

# JAZYK HTML

## *aneb vytváříme vlastní webové stránky*

**WWW (World Wide Web)** je jednou ze služeb internetu. Internet je z technického hlediska tvořen miliony počítačů zapojených v sítích. Počítače v síti internet pracují jednak jako servery, jednak jako klientské stanice – servery své služby poskytují, klientské počítače tyto služby využívají. Web využívá technické struktury internetu a zprostředkovává informace uložené na těchto počítačích v atraktivní grafické podobě.

Na počítači uživatele běží klientský program, kterému se říká **prohlížeč** (browser). Jedná se o program, který dekóduje data a instrukce zaslané webovým serverem, sestaví z nich výslednou stránku jako ze stavebních cihel a tu zobrazí ve svém vlastním okně. Web je založen na principu **hypertextu**, kdy jsou jednotlivé stránky navzájem propojeny pomocí **odkazů** (linků nebo též hyperlinků). Odkazy mohou mířit na jakékoli místo na internetu vzdálené třeba tisíce kilometrů. Mezi nejpoužívanější webové prohlížeče patří například MS Internet Explorer, Mozilla Firefox, Netscape Navigator, Opera apod.

## Jazyk HTML

**Webová stránka** je dokument, který je na webovém serveru reprezentován skupinou souborů, které jsou stavebními cihlami této stránky. Webová stránka také může být vytvářena dynamicky, např. jako výsledek vyhledání nějakých údajů v databázi (typickým příkladem jsou stránky vyhledávače Google nebo stránky různých internetových zpravodajských deníků). Pro dynamické vytváření stránek se používají například tzv. *skriptovací jazyky*, jako je PHP nebo ASP.

Web používá pro popis stránky speciální jazyk, který se jmenuje *HyperText Markup Language* (značkovací jazyk pro hypertext) neboli **HTML**. Jazyk HTML se vyvíjel společně s webovými prohlížeči, které zpětně ovlivňovaly definici jazyka. HTML je základním jazykem pro vytváření webových stránek a jeho zvládnutí je nutné i pro používání složitějších skriptovacích jazyků.

- Stránka v HTML jazyce má vždy textový formát a skládá se z textu, který uživatel vidí, a **značek** (tagů, příkazů) uzavřených do lomených (špičatých) závorek (< >), které prohlížeč nezobrazuje, ale podle nich skládá a formátuje jednotlivé části stránky.
- Pokud se na stránce vyskytují binární data, jako jsou např. obrázky či animace, je na ně odkázáno pomocí příslušné značky, tj. tato data nejsou umístěna do souboru s popisem stránky, ale ve svém vlastním souboru, na který se odkazujeme.
- Značky jsou buď párové nebo nepárové. **Párová značka** slouží k formátování elementu (např. nadpisu článku) – jeho první část je před formátovaným elementem a druhá za ním, čímž je vymezena oblast, na kterou se formátování aplikuje. Příkladem párové značky je například značka pro nadpis <H1> a </H1>. Jak je vidět z příkladu, ukončovací značka je stejná jako počáteční, ale má navíc lomítko. **Nepárová značka** se vztahuje na element, který už je sám o sobě přesně vymezený (např. obrázek – značka <IMG>). Konkrétní příklady viz dále.
- Vícenásobný výskyt mezer za sebou je považován za jednu mezeru, stejně tak výskyt konců řádek, tabelátorů a dalších „netisknutelných znaků“ je také považován za jednu mezeru. Veškeré formátování textu se dělá pomocí značek (např. značkou pro zalomení řádku, odstavce atd.). Proto formátování vlastního HTML kódu slouží spíše k zpřehlednění tohoto kódu, nikoli k formátování stránky (klávesa Enter nemá žádný vliv na formátování textu).
- Zatímco pro zápis značek není důležité, jestli používáte malá či velká písmena nebo jejich kombinaci, pro názvy souborů, na které je na stránce odkazováno (např. soubory s obrázky) je pečlivé dodržování velikosti písmen nezbytné. Pokud bychom nedodrželi přesný zápis názvu souboru včetně malých či velkých písmen, nebude tento soubor serverem nalezen. Na rozdíl od Windows totiž například unixové servery rozlišují v názvech souborů velikost písmen, takže např. `Stranka.html` a `stranka.html` jsou dva různé soubory. Také je vhodné vyhnout se používání dia-

kritiky v názvech souborů, neboť odlišné kódování češtiny na různých systémech může být také zdrojem potíží. Praktické je proto používat pro veškeré názvy souborů **pouze malá písmena bez diakritiky a mezer** (místo mezery můžeme např. používat znak podtržítka \_).

Práce s HTML je mnohem jednodušší než programování v některém vyšším programovacím jazyce (Basic, Pascal, Delphi, C# apod.). Je to také velice tolerantní jazyk – pokud uděláme chybu či použijeme nesprávný příkaz, nebo pokud tento příkaz prohlížeč nezná nebo ho nedovede rozkódovat, prostě jej při zobrazování vynechá a zobrazí jen to, čemu rozumí (což může být na druhou stranu nevýhodou při hledání chyb ve stránce). I začátečník může s pomocí několika základních příkazů vytvořit dobře vypadající www stránky, které může stále zdokonalovat.

Prohlížeče nemusejí stránky vždy jen přijímat od serveru; mohou také přímo číst soubory ve formátu HTML z disku. To je výhodné pro přípravu stránky – jako základní nástroje nám pak stačí obyčejný textový editor (např. Poznámkový blok) a libovolný prohlížeč. Soubory ve formátu HTML mají obvykle příponu **.htm** nebo **.html**.

## Jak stránky připravovat

Protože je formát HTML textovým formátem, není nutné pro přípravu těchto souborů použít nějaký speciální program – stačí použít jakýkoli **textový editor**, například Poznámkový blok z Windows. v tomto případě je však nutné všechny značky psát ručně.

Druhou možností je použít tzv. **editor HTML**, což je program specializovaný na tvorbu www stránek, který obsahuje řadu pomůcek šetřících práci, např. zobrazení značek jinou barvou, automatické vkládání ukončovacích značek apod. Editorů HTML dnes existuje několik desítek (různé úrovně a kvality). Příkladem takového editoru může být např. program PSPad (freeware), Macromedia HomeSite, MS FrontPage apod. Některé editory jsou tzv. WYSIWYG (What You See Is What You Get). To znamená, že to, co vidíte na obrazovce při vytváření www stránky, bude pak také vidět v prohlížeči. Mezi tyto editory patří např. MS Word od verze 97 a MS FrontPage – takové editory tvůrce stránek úplně oprošťují od nutnosti znát HTML jazyk, na druhé straně ale produkují velmi složitý kód, který je pak jinými nástroji těžko použitelný.

## Úvod do jazyka HTML

Stejně jako celý internet i jazyk HTML se vyvíjí velice rychle. HTML utváří **konsorcium W3C** (www.w3c.com). Na jeho webové stránce naleznete vždy údaje o tom, která verze je nejnovější a také **validátor** – nástroj, který ověří syntaktickou správnost zdrojového kódu libovolné webové stránky. Nicméně při tvorbě stránek je především potřeba počítat s tím, jaké prohlížeče jsou uživateli používány – ne každý používá nejnovější verzi podporující nejnovější standardy, takže je záhodno být při tvorbě stránek poněkud konzervativnější.

Jazyk HTML má svou přesnou syntaxi (stavbu). Její pozoruhodnost a unikátnost spočívá v tom, že je sice exaktně definován, ovšem je současně velice flexibilní: i pokud se dopustíte poměrně značných chyb, lze odhalit a správně dekodovat „zdravé části kódu“.

Jazyk HTML je v **textovém formátu** a přímo v tomto formátu je používán – není nikdy kompilován (překládán) do binární nebo jiné podoby. Znamená to, že jakmile vytvoříme stránku HTML v textovém formátu, vytvořili jsme její finální tvar, který přímo čte a interpretuje prohlížeč. Soubory, které jsou v binárním formátu (obrázky, zvuky, animace apod.), nejsou umístěny do souborů HTML – v nich jsou uvedeny pouze **odkazy** na tyto soubory. Odkaz odkazuje na jiná data, která se nemusí nutně nalézat na aktuální stránce. Tedy např. „sem dej obrázek, který najdeš tam a tam“.

