

# XHTML a tvorba webu

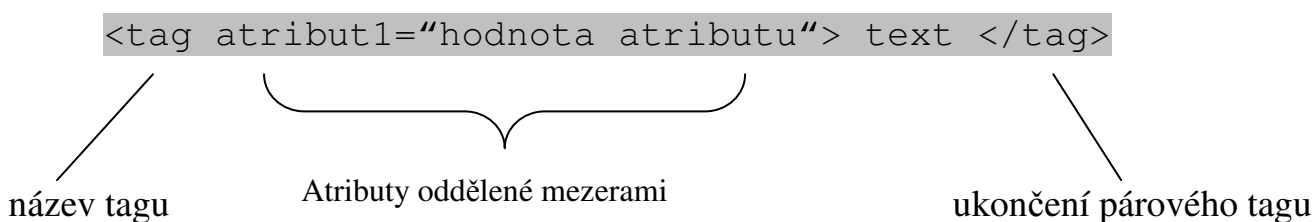
## 1. Historie a základní struktura

### HTML – HyperText Markup Language

- HTML 0.9 – první verze 1991
- Značkový jazyk pro zápis dokumentů na webu
- Vkládání do textu obrázků, hypertextových odkazů, logické členění a zvýraznění
- Existuje několik verzí. Poslední je HTML 4.01.
- HTML se dále nevyvíjí.

HTML se skládá ze značek:

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



Př.:

```
<b>tučné písmo</b>
```

```

```

Pozor! – nepárový tag by měl být ukončen lomítkem

### 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.

### XHTML 1.0 Strict

čistě strukturální značkování, neobsahuje žádné značky spojené s formátováním vzhledu

### XHTML 1.0 Transitional

povoluje atributy pro formátování textu a odkazů v elementu body a některé další atributy

### XHTML 1.0 Frameset

používá se při použití rámců pro rozdělení okna prohlížeče na dvě nebo více částí

### XHTML 1.1

Modulově založené, podobné 1.0 strict, ale umožňující rozšíření...

## **Některé základní odlišnosti od HTML:**

- Všechny atributy tagů mají hodnoty v uvozovkách
- Zákaz křížení tagů ( <b><i> text </i></b> )
- Všechny tagy a atributy jsou malými písmeny (case sensitive)
- Nepárové tagy končí lomítkem např.: <br />
- Párové tagy jsou párové povinně
- Všechny atributy musejí mít hodnotu
- Mírně odlišný zápis javascriptů a stylů
  - o <script><![CDATA[ zde bude kód skriptu ]]></script>
- Dokument má mít XML prolog.
- Dokument požaduje správný doctype.
- Využívání HTML entit pro spec. znaky
  - o Např.: &amp;
  - o <http://www.webtvorba.cz/xhtml/html-entity.html>

Příklad základní stránky:

Deklarace xml s využitím kódování

```
<?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">
```

DTD – deklarace typu dokumentu (zde XHTML 1.0 Trans.)

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

Základní tag HTML s atributy pro jmenný prostor, ...

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

Sekce hlavičky s titulkem a metatagem pro kódování (kompatibilita)

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

Tělo dokumentu (zde s jedním odstavcem)

```
</html>
```

## Poznámka ke kódování češtiny:

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

<http://cs.wikipedia.org/wiki/Ascii>



Nejpoužívanější pro češtinu:

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

## Validita kódu:

HTML i XHTML jsou standardy W3C a norma použitá v záhlaví a definici dokumentu by tedy měla být také v dokumentu striktně dodržována a používána.

Psát kódy podle norem (validní), možno ověřit na internetu

<http://validator.w3.org/>

## ... a 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)
- viz. Např. ACID3 test pro podporu standardů v prohlížeči
  - o <http://acid3.acidtests.org/>

## JAK a v ČEM?

- Editory
  - o WYISWYG vs. klasické (strukturní)
  - o PsPad, Poznámkový blok, PHPeditor, HomeSite, Emacs, Dreamweaver, ..
- www prohlížeče
  - o kontrolovat zobrazení v nejpoužívanějších! (IE, Firefox, Opera, ...)
- další sw
  - o ftp klient, grafické aplikace, ...

