



15

HTML a kaskádové styly

VÝUKOVÝ TEXT PRO STUDENTY INFORMAČNÍCH TECHNOLOGIÍ

Marek Lučný, SŠPU OPAVA 2015

HTML (Hypertext Markup Language = hypertextový značkovací jazyk) je speciální jazyk pro publikování hypertextových dokumentů označovaných jako webové stránky. Za hypertext považujeme text, který obsahuje propojení (odkazy) na jiné texty. Označení značkovací jazyk souvisí se strukturováním dokumentu pomocí značek (tagů).

První verzi jazyka HTML publikoval zakladatel služby WWW Tim Berners Lee v roce 1991. Tato verze umožňovala vkládat do textu obrázky, hypertextové odkazy, vytvořit několik logických úrovní a několik druhů zvýraznění. Byla označena jako HTML 0.9. Požadavky uživatelů se postupně zvyšovaly a promítaly se do vývoje následujících verzí HTML.

Nástupcem velmi rozšířené verze HTML 4.01, která vycházela z metajazyka SGML (Standard Generalized Markup Language), se stal jazyk XHTML, jehož základem je modernější metajazyk XML (eXtensible Markup Language, rozšiřitelný značkovací jazyk). Existují tři různé definice jazyka XHTML: strict (přísná), transitional (přechodná) a frameset (s podporou frames - rámců). Přísná definice vychází ze zásad XML a vyžaduje přísné dodržování pravidel: značky i atributy mají být zapisovány výhradně malými písmeny; všechny značky jsou považovány za párové, a i ty původně nepárové proto musejí být zakončeny lomítkem (např. `
`); všem atributům musí být přiřazena hodnota, která je zapisována mezi uvozovky apod.

Přestože se původně připravovala specifikace XHTML 2.0, iniciativa zejména tvůrců významných prohlížečů vedla ke vzniku současné verze HTML 5. Ta definitivně ukončila závislost na SGML, opravila mnoho chyb předešlých verzí, vyřadila řadu zastaralých a nepoužívaných prvků a zavedla nové sémantické prvky. Zároveň přibyla podpora moderních technologií (multimédia), byl zaveden nový systém vývoje jazyka.

O výběru definice HTML rozhoduje speciální element DOCTYPE, který se uvádí na začátku kódu stránky a v případě XHTML obsahuje odkaz na příslušný DTD (Document Type Definition). DTD představuje návod, jak má dokument, napsaný ve značkovacím jazyce, interpretovat prohlížeč (tj. např. jaké elementy dokument používá, jak s nimi zacházet, jaký význam mají jednotlivé atributy apod.). V případě XHTML Strict vypadá zápis elementu DOCTYPE následovně:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//
EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.
dtd">
```

V případě HTML 5 je zápis mnohem jednodušší:

```
<!DOCTYPE html>
```

Definiční dokument je umístěn na serveru konsorcia W3C (World Wide Web Consortium), které má na starost definování standardů služby WWW.

V HTML se používají speciální značky – tagy. Tagy jsou tvořeny znaky „<“ a „>“, mezi něž se zapisuje název tagu (<tag>). Vše ostatní, co není mezi těmito znaky, se zobrazuje jako výsledný text na stránce. Značky určují, jaký má text význam (např. jestli se jedná o nadpis, tabulku, či hypertextový odkaz). V případě párových značek musí ke každému počátečnímu tagu existovat tag ukončovací – ten se liší od počátečního tím, že před názvem tagu obsahuje lomítko („/“).

```
<p>Text zobrazovaný jako odstavec.</p>
```

Nepárové značky, které neobklopují žádný text ani jiný obsah, se mohou zapisovat jen pomocí počátečního tagu (týká se to např. značky pro zobrazení obrázku nebo značky <hr>, která tvoří horizontální čáru). Pouze v XHTML musí být i tyto značky aspoň symbolicky uzavřeny zápisem <tag />.

Tagy mohou být vnášeny do jiných tagů, ale musí být zachován jejich hierarchický sled, tedy např.:

```
<b><i>tučná kurzíva</i>, pouze tučné písmo</b>
```

Z hlediska významu je možné značky rozdělit na tři základní skupiny:

Strukturální značky. Definují strukturu dokumentu (např. <p>, <h1>, <h2>).

Sémantické značky. Popisují povahu obsahu prvku (<title>, <address>, <article>). Trendem je orientace právě na sémantické značky, které usnadňují automatizované zpracovávání dokumentů a vyhledávání informací na webu.

Stylistické značky. Určují vzhled prvku při zobrazení (např. , <i>). Od tohoto druhu značek se postupně upouští, trendem je používání kaskádových stylů, které vzhled popisují odděleně od obsahu dokumentu. Tyto značky jsou příliš orientovány na prohlížení na obrazovce počítače a nepočítají s používáním alternativních prohlížečů pro postižené (čtečky pro slepce), mobilních zařízení apod. Kaskádové styly umožňují přizpůsobení zobrazení pro různá zařízení.

ATRIBUTY

Atributy přiřazují nějaké vlastnosti určitému elementu. Atributy se píšou po počátečním tagu, může jich být i více (oddělují se mezerou). Atribut má svou hodnotu, jež bývá v zapsána v uvozovkách (zdvojených nebo jednoduchých).

ELEMENTY

Elementem nazýváme celou sekvenci počínaje počátečním tagem a konče tagem ukončovacím. Existují tři základní druhy elementů: blokové, inline a nahrazované. Podle významu, který textu přiřazují, je můžeme rozdělit ještě na elementy pro strukturování dokumentu, textové elementy, elementy pro tvorbu odkazů, elementy pro tvorbu tabulek, elementy pro tvorbu seznamů apod.

Blokové elementy. Tvoří nějaký blok – po takovém elementu je text dokumentu zalomen (odřádkován). Blokovými elementy jsou například nadpis (h1, h2, h3 ...), odstavce (p), oddíly (div), seznamy (ul, ol, dl), tabulky (table) apod.

Inline elementy. Nacházejí se uvnitř textu a nedochází po nich k zalomení. Obvykle plní funkci zvýraznění nějaké části textu (i, em, b, strong, span ...), nebo představují např. hypertextový odkaz (a).

Nahrazované elementy. Jsou nahrazeny nějakým obsahem, pro začlenění dokumentu jsou důležité jejich rozměry. Např. obrázkové elementy (img), vnitřní rámce (iframe) apod.