V současné době se prosazuje používání tzv. **stylů** (CSS – Cascade Style Sheets). Principem je oddělení formátovací informace od textu tak, že se místo přímého zápisu údajů např. o velikosti a barvě písma do HTML kódu uvede pouze odkaz na pojmenovaný styl, který je definovaný v samostatném souboru (je tu jistá analogie ke stylům ve Wordu). Používání CSS přináší celou řadu výhod, např. změnou stylu je možné změnit vzhled mnoha stránek najednou. Pro začátečníky v HTML jazyce je však používání stylů příliš složité, je lépe nejprve zvládnout základy HTML jazyka a pak se věnovat stylům. Proto se v této publikaci stylům nebudeme věnovat.

## Struktura dokumentu – základní části stránky

HTML soubor vždy obsahuje popis webové stránky nebo její části. Stránkou je míněna plocha s obsahem, která se zobrazí v okně prohlížeče. Základní struktura webové stránky je následující:

- <HTML>** Počátek zdrojového textu ve formátu HTML.
- <HEAD>** Počátek definice hlavičky. Vedle názvu stránky obsahuje hlavička další důležité informace vztahující se k celému dokumentu, zejména v jakém kódování češtiny je stránka napsána, klíčová slova, odkaz na soubor s popisem stylů apod., například:

```
<meta http-equiv="content-type" content="text/html"
  charset="windows-1250">
<meta name="description" content="Jak na jazyk HTML">
<meta name="keywords" content="html, značka, tag, atribut">
```

**<TITLE>Jméno dokumentu</TITLE>**

Text, který se objeví v titulkovém pruhu (bude použit např. v seznamu záložek, když si někdo dotyčnou stránku založí jako oblíbenou) – měl by být výstižný a srozumitelný a neměl by překročit 64 znaků.

**</HEAD>** Konec definice hlavičky.

**<BODY>** Počátek definice těla, vlastního zobrazovaného obsahu stránky.

**...tělo dokumentu...**

Vlastní zobrazovaný obsah.

**</BODY>** Konec definice těla stránky.

**</HTML>** Konec zdrojového textu HTML.

Např.:

```
<HTML>
  <HEAD>
    <TITLE>
      Soukromá stránka Pepka námořníka
    </TITLE>
  </HEAD>

  <BODY>
    <H1>
      Já, Pepek námořník
    </H1>
    <P>
      Jmenuji se Pepek, jsem námořník a jsem velmi silný, protože jím špenát. A každé malé dítě ví, že špenát je nejen velmi zdravý, ale že i skvěle chutná.<BR>
      Moje přítelkyně se jmenuje Olie a mým největším rivalem je Brutus.
    </P>
    <IMG SRC="pepek.jpg">
    <P>
      A kdo nemá rád Pepka námořníka, může se podívat třeba na
      <A HREF="http://www.hampsterdance.com">tancující křečky</A>.
    </P>
  </BODY>
</HTML>
```

Abychom se při ladění stránky ve zdrojovém kódu vyznali, musíme dodržovat základní formátování zdrojového textu, aby bylo na první pohled zřejmé, na jakou část kódu budou mít ohraničující značky vliv. Stačí k tomu pouhé odsazování textu od kraje (tabelátorem nebo mezerami) a vynechání řádku mezi jednotlivými částmi HTML kódu (entrem).



A takto vypadá kód trochu vylepšené stránky:

```

<HTML>
<HEAD>
  <TITLE>
    Soukromá stránka Pepka námořníka
  </TITLE>
</HEAD>
<BODY BGCOLOR="yellow" TEXT="black" LINK="blue" VLINK="purple">
  <H1 ALIGN="center">
    Já, Pepek námořník
  </H1>
  <P ALIGN="justify"><FONT SIZE="+1" COLOR="green">
    Jmenuji se Pepek, jsem námořník
    a jsem velmi silný, protože jím špenát.
    a každé malé dítě ví, že špenát je nejen
    velmi zdravý, ale že i skvěle chutná.<BR>
    Moje přítelkyně se jmenuje Olie
    a mým největším rivalem je Brutus.
  </FONT></P>
  <IMG SRC="pepek.jpg" ALIGN="middle" WIDTH="282" HEIGHT="200"
    BORDER="2" ALT="Pepek námořník">
  <P ALIGN="center">
    A kdo nemá rád Pepka námořníka, může se
    podívat třeba na
    <A HREF="http://www.hampsterdance.com">tancující
    křečky</A>.
  </P>
</BODY>
</HTML>

```

Chceme-li do textu umístit *poznámku nebo komentář*, který se nebude zobrazovat ani žádným způsobem neovlivní zobrazení či zpracování, musíme na začátek poznámky vložit posloupnost znaků `<!--` a na konec `-->`. Prohlížeč takto označenou část textu prostě ignoruje, jako kdyby tam vůbec nebyla. Například:

```
<!-- Toto je nějaká poznámka či komentář -->
```

## Tělo dokumentu

Značka **BODY** je párová. Uzavírá do sebe celé vlastní tělo stránky a za jeho ukončovací polovinou (`</BODY>`) se už nalézá úplný konec HTML souboru, tedy uzavírací značka `</HTML>`. Protože stránka má několik vlastností, které se týkají jí celé (jako např. barva pozadí stránky, barva textu, barva odkazu apod.), uvádějí se tyto přímo ve značce **BODY** jako *parametry* neboli *atributy* (jednotlivé atributy se oddělují mezerami, hodnota atributu se umísťuje do uvozovek). Tyto atributy jsou nepovinné – pokud nebudou uvedeny, použije se výchozí nastavení internetového prohlížeče. Syntaxe značky **BODY** je následující:

```
<BODY BACKGROUND="..." BGCOLOR="..." TEXT="..." LINK="..." VLINK="...">
...obsah stránky...
</BODY>
```

**BACKGROUND** je atribut určující obrázek na pozadí. Ten obsahuje jméno obrázku, který vyplní pozadí stránky (ve formátu GIF, JPG nebo PNG).

Např.: `<BODY BACKGROUND="pozadi.gif">`

Obrázky na pozadí mohou být pouze bitmapové, a mají tudíž přesně danou výšku a šířku v pixelech. Obrázek se tedy vyskládá na plochu hlavního okna prohlížeče ve formě dlaždic v řádcích a sloupcích tak, aby ji celou vyplnil.

Dalšími atributy značky **BODY** například jsou:

**BGCOLOR** barva vyplňující pozadí stránky (background color, implicitně obvykle bílá)

**TEXT** základní barva textu (implicitně je obvykle černá – je to dáno nastavením prohlížeče)

**LINK** barva ještě nenavštíveného odkazu (odkaz je zároveň podtržený), implicitně je obvykle modrá (hex. 0000FF) – měla by být výraznější než barva navštíveného odkazu

**VLINK** barva již navštíveného odkazu (visited link), implicitně je purpurová (hex. 800080)

Hodnoty barev je možné zadávat dvěma způsoby: jednak existuje paleta tzv. pojmenovaných barev, jednak je možné použít hexadecimální označení, kterým lze popsat každou z pravých barev, tj. každou barvu z dostupné palety 16,7 mil. barev. Každá taková barva se dá zapsat jako 3 hodnoty: červená, zelená a modrá (Red, Green, Blue = RGB) v rozsahu 0 až 255. z praktických důvodů se používá hexadecimální zápis, tj. 0 až FF. Takže například barva #FF0000 bude obsahovat plnou červenou a žádnou zelenou ani modrou. Barva #FFFF00 míchá aditivně plnou červenou a plnou zelenou (a žádnou modrou), což dává dohromady žlutou (pozor – neplést si se subtraktivním mícháním barev na papíře, kde smícháním všech barev vznikne černá, tady smícháním všech barev #FFFFFF vznikne bílá).

