

1 HTML

HTML - HyperText Markup Language – značkovací jazyk, ve kterém se píše webové stránky. Texty, obrázky, tabulky... jsou v tomto jazyce obstarány **značkami (tagy)**, které jsou povelami pro webový prohlížeč, jak má text, obrázek, tabulku... zobrazit.

Jazyk html procházel bouřlivým vývojem, časem vznikaly jeho **standardy** (něco jako pravidla pravopisu) s různými jmény. V současné době se používá html 4.01 Transitional, nebo přísný html 4.01 Strict.

Formát webových stránek:

Soubory určené pro zobrazení webovým prohlížečem jsou psané v html kódu a mají příponu .html.

Dvojitá tvář html souborů :)

1. Přípona .html je obvykle asociovaná s webovým prohlížečem, takže se soubory html obvykle otevírají ve **webovém prohlížeči**, tak, jak jsme zvyklí.

2. Chceme-li ale vidět zdrojový kód (html kód) souboru, otevřeme jej v obyčejném **textovém editoru**. (Na prohlédnutí html kódu webové stránky, kterou právě prohlížíme v prohlížeči použijeme klávesovou zkratku ctrl+U.)

3. Píšeme-li zdrojový kód webové stránky, můžeme použít jakýkoliv textový editor (je nutná dokonalá znalost html kódu), nebo využijeme editor či generátor html kódu (stačí znát html kód jen orientačně).

Editory html kódu

1. Komerční: Macromedia Dreamweaver (DW), Microsoft FrontPage...

(oba patří do kategorie vizuálních (**WYSIWYG**) editorů - nepíšeme html kód, pracujeme se stránkou tak, jak vypadá v prohlížeči, odpovídající html kód vygeneruje editor)

2. Volně šiřitelné: Golden HTML Editor, EasyPad...

(oba patří do kategorie **nevizuálních editorů** - píšeme html kód)

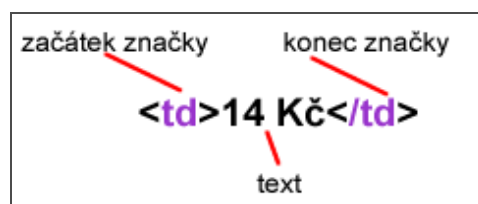
Užitečné webové stránky o HTML kódu a tvorbě webových stránek: www.jakpsatweb.cz

1.1 ZNAČKY (TAGY) A ATRIBUTY

Zdrojový kód html souboru se skládá z **textů, značek a atributů značek**.

Značka (tag):

- **párová:** <xxx> objekt, kterého se značka týká </xxx>
- **nepárová:** <xxx>



Atributy

Atribut značky: upřesnění vlastností značky, píše se: <xxx atribut="... ">, mezi uvozovky se udává hodnota atributu.



Úkol: zjistěte, co znamenají následující značky a atributy:

tag	atribut
<head></head>	src
<body></body>	alt
<html></html>	align
<div></div>	width
<p></p>	height
<h1></h1>	border
	href
<a>	target
 	cellpadding
<hr>	cellspacing
<!-->	
	
	
<u></u>	
<table></table>	
<thead></thead>	
<tr></tr>	
<th></th>	
<td></td>	
	
	
	
 	
znaková sada (příklad):	
html standard (příklad):	

1.2 ZÁKLADNÍ STRUKTURA WEBOVÉ STRÁNKY

doctype

informace o verzi použitého HTML a typu DTD (Document Type Definition)

html

značí začátek a konec html dokumentu

head

hlavička obsahuje meta informace pro webový prohlížeč, např. použitou znakovou sadu, titulek stránky ...

body

samotný obsah stránky

použitý standard

```

<!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">
  <head>
    <meta http-equiv="Content-Type" content="text/html;
    charset=iso-8859-1" /> znaková sada
    <title>Nezajímavý html soubor</title>
  </head>
  <body>
    Jsem obyčejný html soubor, nic zajímavého ve mně není...
  </body>
</html>

```

Diagrammatic labels for the code above:

- html dokument** (bracketed on the left, spanning the entire code)
- hlavička** (bracketed on the left, spanning the <head> section)
- tělo** (bracketed on the left, spanning the <body> section)

1.3 TABULKY

Ve webových stránkách se tabulky používají ve dvou případech:

1. Je-li nutné zobrazit **klasickou tabulku** (tabulka má nějaký rozměr, ohraničení, výplň)
2. K **rozvržení webové stránky** (Tabulka sama není důležitá, jde o přesné rozdělení prostoru webové stránky do oblastí)

K tabulce se vztahují značky a atributy zřejmé z kódu:

```

<table width="500" border="1" cellpadding="2" cellspacing="3">
<tr>
<th scope="col">PO</th>
<th scope="col">ÚT</th>
</tr>
<tr>
<td><p>Matematika</p></td>
<td>Hudební výchova </td>
</tr>
<tr>
<td>Chemie</td>
<td>Hudební výchova </td>
</tr>
</table>

```

Úkol 1: Podle předchozího kódu nakreslete odpovídající tabulku. Vypište význam značek: <table>, <tr>, <th>, <td> a atributů: width, border, cellpadding, cellspacing

Úkol 2: V html kódu vytvořte tabulku pro váš školní rozvrh hodin.

Slučování buněk

U tabulek je běžné **slučování buněk** (podobně jako to znáte z tabulkových procesorů). V html kódu se ke slučování buněk používají atributy (značky <td>) **colspan** a **rowspan**. Například:

- Buňka s atributem **colspan="3"** zabírá místo přes **tři sloupce** (buňky v příslušných sloupcích tím pádem musíme vynechat)
- Buňka s atributem **rowspan="2"** zabírá místo přes **dva řádky** (buňky v příslušných řádcích tím pádem musíme vynechat)

Vyzkoušejte si: vytvořte tabulku 3 x 3 s buňkami A1 až C3 s kódem:

```
<table width="30%" border="2" cellpadding="5">
<tr>
<td>A1</td>
<td>A2</td>
<td>A3</td>
</tr>
<tr>
<td>B1</td>
<td>B2</td>
<td>B3</td>
</tr>
<tr>
<td>C1</td>
<td>C2</td>
<td>C3</td>
</tr>
</table>
```

Tabulka by měla vypadat takto:

A1	A2	A3
B1	B2	B3
C1	C2	C3

A teď vytvořte podobnou tabulku podle kódu:

```
<table width="30%" border="2" cellpadding="5">
<tr>
<td>A1</td>
<td>A2</td>
<td rowspan="2">A3</td>
</tr>
<tr>
<td colspan="2">B1</td>
</tr>
<tr>
<td colspan="3">C1</td>
</tr>
</table>
```

Tabulka by měla vypadat takto:

A1	A2	A3
B1		
C1		

Úkol 4: Vytvořte v html kódu tabulku podle vzoru:

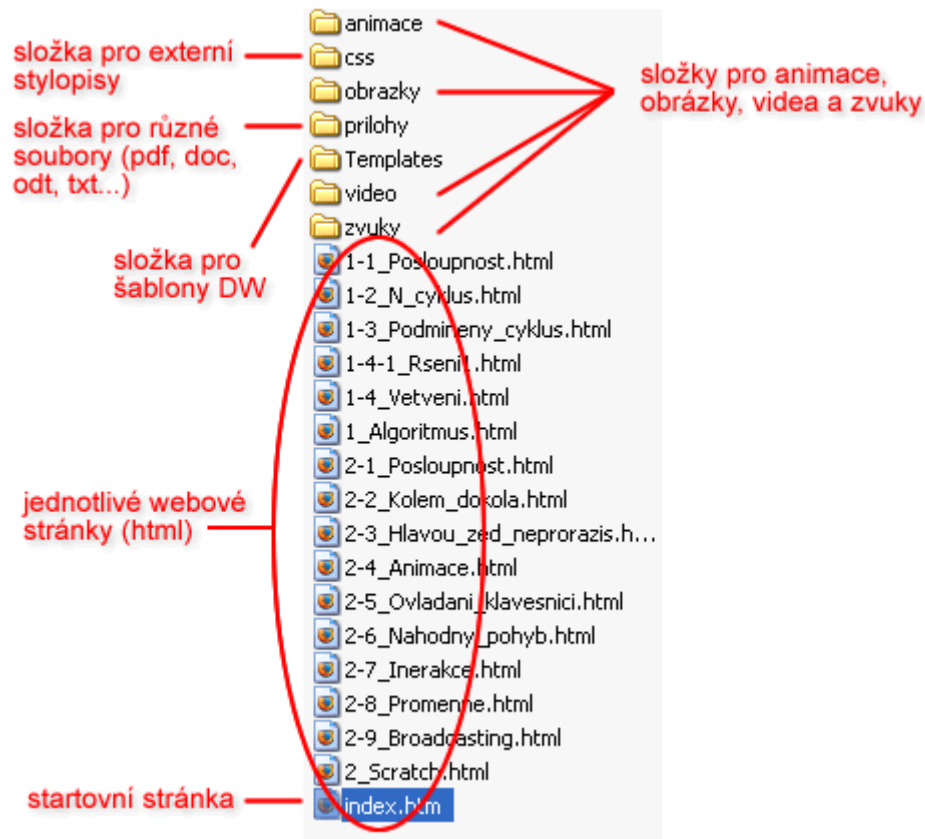
PO	ÚT	ST	ČT	PÁ
Mat	Hv	Nab	Zem	Tev
Chem		Fyz	Bio	
Inf	Anj	Mat	Fyz	Chem
Dej		Mat	Čej	Inf
oběd				

2 WEBOVÝ PROJEKT

2.1 WEB

Web je množina navzájem propojených souborů, které máme uloženy v jednom adresáři a jako **celek** je publikujeme na webovém serveru.

Příklad adresáře ICT_Algoritmy, obsahující jednotlivé webové stránky (soubory html) a složky se všemi souvisejícími soubory:



Konvence, kterou je doporučeno dodržovat:

- **startovní stránka** se jmenuje **index.html**
- v názvech souborů se **nesmí** používat **česká diakritika a mezery!**
- obrázky, stylopisy, šablony a jiné soubory jsou pro přehlednost uloženy ve složkách (podsložkách)

2.2 ODKAZY

Odkaz je v html kódu obstoupen značkami `<a>` ``.

Důležité atributy tagu `<a>` ``:

- **href** - cíl odkazu, přesněji řešeno adresa cíle odkazu.
 - Pokud je **cíl odkazu umístěn v adresáři webu**, pak může být adresa **absolutní** nebo **relativní**. Podrobněji o absolutních a relativních adresách pojednává kapitola [2.3](#).
 - Pokud **cíl odkazu leží mimo náš web**, je nutné zapsat ho pomocí [URL](#).
 - Cílem odkazu může být i **záložka** umístěná v rámci webové stránky.
- **name** - jméno záložky

Například:

Odkaz na soubor, který je součástí webu:

relativní adresa odkazu text odkazu

```
<a href="prilohy/ukazky_animaci.html">zde</a>
```

Odkaz na internetovou stránku:

URL odkazu text odkazu

```
<a href="http://www.jakpsatweb.cz">www.jakpsatweb.cz</a>
```

Odkaz na jiné místo (záložku) v rámci téže webové stránky:

cíl odkazu - záložka
jménem "jednička" text odkazu

```
<p><a href="#jednička">odskok na záložku</a></p>
```

záložka jménem
"jednička" text záložky

```
<p><a name="jednička">záložka</a></p>
```

Nejběžnější odkazy jsou **textové**, jako třeba: [seznam](#)



Oblíbené jsou i **obrázky jako odkazy**, jako třeba:

Takto vypadá kód takového odkazovacího obrázku:

párová značka odkazu

```
<a href="http://www.seznam.cz"></a>
```

cíl odkazu obrázek jako odkaz

2.3 OBRÁZKY

V html souboru je obrázek uložen **jako odkaz**. Vkládá se nepárovým tagem ****.

Důležité **atributy tagu **:

- **src** - umístění obrázku v adresáři, přesněji řečeno jeho **adresa** a to buď **absolutní**, nebo **relativní**
- **alt** - popis obrázku, který se zobrazí v prohlížeči v případech kdy:
 - uživatel má vypnuté zobrazování obrázků
 - pokud obrázek z nějakého důvodu nelze zobrazit

- **width** - šířka obrázku. Je dobré si dopředu v grafickém editoru připravit optimální obrázek včetně jeho rozměrů a tyto už v html dodatečně neupravovat.
- **height** - výška obrázku

Abslutní a relativní adresa obrázku

Atribut **src** udává umístění obrázku v adresáři. Umístění můžeme zadávat dvěma způsoby:

- **absolutní** adresou:

Absolutní adresa obrázku

```

```

pseudoprotokol úplná cesta (adresa) obrázku kopretina.jpg

Absolutní adresa se u obrázků, které jsou součástí adresáře webu nepoužívá, protože v případě, že změníme umístění celého webu, bude absolutní adresa chybná.

- **relativní** adresou:

Relativní adresa obrázku

značka umístění (zdroj) obrázku -
relativní adresa šířka výška alternativní text

```

```

obrázek je v podadresáři vzhledem k webové stránce

```

```

obrázek je ve stejném adresáři jako webová stránka

```

```

obrázek je v nadřazeném adresáři vzhledem k webové stránce

Relativní adresa se vztahuje vzhledem k webové stránce, ve které se má daný obrázek zobrazit. Relativní adresa zůstane správná i v případě přesunu celého webu.

3 STYLY A DESIGN

3.1 CSS

Každý textový dokument má svůj **obsah**, **logickou strukturu** a **vzhled**. Přitom logická struktura (nadpisy, hlavičky, patičky, poznámky, citace, odstavce běžného textu...) se vzhledem (barva písma, umístění odstavce, barva pozadí...) úzce souvisí. Vzhled textového dokumentu může zvyšovat přehlednost textu, je-li v souladu s logickou strukturou. Stejně tak vzhled může srozumitelnost textu znesnadňovat, není-li v souladu s logickou strukturou. V poslední době (aktualizováno 2009) se pro udávání vzhledu webových stránek používají **kaskádové styly - Cascading Style Sheets - CSS**.

CSS XXX je sada pravidel pro formátování webových stránek. Tato pravidla standardizuje mezinárodní konsorcium **W3C** (The World Wide Web Consortium). Standard css1: www.w3.org/TR/CSS1/

Poznámka: Nyní (aktualizováno 2009) je v platnosti standard CSS3.

Vzhled stránek v HTML

V HTML (4,01 Transitional a vyšší) jsou prostředky, kterými lze udat vzhled stránky omezené. Značky sice mají k dispozici atributy, jejichž hodnoty mohou udávat vzhled elementu mezi značkami, ale zdaleka se svými formátovacími možnostmi nevyrovnají CSS.