```
<element atribut="hodnota_atributu">
```

K nejčastěji používaným atributům patří:

Atribut	Význam
style	Definuje přímo vložený styl pro obsah značky.
id	Definuje pro značku unikátní identifikátor.
class	Definuje tzv. třídu - skupinu elementů s určitými vlastnostmi.
title	Titulek, který se nad elementem zobrazuje při najetí myši.
src	Udává URI adresu (cestu) k externímu dokumentu - obrázku, skriptu, multimediálnímu souboru.

ENTITY

Entity vyjadřují pomocí sekvence znaků jeden určitý znak nebo symbol, který je obvykle z určitého důvodu netypický (nelze ho jednoduše zapsat pomocí klávesnice jako např. © symbol pro copyright) nebo problematický (pomocí entit zapisujeme znaky < jako < nebo > jako >, aby nedocházelo k záměně mezi symboly značek). K dodržování typografických pravidel můžeme použít entitu (nezlomitelná mezera). Pomocí entit s číselnými kódy můžeme zadat jakýkoliv znak z kódovací tabulky - např. A vypíše A.

VALIDITA

Validitu kódu webové stránky ověřujeme pomocí speciálního programu - validátoru. Online validátor W3C se nachází na adrese validator.w3.org a nabízí možnosti validace zadáním URI stránky, uploadem souboru nebo přímým vložením kódu HTML. Výsledkem validace je informace o platnosti kódu stránky – stránka je buď validní s bezchybným kódem, nebo se zobrazí přehled chyb a varování s upřesňujícími informacemi (umístění chyby na stránce, charakter chyby, možná řešení apod.). Nevalidní kód může způsobit nesprávné zobrazení stránky; důležitější je však validita pro správnou interpretaci stránky např. hlasovou čtečkou nebo vyhledávacím robotem (může stránku chybně zaindexovat, nebo nezaindexovat vůbec – neobjeví se proto ve vyhledávacích).

Dokument v jazyku HTML má předepsanou strukturu, kterou tvoří:

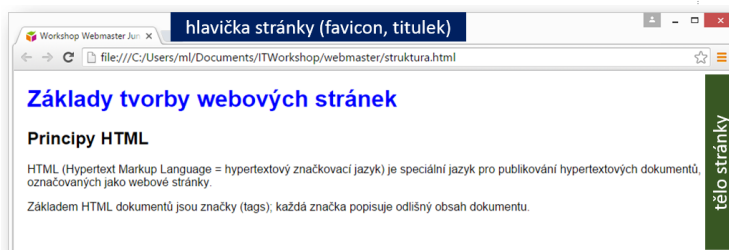
Deklarace typu dokumentu `<!DOCTYPE html>` sděluje prohlížeči, že otevřel HTML dokument; validátor pomocí této deklarace rozlišuje verzi HTML.

Kořenový element `html` (značky `<html>` a `</html>`) reprezentuje celý dokument.

META TAGY

Nepárové značky `<meta>` v hlavičce stránky mají význam pro správné zobrazení stránky v prohlížečích a efektivní zaindexování stránky vyhledávacími stroji. Meta značek je celá řada - kromě těch, obsahujících informace pro vyhledávače (např. `description`, `keywords`, `robots`), se zejména v minulosti využívaly např. meta tagy pro automatické přesměrování či obnovu stránky po určitém čase (`<meta http-equiv="refresh" content="6;URL=http://www.nekam.cz/cokoliv">`) a podobné. Význam některých meta značek se postupně vytratil, některé meta značky jsou ignorovány současnými prohlížeči i vyhledávači. Důležité je stále nastavení správného kódování znaků, které stránka používá (např. `<meta charset="UTF-8">`). Moderní stránky s podporou responzivního designu (přizpůsobení zobrazení podle použitého zařízení) často obsahují meta značku `<meta name="viewport" content="width=device-width,initial-scale=1">`. Slovo `viewport` označuje plochu aktuálního zobrazovacího zařízení. Atribut `width` naznačuje prohlížeči, že má stránku rozšířit podle velikosti této plochy. To `initial-scale=1` znamená, že se při načtení na mobilu stránka nemá zmenšovat.

Struktura stránky



```
1 <!DOCTYPE html>
2 <html lang='cs'>
3   <head>
4     <title>Workshop Webmaster Junior</title>
5     <meta charset='utf-8'>
6     <meta name='description' content='Výukové stránky o jazyku HTML 5'>
7     <meta name='keywords' content='HTML, web, výuka'>
8     <meta name='author' content='Marek Lučný'>
9     <meta name='robots' content='all'>
10    <link href='favicon-sw.png' rel='shortcut icon' type='image/png'>
11    <link href='css/styly.css' rel='stylesheet' type='text/css'>
12  </head>
13  <body>
14    <h1>Základy tvorby webových stránek</h1>
15    <h2>Principy HTML</h2>
16    <p>HTML (Hypertext Markup Language = hypertextový značkovací jazyk)
17      je speciální jazyk pro publikování hypertextových dokumentů,
18      označovaných jako webové stránky.</p>
19    <p>Základem HTML dokumentů jsou značky (tags);
20      každá značka popisuje odlišný obsah dokumentu.</p>
21  </body>
22 </html>
```

1 Typ HTML dokumentu (zde HTML 5)
 2 Začátek HTML dokumentu, jazyk
 3 Začátek hlavičky stránky
 4 Titulek stránky
 5 Kódování stránky (zde UTF-8)
 6 Popis obsahu stránky
 7 Klíčová slova
 8 Jméno autora stránky
 9 Přístupné pro vyhledávací roboty
 10 Odkaz na favicon v liště okna
 11 Odkaz na připojený CSS dokument
 12 Konec hlavičky stránky
 13 Začátek těla stránky
 14 Hlavní nadpis stránky
 15 Podnadpis
 16 Začátek prvního odstavce
 17 ... text odstavce ...
 18 Konec prvního odstavce
 19 Druhý ...
 20 ... odstavce
 21 Konec těla stránky
 22 Ukončení HTML dokumentu

Hlavička dokumentu `head` (značky `<head>` a `</head>`) obsahuje metadata, která se vztahují k celému dokumentu. Definuje např. kódování, název dokumentu, autora, popis, klíčová slova, titulek dokumentu nebo kaskádové styly.

Tělo dokumentu `body` (značky `<body>` a `</body>`) zahrnuje vlastní obsah dokumentu.

KOMENTÁŘE

Součástí HTML kódu mohou být rovněž komentáře, které se zapisují:

```
<!-- Toto je komentář na HTML stránce -->
```

Takto zakomentované části prohlížeče nezobrazují.

Základní blokové elementy

Nadpisy. V HTML může být použito šest úrovní nadpisů (headings), které mohou zároveň tvořit osnovu dokumentu. Na stránce by se měl vyskytovat pouze jeden hlavní nadpis (`<h1>`); někdy bývá součástí tohoto elementu logo stránky. Rozdělení dokumentu pomocí nadpisů různých úrovní je důležité zejména s ohledem na vyhledávací stroje, které obsah nadpisů využívají k indexování.

Odstavce a oddíly. Párový element `<p>` (odstavec) by měl být používán k členění delšího textu do samostatných odstavců. K oddělování odstavců by neměla být používána nepárová značka `
`, která je určena k prostému zalomení řádku ve specifických případech (např. na koncích veršů, při úpravě adresy atd.). Element `<div>` (oddíl) slouží jako kontejner pro další HTML elementy, může obsahovat nadpisy, odstavce, tabulky, seznamy, ale i další vnořené oddíly. Velmi často se využívá pro rozdělení stránky na samostatné oblasti a ve spojení s CSS k definování layoutu stránek (viz níže).

Seznamy. Součástí jazyka HTML jsou značky pro vytvoření tří různých typů seznamů - uspořádaného (``, ``), neuspořádaného (``, ``) a popisného (`<dl>`, `<dt>`, `<dd>`).

Seznamy

Neuspořádaný seznam	Uspořádaný seznam	Seznam pojmů a definic
<ul style="list-style-type: none"> Červená Modrá Zelená 	<ol style="list-style-type: none"> Zlata Stříbrná Bronzová 	<p>HTML Hypertext Markup Language</p> <p>CSS Cascading Style Sheets</p>
<pre><h2>Neuspořádaný seznam</h2> Červená Modrá Zelená </pre>	<pre><h2>Uspořádaný seznam</h2> Zlata Stříbrná Bronzová </pre>	<pre><h2>Seznam pojmů a definic</h2> <dl> <dt>HTML</dt> <dd>Hypertext Markup Language</dd> <dt>CSS</dt> <dd>Cascading Style Sheets</dd> </dl></pre>

definující záhlaví (`<thead>`), tělo (`<tbody>`) a zápatí (`<tfoot>`) tabulky. K rozdělení tabulky na řádky používáme element `<tr>`, jednotlivé buňky tabulky tvoří pomocí značek `<td>` (jedná-li se o buňky obsahující data) nebo `<th>` (pokud jde o buňky se záhlavím). Pomocí atributů `colspan` a `rowspan` můžeme zajistit přesah buněk přes více sloupců respektive řádků.

Rozvrh hodin

Den	1	2	3	4
	8:00 - 8:45	8:50 - 9:35	9:45 - 10:30	10:50 - 11:35
Pondělí	ICT		MAT	ČJ
Školní rok 2015/2016				

Tabulky

```
<h2>Rozvrh hodin</h2>
<table>
```

```
<thead>
  <tr>
    <th rowspan="2">Den</th>
    <th>1</th>
    <th>2</th>
    <th>3</th>
    <th>4</th>
  </tr>
  <tr>
    <th>8:00 - 8:45</th>
    <th>8:50 - 9:35</th>
    <th>9:45 - 10:30</th>
    <th>10:50 - 11:35</th>
  </tr>
</thead>
<tbody>
  <tr>
    <td colspan="2">ICT</td>
    <td></td>
    <td>MAT</td>
    <td>ČJ</td>
  </tr>
</tbody>
<tfoot>
  <tr>
    <td colspan="5">Školní rok 2015/2016</td>
  </tr>
</tfoot>
</table>
```

Důležité značky

`<table>` = tabulka
`<tr>` = řádek tabulky
`<th>` = buňka v záhlaví
`<td>` = datová buňka

Kaskádové styly

