

**Tvorba WWW stránek s využitím technologií (X)HTML, CSS, PHP
a databází**

HTML

Úvod do (X)HTML

„Tagy“ – značky. Slouží ke strukturování dokumentu (párové a nepárové)

```
<tag atribut1="hodnota atributu"> text </tag>
```

HTML – HyperText Markup Language

- HTML – první verze 1991
- Značkovací jazyk pro zápis dokumentů na webu
- Existuje několik verzí. Poslední je HTML 4.01.
- HTML se dále nevyvíjí.

Základní struktura stránky

```
<HTML>  
<HEAD>  
    záhlaví stránky  
</HEAD>  
<BODY>  
    vlastní zobrazený text  
</BODY>  
</HTML>
```

XML - Extensible Markup Language

- Univerzální jazyk pro předávání strukturovaných informací.
- XML definuje jen způsob strukturování dokumentu (syntaktická pravidla).
- Nedefinuje použitelné značky.
- Použitelné značky a další pravidla záleží již na konkrétní aplikaci XML.
- Výhodné pro automatizované zpracování.

XHTML - Extensible HyperText Markup Language

- Jedná se o aplikaci XML.
- Na první pohled podobné HTML (použitelné tagy z něj vychází)
- Novější a modernější.
- Existuje několik verzí (1.0, 1.1, a druhů Strict, Transitional, Frameset)
- Přísnější pravidla pro zápis.

Některé základní odlišnosti od HTML:

- Všechny atributy tagů mají hodnoty v uvozovkách
- Zákaz křížení tagů (<i> text </i>)
- Všechny tagy a atributy jsou malými písmeny (case sensitive)
- Nepárové tagy končí lomítkem např.:

- Párové tagy jsou párové povinně
- Všechny atributy musejí mít hodnotu
- Mírně odlišný zápis javascriptů a stylů
- Dokument má mít XML prolog.
- Dokument požaduje správný doctype.

Příklad základní stránky:

```
<?xml version="1.0" encoding="iso-8859-2" ?>

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

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="cs"
lang="cs">

<head>
    <meta http-equiv="Content-Type" content="text/html;
        charset=iso-8859-2" />
    <title>Titulek stránky</title>
</head>

<body>
    <p>Text ...</p>
</body>

</html>
```

Kódování češtiny:

ASCII tabulka 0-127 – popisuje základní znaky
128 – 255 – použitelné pro národní abecedy

Nejpoužívanější:

iso-8859-2	-	ISO norma (Latin 2) pro východo(středo)evropské jazyky. Standard hlavně na UNIXu (Linuxu)
windows-1250	-	Windows kódování
UTF-8	-	Univerzální kódování Unicode

Validita kódu:

Psát kódy podle norem (validní), možno ověřit na internetu
<http://validator.w3.org/>

V praxi:

- ne každý validní kód správně zobrazí i webový prohlížeč
- ne každý nevalidní kód zobrazí nesprávně
- různé prohlížeče různé chování na stejném kódu (i validním)

(X)HTML – tagy – základní přehled

Opravdu velmi stručný přehled některých tagů. Pro další informace doporučuji např.:

<http://www.jakpsatweb.cz/html/>

U mnoha tagů lze použít různé atributy: border, bordercolor, height, width,

Spoustu atributů na formátování vzhledu a některé tagy je lepší nepoužívat (zastaralé) a nahradit CSS (kaskádovými styly)

Základní tagy – viz. příklad výše:

```
<html><head><body>
```

Tagy v sekci HEAD:

```
<meta><title><link><script>
```

Př.:

```
<link rel="stylesheet" type="text/css" href="format.css" />
```

Poznámky:

```
<!-- poznámka -->
```

Formátování textu (nahrazovat CSS):

<code></code>	-	tučně
<code><i></code>	-	kurzíva
<code><u></code>	-	podtrženo
<code><sup><sub></code>	-	indexy
<code></code>	-	fonty

Formátování odstavců:

<code></code>	-	oddělení části textu s jiným formátováním
<code><div></code>	-	blok textu (zalomuje řádek před i za)
<code><h1> ... <h6></code>	-	nadpis 1. – 6. úrovně
<code><p></code>	-	odstavec (vytváří před a za mezeru)
<code>
</code>	-	odřádkování

<center> - vycentrování
<pre> - předformátovaný text
<hr> - horizontální čára

 - nečíslovaný seznam
 bod 1

 ... - číslovaný seznam

Odkazy:

```
<a href="http://www.mojestranka.cz">odkaz</a>  
<a href="#zacatek">začátek</a>  
    <h1 id="zacatek">Hlavní nadpis stránky</h1>
```

parametry: href, target

Tabulky:

```
<table summary="popis tabulky">  
  <tr>  
    <th>1</th>  
    <th>2</th>  
  </tr>  
  <tr>  
    <td>aaa</td>  
    <td>bbb</td>  
  </tr>  
</table>
```

Pozn. pro slučování buněk slouží atributy colspan a rowspan

Obrázky:

```

```

atributy: src, alt, width, height, border,

Klikací mapy:

```
<map id="mapa" name="mapa">
  <area href="prvni.html" shape="rect" coords="80, 80,
130, 130">
  <area href="druha.html" shape="rect" coords="20, 20,
50, 50">
</map>
```

```

```

Vkládání objektů:

jen některé příklady

```
<object
  type="application/x-shockwave-flash" data="flash.swf"
  width="100" height="40" >
  <param name="movie" value="flash.swf" />
</object>
```

```
<script type="text/javascript">
function podnabidka(id){
  if(document.getElementById(id).style.display=='block')
  document.getElementById(id).style.display='none';
  else document.getElementById(id).style.display='block';
}
</script>
```

Rámce:

```
<frameset cols="250, *">
  <frame name="I1" src="menu.htm">
  <frame name="I2" src="main.htm">
  <noframes> když to nejede a co se vypisuje v některých
vyhledávačích... </noframes>
</frameset>
```

Atributy frameset: cols, rows, border, ...

Atributy frame: name, src, scrolling, noresize, ...

`` - odkaz do rámu

Plovoucí rámce:

```
<iframe name="main" src="pg0.php" width="484"
frameborder="0" scrolling="no" > alternativní obsah
</iframe>
```

Formuláře:

```
<form action="index.php" method="post">
... zde jsou jednotlivé prvky formuláře
</form>
```

Atributy: action, method, accept-charset,

Základní prvky formulářů:

`<input>` - vstupní pole (vypadá různě podle atributů)

atributy: type, name, value, disabled, readonly, (size, maxlength)

Atribut type (jeho hodnota):

- text
- password
- hidden
- radio
- checkbox
- submit
- reset
- file

```
<select> - vybírací nabídka
  <option value="vse">text</option>
  <option value="vse">text</option>
</select>
```

Atributy select: name, multiple, size, disabled

Atributy option: value, selected

Pozn. `<optgroup>`

`<textarea> </textarea>` - textové vstupní pole

Atributy: name, cols, rows, disabled, readonly, ...

Další prvky formulářů:

label, optgroup, button, ...

CSS - Cascading Style Sheets

- CSS (Cascading Style Sheets) je soubor metod sloužících ke grafické úpravě HTML dokumentu.
- Nahrazuje některé HTML tagy a jejich atributy a přináší přehlednější a efektivnější způsob ke stylování dokumentu.
- Pro opravdové ovládnutí vzhledu HTML dokumentu je jejich využití prakticky nutností.
- Jakékoliv statické formátování je zvládnutelné pomocí CSS
- Novější verze (X)HTML vlastní atributy tagu pro stylování nepřipouštějí a počítají tomhle ohledu s využitím CSS.
- I CSS patří pod W3C a je možné ověřit validitu vytvářených dokumentů.

CSS lze v dokumentu využít několika způsoby:

- 1) Zápisem atributu „style“ u jednotlivých tagu – jednoduchý a rychlý způsob pokud chceme jednotlivé některé části nastavit, ale nepřehledné a u opakujících se použití téhož stylu přímo nevhodné.

