

Tvorba první HTML stránky

Formátovací značky v HTML

Každý HTML dokument obsahuje kromě samotného textu stránky také informace o formátování stránky. Tyto informace jsou určeny pro prohlížeč, aby věděl, jak má s každým prvkem (textem, obrázkem, tabulkou atd.) naložit. Tyto informace se prohlížeči předávají pomocí tzv. značek (tagů). V HTML existují dva druhy značek. Jsou to značky párové (<html> - otevírací, </html> - uzavírací) a nepárové (
). Párové značky platí pro text, který je mezi nimi uzavřen. Nepárové platí pouze v místě, kde jsou uvedeny (zalomení řádku, vložení obrázku) nebo pro zbytek dokumentu, který následuje po značce. Nepárových značek je však pouze několik. K většině značek se mohou přidávat parametry. Ty potom mohou měnit například zarovnávání (<p align="center">), druh odrážky (<li type="circle">), barvu textu () a další.

Struktura stránky

Nejenom značky a samotný text dělají stránku stránkou. Každá stránka musí mít pevnou strukturu. Jednotlivé značky se nesmí křížit (<p> ... </p>)! Značky musí být ukončeny v opačném pořadí, než byly otevřeny! V tomto případě to znamená: <p> ... </p>.

Každá stránka by měla obsahovat tuto základní strukturu:

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 TRANSITIONAL//EN">

Sděluje, v jaké verzi HTML je stránka napsána. Uvedením tohoto řádku budete mít jistotu, že stránku bude možné zpracovávat nástroji pro strojové zpracování dokumentů. V praxi se můžete setkat s tím, že se tato značka vynechává, ale doporučuji ji uvést. Neexistuje samozřejmě pouze tato jedna procedura DOCTYPE, ale prozatím s ní vystačíme!

<html>...</html>

Sděluje prohlížeči, kde začíná a končí zdrojový text zobrazované stránky. Celý zdrojový text stránky je umístěn mezi tyto značky.

<head>...</head>

Hlavička je důležitou částí HTML dokumentu. Obsah hlavičky se na stránce nezobrazí. Jsou zde umístěny například: název stránky, metainformace, definice kaskádových stylů a další.

<body>...</body>

Tělo dokumentu. Právě sem se zadává samotný text stránky včetně formátovacích značek.

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

Metainformace o použitém kódování. Je velice důležitá. Kdybychom ji neuvedli, mohlo by dojít k tomu, že se budou chybně zobrazovat znaky jako: "ěščřžýáíé"! Existuje mnoho druhů kódování. U nás se nejčastěji používá windows-1250, iso-8859-2 a také UTF-8. Pokud používáte pro psaní HTML stránek Poznámkový blok, nezbývá vám jiná volba, než windows-1250. Těm, kteří používají například již zmíněný editor PSPad a další editory, doporučuji používat znakovou sadu iso-8859-2.

Samotný HTML dokument má tedy následující strukturu:

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 TRANSITIONAL//EN">

<html>

<head>

<meta http-equiv="content-type"

content="text/html; charset=windows-1250">

</head>

<body>

Tento text se zobrazí v okně prohlížeče.

</body>

</html>

Jak si tuto stránku napsat? Je to velmi jednoduché. Otevřeme si HTML editor (nebo Poznámkový blok), kde opíšeme tento text (místo věty "Tento text" si můžete napsat cokoli). Hotový text uložíme například jako prvni.html. Pokud váš systém nepodporuje čtyřmístnou koncovku, napište prvni.html. Dejte si však pozor. **Poznámkový blok rád ukládá takto: prvni.html.txt. To je samozřejmě špatně! Musíte kolonku *uložit jako typ* změnit na *všechny soubory*. Když bude dokument uložen, stačí na něj poklepat myší a on se sám otevře v okně prohlížeče.**

POZOR:

Formátovací značky lze psát jak <malými písmeny> tak <VELKÝMI PÍSMENY> dokonce i <TakOVÝto Zápis> je možný, ale pokud takto napíšete nějakou složitější stránku, uvidíte, že se v tom sami nevyznáte! Je proto dobré, zvolit si jeden způsob a toho se držet. Dříve se značky psaly převážně velkými písmeny. S příchodem XML a XHTML se od toho však upouští. V tomto dokumentu budu psát značky malými písmeny. Špičaté závorky (<,>) nejsou zahrnuty na české klávesnici. Neznamená to však, že musíme pokaždé měnit klávesnici. Stačí znát jednoduchou kombinaci kláves. Obě závorky se píšou pomocí AltGr a klávesy čárka (<) a tečka (>) na české klávesnici.

Co všechno se dá napsat do hlavičky

Do hlavičky patří například název stránky. Ten se vepisuje mezi značky <title>...</title>. Název stránky se zobrazuje v záhlaví okna prohlížeče. Proto by neměl být příliš dlouhý. Doporučuje se maximálně 60 znaků. Dále by měla každá stránka obsahovat informace, které se stránky přímo týkají (obsah, klíčová slova, autora stránky, kódování). Těmto informacím se říká metainformace. Do hlavičky se vepisují pomocí nepárové značky <meta>, která se doplní o parametry. Parametrem u značky <meta> je *name*. Ten může mít tyto hodnoty:

description
popis
keywords
klíčová slova
author
autor

Metainformace jsou nepovinné, ale jestli uvažujete o umístění vaší stránky na internet, doporučuji je uvést.

