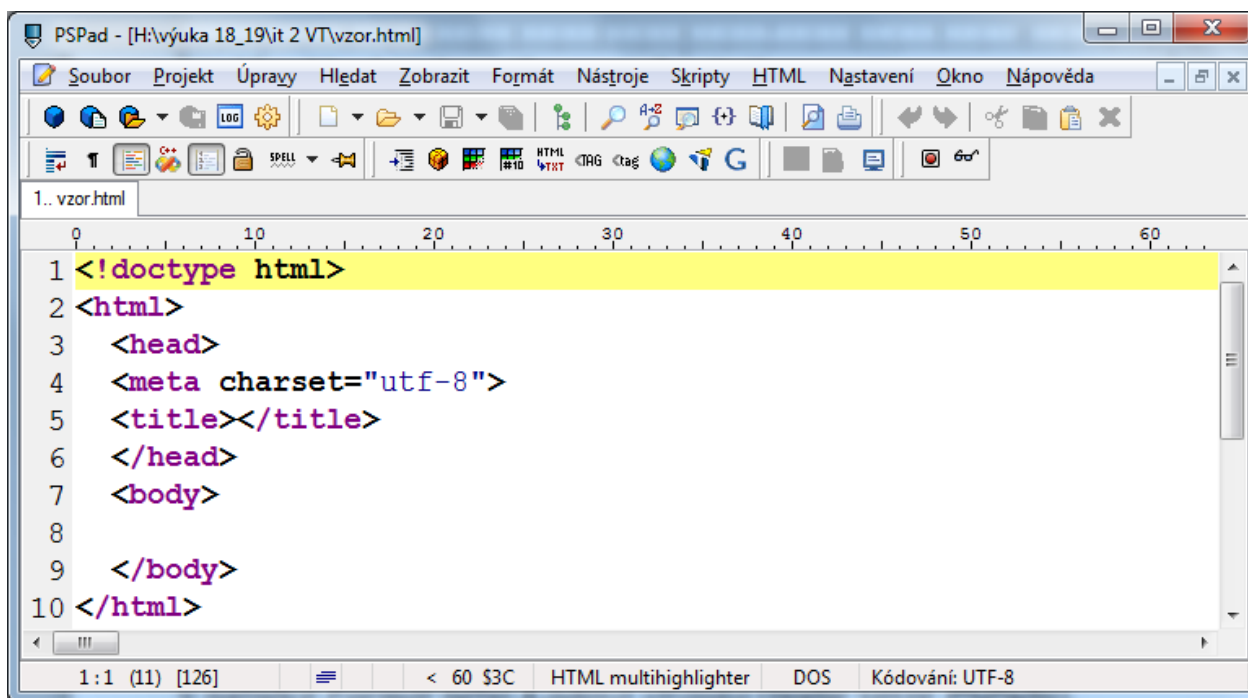


## 1. Základní kostra HTML dokumentu

HTML dokument se skládá z tzv. **značek (tagů)**. Značky zapisujeme mezi znaky < >. Většina značek je párových, což znamená, že značku je třeba ukončit. Koncová značka je stejná jako úvodní, ale má navíc lomítko. Nepárové značky se neukončují

Př. <p>...</p> párová                      <br>                      nepárová



```
1 <!doctype html>
2 <html>
3   <head>
4     <meta charset="utf-8">
5     <title></title>
6   </head>
7   <body>
8
9   </body>
10 </html>
```

Struktura html dokumentu

Prvek **<!doctype html>** uvádí verzi HTML. Ve starších HTML dokumentech má tento prvek složitější zápis. Uvedená verze odpovídá použití HTML5.

Značka **<head>...</head>** vymezuje hlavičku dokumentu. Hlavička je část, která se nezobrazuje. Součástí hlavičky je značka **<title>...</title>**, ve které se uvádí titulek dokumentu. Tento titulek se zobrazí v titulkovém pruhu prohlížeče.

Značky **<meta ... >** mají různý význam, obsahují např. info o stránce, použité znakové sadě, obsahu, přesměrování,... V našem případě definuje kódování češtiny UTF-8. Aby stránka byla správně česky, musí se kódování definované na stránce (v našem případě UTF-8) shodovat s kódováním nastaveným v PS-Padu v nabídce *Formát* nebo *Kódová stránka* (podle verze PSPadu).

Značka **<body>...</body>** vymezuje tělo dokumentu, do kterého se píše veškerý obsah, který má být zobrazen.

Většinu značek lze dále upřesnit pomocí atributů.

**<značka atribut="hodnota atributu" ...>...</značka>**

Ve výše uvedeném příkladu má atribut pouze značka **<meta ...>**.

Některé značky mají část svých atributů povinných, protože bez nich by neměly smysl. Např. když vkládáme obrázek, tak musíme říct jaký, ale už třeba nemusíme upřesňovat zarovnání, velikost apod.

## 2. Členění dokumentu

### Vkládání nadpisů

**<hn>text nadpisu</hn>** *n* – číslo úrovně nadpisu (1-6, menší číslo = vyšší důležitost nadpisu)

Př. <h1> Toto je nadpis úrovně 1</h1>

## Formátování odstavců

Veškerý text umístěný v těle dokumentu by měl být ohraničen značkami pro zalomení odstavce `<p>...</p>`. Chceme-li v rámci odstavce psát text na nový řádek, musíme vložit značku pro zalomení řádku `<br>`.

## Horizontální čára

`<hr>`

## 3. Kaskádové styly - CSS

Použitím kaskádového formátování oddělíme obsah stránky od jejího vzhledu. O obsah se postará HTML kód a vzhled zajistí CSS. Druhým důvodem pro vznik kaskádových stylů bylo rozšířit škálu možností pro formátování vzhledu stránek.

Kaskádový styl můžeme vkládat trojím způsobem:

### a) Řádkový styl

Popis stylu vkládáme přímo do konkrétní značky pomocí atributu **style**

Př. `<h2 style="text-align:center">Toto je nadpis 2. úrovně</h2>`

Takto vložená definice stylu se vztahuje pouze na jeden konkrétní element, v tomto případě nadpis 2. úrovně, který zarovná na střed. Atribut **style** můžeme použít u většiny značek. Jako hodnotu tohoto atributu můžeme uvést i více definic formátu příslušné značky. Jednotlivé definice oddělujeme středníkem.