Jako atribut prakticky jakéhokoliv tagu je možné zapsat style=" " a v uvozovkách uvést jednotlivé vlastnosti pro CSS.

```
<div style="border: 2px white; width: 180px;" >
```

- 2) Stylopisem v hlavičce HTML dokumentu. Styly pro celý dokument jsou uvedeny mezi tagy <head></head>. Uvnitř párového tagu „style“ uvedeme nastavení stylu pro jednotlivé elementy dokumentu.

```
<head>
```

```
<style type="text/css">
```

```
div { border: 2px white; width: 180px }
```

```
</style>
```

```
</head>
```

- 3) Stylopisem v externím souboru. Jedná se o soubor, kde jsou zapsány všechny styly, které se budou opakovaně využívat v celé webové aplikaci a tento soubor se tagem „link“ připojí k html dokumentu. Tento způsob je asi nejvýhodnější, chceme-li mít všechny styly pro aplikaci pohromadě a v jednotlivých souborech aplikace se pak na ně můžeme jen odkazovat a jinak již používat (X)HTML bez CSS.

```
<head>
```

```
<link rel="stylesheet" type="text/css" href="format.css">
```

```
</head>
```

Všechny tři možnosti lze samozřejmě i kombinovat, ale u složitějších stránek je potřeba si v tom zavést určitý pořádek.

Vlastní zápis stylů (u možností 2) a 3)):

```
div { border: 2px white; width: 180px }
```

--nadefinuje styl pro tag div → tag div teď bude vždy vypadat tak jak jej nastavíme

```
.mujstyl { text-color: red; text-decoration: underline; }
```

-- definujeme si vlastní třídu a tu pak budeme používat podle potřeby (přiřazovat ji jednotlivým tagům)

Použití vlastní třídy:

```
<tag class="mujstyl"> .....
```

```
<tag class="prvnistyl druhystyl"> .....
```

Pozn.:

Místo třídy můžeme definovat i identifikátor - #mujstyl { ... }

Přiřazení je pak přes atribut tagu: id="mujstyl"

Obecně je jedno co použijete (doporučuji třídy) rozdíly jsou při zpracovávání skriptů

Deklarace pro více elementů současně:

```
html, body { height: 100%; }
```

Kontextová deklarace:

```
td a { font-size: 10pt; }
```

Deklarace s pseudotřídou – (hlavně) tag a:

```
a {text-decoration: none; }  
a:hover {text-decoration: underline}
```

pseudotřídy: visited, hover, link, active

Přehled některých nejpoužívanějších CSS vlastností

Zápis je vždy vlastnost:hodnota;

Vlastnosti:

width	-	šířka
height	-	výška
color	-	barva, název nebo #RRGGBB
background-color	-	barva pozadí
background-image	-	obr. na pozadí url('cesta k obrazku')
background-repeat	-	opakování pozadí
font-family	-	písma – konkrétní názvy
font-style	-	styl
font-size	-	velikost
font-weight	-	tloušťka
text-decoration	-	dekorace textu
line-height	-	výška řádku
text-indent	-	odsazení 1. řádku
text-align	-	zarovnání text
vertical-align	-	vertikální zarovnání
margin	-	vnější okraj (margin-left,)
padding	-	vnitřní okraj (.....)

border - rámečky (spousta možností)

atd opravdu spousta dalších možností a vlastností

Přehled např:

<http://www.jakpsatweb.cz/css/css-vlastnosti-hodnoty-prehled.html>

př.:

```
.nadpis {  
    color: #454A3D;  
    text-align:center;  
    font-weight:bold;  
    font-style:italic;  
    font-family: Arial, Verdana, Helvetica, sans-serif;  
    font-size: 14pt;  
    text-decoration: none;  
    border-bottom: solid #454A3D 2px;  
    border-top: solid #454A3D 2px;  
}
```

Dynamické www stránky, systémy řízení báze dat a prostředky pro vývoj www aplikací

Systémy www aplikací

WWW (World Wide Web) je dnes společně s elektronickou poštou nejvyužívanější službou internetu. URL (Uniform Resource Locator) slouží ke kompletní adresaci informace na internetu. Udává jak protokol, který je k přístupu využít, tak i **adresu serveru** (doménovou nebo IP) a přesné umístění dokumentu.