Jak by tedy měla správně vypadat předchozí ukázka?

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 TRANSITIONAL//EN">
<html>
<head>
<meta http-equiv="content-type"
  content="text/html; charset=windows-1250">
<meta name="description"
  content="ukázka první WWW stránky určená pro začínající tvůrce">
<meta name="keywords"
  content="ukázka, první www, www stránky, začínající tvůrce">
<meta name="author"
  content="Pepa Jahoda">
<title>První stránka</title>
</head>
<body>
  Tento text se zobrazí v okně prohlížeče.
</body>
</html>
```

Pokud si tento kód opíšeme a poté otevřeme v prohlížeči, uvidíme totéž jako předtím, až na to, že se změnil text v záhlaví okna. Později se ještě k hlavičce vrátíme. Bude to v souvislosti s kaskádovými styly.

Komentáře

Ještě, než postoupíme do další kapitoly, zmíníme si nedílnou součást jazyka HTML. Co jsou komentáře? Komentáře jsou nedílnou součástí zdrojového kódu. Přestože nemají na vzhled stránky žádný vliv, nejsou tak úplně zbytečné. Pomohou nám v orientaci ve zdrojovém kódu. Vkládají se pomocí značky `<!-- text komentáře -->`. Mohou se však hodit i při samotném psaní stránky. Když například chcete na stránce nějakou část vypustit, můžete ji označit jako komentář. Výhodou je, že v případě, kdy budete chtít původní text vrátit, nemusíte ho psát znovu nebo klikat na tlačítko zpět. Stačí pouze vymazat značky, které ohraničují text komentáře.

Opět si uvedeme krátký příklad (v příkladech budu metainformace **vynechávat**):

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 TRANSITIONAL//EN">
<html>
  <head>
    <title>Ukázka komentáře</title>
  </head>
  <body>
    Tento text se zobrazí v okně prohlížeče.
    Následuje komentář.
    <!-- Tento text je ohraničen značkami komentáře
    a proto nebude v prohlížeči zobrazen. -->
    Tento text se opět zobrazí v okně prohlížeče. Dejte
    si proto pozor na to, co do něho napíšete.
  </body>
</html>
```

Formátování textu I.

Takže úvod do jazyka HTML máme úspěšně za sebou a nezbyvá nám tedy nic jiného, než se seznámit s jeho možnostmi. Začneme tedy formátováním textu.

Nadpisy

Základní způsob dělení textu je pomocí nadpisů. HTML podporuje šest úrovní nadpisů. Umístění nadpisů do dokumentu se provádí pomocí párové značky `<hx>...</hx>`, kde *x* označuje úroveň nadpisu.

Zkusme si tedy zobrazit na stránce nadpisy všech šesti velikostí:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 TRANSITIONAL//EN">
<html>
  <head>
    <title>Ukázka nadpisů</title>
  </head>
  <body>
    <h1>Nadpis úrovně 1</h1>
    <h2>Nadpis úrovně 2</h2>
    <h3>Nadpis úrovně 3</h3>
    <h4>Nadpis úrovně 4</h4>
    <h5>Nadpis úrovně 5</h5>
    <h6>Nadpis úrovně 6</h6>
    Toto je běžný (neformátovaný) text stránky.
  </body>
</html>
```

Jak je vidět z této ukázky, nadpisy páté a šesté úrovně jsou menší, než běžný text stránky. Neznamena to však, že se nepoužívají. Téměř každý má rád barevně odlišené nadpisy. Pokud budou tyto nadpisy barevně odlišeny, není nejmenší důvod, je nepoužít.

Odstavce a optické dělení textu

Jelikož prohlížeče ignorují všechny přebytečné mezery a znaky zalomení řádku, je důležité říci prohlížeči, kde mají být odstavce. Označit odstavec je hračka. Když už máme za sebou nadpisy, nebude nám to dělat žádné problémy. Odstavec uzavřete mezi značky `<p>...</p>` (p = paragraph = odstavec).

A tady máme ukázku se dvěma odstavci:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 TRANSITIONAL//EN">
<html>
  <head>
    <title>Ukázka odstavců</title>
  </head>
  <BODY>
    <h1>Na této stránce jsou dva odstavce</h1>
    <p>
      Toto je text prvního odstavce, který se nalézá na této
      stránce. Toto je druhá věta prvního odstavce, ve které
      se však již nic nedozvíte, protože je tu jenom proto, aby
      byl odstavec o trochu delší.
    </p>
    <p>
      Nezapomínejme na to, že tato stránka má také druhý
      odstavec. Toto je druhá věta druhého odstavce, ve které
      se však již nic nedozvíte, protože je tu jenom proto, aby
      byl odstavec o trochu delší.
    </p>
  </body>
</html>
```

Pokud však potřebujeme začít psát na nový řádek (ne do nového odstavce) je tu pro nás nepárová značka `
` (BR = break = zalomení). Problém však nastane, pokud nechceme, aby bylo například víceslovné jméno nějaké společnosti rozděleno na dva řádky. Existuje tady jistá možnost použití párové značky `<nobr>...</nobr>` (nobr = no break = nezalamuj), do které uzavřeme text, který nemá být rozdělen na více řádků. Nevýhodou však je, že není zcela v souladu se standardem HTML. Tuto značku údajně podporují pouze prohlížeče MS Internet Explorer a Netscape Navigator. Protože se ale jedná o nejpoužívanější prohlížeče, myslím si, že to můžeme opomenout. Pokud však chcete napsat stránky podle standardů HTML, doporučuji se této značce vyhnout.