Např.: `TEXT="White"` nebo

`TEXT="#024AC9"` (nezapomeňte na znak hashmark # před číslem barvy – PrALT+X nebo ALT+35)

## Tabulka základních pojmenovaných barev

<b>aqua</b>	#00FFFF	vodní modř	<b>navy</b>	#000080	tmavě modrá
<b>black</b>	#000000	černá	<b>olive</b>	#808000	olivově zelená
<b>blue</b>	#0000FF	modrá	<b>purple</b>	#800080	purpurová
<b>fuchsia</b>	#FF00FF	fialová	<b>red</b>	#FF0000	červená
<b>gray</b>	#808080	šedá	<b>silver</b>	#C0C0C0	stříbrná
<b>green</b>	#008000	zelená	<b>teal</b>	#008080	světle zelená
<b>lime</b>	#00FF00	jasně zelená	<b>yellow</b>	#FFFF00	žlutá
<b>maroon</b>	#800000	kaštanově červená	<b>white</b>	#FFFFFF	bílá

## Formátování textu

---

Text, který je umístěn do sekce BODY bez jakýchkoli atributů, je na obrazovce prohlížeče vykreslen tzv. *základním písmem* (base font), což je obvykle patkové písmo typu Times (Times New Roman) v základním řezu. Typ písma, jeho velikost a řez je v prohlížeči nastavitelný. Počet mezer mezi slovy nemá význam, protože jsou vždy komprimovány na jednu. Formátování písma je řešeno několika značkami, které jsou vždy párové, ohraničují formátovaný text zleva a zprava. Tyto značky se mohou vnořovat.

## Nadpis

---

**Nadpisy** jsou v běžném textu nejčastěji využívaným elementem. Základní syntaxe všech značek pro nadpisy je stejná:

`<Hx>`Zde je znění nadpisu`</Hx>`kde **Hx** může být **H1** až **H6**

Základní formátování nadpisu (dané nastavením konkrétního prohlížeče) určuje, že se jedná o text napsaný tučným písmem, zarovnaným doleva, vytvořený velkým fontem. Nadpis je sám o sobě odstavcem. Mezi jednotlivými značkami pro nadpisy je rozdíl pouze ve velikosti, která plynule klesá od H1 (největší) k H6 (nejmenší). V praxi se nejčastěji používají H1 až H3.

Zarovnání nadpisu je řízeno atributem **ALIGN**

`ALIGN="left"` zarovná nadpis doleva  
`ALIGN="right"` zarovná nadpis doprava (na pravý okraj stránky)  
`ALIGN="center"` vystředí nadpis  
`ALIGN="justify"` nadpis je zarovnaný na oba okraje (do bloku)

Př.: `<H1 ALIGN="center">`Hlavní nadpis je vystředěn`</H1>`  
`<H2 ALIGN="right">`Druhý nadpis je zarovnan doprava`</H2>`

## Text a jeho základní formátování

---

**<BR>** Ukončuje řádek (zalomení konce řádku, nucený skok na nový řádek, line Break). Rozdělení textu na řádky zajišťuje klient automaticky podle momentální šířky okna, ve kterém je dokument zobrazen. Pokud chceme některý z řádků ukončit dříve, máme k dispozici tuto značku.

Např.:

<pre>&lt;BODY&gt;   Takto&lt;BR&gt;   vypadá&lt;BR&gt;   odřádkovaný&lt;BR&gt;   text. &lt;/BODY&gt;</pre>	Takto vypadá odřádkovaný text.
--	---

**<DIV>** Odděluje odstavce – přesněji zahajuje nový odstavec, tj. provede odskok na nový řádek a navíc mírné odsazení před i za odstavcem, je párový. Syntaxe je:

`<DIV>`Zde je nějaký text.`</DIV>`

Zarovnání odstavce se provádí pomocí atributu **ALIGN**, jehož hodnoty jsou stejné jako u značky pro nadpis.

Např.:

`<DIV ALIGN="center">`Jedna, dva, tři, my jsme bratři`</DIV>`

**<P>** Též odděluje odstavce (Paragraph) – zahajuje nový odstavec, tj. provede odskok na nový řádek a navíc mírné odsazení před i za odstavcem, je párový.

Syntaxe je:

`<P>`Zde je nějaký text.`</P>`

Zarovnání odstavce se provádí pomocí atributu **ALIGN**, jehož hodnoty jsou stejné jako u značky pro nadpis.

**<FONT>** Tato značka se používá tam, kde nestačí standardní formátování. Definiuje jednak druh použitého písma (pozor na dostupnost daného písma na straně uživatele – musí ho mít nainstalované ve svém počítači), jeho relativní velikost a barvu.

Syntaxe je následující:

```
<FONT SIZE="velikost" FACE="druh písma" COLOR="barva">Daný text</FONT>
```

Atribut **SIZE** určuje velikost písma – absolutní (může mít hodnotu 1-7) nebo relativní zvětšení nebo zmenšení oproti základní velikosti (BASEFONT). Základní písmo se zvětšuje použitím znaménka plus +, zmenšuje se pomocí znaménka mínus -.

```
<FONT SIZE="+2">Písmo zvětšené o dvě úrovně.</FONT>
```

Atribut **COLOR** určuje barvu písma. Je možné používat jména barev i jejich hexadecimální vyjádření.

Např.: 

```
<FONT COLOR="White">Bílé písmo</FONT>
```

Atribut **FACE** uvádí písmo, kterým bude text znázorněn. Umožňuje zadat několik názvů písem, která jsou pak postupně vyhledávána. Obsahuje-li název písma mezeru, uzavřeme jej do apostrofů (').

Např.: 

```
<FONT FACE="Verdana, 'Arial CE', Helvetica">Text</FONT>
```

**<BASEFONT>** Definiuje velikost základního písma na stránce, tj. toho písma, které není formátováno jinak. Jeho syntaxe je:

```
<BASEFONT SIZE="n"> kde n je velikost písma v 7 úrovních, od 1 do 7
```

**<BLOCKQUOTE>** Provede odsazení citace.

Např.:

```
<BLOCKQUOTE>
  <DIV>
    „Když už člověk jednou je, tak má koukat, aby byl. a když
    kouká, aby byl, a je, tak má být to, co je, a nemá být to,
    co není, jak tomu v mnoha případech je.“
  </DIV>
  <DIV ALIGN="right">
    „Jan Werich“
  </DIV>
</BLOCKQUOTE>
```

## Značky pro změnu písma

**<EM>** kurzíva (emphasize, dříve **<I>** italic)  
**<STRONG>** tučné písmo (dříve **<B>** bold)  
**<CODE>** neproporcionální písmo (psací stroj), ukázka kódu **<TT>**  
**<STRIKE>** přeškrtnuté písmo  
**<SUB>** dolní index (subscript)  
**<SUP>** horní index (superscript)  
**<BIG>** velké písmo (zvětšit o stupeň)  
**<SMALL>** malé písmo (zmenšit o stupeň)

Značky jsou vždy párové.

Např.:

Slovo **<I>**kurzíva**</I>** bude kurzívou, slovo **<B>**tučný**</B>** bude napsáno tučně a tento text **<I>**bude **<B>**všemi**</B>** kombinacemi**</I>** současně.

Na obrazovce to pak bude vypadat takto:

Slovo *kurzíva* bude kurzívou, slovo **tučný** bude napsáno tučně a tento text *bude všemi kombinacemi* současně.

## Symboly pro speciální znaky

Pokud chceme do běžného textu umístit tzv. *rezervované znaky*, tj. znaky, které mají v HTML jazyce speciální význam, vložíme je tam pomocí příkazů zvaných symboly.

<b>&amp;lt;</b>	znak < (lt=less than)	<b>&amp;laquo;</b>	znak «
<b>&amp;gt;</b>	znak > (gt = greater than)	<b>&amp;raquo;</b>	znak »
<b>&amp;amp;</b>	znak & (amp = ampersand)	<b>&amp;deg;</b>	znak stupeň (deg = degree – např. °C)
<b>&amp;nbsp;</b>	pevná neboli <b>nedělitelná mezera</b> (nbsp = non breaking space). Slova spojená nedělitelnou mezerou se chovají jako jedno slovo a nedojde tedy k jejich automatickému oddělení na konci řádku.		

Stejným způsobem lze do textu vložit také znaky z jiných abeced, které nejsou v daném kódování k dispozici. Například španělské ñ (n s tildou) se píše **&ntilde;**.

## Seznamy a výčty

**<UL>** *Nečíslovaný seznam* (netříděný, neuspořádaný – Unordered List). Tato párová značka do sebe uzavírá výčet položek, kdy je na každá položka odsazena doprava a označena grafickým symbolem. Lze vytvářet i vnořené seznamy. Syntaxe je:

```
<UL>
  <LI>První položka seznamu</LI>
  <LI>Druhá položka seznamu</LI>
  <LI>Třetí položka seznamu</LI>
</UL>
```

Atribut **TYPE** ovlivňuje vzhled grafické značky: "**disc**" (tečka), "**circle**" (kolečko), "**square**" (čtvereček)

**<LI>** je položka seznamu (List Item)

Např.:

```
Dny v týdnu:
<UL>
  <LI>Pondělí</LI>
  <LI>Úterý</LI>
  <LI>Středa</LI> ...
</UL>
```

Dny v týdnu:

- Pondělí
- Úterý
- Středa ...