*Pozn.: Doporučuji používat krátké názvy souborů bez diakritiky a mezer!*

*Pozn.2: Hlavní soubor webu se většinou jmenuje index.html (nebo obdobně).*

## 2. XHTML tagy – základní přehled

`<tag> text </tag>` - párový xhtml tag (pozor xhtml je case-sensitive)

Podle normy jsou všechny xhtml tagy párové, ale u tagů kde není třeba aby obsahoval nějaký text je možné zápis zkrátit (budeme jim říkat tagy „nepárové“)

`<tag />` - “nepárový” tag

Uvnitř tagu nemusí být jen text, ale i další tag. Tagy se však nesmí vzájemně křížit!

```
<tag1>
  cokoli nebo nic
  <tag2> ...text... </tag2>
  cokoli nebo nic
</tag1>
```

Pozn.:  
Formátování kódu pro lepší přehlednost!  
  
ENTER v kódu se jako odřádkování při zobrazení neprojeví! Veškeré sekvence mezer jsou brány jako jedna mezera ....

### Atributy tagů:

- Tagy mohou dále mít různé **atributy**.
- Atributy slouží k určení vzhledu tagu nebo k dodání další funkční informace pro tag.
- Striktnější verze xhtml nedovolují atributy na formátování vzhledu dokumentu.
- Atributy se zapisují za název počátečního tagu
- V případě více atributů se oddělují mezerou
- Každý! atribut má svoji hodnotu – hodnota je zapsána v uvozovkách

```
<tag atribut1="hodnota1" atribut2="hodnota2"> text </tag>
```

Pozn.:

Sekvenci od začátku tagu po jeho ukončení nazýváme *elementem* xhtml.

- Blokové – zalomený blok – např. odstavec
- Inline – uvnitř textu nezalomené
- Nahrazované – nahrazené obsahem, např. obrázek
- *Není obecně možné vkládat libovolný tag do libovolného – např. do inline blokový*

## Přehled vybraných (X)HTML tagů

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). Z tohoto důvodu nejsou formátovací atributy příliš zmiňovány.

Základní tagy – viz. příklad výše:

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

Tagy v sekci HEAD:

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

Př.:

```
<title> titulek zobrazený jako název okna/záložky </title>
```

Meta tagy:

Popisek stránky

```
<meta name="description" content="textík" />
```

Obsah - čeština

```
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-2" />
```

Klíčová slova, autor, pro vyhledávače, google, ...

```
<meta name="keywords" content="klíčová, slova" />
```

```
<meta name="author" content="Jméno Příjmení" />
```

```
<meta name="robots" content="index, follow" />
```

```
<meta name="googlebot" content="snippet, archive" />
```

Vložení externího souboru s definicemi CSS stylů (trochu předbíháme):

