



Tvorba www stránek v HTML a CSS

Marek Laurenčík

- Základy jazyka HTML
- Základy CSS stylů a jejich použití na webových stránkách
- Práce s databází MySQL
- Úvod do využití jazyka Javaskript, použití na webových stránkách
- Základy jazyka PHP a jeho využití na webových stránkách
- Využití databáze MySQL pro tvorbu webových stránek



Tvorba www stránek v HTML a CSS

Marek Laurenčík

Grada Publishing

Upozornění pro čtenáře a uživatele této knihy

Všechna práva vyhrazena. Žádná část této tištěné či elektronické knihy nesmí být reprodukována a šířena v papírové, elektronické či jiné podobě bez předchozího písemného souhlasu nakladatele. Neoprávněné užití této knihy bude **trestně stíháno**.

Marek Laurenčík

Tvorba www stránek v HTML a CSS

Vydala Grada Publishing, a.s.
U Průhonu 22, Praha 7
obchod@grada.cz, www.grada.cz
tel.: +420 234 264 401, fax: +420 234 264 400
jako svou 7289. publikaci

Spoluautor Michal Bureš
Odpovědný redaktor Petr Somogyi
Sazba Petr Somogyi
Ikony Freepik (www.flaticon.com)
Počet stran 224
První vydání, Praha 2019
Vytiskly Tiskárny Havlíčkův Brod, a. s.

© Grada Publishing, a.s., 2019
Cover Design © Grada Publishing, a. s., 2019
Cover Photo © Depositphotos/fraeje

Názvy produktů, firem apod. použité v knize mohou být ochrannými známkami nebo registrovanými ochrannými známkami příslušných vlastníků.

ISBN 978-80-271-2559-3 (ePub)
ISBN 978-80-271-2946-1 (pdf)
ISBN 978-80-271-2241-7 (print)

Obsah

	Předmluva	13
1	Úvod	15
1.1	Soubory tvořící webové stránky	15
1.2	Postup při tvorbě webových stránek	16
1.3	Struktura webu a názvy souborů	16
1.4	Publikování webu	17
1.5	Kopírování souborů na webový server	17
2	Tvorba kódu HTML	19
2.1	Elementy jazyka HTML	19
2.1.1	Struktura souboru HTML	21
2.2	Tvorba textu	23
2.2.1	Odstavce	24
2.2.2	Pevná mezera a další entity	25
2.2.3	Rozklikávací text	26
2.2.4	Změna vzhledu písma	26
2.2.5	Nadpisy	27
2.2.6	Kopírování textu z Wordu	27
2.3	Tvorba seznamů	28
2.3.1	Seznamy s odrážkami	28
2.3.2	Číslované seznamy	29
2.3.3	Víceúrovňové seznamy	30
2.3.4	Kopírování seznamu z Wordu	30
2.3.5	Definiční seznamy	30
2.4	Obrázky, multimédia a grafické efekty	31
2.4.1	Obrázek v samostatném odstavci	32
2.4.2	Umístění obrázku do odstavce s textem	33
2.4.3	Obrázek v elementu <figure>	34
2.4.4	Videoklipy	34
2.4.5	Audio (zvukový klip)	35
2.4.6	Vodorovná čára	36
2.4.7	Běžící nápis	36
2.5	Hypertextové odkazy	38
2.5.1	Odkaz na jinou stránku ve stejném webu	38
2.5.2	Odkaz na obrázek	39
2.5.3	Odkaz na jiný soubor	39
2.5.4	Odkaz na videoklip	39
2.5.5	Odkaz na adresu elektronické pošty	40
2.5.6	Odkaz na jiný web	40
2.5.7	Práce se záložkami	40



2.5.8	Hypertextový odkaz z obrázku.....	41
2.5.9	Obrázková mapa.....	41
2.5.10	Odkaz v rámečku.....	42
2.6	Tvorba tabulek.....	43
2.6.1	Struktura tabulky.....	43
2.6.2	Orámování a umístění tabulky.....	43
2.6.3	Výška a šířka.....	44
2.6.4	Písmo.....	44
2.6.5	Vodorovné a svislé zarovnání.....	45
2.6.6	Nadpis tabulky.....	45
2.6.7	Barvy rámečků a pozadí.....	45
2.6.8	Slučování buněk.....	46
2.6.9	Rozšířené možnosti orámování v tabulce.....	46
2.6.10	Formátování celého sloupce.....	48
2.7	Formuláře pro zadávání dat.....	49
2.7.1	Element pro tvorbu formuláře.....	49
2.7.2	Tvorba příkazových tlačítek.....	50
2.7.3	Popisky.....	50
2.7.4	Textová políčka.....	50
2.7.5	Volby.....	52
2.7.6	Výběr pomocí tlačítka.....	52
2.7.7	Textové okénko.....	53
2.7.8	Seznamy.....	53
2.7.9	Skupina prvků.....	54

3

	Kaskádové styly.....	56
3.1	Použití stylů na webové stránce.....	56
3.1.1	Tabulky kaskádových stylů.....	56
3.1.2	Vnořené styly.....	57
3.1.3	Externí styly.....	57
3.1.4	Selektory a jejich použití.....	57
3.1.5	Pravidla pro psaní stylů.....	59
3.1.6	Dědičnost a hierarchie stylů.....	59
3.1.7	Slučování stylů, konflikty, chyby a výběr pravidel.....	60
3.1.8	Pseudotřídy.....	60
3.1.9	Jednotky, používané ve stylech.....	61
3.1.10	Zadání barev a obrázků.....	61
3.2	Formátování textu.....	62
3.2.1	Typ a velikost písma.....	62
3.2.2	Zvýraznění písma.....	62
3.2.3	Barva a stín písma.....	62
3.2.4	Další možnosti pro vzhled písma.....	62
3.2.5	Úprava textu.....	63
3.2.6	Vzdálenosti v textu.....	63
3.2.7	První písmeno v odstavci.....	64
3.3	Formátování odstavce.....	64
3.3.1	Zarovnání textu a šířka textu.....	64
3.3.2	Orámování odstavce.....	65
3.3.3	Další možnosti rámečků.....	66
3.3.4	Okraje odstavce.....	66

