále je součástí každé podstránky nadpis, opět vytvořený elementem `<h1>`. Styl pro tento nadpis je totožný s nadpisem na úvodní straně. To samé platí pro oddělovací linku a pro element `<div class="text">`.

Druhý se liší tím, že dané podstránky v pravé části obsahují fotogalerii (jedná se zejména o stránky pro jednotlivá hřiště) – tedy sadu několika obrázků. Kód HTML souboru pak vypadá následovně:

```

<div class="prave_menu">
  <h1>Photogallery </h1>
  
  <br />
  <table border="0" width="100%">
    ... jednotlivé fotografie ...
  </table>
</div>
<div class="stred">
  <h1><span class="velka">nadpis</span></h1>
  
  <div class="text">
    ....obsah elementu...
  </div>

```

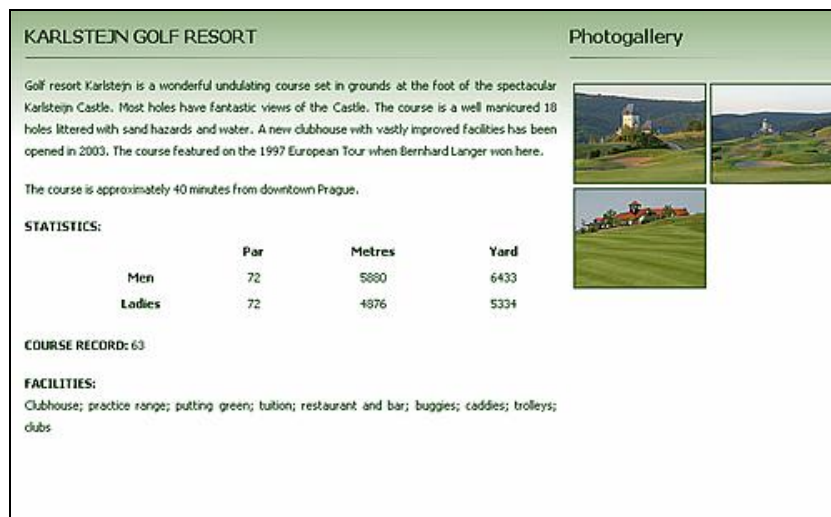
Element `<div class="stred">` má totožné vlastnosti jako první typ podstrany, jen má menší šířku.

Element `<div class="prave_menu">` má následující kód:

```
.prave_menu {height:557px;
width:250px;
margin-top: 15px;
margin-right: 7px;
background-color: transparent;
float: right;}
```

Obsahem tohoto elementu je tabulka, která obsahuje náhledy fotek. Náhledy jsou ve velikosti 120x90px a fungují jako odkaz na fotografie v jejich původní velikosti. Fotografie dodal klient, náhledy jsem si připravil sám v programu Macromedia Fireworks. Jsou to buď zmenšené fotografie, nebo jejich výřezy.

Obrázek 13 – Podoba sekce s fotogalerií



Fotografie se zobrazují se vždy do nového vyskakujícího okna (pop-up okno), jehož velikost se přizpůsobí velikosti dané fotografie (do maximálních rozměrů 600x60px). Po kliknutí do tohoto okna se okno samo zavře. Tento efekt je vytvořen pomocí javascriptu (viz. Příloha 4 – Javascript JustSoPicWindow).

6.2.5 Problémy se stránkami

Poté, co jsou naplnil informacemi jednotlivé sekce, se ukázala jedna chyba. Levé menu mělo pokračovat zelenou barvou až do konce stránky (tak jako je to teď). Ale nepokračovalo. Problém byl i v tom, že jednotlivé sekce jsou různě dlouhé, takže se jednoduše nedalo v CSS stylu nastavit pro levé menu výška napevno.

Nastavit novou barvu u elementu `<body>` také nešlo bez toho, aniž bych porušil barevný přechod tvořící pozadí.

Nakonec jsou dospěl k téměř vyhovujícímu výsledku. Zavedl jsou novou třídu a té jsem zadal potřebnou barvu pozadí. Kód je následující:

```
.pozadi{background: url(images/pozadi_leve_menu.jpg)
background-repeat: repeat-y;
width: 1000px;
overflow: hidden;}
```

Obrázek umístěný na pozadí má 1px na výšku. Šířka odpovídá šířce levého menu. Vlastnosti `background-repeat` s hodnotou `repeat-y` zajistí opakování obrázku, tentokrát nejen na šířku stránky, ale na délku. Kolikrát se bude opakovat závisí na délce stránky, tj. na délce elementu `<div="stred">` nebo `<div="siroky_stred">`. Vlastnost `overflow` s hodnotou `hidden` způsobí nezobrazení přesahující části pozadí.