Př. Chceme-li nadpis zarovnat na střed a obarvit červeně:

`<h2 style="text-align:center; color:red">Toto je nadpis 2. úrovně</h2>`

Pokud bychom danou vlastnost chtěli uplatnit i u dalšího nadpisu 2. úrovně, museli bychom styl definovat znovu.

Řádkový styl odporuje filozofii kaskádového formátování, tedy oddělení vzhledu od obsahu. Používáme ho, pokud chceme změnit vlastnost pouze jedné značky. V HTML5 je doporučeno místo řádkového stylu používat výhradně třídy (vysvětlíme si později).

### b) Hlavičkový styl

Popis stylu vkládáme do hlavičky dokumentu pomocí značky `<style>...</style>`.

```
Př. <head>
  <title>...</title>
  <style>
    h2 { text-align:center;
        color:red }
  </style>
</head>
<body>
  ...
  <h2>Toto je nadpis 2. úrovně</h2>
  ...
</body>
```

Styl definovaný v hlavičce dokumentu se vztahuje na všechny značky dané stránky, tj. pokud kdekoliv na stránce použijeme značku `<h2>`, vždy bude červeně a na střed.

Pokud se v dokumentu setkají (v definici dané vlastnosti) řádkový a hlavičkový styl, pak má přednost řádkový.

Př. Pokud definujeme hlavičkový styl jako v předchozím příkladu a v těle dokumentu použijeme zápis:

`<h2 style="color:blue">Toto je nadpis 2. úrovně</h2>`,  
pak bude tento konkrétní nadpis sice na střed, ale modrý.

Pokud vytváříme rozsáhlejší stránku, pak by definice stylu všech prvků stránky učinily hlavičku dokumentu velmi rozsáhlou a nepřehlednou. Jestliže navíc budeme chtít použít jednotný styl pro více stránek, museli bychom definici kopírovat do každého dokumentu zvlášť a jakékoliv dodatečné změny bychom museli psát znovu do každého zvlášť. Proto hlavičkový styl používáme tam, kde chceme definovat vlastnosti pouze pro jednu konkrétní stránku.

### c) Externí dokument

Veškeré informace o formátech prvků stránky jsou uloženy v samostatném souboru s příponou **.css**.

Každá stránka, na které chceme styl použít, obsahuje v hlavičce značku **<link>**.

Př. Soubor se stylem se jmenuje **styl.css** a je ve stejné složce jako daná stránka

Obsah souboru:

```
h2 { text-align:center; color:red }
```

V hlavičce stránky:

```
<link rel="stylesheet" href="styl.css" >
```

Definovaný formát se projeví na všech stránkách najednou. Pokud se rozhodneme některou z definic změnit nebo přidat, pak změnu uvádíme pouze jednou, v externím souboru, a nemusíme zapisovat do každé stránky zvlášť.

Pokud je k dokumentu připojena externí definice stylu a navíc obsahuje hlavičkový styl, pak má u definic značek, ve kterých se překrývají, přednost ten styl, který je uveden jako druhý. V níže uvedeném příkladu to tedy bude styl hlavičkový. Je to tak i vhodnější uvádět.

```
<head>
```

```
  <title>...</title>
```

```
  <link rel="stylesheet" href="styl.css">
```

```
  <style>
```

```
    ...
```

```
  </style>
```

```
</head>
```

Toto pořadí připojení stylů, tedy nejdřív **<link>** potom **<style>...</style>**, je vhodnější, protože umožňuje hlavičkovému stylu blíže upřesnit vlastnosti nastavené pro všechny ve stylu externím.

### Zápis stylu

Za názvem značky, kterou stylem formátujeme, jsou vypsány její CSS vlastnosti ve složených závorkách.

Vlastnosti se zapisují ve tvaru: *název vlastnosti:hodnota*. Jednotlivé vlastnosti se oddělují středníkem.

Př. `p {text-align:left; color:navy; font-size:12pt;}`

Za poslední hodnotou ; být nemusí, ale je vhodné ho psát, abychom při přidávání dalších vlastností na něj nezapomínali.

Pro formátování více značek najednou máme dvě možnosti:

- názvy značek oddělíme čárkou a vlastnosti v {} platí pro každou značku zvlášť
- názvy značek neoddělíme čárkou a vlastnosti v {} platí pouze při kombinaci značek v stejném pořadí, v jakém byly uvedeny v definici stylu.

Př. definice stylu: `h1, h2, h3 {color:red; text-align:center;}`

při použití budou všechny nadpisy h1, h2 a h3 červené a zarovnané na střed.

definice stylu: `a img {...}`

při použití se vlastnosti definované v {...} jen pro obrázky, které jsou uvnitř odkazu.

### Třídy

Při výše uvedeném způsobu definice CSS lze formátovat vždy pouze konkrétní značku a pouze jedním způsobem. Někdy potřebujeme mít pro určitou značku více variant. Tuto možnost nám dává využití tříd.

Třídy se definují dvojím způsobem:

- a) za název značky napíšeme tečku, název třídy a pak ve složených závorkách CSS vlastnosti. Použití vlastností pak vyvoláme užitím atributu **class** u názvu značky. Takto definovanou třídu lze použít pouze u té značky, pro kterou jsme jí definovali.
- b) nepíšeme název značky, ale rovnou tečku, název třídy a pak ve složených závorkách CSS vlastnosti. Použití třídy vyvoláme opět pomocí **class**. Takto definovanou třídu lze použít u všech značek, pro které má smysl.