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

## Skripty

```
<script type="text/javascript" src="skript.js"></script>
```

```
<script language="javascript" type="text/javascript">  
... kód skriptu  
</script>
```

```
<script type="text/javascript">  
/*  */<br/>... kód skriptu ...<br/>/* ]]&gt; */<br/>&lt;/script&gt;</pre></div><div data-bbox="93 334 204 353" data-label="Section-Header"><h2><u>Poznámky:</u></h2></div><div data-bbox="93 372 963 411" data-label="Text"><p>Zakomentovaný kód nebude zobrazen. Také vhodné jako popisky u složitějších konstrukcí (ne že by v XHTML mohlo být něco složitějšího)</p></div><div data-bbox="152 429 397 448" data-label="Text"><pre>&lt;!-- poznámka --&gt;</pre></div><div data-bbox="93 486 314 505" data-label="Section-Header"><h2><u>Formátování odstavců:</u></h2></div><div data-bbox="93 525 734 585" data-label="Text"><table><tr><td><code>&lt;span&gt;</code></td><td>-</td><td>oddělení části textu s jiným formátováním</td></tr><tr><td><code>&lt;div&gt;</code></td><td>-</td><td>blok textu (zalomuje řádek před i za)</td></tr><tr><td><code>&lt;p&gt;</code></td><td>-</td><td>odstavec (vytváří před a za mezeru)</td></tr></table></div><div data-bbox="93 603 796 625" data-label="Text"><table><tr><td><code>&lt;h1&gt; ... &lt;h6&gt;</code></td><td>-</td><td>nadpis 1. – 6. Úrovně – předdefinované velikosti</td></tr></table></div><div data-bbox="93 644 570 664" data-label="Text"><table><tr><td><code>&lt;br&gt;</code></td><td>-</td><td>odřádkování spíše <code>&lt;br /&gt;</code></td></tr></table></div><div data-bbox="93 685 541 743" data-label="Text"><table><tr><td><code>&lt;center&gt;</code></td><td>-</td><td>vycentrování</td></tr><tr><td><code>&lt;pre&gt;</code></td><td>-</td><td>předformátovaný text</td></tr><tr><td><code>&lt;hr&gt;</code></td><td>-</td><td>horizontální čára</td></tr></table></div><div data-bbox="93 782 450 801" data-label="Section-Header"><h2><u>Formátování textu (nahrazovat CSS):</u></h2></div><div data-bbox="93 820 521 923" data-label="Text"><table><tr><td><code>&lt;b&gt;</code></td><td>-</td><td>tučně</td></tr><tr><td><code>&lt;i&gt;</code></td><td>-</td><td>kurzíva</td></tr><tr><td><code>&lt;u&gt;</code></td><td>-</td><td>podtrženo</td></tr><tr><td><code>&lt;sup&gt;&lt;sub&gt;</code></td><td>-</td><td>indexy</td></tr><tr><td><code>&lt;font&gt;</code></td><td>-</td><td>fonty – ne XHTML</td></tr></table></div>
```

## Seznamy

```
<ul>
  <li>bod 1</li>
  <li>bod 2</li>
</ul>
```

- nečíslovaný seznam (unordered list)  
- položka seznamu (list item)

```
<ol> ... </ol>
```

- číslovaný seznam – položky jsou opět <li>

```
<dl>
  <dt> definiční termín</dt>
  <dd> definice ...</dd>
  <dt> ... </dt>
  <dd> ... </dd>
  ...
</dl>
```

- seznam definic

## Odkazy:

parametry: href, target, title

Zobrazený text  
(může být i jiný tag  
např. pro obrázek)

Odkaz na www umístění: /

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



href značí cíl, psát včetně protokolu

Absolutní odkaz:

```
<a href="C:\složka\podsložka\soubor.html">odkaz</a>
```

Relativní odkaz:

```
<a href="soubor.html">odkaz</a>
<a href="složka\soubor.html">odkaz</a>
<a href="..\složka\soubor.html">odkaz</a>
```

- .. o úroveň výše



- . aktuální složka

Odkaz na místo v dokumentu:

```
<tag id="umistenil">text</tag>
```

- U libovolného tagu použijeme libovolné pojmenování, na které se následně budeme odkazovat. Po odkázání se zobrazí přímo ta část v dokumentu (odrolování záleží na délce obsahu)

```
<a href="#umistenil">začátek</a>
```

```
<a href="soubor.html#umistenil">začátek</a>
```

Další atributy tagu a:

```
title="titulek zobrazený po najetí myši na odkaz"
```

```
target="_blank" - otevření stránky v novém okně
```

Atribut target lze dále použít k otevírání v různých rámcích (částech stránky) viz. níže.