3.3.5	Pozadí odstavce.....	67
3.3.6	Stínování odstavce.....	67
3.3.7	Dělení textu do sloupců.....	68
3.3.8	Zalamování textu v odstavci.....	68
3.4	Formátování hypertextových odkazů, seznamů a obrázků.....	69
3.4.1	Formátování hypertextových odkazů.....	69
3.4.2	Nastavení odrážek a číslování.....	69
3.4.3	Odrážky vytvořené obrázkem.....	70
3.4.4	Formátování víceúrovňových seznamů.....	70
3.4.5	Formátování obrázků.....	71
3.5	Formátování tabulek.....	71
3.5.1	Nastavení písma.....	71
3.5.2	Šířka.....	71
3.5.3	Zarovnání.....	72
3.5.4	Odsazení.....	72
3.5.5	Orámování buněk.....	73
3.5.6	Pozadí a stínování.....	73
3.5.7	Formátování celého sloupce najednou.....	74
3.6	Práce s bloky.....	74
3.6.1	Zobrazení bloků.....	74
3.6.2	Rozměry a přetékání bloků.....	75
3.6.3	Umístění bloků.....	75
3.6.4	Obtékání bloků.....	75
3.7	Praktické použití bloků.....	77
3.7.1	Tři různě umístěné bloky.....	77
3.7.2	Obtékáný text.....	77
3.7.3	Obtékané první písmeno v odstavci.....	78
3.7.4	Text s obtékajícími obrázky.....	78
3.7.5	Dva sloupce vedle sebe.....	78
3.7.6	Tři sloupce stejné šířky s pevným obsahem.....	79
3.7.7	Hlavička, patička, okraje.....	80
3.7.8	Kartičky.....	80
3.7.9	Galerie fotografií.....	81
3.8	Tvorba nabídek.....	82
3.8.1	Tlačítka pod sebou.....	82
3.8.2	Tlačítka vedle sebe.....	82
3.8.3	Nabídkový panel.....	82
3.8.4	Záložky.....	83
3.9	Tvorba vrstev.....	84
3.9.1	Obrázek na pozadí.....	84
3.9.2	Vrstvení bloků.....	85
3.10	Formátování formulářů.....	86
3.10.1	Formátování formuláře.....	86
3.10.2	Popisky ve formuláři.....	87
3.10.3	Formátování tlačítek.....	87
3.10.4	Formátování seznamů.....	87
3.10.5	Formátování elementu <textarea>.....	88
3.10.6	Formátování skupiny elementů.....	88
3.10.7	Formátování legendy.....	88
3.11	Přehled formátovacích atributů.....	89

Databázový systém MySQL	91
4.1 Vytvoření nové tabulky a její struktura	93
4.1.1 Vytvoření nové tabulky	93
4.1.2 Tvorba názvů	94
4.1.3 Úložiště tabulek	94
4.1.4 Textové typy polí	94
4.1.5 Celočíselné typy polí	95
4.1.6 Číselné typy s desetinnou čárkou	95
4.1.7 Typy pro kalendářní data	95
4.1.8 Speciální typy polí	95
4.1.9 Primární klíč a další indexy	96
4.1.10 Změna ve struktuře tabulky	96
4.1.11 Seznam indexů	97
4.1.12 Relace mezi tabulkami	97
4.1.13 Hodnota Null	98
4.1.14 Automatické číslo (Autoincrement)	99
4.1.15 Výchozí hodnoty u nového záznamu	99
4.2 Práce s tabulkami	99
4.2.1 Další akce s tabulkami	100
4.3 Jednoduchý příkaz SELECT	100
4.3.1 Základní části příkazu SELECT	101
4.3.2 Označení sloupce v dotazu	101
4.3.3 Výběr jedinečných hodnot	101
4.3.4 Omezení počtu zobrazených řádků	102
4.3.5 Řazení v dotazu	102
4.3.6 Výběr nejmenších a nejvyšších hodnot	102
4.3.7 Tvorba pohledů	103
4.4 Filtrace	104
4.4.1 Negace výrazu	104
4.4.2 Logické a nulové hodnoty	105
4.4.3 Kombinace výrazů v podmínce	105
4.4.4 Filtrace podle rozsahu	106
4.4.5 Filtrace podle seznamu	106
4.4.6 Filtrace podle části textu	106
4.4.7 Fulltextová filtrace	107
4.4.8 Využití více polí při filtraci	107
4.5 Využití regulárních výrazů	108
4.5.1 Použití regulárního výrazu ve filtru	109
4.5.2 Text na začátku a na konci	109
4.5.3 Seznam znaků	109
4.5.4 Zástupný symbol	110
4.5.5 Opakování	110
4.5.6 Výskyt zástupných znaků ve filtrovaném textu	110
4.5.7 Kvantifikátory	110
4.5.8 Kontrola e-mailové adresy	111
4.6 Výpočty a standardní funkce	112
4.6.1 Podmíněný výpočet	112
4.6.2 Ošetření hodnoty NULL	113
4.6.3 Matematické funkce	113
4.6.4 Výrazy s hodnotami kalendářního data	113
4.6.5 Práce s textovými hodnotami	114