Např.:

```
<p align="center">Jsem odocela obyčejný odstavec </p>
```

udává, že odstavec mezi značkami má být zarovnaný na střed.

3.2 STYL

Styl znáte již z tvorby textových dokumentů. Styl je **pojmenovaná sada formátování znaku, odstavce**, či celého **bloku textového dokumentu**.

Kaskádové styly se ve webdesignu prosadily zejména pro následující výhody:

1. **Obsah** stránek je **oddělen** od jejich **vzhledu** - umožňuje rozdělit práci na webových stránkách mezi specializované pracovníky v týmu.
2. Je možné **snadno měnit vzhled** webových stránek.
3. Stránky s externím souborem pro styly (externím stylopisem) se **rychleji načítají** (css soubor se při prvním přístupu návštěvníka webu načte do Cache paměti prohlížeče a nemusí se již znovu načítat při brouzdání ostatními stránkami webu).
4. Umožňují snadné **sjednocení vzhledu** všech stránek webu.

3.3 ZÁKLADNÍ POJMY CSS

Jazyk CSS má přísná syntaktická pravidla, která nám pomůže dodržovat DW :) . Abychom si navzájem rozuměli, musíme znát základní pojmy:

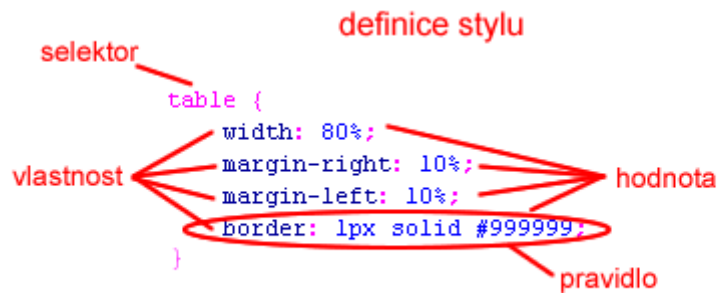
Selektor - značka, třída, nebo identifikátor, jehož styl definujeme (např. h1, p, table, .levyblok, #paticka...)

Vlastnost - jeden parametr vzhledu (např. **font-family** - druh písma, **word-spacing** - mezera mezi slovy, **color** - barva písma, **background-color** - barva pozadí, **width** - šířka elementu, **float** - obtékání elementu, **margin** - vnější okraj elementu, a desítky dalších...)

Hodnota - konkrétní informace o vlastnosti (např. **1px** - tloušťka rámečku, **#CCCCCC** - barva písma...)

Pravidlo = vlastnost + hodnota (např. **border: 1px solid #CCCCCC** - rámeček 1 px široký, plnou čarou, šedou barvou)

Definice stylu - seznam pravidel oddělených středníkem, uzavřený do složených závorek.



3.4 TROJÍ DEFINICE STYLU

Styly se do webových stránek zapisují **trojím způsobem**:

1. Individuální definice - přímo ke konkrétní značce pomocí atributu **Style**:

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>Untitled Document</title>
</head>
<body>
<p style="font-family:'Courier New'; color:#FF0000; background-
color:#66CCFF" >
červený Courier s modrým pozadím </p>
</body>
</html>

```

To se příliš nepoužívá, snad jenom chceme-li pro nějaký element vyjímku ze stylu.

2. Stylopis v hlavičce souboru (styly platné pouze pro jednu webovou stránku), analogické se styly v textových dokumentech .odt nebo .doc

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>Untitled Document</title>
<style type="text/css">
<!--
p {color: #FF0000;
background-color:#66CCFF;
font-family:"Courier New"}
-->
</style>
</head>
<body>
<p>červený Courier s modrým pozadím </p>
</body>
</html>

```

3. **Externí stylopis** - definice stylů v samostatném souboru s příponou css. Používá se nejčastěji, protože udává vzhled celého webu.

Do **samostatného souboru xxxxx.css** zapíšeme definici stylu pro odstavec:

```
p {color: #FF0000;
background-color: #66CCFF;
font-family: "Courier New
```

a do **hlavičky html souboru** zapíšeme odkaz na externí stylopis **xxxx.css**:

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01
Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html;
charset=utf-8">
<title>Untitled Document</title>
<link href="css/stylopisy.css" rel="stylesheet"
type="text/css">
</head>
<body>
<p>červený Courier s modrým pozadím </p>
</body>
</html>

```

Použijete-li (výjimečně) pro jeden element všechny tři definice, má přednost individuální definice před stylopisem v hlavičce a ten má přednost před externím stylopisem.

3.5 VRSTVENÍ, SESKUPOVÁNÍ, DĚDIČNOST

Vrstvení (kaskáda)

Konečný vzhled html dokumentu, tak jak jej zobrazí prohlížeč, je podřízen mnoha vlivům:

1. Vzhled předepíše **autor stránky** pomocí stylopisu. Stylopis může i importovat z jiných webových stránek.
2. Vzhled může upravit sám **čtenář webových stránek** ve svém prohlížeči
3. Každý **prohlížeč** má definovanou svoji sadu výchozích stylů.

Přednost (nejvyšší váhu) mají definice stylů čtenáře před styly autora (lze změnit v prohlížeči) a ty před výchozími styly prohlížeče.

Při **konfliktu definic stylů stejné váhy** se posuzuje jejich **specifičnost** (algoritmus přesného výpočtu specifičnosti najdete na stránkách <http://www.w3.org/TR/CSS1/#the-cascade>):

1. Pravidla definovaná u html značek (p, h1, h2..., table...) nebo pravidla html značek zděděná.
2. Pravidla definovaná třídou.
3. Pravidla definovaná identifikátorem nebo individuální definicí.

Přednost (vyšší specifičnost mají) pravidla v bodě 3., před pravidly v bodě 2. a ta před pravidly v bodě 1.

Při **konfliktu definic stylů stejné váhy i specifičnosti** se posuzuje jejich **aktuálnost**:

Přednost má ta definice stylu, která byla definována **později** (míněno později v toku textu). Přitom stylopis importovaný se považuje za dřívější než externí stylopis odkazovaný značkou <LINK rel="stylesheet"...> v hlavičce souboru. Tomuto pravidlu odpovídají přednosti definic stylů popsané v kapitole Trojí definice stylu.

Pravidlu aktuálnosti také podléhá rozhodování mezi použitím pravidla definovaného identifikátorem a zároveň individuální definicí stylu (v bodě 3.) **Například**:

```
<STYLE TYPE="text/css">
  #zelny { color: green }
</STYLE>
...
<P ID=zeleny STYLE="color: red"> text </p>
```

Výsledný text bude mít červenou barvu, přestože identifikátor "#zeleny" předepisuje zelenou barvu. Obě definice mají stejnou váhu i stejnou specifičnost, ale individuální definice je v toku textu později než definice pomocí identifikátoru.

Chceme-li (ať jako autor či jako čtenář) zvýšit váhu některého pravidla, použijeme příkaz ! important. Například:

```
H1 { color: black ! important;
    background: white ! important }
P { font-size: 12pt ! important;
   font-style: italic }
```

Seskupování selektorů

Podle pravidel CSS lze současně definovat styl pro více selektorů. Selektory jsou odděleny čárkami:

```
h1, h2, h3, h4, h5, h6 {color: #FF6600}
```

Všechny nadpisy mají oranžovou barvu.

Dědičnost

Definujete-li styl například pro selektor **Body** (bezpatekové písmo), styl převezmou (zdědí) některé **vnořené** elementy - nadpisy, odkazy, odstavce atd. Všechny budou vysázeny bezpatkovým písmem (neurčíte-li jinak). Nadřazeným elementům se říká **rodičovské**.

Pravidla CSS určují, která vlastnost je pro ten který element dědičná a která není.

Například v této struktuře webové stránky:

```
<body>
  <div>
    <img>
    <p>Text</p>
  </div>
</body>
```

Element **div** je **rodičovský** element pro **p**, má-li **div** definované pravidlo pro velikost písma, element **p** je **zdědí**.

Element **div** je sice **rodičovský** element k **img**, ale má-li **div** definované pravidlo pro velikost písma, element **img** je **nezdědí**.

3.6 TŘÍDY A IDENTIFIKÁTORY

Třídy (class)

Kromě toho, že v CSS můžeme definovat styl pro každou značku, můžeme definovat styl nezávislý na značkách, **vlastní styl**, kterému se říká třída. Třída má své **jméno**, za kterým následuje **definice stylu**. Jméno třídy je ve stylopisu uvozeno **tečkou!**

Například definice třídy jménem **.ukol**:

```
.ukol {
  font-style: italic;
  border: medium solid #0066FF;
  padding: 5px;
```

```
margin-top: 10px;
margin-bottom: 10px;
font-family: Georgia, "Times New Roman", Times, serif;
text-indent: 0px;
}
```

Vlastní styl třídy **.ukol** pak můžeme použít u libovolné značky atributem **class**, mnohokrát v jedné webové stránce. Například odstavec třídy **ukol**:

```
<p class="ukol">Zkomprimujte do formátu zip, vypalte na CD-RW vaši složku
Dokumenty a Data aplikací.
Poté tyto složky obnovte. </p>
```

Kříží-li se pravidla stylu pro selektor a třídu (<p> má odsazený první řádek, **.ukol** nemá odsazený první řádek), platí styl třídy.

Identifikátory (id)

Pro **jednoznačný popis** nějakého elementu (obvykle pro skripty nebo pozicování) existuje atribut **id**. I jemu se může ve stylopisu přiřadit nějaká definice, ale na rozdíl od třídy **nezačíná tečkou**, ale dvojkřížkem **#**. V těle dokumentu by se element s jedním identifikátorem měl vyskytovat **jenom jednou**.

Například definice identifikátoru **#hlavicka**:

```
#hlavicka {
background-color: #D3CEC7;
padding: 10px;
margin-bottom: 10px;
}
```

Bloku textu pak přiřadíme styl identifikátoru **#hlavicka**:

```
<div id="hlavicka">
<p> <a href="index.htm">Úvodní strana</a></p>
<p>Učebnice tvorby webových stránek</p>
</div>
```

3.7 VZHLED PÍSMO A TEXTU, UŽITEČNÁ ZNAČKA SPAN

Tato kapitola přináší výčet **nejpoužívanějších CSS vlastností** pro formátování **písma a textu**.

Písmo

- **font-family**: Druh písma, font
- **font-style**: řez písma (ne tučné)
- **font-size**: velikost písma v [délkových jednotkách](#)
- **font-weight**: tloušťka písma
- **color**: barva písma

(Tyto vlastnosti DW ukrývá v kategorii **Type**.)

Text

- **text-decoration**: podtržení, nadtržení, přeškrtnutí, blikání
- **letter-spacing**: proklad znaků
- **line-height**: výška řádku
- **text-indent**: odsazení prvního řádku v [délkových jednotkách](#)
- **text-align**: horizontální zarovnání
- **vertical-align**: vertikální zarovnání

(Tyto vlastnosti DW ukrývá v kategorii **Block**, kromě **line-height**, to je v **Type**)

Span

Značka **Span** nemá v html definovaný žádný vzhled ani význam. Je to **neutrální značka**, která pouze **označuje část textu a nezalamuje řádek**. Její důležitost vzrostla **v kombinaci s CSS**. Chceme-li totiž nějak odlišit část textu, věty nebo jen jedno slovo či písmeno, obstavíme tento úsek značkou **Span** a přidělíme jí styl (atributem **Style**) nebo třídu stylu (atributem **class**).

Například:

```
<span class="pc_menu">Soubor/ uložit jako</span>
```

3.8 DÉLKOVÉ JEDNOTKY V CSS

Používané délkové jednotky jsou **px, mm, cm, in, pt, pc, em a ex**.

- desetinné číslo se v css zadává pomocí tečky, nikoliv desetinné čárky
- V CSS se mezi číslem a jednotkou **nedělá mezera**.
Např.:font-size: 14pt;

px

px - pixel, jeden obrazový bod

- V HTML (ne v CSS) se všechny rozměry (kromě procent) udávají v pixelech, netřeba psát jednotku.
Např.:<table width="100%" border="1" cellspacing="0" cellpadding="2">
- V CSS se pixely vždy musejí zapsat s jednotkou px (bez mezery před).
Např.:margin-top: 10px;

mm, cm, in, pt, pc

mm, cm - fyzikální délkové jednotky o jejichž velikosti by měl mít každý představu. Jak je zobrazí prohlížeče je ovšem otázka, proto je radši nepoužívejme.

in - inch, palec; **1 in = 2,54 cm**.

pt - typografická délkové jednotka, odvozuje se od anglického palce. **72 pt = 1 in**. Textové editory MS Word a OpenOffice Writer používají **pt** pro velikost písma.

pc - (pica) typografická délková jednotka, **1 pc = 12 pt**.

em, ex

Em a ex jsou **relativní délkové jednotky**. Vztahují se k aktuální velikosti písma. Jednotka **em** se v designu webových stránek používá často.

em - šířka velkého M

ex - výška malého x.

Procenta

Procenta se často používají u rozměrů tabulek, velikosti písma nadpisů, u rozměrů bloků. Jako základ pro výpočet procent prohlížeče berou:

- základní rozměry písma a výšky řádku (běžné písmo = 100 %, nadpis první úrovně = 150 %...)
- rozměry rodičovského elementu (buňka tabulky = 10 % rozměrů celé tabulky)
- rozměry okna

3.9 POZADÍ A BARVY

Tato kapitola přináší výčet **nejpoužívanějších CSS vlastností** pro práci s pozadím, barvami.

Pozadí a barvy

- **background-color**: barva pozadí
- **background-image**: obrázek na pozadí
- **background-repeat**: opakování obrázku na pozadí

- **background-position**: poloha obrázku na pozadí

(Tyto vlastnosti DW ukrývá v kategorii **Background**.)

3.10 VZHLED ODKAZŮ

Vzhled odkazů v CSS definujeme pomocí **pseudotříd**:

- **a:link**: nenavštívený odkaz
- **a:visited**: navštívený odkaz
- **a:hover**: odkaz přejížděný myší
- **a:active**: aktivní odkaz

3.11 VZHLED BLOKŮ

Blokem se rozumí část html kódu ohraničená značkami, které zalamují řádky (p, br, div, h, pre, hr...).

Velikost a obtékání bloku

- **width**: šířka bloku v [délkových jednotkách](#)
- **height**: výška bloku v [délkových jednotkách](#)
- **float**: obtékání bloku
- **clear**: čekání na konec obtékání

(Tyto vlastnosti DW ukrývá v kategorii **Box**.)

Okraje a rámečky bloku

- **margin**: vnější okraj bloku v [délkových jednotkách](#)
- **padding**: vnitřní okraj bloku v [délkových jednotkách](#)
- **border-style**: styl čáry rámečku kolem bloku
- **border-color**: barva čáry rámečku kolem bloku
- **border-width**: tloušťka čáry rámečku kolem bloku v [délkových jednotkách](#)

(Tyto vlastnosti DW ukrývá v kategorii **Box a Border**.)

3.12 UŽITEČNÁ ZNAČKA DIV

Oddílem se rozumí část html kódu ohraničená značkami **Div**. Oddíl může být složen třeba:

- z jednoho nebo více odstavců textu
- z odstavců textu a obrázků
- z odstavců textu, obrázků, tabulek...

Značka **Div** nemá v html definovaný žádný vzhled ani význam. Je to **neutrální značka**, která pouze **označuje část html kódu a zalamuje řádek**. Její význam je v kombinaci s CSS obrovský.

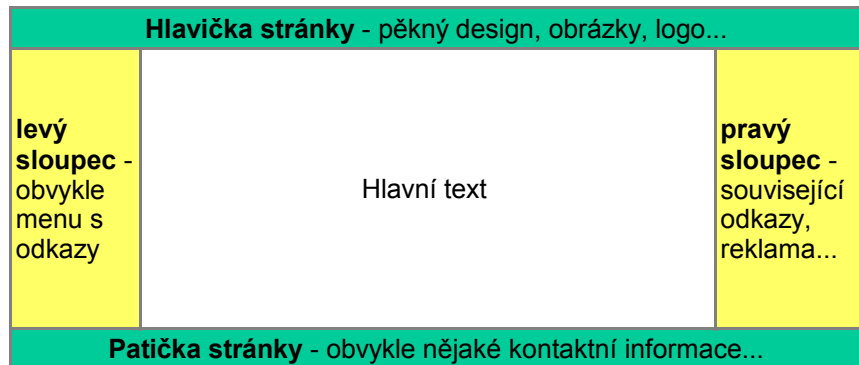
3.13 DESIGN WEBOVÝCH STRÁNEK

Webdesigner si samozřejmě může zvolit vzhled a rozvržení webových stránek podle libosti. Je třeba brát v úvahu několik skutečností:

- **jak rychle** se stránka načítá a zobrazuje v prohlížeči
 - načítání zrychlí - optimalizovaná velikost obrázků, používání CSS
- jak dobře se v ní návštěvník **orientuje**
 - je dobré používat zaběhnutá schemata rozvržení obsahu
 - orientaci zvýší krátké, stručné texty, barevně rozlišené nadpisy a bloky
- jak je **čitelná** a na pohled příjemná
 - čitelnost zvyšuje kontrast písma na pozadí
 - které kombinace barev jsou příjemné a které ne, o tom je celá věda...

- jak je **zobrazí** různé webové prohlížeče
 - v interpretaci CSS se nejpoužívanější prohlížeče (Mozilla Firefox a Internet Explorer) dramaticky liší, design je třeba odlaďovat pro oba (to vám tedy přeji pevné nervy :)).

Obvyklé schema rozvržení webových stránek:



Schema symbolicky znázorňuje rozvržení obsahu webové stránky. Obvyklé je i vypuštění některého oddílu - pravého sloupce nebo patičky.

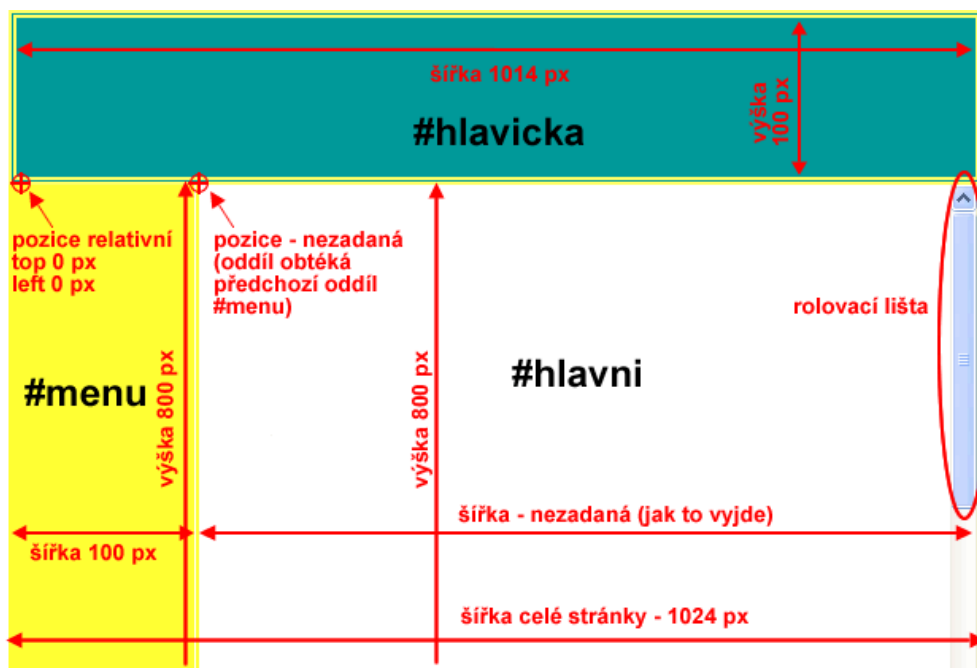
Takového rozvržení lze docílit třemi způsoby:

1. **Rozvržením do oddílů** se zadáním obtékání a rozměrů v CSS
2. **Tabulkou**

3.14 ROZVRŽENÍ STRÁNKY POMOCÍ CSS

Webová stránka s rozvržením - **hlavička**, **levý sloupec**, **hlavní text** může být realizována pomocí **třech Div oddílů** s přesným nastavením pozice, okrajů, rozměrů a obtékání. K definování stylů jednotlivých oddílů se výborně hodí identifikátory id.

Například stránka může mít tři hlavní oddíly, styl každého z nich může být udán identifikátorem:



- oddíl s id **#hlavicka**

- oddíl s id **# menu**
- oddíl s id **#hlavni**

Celá stránka - Body

Celá stránka zabírá **1024 px**, je v otevřeném okně prohlížeče **centrovaná**. To zajistíme pomocí vnějších okrajů - levého a pravého - zadáním hodnoty auto budou oba okraje stejně široké.

```
body {
font-family: Geneva, Arial, Helvetica, sans-serif;
font-size: 14px;
width: 1024px;
margin-top: 0px;
margin-right: auto;
margin-left: auto;
}
```

Oddíl #hlavicka

Hlavička stránky: **šířka** 1024 px mínus její okraje (v procentech asi tak **95%**), **výška 100 px**, bez obtékání, pravý a levý vnitřní okraj 20 px, barva tmavě tyrkysová, rámeček pastelově žlutý, dvojité.

```
#hlavicka {
background-color: #009999;
height: 100px;
width: 95%;
border: thick double #FFFF66;
float: none;
padding-top: 0px;
padding-right: 20px;
padding-bottom: 0px;
padding-left: 20px;
}
```

Oddíl levý sloupec - #menu

Levý sloupec - menu: **šířka** je jasná - **100 px**, otázka je, jak má být vysoký, protože v něm bude mnohem méně řádků, než v hlavním textu, ale měly by být oba oddíly stejně vysoké. Jedno z řešení je, že zadáme oběma oddílům stejnou výšku, řekněme **800 px**, a bude-li hlavní text delší, nastavíme mu rolovací lištu. Jelikož další oddíl - hlavní text bude vpravo od oddílu menu, musíme **menu** nastavit **levé obtékání**. Okraje a barvu včítete níže z definice id #menu.