Obecné schéma: protocol://[login_name[:heslo]@]adresa_serveru[:port][cesta]

Základním protokolem pro www je http (HypeText Transfer Protocol). Slouží k přenosu HTML dokumentů. HTTP protokol pracuje formou klient-server. Klient zasílá požadavky a server na ně reaguje odpovědí. HTTP existuje v několika verzích 0.9, 1.0 a 1.1. Jsou specifikovány RFC (Request for Comments), které stanoví normy pro protokoly a systémy. HTTP 1.0 – RFC 1945, HTTP 1.1 – RFC 2068.

Pro provoz HTTP a tedy zpracovávání požadavků od klienta je potřeba mít nainstalovaný **http server**. Serverů existuje celá řada. Jako nejvhodnější se mi jeví Apache [1], který je dostupný zdarma formou open-source a pro více platforem (v Linuxu bývá obvykle součástí většiny distribucí). Bylo by ale možné využít i např. MS IIS (Internet Information Server). Apache HTTP server je jedním z nejpoužívanějších serverů na internetu. Podle [1] ho využívá až 70% webových stránek na internetu. Apache http server je produktem Apache Software Foundation, která se zabývá open-source softwarovými projekty. Apache se neustále vyvíjí a podporuje nové vlastnosti.

HTTP server zpracovává požadavky klientů, kteří se k němu připojují za účelem získání HTML stránek.

Dynamické WWW stránky

Využití některé verze HTML je sice základem pro www aplikaci, ale pokud chceme, aby stránky aktivně prováděly výpočty či vyhodnocování, aby spolupracovaly s databází a vytvářely kvalitní rozhraní celé aplikace, musíme použít další nástroje.

První možností je použití skriptů a některého skriptovacího jazyka. Jsou to především CGI (Common Gateway Interface) skripty zpracovávané **na straně serveru**. CGI skripty jsou nezávislé na programovacím jazyku. Využití CGI skriptů je dnes výhodné jen za určitých podmínek (dlouhé a složité výpočty a jen krátké zobrazovací výpisy). Další možnostmi jsou JavaScript a VBScript zpracovávané **na straně klientského počítače**.

JavaScript je jazyk interpretovaný prohlížečem a není tedy zpracováván na straně serveru. Díky tomu může být využit k účelům, kdy by bylo využití například PHP krajně nevhodné, například k aktivní stylizaci dokumentů (tvorbě různých menu). Je ale možné ho využít i k jednoduchým výpočtům, interakci s dokumentem či řízení prohlížeče. Jazyk podporuje většinu obvyklých příkazů, výrazů i operátorů, netypané proměnné a využívá objektových vlastností. Do dokumentu se zapisuje tagem `<script language="javascript">`
. . . `</script>`.

Mezi modernější jazyky zpracovávané serverem patří ASP (Active Server Pages) a PHP (PHP Hypertext Preprocessor). ASP je skriptovací jazyk z dílny firmy Microsoft. Funguje na principu vkládání vsuvek a příkazů serverem a jejich interpretaci. Vlastní ASP se píše za pomoci VBScriptu nebo JScripu (implementace JavaScriptu od Microsoftu). Novější možností je využití ASP.NET. Nejedná se již o skriptovací jazyk, ale o plnohodnotný programovací jazyk využívající jádra .NET framework. ASP i ASP.NET je komerční aplikací.

Podobné možnosti jako ASP nabízí i PHP. **PHP** je na rozdíl od ASP open source [19], takže jsou dostupné i kompletní zdrojové kódy a je k dispozici na více platformách. Opět pracuje na principu vkládání částí kódu do HTML stránek. Tento kód interpretuje server a musí tedy na něm být podpora instalována. Klient, který výslednou stránku zobrazuje, nepotřebuje žádnou dodatečnou instalaci aplikace nebo nastavení. Klientu (www prohlížeči) je prakticky jedno zobrazuje-li stránku generovanou pomocí ASP, PHP nebo jen statickou HTML stránku. PHP podporuje celou řadu databázových SQL serverů, se kterými dokáže pomocí vlastní sady funkcí komunikovat. Pro administraci MySQL databáze pomocí PHP existují dokonce zvláštní nástroje jako je phpMyAdmin [24].