Tabulky:

```
<table summary="popis tabulky"> - tabulka
  <caption> nadpis </caption> - nadpis tabulky
  <tr> - řádek
    <th>1</th> - hlavičková buňka
    <th>2</th>
  </tr>
  <tr>
    <td>aaa</td> - buňka tabulky
    <td>bbb</td>
  </tr>
</table>
```

atributy `colspan="n"` a `rowspan="n"` slouží ke sloučení n (počet) buněk

Obrázky:

```

```

atributy:

```
src - adresa zdroje obrázku, pozor na pojmenování a příponu
```

```
alt - alternativní text - povinné
```

`width, height` - šířka, výška – vhodné zadávat, nahrazovat CSS, používat spíše reálnou velikost – tedy velikost předem upravit v grafickém editoru  
`border, ....`

### Klikací mapy:

Pojmenování mapy (name – kompatibilita)

```
<map id="mapa" name="mapa">
```

```
<area href="prvni.html" alt="" shape="rect"
      coords="80, 80, 130, 130">
<area href="druha.html" alt="" shape="rect"
      coords="20, 20, 50, 50">
```

Libovolný počet oblastí

```
</map>
```

### Parametry area:

`shape` - vzhled oblasti pro kliknutí: rect, poly, circle

`coords` - souřadnice podle typu oblasti, 0,0 je vlevo nahoře

### Samotný obrázek s použitou mapou:

```

```

### Vkládání objektů:

*jen některé příklady pro doplnění mimo probíranou látku*

```
<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>
```

```
<tag onmouseover="funkceJavaScriptu1(2,50);" />
<tag onmouseover="this.style.backgroundColor='#FFDFD8'"
onmouseout="this.style.backgroundColor='#f3aea2'">
```

### Rámce:

Umožňují rozdělit stránku na několik podstránek (např. levé menu s navigací a vpravo obsah). Nebo umožňují vložit na určité místo stránky rámec s jinou stránkou. Dříve hojně používané, dnes nevhodné díky nedostatkům při navigaci, indexování vyhledávači, tisku, apod.

```
<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, ...

```
<a target="I2" href="002.htm"> - odkaz do rámu
```

### Plovoucí rámce:

Nerozdělují celou stránku, ale na jistém místě ve stránce umožňují vložení jiné stránky. O něco vhodnější využití.

```
<iframe id="main" name="main" src="stranka.html" width="484"
frameborder="0" scrolling="no" > alternativní obsah
</iframe>
```

```
<a target="main" href="002.html">
```

### Formuláře:

Vytvoření formuláře je velmi snadné. Pro jeho zpracování se však často používají různé (serverové) skripty – php apod.

```
<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

Př.:

```
<input type="text" maxlength="64" size="50"
class="formvstup" name="jméno" value="Martin">
```

```
<select name="vybersi"> - vyběrací nabídka
  <option value="jedna">jedna text</option>
  <option value="dva">dva 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, ...

### 3. CSS – kaskádové styly

#### *CSS - Cascading Style Sheets*

- CSS (Cascading Style Sheets) je soubor metod sloužících ke grafické úpravě (X)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 v tomhle ohledu s využitím CSS.
- I CSS patří pod W3C a je možné ověřit validitu vytvářených dokumentů.
- Většina vlastností se dědí níže strukturou HTML dokumentu

#### **Základní zjednodušená CSS syntaxe:**

Další podrobnosti jsou uvedeny níže

```
Název_vlastnosti1: hodnota;  
Název_vlastnosti2: hodnota1 hodnota2; Název_vlastnosti3:  
hodnota;
```

#### **CSS lze v dokumentu využít několika způsoby:**

Všechny tři možnosti lze samozřejmě i kombinovat, ale u složitějších stránek je potřeba si v tom zavést určitý pořádek. Vhodnější je pak mít veškeré styly externě v souboru (souborech) a oddělit tak obsah od formy ...

- 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. Pozor - uvnitř uvozek atributu style se zapisuje CSS kód (a NE HTML) - syntaxe.