4.7	Spojování více tabulek.....	117
4.7.1	Vnitřní spojení dvou tabulek.....	117
4.7.2	Spojení více tabulek.....	118
4.7.3	Vypočtené sloupce.....	118
4.7.4	Filtrace a řazení.....	119
4.7.5	Vnější spojení.....	119
4.8	Sjednocovací dotazy.....	120
4.8.1	Filtrace a řazení ve sjednocovacím dotazu.....	121
4.8.2	Plné spojení mezi tabulkami.....	121
4.9	Tvorba souhrnů.....	122
4.9.1	Seskupování.....	122
4.9.2	Agregační funkce.....	122
4.9.3	Použití funkce COUNT.....	123
4.9.4	Celkový souhrn na konci dotazu.....	123
4.9.5	Seskupení záznamů bez agregačního výpočtu.....	123
4.9.6	Řazení.....	124
4.9.7	Filtrace.....	124
4.9.8	Vypočtené sloupce.....	125
4.9.9	Vnitřní spojení tabulek v dotazu se seskupením.....	126
4.9.10	Vnější spojení tabulek v dotazu se seskupením.....	126
4.10	Poddotazy.....	127
4.10.1	Skalární poddotazy.....	128
4.10.2	Sloupcové poddotazy.....	128
4.11	Akční dotazy.....	130
4.11.1	Vkládání záznamů do tabulky.....	130
4.11.2	Přepis záznamů v tabulce.....	131
4.11.3	Odstraňování záznamů v tabulce.....	132
4.12	Údržba databáze.....	132
4.12.1	Archivace a obnova dat.....	132
4.12.2	Export dat do Excelu.....	133
4.12.3	Import dat z Excelu.....	134
4.13	Tvorba vlastních funkcí.....	136
4.13.1	Tvorba kódu.....	136
4.13.2	Psaní kódu.....	137
4.13.3	Příklady kódu.....	138

5

	Úvod do JavaScriptu.....	140
5.1	Základy jazyka.....	140
5.1.1	Použití proměnných.....	141
5.1.2	Tvorba vzorců.....	142
5.1.3	Výstup výsledků na webovou stránku.....	142
5.1.4	Podmíněné provádění příkazů.....	142
5.1.5	Podmínečný výraz.....	144
5.1.6	Přepínač.....	144
5.1.7	Nepřímý výraz.....	145
5.1.8	Psaní vlastních funkcí.....	145
5.2	Dialogová okna JavaScriptu.....	147
5.2.1	Zpráva na obrazovce.....	147
5.2.2	Vstup z klávesnice.....	147
5.2.3	Kontrolní dotaz.....	148



5.3	Práce s poli a cykly	148
5.3.1	Tvorba pole	149
5.3.2	Cyklus s konečným počtem kroků	149
5.3.3	Cyklus s neznámým počtem kroků	150
5.3.4	Zjištění počtu prvků pole – vlastnost length	151
5.3.5	Hlavní metody pro práci s polem	151
5.3.6	Pole jako návratová hodnota funkce	152
5.4	Práce s texty	153
5.4.1	Slučování textových a číselných hodnot	153
5.4.2	Délka textu	153
5.4.3	Úpravy textů	153
5.4.4	Vyjímání části textového řetězce	154
5.4.5	Hledání a nahrazování	154
5.4.6	Rozdělování a slučování textu	155
5.4.7	Použití regulárních výrazů v JavaScriptu	156
5.5	Práce s kalendářním datem a časem	157
5.5.1	Tvorba objektu pro datum a čas	157
5.5.2	Převod kalendářní a časové hodnoty na text	158
5.5.3	Převod kalendářní a časové hodnoty na číslo	158
5.5.4	Změna kalendářního data a času	158
5.6	Události a časování	160
5.6.1	Spojení skriptu s událostí	160
5.6.2	Události myši	160
5.6.3	Události stránky	161
5.6.4	Časování	162
5.7	Otvírání oken	162
5.7.1	Metoda open	162
5.7.2	Další možnosti metody open	163
5.7.3	Zavření vytvořeného okna	164
5.7.4	Spuštění metody open	164
5.8	Práce s cookies	165
5.8.1	Čtení cookies	165
5.8.2	Tvorba cookie	165
5.8.3	Odstranění cookie	166
5.9	Práce s obrázky	166
5.9.1	Nastavení odkazu na obrázek	166
5.9.2	Využití kolekce images pro záměnu obrázku	167
5.9.3	Dynamické orámování obrázku	167
5.9.4	Preload obrázku	168
5.10	Kód ve formuláři	169
5.10.1	Využití prvku button	169
5.10.2	Identifikace formuláře a jeho prvků	169
5.10.3	Načtení hodnot z formuláře do JavaScriptu	170
5.10.4	Hodnoty ze seznamů	171
5.10.5	Manipulace s prvky na formuláři	172
5.10.6	Události formuláře	173
5.10.7	Funkce pro kontrolu údajů ve formuláři	174
5.10.8	Kontrola kalendářních údajů	175
5.11	Dynamické nastavení vzhledu stránky	176
5.11.1	Změna pozadí stránky	176
5.11.2	Využití objektu style	176