**<OL>** *Číslovaný seznam* (tříděný, uspořádaný – Ordered List) – použití je obdobné jako u značky UL. Atribut **TYPE** udává typ číslování seznamu:

TYPE="A"	číslování výčtu pomocí velkých písmen, tj. A, B, C, ...
TYPE="a"	číslování výčtu pomocí malých písmen, tj. a, b, c, ...
TYPE="I"	číslování výčtu pomocí velkých římských číslic, tj. I, II, III, ...
TYPE="i"	číslování výčtu pomocí malých římských číslic, tj. i, ii, iii, ...
TYPE="1"	číslování výčtu pomocí arabských číslic, tj. 1, 2, 3, ...

Pomocí kombinace různých typů seznamů můžeme vytvářet vnořené (víceúrovňové) seznamy.

Např.:

```
Nákup:
<OL>
  <LI>Vino</LI>
  <UL>
    <LI>červené</LI>
    <LI>bílé</LI>
  </UL>
  <LI>Pivo</LI>
  <UL>
    <LI>světlé</LI>
    <LI>tmavé</LI>
  </UL>
</OL>
```

Nákup:

1. Víno
  - červené
  - bílé
2. Pivo
  - světlé
  - tmavé



## Vodorovná linka

---

**<HR>** Vodorovná linka (Horizontal Rule) se používá pro výrazné oddělení dvou částí textu (doporučuje se před i za vložit značku **<P>**, aby text nebyl nalepen přímo na lince. Linka se vykresluje přes celou šířku okna s určitým stejným odsazením zleva i zprava. Tato značka může mít několik atributů:

**SIZE="n"**, kde *n* je tloušťka linky v pixelech

**WIDTH="n"**, kde *n* je šířka linky buď v pixelech (přirozené číslo) nebo v procentech šířky okna (číslo se znakem procent). Implicitně je linka kreslena přes celou šířku okna.

**ALIGN="left"**, "right" nebo "center" provede zarovnání linky k levému, pravému okraji okna či na střed okna. Implicitní zarovnání je na levý okraj.

**COLOR="jméno"** či hex. číslo, barva linky (implicitně je šedá)

**NOSHADE** vypne stín pod linkou (implicitně je zapnutý)

Např.:

```
<HR ALIGN="left" SIZE="1" NOSHADE>
```

```
<HR SIZE="2" COLOR="green" WIDTH="50%" NOSHADE>
```

## Odkazy

---

Odkaz definuje prvek na stránce, který přímo vede k prvku jinému (zpravidla na jiné stránce) – vytváří mezi nimi vazbu. Prvek, který je odkazem, je graficky zvýrazněn (implicitně je text modrý a modře podtržen, obrázek je modře orámován), vazba je dána v definici tohoto prvku a realizuje se ťuknutím myši na tento prvek.

**<A>** Základní značkou odkazující na jiný element je kotva (Anchor). Jeho syntaxe je:

```
<A HREF="externí odkaz" NAME="interní odkaz" TITLE="jméno"
TARGET="jméno cíle">Zobrazený text</A>
```

**HREF** znamená odkaz na jiný soubor (Hyperlink REFERENCE), externí soubor obecně kdekoli na světě – může se nalézat ve stejném adresáři jako je otevřený dokument, v jiném adresáři na totéž serveru nebo na jiném serveru na druhém konci světa.

Odkaz může být umístěn uvnitř věty nebo může tvořit samostatnou větu či několik vět (což ale není účelné). Pokud odkaz míří na soubor na lokálním počítači, stačí pouze specifikace umístění souboru.

Např.:

Chcete-li se od srdce zasmát, podívejte se na

```
<A HREF="C:\humor\kresleny\vtip.gif">kreslený vtíp</A>, který se
vám určitě bude líbit.
```

Pokud je na serveru kdekoli na světě, je nutné zadat jeho univerzální internetovou adresu (URL – Universal Resource Locator).

Např.:

```
Chcete-li radostí slzet a smíchy se za břicho popadat, podívejte
se na <A HREF="http://websmile.flamenco.cz">Pavučinový úsměv-
ník</A>, který se jistě stane vašimi oblíbenými stránkami.
```

Pokud zadáváte jméno souboru v odkazu, musí přesně odpovídat realitě – udáte-li například *zprava.htm* a hledaný soubor se jmenuje *zprava.html*, prohlížeč jej nenalezne (obě přípony jsou totiž přípustné vzhledem k tomu, že některé systémy nezvládají dlouhá jména souborů). Stejně tak je potřeba naprosto přesně psát malá a velká písmena v názvu souboru. Pokud používáme odkaz na externí soubor, měli bychom se přesvědčit, zda tento soubor existuje, zdali bude výsledný odkaz platný.

Odkaz na jiný dokument může být nejen ve formě textu, ale také například ve formě obrázku.

Např.:

```
<A HREF="zvetseny_obrazek.jpg"><IMG SRC="obrazek.jpg"></A>
```

```
<A HREF="stranka.html"><IMG SRC="obrazek.jpg"></A>
```

Vedle odkazu na další stránky můžeme vytvářet být odkazy i na další typy souborů (např. DOC, PDF, ZIP apod.). Prohlížeč je buď umí zobrazit přímo ve svém okně nebo nabídne jejich uložení na disk. Pro odkaz na jakýkoli soubor ležící ve stejné složce jako daná stránka (viz níže) můžeme například použít zápis

```
<A HREF="jméno souboru">Libovolný text</A>
```

Např.:

```
Studijní materiál<A HREF="texty.doc">Textové editory</A>
```

Pokud bychom chtěli spustit klienta elektronické pošty přímo z webové stránky, použijeme následující zápis odkazu

```
<A HREF="mailto:e-mail">text, např. přímo elektronická adresa</A>
```

Např.:

```
<A HREF="mailto:simona@atlas.cz">simona@atlas.cz</A>
```

```
<A HREF="mailto:simona@atlas.cz">Napište nám</A>
```

**NAME** Pokud chceme odkaz na jiné místo aktuálního dokumentu, musíme na tomto místě vytvořit zarážku danou deklarací A NAME, tedy jakési *jméno daného místa*. V odkazu A HREF i A NAME se pak použije před *jménem místa* znak #. Odkazy na jiná místa téhož dokumentu se používají pro větší přehlednost a snazší orientaci v rozsáhlejších dokumentu, který není vhodné např. z důvodu provázanosti a sledu informací rozdělit na samostatné stránky a propojit je pomocí odkazů. Na začátku dokumentu tak můžeme vytvořit jakýsi obsah sestávající z odkazů, které nás po ťuknutí myši přenesou na zadané místo.

Např.:

Dejme tonu, že existuje dokument rozdělený do následujících celkem obsáhlých částí.

```
<BODY>
```

```
<H2>Jazyk HTML</H2>
```

```
1. <A HREF="#kapitola1">Formátování textu</A><BR>
```

```
2. <A HREF="#kapitola2">Seznamy</A><BR>
```

```
3. <A HREF="#kapitola3">Odkazy</A><BR>
```

```
4. <A HREF="#kapitola4">Obrázky</A><BR>
```

```
5. <A HREF="#kapitola5">Tabulky</A><BR>
```

```
...
```

```
<A NAME="kapitola1"></A>
```

```
<P>1. Formátování textu</P>
```

```
...
```

```
<A NAME="kapitola2"></A>
```

```
<P>2. Seznamy</P>
```

```
...
```

```
<A NAME="kapitola3"></A>
```

```
<P>3. Odkazy</P>
```

```
...
```

```
<A NAME="kapitola4"></A>
```

```
<P>4. Obrázky</P>
```

```
...
```

```
<A NAME="kapitola5"></A>
```

```
<P>5. Tabulky</P>
```

```
</BODY>
```

## Jazyk HTML

1. Formátování textu

2. Seznamy

3. Odkazy

4. Obrázky

5. Tabulky

...

1. Formátování textu

...

2. Seznamy

...