Další možností, jak vizuálně oddělit text, je vložení horizontální čáry. Čáru vložíme do dokumentu pomocí nepárové značky `<hr>`:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 TRANSITIONAL//EN">
<html>
  <head>
    <title>Ukázka horizontální čáry</title>
  </head>
  <body>
    <h1>Na této stránce jsou dva odstavce</h1>
    <p>
      Toto je text prvního odstavce, který se nalézá na této
      stránce. Toto je druhá věta prvního odstavce, ve které
```

se však již nic nedozvíte, protože je tu jenom proto, aby byl odstavec o trochu delší.

```
</p>
```

```
<hr>
```

```
<p>
```

Nezapomínejme na to, že tato stránka má také druhý odstavec. Tento odstavec je od prvního navíc oddělen horizontální čarou. Toto je třetí věta druhého odstavce, ve které se však již nic nedozvíte, protože je tu jenom proto, aby byl odstavec o trochu delší.

```
</p>
```

```
</body>
```

```
</html>
```

Tato ukázka asi vyvolá debatu o tom, že není čára jako čára. Já s vámi souhlasím. Proto si řekneme, jak lze tuto čáru změnit. Je to velmi jednoduché. Stačí přidat ke značce `<hr>` parametr *size*, který udává tloušťku čáry v pixelech (obrazových bodech). Délku čáry lze samozřejmě měnit také. To umožňuje parametr *width*. Hodnotu tohoto parametru lze zadat buď v pixelech (obrazových bodech) nebo v procentech. Zadávat délky v pixelech má jednu velkou nevýhodu. Ta je způsobena tím, že všichni nepoužívají stejné obrazové rozlišení jako my. To znamená, že my si uděláme čáru, ale ta se v jiném rozlišení zobrazí úplně jinak. Je proto výhodnější zadávat tuto hodnotu v procentech. Někdy se však stane, že máte na stránce nějaký odstavec, který má pevnou šířku (například levé menu na této stránce). Potom je možná vhodnější použít pevnou hodnotu v pixelech.

Zarovnávání

V textu se samozřejmě neobejdeme bez zarovnávání. Nadpis někdy vypadá lépe, je-li zarovnán na střed. Další využití bude mít také u obrázků a tabulek. Tabulky sice přijdou až později, ale určitě nebude na škodu, když si něco o zarovnávání řekneme už teď. Celá věda tkví v tom, že přidáme ke značce (například `<p>`) parametr *align*. Tento parametr může mít tři hodnoty:

`left`

vlevo

`center`

na střed

`right`

vpravo

Parametr `align` lze přidat k mnoha značkám. Budeme se s ním tedy hojně setkávat i v příštích kapitolách.

Ukážeme si to na příkladu:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 TRANSITIONAL//EN">
```

```
<html>
```

```
<head>
```

```
<title>Ukázka zarovnávání</title>
```

```
</head>
```

```
<body>
```

```
<h1 align="center">
```

```
Nadpis zarovnaný na střed
```

```
</h1>
```

```
<p align="left">
```

```
Toto je text prvního odstavce, který se nalézá na této stránce. Tento odstavec je zarovnán vlevo.
```

```
</p>
```

```
<p align="right">
```

```
Toto je text druhého odstavce, který se nalézá na této
```

```
    stránce. Tento odstavec je zarovnán vpravo.  
</p>  
</body>  
</html>
```

Zobrazení v této ukázce je samozřejmě hezké, ale často se stává, že potřebujeme zarovnat několik odstavců například vpravo. To by znamenalo, že u každé značky `<p>` bychom museli uvést parametr `align` s hodnotou `right`. Jazyk HTML však pamatuje i na toto. Pro nastavení společných vlastností několika odstavců (v našem případě zarovnání) existuje párová značka `<div>...</div>`. Odstavce, které budou mezi tyto značky uzavřeny, budou mít stejné vlastnosti.