5.11.3	Využití objektu style v událostech na stránce	177
5.11.4	Využití objektu style ve formuláři	178

6

	Úvod do jazyka PHP	179
6.1	Tvorba kódu	179
6.1.1	Komentáře	180
6.1.2	Syntaktická pravidla	180
6.1.3	Výstup na obrazovku	180
6.1.4	Ukončení kódu	181
6.	Programové konstrukce	182
6.2.1	Přiřazování hodnot	182
6.2.2	Konstanty	182
6.2.3	Podmíněné příkazy	183
6.2.4	Podmínky a logické výrazy	183
6.2.5	Logická hodnota, vrácená dosazovacím příkazem	184
6.2.6	Podmíněný výraz	184
6.2.7	Přepínač	184
6.2.8	Cyklus se známým počtem kroků	185
6.2.9	Cyklus s neznámým počtem kroků	186
6.3	Použití funkcí	186
6.3.1	Matematické funkce	186
6.3.2	Tvorba vlastních funkcí	187
6.3.3	Umístění vlastní funkce v kódu PHP	188
6.4	Práce s poli	188
6.4.1	Naplnění pole hodnotami	188
6.4.2	Cyklus pro prvky pole	189
6.4.3	Pole ve vlastní funkci	189
6.4.4	Počet prvků v poli	190
6.4.5	Ukazatel pole	190
6.4.6	Testování pole	190
6.4.7	Přiřazení hodnot z pole proměnným	191
6.5	Práce s texty a regulárními výrazy	191
6.5.1	Malá a velká písmena	192
6.5.2	Délka textu a uřezávání	192
6.5.3	Vyhledávání	193
6.5.4	Nahrazování	193
6.5.5	Převod čísla na text	194
6.5.6	Textové hodnoty a pole	194
6.5.7	Úprava kódu HTML	195
6.5.8	Použití regulárních výrazů	196
6.6	Práce s kalendářními a časovými hodnotami	197
6.6.1	Časové razítko	198
6.6.2	Převod časového razítka na text a pole	198
6.6.3	Tvorba kalendářní proměnné	199
6.6.4	Posuv kalendářních hodnot	199
6.6.5	Rozdíl kalendářních hodnot	200
6.6.6	České názvy v kalendářních údajích	201
6.7	Komunikace kódu s okolím	201
6.7.1	Cookies	201
6.7.2	Sessions	202

6.7.3	Práce s textovými soubory.....	203
6.7.4	Posílání e-mailových zpráv.....	204
6.8	Zpracování údajů z formuláře	205
6.8.1	Textová pole.....	205
6.8.2	Přepínač a zaškrtnávací volba	206
6.8.3	Element <textarea>	206
6.8.4	Seznamy	207
6.8.5	Více odesílacích tlačítek na formuláři	207
6.8.6	Stažení souboru na server	207
6.9	Použití databáze MySQL	208
6.9.1	Spojení s databází.....	209
6.9.2	Nastavení znakové sady.....	209
6.9.3	Spuštění dotazu.....	209
6.9.4	Zpracování výsledků dotazu.....	210
6.9.5	Zobrazení výsledků dotazu.....	211
6.9.6	Tvorba tabulky	212
6.9.7	Práce s kalendářními hodnotami.....	213
6.9.8	Využití databáze při tvorbě formuláře	213
6.9.9	Práce s daty v tabulce	213
6.10	Tvorba nabídek.....	214
6.10.1	Předání kódu stránky přes URL.....	215
6.10.2	Využití formuláře pro tvorbu menu.....	215
6.10.3	Zobrazení obsahu stránky	216
6.10.4	Tvorba nabídky z tabulky.....	216
	Rejstřík	218

Předmluva

Tato kniha pojednává o tvorbě webových stránek. V současné době zahrnuje tato práce činnosti z několika oblastí. Doba, kdy tvůrci webových stránek stačilo ovládat jazyk HTML a práci s kaskádovými styly, již patří minulosti. Moderní přístup ke tvorbě webu vyžaduje využití databáze, což vyžaduje ovládání základů jazyka SQL. Pro spojení databáze s webovou stránkou ovšem musíte ovládat aspoň základy jazyka PHP, pro tvorbu dynamických webových stránek se tvůrce zase neobejde bez základů dalšího programovacího jazyka – Javascriptu.

Jedná se o značně širokou oblast témat, jejíž jednotlivé složky jsou mnohdy uváděny odděleně. Cílem této knihy je představit základy všech pěti oblastí využívaných při tvorbě webových stránek: jsou to jazyk HTML, kaskádové styly CSS, práce s databází My SQL a základy jazyka SQL, úvod do dvou uvedených programovacích jazyků: Javascriptu a PHP. Přitom je samozřejmě kladen důraz i na jejich vzájemné souvislosti.

Rozsah knihy samozřejmě neumožňuje podat u všech pěti oblastí látku do hloubky. Týká se to zejména posledních dvou kapitol zabývajících se programováním. To však ani nebylo naším cílem. Studijního materiálu o Javascriptu a PHP je k dispozici poměrně dost a tato kniha by měla posloužit jak pro zvládnutí základních a nejčastěji používaných programovacích postupů, tak se stát dostačujícím úvodem pro seznámení se s pokročilými programovacími postupy.

Knihy je určena především pro čtenáře, kteří s tvorbou webových stránek začínají, a měla by jim umožnit od začátku přistupovat k této práci v plné komplexnosti. Užitečná může být i pro všechny, kdo již s webovými stránkami nějaké zkušenosti mají a chtěli by si své znalosti rozšířit a prohloubit.