Př. a) definice stylu: `p.textbr {font-weight:bold, color:red}`  
 v dokumentu pouze: `<p class="textbr">...</p>`  
 b) definice stylu: `.textbr {font-weight:bold, color:red}`  
 v dokumentu: `<p class="textbr">...</p>`,  
 ale i např.: `<h1 class="textbr">...</h1>`

## Řádkový element SPAN

Chcete-li měnit formát textu uvnitř jedné značky (např. několik slov v odstavci, ne celý odstavec), můžete vložit značku `<span>...</span>`

Př. definujeme třídu

```
.zelena {color:green}
```

a použijeme zápis

```
<p>v textu je <span class="zelena">zelené</span> jen jedno slovo</p>
```

## 4. 1. Formátování textu pomocí CSS

### font-family (rodina písma)

Jako hodnota této vlastnosti se používá buď název konkrétního fontu, nebo obecný název fontu, tzv. rodinu. **Pozor** na používání **nestandardních** (exotických) **fontů!** Pokud uživatel nebude mít tento font k dispozici, dojde k deformaci stránky popř. k chybné interpretaci českých znaků. Proto uvádíme raději standardní (běžně používané) fonty, popřípadě můžeme uvést více fontů oddělených čárkou. V takovém případě se použije první uvedený font, není-li k dispozici, použije se druhý, třetí atd. Nebo lze uvádět **rodinu písma**, pak se použije písmo z této rodiny podle nastavení prohlížeče.

```
Př. h1 {font-family: Arial, Helvetica, sans-serif;}
```

Rodiny písma:

Název	Popis	Příklad
<b>serif</b>	patkové písmo, obvykle proporcionální	Times New Roman, Garamond
<b>sans-serif</b>	bezpatkové písmo, obvykle proporcionální	Arial, Verdana, Futura, Univers
<b>cursive</b>	imitace psacího písma	Adobe Poetica, Snell Roundhand
<b>fantasy</b>	ozdobné písmo	Cottonwood, Sloha Geometrique
<b>monospace</b>	neproporcionální písmo	Courier

### color (barva písma)

Barvu definujeme buď názvem, nebo **RGB** definicí stejně jako např. u barevného pozadí stránky viz. kapitola 5.

### font-size (velikost písma)

Velikost písma uvádíme v **pt** (typografický bod), **px** (obrazkový bod - pixel), **in** (palec), **mm** (milimetr), **cm** (centimetr), **em** (šířka velkého písmene „M“), **ex** (výška malého písmene „x“), **%** (procento).

Číselná hodnota a jednotka se neoddělují mezerou!

Kromě zadání přesné velikosti písma lze použít i následující slovní hodnoty (konstanty) –

**xx-large** (cca. 30px, podle nastavení prohlížeče), **x-large**, **large**, **medium** (cca. 16px, podle nastavení prohlížeče), **small**, **x-small**, **xx-small** (cca. 10px, podle nastavení prohlížeče).

## font-style (sklon písma)

Hodnota	Popis zobrazení znaku
<b>normal</b>	běžné písmo – výchozí hodnota
<b>italic</b>	kurzíva, použije se znak kurzívy
<b>oblique</b>	skloněné písmo, „rovný znak bude upraven výpočtem“

Rozdíl mezi kurzívou a skloněným písmem je nepatrný. Pro kurzívu existují znaky, skloněné se vykresluje výpočtem. Pokud použijeme kurzívu a příslušný znak není k dispozici, bude skloněn taky výpočtem. V praxi se doporučuje používat raději *italic*.

## font-weight (tučnost písma)

Hodnota	Tučnost písma		Hodnota	Tučnost písma
<b>normal</b>	standardní		400	stejně jako normal
<b>lighter</b>	méně tučné		500	mírně tučné
<b>bold</b>	tučné		600	téměř jako bold
<b>bolder</b>	ještě tučnější		700	stejně jako bold
<b>100</b>	velmi tenké		800	tučnější než bold
<b>200</b>	tenké		900	nejtučnější
<b>300</b>	téměř jako normal			

Ne všechny prohlížeče podporují všechny uvedené hodnoty. V praxi se používá pouze **normal** a **bold**.

## font-variant (kapitálky)

**KAPITÁLKY JSOU VELKÁ PÍSMENA zmenšená na velikost malých písmen (tvar velkých písmen, ale píše se menší, takže lze rozlišit malé a velké písmeno). Starší prohlížeče kapitálky nepodporují, novější ano, pouze pokud pro daný znak neexistuje příslušný řez písma, nahradí kapitálky **verzálkami** (velkými písmeny).**

Hodnota	Vzhled písma
<b>normal</b>	standardní písmo (výchozí hodnota)
<b>small-caps</b>	kapitálky

## line-height (výška řádku - řádkování)

Hodnota	Vzhled řádku
<b>normal</b>	běžná výška řádku
<b>procento</b>	např. 130% - výška zvětšená o 30% oproti normálu
<b>násobek</b>	např. 1.3 – výška zvětšená 1.3 krát oproti normálu
<b>přesná výška</b>	např. 20px – přesně zadaná výška

## text-decoration (podtržení, nadtržení, přeškrtnutí)

Ne všechny prohlížeče podporují všechny hodnoty

Hodnota	Úprava textu
<b>none</b>	bez úpravy
<b>underline</b>	podtržené
<b>overline</b>	nadtržené (čára nad textem)
<b>line-through</b>	přeškrtnuté

Vlastnosti čáry lze blíže specifikovat:

### text-decoration-style

Hodnota	Druh podtržení
<b>solid</b>	plné
<b>double</b>	dvojité
<b>dotted</b>	tečkované
<b>dashed</b>	čárkované
<b>wavy</b>	vlnovka

### text-decoration-color

Barvu čáry definujeme buď názvem, nebo **RGB** definicí stejně jako např. u barevného pozadí stránky viz. kapitola 5.

## **text-transform (převod malá ↔ velká písmena)**

K užití této vlastnosti není žádný důvod, snad jen ošetření vstupních dat od uživatele.

Hodnota	Úprava textu
<b>none</b>	bez úpravy
<b>uppercase</b>	velká se převedou na malá
<b>lowercase</b>	malá se převedou na velká
<b>capitalize</b>	Všechna první velká

## **letter-spacing (vzdálenost mezi znaky)**

Hodnota **normal** znamená normální vzdálenost. Pro jiné vzdálenosti se uvádí hodnota ve všech jednotkách pro velikost písma (mm, cm, px, pt,...). Při kladné hodnotě se mezery zvětšují, při záporné se zmenšují (pozor na čitelnost).

## **word-spacing (vzdálenost mezi slovy)**

Princip i hodnoty jsou totožné jako u předchozí vlastnosti

## **text-indent (odsazení prvního řádku odstavce)**

Hodnota se zadává buď v délkových jednotkách (cm, mm, ...) nebo v % šířky bloku.