3. Odkazy

...

4. Obrázky

...

5. Tabulky

**TITLE** – tento atribut uvádí alternativní název odkazu, který se zobrazí, podržíte-li nad odkazem vteřinku myš. Obsahuje tedy doplňkové nebo alternativní vysvětlení odkazu.

**TARGET** – při ťuknutí na odkaz otevře automaticky nové okno prohlížeče místo toho, aby se stránka načetla do existujícího otevřeného okna prohlížeče.

Např.:

Odkaz na `<A HREF=http://www.atlas.cz TARGET="_blank">Atlas</A>` se otevře v novém okně.

Pokud odkaz vede na další stránku nalézající se na stejném serveru, je vhodné používat tzv. *relativní cesty*. Například:

Stránka se jmenuje *index.html* a nalézá se na serveru *www.mgplzen.cz* (v kořenovém adresáři), takže její plná specifikace se dá zapsat:

```
http://www.mgplzen.cz/index.html
```

Předpokládejme, že z této stránky vede odkaz na jinou stránku (např. *novinky.html*), která je též umístěna v kořenovém adresáři na *www.mgplzen.cz*. Plná specifikace odkazu by se dala zapsat:

```
<A HREF="http://www.mgplzen.cz/novinky.html">...</A>
```

Je však lépe do HTML kódu zapsat odkaz takto:

```
<A HREF="novinky.html">...</A>
```

Pokud je specifikace serveru vynechána, doplní se automaticky stejná, jaká je u odkazující stránky.

Předpokládejme dále, že z této stránky vede odkaz na obrázek *budova.jpg*, který se nalézá ve složce *obrazky*. Plná specifikace odkazu by se dala zapsat:

```
<IMG SRC="http://www.mgplzen.cz/obrazky/budova.jpg">
```

Namísto této formy však stačí napsat:

```
<IMG SRC="/obrazky/budova.jpg">
```

nebo

```
<IMG SRC="obrazky/budova.jpg">
```

Rozdíl mezi prvním a druhým způsobem (všimněte si chybějícího lomítka v druhém případě) je ten, že v prvním případě se jedná o tzv. *absolutní specifikaci* v rámci daného serveru, tj. takto se může na obrázek odkázat stránka z kterékoliv podsložky na serveru, zatímco v druhém případě jde o *relativní specifikaci*, tj. složka *obrazky* bude hledána ve složce, ve které se nachází odkazující stránka. Ačkoliv se druhá varianta zdá být méně výhodná než první, ve skutečnosti je tento druhý způsob zápisu nejvýhodnější pro možné přenášení a zejména pro možnost testování stránek na lokálním disku.

## Umístování odkazů do textu

- Tvořte text tak, jako kdyby v něm žádné odkazy nebyly. Odkazy jsou čitelné již modrou barvou a podtržením, není zapotřebí na ně v textu poukazovat. Odkaz by měl být přirozenou součástí věty, která jej obklopuje. Do odkazu též nezařazujeme interpunkční znaménka a okolní mezery.

Např.:

Ve Smrkové ulici se nalézá Muzeum orientálního umění, které založil známý orientalista Jan Bouzek.

Nikoli:

Ve Smrkové ulici naleznete Muzeum orientálního umění, jehož domovská stránka je zde. Kliknete-li sem, můžete poslat e-mail jejímu tvůrci, orientalistovi Janu Bouzkovi.

- Odkazy neplýtvajte a myslte na jejich vhodnou délku. Odkaz by se neměl vléci přes mnoho slov, protože podtržení a jiná barva jen zhoršuje čitelnost.

Např.:

Americká firma WebResources, Inc. z Colorada udržuje na svých stránkách velmi obsáhlou kolekci webových grafiky, použitelných appletů a pomůcek pro práci s internetem.

Lépe:

Americká firma WebResources, Inc. z Colorada udržuje na svých stránkách velmi obsáhlou kolekci internetových materiálů: webové grafiky, použitelných appletů a pomůcek pro práci s Internetem.

- Odkaz je současně zvýrazněním, proto se vždy zamyslete nad jeho vhodností.

Např.:

Dále naleznete:

- studijní materiály pro I. ročník
- studijní materiály pro II. ročník
- studijní materiály pro III. ročník

Lépe:

Dále naleznete studijní materiály pro I. ročník, pro II. ročník a pro III. ročník

- U dlouhých dokumentů přetékajících přes několik stránek hrozí, že čtenář ztratí orientaci. Udělejte na začátku takového dokumentu obsah (nadpisy jednotlivých částí jako kapitoly) a proveďte jeho provázání pomocí odkazů. Stránkujte dokument také například oddělovacími linkami s tím, že na začátku každé „stránky“ je uveden skok na začátek, může zde být i číslování.
- Dobře nazvaný textový odkaz dává uživateli jasnou informaci o stránce, na kterou vede, a pomáhá mu nalézt požadovanou informaci.

## Obrázky

---

Abychom mohli zařadit obrázek do webového dokumentu, musíme jej převést do bitmapového formátu GIF, JPG nebo PNG (jiné bitmapové formáty a vektorové obrázky neumí současné prohlížeče samy o sobě zobrazovat). Každý z těchto formátů má své výhody i nevýhody:

**GIF** má omezený počet barev (max. 256), dobře však zvládá obrázky s ostrými přechody barev (nápis, loga, kresby, černobílé fotografie), má efektivní bezztrátovou komprimaci výše zmíněných obrázků (monotónní plochy zabírají minimum místa), je velmi dobrý pro malé obrázky, umožňuje animaci, prokládání a průhlednost. Nehodí se pro fotorealistické obrázky (především plynulé přechody barev). Obdobně formát **PNG** – bezztrátová komprimace, průhlednost, neumožňuje animaci, ale zato disponuje plnými barvami.

**JPG** je výborný pro fotografie a obrázky podobného charakteru (plynulé přechody barev), zachovává plnou barevnou informaci (24 bitů – 16,7 mil. barev), použitá komprimace je ztrátová – nastavená kvalita obrázku je přímo spojená s velikostí obrázku (čím větší komprimace obrázku, tím menší velikost souboru, ale také tím horší kvalita obrázku). Nehodí se pro kresby a nápis, není příliš efektivní pro souvislé jednobarevné plochy a malé obrázky, neumožňuje průhlednost ani animaci.

**<IMG>** je jediná univerzální značka pro umístování obrázků (IMaGe). Základní syntaxe je:

```
<IMG SRC="jméno souboru">
```

Značka je nepárová, pro soubor zadáváme celou adresářovou cestu popř. URL

Např.:

```
<IMG SRC="obr1.gif">
```

Obrázek je umístěn přímo v adresáři (složce), ve kterém je i aktuální dokument.

```
<IMG SRC="../obrazky/obr2.gif">
```

Obrázek je umístěn ve složce *obrazky* včetně celé adresářové cesty (symbol .. znamená návrat do nadřazené složky).


```
<IMG SRC="http://www.cpress.cz/obrazky/obr3.gif">
```

Obrázek je zadán s celou URL i adresářovou adresou.


Značka IMG má řadu upřesňujících atributů. Prvním z nich je zarovnání obrázku vzhledem k okolnímu textu pomocí atributu **ALIGN**:

- "left"** obrázek je umístěn na levý okraj okna a text jej obtéká bezprostředně zprava
- "right"** obrázek je umístěn na pravý okraj okna a text jej obtéká bezprostředně zleva
- "top"** zrovná horní okraj obrázku s nejvyšším elementem v daném řádku
- "texttop"** zrovná horní okraj obrázku s nejvyšším písmenem v daném řádku (obvykle se rovná atributu „top“)
- "middle"** zrovná základnu daného řádku textu s prostředkem obrázku
- "absmiddle"** zrovná prostředek daného řádku textu s prostředkem obrázku
- "baseline"** zrovná spodní okraj obrázku se základnou daného textu
- "bottom"** stejné jako "baseline"
- "absbottom"** zrovná spodní okraj obrázku s nejspodnější linkou textu


Implicitním zarovnáním je volba „bottom“. Pokud chcete, aby byl obrázek plynule shora obtékán textem, používá se volba „top“ nebo „left“.