```
<div style="border: 2px solid black; width: 180px;" > ...
</div>
```

Rámeček kolem odstavce, šířka odstavce

- 2) Stylovisem v hlavičce HTML dokumentu. Styly pro celý dokument (soubor) jsou uvedeny mezi tagy `<head></head>`. Uvnitř párového tagu (ne atributu) `<style>` uvedeme nastavení stylu pro jednotlivé elementy dokumentu.

```
<head>
```

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

```
div {
    border: 2px solid black;
    width: 180px;
}
.muystyl {
    text-color: red;
    text-decoration: underline;
}
```

Dva definované styly pro odstavce (div) a vlastní pojmenování. Pozor zde jsou v syntaxi složené závorky.

```
</style>
```

```
</head>
```

- 2) Stylovisem 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>
```

(jiné varianty:)

```
<style type="text/css">@import "styl.css";</style>
<style type="text/css">@import
  url("http://www.neco.cz/styl.css");</style>
```

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

„selektor“ určuje element platnosti ...

Css vlastnosti oddělené středníkem

```
div { border: 2px white; width: 180px }
/* nadefinuje styl pro tag div, tag div teď bude vždy vypadat, tak jak jej nastavíme */
```

komentář

Nemusíme se však omezovat pouze na selektory v podobě existujících XHTML tagů, ale použít vlastní pojmenování:

```
.mujstyl { text-color: red; text-decoration: underline; }
/* definujeme si vlastní třídu a tu pak budeme používat podle potřeby (přiřazovat ji jednotlivým tagům)*/
(tag.mujstyl { ... } – aplikuje se jen na elementy tag)
```

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 hlavně při zpracovávání skriptů a v tom, že id je unikátní pro jeden element na jedné stránce

Deklarace pro více elementů současně:

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

Kontextová deklarace (pro tag a uvnitř tagu td):

```
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
background-position	-	pozice pozadí
font-family	-	písma – konkrétní názvy
font-style	-	styl (italic, normal, ...)
font-size	-	velikost (% , px, large, pt ...)
font-weight	-	tloušťka (normal, bold, ...)
text-decoration	-	dekorace textu (underline, overline, ..)
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)
list-style-type	-	typ odrážky seznamu
list-style-image	-	obrázek místo odrážky
list-style-position	-	pozice odrážky



atd .... opravdu spousta dalších možností a vlastností

Přehled např:

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

<http://www.webtvorba.cz/css/css-vlastnosti.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;  
}
```

Pro písma se používá rodina písem. Písma se nepřenaší, takže bude použito první nalezené na cílovém PC. Použít podobná písma (patková, bezpatková, ...)

U border se udává šířka, styl a barva

## Pozicování v CSS

Pomocí CSS je možné HTML entity pozicovat a určit tak jejich umístění na stránce:

```
<span style="position: relative; top: 20px">  
    relativně umístěný text  
</span>
```

```
<span style="position: absolute; top:10px; left: 10px">  
    absolutně umístěný text  
</span>
```

Další případné atributy jako `z-index` určí, jak bude viditelné případné překrývání. Při centrování: v IE se vystředí body a v Mozille se divu dá `margin: auto`

## Barvičky v CSS

- Slovník (anglickým) zápisem

```
o color: blue
```

- RGB

```
o color: rgb(100%, 0%, 0%)
```

```
o color: rgb(255, 0, 0)
```

```
o color: #FF0000
```

Na zobrazení barev na webu obecně pozor, různé prohlížeče, různé monitory, různé platformy, apod.

## 4. Tvorba web layoutu a grafika na webu

Vytvoření kompletní stránky pro web, která má mít také zajímavý design, úpravu, navigaci a další atributy není tak snadné jak by se mohlo podle výše uvedených informací zdát. Nejde jen o pochopení a zvládnutí XHTML a CSS (případně dalších), ale také o to jak s nimi správně pracovat a využívat je v praxi. Tato stránka se učí o poznání hůře než syntaxe. Dále je třeba umět pracovat s grafickými aplikacemi a k tomu mít určité grafické cítění, nápad apod.

