

1 Úvod

V tomto studijním textu se seznámíme s tvorbou webových stránek a prezentací a se základy syntaxe jazyka HTML.

K tomu, abychom byli schopni vytvořit internetovou stránku, nemusíme používat žádný drahý software. Postačíme si se základním softwarovým vybavením, které je součástí jakéhokoliv operačního systému. Popřípadě můžeme využít jakékoliv volně stažitelný nebo komerční program.

V případě, že bychom chtěli použít například poznámkový blok z MS Windows, bude práce jednoduchá, ale ne tak komfortní jako v nějakém specializovaném programu. Ten je vybaven specializovanými nástroji, které práci zjednoduší a skoro ve všech případech nám dokáže i barevně oddělit jednotlivé části programu.

2 Tvorba internetových prezentací

Internetové, neboli WWW prezentace vytváří takzvaný webmaster. Jeho funkcí není jen prezentaci vytvořit, ale také spravovat a udržovat aktualizovanou. Další jeho funkcí často bývá programátor a designér.

Při tvorbě prezentace je třeba dobře zvážit, o jak velký projekt půjde a co vše bude obsahovat. Podle toho je pak nutné zvážit, jakým způsobem se bude projekt vytvářet. Tedy zda půjde o prezentaci statickou nebo dynamickou.

Statická prezentace je vhodná pro malé projekty, u kterých není počítáno s častou aktualizací, a jejíž celý obsah bude veřejně přístupný. Na druhou stranu pro rozsáhlé projekty s velkým množstvím informací rozdělených do sekcí s rozdělením pro veřejné a soukromé informace, popřípadě se zabezpečeným obsahem, je třeba zvolit prezentaci dynamickou. U ní je také možné provádět vyhledávání, třídění a jiné vlastnosti, které závisí na rozhodnutí uživatele. Statická prezentace se tvoří jazykem HTML. Dynamická už je tvořena kombinací jazyka HTML a některého z webových programovacích jazyků (PHP, ASP, .NET a jiné). Jak již bylo řečeno, v tomto textu se budeme zabývat statickou tvorbou prezentace za pomocí jazyka HTML.

2.1 Nástroje pro tvorbu prezentace

Nyní se podívejme podrobněji na možnosti, v čem se dají prezentace vytvářet. Od základních textových editorů typu poznámkový blok, přes NonWYSIWYG až po VYSIWYG editory. Jednoduchým textovým editorem je vybaven každý operační systém. Má své limity, které nám práci nijak neznepříjemní, ale vyžaduje to už znalost jazyka HTML. Omezení může nastat v okamžiku, kdy potřebujeme provést změnu kódování textu. Jejich výhodou je, že jsou zdarma a vždy a všude po ruce.

NonWYSIWYG editor (WYSIWYG – What You See Is What You Get, to co vidíš je to, co dostaneš) je nástroj, ve kterém je už zakomponováno vývojové prostředí a nástroje pro zjednodušení práce. Může je jednat o zjednodušenou práci s odkazy a obrázky, editory barev pozadí a spoustou jiných funkcí končících u podpory skriptů a programovacích jazyků. Velké množství těchto editorů je zdarma ke stažení, popřípadě zdarma pro nekomerční účely (např. 1st Page 2000, nebo PSPad). Těmito typy editorů můžeme vytvářet prezentace jak statické tak dynamické. Vyžadují však znalost programovacího jazyka.

WYSIVYG editor je už rozsáhlý a komplexní nástroj pro tvorbu prezentace. Dají se v něm vytvářet prezentace, aniž by jejich autor měl ponětí o nějakém programování. Tyto programy pak na základě požadavku uživatele vygenerují zdrojový kód. Tento programový kód nebývá vždy nejlepší a jeho interpretace jednotlivými internetovými prohlížeči nemusí být vždy stejná. Navíc pokud nemáte hlubší znalosti takovéhoto programu, dokážete v něm vytvořit pouze statickou prezentaci.



Dalším důležitým vybavením je internetový prohlížeč. Nejlépe je provozovat několik prohlížečů současně a průběžně v nich kontrolovat své výtvory. Výhodou tohoto bodu je, že internetové prohlížeče jsou již součástí operačních systému, a pokud ne, tak jsou volně ke stažení.