```
<style>
table {
  width: 500px;
  border-spacing: 0;
  border-collapse: separate;
  border: 2px solid black;
}
thead {
  background-color: #fdd;
}
tfoot {
  background-color: #ddf;
}
th, td {
  border: 1px solid gray;
  padding: 10px;
  text-align: center;
}
</style>
```

Tabulky. Strukturu tabulky může tvořit hned několik párových elementů. Základním je element `<table>`, do něhož mohou být vnořeny nepovinné elementy

INLINE ELEMENTY

Inline elementy jsou vkládány do blokových elementů (kontejnerů), protože obvykle ohraničují jen kratší úsek textu. Univerzálním inline elementem je značka ``, která se používá na těch místech, kde je potřeba změny formátování.

Starší verze HTML obsahovaly řadu značek (např. ``, `<center>`, `<big>` a jiné), ale i atributů (color, border, background) které se používaly k formátování textu. Přestože současné prohlížeče kvůli zpětné kompatibilitě stále ještě větší podobné úpravy zobrazují, jejich použití v HTML kódu není vhodné. HTML kód by měl obsahovat především sémantické značky, které definují význam určité části stránky, zatímco grafické úpravy textu by měly být výhradně přesunuty do kaskádových stylů. Například pro zvýraznění důležitého pojmu by měl být použit spíše element `` než element ``.

Tag	Význam	Tag	Význam
<code></code>	tučný text	<code><sub></code>	dolní index
<code></code>	zvýrazněný text	<code><sup></code>	horní index
<code><i></code>	kurzíva	<code><ins></code>	vložený text
<code><small></code>	zmenšený text	<code></code>	vymazaný text
<code></code>	významný text	<code><mark></code>	označený text

ELEMENTY PRO CITACE

Tag	Význam
<code><abbr></code>	Definuje zkratku nebo akronym
<code><address></code>	Definuje kontaktní adresu
<code><bdo></code>	Určuje směr text (např. <code><bdo dir="rtl"></code> - zprava doleva)
<code><blockquote></code>	Vyznačuje víceřádkový citovaný text
<code><cite></code>	Definuje název díla (např. <code><cite>Krakatit</cite></code>)
<code><q></code>	Vyznačuje kratší citaci, přidává uvozovky

ELEMENTY PRO VÝPIS POČÍTAČOVÉHO KÓDU

Tag	Význam
<code><code></code>	Slouží k výpisu programového kódu
<code><kbd></code>	Používá se k označení vstupu z klávesnice
<code><samp></code>	Slouží k vyznačení počítačových výstupů
<code><var></code>	Používá se k zvýraznění proměnných
<code><pre></code>	Předformátovaný text, včetně mezer, konců řádků...

NOVÉ SÉMANTICKÉ ZNAČKY V HTML5

Ve verzi HTML5 byly přidány některé nové značky, které mají omezit nadužívání značky `<div>` a v duchu tzv. sémantického webu označovat obvyklé bloky webové stránky podle charakteru jejich obsahu:

Tag	Význam
<code><article></code>	Označuje začátek a konec článku (např. v blogu)
<code><aside></code>	Vyznačuje postranní blok na stránce
<code><details></code>	Obsahuje detailní informace (primárně skryté)
<code><figcaption></code>	Definuje popis textu vztahovaný k elementu <code><figure></code>
<code><figure></code>	Specifikuje přílohy v podobě fotek, diagramů, kódů...
<code><footer></code>	Vymezuje zápatí dokumentu nebo sekce
<code><header></code>	Vymezuje záhlaví dokumentu nebo sekce
<code><main></code>	Specifikuje hlavní část obsahu
<code><nav></code>	Obsahuje navigaci v podobě odkazů
<code><section></code>	Vymezuje určitou sekci v rámci HTML stránky
<code><summary></code>	Viditelné záhlaví elementu <code><details></code>
<code><time></code>	Obsahuje časový údaj (datum/čas)

Odkazy jsou klíčovými prvky webových stránek, protože umožňují jejich vzájemné provázání. Odkaz se vytváří pomocí značky `<a>`:

```
<a href="http://validator.w3c.org"
target="_blank">W3C</a>
```

Povinnou částí je atribut href, který obsahuje URL adresu, na níž odkaz směřuje. Atribut target specifikuje způsob zobrazení výsledku; hodnota `"_blank"` znamená, že odkazovaná stránka bude otevřena v nové okně.

Absolutní odkazy

Používají se k odkazování na dokumenty umístěné na jiném serveru. Obsahují úplnou URL adresu:

```
<a href="http://w3schools.com/links.asp">Odkaz</a>
```

Relativní odkazy

Odkazují na dokumenty umístěné na stejném serveru nebo disku. Takové odkazy nemusí obsahovat kompletní URL adresu, ale relativní umístění vzhledem k výchozí stránce:

```
<a href="../sport/index.html">Relativní odkaz</a>
```

Odkazy na záložky

Odkazovat je možné i na určité vyznačené místo HTML dokumentu. Je-li například na stejné stránce použito označení určité sekce pomocí id takto:

```
<section id="zpravy">... obsah sekce ...</section>
```

je možné pro přesun na místo začátku této sekce použít odkaz:

```
<a href="#zpravy">Odkaz na záložku zpravy</a>
```

Součástí odkazu nemusí být pouze text, ale také obrázek:

```
<a href="../sport/index.html"></a>
```

NAVIGACE

Hlavní navigace bývá nejvýraznější navigační strukturou na každé stránce. Většinou má podobu horizontálního pruhu nebo je umístěna vertikálně po straně hlavního sdělení. V případě členitějšího webu bývá hlavní navigace doplněná o vysouvací menu. Na velmi rozsáhlých webech lze v rámci hlavní navigace ještě rozlišit navigaci globální (umožňuje přecházení mezi hlavními kategoriemi webu) a lokální (upozorňuje na další stránky dané kategorie).

Hierarchická navigace (též stromová navigace) vychází z principu větvení struktury webu na nižší kategorie. Uživatel postupně prochází stromem shora dolů a v každém uzlu vybírá nejvhodnější podkategorii. Používá se zejména na velmi rozsáhlých webech, např. v internetových obchodech.

Kontextovou navigaci tvoří obvykle seznam souvisejících odkazů, které jsou připojeny na konci stránky, někdy se používá i kontextová navigace v podobě panelů umístěných po straně stránky. Výborně se hodí k vytváření zkratk skrz web, umožňuje poslat návštěvníka přímo k informacím, které hledá, nezávisle na hlavní navigaci a struktuře webu.

Drobečková navigace (anglicky Breadcrumb Navigation) představuje seznam odkazů na nadřazené sekce, odkazy jsou odděleny šipkou či lomítkem. Název drobečkové navigace pochází z pohádky o Jeníčkovi a Mařence, ve které Jeníček odhazuje po cestě lesem drobečky, které mu mají umožnit cestu zpátky domů (v tomto případě návrat na homepage).

Úvod > Znalosti > Slovníček > Drobečková navigace

Úkolem drobečkové navigace je ukázat návštěvníkům, kde se v hierarchii webu právě nacházejí a její význam stoupá s rostoucím rozsahem a složitostí webu. Obvykle je umístěna v hlavičce stránky, nejlépe poblíž hlavního nadpisu.

Alternativními způsoby navigace jsou **mapa webu** (zpravidla hierarchicky uspořádané odkazy na všechny dostupné stránky webu) a **fulltextové vyhledávání**.

OBRÁZKY V HTML

Na webovou stránku je možné vkládat různé typy grafických souborů. Nejčastěji to jsou rastrové obrázky ve formátech JPEG, PNG nebo GIF (ty mohou obsahovat i animace). Používáme k tomu značky ``:

```

```

Povinné jsou atributy `src` (URL adresa grafického souboru) a `alt` (alternativní popis obrázku, který je použit tehdy, kdy obrázek nemůže být zobrazen; využívají toho např. hlasové čtečky stránek). Rozměry obrázku mohou být nastaveny atributy `width` a `height`, nebo pomocí kaskádových stylů. Pokud zadáme pouze jeden z rozměrů, bude druhý rozměr automaticky přizpůsoben tak, aby byl zachován správný poměr stran.

OBRAZOVÉ MAPY

Jedná se o obrázek, na který jsou pomocí souřadnic „namapovány“ aktivní geometrické oblasti spojené s odkazy. Po kliknutí do některé z oblastí může být provedena určitá akce - změna stránky, javascriptová funkce:

```

<map name="mapa">
  <area shape="rect" coords="0,0,82,126" alt="Asie"
href="asie.html">
  <area shape="circle" coords="90,58,3" alt="Evropa"
href="evropa.html">
</map>
```

PODPORA MULTIMÉDIÍ V HTML5

Webové stránky mohou obsahovat také nejruznější multimediální soubory - vedle běžných obrázků (fotek) také videa, animace nebo zvuky. Ve starších

prohlížečích a dřívějších variantách HTML byla podpora přehrávání multimédií omezená a velmi odlišná. Rozsáhlá podpora multimédií se stala standardem až v HTML5. Příkladem je vložení elementu pro přehrávání videa:

```
<video width="320" height="240" controls>
  <source src="movie.mp4" type="video/mp4">
  <source src="movie.ogv" type="video/ogg">
Váš prohlížeč nepodporuje přehrávání videa.
</video>
```

Atribut `controls` zajistí vložení potřebných ovládacích prvků pro přehrávání videa. Tag `source` umožňuje vložení videa v různých formátech.

Velmi podobně lze přehrávat rovněž zvukové soubory:

```
<audio controls>
  <source src="horse.ogg" type="audio/ogg">
  <source src="horse.mp3" type="audio/mpeg">
Váš prohlížeč nepodporuje přehrávání zvukových souborů.
</audio>
```

Díky připojeným pomocným programovým modulům (tzv. pluginům) lze rozšířit funkčnost prohlížeče tak, aby umožnil zobrazení i jiných dokumentů - např. PDF nebo flashových animací. Lze k tomu využít elementu `<object>`:

```
<object width="400" height="300"
data="animace.swf"></object>
```

Modernější a lepší variantou pro interaktivní grafické aplikace a animace je nový element HTML5 `<canvas>`. Jde o virtuální plátno, na kterém mohou být prostřednictvím skriptů prováděny nejruznější grafické operace:

```
<canvas id="platno" width="400" height="300">
</canvas>
```

PŘÍRAZENÍ STYLŮ K WEBOVÉ STRÁNCE

Kaskádové styly, známé také pod zkratkou CSS (z anglického Cascading Style Sheets = tabulky kaskádových stylů) jsou moderním jazykem umožňujícím účinné formátování stránek v jazycích HTML, XHTML či XML. Jazyk byl navržen organizací W3C. Existuje několik specifikací CSS, tou poslední je zatím verze CSS3. Hlavním smyslem kaskádových stylů je umožnit návrhářům oddělit vzhled dokumentu od jeho struktury a obsahu. Slovo kaskádové odkazuje na typickou vlastnost CSS – jednotlivá pravidla se mohou vzájemně překrývat.

Stále platí určitá omezení využití stylů v různých prohlížečích i na odlišných platformách. Významné rozdíly v interpretaci některých vlastností CSS existují zejména u starších verzí prohlížečů Internet Explorer.

VÝHODY POUŽITÍ CSS

Oddělení struktury textu od jeho vzhledu. Oddělení obou vrstev (strukturnální a prezentační) zvyšuje přístupnost i přehlednost webu.

Rozsáhlejší formátovací možnosti. CSS nabízejí širokou škálu vlastností, díky nimž je možné provést téměř libovolnou grafickou úpravu.

Konzistentní úprava souvisejících stránek. Je mnohem jednodušší udržovat jednotný styl rozsáhlých webových prezentací.

Dynamická práce se styly. Provést změnu stylu webu, který pro formátování vzhledu využívá jen možnosti HTML, znamená najít a nahradit všechny značky a změnit atributy mnoha dalších značek. V případě používání CSS znamená změna stylu webu přepsání jediného souboru – souboru stylů.

Kratší doba načítání stránky. Veškerý design a formátování se může načítat ze souborů .css, které jsou společné pro celý web. K urychlení může přispět zkomprimování souborů css (odstranění nadbytečných prázdných znaků).

Větší kompatibilita alternativních webových prohlížečů.

Přizpůsobení možnostem prohlížeče. Je možné upravit formátování podle prohlížeče, kterým si uživatel danou stránku zobrazuje.

Použití různých stylů pro různá výstupní zařízení. Pomocí stylů lze určit, jak bude vypadat dokument na různých zobrazovacích zařízeních, jak bude vytištěn na tiskárně apod.; specifikace CSS nezapomínají dokonce ani na zrakově postižené - je možno napsat styly pro hlasový syntetizátor nebo hmatovou čtečku Braillova písma.

Vložené styly. Nejjednodušší metoda – v rámci značky pouze uvedeme parametr `style`, který bude obsahovat seznam vlastností a jejich hodnot. Styl je platný pouze pro příkaz (tag), v němž byl uveden. Vzhledem ke komplikované aktualizaci vložených stylů je používáme pouze minimálně a ve specifických případech.