Do takto vytvořené třídy jsem teprve umístil element `<div class="lm">`.

Tento způsob fungoval. Ale opět s jedním problémem. Způsobil objevení se rolovací lišty (scrollbar). Pokud by byla ve vertikálním směru, tak by to nevadilo. Objevila se ale horizontální. To bylo nevyhovující. Ale přes veškerou snahu jsem ji odstranit nedokázal. Musel mi poradit jeden kolega. Jeho řešení je následující.

```
#pozadi{background: url(images/pozadi_leve_menu.jpg) left top repeat-y;
width: 1000px;
overflow: hidden;}
```

V podstatě jen změnil již vytvořenou třídu na selektor ID a dopsal vlastnosti `left` a `top`. Až zahanbujícím způsobem jednoduché.

6.3 Optimalizace stránek

K žádné optimalizaci stránek víceméně nedošlo. Důvod pro to je prozaicky. Klient odmítl optimalizaci zaplatit, takže z ní sešlo. Jediným jejím pozůstatkem je pojmenování souborů obsahujících jednotlivé sekce. Názvy jsou uloženy v PHP souboru, kde se dají libovolným způsobem přejmenovat jen přepsáním původního názvu na nový.

6.4 Závěrečná korektura

Poslední fází, kterou musely stránky projít, byla závěrečná korektura. Její první část byla v rukou grafika, který kontroloval vzhled stránek. Druhá a důležitější korektura byla ze strany zákazníka, která byla nejen vzhledová, ale i obsahová.

Protože korektury probíhaly průběžně s tím jak se stránky vytvářely, bylo málo změn a připomínek. Jednalo se v podstatě jen o drobnosti, které se daly změnit či opravit velmi snadno. Po jejich nápravě a finálním schválení zákazníkem byly stránky připraveny na to, aby se mohly umístit na internet (ve fázi vývoje běžely stránky na zvláštní adrese).

Výsledek mé téměř měsíční práce (a dvousemestrové výuky na škole) si můžete prohlédnout na již zmíněné adrese - www.praguegolftours.com. (většinu času jsem ale strávil čekáním na podklady od zákazníka).

7. Použité zdroje:

Literatura:

- [1.] BRÁZA, Jiří. *PHP5 - začínáme programovat*. Praha: Grada Publishing, spol. s r. o., 2005. 244 s. ISBN 80-247-1146-x
- [2.] CEDERHOLM, Dan. *Webdesign s webovými standardy*. Brno: ZonerPress, s.r.o., 2004. 254 s. ISBN 80-86815-15-3
- [3.] GRAY, Daniel. *Profesionální design na webu*. Brno: SoftPress, s.r.o., 1999. 223 s. ISBN 80-902824-1-5
- [4.] KADLEC, Václav. *Mozilla Firefox – Komplettní uživatelská příručka*. Brno: CP Books, a.s., 2005. 167 s. ISBN 80-251-0644-6
- [5.] KOSEK, Jiří. *Tvorba dokonalých www stránek – podrobný průvodce*. Praha: Grada Publishing, spol. s r. o., 1998. 291 s. ISBN 80-7169-608-0
- [6.] PROKOP, Marek. *CSS Kaskádové styly pro webdesignéry*. Mobil Media, a.s , 2003, 288 s. ISBN: 80-86593-35-5
- [7.] RENDA, Miroslav. *Internet Explorer 6.0* Praha: Grada Publishing, spol. s r.o., 2002. 128 s. ISBN 80-24700153-7

Elektronické dokumenty

- [8.] World Wide Web Consortium. Dostupné z: <http://www.w3c.org>. Dne: 14.4.2006
- [9.] JANOVSKEJ Dušan , *Popis tvorby a publikování WWW stránek pro začátečníky i profesionály*. Dostupné : www.jakpsatweb.cz. . Dne: 15.4.2006
- [10.] GRIMMICH, Simon. *Tvorba webu*. Dostupné z: www.tvorba-webu.cz. Dne: 15.4.2006
- [11.] KOSEK, Jiří. *Vše o www co jste chtěli vědět ale báli jste se zeptat*. Dostupné z: www.kosek.cz. Dne 20.4.2006
- [12.] ROLNÝ, Marek. *Tvorba kvalitních webových stránek*. Dostupné z: <http://www.tvorba-www-stranek.kvalitne.cz/>. Dne 20.4.2006
- [13.] VIDIA-DESIGN. Dostupné z: www.vidia-design.cz. Dne: 29.5.2006
- [14.] Optimalizace pro vyhledávače–SEO. Dostupné z: <http://www.optimalizace-pro-vyhledavace.cz/>. Dne: 29.5.2006
- [15.] SEO Asistent. Dostupné z : <http://www.seo.unas.cz/>. Dne: 29.5.2006