Pro tvorbu grafického obsahu stránek je zapotřebí ještě mít grafický editor schopný zpracovat bitmapové obrázky. Např. Photoshop nebo volně šiřitelný GIMP.

S tímto vybavením obsahující HTML editor, internetový prohlížeč a grafický program, jsme si schopni vystačit při tvorbě statických prezentací.

3 Jazyk HTML

Už jen slova programovací jazyk můžou v některých z Vás vyvolat mírnou nejistotu. Ale není třeba se bát. Nejedná se o nic komplikovaného a v začátcích stačí znát pár základních příkazů a můžete být schopni vytvořit jednoduchou internetovou stránku. Jak již bylo řečeno, není třeba k tomu mít speciální vybavení, ale vystačíme si s tím, co je součástí operačního systému, popřípadě s nějakými volně stažitelnými programy. Na rozdíl od jiných programovacích jazyků, se jazyk HTML nemusí nijak převádět do spustitelné podoby. O jeho přeložení do podoby čitelné počítačem se stará samotný internetový prohlížeč. Z toho plyne výhoda, že kdykoliv, když budeme potřebovat něco upravit, nemusíme nějak složitě dopisovat příkazy a provádět překlad (kompilaci) programu do podoby čitelné pro počítač.

3.1 Základní značky v HTML

V jazyce HTML se pracuje s takzvanými značkami neboli tagy. Značky mohou být párové a nepárové a poznáme je v programu podle ostrých závorek <>¹. Párové značky začínají <a> a jsou ukončeny , nepárové jsou pouze
 a nemají zakončení. Samotné značky píšeme malým písmem. Psát značky malým písmem je pouze doporučení. Je pak jednodušší přepsat HTML program například do XML.

Každý HTML dokument začíná a končí značkami <html> a </html>. Tím říkáme prohlížeči, že daný dokument bude psaný jazykem HTML. Dalším krokem je rozdělení dokumentu na dvě části, hlavičku <head> </head> a tělo <body> </body>. Hlavička obsahuje značky pro název stránky, její klíčová slova a spoustu dalších. V těle stránky jsou obsaženy značky a texty, které mají co dočinění se samotným obsahem stránky. Teda tím co se má uživateli zobrazit.

Zaměřme se nesamotnou syntaxi (tedy zápis) naší první internetové stránky. V poznámkovém bloku, nebo jiném textovém editoru napíšeme:

Moje prvni stranka - Windows Internet Explorer
🔾 🗢 🙋 C:\wamp\www\pré 👻 🍕 🗙 Google
🔶 Oblíbené položky 🛛 🏉 Moje první stránka
Moje prvni stránka

Ve zdrojovém kódu přibyla nová párová značka <title>. Text uvedený mezi těmito značkami se nám zobrazí v záhlaví okna prohlížeče. A text v těle stránky se nám zobrazí v prohlížeči.

¹ <> napíšeme pomocí "pravého tlačítka alt + ,(čárka)" a " pravého tlačítka alt + .(tečka)"



3.2 Práce s textem

V této kapitole se podíváme na značky pro práci s textem a jejich parametry.

Pokud jste zkoušeli napsat do těla internetové stránky delší text, který by obsahoval formátovaný text (odstavce, nadpisy, zarovnání a jiné), na výsledné zobrazení to vliv nemělo. Jazyk HTML nedokáže zpracovat konec řádku jako textový editor. Tím, že za koncem odstavce zmáčkneme ENTER se kurzor posune na další řádek, ale na zobrazení v prohlížeči to mít vliv nebude. Tento problém nám vyřeší přidání na konec řádku nepárovou značku

br>. Tato značka prohlížeči sdělí, že se jedná o konec řádku a další text bude následovat na dalším řádku.

<html> <head> </head></html>	<title>Moje první stránka</title>
<body></body>	Moje první stránka.
	Text na dalším řádku.



Nadpisy

Neméně důležitým prvkem formátování textu je nadpis. Jeho značka je párová a vychází z anglického názvu pro nadpis heading. Pro orientaci v textu, tak i v textu generovaném pomocí HTML, máme nadpisy několika úrovní. Pro internetové stránky je definováno 6 úrovní nadpisů <h1> - <h6>, které se od sebe liší jak velikostí, tak i druhem písma v něm použitým. Nadpis je brán jako samostatný prvek v textu a proto se po něm automaticky zalamuje řádek. Není tedy nutné používat značku pro zalomení řádku
br>.