	Jmenuji se Pepek, jsem námořník a jsem velmi silný, protože jím špenát. a každé malé dítě ví, že špenát je nejen velmi zdravý, ale že i skvěle chutná.
---	--


...zarovnání "left"


Jmenuji se Pepek, jsem námořník a jsem velmi silný, protože jím špenát. a každé malé dítě ví, že špenát je nejen velmi zdravý, ale že i skvěle chutná.	
--	--


...zarovnání "right"

	Toto je zarovnání "top" – horní okraj obrázku je zarovnán s nejvyšším písmenem v tomto řádku a dále pokračuje pod obrázkem zleva.
---	---

	Toto je zarovnání "middle" – zrovná linku, na níž je posazen text, na střed daného obrázku, a dále pokračuje pod obrázkem zleva.
---	--

	Toto je zarovnání "absmiddle" – zrovná prostředek daného řádku na střed obrázku, a dále pokračuje pod obrázkem zleva.
---	---

	Toto je zarovnání "bottom" – spodní okraj obrázku je zarovnán s linkou, na níž je posazen text, a dále pokračuje pod obrázkem zleva.
---	--

	Toto je zarovnání "absbottom" – spodní okraj obrázku je zarovnán s nejspodnějším písmenem v textu a dále pokračuje pod obrázkem zleva.
---	--

**WIDTH** a **HEIGHT** jsou atributy pro zadání výšky a šířky obrázku:

WIDTH="n" , kde *n* je šířka obrázku v pixelech

HEIGHT="n" , kde *n* je výška obrázku v pixelech

Např.:

```
<IMG SRC="obr1.gif" WIDTH="40" HEIGHT="60">
```

Toto zadání vykreslí obrázek do rámečku o rozměru 40x60 pixelů bez ohledu na to, jaký byl původní rozměr obrázku. Velikost, kterou zadáváte přímo ve značce zadávejte přesně stejnou jako měl původní obrázek, protože jinak dojde ke zkreslení obrázku. Zadáte-li velikost menší, kromě zkreslení plýtváte i datovým objemem (obrázek se tahá ve své původní velikosti a až při vykreslování se zmenší). Pokud jej zvětšuje, dojde opět k nehezkému zkreslení, neboť jeden obrazový bod se roztáhne přes více pixelů. Pokud zadáme jen jeden atribut, prohlížeč ho akceptuje a zbylý ve správném poměru stran dopočítá sám.

**ALT** je atribut, který definuje alternativní text (popisek obrázku). Popisky obrázků je vhodné používat zejména tehdy, pokud předpokládáme, že uživatel může vypnout zobrazování obrázků.

Např.:

```
<IMG SRC="karlmost.jpg" ALT="Karlův most">
```

**BORDER** je atribut vykreslující rámeček okolo obrázku. Syntaxe je:

BORDER="n", kde *n* je číslo udávající tloušťku rámečku v pixelech.

Barva rámečku je vždy černá kromě případu, kdy je obrázek současně odkazem. Tehdy je barva rámečku implicitně modrá. Je-li hodnota nulová, obrázek je vždy bez rámečku (a to i tehdy, je-li odkazem).

Např.:

```
<IMG SRC="obr4.gif" BORDER="6">
```

**HSPACE** a **VSPACE** jsou atributy určující volný prostor okolo obrázků a jsou pro estetickou tvorbu stránky velice důležité. Nejsou-li totiž uvedeny, je okolní text přilepený těsně k obrázku, což působí nehezky.

HSPACE="n", kde *n* je velikost volného prostoru v pixelech po levé a pravé straně obrázku

VSPACE="n", kde *n* je velikost volného prostoru v pixelech po horní a dolní straně obrázku

## Tabulky

---

Dalšími prvky jazyka HTML jsou **tabulky**. Slouží k přehlednému formátování dat do políček (nejen textových, číselných, ale také například k přehlednému uspořádání obrázků) a k mnoha dalším grafickým zpracováním stránky. Tabulka umožňuje ukládat prvky stránky vedle sebe i pod sebe a diktovat jejich vzájemnou polohu. Pomocí skrytých tabulek (bez ohraničení) lze vytvářet velice efektivní sloupcovou sazbu.

Syntaxe tabulky je následující:

```
<TABLE>          . . . začátek tabulky
<TR>            . . . začátek 1. řádky
<TD></TD>       . . . začátek a konec 1. sloupce
<TD></TD>       . . . začátek a konec 2. sloupce
</TR>          . . . konec 1. řádky
</TABLE>       . . . konec tabulky
```

Např.:

```
<TABLE>
  <TR>
    <TD>&nbsp;</TD><TD>1.</TD><TD>2.</TD>
  </TR>
  <TR>
    <TD>Pondělí</TD><TD>matematika</TD><TD>čeština</TD>
  </TR>
  <TR>
    <TD>Úterý</TD><TD>chemie</TD><TD>matematika</TD>
  </TR>
</TABLE>
```

1.	2.
Pondělí	matematika čeština
Úterý	chemie matematika

Pokud nezadáme jinak, políčka (buňky) jsou přesně tak velká, aby se do nich vešel obsah. Chceme-li mít text v jednom políčku ve více řádkách, ukončíme každou řádku značkou <BR>.

Pro vytvoření záhlaví sloupců nebo řádků můžeme použít značku <TH>, která nahradí značky <TR> nebo <TD>. Má výhodu v tom, že hlavičkový text je automaticky tučný a zarovnaný na střed.

Pro vytvoření nadpisu tabulky, který napovídá, co vlastně tabulka zobrazuje, slouží značka <CAPTION>. Nadpis je standardně umístěn na střed tabulky. Pro jiné zarovnání použijeme atribut ALIGN="left", "center" nebo "right".

Např.:

```
<TABLE BORDER="1" ALIGN="center">
  <CAPTION>Rozvrh hodin</CAPTION>
  <TR>
    <TD>&nbsp;</TD><TH>1.</TH><TH>2.</TH>
  </TR>
  <TR>
    <TH>Pondělí</TH><TD>matematika</TD><TD>čeština</TD>
  </TR>
  <TR>
    <TH>Úterý</TH><TD>chemie</TD><TD>matematika</TD>
  </TR>
</TABLE>
```

**BORDER="n"**, kde *n* je šířka ohraničení v bodech. Pokud BORDER="0" nebo není uveden, nebude mít tabulka ani žádná buňka rámeček. Je-li rámeček přítomen, budí tvar prostorovosti díky efektu stínování dvěma odstíny.

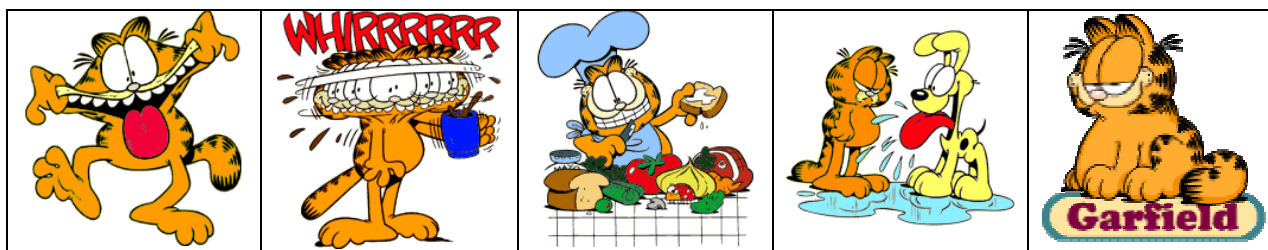
**ALIGN="left", "center" nebo "right"** provede zarovnání tabulky na stránce.

Rozvrh hodin

	1.	2.
Pondělí	matematika	čeština
Úterý	chemie	matematika

Např.:

```
<TABLE BORDER="1" ALIGN="center">
  <TR>
    <TD><IMG SRC="garfield1.gif"></TD>
    <TD><IMG SRC="garfield2.gif"></TD>
    <TD><IMG SRC="garfield3.gif"></TD>
    <TD><IMG SRC="garfield4.gif"></TD>
    <TD><IMG SRC="garfield5.gif"></TD>
  </TR>
</TABLE>
```



Tabulku bez ohraničení jednotlivých buněk můžeme také využít ke skládání obrázků – tedy k vytvoření jakési obrázkové mozaiky.