- [16.] HODBOŮ, Tomáš. *Oříšek.net*. Dostupné z : z <http://www.orisek.net/?article=co-je-to-seo>. Dne: 29.5.2006
- [17.] OBORSKÝ Lukáš. *Sova v síti*. Dostupné z: http://sovavsiti.cz/2003/mod_rewrite.html. Dne: 1.6.2006
- [18.] FÁRA Jan. *Grafika&Design*. Dostupné z: <http://www.volny.cz/webgrafika/index.htm>. Dne: 10.6.2006
- [19.] Design a barvy. Dostupné z : <http://tvorba-www.bravepages.com/barvy/azdesign.htm>. Dne: 10.6.2006
- [20.] MASARYKOVA UNIVERZITA. *Dublin Core Czech*. Dostupné z : http://www.ics.muni.cz/dublin_core. Dne: 15.7.2006
- [21.] Dublin Core. Dostupné z : http://cs.wikipedia.org/wiki/Dublin_Core. Dne: 15.7.2006

Příloha 1 – Seznam obrázků

Obrázek 1 - Jednosloupcová struktura.....	11
Obrázek 2 - Vícesloupcová struktura	12
Obrázek 3 - Kruh barev.....	15
Obrázek 4 – Paleta barev	15
Obrázek 5 - Boxová koncepce.....	26
Obrázek 6 - Náhled stránky www.praguegolftours.com	35
Obrázek 7 - Navigační menu.....	39
Obrázek 8 - Položka navigačního menu	40
Obrázek 9 - Podoba hlavičky	42
Obrázek 10 - Obsah úvodní strany	44
Obrázek 11 - Detail části speciální nabídky.....	45
Obrázek 12 - Detail části partnerů.....	46
Obrázek 13 - Podoba sekce s fotogalerií.....	49

Příloha 2 – Seznam tabulek

Tabulka 1 - Kategorie odpovědí serveru.....	8
Tabulka 2 – Typy služeb.....	8
Tabulka 3 - Srovnání grafických formátů GIF a JPEG	18
Tabulka 4 – Selektory.....	21
Tabulka 5 - Standard Dublin Core.....	31

Příloha 3 - CSS styly pro stránky www.praguegolftours.com

```
body {background-color: #ffffff;
      background-image: url(images/pozadi.jpg);
      background-repeat: repeat-x;
      padding: 0;
      margin: 0;}

#pozadi{background: url(images/pozadi_leve_menu.jpg) left top repeat-y;
        width: 1000px;
        overflow: hidden;}

/*hlavička - obr, zelený řádek*/
.hlavicka {height: 173px;
           width: 100%;
           background-color: #1462AC;
           background-image: URL(images/image_2.jpg);
           float:left;
           margin: 0px;
           padding: 0px; }

.hlavicka2 {width: 749px;
            height: 52px;
            padding-left: 15px;
            background-color: #315E25;
            color: white;
            font-family: Tahoma;
            font-size: 12px;
            float:left;
            overflow: hidden;}

a.bila:link,
a.bila:visited,
a.bila:hover {color:white;
              text-decoration:none;}
```

```

a.zelena:link,
a.zelena:visited,
a.zelena:hover{color:#003300;
                text-decoration:none;}

.hl_text{width: 60px;
          font-family: Tahoma;
          font-size: 9px;
          color:white;
          float: right;
          position: relative;
          top: 32px;}

.logo { width: 112px;
        display: block;
        float: right;}

/*levé menu*/

.leve_menu {float:left;
            height:56px;
            width: 226px;}

.lm { width: 226px;
      height: 835px;
      float:left;
      background-color: #315E25;}

.lm img { display: block;}

a.current {background-image: URL(images/ico_golf_micek.jpg);
           background-repeat: no-repeat;
           background-position: 0 -6px;}