<html></html>	🏉 Moje první stránka - Windows Internet Explorer
<pre>Moje první stránka</pre>	🕞 🕞 🗢 😰 C:\wamp\www\prá 🕶 🐓 🗙 🚼 Google 🛛 🔎 🔹
	🖕 Oblíbené položky 🖉 Moje první stránka 🏼 🎽 👻 🔌
<body> <h1>Moje první stránka.</h1></body>	Moje první stránka.
Text na 1. řádku. Text na 2. řádku.	Text na 1. řádku.
<h2>Nadpis 2. úrovně</h2> Další text…	Text na 2. řádku.
<h6>Nadpis 6. úrovně</h6>	Nadpis 2. úrovně
 	Dalši text
	Nadpis 6. úrovně

Jak je z předchozího zdrojového kódu patrné, je opravdu nadpis samostatným prvkem ve formátování. Tudíž není nutné ani před ani za značkami pro nadpis dávat značku pro zalomení řádku.



Tučné, kurzíva, podtržené

Všichni jsme se setkali s nějakým textovým editorem a dobře víme, jakým způsobem jsou v něm provedeny tlačítka pro změnu typu písma na tučné - **B**, kurzíva - *I* a podtržené - <u>U</u>. Jelikož tyto zkratky mají základ v angličtině, je jisté že stejně budou vypadat i značky v jazyce HTML. Jedná se o párové značky, které lze navzájem mezi sebou kombinovat.



Pro lepší orientaci jsou značky v textu zvýrazněny barevně. A jak vyplývá z uvedeného příkladu, značky lze kombinovat nejen mezi sebou, ale i s jinými značkami pro úpravu a formátování textu. Obecně toto pravidlo lze aplikovat na všechny značky týkající se práce s textem.

Zarovnání textu

Další funkcí textového editoru, kterou lze použít i v jazyce HTML, je funkce zarovnání textu. Standardně se text zarovnává k levému okraji. Pro jiná zarovnání musíme použít funkci oddílu a na ni aplikovat patřičné zarovnání.

Funkce oddílu je definována mezi párovými značkami <div>. Nyní si budeme muset vysvětlit pojem parametr. Většina značek v HTML definuje pouze základní možnost výsledného zobrazení. Pokud chceme možnosti rozšířit, přidáme k značce její parametru hodnotu parametru. Ve výsledku to vypadá asi takto: <značka parametr1="hodnota_parametru" parametr2="hodnota_parametru">. Jak je vidět parametry se od sebe oddělují pouze mezerami a hodnoty parametru se udávají za znaménko rovná se do uvozovek.

Nyní se vraťme k oddílu a jeho parametru pro zarovnání textu. Parametr se nazývá align a jeho parametry jsou left, right, center, justify. Přeloženo do češtiny jsou to parametry pro zarovnání k levému okraji, pravému okraji, na střed a do bloku (od kraje ke kraji).



<html> <head> <title>Moje první stránka</title> </head>

<body> <div align = "center"> <h1>1 Úvod</h1> </div>

<div align = "justify">

V tomto studijním textu se seznámíme s tvorbou webových stránek a prezentací a se základy syntaxe jazyka HTML. K tomu, abychom byli schopni vytvořit internetovou stránku, nemusíme používat žádný drahý software. Postačíme si se základním softwarovým vybavením, které je součástí jakéhokoliv operačního systému. Popřípadě můžeme využít jakékoliv volně stažitelný nebo komerční program.
br>

</div> <div align = "right">

V případě, že bychom chtěli použít například poznámkový blok z MS Windows, bude práce jednoduchá, ale ne tak komfortní jako v nějakém specializovaném programu. Ten je vybaven specializovanými nástroji, které práci zjednoduší a skoro ve všech případech nám dokáže i barevně oddělit jednotlivé části programu.

</div> </body>

</html>



V příkladu je použito zarovnání nadpisu na střed stránky. První odstavec je zarovnán do bloku a druhý odstavec je zarovnán k pravému okraji. Jednotlivé bloky musí být vždy uzavřeny.