## **text-align (horizontální zarovnání textu)**

Hodnota	Zarovnání textu
<b>left</b>	vlevo, výchozí nastavení
<b>right</b>	vpravo
<b>center</b>	doprostřed
<b>justify</b>	do bloku

## **vertical-align (vertikální zarovnání textu)**

Hodnota	Zarovnání textu
<b>sub</b>	horní index
<b>super</b>	dolní index

## **4. 2. Formátování textu pomocí značek – textové tagy**

**<em>** část textu, která má větší význam než okolní text – zobrazuje se kurzívou

**<strong>** část textu, která má větší význam než okolní text, ještě větší než **<em>** – zobrazuje se tučně

**<sub>** dolní index

**<sup>** horní index

**<abbr>** zkratka v textu

Př. `<p>Vítejte na stránkách <abbr title="Střední škola spojů a informatiky">SŠSI</abbr>!</p>`

## **5. Pozadí stránky**

### **Barevné pozadí**

Nastavení barvy pozadí zajistí CSS vlastnost **background-color**.

Př. `body {background-color:red;} – nastaví barvu pozadí stránky na červenou`

Barvu lze zadat dvojím způsobem

a) Názvem barvy – nejčastěji používané barvy mají pro jednoduchost své jméno.

16 základních barev:



white	bílá	lime	citronově zelená
yellow	žlutá	olive	olivová
red	červená	purple	purpurová
green	zelená	fuchsia	anilínová červená
blue	modrá	silver	stříbrná
navy	tmavá modrá	gray	šedivá
teal	tmavá modrozelená	maroon	kaštanová
aqua	jasně modrozelená	black	černá

b) **RGB** definicí – hodnotu vlastnosti **background-color** zapíšeme jako podíl jednotlivých základních barev (R-červená, G-zelená, B-modrá). K dispozici máme tři možnosti zápisu:

- Desítkový zápis – mohutnost každé barvy zapíšeme hodnotou 0 – 255.

Př. body {background-color: rgb(255, 0, 0); }

- Procentový zápis – místo hodnot 0- 255 píšeme procenta

Př. body {background-color: rgb(100%, 0, 0); }

- Šestnáctkový zápis – podíl barvy je opět hodnota 0 – 255, ale v šestnáctkové soustavě, tj. 00 – ff. Zápis je proveden jako šestimístné číslo podle vzorce *rrggbb*. Před číslo se píše znak #.

Př. body {background-color: #ff0000; }

## Obrázek jako pozadí stránky

K umístění obrázku na pozadí stránky se používá vlastnost **background-image**.

Př. body {background-image: url(pozadi.jpg); } – soubor *pozadi.jpg* je obrázek umístěný ve stejné složce jako definice stylu.

Ve stylu se obvykle ponechává i barva pozadí pro případ, že by se obrázek nenačetl, případně barevné pozadí může pod obrázkem prosvítat.

Pokud nechceme obrázek na pozadí opakovat, nastavíme vlastnost

**background-repeat:no-repeat;** a obrázek se zobrazí jen jednou v levém horním rohu.

Pro pozici obrázku použijeme vlastnost **background-position:**

**left top** - levý horní roh

**center top** – horní střed

**right top** – pravý horní roh

**left center** – levý střed

**center center** – střed

**right center** – pravý střed

**left bottom** – levý dolní roh

**center bottom** – levý střed

**right bottom** – pravý dolní roh

**x% y%** - souřadnice v %

**xpos ypos** – souřadnice v pixelech

Opakovat obrázek můžeme jen v řádku nebo sloupci, pak zvolíme **background-repeat:repeat-x;** nebo **background-repeat:repeat-y;**

## Relativní a absolutní adresování

Pro uvedení adresy obrázku platí stejná pravidla jako pro adresu externího dokumentu se stylem a stejná pravidla budou platit i pro ostatní adresy, např. u odkazů.

Adresu lze zadat dvojím způsobem:

- **Absolutní adresa** – úplné zadání cesty k cílovému souboru

Př. <http://www.firma.cz/kontakt.html>

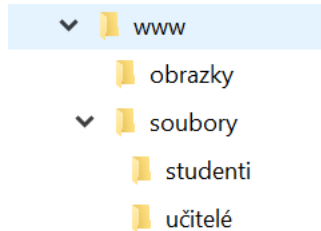
nebo <C:/www/soubory/obrazky/pozadi.jpg>.

První způsob se používá při odkazu na stránky mimo naši prezentaci, druhý se nepoužívá, protože při jakémkoliv změně umístění naší webové prezentace se adresy změní a odkaz by nefungoval.

- **Relativní adresa** – jako výchozí bod cesty se bere aktuální adresa stránky, kterou vytváříme. Je-li soubor, na který se odkazujeme ve stejné složce, pak uvádíme pouze název včetně přípony. Pokud je

tento soubor v podsložce, pak píšeme cestu po jednotlivých podsložkách a názvy podsložek oddělujeme znakem / (lomítko). Pokud je soubor, na který se odkazujeme ve stromové struktuře složek výš, než se právě nacházíme, pak za každou úroveň směrem nahoru píšeme .. (dvě tečky).

Př. Ze souboru **index.html** na soubor **styl.css** se odkazujeme adresou *styl.css*.

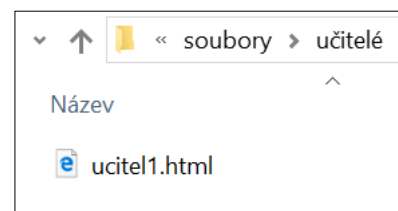
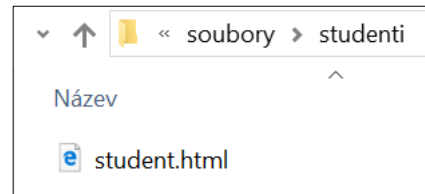
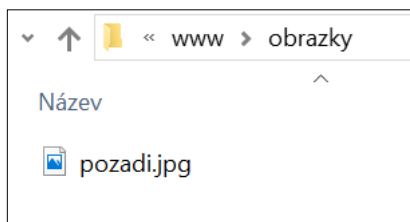
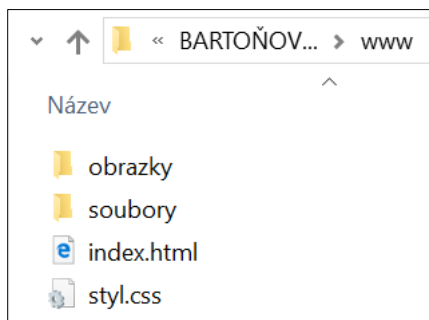


Ze souboru **index.html** na soubor **ucitel1.html** se odkazujeme adresou *soubory/učitelé/ucitel1.html*

Ze souboru **student1.html** na soubor **styl.css** se odkazujeme adresou *../../styl.css*.