```



```
.lm_polozka { width: 226px;
              height: 22px;
              background-image: url(images/pozadi_menu_buton.jpg);
              background-repeat: repeat-x;
              font-family: Tahoma;
              color: white;
              font-size: 12px;
              font-weight: bold;
              vertical-align: middle;
              padding-top: 5px; }
```

```
.lm_polozka a { text-decoration: none;
               color: #FFFFFF;
               padding-left: 25px;
               display: block;}
```

```
.lm_polozka a:hover { text-decoration: underline;}
```

```
.leve_spodni { width: 201px;
               background-color: #315E25 ;
               color: white;
               font-size: 11px;
               font-weight: bold;
               padding-left: 20px;
               line-height: 2.2;
               font-family: Tahoma;}
```

```
.vlajka { background-image: url(images/ico_jamka.gif);
          background-repeat: no-repeat;
          margin-left: 5px;
          padding-left: 11px;}
```

```
.leve_spodni a { color: #FFFFFF;
                 text-decoration: none;}
```

```
.leve_spodni a:hover {text-decoration: underline;}
```

```
.micek {float:left;
        width: 18px;
        height: 27px;}
```

/*střední část*/

```
.stred{width: 490px;
        margin-left: 15px;
        margin-top: 15px;
        float:left; }
```

```
.siroky_stred {width: 750px;
                margin-top: 15px;
                margin-left: 15px;
                float:left;}
```

```
.stred a,
.siroky_stred a {text-decoration: none;
                 color:#003300;
                 font-weight: bold; }
```

```
.stred a:hover,
.siroky_stred a:hover {text-decoration: underline;
                       color:#003300;
                       font-weight: bold; }
```

/*pravé menu*/

```
.prave_menu {height:557px;
              width:250px;
              margin-top: 15px;
              margin-right: 7px;
              background-color: transparent;
              float: right;}
```

```
.pm_more { position: relative;
            top: 0px;
            left: 1px;
            width: 40px;
            height: 19px;
            font-family: Tahoma;
            font-size: 11px;
            color: white;
            background-color: #315E25;
            padding-left: 8px;
            float: right; }
```

```
.pm_offer { width: 90;
            font-weight: bold;
            font-family: Tahoma;
            font-size: 12px;
            color: #003300;
            text-transform: uppercase;
            float: left; }
```

```
.pm_text { font-family: Tahoma;
           font-size: 11px;
           color: #003300;
           line-height: 1.5;
           padding-bottom: 5px;
           margin-top: 2px; }
```

/* design by*/

```
.design { width: 114px;
         font-family: Tahoma;
         font-size: 9px;
         color: #003300;
         float: right;
         margin-right: 2px; }
```

```
#footer {width: 100%;  
clear: both;}
```

```
/*copyright*/
```

```
.spodek {width: 160px;  
height: 10px;  
font-family: Tahoma;  
font-size: 9px;  
color:#003300;  
float:left;  
margin-top: 10px;  
margin-bottom: 2px;}
```

```
/*zbytek*/
```

```
img.text_t_img {border: 1px solid #003300;  
margin-bottom: 0px; }  
img.jedna {border-top: 1px solid #315E25;  
border-bottom: 1px solid #315E25;}  
img.sekce_more {border: 1px solid #003300;  
float:left;}  
  
.text {font-family: Tahoma;  
font-size: 11px;  
color:#003300;  
line-height: 1.8;  
text-align: justify; }  
  
h1 {font-family: Tahoma;  
font-size: 16px;  
color:#003300;  
margin-bottom: 10px;  
margin-top: 0px; }
```

```
.text_t {font-family: Tahoma; /*text -tucne*/
  font-size: 11px;
  color:#003300;
  font-weight: bold;
  padding-bottom: 10px;
  margin-top: 8px;}

.copyright {font-family: Tahoma;
  font-size: 9px;
  color:#003300;}

.dvanacka {font-family: Tahoma;
  font-size: 12px;
  color: white;}

.velka {text-transform: uppercase;}

.zarovnani_vpravo {text-align: right;}

p {margin-top: 15px ;
  margin-bottom: 15px;}

ol {margin-top: 0px;
  margin-bottom: 30px;}

.kurziva {font-style: italic;}

a.spodek {text-decoration: none;
  color:#003300;
  font-weight: normal;}

a.spodek:hover {text-decoration: underline;
  color:#003300;
  font-weight: normal;}
```

```

a.tlaciditko { font-size: 12px;
               font-weight: bold;
               text-decoration: none;
               padding: 4px 4px 6px 4px;
               float:right;
               margin-right: 18px; }

a.tlaciditko:link,
a.tlaciditko:visited { background: #315E25;
                      color: #ffffff }

a.tlaciditko:hover { background: #315E25;
                    color: #ffffff; }