**BGCOLOR**="barva" nastaví barvu pozadí tabulky

**BORDERCOLOR**="barva" nastaví barvu ohraničení tabulky

Aby text nebyl na okraje buněk nalepen, používají se atributy:

**CELLSPACING**="n", kde *n* je mezera mezi políčky v pixelech

**CELLPADDING**="n", kde *n* je mezera mezi textem a okrajem buňky v pixelech

Např.:

```
<TABLE BORDER="1">
```

	1.	2.
Pondělí	matematika	čeština
Úterý	chemie	matematika

```
<TABLE BORDER="1" CELLSPACING="10">
```

	1.	2.
Pondělí	matematika	čeština
Úterý	chemie	matematika

```
<TABLE BORDER="1" CELLPADDING="10">
```

	1.	2.
Pondělí	matematika	čeština
Úterý	chemie	matematika

```
<TABLE BORDER="1" CELLPADDING="10"
CELLSPACING="0">
```

	1.	2.
Pondělí	matematika	čeština
Úterý	chemie	matematika

V tomto případě budou políčka přiřazena přímo na sebe, bez mezery, a mezera mezi textem a okrajem buňky bude 3 pixely.

**FRAME** je atribut, který umožňuje zobrazit pouze část ohraničení buněk nebo tabulky, hodnoty tohoto atributu mohou být:

**"void"** odstraní celý vnější rámeček tabulky, takže bude vidět jen ohraničení buněk

**"above"** zobrazí se pouze část rámečku nad tabulkou



- "below" zobrazí se pouze část rámečku pod tabulkou
- "lhs" zobrazí se pouze část rámečku na levé straně tabulky
- "rhs" zobrazí se pouze část rámečku na pravé straně tabulky
- "hside" zobrazí se pouze část rámečku nad a pod tabulkou
- "vside" zobrazí se pouze část rámečku na levé a pravé straně tabulky

Např.:

<TABLE BORDER="1" FRAME="void">

	1.	2.
Pondělí	matematika	čeština
Úterý	chemie	matematika

<TABLE BORDER="1" FRAME="hsides">

	1.	2.
Pondělí	matematika	čeština
Úterý	chemie	matematika

<TABLE BORDER="1" FRAME="above">

	1.	2.
Pondělí	matematika	čeština
Úterý	chemie	matematika

<TABLE BORDER="1" FRAME="vsides">

	1.	2.
Pondělí	matematika	čeština
Úterý	chemie	matematika

**RULES** je atribut, pomocí kterého můžeme odstraňovat rámečky okolo buněk:

- "none" odstraní všechny vnitřní okraje oddělující buňky
- "all" zobrazí veškerá ohraničení (implicitní nastavení)
- "cols" zobrazí všechny okraje mezi sloupcečky tabulky
- "rows" zobrazí všechny okraje mezi řádky tabulky

Např.:

<TABLE BORDER="1" RULES="all">

	1.	2.
Pondělí	matematika	čeština
Úterý	chemie	matematika

<TABLE BORDER="1" RULES="none">

	1.	2.
Pondělí	matematika	čeština
Úterý	chemie	matematika

<TABLE BORDER="1" RULES="cols">

	1.	2.
Pondělí	matematika	čeština
Úterý	chemie	matematika

<TABLE BORDER="1" RULES="rows">

	1.	2.
Pondělí	matematika	čeština
Úterý	chemie	matematika

Jednotlivé řádky tabulky jsou uvedeny značkou <TR>, která definuje veškerá nastavení týkající se právě této části tabulky. Můžeme u ní použít atributy **BORDERCOLOR**, **BGCOLOR**

**ALIGN**="left", "right", "center"

**VALIGN**="top", "bottom", "center"/"middle", "baseline"

Pondělí	matematika	čeština	..."top"
Úterý	chemie	cvičení z matematiky	..."bottom"
Středa	fyzika	angličtina	..."center"
Čtvrtek	cvičení z češtiny	chemie	..."baseline"
Pátek	<b>tělocvik</b>	fyzika	..."baseline"

Jednotlivá políčka tabulky jsou uvedeny značkou <TD>. Veškerá níže uvedená nastavení mají vliv pouze na aktuální buňku. Můžeme použít například tyto atributy:

### **BORDERCOLOR, BGCOLOR, ALIGN, VALIGN**

Můžeme též nastavovat šířku a výšku buněk:

**WIDTH="n"**, kde *n* je šířka v pixelech nebo procentuální vyjádření šířky (číslo se znakem procent) celé tabulky (zároveň se rozšíří všechny buňky ve stejném sloupci – sloupec bude mít takovou výšku, jakou má nejširší buňka)

**HIGHT="n"**, kde *n* je výška v pixelech (řádek bude mít takovou výšku, jakou má nejvyšší buňka)

Např.:

```
<TABLE BORDER="1">                                <TABLE BORDER="1">
  <TR>                                              <TR>
    <TD WIDTH="100">Ráno</TD>...                <TD WIDTH="30%">Ráno</TD>...
```

Buňky můžeme také spojovat pomocí atributů:

**ROWSPAN="n"**, kde *n* je celkový počet řádků, přes které bude políčko rozloženo

**COLSPAN="n"**, kde *n* je celkový počet sloupců, přes které bude políčko rozloženo

Např.:

```
<TABLE BORDER="1">
  <TR>
    <TD>&nbsp;</TD><TD>1.</TD><TD>2.</TD><TD>3.</TD>
  </TR>
  <TR>
    <TD>Pondělí</TD><TD>matematika</TD>
    <TD ROWSPAN="2">čeština</TD><TD>fyzika</TD>
  </TR>
  <TR>
    <TD>Úterý</TD><TD>chemie</TD><TD>angličtina</TD>
  </TR>
</TABLE>
```

	1.	2.	3.
Pondělí	matematika	čeština	fyzika
Úterý	chemie		angličtina

nebo

```
<TABLE BORDER="1">
  <TR>
    <TD>&nbsp;</TD><TD>1.</TD><TD>2.</TD><TD>3.</TD>
  </TR>
  <TR>
    <TD>Pondělí</TD><TD>matematika</TD><TD>čeština</TD><TD>fyzika</TD>
  </TR>
  <TR>
    <TD>Úterý</TD><TD COLSPAN="2">chemie</TD><TD>angličtina</TD>
  </TR>
</TABLE>
```

	1.	2.	3.
Pondělí	matematika	čeština	fyzika
Úterý	chemie		angličtina

## Kontrola, ověřování a další doporučení

---

Stali jste se vydavatelem, jehož dílo může číst kdokoli na světě. Proto:

- **Pečlivě testujte veškeré odkazy**, které na svých stránkách vytvoříte. Neplatné odkazy jsou velice nepříjemné a také cosi nelichotivého vypovídají o vaší pečlivosti.
- **Ověřujte i správnou syntaxi HTML**. To, že se na stránce v prohlížeči cosi objeví, a že se to dokonce blíží vašim záměrům, ještě neznamená, že je HTML syntaxe správná. Důsledkem pak mohou být špatně zobrazené stránky v jiných prohlížečích, než ve kterém jste vzhled stránky kontrolovali. Existují programy pro kontrolu HTML kódu, které vám mohou s kontrolou pomoci.
- **Ověřujte aktuálnost** a datujte poslední aktualizaci stránek.
- **Koncipujte stránku** tak, aby se správně zobrazovala bez závislosti na rozlišení obrazovky. Proto je lepší (než se stanete zkušeným tvůrcem) vyhýbat se všem prvkům, jejichž tvar nebo umístění jsou závislé na šířce okna prohlížeče. V dnešní době je vhodné počítat s nejrozšířenější šířkou obrazovky 1024 pixelů, ale je nutné mít na paměti, že uživatel může mít i šířku dvojnásobnou, nebo naopak podstatně menší (např. u kapesních počítačů).
- **Dejte svým uživatelům zpětnou vazbu**. Je rozumné dát na stránky odkaz na autora či odpovědného správce těchto stránek, uvidíte, jak často vás právě čtenáři vašich stránek upozorní na nalezené chyby.
- Neuškodí, když budete dbát o **co nejmenší velikost obrázků** – u stolních počítačů s rychlým připojením to již není tak významný faktor, ale roste počet kapesních počítačů, jejichž uživatelé efektivitu přenosu dat stále ocení. Existuje ještě několik dalších tipů týkajících se umístování obrázků do webové stránky:
- Přidáme-li atributy HEIGHT a WIDTH, tj. **informaci o velikosti obrázku**, může prohlížeč ihned vykreslit rámeček, do kterého se bude obrázek postupně načítat a pokračovat v zobrazování textu, které je rychlejší (text nám pak nepříjemně neuskočí před očima).
- Nenechejte se unést obrazovým pozadím. Používejte pozadí nenápadné – jednoduché neagresivní barvy, které se považují za nejelegantnější (nejlépe bílou) nebo nerušivé vzorky (žádné hrany ani kontrasty) – aby nekonkurovalo vlastnímu obsahu stránky. Snažte se, aby **pozadí příliš nezhoršovalo čitelnost textu**. Soustřeďte se spíše na práci s typografií.
- Pokud umístíte na stránku s textovou informací jakékoli **pohybující se komponenty**, radikálně zhoršíte její čitelnost. Dopřejte svým čtenářům klid na čtení! Animovaný prvek totiž na sebe velmi hrubým způsobem stáhne uživatelskou pozornost. Ostatní součásti stránky mu v tomto směru nemohou konkurovat.
- **Hudba na pozadí** sice působí velmi originálně, ale také poněkud vlezle, takže po chvíli prohlížení takové stránky si už většinou budete přát, aby už konečně bylo ticho. Proto si dobře rozmyslete, jestli nekonečná hudební smyčka je přesně to, na co vaši čtenáři čekají.