```
#menu {
padding: 5px;
height: 800px;
width: 100px;
float:left;
background-color: #FFFF33;
border-left-width: thick;
border-left-style: double;
border-left-color: #FFFF33;
}
```

Oddíl hlavní text - #hlavni

Hlavní text: **Výška 800 px**, **šířka nezadaná** - jak to vyjde. Oddíl obtéká oddíl #menu. Pokud je text oddílu delší než 800 px na výšku je nutná **rolovací lišta**. Tu zajistíme pravidlem **overflow: scroll** nebo **overflow:auto**. Okraje a barvu oddílu včítete z kódu níže:

```
#hlavni {
height: 800px;
padding: 10px;
border-top-width: thick;
border-right-width: thick;
```



```
border-bottom-width: thick;
border-left-width: thick;
border-left-style: double;
border-left-color: #FFFF33;
border-right-style: double;
border-right-color: #FFFF33;
overflow: scroll;
}
```

Tip: mnohé firmy tvořící design pro webové stránky v rámci reklamy poskytují některé webové šablony zadarmo. My je s výhodou můžeme využít:

- www.nuviotemplates.com
- www.webove-sablony.cz/webove-sablony/zdarma
- www.breezy.cz/reference/webove-sablony-zdarma
- www.freelayouts.com/websites/html-templates

3.15 STEJNÉ ROZVRŽENÍ VÍCE WEBOVÝCH STRÁNEK

Součástí webového projektu bývají desítky stránek, obvykle s podobným vzhledem, rozvržením a v hlavičce či menu dokonce stejnými texty a obrázky. Vytváření podobných stránek nám usnadní **šablony DW**.

Pozorujte rozvržení webových stránek od profesionálů. Jakou podobu mají stránky, které se skrývají za všemi interními odkazy?

- www.giant.cz
- auto.honda.cz

3.16 SKRIPTY NA WEBOVÝCH STRÁNKÁCH

Kromě HTML kódu a CSS definic mohou webové stránky obsahovat i kódy programovacích jazyků.

Statické webové stránky

Webové stránky, které **neobsahují** kód programovacího jazyka, jehož příkazy **vykonává webový server** se nazývají **statické**.

Nenechte se mýlit tímto označením i statické stránky mohou být oživeny animacemi a interaktivními prvky (flash animace, tlačítka, rozbalovací menu...), ale příkazy jejich kódu vykonává prohlížeč, tedy webový klient.

Programovací jazyk, který se obvykle používá pro oživení statických webových stránek je **Javascript**. Javascript se používá typicky pro:

- **Rozbalovací menu.** Prohlédněte si na www.audi.cz.
- **Aktuální datum a čas:** Dnes je 31. 12. 2009. Čas: 10:51.
- **Nabídku** data do formulářových políček: http://www.dobre-svetlo.com/tools_sun.php
- atd.

Statické webové stránky jsou již dnes na ústupu, využívají se hlavně pro firemní prezentace.

Více o javascriptu na <http://www.jakpsatweb.cz/javascript/>

Hotové javascripty zdarma ke stažení a použití najdete ve vyhledávači například pod heslem "free javascript".

Dynamické webové stránky

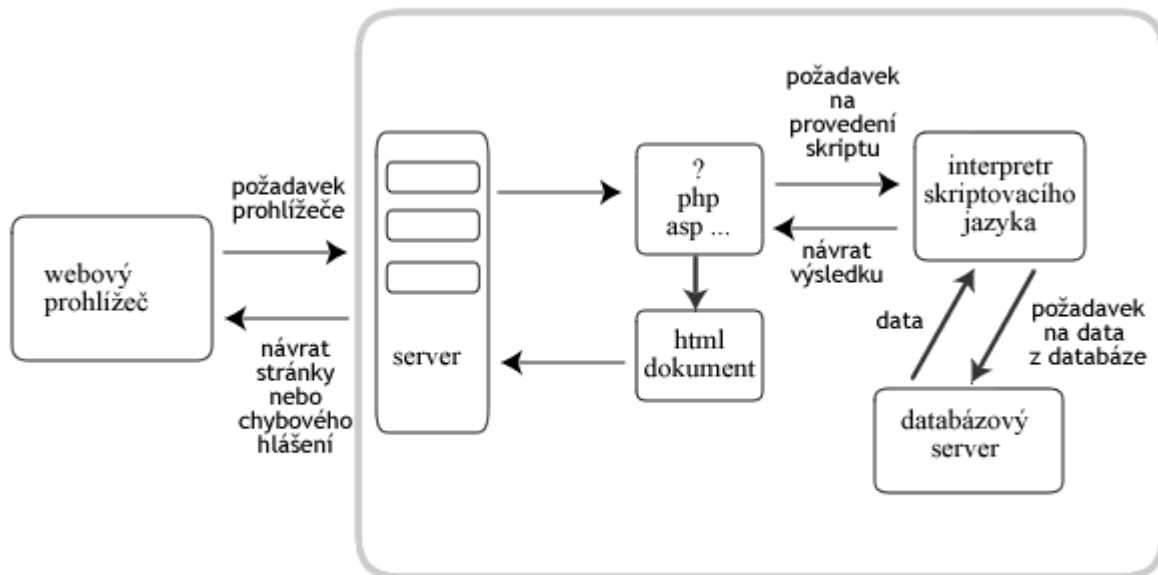
Webové stránky, které **obsahují** kód programovacího jazyka, jehož **příkazy vykonává webový server** se nazývají **dynamické**. Dynamické stránky mohou spolupracovat s databází a na požadavky uživatelů mohou odesílat i přijímat data.

Programovací jazyky dynamických webových stránek: PHP, JSP a ASP..., databáze: SQL, MySQL...

Dynamické webové stránky se používají pro:

- internetové obchody a aukce
- internetové bankovníctví

- webová alba, alba videoklipů
- sociální sítě (Spolužáci, Facebook, Hi5...)



Dynamické webové stránky

4 PUBLIKACE WEBOVÝCH STRÁNEK

Webové stránky nevytváříme pro sebe, ale pro zveřejnění "na Internetu". K tomu máme dvě možnosti:

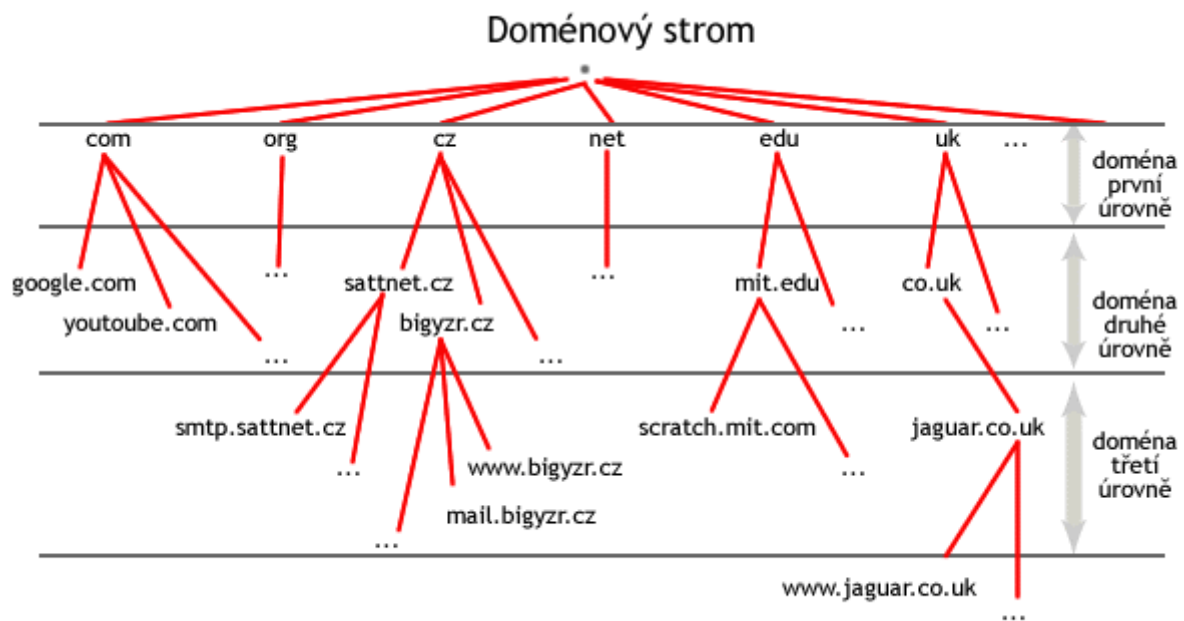
- Webové stránky necháme **na našem počítači** a zpřístupníme ho pro veřejnost tato možnost je teoreticky snadná, vyžaduje pouze:
 - **kvalitní internetové připojení**
 - **veřejnou IP adresu** našeho počítače (poskytne náš provozovatel internetu)
 - **webový server** (příslušný SW - Apache, IIS, ...)
 veřejná IP adresa však s sebou nese velké bezpečnostní riziko, proto pro běžného uživatele je vhodnější využít služeb profesionálů v oboru poskytování webových služeb:
- **Webhosting**