### Grafika na webu

- Vkládané obrázky na webu by měly být obrazově kvalitní (rozlišení, barevná paleta) a většinou komprimované. Nejčastěji využíváme formátů JPEG, GIF, PNG, apod.
- Obrázky na webu vystupují ve dvou rovinách. Jedna pro vlastní obsah webu a jednak pro grafický vzhled webu.
- Grafika na webu je většinou umísťována jako pozadí HTML elementů.
- V praxi je možné kombinovat při návrhu jak vektorové, tak i rastrové grafické aplikace.
- Nejlépe je začít „skicou“ stránky.
- Mezi aplikacemi je široký výběr (i freeware).
  - Adobe Photoshop, Adobe Fireworks, Pain.Net, GIMP, Pixel, PaintShop, ImageReady, ...
  - Některé aplikace umí vygenerovat i příslušné HTML/CSS
- Dobrý grafik udělá hezkou grafiku i v Malování, špatný ani v nejnovějším Photoshopu.

### Layout stránky

Pod tímto pojmem se většinou rozumí rozvržení stránky jako celku. Tedy kde co na stránce bude umístěno. Jak bude řešena navigace, menu, text, reklamy, rámečky, obrázky apod.

Ideální je začít s tužkou a papírem nebo grafickou aplikací (typu např. Photoshop) a vzhled stránky si rozvrhnout a nakreslit. Dál takto získáme grafiku (obrázky) a tvoříme pomocí HTML, CSS kód layoutu stránky.

Šířka, výška, x, y pozice		
Šířka, výška, x, y pozice	Šířka, výška, x, y pozice	Šířka, výška, x, y pozice
	Šířka, výška, x, y pozice	

Pro vytvoření celku stránky máme v zásadě několik možností (v praxi často budeme kombinovat a není nutné striktně dodržet jednu):

### 1. Ne HTML technologie

Stránku samozřejmě můžeme tvořit pomocí i jiných technik než HTML a spol. Nejčastěji se v této souvislosti využívá Adobe Flash. Tento přístup má jistě své výhody i nevýhody. Takováto tvorba ale není cílem tohoto předmětu.

### 2. Rámcový layout

Využívá tagů `<frame>` nebo `<iframe>`. Tento způsob je zastaralý, ale i v XHTML podporovaný.

Pro vlastní tvorbu je to velmi jednoduchý postup.

#### FRAME:

Hlavní stránku tvoří definice rámců. V jednom rámcu je umístěno např. menu a v druhém vlastní stránka (rámců samozřejmě může být více). V menu jsou pak odkazy na jednotlivé stránky směřovány do rámce s obsahem (parametr `target`).

#### IFRAME

Stránka je „klasicky“ rozvržena, ale na místě např. vlastního obsahu je vložen `iframe` odpovídající velikosti a do něj jsou opět směřovány odkazy z menu.

Oba postupy mají nevýhody pro indexování vyhledávači, tisk, odkazy na konkrétní stránky, omezený obsah a velikost rámce, problémy na prohlížečích nepodporujících rámce, apod.

### 3. Tabulkový layout

Při tomto přístupu využíváme pro rozvržení stránky `<table>` a vše co s tím souvisí. Celá stránka může být tvořena tabulkou (i poměrně komplikovanou) a

případně v jednotlivých buňkách tabulky mohou být umístěny další tabulky. Díky vlastnostem tabulek (i CSS) je možné poměrně kvalitně a přesně rozvrhnout celý web a jeho strukturu.

Nevýhodami jsou hlavně komplikovaná struktura celého webu, která je někdy špatně přehledná. Většinou (pokud striktně nepoužíváme CSS) ne dobré oddělení vzhledu od obsahu. Případné problematické načítání a zobrazení v prohlížečích.