## Netiquette aneb pravidla slušného chování v počítačové síti

Netiquette (zkratka z Network Etiquette) je soubor doporučení, nikoli zákon. Dodržování pravidel slušného chování na internetu není jen věcí osobní slušnosti. Vypovídá také velice silně o tom, zda jste už profesionální autor a znalec internetu, či zda jste začátečník, zelenáč. Obsahuje jak obecná pravidla pro práci s počítačem, tak pravidla chování v počítačové síti. Mimo jiné například:

- **Pozor na autorská práva a ochranné známky**. Pokud chcete vystavovat materiál, jehož autorem je někdo jiný, je nutné získat povolení od autora, jinak porušujete autorský zákon.
- **Neurážejte žádného z potenciálních čtenářů**. I když jsou vaše stránky zaměřené pouze na určitou skupinu lidí, mějte na paměti, abyste se nedotkli ostatních – rasových menšin, nemocných lidí, lidí různého náboženského vyznání apod.
- **Nepublikujte obscénnosti, perverznosti či pornografii** – internet je sice svobodné médium, ale svobodu nelze zneužívat a také má určité hranice.

## Hierarchie a obsah

Pravděpodobně se vám nepodaří umístit vše, co chcete sdělit, na jedinou webovou stránku – musíte tedy uvažovat o hierarchii těchto stránek. Základní koncept pyramidy je jasný: existuje jedna centrální – **domovská stránka** (Home Page – obvykle s názvem *index.html*), která je pro návštěvníka výchozí a proto by měla obsahovat srozumitelnou navigaci, aby se snadno dostal k těm stránkám, které ho zajímají. Je-li stránek větší množství, bývá obvyklé na hlavní (nebo i na každou) stránku umístit strom s hierarchií stránek pro lepší orientaci.

## Na závěr

---

Jednotlivých značek má jazyk HTML velice mnoho (a stejně tak i různých atributů). Dalšími prvky jazyka HTML jsou například **rámy**, **formuláře** a **styly**. Informace o nich však již překračuje rámec této základní příručky.

Své stránky také můžeme oživit pomocí modulů **ActiveX** nebo prvků programovacího jazyka **JavaScript** nebo **VBScript** (Visual Basic Script), které lze do zdrojového kódu implementovat a vytvářet tak živé či interaktivní stránky reagující na impulsy uživatele. Pro zájemce, kteří chtějí dále rozvíjet své znalosti v oblasti tvorby webových stránek je přiložen seznam použité odborné literatury.

Původní jazyk pro publikování na webu HTML již přestal vyhovovat především pro své omezené možnosti.

Následníkem jazyka HTML, jehož vývoj byl ukončen, je **XHTML** (eXtensible HyperText Markup Language – rozšiřitelný značkovací jazyk pro hypertext). Na rozdíl od svého předchůdce se jedná o aplikaci XML. Jazyk **XML** (eXtensible Markup Language, rozšiřitelný značkovací jazyk) je určen především pro výměnu dat mezi aplikacemi a pro publikování dokumentů. Tento jazyk umožňuje popsat strukturu dokumentu z hlediska věcného obsahu jednotlivých částí, nezabývá se sám o sobě vzhledem dokumentu nebo jeho částí.

**CSS** (Cascading Style Sheets, tabulky kaskádových stylů) je jazyk pro popis způsobu zobrazení stránek napsaných v jazycích HTML, XHTML nebo XML. Hlavním smyslem je umožnit návrhářům oddělit vzhled dokumentu od jeho struktury a obsahu.

**PHP** (Hypertext Preprocessor, původně Personal Home Page) je programovací jazyk určený především pro programování dynamických internetových stránek (stránka není statická, ale vytváří se dynamicky na serveru až na základě požadavku od klienta). Jazyk PHP se stal velmi oblíbeným především díky jednoduchosti použití a tomu, že kombinuje vlastnosti více programovacích jazyků a nechává tak vývojáři částečnou svobodu v syntaxi. V kombinaci s databázovým serverem MySQL a webovým serverem Apache je často využíván k tvorbě webových aplikací.

**ASP** (Active Server Pages) je dalším z rozšířených nástrojů pro tvorbu dynamických stránek (jedná se o komerční produkt firmy Microsoft).

**Flash** je vektorový grafický editor používaný především pro tvorbu interaktivních animací, prezentací a her. Flash také disponuje vlastním programovacím jazykem ActionScript, který slouží k vytváření interaktivních animací. Flashová animace je skupina snímků, které se mezi sebou vyměňují (podobně jako u animovaného GIFu). Animace lze dosáhnout minimálně třemi způsoby: animace pohybem (stačí pouze první a poslední snímek – ostatní se vytvoří samy), animace změnou tvaru nebo animace snímek po snímku (stejně jako u GIFu). Na rozdíl od formátu GIF, který je bitmapový (obrázek je tvořen jednotlivými body), se u Flashe jedná o vektorovou grafiku (obrázek je tvořen pomocí čar a výplní).

## Obsah

Jazyk HTML .....	1
Jak stránky připravovat.....	2
Úvod do jazyka HTML.....	2
Struktura dokumentu – základní části stránky .....	3
Tělo dokumentu.....	5
Tabulka základních pojmenovaných barev .....	5
Formátování textu .....	6
Nadpis.....	6
Text a jeho základní formátování.....	6
Značky pro změnu písma .....	7
Symboly pro speciální znaky .....	8
Seznamy a výčty.....	8
Vodorovná linka .....	9
Odkazy.....	9
Umístování odkazů do textu .....	11
Obrázky .....	12
Tabulky.....	14
Kontrola, ověřování a další doporučení .....	19
Netiquette aneb pravidla slušného chování v počítačové síti.....	19
Hierarchie a obsah .....	20
Na závěr .....	20
Obsah.....	21
Literatura a zdroje informací .....	21

## Literatura a zdroje informací

Petr Broža: **Tvorba WWW stránek pro úplné začátečníky**, Computer Press, Praha 1999

Jana Hanzlíková: **Webdesign pro úplné začátečníky**, Computer Press, Brno 2004

Steve Krug: **Web design Nenuťte uživatele přemýšlet!**, Computer Press, Brno 2003

David Morkes: **Ožívování WWW stránek pomocí skriptů**, Grada, Praha 2002

Jiří Hlavenka, Radek Sedlář, Tomáš Holčík, Martin Šebesta a Richard Botík: **Vytváříme WWW stránky a spravujeme moderní web site**, Computer Press, Praha 1997

Pavel Satrapa: **World Wide Web pro čtenáře, autory a misionáře**, Neokortex, Praha 1996

Pavel Satrapa: **Web Design**, Neokortex, Praha 1997

Lubor Mrázek: **První kroky s Internetem aneb Je to na WWW!**, KOPP, České Budějovice 1995

Webové stránky:

[www.jaknaweb.cz](http://www.jaknaweb.cz)

[www.jakpsatweb.cz](http://www.jakpsatweb.cz)

[www.webtvorba.cz](http://www.webtvorba.cz)

[cs.wikipedia.org](http://cs.wikipedia.org)

*Zpracovala: Ing. Simona Martínková*

*© update listopad 2009*