Ze souboru **student1.html** na soubor **pozadi.jpg** se odkazujeme adresou *../obrazky/pozadi.jpg*

Ze souboru **student1.html** na soubor **ucitel1.html** se odkazujeme adresou *../učitelé/ucitel1.html*.



## 6. Vkládání obrázků

Pro vkládání obrázků do webových stránek se používají formáty **.jpg**, **.png** případně **.gif**

Obrázek vložíme užitím značky **<img>**. Povinný je atribut **src**, jehož hodnotou je adresa obrázku. Dále je vhodné použít atribut **alt**, jehož hodnotou je tzv. alternativní text, tj. text, který se zobrazí v případě nenačtení obrázku, a atribut **title**, což je bublinová nápověda. Použití atributu **alt** je vhodné kvůli vyhledávačům obrázků nebo hlasovým čtečkám. Atribut **title** jsme již zaznamenali u značky **<abbr>**. Použití ho lze u libovolné značky. Vhodný je například i u odkazů nebo formulářových polí.

Př. ``

Adresu vždy uvádíme relativně viz. předchozí kapitola.

### Svislé zarovnání obrázku vzhledem k textu

Pro zarovnání obrázku vzhledem k textu se dříve používal atribut **align**, který je z pohledu HTML5 zastaralý.

Pro svislé zarovnání obrázku vzhledem k okolnímu obsahu musíme vytvořit třídy

**.nahoru {vertical-align:top;}** text zarovnán k hornímu okraji obrázku

**.nastred {vertical-align:middle;}** text zarovnán ke středu obrázku

**.dolu {vertical-align:bottom;}** text zarovnán k dolnímu okraji obrázku

Př. ``

### Obtékání obrázku

Pro obtékání obrázku textem se dříve také používal zastaralý atribut **align**.

Musíme vytvořit třídy

**.vlevo {float:left;}** obrázek vlevo a text vpravo

**.vpravo {float:right;}** obrázek vpravo a text vlevo

Př. ``





Pokud odkazu odebereme jeho barevné rozlišení a podtržení, měli bychom ho nějak jinak odlišit od ostatního textu, třeba barvou, velikostí nebo řezem písma. Podtržení bychom odkazům v běžném textu měli nechat, aby bylo jasné, že jde o odkazy. V nabídkách je naopak možné podtržení odebírat. Je jasné, že jde o odkaz, když je to v nabídce. Také můžeme zajistit změnu textu odkazu, jestliže na něj najedeme myší.

Tohoto efektu dosáhneme pomocí pseudotřídy **a:hover**

```
Př. a {color:navy;
      font-size:12pt;
      text-decoration:none}
a:hover {color:navy;
         font-size:14pt;
         text-decoration:underline}
```

při najetí myší se text zvětší a podtrhne

Další pseudotřídy, které souvisí s odkazy jsou **a:link**, **a:activ**, **a:visited**, které se vztahují k nenavštívenému odkazu, k právě volanému odkazu a k navštívenému odkazu.

K textu odkazu můžeme připojit bublinovou nápovědu pomocí atributu **title** (jako u obrázku).

Odkaz můžeme otevírat do nového okna pomocí atributu **target** s hodnotou **\_blank**.

```
Př. <a href="..." target="_blank">...</a>
```

## 8. Orámování objektů

### Barva rámečku

**border-color:** definice barvy;

Pro definici barvy používáme stejná pravidla jako u pozadí stránky

### Šířka orámování

**border-width:** hodnota;

Hodnota se nejčastěji uvádí v pixelech (px). Mezi číslo a jednotku se nepíše mezera. Pokud zapíšeme bez jednotky, pak to většina prohlížečů interpretuje jako pixely, ale některé prohlížeče by mohly hodnotu bez udané jednotky ignorovat.

Kromě pixelů se užívají i jiné jednotky:

- mm (milimetr)
- cm (centimetr)
- in (palec - inch), 1 in = 2,6 cm
- pt (typografický bod), 72 pt = 1 in

Místo přesné hodnoty lze udávat i předdefinované hodnoty:

- **thin** tenké orámování
- **medium** středně silné orámování
- **thick** silné orámování

### Styl orámování

**border-style:** hodnota;

Hodnota	Popis	Hodnota	Popis
solid	nepřerušovaná čára	groove	drážka (příkop)
dotted	tečkovaná čára	ridge	val
dashed	přerušovaná čára	inset	údolí (obsah zanořený)
double	dvojitá čára	outset	návrsí (obsah vyvýšený)
		none	bez orámování

### Všechny vlastnosti najednou

**border:** šířka styl barva;

```
Př. border:3px solid red;
```

## Definice orámování jednotlivých stran objektu

	Definice barvy	Definice šířky	Definice stylu
Horní okraj	<code>border-top-color</code>	<code>border-top-width</code>	<code>border-top-style</code>
Dolní okraj	<code>border-bottom-color</code>	<code>border-bottom-width</code>	<code>border-bottom-style</code>
Pravý okraj	<code>border-right-color</code>	<code>border-right-width</code>	<code>border-right-style</code>
Levý okraj	<code>border-left-color</code>	<code>border-left-width</code>	<code>border-left-style</code>

Vlastnosti jednotlivých stran lze také zadat najednou, a to od horního rámečku ve směru hodinových ručiček (horní, pravý, dolní, levý)

Př. `border-color:red green blue black;` horní červený, pravý zelený, dolní modrý a horní černý

## 9. Seznamy a výčty

### Seznamy s odrážkami

Seznam je vymezen značkami `<ul>...</ul>`. Jednotlivé položky jsou mezi značkami `<li>...</li>`.

Př.