Stránky obsahující PHP musí být na serveru rozpoznány (například díky příponě php) a potom může být kód zpracován. V praxi se v jednom dokumentu často kombinuje PHP kód spolu s HTML. K oddělení se PHP zapisuje do zvláštních značek:

- `<? . . . ?>`
- `<?php . . . ?>`
- `<SCRIPT LANGUAGE="php"> . . . </SCRIPT>`

Obecně lze využít všech třech způsobů s tím rozdílem, že druhá varianta je vyžadována normou XML.

PHP podporuje proměnné různých typů včetně složených polí a objektů i příkazy typu if, while a podobně. Samozřejmě je celá řada využitelných funkcí pro matematické výpočty, spolupráci s databázemi, soubory, protokoly a mnohé další [19]. Jeho použití je do jisté míry podobné jazykům C/C++.

Systemy řízení báze dat

Databází rozumíme uspořádanou množinu informací (dat) uloženou na paměťovém médiu. Systém řízení báze dat (SŘBD) je databáze včetně softwarových prostředků umožňující přístup k těmto datům a manipulaci s nimi. Obvykle je pro jednoduchost pojmem databáze myšlen celý SŘBD.

Databáze slouží k přehlednému ukládání většího množství dat a snadné a efektivní další práci s nimi. Z hlediska ukládání dat a vazeb mezi nimi můžeme rozdělit databáze na následující typy:

- Síťové databáze
- Hierarchické databáze
- **Relační databáze**
- Objektové databáze
- Objektové relační databáze
- Ostatní typy

Nejrozšířenějším typem jsou databáze relační (případně objektové relační). Relační databáze je založena na relačním matematickém modelu. Ze dvou množin dat je možné vytvořit kartézský součin a relace je podmnožinou tohoto součinu odpovídající jisté skutečnosti. Data jsou uložena do tabulek (n-tic) s pevně stanovenými typy položek (sloupce) a jednotlivé položky jsou uloženy v řádcích. Mezi tabulkami majícími stejný typ sloupce mohou pak existovat vazby (relace).

Jazykem pro ovládání databáze je SQL (Structured Query Language - strukturovaný dotazovací jazyk). SQL se však prosadil jako de facto standard a ANSI (American National Standards Institute) založil standard na tomto jazyku. S jazykem se pracuje pomocí sady příkazů pro definici, manipulaci a řízení dat.

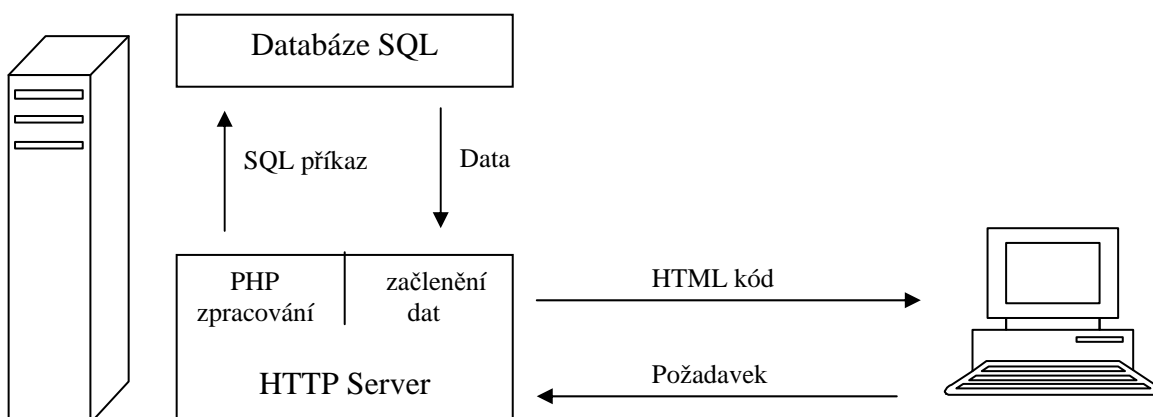
Jazyk SQL využívá celá řada databázových systémů a určit, který je nejlepší, není prakticky možné. Vždy záleží na konkrétních požadavcích, které od systému očekáváme. Mezi nejznámější databázové systémy patří: Oracle, MS SQL, **MySQL**, PostgreSQL:

Každý databázový systém má svoje klady a zápory. Výběr záleží na konkrétní aplikaci, na použitelnosti ve vybraném implementačním jazyku a platformě.

Princip

Za pomoci PHP (a ostatních jazyků) je možné vytvořit distribuovanou webovou aplikaci. Celý systém webového serveru s podporou PHP a databáze SQL pracuje následujícím způsobem. Klient přes www prohlížeč odesílá na server požadavky, server za pomoci PHP a databáze zpracuje výsledný kód a odesílá uživateli čistý HTML kód (případně včetně CSS a JavaScriptu). Požadavky klienta jsou získávány většinou odesláním HTML formulářů a jejich zpracováním pomocí PHP na straně serveru.

Proces je vidět na následujícím obrázku:



Instalace a zprovoznění www stránek

Jak tedy zprovoznit svoje vlastní www stránky? Existuje několik možností. V první řadě si stránky musím vytvořit (nebo nechat vytvořit). Pokud budu používat některé serverové skripty, jako je PHP musím mít při testování (a to je třeba často) nainstalovanou podporu. Mohu si tedy na svůj počítač nainstalovat http server (např. Apache) a PHP. Pak mohu jednoduše lokálně testovat svoje stránky. Pokud budu chtít využívat i databáze, a pro větší aplikace je to téměř nezbytnost musím nainstalovat i databázový stroj. Ke konfiguraci a vytvoření databáze bych ale měl znát, alespoň základy SQL.

Jakmile mám stránky vytvořeny, chci je vystavit na internetu. To modu udělat více způsoby:

- vlastní PC s veřejnou IP adresou, PC musí být stále zapnuté (tedy pokud se mají uživatelé připojit a prohlížet si stránky či pracovat v aplikaci).
- zakoupení webhostingu – za měsíční poplatek si pronajmu místo na serveru a s tím i služby, tam nahraji svoje stránky.

- freehosting – pokud nechci platit existují i servery, na které mohu umístit svoje stránky zdarma, zde ale většinou nemám všechny služby (např. PHP) a také adresa není reprezentativní.

Stránky mohou být samozřejmě viděny a provozovány pod IP adresou, ale lepší je mít adresu v doménovém tvaru (např. www.mojestranka.cz).

Doménu v podobném tvaru, tedy přímo v nějaké národní nebo generické doméně první úrovně, si mohu za roční poplatek zaregistrovat (např. www.nic.cz). Na freehostingu dostanu většinou doménu nižší úrovně (např. mojestranka.unas.cz).

Pokud mám dynamickou IP adresu na svém počítači a chci na ní provozovat server s doménovou adresou, která bude stálá, existují i zde služby (<https://www.dyndns.com/>).

Základy PHP

PHP skripty jsou programy běžící na serveru. Píší se podobně jako normální (X)HTML, jenom občas obsahují `<? kousek PHP kódu ?>`, který je serverem při odesílání vyhodnocován. Kdekoliv na stránce se prostě napíše znaky `<? , PHP kód ... ?>`.

Doporučuji např. <http://www.jakpsatweb.cz/php/moznosti-php.html>
www.php.net

Některé příkazy PHP:

```
echo "text text text";           // vypíše text – tedy i HTML tagy a podobně.
```

```
INCLUDE "menu.php";             // vloží na místo obsah jiného souboru (HTML, PHP)
```

```
Exit;                            // ukončí provádění skriptu
```

Proměnné:

```
$navez = .....;                // nadefinování jména proměnné a popř. přiřazení hodnoty
```

```
IsSet();                         // zjistí jestli proměnná zadaná jako parametr existuje
```