Opět si uvedeme ukázkou:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 TRANSITIONAL//EN">  
<html>  
<head>  
<title>Ukázka DIVu</title>  
</head>  
<body>  
<p>  
    Toto je text prvního odstavce, který se nalézá na této  
    stránce. Tento odstavec je zarovnán vlevo.  
</p>  
<div align="right">  
<p>  
    Toto je text druhého odstavce, který se nalézá na  
    této stránce. Tento odstavec je zarovnán vpravo,  
    protože je pod vlivem značky DIV.  
</p>  
<p align="left">  
    Toto je text třetího odstavce, který se nalézá na  
    této stránce. Tento odstavec je zarovnán vlevo  
    (i když má být pod vlivem DIVu), protože má  
    nastaveno vlastní zarovnávání.  
</p>  
<p>  
    Toto je text čtvrtého odstavce, který se nalézá na  
    této stránce. Tento odstavec je zarovnán vpravo,  
    protože je pod vlivem značky DIV.  
</p>  
</div>  
</body>  
</html>
```

Tato ukázka je pouze jednou z možností zarovnání více odstavců. Další možnost si ukážeme, až dojdeme ke kapitole **Kaskádové styly**.

Odkazy

Tag	Význam	Párový	Výskyt
a	odkaz, hyperlink, záložka	ano	kdekoliv kromě <a>

a

Odkaz (z angl.anchor).

Obsah odkazu

Vše, co je mezi značkami <a> a bude sloužit jako odkaz (tzn. bude klikatelné).

Např.

`Hlavní stránka` se zobrazí jako *klikatelný nápis Hlavní stránka*.

Uvnitř elementu se může objevit skoro cokoli (např. obrázek, který tak bude celý sloužit jako odkaz). Uvnitř odkazu nesmí být

- další odkaz (tag <a>),
- formulář (tag <form>),
- tabulka (tag <table>) uvnitř odkazu nefunguje jako odkaz a podobně se mohou chovat i jiné složité tagy.

Atributy tagu <a>		
Atribut	Význam	Hodnota
href	cíl odkazu	URL
name	jméno záložky	libovolné jméno
target	cílový rám	jméno rámu
rel	druh odkazu	několik vyjmenovaných hodnot
hreflang	jazyk cíle	kód jazyka
ping	měření prokliků	sledovací adresa

Href

Nejdůležitější atribut, cíl odkazu. Zjednodušeně pojato tato hodnota říká, jaká stránka se objeví po kliknutí. Href se zapisuje jako [URL](#).

Příklad:

`Portál Seznam`

Name

Jméno záložky. Tag <a> může sloužit kromě odkazu i jako záložka. Záložka je dobrá k tomu, že definuje místo v dokumentu, na které může mířit odkaz (jiný tag a s href="#jmenoZalozky"). Po kliknutí na takový odkaz se stránka odroluje na pozici záložky.

Tag <a> definující záložku neobsahuje atribut href, ale name. Obsah tagu může být prázdný.

Příklad:

 Text příkladu

Po kliknutí na tento odkaz odroluje stránka na záložku příkladu.

Target

Cílové okno, iframe, nebo rám, ve kterém se obsah zobrazí. Pokud se ve stránkách používají [rámy](#), má každý z nich jméno. Jméno rámu je obsaženo v atributu name v tagu <frame> (případně <iframe>).

Příklad:

hlavní stránka, se otevře v rámu nebo okně "okenko".

Kromě jmen existujících ráků se jako hodnota atributu target mohou použít speciální jména virtuálních ráků:

Jména virtuálních ráků	
target=	Kde se to otevře
_blank	nové okno nebo nový tab
_top	nejvyšší okno hierarchie ráků
_parent	nadřazený rám
_self	tentýž rám

Příklad:

Yuhůva stránka, která se otevře v novém okně.

Pokud se v atributu target objeví jméno neexistujícího okna nebo rámu, otevře se nové okno. Je to velmi podobné hodnotě _blank s tím rozdílem, že nové okno (nebo tab) si svoje jméno podrží, takže případné další podobné odkazy se budou zobrazovat v tomto novém okně (nebo tabu).

Otevírání odkazů v rámech lze ovlivnit také tagem <base> v hlavičce stránky, jehož atribut target nastaví cíl pro všechny odkazy, kterým nebude cíl předefinován přímo v odkazu.

Pozor na častý překlep: traget je špatně, target je správně.

Rel

Atribut rel je ve specifikaci dlouho, ale téměř se nepoužívá (spíše u tagu <link>). Teorie říká, že když se do odkazu (do tagu <a>) přidá třeba rel="copyright", že to sděluje, že odkaz míří na stránku s popisem autorských práv k textu. Další hodnoty jsou alternate, stylesheet, start, next, prev, contents, index, glossary, chapter, section, subsection, appendix, help a bookmark a v praxi jsou všechny totálně k ničemu.

Počátkem roku 2005 Google doporučil používat novou (nestandardní) hodnotu **rel="nofollow"** (= angl. "nesledovat") ve všech odkazech, u kterých nechcete, aby byly Googlem brány v úvahu.

Příklad:

odkaz, který nebude sledován

Jak přesně to funguje, se zatím neví. Google to udělal kvůli tomu, aby se daly zablokovat odkazy spammerů z diskusí. Postupně se přidávají další vyhledávače, takže počátkem roku 2008, kdy tento text aktualizují, ignorují takový odkaz všechny důležité vyhledávače včetně Yahoo, Bing a Seznamu.

Hreflang

Označení jazyka stránky, na kterou odkaz ukazuje. Např. `hreflang="en"` znamená, že odkaz míří na anglicky psanou stránku. Neznám žádné rozumné použití a pochybuji, že si tohoto atributu všímají vyhledávače. Další kódy jazyka jsou např. `cs`, (nikoli `cz`), `de`, `sk`, `fr` apod.

Ping

V prohlížeči Mozilla od verze 1.6 je zavedena podpora nového atributu `ping`. Způsobí, že po kliknutí na odkaz prohlížeč zároveň přistoupí i na URL obsaženém v atributu `ping`. Usnadní to měření počtu prokliků, ale v praxi to moc použitelné není. Symbolicky zapsáno:

```
<a href="adresa-odkazu" ping="adresa-měřící-tečky">
```

Mezery v textu

Prohlížeče podtrhávají text odkazu včetně mezer.

- Pokud text odkazu mezerou *začíná*, mezeru se *nepodtrhne*.
- Pokud text odkazu mezerou *končí*, **mezeru se podtrhne**.

Proč to zdůrazňuji -- protože jako mezeru se v HTML chápe i zalomení řádku. Pokud máte ve zvyku kód krásně strukturovat do řádků, tak se vám na konci odkazu podtrhne mezeru a vypadá to fakt divně. V případě odkazů tedy doporučuji psát je na jeden řádek.

Příklad kódu odkazu s podtrženou mezerou na konci:

```
<a href="kamkoli"> text odkazu s podtrženou mezerou na konci </a>
```

nebo

```
<a href="kamkoli">  
  text odkazu s podtrženou mezerou na konci  
</a>
```

Další varianty odkazů

Prvky stránky, které se chovají jako odkazy, lze vytvořit i jinými způsoby:

- Obrázky s klikacími mapami (tag `` s atributy `ismap` a `usemap` při použití tagu `<map>`).
- Potvrzovacím tlačítkem formuláře (tag `<form>`, který má atribut `action` nastaven na cílovou stránku).
- Skriptem svázaným s libovolným elementem pomocí `události` `onclick`, který mění `document.location`.

Obrázky

Tag	Význam	Párový	Výskyt
img	obrázek	ne	
map	klikací mapa	ano	
area	oblast v klikací mapě	ne	<map>

Img

Obrázek (angl. image). Nepárový tag. Do stránky se vloží obrázek načtený z jiného souboru. Tag má kromě [obecných atributů](#) spoustu dalších:

Atributy tagu 			
Atribut	Význam	Hodnoty	Nahradíte CSS stylem
src	umístění souboru s obrázkem	URL	
alt	alternativní popis	libovolný text	
lowsrc	náhradní obrázek pro malé displeje	URL	
width	šířka	délka nebo procento	width
height	výška	délka nebo procento	height
border	tloušťka rámečku	délka	border
vspace	vertikální okraj	délka	margin
hspace	horizontální okraj	délka	
align	zarovnání obrázku	left, right, top, texttop, middle, absmiddle, baseline, bottom, absbottom	float (left, right) vertical-align (ostatní hodnoty)
usemap	použití klikací mapy	"#" + jméno mapy, nebo URL	
ismap	přidání souřadnic kliknutí	bez hodnoty	
longdesc	dlouhý popis	text nebo URL	

Src

Zdój obrázku, prostě cesta k tomu souboru (angl. source = zdroj). Cesta k obrázku se udává jako [klasické URL](#) absolutně nebo relativně. Příпустné jsou soubory typu *.gif, *.jpg, v novějších prohlížečích také *.png a v Internet Exploreru *.bmp. (Jiné typy souborů se musejí vkládat jako objekty pomocí doplňku prohlížeče). Pokud soubor obrázku nebude nalezen, zobrazí se ve stránce zástupný symbol nenalezeného obrázku (nejčastěji červený křížek) se zástupným textem.

Příklad vložení obrázku strom.gif, který se nachází ve stejném adresáři (relativní cesta):

```

```

Příklad vložení obrázku absolutní cestou:

```

```

Pokud je src zadané špatně, objeví se místo obrázku červený křížek. V nových prohlížečích (v IE až od verze 8) lze obrázek zapsat také přímo do HTML nebo CSS souboru pomocí schématu data.

Alt

Zástupný text (alternative text). Zobrazuje se ve stránce v oblasti nenačteného obrázku. Spousta lidí prochází web s vypnutými obrázky (třeba já kvůli slabému připojení), mnoho uživatelů webu je slabozrakých, slepci si nechávají stránky předčítat čtečkami nebo převádět do braila. Pro všechny tyto uživatele jsou obrázky neviditelné. Oni vědí, že na stránce je obrázek, ale nevědí, co na něm je. A proto (a nejen proto) se tam píše alt.

Příklad zástupného textu "strom":

```

```

Pokud je obrázek načten, zobrazuje se v Internet Exploreru hodnota altu při přejíždění obrázku ve žluté bublině. Chcete-li udělat žlutou bublinu standardně, funkční v Exploreru i v jiných prohlížečích, nepoužívejte na to alt, ale atribut [title](#). Pokud chcete zadat alt, ale nechcete, aby jej Explorer zobrazoval ve žluté bublině, přebijte alt prázdným title, bublina v Exploreru nebude.

Lowsrc

Zástupný obrázek pro malé displeje. Nikdy jsem neviděl použití. V mnoha návodech se uvádí, že se tento obrázek načítá před načtením vlastního src. To ale platí pouze pro starý Netscape, který už nikdo nepoužívá. Osobně doporučuji lowsrc nepoužívat.

Width, height

Šířka (width) a výška (height) obrázku. Udává se v [píxelech](#) (normální číslo) nebo [procentech](#). Pokud rozměry nejsou uvedeny, zobrazí se obrázek ve své normální velikosti. V praxi se doporučuje výšku a šířku u obrázku zadávat i v případě, že obrázek bude přirozeně veliký. To kvůli rychlejšímu a lepšímu načítání stránky.

Pokud se zadají jiné rozměry, než má obrázek ve skutečnosti, pak se ve výsledku obrázek zvětší, zmenší nebo zdeformuje. V případě, že je uveden jenom jeden rozměr (šířka nebo výška), upraví se druhý rozměr tak, aby obrázek zachovával proporce (takže může být jinak velký, ale nebude deformovaný).

Pozor na častou chybu přehazování písmenek t a h. **Widht** je špatně, správně je **width**. **Heigth** je špatně, správně je **height**.

Příklad: mám obrázek strom.gif o šířce 50 a výšce 100 pixelů. Nejlépe je zadat jej takto:

```

```

V případě, že rozměry nezadám, obrázek se nakonec načte stejně velký. Jenomže při načítání si nevyhradí místo a způsobí, že dokument po načtení obrázku poskočí:

```

```

Zadají-li se jiné rozměry, obrázek bude deformován, ale rozměry v dokumentu dodrží:

```

```

V případě, že zadáte do width nebo height nulové hodnoty, Internet Explorer stejně vykreslí 1px velký obrázek. Chcete-li obrázek nulové velikosti, zadejte mu rozměry pomocí CSS stylů [width](#) a [height](#).

Border

Tloušťka rámečku. Lze zadat nula, pak nebude obrázek orámečkován. Různé prohlížeče se liší v tom, zda zobrazují rámeček u obrázku, u kterého border nebyl zadán. Proto komu záleží na přesném vzhledu, měl by border vždy zadávat. Poslední dobou se ale doporučuje rámeček spíše zadávat css [stylem border](#).

Příklad nejčastějšího použití: je-li obrázek v odkazu, přirozeně se kolem něj tvoří barevný rámeček:

```
<a href="stranka.html"></a>
```

Nejprostším řešením je zadat obrázku nulový okraj border="0":

```
<a href="stranka.html"></a>
```

V moderní praxi se tento postup už doporučuje nahradit spíše globálním stylem:

```
<style>
a img {border-width: 0px;}
</style>
```

Další údaje [o odkazech v HTML](#).

Vspace, hspace

Vertikální a horizontální okraje, prostě prázdné místo okolo. Zadává se v pixelech (tedy bez hodnoty a znamená to pixely), nebo v procentech. Procenta se počítají z rozměru obrázku. V praxi se zastaralé atributy vspace a hspace nahrazují [CSS stylem margin](#).

Align

Zarovnání obrázku s okolím.

Tag , atribut align			
	align=	Význam	Nahradíte CSS zápisem
Obtékání obrázku	left	Obrázek je umístěn k levému okraji a obtékán zprava.	float: left;
	right	Obrázek je umístěn k pravému okraji a obtékán zleva.	float: right;
Vertikální umístění obrázkuna řádku	top	Vršek obrázku je zarovnán s nejvyšším bodem libovolného objektu řádku.	vertical-align: top;
	texttop	Vršek obrázku je zarovnán s nejvyšším bodem textu (většinou je to totéž, co top).	vertical-align: text-top
	middle	Střed obrázku je zarovnán s účařím písma řádku.	-
	absmiddle	Střed obrázku je zarovnán se středem řádku.	vertical-align: middle;
	baseline	Spodek obrázku je zarovnán na účaři.	vertical-align:

			text-bottom;
	bottom	Spodek obrázku je zarovnán na spodek písma	-
	absbottom	Spodek obrázku bude zarovnán na nejnižší místo řádku.	vertical-align: bottom;

Osobně považuji milión hodnot pro umístění na řádku za zbytečnost a nepoužívám je. Má to smysl pouze u malých obrázkových matematických vzorců. Velký obrázek se v praxi většinou nechává obtékat nebo umísťuje do samostatného odstavce.

Obtékané obrázky se vertikálně nacházejí **pod** řádkou, v níž jsou uvedeny (to je trochu nešikovné, ale logické). Jedinou výjimkou je, kdy obtékaným obrázkem řádka začíná - pak je vrchol obrázku v úrovni vrcholu řádky.

Atribut align se dá nahradit [css stylem float](#) (pro hodnoty obtékání right a left) a css stylem [vertical-align](#).

Chcete-li obrázek umístit na střed, přes atribut align to nejde. Obalte obrázek nějakým blokovým prvkem (třeba tagem <div>) a zarovnání na střed zadejte tomuto obalovacímu prvku.

Style

Všechny atributy kromě src, alt a usemap se dají nahradit CSS deklarací pomocí obecného atributu [style](#).

Příklad: Následující dva obrázky se zobrazí stejně:

```

```

```

```

Usemap

Přítomnost tohoto atributu dělá z obrázku klikací mapu. Hodnotou atributu je křížek # plus jméno mapy. Mapa je element vyznačený tagem <map> obsahující definici aktivních oblastí. Popis mapy je umístěn v tomtéž HTML dokumentu. Jméno mapy je určeno atributem name u tagu <map>. Příklad viz u popisu tagu [<map>](#).

Tag <map> kdysi mohl být umístěn i v jiném souboru. Fungovalo to v Netscape 3, dnes to nefunguje. Hodnotou atributu usemap bývalo jméno souboru s křížkem # a jménem mapy.

Ismap

Pokud je obrázek v odkazu, při kliknutí se za adresu připojí otazník a souřadnice kliknutí oddělené čárkou. Umějí to zpracovávat některé serverové skripty. Atribut ismap je považován za zastaralý a v praxi se nepoužívá.

Longdesc

Dlouhý popis se používá málo, protože na to autoři stránek kašlou, a tak na to kašlou i prohlížeče. Do tagu se dá dát kromě atributu alt, který má obsahovat popis obrázku, také atribut longdesc, který odkazuje na jiné URL, na kterém je obrázek popsán. Podporuje to Opera a některá rozšíření jiných prohlížečů (zobrazují možnost přechodu na to jiné URL v kontextovém menu).

Nestandardní atributy

Přidá-li se od html tagu atribut **galleryimg** s hodnotou false (tedy), prohlížeč Internet Explorer nebude po přejetí obrázku myší zobrazovat takové ty tři hnusné ikonky (uložit vytisknout a ještě něco). Na Firefox ani jiné prohlížeče to vliv nemá.

Dalším nestandardním atributem je **dynsrc**. Pokud se v tagu místo atributu **src** uvede tag **dynsrc**, jde o video. Jde pravděpodobně o velmi nestandardní rozšíření Internet Exploreru, většinou nefunguje.

Atributy tagu		
Atribut	Význam	Hodnoty
dynsrc	adresa videa	URL
loop	počet přehrávání	číslo nebo "infinite"
loopdelay	prodleva mezi přehráváními	čas v milisekundách
start	kdy se má začít přehrávat	"mouseover" -- při přejetí myši, jinak po natažení stránky
controls	zobrazení ovládání	atribut bez hodnoty (Netscape nepodporuje)

Prohlížeče podporují takové video formáty, na které mají v operačním systému kodeky. Trochu lepší podporu videa lze dosáhnout pomocí [objektů](#).

Map

Klikací mapa obrázku. V dokumentu se tento element nijak nezobrazuje, pouze způsobuje, že obrázek bude ně některých místech klikat jako odkaz. Mapa obsahuje popis aktivních oblastí, které jsou reprezentovány tagy <area>. Pomocí jména -- atributu **name** -- je mapa spojena s obrázkem (atribut usemap u). Name může být libovolný název. Jedna mapa může být teoreticky přiřazena k více obrázkům, nenapadá mě ale situace, kdy je to užitečné.

Příklad: <map name="ctverecky">

<area href="prvni.htm" shape="rect" coords="80, 80, 130, 130">

<area href="druha.htm" shape="rect" coords="20, 20, 50, 50">

</map>

Area

Aktivní oblast v mapě. Nepárový tag. Nezobrazuje se. Může se vyskytovat pouze v elementu <map>. Area vymezuje oblast obrázku, na kterou se dá kliknout jako na odkaz. V jednom elementu <map> může být neomezeně elementů <area>, každá pro jednu aktivní oblast.

Atributy tagu <area>		
Atribut	Význam	Hodnoty
href	cíl odkazu	URL
target	cílový rám	jméno rámu
shape	tvar oblasti	rect, circle, polygon

coords	souřadnice oblasti	čísla oddělená čárkami
nohref	oblast není odkazem	bez hodnot
alt	alternativní text	libovolný text

Atributy [href](#) a [target](#) jsou stejné jako u [odkazů](#). Z [obecných atributů](#) vypichuji atribut [title](#), který v novějších prohlížečích zobrazuje u myši nad oblastí bublinu s textem (podobně jako [alt](#) u obrázků). Atribut **alt** takto nefunguje, ten má význam pouze pro negrafické klienty.

Atribut **shape** určuje tvar oblasti, podle toho se potom mění význam hodnot atributu **coords**. Ten obsahuje souřadnice [v pixelech](#), přičemž počátek souřadnic je levý horní roh mapy (tedy obrázku) a y roste směrem dolů.

Tag <area>, atributy shape a coords			
shape=	Význam	zadáva se	coords=""
rect	obdélník	levý horní roh, pravý dolní roh	lh-x, lh-y, pd-x, pd-y
circle	kruh	střed, poloměr	střed-x, střed-y, poloměr
polygon	mnohoúhelník	souřadnice každého vrcholu za sebou	vrchol1-x, vrchol1-y, vrchol2-x, vrchol2-y, atd.

Jsou možné i alternativní hodnoty circ, poly a rectangle se stejnou funkcí. Pokud se dvě oblasti překrývají, je ta pozdější jakoby vespod (dlouho jsem zde měl chybnou informaci, že je navrchu). Jinak řečeno v překrývaných kouscích vyhrává ta area, která je první.

Atribut nohref byl vymyšlen pro tvorbu tvarů, které nikam neklikají. To používá např. pro tvorbu klikacích mezikruží, kdy větší circle někam kliká. Uvnitř většího kruhu je menší (v kódu uvedený jako první) s atributem nohref.

HTML zápis klikacích map je poněkud složitý. Naštěstí každý lepší HTML editor zvládá jejich zápis tak, že autorovi pouze stačí šoupat myší.

Seznamy

Tag	Význam	Párový	Výskyt
li	položka seznamu	nepovinně	, <menu> nebo <dir>
ol	číslovaný seznam	ano	
ul	odrážkový seznam	ano	
dir	zvláštní druh seznamu	ano	
menu	typ seznamu	ano	
dl	seznam definic	ano	
dt	definovaný termín	ano	<dl>
dd	definice termínu	ano	<dl>

li

Položka seznamu (z angl. list item). Pro správné zobrazení musí být uvnitř seznamu -- to znamená uvnitř elementu , , <dir> nebo <menu>.

Zobrazuje se vždy na novém řádku za číslem (pokud jde o) nebo za odrážkou (v ostatních případech). Mezi jednotlivými položkami nejsou zvětšené vertikální mezery.

- Ačkoli se jedná o tag, který se nemusí zavírat, doporučuji v rámci neprasení psát i koncové . Chybí-li koncové , prohlížeč si ho domyslí, jakmile uvidí konec seznamu (, apod.), nebo další .

Atributy tagu

Atribut	Význam	Hodnoty	Význam hodnot
type	druh odrážky	disc circle square	puntík kolečko čtvereček
	druh číslování	1 A a I i	normální číslování velké písmenkování malé písmenkování římské číslice malé římské číslice

value	změna čísla	číslo	číslo položky (převedené podle typu číslování) pozor, dříve jsem zde měl uveden atribut start, který u li nefunguje
-------	-------------	-------	--

ol

Číslovaný seznam (ordered list). Uvnitř se smějí vyskytovat pouze elementy (číslované řádky). (V zanořeném tagu se ale pak mohou objevovat další tagy.) Čísla k položkám samozřejmě přidá prohlížeč, o to není třeba se starat.

Například takto v HTML kódu vypadá číslovaný seznam se dvěma položkami:

```
<ol>
  <li>První položka</li>
  <li>Druhá položka</li>
</ol>
```

 má atributy **type** a **start**, které mají stejný význam, jako type a value u , případné nastavení téhož u jednotlivých má ale větší prioritu.

Příklad: pokud nastavím různé start a value, tak vyhraje ta pětka u :

```
<ol start="3">
  <li value="5">před tímto textem bude pětka</li>
</ol>
```

Zanořením dalšího tagu do elementu se dá udělat víceúrovňové číslování s větším levým odsazením.

Před a po celém seznamu se udělá vertikální mezeru velikosti jednoho řádku. V tomto se seznamy podobají odstavcům (<p>). Levé odsazení se zobrazuje zpravidla ve velikosti 40 pixelů, čímž se seznamy podobají tagu <blockquote>.

ul

Nečíslovaný seznam (unordered list). Uvnitř mohou být pouze elementy .

Před položkami se jako odrážka zobrazí puntík, lze to ale změnit atributem **type** (hodnoty stejné jako u). Víceúrovňové odrážky se dají dělat zanořováním dalšího do . Druhá úroveň zobrazuje přednastaveně kolečko, třetí čtvereček.

Seznamy s obrázkovou odrážkou lze udělat pomocí CSS stylů a vlastosti list-style-image.

Příklad:

```
<ul style="list-style-image: URL('puntik.gif')">
  <li>položka s obrázkovou odrážkou</li>
</ul>
```

dir, menu

Tyto tagy funkcí ani zobrazením neumím odlišit od , zobrazují a chovají se totiž ve všech směrech naprosto stejně.

Další tagy se týkají definičních výčtů

dl

Seznam definic (definition list). Obsahuje na střídačku tagy <dt> a <dd>, což by měl být termín a jeho vysvětlení.

Definice se dnes v HTML používají velmi zřídka. Ne každý autor totiž píše encyklopedie.

Jediný atribut **compact** způsobuje, že pokud to jde, zobrazuje se <dd> na stejném řádku jako <dt>. (V Exploreru funguje, ale dd musí začínat prostým textem, nikoli dalším tagem). Atribut compact je zastaralý a doporučuje se nepoužívat jej.

Příklad správně strukturovaného definičního výčtu:

```
<dl>
  <dt>První termín</dt>
  <dd>Výklad prvního termínu</dd>
  <dt>Druhý termín</dt>
  <dd>Výklad druhého termínu</dd>
</dl>
```

Další informace o [definicích v HTML](#).

dt

Definovaný termín (defined term). Může se vyskytovat pouze uvnitř elementu <dl>. Zobrazuje se vždy na novém řádku u levého okraje stránky.

dd

Definice předchozího termínu. Může se vyskytovat pouze uvnitř elementu <dl>. Zobrazuje se s levým odsazením. Mezi <dt> a <dd> není žádná vertikální mezera. Klasická velikost odsazení <dd> je 40 pixelů, ale dá se měnit pomocí CSS vlastnosti [margin-left](#).

Seznam

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2//EN">
<!--MYCHARSET=CP1250-->

<HTML>
<HEAD>
<TITLE>Vnořené seznamy</TITLE>
</HEAD>

<BODY>
Pro tvorbu stránek v jazyce HTML můžeme použít:
<OL>
  <LI>Obyčejné textové editory
    <UL>
      <LI>Notepad
      <LI>Programmer's File Editor
      <LI>Emacs
    </UL>
  <LI>Textové editory podporující vytváření struktury HTML
    dokumentu
    <UL>
      <LI>HotMetal Pro
      <LI>asWedit
    </UL>
  <LI>WYSIWYG editory HTML stránek
    <UL>
      <LI>FrontPage
      <LI>Netscape Editor
      <LI>WebMagic Pro
    </UL>
</OL>
<script src="http://www.google-analytics.com/urchin.js" type="text/javascript">
</script>
<script type="text/javascript">
_uacct = "UA-1085576-1";
urchinTracker();
</script></body>
</HTML>
```