webhostingové firmy poskytují datový prostor na svých serverech, musíme tedy svůj web přesunout na jejich počítač. Kromě datového prostoru garantují i funkčnost serveru, zálohování, ochranu proti napadení.
 - placený - s veškerou podporou, s možností zakoupení [domény 2. řádu](#)
 - zdarma - s drobným omezením služeb nebo datového prostoru s [doménou nejvýše třetího řádu](#)

Náš web přenášíme na webový server webhostingové firmy přenášíme obvykle pomocí [FTP klienta](#).

4.1 DOMÉNOVÉ JMÉNO A DNS

Jednotlivé počítače se v internetové síti identifikují svojí **IP adresou**. IP adresa může být až 12-ti ciferná (v systému IPv6 ještě víc). Takové číslo se však špatně pamatuje, proto se IP adresám může přiřadit **doménové jméno**. Doménová jména tvoří systém **DNS (Domain Name System)**.

DNS je hierarchický systém doménových jmen, který je technicky realizován DNS servery komunikujícími mezi sebou protokolem DNS. Jeho hlavním úkolem je **převod doménových jmen na IP adresy uzlů sítě nebo naopak**. Tento systém doménových jmen tvoří stromovou strukturu:



Celé doménové jméno se skládá z **několika částí navzájem oddělených tečkami**.

TLD (Top Level Domain) - doména prvního řádu je skupinka znaků za poslední tečkou v doménovém jméně. Např: *.com*, *.edu*, *.net* (skupina generických domén, *.com* znamená komerční sektor, *.edu* vzdělávací, *.org* pro neziskové organizace...) nebo *.cz*, *.sk*, *.us*, *.de*, *.uk*... (skupina národních domén spjatá s geografickou polohou).

Každá doména prvního řádu musí mít svého **správce** (osoba, či organizace), který zajišťuje:

- přidělování doménových jmen nižšího řádu v rámci své větve
- zápis všech doménových jmen v jeho působnosti do registru domén

Stejně tak musí mít svého správce i každá doména druhého řádu.

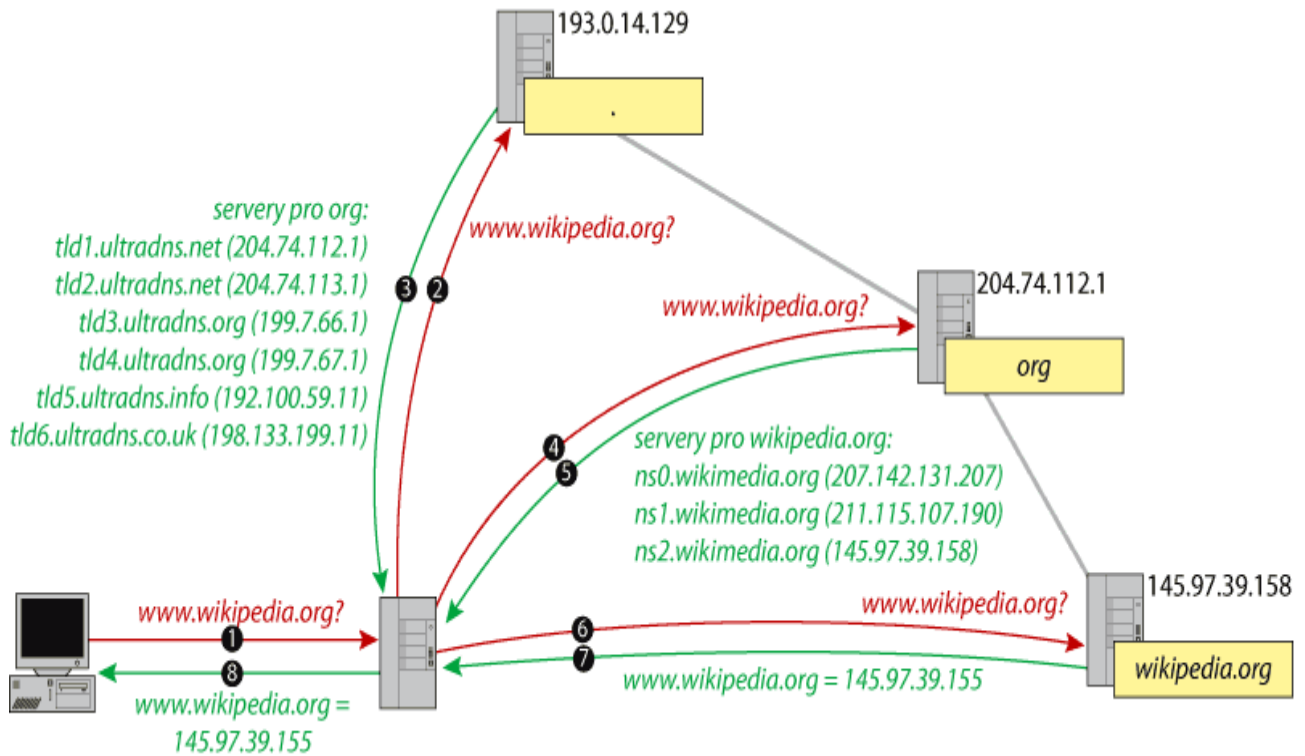
Všechny správce domén prvního řádu registruje mezinárodní organizace Internet Assigned Numbers Authority ([IANA](#)). Doménu *.cz* spravuje [CZ.NIC, z. s. p. o.](#), zájmové sdružení právnických osob.

Úkol 1: Zjistěte na webu CZ.NIC u některého z registrátorů domén druhého řádu (v rámci .cz) ceny za registraci domény.

Služba DNS

Služba DNS poskytuje překlad doménových jmen na IP adresy uzlů sítě a naopak.

Každý počítač v síti má nastaveno, jaký DNS server má pro překlad jmen používat. Služeb DNS serveru využíváme pokaždé, zadáme-li v prohlížeči (či správci souborů) adresu některého uzlu sítě pomocí [URL](#) s doménovým jménem. Prohlížeč vnese požadavek na překlad doménového jména na IP adresu. DNS server buď doménové jméno zná a pošle příslušnou adresu, nebo nezná a začne po ní pátrat u nadřazených DNS serverů (začne u **kořenového**). Nakonec předá prohlížeči zjištěnou IP adresu nebo v případě neúspěchu chybové hlášení. Fungování je naznačeno na schematicém obrázku (obrázek byl stažen z [cs.wikipedia.org](#)):

**Úkol 2:**

- 1) Zjistěte IP adresu vašeho DNS serveru ve škole.
- 2) Přeložte doménová jména: `mail.bigyzzr.cz`, `www.bigyzzr.cz` a `zed.bigyzzr.cz` na IP adresy
- 3) Zjistěte doménové jméno počítače s IP adresou 8.8.8.8

(nápopvěda - v příkazové řádce (windítko + R, cmd) příkazy `ipconfig /all`, `nslookup`)

4.2 URL

Doménové jméno či IP adresa identifikuje počítač v síti, nikoliv konkrétní objekt (soubor, aplikaci či službu) na něm uložený. Proto se k přesné identifikaci objektu na počítači v síti používá **URL - Uniform Resource Locator** (Jednotný ukazatel na zdroje). Struktura URL je vysvětlena na obrázku:

URL

http://uzivatel:jehotajneheslo@212.13.13.13:80/vyuka/testik.gif

1 2 3 4 5 6

- 1 - protokol (povinné)
- 2 - jméno uživatele (nepovinné)
- 3 - heslo uživatele (nepovinné)
- 4 - adresa serveru (povinné)
- 5 - port (v případě http protokolu je nastaven výchozí port 80, proto se nemusí uvádět)
- 6 - adresa souboru v adresáři webu (povinné)

Adresa serveru nemusí být pomocí IP, ale i doménová. S vynecháním nepovinných údajů by URL souborů mohlo být například:

http://212.233.13.13/vyuka/testik.gif

http://www.bigyzt.cz/prispevek.aspx

ftp://192.168.1.241/home/neco.txt

Poznámka: například pro webovou adresu www.seznam.cz je celé URL:

protokol: http

adresa serveru: www.seznam.cz

port: 80

adresa souboru v adresáři: index.html

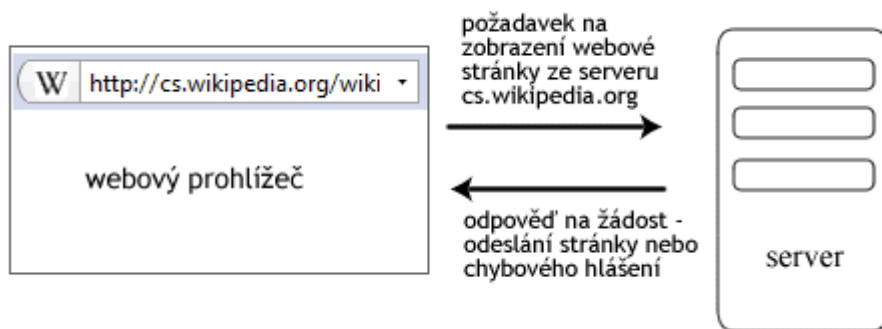
tedy: http://www.seznam.cz:80/index.html (ověřte v prohlížeči)

4.3 WWW SLUŽBA, HTTP A FTP PROTOKOL

WWW

Jedna z mnohých služeb dostupných na Internetu je služba WWW (World Wide Web) poskytující zobrazování webových stránek.

- **WWW klient** (sw na straně uživatele): Kterýkoliv internetový prohlížeč (Internet Explorer, Mozilla Firefox, Opera...)
- **WWW server** (sw na straně serveru): Apache, IIS, ...
- **protokol** pro komunikaci mezi klientem a serverem: **HTTP** (Hypertext Transfer Protocol)
- **popis služby:** uživatel zadá požadavky na zobrazení příslušné webové stránky prostřednictvím URL ve WWW klientovi, ten naváže spojení s webovým serverem, webový server požadavek vyhodnotí a klientovi odešle příslušná data, či chybové hlášení. Klient zobrazí data uživateli. Nutno podotknout, že na jedno navázané spojení odešle server jen jeden soubor. Ale webová stránka dnes obsahuje desítky, či stovky objektů - obrázků, skriptů, videoklipů atd. Pro stažení každého z nich musí prohlížeč znovu navázat spojení se serverem a obdržet příslušný objekt.



FTP

FTP (File Transfer Protocol) je služba umožňující přenos souborů mezi FTP klientem a FTP serverem, také správu souborů v přístupných adresářích serveru. Je obvyklá pro nahrávání souborů (např. webových stránek) na server (**upload**) a stahování souborů (**download**) ze serveru. Pro download není nezbytná, je možné stahovat soubory i pomocí služby WWW.

- **ftp klient:** Internet Explorer, Mozilla Firefox, Total Commander, WinSCP (podporuje i bezpečný FTPS (s znamená secure) protokol), Dreamweaver, FrontPage, v Linuxu Conqueror, Quanta+,...
- **ftp server:** IIS, NASLite, ...
- **protokol:** FTP
- **popis služby:** FTP klient naváže spojení s FTP serverem, předá přístupové jméno a heslo, od uživatele přijímá příkazy (pohyb v adresáři serveru, mazání, přejmenování souborů na serveru, stahování nebo nahrávání souborů do adresáře na serveru) a server je vykonává.

Úkol: Zjistěte několik webhostingových firem, které poskytují datový prostor zdarma. Vyberte si jednu z nich a publikujte své webové stránky s ročníkovou prací.

4.4 REDAKČNÍ SYSTÉM

Český termín "**Redakční systém**" je volným překladem anglického **Content Management System (CMS)** - systém pro správu obsahu (webu).

Tyto systémy se používají pro **snadnou aktualizaci a správu webů**, kterou nemusí provádět samotný webmaster. V současné době jsou webové stránky s redakčním systémem obvyklé:

- v komerční sféře (internetové obchody např. www.megapixel.cz)
- ve zpravodajství (zpravodajské servery, elektronické časopisy např. www.idnes.cz)
- osobní stránky - blogy
- weby pro sociální sítě (spolužáci, hi5, facebook...)
- firemní stránky, firemní intranet

Základní role uživatelů CMS:

- **čtenář** stránek (pouze sleduje obsah stránek, nemůže do jejich obsahu zasahovat)
- **editor** stránek (mění, přidává příspěvky do vymezeného prostoru webu)
- **administrátor** stránek (oproti editorovi může navíc přidělovat přístupová práva, zasahovat do stromové struktury webu...)

Základní funkce CMS:

- tvorba, úprava a publikace článků prostřednictvím webového rozhraní
- správa uživatelů a přístupových práv
- řízení přístupu k dokumentům
- správa diskusí, komentářů
- správa obrázků, galerií
- kalendář, statistiky přístupů

CMS je software naprogramovaný v jazyce, který je podporován webovým serverem (PHP, Java, ASP.NET).

Na trhu s CMS je k dispozici kromě nepřeberného množství komerčních systémů i systémy volně šiřitelné, např.: **Drupal**, **Joomla**, **WordPress**...