```

/* nasteveni inputů v casti book here */

```

input.green,
select.green,
textarea.green {color: #003300;
                font-size: 11px;}

```

/*testamonials*/

```

.testamontials {width: 750px;
                padding-left: 15px;
                padding-top: 15px;
                background-image: url(images/pozadi_prechod.jpg);
                background-repeat: repeat-x;
                float: left;}

.testam {margin-top: 11px;}

.testam_head {font-size: 12px;
              font-weight: bold;
              color: #006600;}

.testam_datum {font-size: 12px;
               color: #666666;}

```

```
.testam_body { margin-top: 5px;
                margin-bottom: 11px;
                width: 600px;
                float: left;estam_img
                width: 130px;
                float: left;
                margin-left: 12px;
                margin-bottom: 11px;
                text-align: right;}

.testam_img img { border: 1px solid #003300;}

.testam_active { font-weight: bold;
                 color: #CC0000;
                 text-decoration: underline;}
```

Příloha 4 – Javascript JustSoPicWindow.

```
<SCRIPT Language="JavaScript">
```

function

```
JustSoPicWindow(imageName,imageWidth,imageHeight,alt,bgcolor,hugger,hugMargin) {
```

```
// by E Michael Brandt of ValleyWebDesigns.com - Please leave these comments intact.
```

```
// version 3.0.4
```

```
if (bgcolor=="") {bgcolor="#FFFFFF";} 
```

```
var adj=10
```

```
var w = screen.width;
```

```
var h = screen.height;
```

```
var byFactor=1;
```

```
if(w<740){ var lift=0.90;}
```

```
if(w>=740 & w<835){var lift=0.91; }
```

```
if(w>=835){var lift=0.93;}
```

```
if (imageWidth>w){
```

```
    byFactor = w / imageWidth;
```

```
    imageWidth = w;
```

```
    imageHeight = imageHeight * byFactor;
```

```
}
```

```
if (imageHeight>h-adj){
```

```
    byFactor = h / imageHeight;
```

```
    imageWidth = (imageWidth * byFactor);
```

```
    imageHeight = h;
```

```
}
```

```
var scrWidth = w-adj;
```

```
var scrHeight = (h*lift)-adj;
```

```
if (imageHeight>scrHeight){
```

```
    imageHeight=imageHeight*lift;
```

```
    imageWidth=imageWidth*lift;
```

```
}
```

```
var posLeft=0;
```

```
var posTop=0;
```



```

if (hugger == "hug image"){
    if (hugMargin == ""){hugMargin = 0; }
    var scrHeightTemp = imageHeight - 0 + 2*hugMargin;
    if (scrHeightTemp < scrHeight) {scrHeight = scrHeightTemp; }
    var scrWidthTemp = imageWidth - 0 + 2*hugMargin;
    if (scrWidthTemp < scrWidth) {scrWidth = scrWidthTemp; }
    if (scrHeight<100){scrHeight=100;}
    if (scrWidth<100){scrWidth=100;}
    posTop = ((h-(scrHeight/lift)-adj)/2);
    posLeft = ((w-(scrWidth)-adj)/2);
}
if (imageHeight > (h*lift)-adj || imageWidth > w-adj){
    imageHeight=imageHeight-adj;
    imageWidth=imageWidth-adj;
}
posTop = parseInt(posTop);
posLeft = parseInt(posLeft);
scrWidth = parseInt(scrWidth);
scrHeight = parseInt(scrHeight);
var agt=navigator.userAgent.toLowerCase();
if (agt.indexOf("opera") != -1){
    var args= new Array();
    args[0]='parent';
    args[1]=imageName;
    var i ; document.MM_returnValue = false;
    for (i=0; i<(args.length-1); i+=2) eval(args[i]+" .location='"+args[i+1]+'");
}
else {
newWindow =window.open("vwd_justso.htm","newWindow","width="+scrWidth+",
    height="+scrHeight+",left="+posLeft+",top="+posTop);
newWindow.document.open();
newWindow.document.write('<html><title>'+alt+'</title><body leftmargin="0"

```

```
topmargin="0" marginheight="0" marginwidth="0" bgcolor='+bgcolor+'
onBlur="self.close()" onClick="self.close()">');

newWindow.document.write('<table width='+imageWidth+' border="0" cellspacing="0"
cellpadding="0" align="center" height='+scrHeight+' ><tr><td>');

newWindow.document.write('');

newWindow.document.write('</td></tr></table></body></html>');
newWindow.document.close();
newWindow.focus();
    }
}
</SCRIPT>
```