Druhým možným způsobem, jak zarovnat část textu, je místo oddílu požít párovou značku pro odstavec . Název parametru i jeho hodnota je naprosto shodná. Výhodou je však možnost použití většího množství rozšiřujících parametrů. Navíc odpadá nutnost použití značky pro zalomení řádku. Stejného výsledku jako u předchozího příkladu dostaneme i tímto způsobem, který je značně jednodušší a co se týče složitosti zdrojového kódu.



<html> <head> <title>Moje první stránka</title> </head> <body>

<h1 align = "center">1 Úvod</h1>

V tomto studijním textu se seznámíme s tvorbou webových stránek a prezentací a se základy syntaxe jazyka HTML. K tomu, abychom byli schopni vytvořit internetovou stránku, nemusíme používat žádný drahý software. Postačíme si se základním softwarovým vybavením, které je součástí jakéhokoliv operačního systému. Popřípadě můžeme využít jakékoliv volně stažitelný nebo komerční program.

V případě, že bychom chtěli použít například poznámkový blok z MS Windows, bude práce jednoduchá, ale ne tak komfortní jako v nějakém specializovaném programu. Ten je vybaven specializovanými nástroji, které práci zjednoduší a skoro ve všech případech nám dokáže i barevně oddělit jednotlivé části programu.

 </body>

</html>



Na první pohled vypadá výsledek stejně, jedinou změnou je mezera mezi samotnými odstavci.

3.3 Hypertextový odkaz

Do této chvíle jsme si ukazovali jak upravit na stránce text, popřípadě jak vůbec text na danou stránku dostat. Sami však víme, že internetová stránka není jen prostý text, ale že se v ní nacházejí i odkazy. Správně řečeno hypertextové odkazy.

Hypertextový odkaz je část textu, který slouží jako odkaz na jinou stránku. To znamená, při kliknutí myší na hypertextový odkaz, se nám načte stránka, na kterou hypertextový odkaz ukazuje. Značka pro odkaz je <a> a jedná se o párovou značku. Ta sama o sobě nemá žádný význam a musí být doplněna parametry.



Základní syntaxe značky <a> je:

odkaz

Položka nazev_souboru může obsahovat jak HTML soubor, tak i soubor s jakoukoliv jinou koncovkou. Záleží pak na celém webovém projektu, zda soubor zobrazí v prohlížeči jako další internetovou stránku, nebo jako obrázek, popřípadě nabídne daný soubor ke stažení. Jinak řečeno, pokud si internetový prohlížeč se souborem s danou koncovkou bude umět poradit, zobrazí jej. Pokud si nebude vědět rady, nabídne soubor ke stažení.

Položka odkaz pak obsahuje text, který nám internetový prohlížeč zobrazí a nabídne jej jako položku ke kliknutí.

Podívejme se, jak takový hypertextový odkaz vypadá v praxi, a s čím vším se u něj můžeme setkat.

Chceme-li, aby náš hypertextový odkaz ukazoval na stránku, která již na internetu existuje a má svou přesnou internetovou adresu, zadáme ji v přesném tvaru místo <code>nazev_souboru</code>. Například Internetový portál seznam.cz. V internetovém prohlížeči se nám zobrazí text Internetový portál seznam.cz jako hypertextový odkaz a kliknutím myši na tento odkaz nám okno prohlížeče zobrazí definovanou stránku.

```
<html>
<head>
<title>Moje první stránka</title>
</head>
<body>
Moje první stránka.<br>
<a href ="http://www.seznam.cz">Internetový portál seznam.cz</a>
</body>
</html>
```