Př. Seznam

Ovoce:

- jablka
- banány
- pomeranče

Zdrojový kód:

Ovoce:

```
<ul>
  <li>jablka</li>
  <li>banány</li>
  <li>pomeranče</li>
</ul>
```

Typ odrážky ovlivníme CSS vlastností **list-style-type**, kterou přiřadíme značce `<ul>`.

Hodnota	Zobrazovaný znak
<b>none</b>	žádný znak
<b>disc</b>	plné kolečko, případně kosočtverec
<b>circle</b>	nevyplněné kolečko, případně kosočtverec
<b>square</b>	plný čtvereček

Vlastní obrázek místo odrážky lze definovat vlastností **list-style-image** jejíž hodnotou je adresa obrázku ve tvaru: **url(adresa)**

Vlastností **list-style-position** určíme, zda bude odrážka předsazena před textem (hodnota *outside*) nebo vnořená do textu (*inside*). Výchozí hodnota je *outside*.

Viz příklad *seznamy.html*

### Výčty – číslované seznamy

Výčet vymezují značky `<ol>...</ol>`. Odrážky se definují stejně jako u seznamu.

Hodnoty vlastnosti **list-style-type**:

Hodnota	Zobrazovaný znak
<b>none</b>	žádný znak
<b>decimal</b>	desítkové číslování (1. 2. 3. ...)
<b>lower-alpha</b>	malá písmena (a. b. c. ...)
<b>upper-alpha</b>	velká písmena (A. B. C. ...)
<b>lower-roman</b>	malé římské číslice (I. II. III. ...)
<b>upper-roman</b>	velké římské číslice (I. II. III. ...)
<b>lower-greek</b>	malá písmena řecké abecedy

Kromě těchto hodnot jsou definovány i další, jejichž zobrazení závisí na podpoře jednotlivých prohlížečů. Pokud prohlížeč hodnotu nepodporuje, použije výchozí hodnotu *decimal*.

U číslovaného seznamu máme možnost začít i od jiné hodnoty než od první (1., a., I., atd.). Buď u značky `<ol>` použijeme atribut **start**, jehož hodnotou je číslo, od kterého budeme začínat, nebo u konkrétní položky `<li>` použijeme atribut **value**.

Viz příklad *seznamy.html*

## 10. Tabulky

Tabulky ve webových prezentacích využíváme dvojím způsobem. Buď přímo jako tabulky, většinou ohraničené, nebo jako neviditelné tabulky, které slouží k rozvržení prvků na stránce.

`<table> ... </table>` vloží tabulku  
`<tr> ... </tr>` vloží řádek  
`<td>...</td>` vloží buňku  
`<th>...</th>` vloží buňku se zvýrazněným obsahem (záhlaví)

Obsahem buňky je většinou text, ale může obsahovat i obrázky, seznamy atd., dokonce i další tabulku.

Př. `<table>`

```

<tr>
  <th>Pololetí</th>
  <th>Tržba</th>
  <th>Náklady</th>
</tr><tr>
  <th>1.</th>
  <td>250 000,-</td>
  <td>180 000,-</td>
</tr><tr>
  <th>2.</th>
  <td>360 000,-</td>
  <td>250 000,-</td>
</tr>
</table>

```

Pololetí	Tržba	Náklady
1.	250 000,-	180 000,-
2.	360 000,-	250 000,-

Takto nadefinovaná tabulka ovšem nebude mít rámečky, šířka a výška buněk bude podle obsahu.

### Ohraničení, pozadí

Při ohraničení tabulky nebo jednotlivých buněk využijeme stejné vlastnosti orámování objektů viz. předchozí kapitola.

Orámování buněk musíme definovat v každé buňce zvlášť. Abychom nemuseli při změně rámečku měnit každou buňku, je vhodné definovat třídu (např. `.ram` nebo `.bunka`, `.tabulka`).

Jednotlivé buňky tabulky jsou od sebe odděleny malou mezerou. Vlastnost **border-collapse** určuje zda se tyto mezery zobrazí. Jako výchozí hodnota je nastaveno **separate** (zobrazuje mezery), velikost mezer ovlivníme vlastností **border-spacing** (`border-spacing:10px`). Aby se mezery nezobrazovaly, musíme nastavit hodnotu **collapse**. (`border-collapse:collapse;`)

Vlastnost **padding:hodnota**; uvádí šířku vnitřního rámečku.

Př. `td {padding:0px;}`

Pro nastavení pozadí tabulky nebo buňky, platí stejné vlastnosti jako u pozadí stránky.

## Zarovnání obsahu

Pro horizontální zarovnání použijeme vlastnost **text-align** (hodnoty: *left*, *right*, *center*).

Pro vertikální zarovnání použijeme vlastnost **vertical-align** (hodnoty: *top*, *middle*, *bottom*).

## Šířka a výška

Šířka tabulky nebo buňky se tak jako u obrázku nastavuje atributem **width**. Hodnota se udává buď v pixelech nebo v procentech. Zadáme-li šířku buňky, pak se změní šířka celého sloupce.

Výška tabulky nebo buňky se nastavuje atributem **height**.

## Slučování buněk

Ke sloučení buněk pod sebou (v rámci sloupce) slouží atribut **rowspan**, pro sloučení buněk vedle sebe (v rámci řádku) slouží **colspan**. Hodnota obou atributů je počet buněk jež se sloučí.

A	B	
C	D	E
	F	G

```
<table>
<tr>
  <td>A</td>
  <td colspan="2">B</td>
</tr><tr>
  <td rowspan="2">C</td>
  <td>D</td>
  <td>E</td>
</tr><tr>
  <td>F</td>
  <td>G</td>
</tr>
</table>
```

## Prázdné buňky

Buňka `<td></td>` nebo `<td> </td>` je prázdná buňka. Některé prohlížeče ji zobrazí jako prázdnou ohraničenou buňku, ale některé nezobrazí a pak se může dojít k estetické deformaci tabulky. Zabráníme tomu tím, že do prázdných buněk budeme psát tzv. pevnou (tvrdou) mezeru - **&nbsp;**;

Př. `<td>&nbsp;</td>`

## Zalamování řádků

Text v buňkách se automaticky zalamuje. Pokud chceme zalomení zabránit (zobrazit celý obsah v jednom řádku), použijeme CSS vlastnost **white-space:nowrap;**

## 11. Formuláře

Formuláře se na stránkách uplatní tam, kde potřebujeme získat nějaké informace od uživatele. Jazyk HTML umí formuláře vytvářet, ale neumí je zpracovat.