V knize jsme použili následující typografické konvence:

- Kód HTML, stylů CSS a jazyků SQL, Javascript a PHP je psán neproporčním písmem.
- Příkazy v programech a názvy adresářů jsou zvýrazněny **tučně**.
- Názvy kláves jsou odlišeny pomocí KAPITÁLEK.

Ať se Vám daří!

Autoři

1 Úvod

Celosvětová internetová síť propojuje dva základní druhy počítačů:

- servery, které uchovávají a nabízejí dokumenty s informacemi,
- klienty, které tyto dokumenty přejímají, zobrazují a nabízejí uživatelům.

Internet byl od počátku vyvíjen jako prostředek zcela nezávislý na počítačovém vybavení na straně klienta, internetové dokumenty proto mohou obsahovat pouze text. Všechny obrázky, fotografie a další grafické objekty musí být uloženy v samostatných souborech a internetový dokument se na tyto soubory odvolá pomocí odkazů.

Zobrazování internetového dokumentu na straně klienta zajistí webový prohlížeč. Jedním z nich je Microsoft Internet Explorer, který je součástí operačního systému Windows. Jiné často používané prohlížeče (Firefox, Google Chrome, Edge, Opera apod.) lze bezplatně nainstalovat z řady webů.

Správné zobrazení internetového dokumentu v prohlížeči zajistí řídící příkazy pro práci prohlížeče, které se píší v jazyku HTML (Hypertext Markup Language). Takto vytvořený dokument má příponu HTM nebo HTML. Soubor typu HTML se často označuje také jako webová stránka. Součástí každého souboru HTML je rovněž formátování zobrazovaného textu. To je možné provést dvěma způsoby:

- údaje o vzhledu písma, odstavce apod. zapsat přímo pomocí příkazů jazyka HTML,
- zapsat informace o formátování odděleně pomocí nástroje zvaného „style“ a v jazyku HTML se na vytvořené style odvolávat.

Současným trendem při tvorbě webových stránek je přesouvat co nejvíce formátovacích informací do stylů.

Kromě jazyka HTML se k tvorbě webových stránek často používají programovací jazyky JavaScript a PHP. JavaScript provádí kód na straně klienta a umožňuje oživit webové stránky dynamickými efekty. Jazyk PHP obsahuje nástroje pro webové stránky, prováděné na straně serveru. Prostředky jazyka PHP umožňují například měnit obsah webové stránky podle různých podmínek, odesílat e-mailové zprávy a komunikovat s databází, v níž jsou uchovány informace pro tvorbu webových stránek.

1.1 Soubory tvořící webové stránky

Pojmem „webové stránky“ nebo „web“ se označuje série dokumentů od jednoho autora nebo firmy, nabízených veřejnosti na straně serveru. Každý web obsahuje zpravidla následující soubory:

- Soubor s názvem **index**. Jedná se o úvodní stránku celého webu, která se při připojení na web zobrazí každému návštěvníkovi jako první. Soubor **index** musí být umístěn v kořenovém adresáři, který obsahuje publikované webové stránky.

- Další soubory, tvořící vlastní webové stránky.
- Obrázky a další grafické soubory.
- Soubory typu PDF, které lze prohlížet pomocí programu Acrobat Reader.
- Soubory typu CSS: tyto soubory nemají vlastní obsah, ale obsahují definice použitých stylů. Použitím souborů CSS dosáhneme toho, aby byl vzhled celého webu jednotný.
- Podle potřeby mohou webové stránky obsahovat také dokumenty Wordu, sešity Excelu nebo jiné soubory, které jsou zpravidla určeny ke stažení ze serveru do počítače na straně klienta.

1.2 Postup při tvorbě webových stránek

Tvorba webových stránek má zpravidla následující fáze:

- Příprava obsahu stránek (tedy textů, obrázků, tabulek apod.).
- Návrh rozvržení webu a jeho logiky.
- Tvorba úvodní stránky (**index**) a ostatních stránek včetně programového kódu.
- Příprava souborů se styly (CSS) pro jednotné formátování webu.

Pro přípravu webových stránek budeme potřebovat následující programové vybavení:

- Program pro editaci souborů HTML. Protože se jedná o textové soubory, můžete je vytvářet i v Poznámkovém bloku. Lepší je však použít některý z editorů HTML, které barevně odlišují jednotlivé části kódu – práce s nimi je pohodlnější. Mezi často používané programy tohoto druhu patří například program **PsPad** (je k dispozici v češtině a bezplatně na webech jako Stahuj, Kopretina apod.).
- Program pro práci s grafickými soubory. Můžeme použít libovolný z programů, jež umožňují úpravu obrázků: změnu rozměrů, změnu typu souboru (přípony), zesvětlení obrázku.
- Několik webových prohlížečů. Kód HTML není ve všech webových prohlížečích interpretován zcela stejně. Totéž platí i pro styly. Výsledkem je, že jeden a týž soubor může mít v různých prohlížečích poněkud odlišný vzhled. Největší rozdíly jsou mezi MS Internet Exporem a ostatními prohlížeči, avšak ani další často používané prohlížeče nefungují ve všem zcela stejně. Proto je nutné připravované webové stránky vyzkoušet v různých prohlížečích (například Google Chrome, FireFox, Opera, Edge atd.).

1.3 Struktura webu a názvy souborů

Tvorbu webu a jeho zkoušení provádíme zpravidla nejprve na lokálním počítači. Vytvářený web umístíme do samostatného adresáře. Aby byla jeho struktura přehledná, je vhodné využívat podsložky (umístíme do nich například obrázky, soubory PDF, soubory stylů CSS či další soubory). Na webovém serveru potom vytvoříme přesně stejnou strukturu.