Další využití hypertextového odkazu je k propojení mezi jednotlivými stránkami v daném webovém projektu. Zde platí několik pravidel pro vytvoření správného propojení. Jedno z nedůležitějších pravidel, které platí je: **Nikdy při vytváření hypertextových odkazů nepoužívat absolutní cesty k souboru**. Co to v praxi znamená? Máme-li rozpracovaný webový projekt uložen ve složce ve svých osobních dokumentech, vede k němu absolutní cesta, která má podobu c:\users\jmeno\Documents\slozka_projektu (popřípadě podobnou). Na první pohled se bude zdát, že vše je funkční, protože cesta k souborům existuje. Avšak v okamžiku vystavení stránek na webový server už cesta nebude platná (určitě se na serveru nebude nacházet složka dokumenty s vašim jménem). Cesta k další internetové stránce obsahuje pouze název souboru, pokud se nachází v aktuální složce. Pokud se nachází ve vnořené složce, zadáme



pouze název složky a název souboru (např.: slozka_1/soubor1.html). Je-li soubor uložen v nadřazené složce, cesta k němu má podobu ../soubor1.html. V praxi to vypadá následovně:

Odkaz na soubor - Odkaz na soubor nacházející se ve stejné složce jako soubor, ze kterého je na něj odkazováno.

Odkaz na soubor - Odkaz na soubor nacházející se ve vnořené složce s názvem slozka1 jako soubor, ze kterého je na něj odkazováno.

Odkaz na soubor - Odkaz na soubor nacházející se v nadřazené složce jako soubor, ze kterého je na něj odkazováno.

Tyto tři příklady nám ukazují nejčastěji používané typy odkazů. Není však výjimkou použití vícenásobného vnoření složky, či několika složek na stejné úrovni. K odkazování se mezi nimi pak slouží kombinace nebo sloučení výše zmiňovaných ukázek.

../../soubor.html - soubor se nachází o dvě složky výše

slozka1/slozka2/soubor.html - soubor se nachází v podložce slozka2 vnořené složky slozka1

../slozkal/soubor.html – soubor se nachází ve složce slozkal, která je na stejné úrovni jako složka ze souborem, ze kterého bylo na něj odkazováno. Jedná se však o složku s jiným názvem.

Ke značce <a> se mohou přidávat další parametry. Jeden z nejpoužívanějších je parametr target. Slouží k označení cíle, kde se má odkazovaná stránka otevřít. Obsahuje přepínače, které se odkazují na takzvané rámy, kde se má stránka otevřít. Jelikož se dnes rámy používají velice zřídka, převzaly přepínače trošku jinou funkci. Parametr target = "_blank" otevře odkaz v novém okně. Pokud použijeme přepínač s vlastním názvem, který není nikde v kódu jinak specifikován, otevře se odkaz také v novém okně, ale specifický název bude sloužit jako identifikátor tohoto okna (v praxi se můžete setkat pouze výjimečně).

Internetový portál seznam.cz - odkaz se otevře v novém okně

Další a velice hojně užívaný parametr je name. Slouží k odkazování se na části stránky a nahrazuje parametr href. Pokud máme na stránce velké množství textu, můžeme si vytvořit obsah a zněj se odkazovat například na jednotlivé kapitoly v textu.

```
<a href = "#zalozka_1">Odkaz na záložku</a>
.
.
.
<a name = "zalozka_1"></a>zde se nachází text (oblast), na kterou se odkazuje záložka
```

3.4 Obrázky

Vložit obrázek do webové prezentace je záležitost nepárového tagu s parametry. Nejlépe je si to ukázat na příkladu, jak takový obrázek vložit, a které parametry jsou, popřípadě nejsou důležité.

Povinný parametr je pouze cesta_k_obrazku. Je to cesta na místo, kde je obrázek uložen. Může se nacházet ve složce s projektem, popřípadě v podsložce. Ale také může být uložen někde na internetu a cesta k němu je potom přesná internetová adresa

(např. http://www.nazev serveru.cz/obrazek.jpg).

Nejčastější chybou pro vkládání cesty k obrázku je zadání takzvané absolutní cesty. Jedná se o přesné místo uložení a je třeba si uvědomit, že při nahrávání obsahu stránek na internetový server se již obrázek (obecně jakákoliv položka – soubor, složka) nenachází na disku Vašeho počítače ve Vaší složce například s dokumenty.



Parametry width a height udávají výšku a šířku obrázku v obrazových bodech. Má to své výhody používat tyto parametry, pokud chceme, aby se ještě před načtením samotného obrázku udělalo pro něj vyhrazené místo. Tyto parametry nejsou povinné, stejně jako parametr alt. Ten nám vytvoří popis obrázku před jeho načtením, a zároveň je schopen jej zobrazit jako text v bublině při najetí kurzoru myši na již načtený obrázek.