```
<h1 style="color: blue; font-style: italic">To je
modrá</h1>
```

Styly na úrovni dokumentu. Uvádějí se zpravidla v hlavičce HTML dokumentu mezi tagy `<style></style>`. Takto definované styly jsou platné pro celý dokument, pokud zrovna nejsou překryty vloženým stylem.

```
<style>
body {font-family: Arial, sans-serif; font-size:
10pt; text-align: left}
h1 {color: red}
</style>
```

Externí styly. Definice stylů je možné umístit do externího textového souboru (nejčastěji s příponou CSS) a tento soubor pak do dokumentu HTML importovat. Takto je možné používat i styly z cizích zdrojů. Velkou výhodou je, že jeden soubor s definicemi stylů je možné aplikovat třeba na celé webové sídlo. Definice stylů přitom zůstává stejná jako u stylů na úrovni dokumentů.

Externí definice stylů můžeme do HTML dokumentu nahrát dvěma různými způsoby:

1. Jako vsunuté styly příkazem `<link>`:

```
<link rel="stylesheet" type="text/css" href="http://www.
seznam.cz/css/zakladni.css" title="základní">
```

2. Jako importované styly pomocí speciálního příkazu `@import` v tagu `<style>`:

```
<style>
<!--
@import url(http://www.seznam.cz/css/zakladni.css);
h1 {color: blue; font-style: italic}
-->
</style>
```


1

Syntax CSS - selektory a deklarace

Základem kaskádových stylů jsou pravidla (rules). Každé pravidlo obsahuje selektor a blok deklarací. Jednotlivé deklarace jsou odděleny středníky. Každou deklaraci tvoří dvojice vlastnost:hodnota. Díky CSS se na jeden element může navrstvit mnoho různých deklarací (proto styly kaskádové), někdy i protichůd-



ných. Platná je zpravidla poslední deklarace s výjimkou použití příznaku !important za deklarací. Taková nemůže být potlačena pozdější běžnou deklarací.

TŘÍDY A IDENTIFIKÁTORY

Třídy a identifikátory v CSS slouží k tomu, abychom mohli různé elementy formátovat různě (např. odkazy na stránce). Třídy vytvoříme tak, že k elementu v HTML přidáme atribut class. Jeho hodnotou bude řetězec písmen, který pak můžeme používat jako selektor.

```
<p class="poznamka">Nějaký text</p>
.poznamka {font-size:x-small; color: black}
```

Tento odstavec bude formátován podle pravidel třídy poznamka, na formátování ostatních odstavců se tato pravidla neprojeví.

Identifikátor se od třídy liší tím, že se jedná vždy o jednoznačný identifikátor. To znamená, že ho na každé stránce můžeme použít jen jednou (zatímco třídu opakovaně). Identifikátory se označují mřížkou (#).

```
<div id="menu"> ... </div>
#menu {width:14em; background-color:black}
#menu a {color: white}
```

PSEUDOTŘÍDY A PSEUDOELEMENTY

Pseudotřída je využívána k označení speciálního stavu elementu. Příkladem je pseudotřída `:hover`, která definuje akci najetí kurzoru myši nad odkaz:

```
a:hover {color: #FF00FF;}
```

Pseudoelement je používán pro změnu stylu specifické části elementu. Příkladem je změna prvního řádku odstavce:

```
p::first-line { font-variant: small-caps;}
```

DĚDIČNOST

Třída každého elementu bude dědit všechny vlastnosti daného elementu a navíc bude mít své vlastní, tedy např.:

```
p {text-align: center; color: blue}
```

```
.poznamka {font-size:x-small; color: black}
```

Tímto zápisem je řečeno, že všechny odstavce budou zarovnaný na střed a jejich text bude mít modrou barvu (první deklarace). Toto obecně platí pro všechny odstavce stránky. Odstavec s třídou poznamka bude mít navíc ještě menší písmo (druhá deklarace). Barva je určená v obou deklaraci, v takovém případě má vyšší prioritu deklarace zadaná později, tedy odstavce s třídou poznamka bude mít černý text. Všechny odstavce tedy budou modré, zarovnané na střed, kromě odstavce s třídou poznamka (ten bude černý, s malým písmem, zarovnaný na střed).

Kombinátory. Označujeme tak symboly, které definují vztahy mezi selektory:
vztah předek > přímý potomek, vztah příbuzný + nejbližší příbuzný,
vztah příbuzný ~ příbuzný (obecně).

SELEKTORY

Selektory slouží k výběru prvků webové stránky, na které má být aplikováno uvedené pravidlo a deklarace v něm obsažené. V nejnovější verzi CSS3 je k dispozici mnoho různých selektorů, zde je včetně nejpoužívanějších:

Selektor	Příklad	Rozsah platnosti pravidel
*	*	Platí pro všechny elementy na stránce
element	p	Platí pro všechny elementy p
.class	.lichy	Platí pro elementy s atributem class="lichy"
#id	#prvni	Platí pro element s atributem id="prvni"
selektor, selektor	h1, .lichy	Hromadná deklarace - platí pro všechny vybrané prvky seznamu
selektor selektor	p .lichy	Kontextová deklarace - platí pro všechny odstavce, které obsahují element s atributem class="lichy"
element.class	p.lichy	Platí pro všechny odstavce s atributem class="lichy"
selektor > selektor	div > p	Platí pro všechny p, jejichž rodičem je div
selektor + selektor	div + p	Platí pro p, který následuje hned po div
selektor ~ selektor	div ~ p	Platí pro všechny p, které jsou příbuzné s div
[atribut]	[target]	Platí pro všechny prvky s atributem target
[atribut=hodnota]	[type=text]	Platí pro prvky s atributem type=text
[atribut*=hodnota]	[title*=ČR]	Platí pro všechny prvky, které mají v atributu title obsaženu zkratku ČR
:not(selector)	:not(p)	Platí pro všechny prvky, které nejsou p
::selection	::selection	Platí pro uživatelem vyznačené prvky

J

Běžné vlastnosti a hodnoty v CSS

Pomocí kaskádových stylů můžeme provést prakticky libovolné formátovací úpravy webové stránky. Rozsah možných úprav závisí na použité verzi CSS a schopnostech prohlížečů formátovací příkazy akceptovat. Mezi základní vlastnosti, které byly z větší části k dispozici již v CSS1, patří:

- Vlastnosti písma - vlastnost font a podřízené vlastnosti, které se připojují za pomlčku - např. font-family, font-size, font-weight atd.)
- Vlastnosti textu - text-align, text-decoration, text-indent a další.
- Barvy textu a vlastnosti pozadí - vlastnost color a množina vlastnosti pozadí, např. background-color, background-image, background-repeat atd.
- Vlastnosti blokových elementů - margin, border, padding, width, height...
- Vlastnosti spojené se seznamy - list-style-type, list-style-position apod.
- Způsoby zobrazení prvků - display, visibility, opacity atd.
- Řízení pozice - position, float, clear, top, bottom, left, right.

HODNOTY A JEDNOTKY

Jednotky délky

- (žádné) – u bezrozměrných vlastností (např. line-height); u desetinných hodnot se používá desetinná tečka
- % – procenta, jednotka relativní vůči implicitnímu rozměru, zapisována bez mezery, např. font-size:110%, tj. o 10 % větší než výchozí

- pt – typografický bod, výchozí jednotka je 1/72 palce
- px – 0,75 pt
- em – čtverčík, je rovna základní výšce písma
- ex – výška písmene „x“, je relativní k použitému písmu

Barvy mohou být zapsány buď:

- některou z předdefinovaných konstant, např. black (černá), blue (modrá), fuchsia (anilinová červen), gray (šedivá), green (zelená)...
- #rgb, barevné složky specifikované jako hexadecimální číslovky (0 až f)
- #rrggbb, na barevnou složku vychází dvouciferná hodnota (00 až ff)
- rgb(r,g,b), hodnoty jsou v desítkové soustavě (0 až 255)
- rgb(r%,g%,b%), hodnoty jsou v rozsahu (0 až 100)

URL

Používá se konstrukce `url()`, kde se mezi závorky specifikuje adresa zdroje. Ta může být absolutní, např. `url(http://www.server.com/images/logo.jpg)`, relativní vůči serveru, např. `url(/images/logo.jpg)` nebo relativní vůči aktuálnímu adresáři, např. `url(images/logo.jpg)`. Obsahuje-li URL čárky, mezey, uvozovky nebo konec kulaté závorky, tyto znaky se dají eskapovat pomocí zpětného lomítka.

Pojmem layout označují webdesignéři rozmístění základních prvků na stránce. Layout je tedy jakési schéma, jež říká, kde bude umístěn logotyp, hlavní navigace, drobečková navigace, formulář pro fulltextové vyhledávání a další obvyklé součásti stránky.

Dobrý layout je základem použitelnosti každé webové prezentace. Přitom navrhnout layout není nic jednoduchého, zejména u větších webů. Z jedné strany vás omezují pravidla informační architektury, z druhé strany volání grafika po kreativitě vzhledu webu, třetí stranou omezující vytvářený layout je již zmíněná použitelnost – s některými prvky by se na stránce nemělo příliš hýbat, uživatelé jsou zvyklí na jejich přibližné umístění (např. logo v levém horním rohu) a layout, který tato očekávání naruší, nebude zřejmě patřit k úspěšným.

K pojmu layout také patří další dva často používané termíny – **tabulkový layout** a proti němu stojící **beztabulkový layout**. Oba se týkají kódu stránek, přesněji způsobu, jakým je layout v rámci kódu stránky realizován. Starší varianta, tabulkový layout, využívá k rozmístění prvků na stránce neviditelné tabulky, často i velké množství tabulek zanořených do sebe. Přestože je tento způsob velmi zastaralý, stále je možné na tabulkový layout narazit.

Technologicky modernější varianta, beztabulkový layout, využívá k rozmístění prvků na stránce kaskádové styly. Mezi výhody tohoto řešení, které považujeme za základ současné webové prezentace, patří rychlejší načítání stránky v prohlížeči, celkově menší objem kódu a jeho správnější sémantická struktura. Beztabulkový layout se používá i při optimalizaci pro vyhledávače, kladem bude také jednodušší a tudíž levnější správa celého webu.

Třetím způsobem, umožňujícím vytvořit layout stránky, jsou tzv. **rámy**.

Rámy (anglicky frames) jsou speciální technologií, umožňující vystavět layout stránky z několika stránek jiných. Obvykle se například vytvoří stránka s hlavní

navigací a samotný obsah se umístí na další stránky. Nakonec se vše spojí přes třetí stránku s rámy. Stejně tak lze přidat další rámy např. s hlavičkou a s patičkou.

Hlavní výhodou rámu je, že při aktualizaci navigace není třeba zasahovat do všech ostatních stránek, správa webu s rámy je tak velice rychlá. Další předností rámu je i malá velikost celého webu, každá informace je uložena jen jednou. A protože uživatel při procházení načítá vždy jen obsahové stránky, zobrazují se mu velice rychle. Velmi rychle se po nich také pohybuje, neboť navigace v rámech zůstává na svém místě, ačkoliv se zbytek stránky roluje.

Dnes se však na běžných webových prezentacích rámy dávno nepoužívají. Výše zmíněné výhody lze získat i jinými způsoby (byť náročnějšími na znalosti) a nevýhody rámu proto převažují. Patří k nim zejména problémy s použitelností webu (např. nelze odkázat na konkrétní podstránku, je obtížné udělat tiskovou verzi,...).

Rámy částečně brání i přístupnosti webu (např. problémy v textových prohlížečích), jeho viditelnosti (uživatel přichází z vyhledávače na podstránku, tedy nevidí rám s navigací). Nevýhodou rámu je i malá kreativita grafického návrhu, design webu využívajících rámy se vzájemně podobá.

Přes tyto nevýhody platné pro běžné webové prezentace však není možné rámy úplně zatracovat. Hodí se například pro některé webové aplikace. Příkladem dosud používaného elementu, který souvisí s rámy, je `iframe`:

```
<iframe src="iframe.html" name="iframe_a"></iframe>
```

A zde je příklad odkazu směřovaného na uvedený `iframe`:

```
<p><a href="http://www.w3schools.com" target="iframe_a">W3Schools.com</a></p>
```

Přístupnost webu (web accessibility) je soubor pravidel, které zajišťují jeho bezbariérovost. Znamená to, že informace na webu obsažené jsou dostupné nezávisle na zobrazovacím zařízení, jeho nastavení (např. přítomnost různých pluginů) a také na fyzickém stavu uživatele.

Přístupnost se samozřejmě netýká jen slepců s jejich slepeckými čtečkami nebo lidí, kteří nemohou ovládat webovou prezentaci myší. Stále se zvětšuje počet seniorů užívajících internet. Pro ně webová přístupnost znamená možnost zvětšit si na stránkách písmo, zvýšit kontrast písma. Přístupnost webu závisí také na použitých barvách. Podle statistik je totiž každý dvacátý muž barvoslepý; například neodliší nepodtržené odkazy, může mít problémy s navigací apod. Také vyhledávací roboty vnímají pouze textový obsah webu (bez obrázků a multimédií) a stále neumí dobře JavaScript. Webová přístupnost je tedy také základem optimalizace pro vyhledávače (SEO).

Obsah webových stránek je dostupný a čitelný

Každý netextový prvek nesoucí významové sdělení má svou textovou alternativu. Informace sdělované prostřednictvím skriptů, objektů, appletů, kaskádových stylů, obrázků a jiných doplňků na straně uživatele jsou dostupné i bez těchto doplňků. Informace sdělované barvou jsou dostupné i bez barevného rozlišení. Barvy popředí a pozadí jsou dostatečně kontrastní. Na pozadí není vzorek, který snižuje čitelnost. Předpisy určující velikost písma nepoužívají absolutní jednotky. Předpisy určující typ písma obsahují obecnou rodinu písem.

Práci s webovou stránkou řídí uživatel

Obsah WWW stránky se mění, jen když uživatel aktivuje nějaký prvek. Stránka bez přímého příkazu uživatele nemanipuluje uživatelským prostředím. Nová okna se otevírají jen v odůvodněných případech a uživatel je na to předem upozorněn. Na webové stránce nic neblíká rychleji než jednou za sekundu. Webová stránka nebrání uživateli posouvat obsahem rámu. Obsah ani kód webové stránky nepředpokládá ani nevyžaduje konkrétní způsob použití ani konkrétní výstupní či ovládací zařízení.

Informace jsou srozumitelné a přehledné

Webové stránky sdělují informace jednoduchým jazykem a srozumitelnou formou. Úvodní webová stránka jasně popisuje smysl a účel webu. Název webu či jeho provozovatele je zřetelný. Webová stránka i jednotlivé prvky textového obsahu uvádějí své hlavní sdělení na začátku. Rozsáhlé obsahové bloky jsou

rozděleny do menších, výstižně nadepsaných celků. Na webové stránce je uveden kontakt na technického správce a prohlášení jasně vymezující míru přístupnosti webu. Na tuto webovou stránku odkazuje každá stránka webu.

Ovládání webu je jasné a pochopitelné

Každá webová stránka má smysluplný název, vystihující její obsah. Navigační a obsahové informace jsou na webové stránce zřetelně odděleny. Navigace je srozumitelná a je konzistentní na všech webových stránkách. Každá webová stránka (kromě úvodní webové stránky) obsahuje odkaz na vyšší úroveň v hierarchii webu a odkaz na úvodní WWW stránku. Všechny webové stránky rozsáhlejšího webu obsahují odkaz na přehlednou mapu webu. Obsah ani kód webové stránky nepředpokládá, že uživatel již navštívil jinou stránku. Každý formulářový prvek má přiřazen výstižný nadpis. Každý rám má vhodné jméno či popis vyjadřující jeho smysl a funkčnost.

Odkazy jsou zřetelné a návodné

Označení každého odkazu výstižně popisuje jeho cíl i bez okolního kontextu. Stejně označené odkazy mají stejný cíl. Odkazy jsou odlišeny od ostatního textu, a to nikoli pouze barvou. Obrázková mapa na straně serveru je použita jen v případě, že nebylo možné pomocí dostupného geometrického tvaru definovat oblasti v obrázkové mapě. V ostatních případech je použita obrázková mapa na straně uživatele. Obrázková mapa na straně serveru je vždy doprovázena alternativními textovými odkazy. Uživatel je předem jasně upozorněn, když odkaz vede na obsah jiného typu, než je webová stránka. Takový odkaz je doplněn sdělením o typu a velikosti cílového souboru.

Kód je technicky způsobilý a strukturovaný

Kód webových stránek odpovídá nějaké zveřejněné finální specifikaci jazyka HTML či XHTML. Neobsahuje syntaktické chyby, které je správce webových stránek schopen odstranit. V metaznačkách je uvedena použitá znaková sada dokumentu. Prvky tvořící nadpisy a seznamy jsou korektně vyznačeny ve zdrojovém kódu. Prvky, které tvoří nadpisy či seznamy, naopak ve zdrojovém kódu takto vyznačeny nejsou. Pro popis vzhledu webové stránky jsou upřednostněny stylové předpisy. Je-li tabulka použita pro rozvržení obsahu webové stránky, neobsahuje záhlaví řádků ani sloupců. Všechny tabulky zobrazující tabulková data naopak záhlaví řádků a/nebo sloupců obsahují. Všechny tabulky dávají smysl čtené po řádcích zleva doprava.