Webové servery využívají operační systém Unix, který používá odlišná pravidla než operační systém Windows:

- nepřipouští názvy s českou diakritikou, mezerami, čárkami a podobnými znaky,
- v názvech **rozlišuje** malá a velká písmena,
- při zápisu cest k souborům se používá obyčejné (nikoliv zpětné) lomítko.

Všechny názvy souborů a podsložek v adresáři s webem proto upravíme, aby obsahovaly jen:

- malá písmena abecedy bez písmen s háčky a čárkami,
- číslice,
- pomlčku a podtržítko.

Tyto zásady je nutné důsledně dodržovat i při přípravě webu na lokálním disku. Jejich nedodržení může vést k tomu, že například odkaz na špatně pojmenovaný obrázek bude na lokálním disku fungovat, ale na webovém serveru již nikoliv.

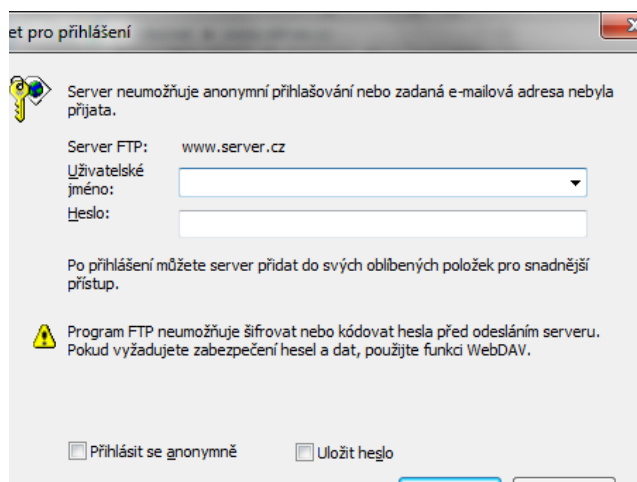
1.4 Publikování webu

Vytvořené webové stránky je třeba zpřístupnit veřejnosti. Provedeme to ve třech krocích:

1. **Registrace domény** – tímto krokem si zarezervujeme jedinečný název, pod nímž budou webové stránky nabízeny uživatelům. Poslední část názvu (přípona) může být národní (cz) nebo mezinárodní (eu, com apod.). Registraci domén provedeme pomocí internetu, cena za doménu se pohybuje řádově ve stovkách korun za rok.
2. **Objednání webhostingu** – jedná se o pronájem diskové plochy na webovém serveru, který zpřístupní webové stránky ostatním uživatelům. Pronájem provedeme opět pomocí internetu, a to bezplatně, nebo za poplatek. Bezplatný webhosting nabízí méně služeb než placený (například využívání databáze je omezené).
3. **Překopírování webových stránek a dalších potřebných souborů na webový server** – tento přenos se provede pomocí protokolu FTP (File Transfer Protocol) a je podrobně popsán dále.

1.5 Kopírování souborů na webový server

Pro kopírování potřebných souborů na webový server využijeme internet, spuštěný pod protokolem FTP. Na rozdíl od běžně používaného protokolu HTTP neumožňuje soubory na webovém serveru prohlížet, umožňuje však s nimi pracovat obdobným způsobem jako programy Průzkumník nebo Tento počítač (tedy soubory kopírovat, mazat, vytvářet nové složky apod.).



Obrázek 1.1 Přihlašovací okno v protokolu FTP

Pro přístup na webový server, kde budou umístěny vytvořené webové stránky, potřebujeme adresu webového serveru, přihlašovací jméno a heslo. Tyto informace sděluje provozovatel webového serveru po objednání webhostingu, zpravidla je pošle e-mailem.

Při práci s webovým serverem postupujeme takto:

1. Spustíme program **Tento počítač**.
2. Do adresní řádku zapíšeme adresu webového serveru včetně protokolu (například *ftp://www.server.cz*).
3. V zobrazeném okně vyplníme přihlašovací jméno a heslo.
4. Klepneme na tlačítko **Přihlásit se**.

V okně programu **Tento počítač** se zobrazí přidělená plocha na webovém serveru.

Pohodlnější práci s protokolem FTP nabízejí některé programy pro práci se soubory, například Total Commander. Z internetu si můžeme bezplatně stáhnout i programy určené přímo pro práci s protokolem FTP, například program FileZilla.

2 Tvorba kódu HTML

Webová stránka, zapsaná v kódu HTML, se skládá ze dvou částí: textů, které budou na webové stránce zobrazeny, a řídicích příkazů jazyka HTML, označovaných jako elementy (někdy se také používá název tagy).

Při tvorbě webové stránky si je třeba uvědomit, že všechny texty, které do webové stránky zapíšeme, slouží pouze jako informace pro webový prohlížeč. Způsob, jak prohlížeč v okně webové stránky zobrazí, můžeme ovlivnit jedině použitím některého elementu jazyka HTML. Při zobrazování zapsaného textu platí tato pravidla:¹

- Prohlížeč ignoruje všechny přechody na nový řádek, tedy znaky ENTER.
- Prohlížeč nezobrazí prázdné řádky.
- Mezera je chápána jako oddělovač mezi slovy, nikoliv jako znak. Několik mezer se tedy zobrazí jako jediná mezera.
- Tabulátory se interpretují jako jedna mezera.

Tyto zásady jsou pochopitelné, protože prohlížeč musí zobrazovaný text přizpůsobit počítači, na kterém se webové stránky spustí (tedy grafické kartě a rozlišení, nastaveném na monitoru). Stejný odstavec textu může na různých počítačích zabrat různý počet řádků.