Nejprve se budeme zabývat vytvoření formuláře pomocí HTML značek a až v dalších kapitolách se naučíme získaná data zpracovat pomocí PHP.

Formulář vkládáme do stránky pomocí značek `<form>...</form>`.

Na jedné stránce může být víc formulářů, ale nesmí se křížit ani vnořovat.

Značka `<form>` má dva povinné parametry – **action** a **method**.

Hodnotou parametru **action** je adresa stránky, kam mají být data po odeslání formuláře předána ke zpracování. Často data zpracovává stejná stránka, pak zadáme jen název dané stránky. Pokud parametr **action** chybí, pak si s tím většina prohlížečů poradí tak, že odešle data do stejné stránky, ale chybějící parametr odporuje specifikaci a stránka není validní.

Parametr **method** určuje, jakým způsobem budou data z formuláře přenesena.

Hodnotou **get** se data přenáší pomocí adresy – názvy polí a hodnoty, které uživatel vyplní, se stanou součástí webové adresy. Pokud jsou data důležitá pro zobrazení konkrétní stránky s konkrétním obsahem, pak je vhodné použít právě metodu **get**. Cílovou stránku můžeme uložit k oblíbeným a tak ji vyvolat znovu i s daným obsahem.

Metodou **post** se data přenáší v těle dokumentu a nejsou tedy vidět v adrese cílové stránky. Metodu **post** použijeme, když se jedná o citlivá data, jako např. přihlašovací jméno a heslo.

## 12. Formulářové prvky

### Textové pole

Textové pole slouží k zadávání krátkých textů. Textové pole vložíme značkou `<input type="text">`.

Značkou `<input...>` lze vkládat i jiné formulářové prvky, proto je v ní povinný parametr **type**, kterým určíme, o jaký prvek se jedná.

Dalším povinným parametrem je u všech formulářových prvků parametr **name**, který určuje jméno tohoto prvku a je tedy jedinečný (tj. každý prvek se v daném formuláři musí jmenovat jinak). Názvu prvku bude po odeslání formuláře přiřazena hodnota zadaná uživatelem. Např. pokud textovému poli s názvem *jméno* zadáme hodnotu *Alois*, vznikne dvojice *jméno=Alois*.

Nepovinné parametry:

Parametr **size** udává šířku textového pole ve znacích.

Parametr **maxlength** udává maximální počet znaků, které lze zadat. Je vhodné omezit maximální délku textového pole, aby nedošlo k zahlcení stránky odesláním nesmyslně dlouhých textových řetězců.

Parametrem **value** můžeme zadat výchozí hodnotu, která se do formuláře vloží.

Parametrem **placeholder** zadáme výchozí text, který není hodnotou.

CSS vlastnostmi můžeme ovlivnit vzhled textového pole například tak, že definujeme třídu, které nadefinujeme např. *background-color*, *background-image*, *color*, *font-size*, *border-style*, *border-color* a další vlastnosti pro formátování textu nebo pro orámování pole.

### Pole pro vložení hesla

Pole pro vložení hesla je textové pole, do kterého se vypisují kolečka či hvězdičky (podle prohlížeče). Vkládá se `<input type="password">`. Všechny ostatní parametry a vlastnosti má jako textové pole.

### Textová oblast

Textovou oblast použijeme tam, kde potřebujeme textové pole o více řádcích. Vkládáme pomocí značek `<textarea>...</textarea>`. Povinným parametrem je, tak jako u všech prvků, **name**. Další povinné parametry jsou **cols** a **rows**. Parametr **cols** určuje šířku textové oblasti ve znacích a **rows** udává počet řádků. Maximální délku zadávaného textu nelze stanovit parametrem **maxlength**. Mezi značky `<textarea>...</textarea>` můžeme vložit výchozí text.

### Zaškrťovací pole

Zaškrťovací pole má dva stavy *zaškrtnuto* a *nezaškrtnuto*. Užívá se např. pro potvrzení souhlasu nebo k výběru z více možností (je možno zaškrtnout více najednou). Není vhodné tam, kde máme vybrat z více možností právě jednu (např. pohlaví, nejvyšší dosažené vzdělání, atd.)

Vkládáme `<input type="checkbox">` Povinné parametry jsou **name** a **value**. Parametr **value** určuje hodnotu pole. Hodnota se přiřadí jménu je-li pole při odeslání zaškrtnuto.

Př. `<input type="checkbox" name="souhlas" value="ano">`

Pokud je zaškrtnuto, vznikne dvojice *souhlas=ano*.

Chceme-li, aby výchozí stav pole byl *zaškrtnuto*, pak použijeme parametr **checked** (bez hodnoty).



## Přepínač

Přepínač je vhodné použít, pokud máme vybrat z více možností právě jednu. Zatímco zaškrťovací pole se používá samostatně, přepínače musí být ve skupině nejméně dvou (podle počtu volených variant). Skupinu přepínačů vytvoříme tím, že jim dáme stejnou hodnotu parametru **name**. Výběrem jednoho přepínače ze skupiny se předtím označený přepínač odznačí.

Vkládáme `<input type="radio">`. Povinné parametry jsou **name** (stejně pro všechny ve skupině) a **value** (jako u checkboxu).

Parametrem **checked** nastavíme jeden ze skupiny nastavit jako výchozí.

## Výběr ze seznamu

Tento prvek použijeme k výběru z více možností, tam kde se nehodí použití přepínačů ne zaškrtn. polí (např. když je variant hodně).

Vkládá se `<select>...</select>`. Jednotlivé položky určujeme značkami `<option value="...">...</option>`. Povinný parametr **value** určuje hodnotu při výběru dané položky seznamu. Značka `select` má povinný parametr **name**.

Nepovinné parametry pro **select**:

Parametr **size** určuje počet zobrazovaných řádků seznamu (není-li uveden, bere se jako výchozí hodnota 1).