#### 4. DIV/CSS layout

Celá stránka je rozvržena jako množina tagů `<div>` (je ale možné obdobně pracovat i s dalšími tagy). Tyto tagy jsou následně přes CSS pozicovány a určován další jejich vzhled. Výhoda je v relativně přehledné kódu a oddělení obsahu od určení vzhledu (externí CSS soubor).

Nevýhodami jsou ne vždy stejná podpora pro CSS styly v jednotlivých prohlížečích. Složitější CSS kód i jeho tvorba.

<http://interval.cz/hotova-reseni/pro-css/>

#### 5. Skriptování

K tvorbě layout je možné využít i další technologie jako JavaScript, PHP (hlavně include) apod. Nevýhodou je nutnost alespoň elementární znalosti těchto technologií a případná podpora ze strany serveru. Složitější testování.

## 5. Umístění na web

### Doména

- Není nutně třeba, můžeme pracovat jen s IP adresou, ale nevhodné
- Placená doména druhého řádu
  - o [www.domena.cz](http://www.domena.cz) (sk, com, eu, ...)
  - o nutno registrovat u registrátora
  - o správce cz domény: <http://www.nic.cz/>
  - o cena několik desítek až stovek ročně
- Doména třetí úrovně nebo „složka“ na doméně druhého řádu
  - o [domena.poskytovatel.cz](http://domena.poskytovatel.cz)
  - o [www.poskytovatel.cz/domena](http://www.poskytovatel.cz/domena)
  - o většinou zdarma v rámci freehostingu nebo podobných služeb

### Hosting

- **placený X freehosting**
- Rozdíly v parametrech:
  - o Cena 0 – tisíce za měsíc (v případě (dedikovaného) serverhostingu i více)
  - o Kapacita pro vlastní obsah
  - o Připojení, agregace, dostupnost, ...
  - o Instalované služby (ASP, PHP, databáze, mail, admin, .... )
- Spousta freehostingů na webu (googluj)
  - o Např.: <http://www.hostingy.cz/hosting-zdarma.html>
- Přenos souborů většinou přes FTP nebo www rozhraní

**Pozor** na přenos na server a zvolený hosting: Unix (like) systémy jsou case-sensitive na rozdíl od MS Windows. Možné jsou i problémy s českými znaky v názvech souborů (odkazů).

### Vyhledávače

Pro fulltextové vyhledávání jsou informace z webu automaticky sbírány roboty a stránky tak jsou indexovány. Prvním kritériem aby stránka byla nalezena, je tedy aby na ni vedl nějaký odkaz z již indexované stránky.

Parametrem pro fulltext vyhledávání je existence klíčového slova na dané stránce. Záleží ale i na tom, kde je slovo umístěno (adresa, titulek, ...), kolik odkazů vede tam a zpět, ... Je tedy vhodné aby na stránku vedl odkaz (třeba z katalogu na Seznamu).

Pro zaindexování je vhodné i mít validní stránky (nebo alespoň bez určitých chyb). Vyhledávače indexují podle obsahu a tak například neuzavření uvozovek může vést k ignorování obsahu dokumentu.

### *PageRank (PR)*

- Číslo „popularity“ webu pro vyhledávače apod.
- Vymyslel Larry Page (Google)
- Základní princip je: pokud na web odkazuje více webů (s vysokým pagerankem) pak i tento web je kvalitní
- Počítaný zvlášť pro každou stránku webu
- Google:
  - o „link:www.stranka.cz“ vypíše odkazující stránky na web (ne vše)
- Zvyšovat přidáváním odkazů na moje stránky „někam“

### *SEO*

[http://cs.wikipedia.org/wiki/Search\\_Engine\\_Optimization](http://cs.wikipedia.org/wiki/Search_Engine_Optimization)

Zdroje

<http://www.webtvorba.cz/>

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

<http://www.google.cz/>