Z toho ovšem plyne určitá nepříjemnost, protože odsunutí textu doprava pomocí mezer nebo tabulátorů není možné a rozmístění textu na stránce musíme provést zcela jinými prostředky.

Co se stane, pokud se při psaní kódu HTML dopustíme chyby? Většina programovacích jazyků včetně jazyka PHP se chybnému zápisu vzepře, ohlásí chybu a odmítne dále pokračovat. Webové prohlížeče však žádnou chybu kódu neohlásí a pokusí se stránku za každou cenu zobrazit. Výsledek může být ovšem poněkud podivný. Musíme si zkrátka zvyknout na to, že se při tvorbě webových stránek naplno uplatňuje stará moudrost „programátor dělá, co umí, počítač si dělá, co chce“.

2.1 Elementy jazyka HTML

Elementy jazyka HTML se uzavírají do znaků `< a >`, mezi které napíšeme identifikační kód zvoleného elementu. V jazyku HTML se rozlišují dva typy elementů:

- Párové elementy zahrnují nějaký obsah, především text. U těchto elementů je nutné vyznačit i konec jejich platnosti. Provedeme to pomocí koncového elementu, který se zapíše stejně jako počáteční element, ale před kód elementu zapíšeme lomítko.

¹ Tyto zásady neplatí zcela stoprocentně. Pomocí vhodného stylu lze zajistit, aby prohlížeč zobrazil text přesně tak, jak se zapíše do kódu webové stránky, tedy i s několika mezerami, zalomeními řádků pomocí klávesy ENTER atd. Toto je však výjimka, používaná poměrně zřídka.

- Nepárové elementy obsah nezahrnují a v dokumentu HTML stojí samostatně, bez koncového elementu.

U řady elementů je zapotřebí uvést další informace, potřebné pro jeho správnou funkci. Proto má většina elementů k dispozici řadu atributů, kterým přiřadíme potřebnou hodnotu. Provedeme to pomocí rovnítka, za kterým následuje hodnota zapsaná v uvozovkách. Atribut i s hodnotou zapíšeme dovnitř elementu mezi znaky `<` `>`, od kódu elementu jej oddělíme mezerou.

Jeden element může obsahovat i více atributů, které od sebe oddělíme opět mezerou. Jestliže element obsahuje více atributů, můžeme je napsat v libovolném pořadí. Na prvním místě mezi znaky `<` `>` však musí být vždy kód elementu.

Řada atributů má některou možnou hodnotu definovanou jako výchozí. Pokud tento atribut v elementu vynecháme, použije se tato výchozí hodnota.

Jak jsme si uvedli, webové prohlížeče ignorují všechny přechody na nový řádek. Proto můžeme elementy psát na samostatnou řádku nebo do jedné řádky společně se zahrnutým textem. Na dva řádky je možné rozdělit i samotný element. Rozdělení elementu na dva řádky je vhodné zejména v případě, kdy obsahuje více atributů a jeho text je dlouhý.

Při interpretaci kódu zapsaného jazykem HTML webové prohlížeče zpravidla nerozlišují malá a velká písmena. Pro korektní zápis kódu však platí zásada, že počáteční i koncový element by měl být napsán stejně, tedy buď malými, nebo velkými písmeny.

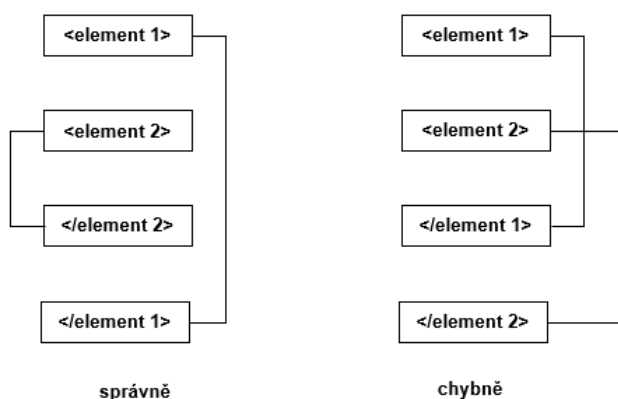


Webové prohlížeče se zpravidla dokážou vyrovnat i s nekorektnostmi v zapsaném kódu. Na to bychom však neměli hřešit. Vývoj webových prohlížečů sice směřuje na jedné straně k stále širším možnostem, ale na druhé straně se mohou zvyšovat požadavky prohlížečů na korektnost zapsaného kódu.

Obecný tvar kódu HTML s párovým a nepárovým elementem tedy vypadá takto:

```
<element1 atribut="hodnota">
Vnořený text
</element1>
<element2 atribut1="hodnota1" atribut2="hodnota2">
```

Párové elementy je možné do sebe vnořovat a vytvářet tak hierarchii do více úrovní. Přitom musíme důsledně dbát na to, aby se vnoření nekřížilo. Element, který otevřeme jako první, musíme vždy uzavřít jako poslední.



Obrázek 2.1 Vnořování elementů

Při psaní kódu HTML je vhodné dodržovat tyto zásady:

- Všechny texty v elementech píšeme malými písmeny: kód elementu, názvy atributů a použité hodnoty.
- Všechny párové elementy je nutné důsledně uzavírat.
- Všechny atributy, uvedené v elementech, by měly mít hodnotu.
- Hodnoty atributů (včetně hodnot číselných) zapisujeme do uvozovek.