3.5 Tabulky

Jednou z nejsložitějších částí v HTML jsou právě tabulky. Dnes jsou sice na ústupu a nahrazují je mnohem výkonnější objekty, ale abychom pochopili další text, je dobré se jim věnovat.

Základem pro tabulku je párový tag . Ten vymezuje oblast samotné tabulky. Do něj vkládáme další párový tag
 vkládáme další párový tag
 určující řádek a mezi tyto tagy vkládáme další párovou značku pro jednotlivou buňku v řádku. Z tohoto vyplývá, že tabulku sestavujeme po jednotlivých řádcích. Omezení, které z tohoto vyplývá, každý řádek v tabulce musí mít stejný počet buněk. Jednotlivé buňky v řádcích pod sebou mají stejnou šířku. Jediná možnost jak toto obejít (za pomocí tabulek) je použít slučování jednotlivých sloupců a řádků.

Podívejme se prvně na jednoduchou tabulku a její základní parametry.

<pre></pre>	-	-	
buňka 1 - 1	buňka 1 - 1	buňka 1 - 2	buňka 1 - 3 buňka 1 - 4 buňka 1 - 5
buňka 1 – 2			
buňka 1 – 3			
buňka 1 – 4			
buňka 1 – 5	buňka 2 - 1	buňka 2 - 2	buňka 2 - 3 buňka 2 - 4 buňka 2 - 5
buňka 2 – 1	1×1	1×1 2 - 2	hužie 2, 2 hužie 2, 4 hužie 2, 6
buňka 2 - 2	Dunka 5 - 1	bunka 5 - 2	bunka 5 - 5 bunka 5 - 4 bunka 5 - 5
buňka 2 – 3	buňka 4 - 1	buňka 4 - 2	buňka 4 - 3 buňka 4 - 4 buňka 4 - 5
buňka 2 – 4			
buňka 2 – 5			
buňka 3 – 1			
buňka 3 – 2			
buňka 3 – 3			
buňka 3 – 4			
buňka 3 – 5			
buňka 4 – 1			
buňka 4 – 2			
buňka 4 – 3			
buňka 4 – 4			
buňka 4 – 5			

U této tabulky byl použit parametr border=1, který nám udává šířku okraje v obrazových bodech. Pokud parametr border nezadáme, je bráno jeho nastavení na hodnotu 0, tudíž okraje tabulky nebudou vidět. Dalším parametrem je parametr height, který nám udává výšku řádku. Tento parametr nelze použít u tagu pro buňku. Naopak zase jen pro buňku je v příkladu použit parametr width určující šířku buňky. Pro názornost je předvedeno, že nezáleží, u kterého řádku tento parametr použijeme, bude platný vždy pro celý sloupec.



Parametry height a width lze použít i pro tag , kde určují výšku a šířku celé tabulky. Pokud již nejsou zadány další parametry u jednotlivých řádků a buněk, je jejich výška a šířka přizpůsobena textu, nebo rozdělena stejně mezi řádky a sloupce.

Parametry pro slučování jednotlivých buněk jsou colspan a rowspan. Colspan nám slučuje buňky na řádku a rowspan zase slučuje buňky ve sloupci. Je si třeba ale uvědomit, že pokud buňky sloučíme, budou jejich pozice obsazeny. Uveďme na příkladu.

buňka 1 – 1
buňka 1 – 2
buňka 1 - 3
buňka 2 – 1
buňka 2 – 2
buňka 2 – 3
buňka 2 – 4
buňka 2 – 5
buňka 3 – 1
buňka 3 – 2
buňka 3 – 3
buňka 4 – 1
buňka 4 – 2

buňka 1 - 1		buňka 1 - 2	buňka 1 - 3
buňka 2 - 1 buňka 2 - 2	buňka 2 – 3		buňka 2 - 5
buňka 3 – 1	buňka 3 – 2	buňka 2 - 4	buňka 3 – 3
	buňka 4 - 1		buňka 4 – 2

Buňka 1-1 má parametr colspan=3 a tudíž zabírá místo pro 3 buňky v prvním řádku. Buňka 2-4 má parametr rowspan=3 a zabírá pozici pro další 2 buňky v následujících sloupcích a je nutno s tímto faktem počítat. Buňka 3-1 má parametr pro sloučení dvou řádků a dvou sloupců.