Jestliže v programu **PSPad** klepneme do párového elementu, barevně se zvýrazní a stejnou barvou se zvýrazní i odpovídající koncový element (pokud už na stránce existuje). Totéž platí i při klepnutí do koncového elementu.

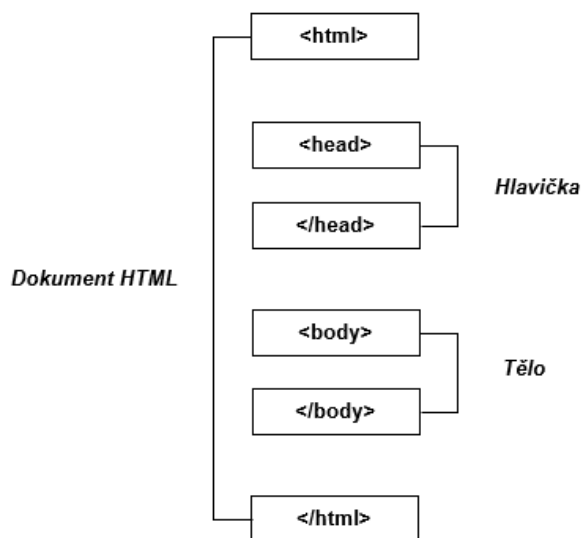
Skutečnost, že webové prohlížeče ignorují přechody na nový řádek a prázdné řádky nezobrazí, umožňuje oddělit jednotlivé části kódu stránky prázdným řádkem, takže se zvýší její přehlednost.

Pro lepší orientaci v kódu je možné použít komentář. Komentář je tvořen nepárovým elementem, jehož prvním znakem je vykřičník, poté následuje mezera a za ní libovolný text včetně české diakritiky. Příklad:

```
<! Toto je komentář k webové stránce>.
```

2.1.1 Struktura souboru HTML

Veškerý obsah webové stránky je uzavřen mezi elementy `<html>` a `</html>`. Ostatní elementy musí být do tohoto elementu vnořené. Element `<html>` může obsahovat atribut `lang`, což vyjadřuje jazyk webové stránky. Hodnotami atributu mohou být například `"cs"` (čeština) nebo `"en"` (angličtina). Tento atribut však není nutný, daleko důležitější je údaj o způsobu, jak je na stránce kódována čeština.



Obrázek 2.2 Struktura webové stránky

Mezi elementy `<html>` a `</html>` jsou vnořeny dvě sekce webové stránky:

- Hlavička stránky je uzavřena mezi elementy `<head>` a `</head>`. V této sekci jsou zapsány údaje určující způsob zobrazení webové stránky v prohlížeči (například kódování češtiny nebo definice stylů). Sekce dále může obsahovat další informace, které určují chování webové stránky a její vztah k webovým vyhledávačům.
- Tělo stránky je uzavřeno mezi elementy `<body>` a `</body>`. V těle stránky je zapsán vlastní obsah, zobrazovaný prohlížečem.

V sekci `head` jsou nejčastěji obsaženy tyto elementy:

- Párový element `<title>` a `</title>` pro zadání titulu webové stránky. Titulek se zobrazí jako nadpis v okně nebo na záložce prohlížeče.
- Elementy definující styly pro formátování webové stránky (struktura těchto elementů bude podrobně popsána v kapitole věnované stylům).
- Nepárové elementy `<meta>`, které definují další vlastnosti webové stránky.

Nejdůležitějším použitím elementu `<meta>` je určení způsobu, jak je kódována čeština na stránce. V současné době se používají především dvě možnosti:

- Kódování **windows-1250** využívají programy balíku MS Office. Webovou stránku můžeme vytvořit také v tomto kódování.
- Kódování **utf-8** se běžně používá v databázi My SQL. Toto kódování umožňuje větší výběr znaků než windows-1250 (například čínské nebo arabské znaky). Z těchto důvodů se při tvorbě webových stránek zpravidla používá kódování utf-8. Je univerzální, textový soubor však zabírá více místa.

V elementu `<meta>` určuje kódování stránky atribut `charset`, hodnotou je označení názvu kódové stránky, například:

```
<meta charset="utf-8">
```

Pokud vytvoříme novou webovou stránku v programu PSPad, vloží se do ní automaticky elementy `<html>`, `<head>` a `<body>`. Do hlavičky stránky se vloží ještě prázdný element `<title>` a dva elementy `<meta>`:

```
<meta http-equiv="content-type" content="text/html; charset=windows-1250">
```

nebo

```
<meta http-equiv="content-type" content="text/html; charset=utf-8">
```

(podle toho, jakou kódovou stránku použije program jako výchozí) a dále element:

```
<meta name="generator" content="PSPad editor, www.pspad.com">
```

První element `<meta>` můžeme ponechat ve vložené podobě nebo použít zjednodušenou verzi, uvedenou výše. Druhý element `<meta>` jen popisuje způsob, jak byla stránka vytvořena. Na funkčnost stránky nemá vliv, lze jej tedy vymazat.

Pokud potřebujeme změnit u webové stránky kódování češtiny, nestačí jen přepsat element `<meta>` v hlavičce stránky. Soubor HTML ještě musíme konvertovat do jiné kódové stránky. V programu PSPad je úprava kódování velmi jednoduchá: klepneme na příkaz **Formát**, v zobrazené nabídce označíme potřebné kódování a soubor uložíme.

Úpravu kódování můžeme provést i pomocí programu Poznámkový blok:

1. Soubor s webovou stránkou otevřeme v Poznámkovém bloku. Při otvírání je nutné použít seznam **Soubory typu** a nastavit v něm volbu **Všechny soubory**.