Dalším parametrem, který lze použít jak na celý řádek (přidáním k tagu) nebo jen na samostatnou buňku (přidáním k tagu), je parametr pro zarovnání textu horizontálně align (center, left, justify) anebo vertikálně valign (top - k hornímu okraji buňky, bottom - k dolnímu okraji buňky)

<u>Důležité upozornění:</u> každý řádek musí obsahovat stejný počet buněk. Je třeba vždy dávat veliký pozor na sloučené buňky jak v řádcích tak sloupcích. Jakákoliv chybějící buňka, či buňka navíc může znamenat chybné zobrazení celé tabulky a tím také znehodnocení celého zdrojového kódu.



3.6 Seznamy

Seznamy v HTML mají podobné vlastnosti jako seznamy například v textovém editoru. Má dvě varianty. První varianta je, seznam s odrážkami a druhá varianta je seznam číslovaný. Obě varianty jsou bloky ohraničeny párovým tagem, a v něm jsou vnořeny další párové tagy určující jednotlivé položky seznamu.

Odrážkovaný seznam

Je ohraničen párovým tagem ... a položky se nachází mezi tagy ... Takový seznam potom v praxi vypadá takto:

```
        1. položka seznamu
        2. položka seznamu
        3. položka seznamu
        4. položka seznamu

    4. položka seznamu
```

Seznamy mohou mít více úrovní. Jednotlivé úrovně se nachází jako nové, ale vnořené seznamy. Jsou odsazeny, a položky mají jiný typ odrážky. Typ odrážky je možno pomocí parametru upravit.

```
<111>
   1. položka seznamu

    1. položka seznamu

   2. položka seznamu

    2. položka seznamu

          <111>

    1. podpoložka seznamu

                 1. položka seznamu

    2. podpoložka seznamu

                 2. položka seznamu

    3. podpoložka seznamu

                 3. položka seznamu
                 4. položka seznamu

    4. podpoložka seznamu

    3. položka seznamu

   3. položka seznamu

    4. položka seznamu

   4. položka seznamu
</11]>
```

Pro oba tagy je používán parametr pro změnu odrážky - type. Pokud jej zadáme k tagu ul, změní se odrážky v celém seznamu, ale neovlivní typ odrážky v podseznamech. Když se type zadá k tagu li, změní se odrážka pouze u dané položky seznamu. Parametr type má tři hodnoty: type="disc" pro tečku, type="circle" pro kolečko a type="square" pro čtvereček.

Číslovaný seznam

Má stejné vlastnosti i možnosti jako seznam odrážkovaný. Jen místo úvodního párového tagu ... se používá tag Parametr type má v tomto případě jiné hodnoty, ale jeho použití má stejné pravidla.

```
    > položka seznamu
    > li> položka seznamu
    > položka seznamu
```

Pomocí parametru type můžeme změnit typ číslování. Primárně je nastaven na číslování arabskými číslicemi type = 1. Pomocí type = I nastavíme číslování na římské číslice. Type = i nastaví číslování na malé římské číslice. Type = A změní číslování na velká písmena a type = a je změní na malá písmena.

Druhým parametrem číslovaného seznamu je parametr value. Tento parametr nastavuje hodnotu, od kterého čísla (písmena) číslování začne. Jeho drobnou nevýhodou je, že pokud máme nastaven místo číselného seznamu seznam s písmeny, parametr value má stále hodnotu čísla odpovídajícího pořadí písmene v abecedě. Například a = 1, b = 2 atd.



19 19

INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

\check{C} íslovaný seznam s parametry type a vlaue

```
   > položka seznamu
   > položka seznamu
   > položka seznamu
   > cli>podpoložka seznamu
   > cli>podpoložka seznamu
   > value = 7>podpoložka seznamu
   > value = 5> položka seznamu
   > položka seznamu
```

- I. položka seznamu
- II. položka seznamu
 - a. podpoložka seznamu
 - b. podpoložka seznamu
 - g. podpoložka seznamu
- V. položka seznamu
- VI. položka seznamu