Parametr **multiple** povoluje zadání více hodnot. Více hodnot vybíráme za použití kláves *ctrl* (pro nespojitou oblast) a *shift* (pro souvislou oblast).

## Skryté pole

Skryté pole umožní odeslat s formulářem data, o kterých uživatel nemusí vědět (např. IP adresa, operační systém, rozlišení, prohlížeč atd.). Používá se také při posílání dříve zadaných hodnot.

Vkládáme `<input type="hidden">`. Povinné parametry jsou **name** a **value**.

## Odesílací tlačítko

Vkládáme `<input type="submit">`. Vzhled tlačítka nelze ovlivnit (záleží na prohlížeči). Pokud chceme ovlivnit text na tlačítku použijeme tento text jako hodnotu parametru **value**. Nemusí mít **name** (pokud to situace nevyžaduje).

## Tlačítko pro vymazání formuláře

Toto tlačítko vynuluje všechny vyplněné hodnoty (nastaví přednastavené).

Vkládáme `<input type="reset">`.

Vlastnosti má jako odesílací – neovlivníme vzhled, **value** nastaví text, nemá **name**.

## Grafické odesílací tlačítko

`<input type="image">`. Povinným parametrem **src** vložíme adresu obrázku (jako u `<img>`)

## Uzamčení formuláře

Pokud chceme u textového pole nebo u textové oblasti zamezit přepsání (vypadá stejně, ale nelze do něj psát), přidáme danému prvku parametr **readonly**.

Chceme-li nějaký prvek zcela deaktivovat, přidáme mu parametr **disabled**.

## Další formulářové prvky a další atributy

Výše uvedené prvky a jejich vlastnosti jsou ty nejčastěji používané. Pro běžné použití stačí. HTML5 nabízí celou řadu dalších možností. Jejich nastudování nechávám „fajnšmekrům“ k samostudiu. Najdete např. na

[www.jakpsatweb.cz/html/formulare.html#input](http://www.jakpsatweb.cz/html/formulare.html#input) nebo

[www.itnetwork.cz/html-css/html-manual/html-5-abecedni-rejstrik-tagu](http://www.itnetwork.cz/html-css/html-manual/html-5-abecedni-rejstrik-tagu) - část formuláře

## 13. Pozicování

Při rozvržení jednotlivých částí stránky lze použít tři varianty řešení:

### - Rámce (frame)

Používání rámců je zastaralé. Šlo o „slepou uličku“ při vývoji jazyka HTML – **NEPOUŽÍVÁ SE**

### - Skryté tabulky

Vhodný a jednoduchý způsob pro toho, kdo neovládá CSS. Nevýhodou je, že se musí definovat v každé stránce zvlášť a při doplnění nebo změnách je nutné měnit kód všech stránek. Používá se spíše při rozvržení prvků v hlavní části konkrétní stránky.

viz. př. **pozicování - tabulka**

### - Užití CSS

Vhodný způsob pro pozicování všech stránek webové prezentace najednou. Je možné použít i třídy, ale vhodnější je vymezit jednotlivé části stránky pomocí identifikátorů.

Identifikátory se definují v CSS stylu ve tvaru: `#název identifikátoru { popis CSS vlastností }`

Pro definování obsahu jednotlivých částí je použito značky: `<div id=" název identifikátoru ">...</div>`

Identifikátor je prvek, který se na dané stránce vyskytuje pouze jednou, jako například právě hlavička, nabídka nebo tělo stránky.

viz. př. **pozicování**

## Vycentrování stránky

Problémem při formátování jednotlivých prvků na stránce může být její šířka. Ta může být různá vzhledem k šířce okna prohlížeče a rozlišení. Řešením může být nastavení pevné šířky obsahu a vycentrování v okně prohlížeče.

U stránek pozicovaných tabulkou to není problém, stačí celou tabulku zarovnat na střed stránky a dát celé tabulce nebo jednotlivým buňkám pevnou šířku.

viz. př. **pozicování - tabulka**

U stránek pozicovaných pomocí CSS je několik možností. Uvedu dva konkrétní příklady.

viz. př. **pozicování – na střed 1**

viz. př. **pozicování – na střed 2**

viz. př. **pozicování – na střed 3** (stejně jako varianta 2 + zvláštní řešení pro starší verze Internet Exploreru)

## 14. Úvod do PHP

PHP je serverový skriptovací jazyk navržený pro potřeby webových stránek.

Vše, co PHP provádí, neprobíhá na straně klienta jako například u JavaScriptu, ale interpretuje se na straně serveru a generuje HTML (či jiný) výstup, který vidí uživatel.

Zdrojový kód PHP skriptu se vkládá přímo do zdrojového kódu HTML např.

```
<?php
```

```
...
```

```
?>
```

Syntaxe je velmi podobná syntaxi programovacího jazyka C. Každý příkaz musí být oddělen středníkem.

Příkaz **echo** slouží k vypisování textu. V příkazu **echo** můžeme zapisovat i příkazy HTML.

Komentář v PHP skriptech se píše buď jednořádkový

Př. `//text komentáře`

nebo víceřádkový  
Př. */\*text komentáře  
i ve více řádcích\*/*

U příkazů PHP nerozlišuje mezi malými a velkými písmeny ovšem u názvů proměnných ano, proto je lepší vyvarovat se experimentů a používat pouze malá písmena.

Pokud je ve stránce použit kód PHP, musí mít soubor příponu **.php**

K zobrazení stránky včetně PHP skriptu musíme mít nainstalovaný překladač jazyka PHP a webserver nebo musíme publikovat na Internetu.

## 15. Získání dat z formuláře

Značka **<form>** má povinné parametry **action** (určuje stránku, kam jsou data odeslána) a **method** (určuje způsob přenesení dat - metody **get** a **post**). Každý formulářový prvek má parametrem **name** zadáno jedinečné jméno, které se v cílovém skriptu stane proměnnou s hodnotou zadanou ve formuláři.

Všechny proměnné předané prostřednictvím formuláře jsou uloženy v poli:

- **\$\_POST**, pokud byla použita metoda **post**
- **\$\_GET**, pokud byla použita metoda **get**

Př. Textové pole ve formuláři má `name="jmeno"` a `method="post"`.  
`$jmeno=$_POST[jmeno];`

Pro návrat k formuláři a zadání jiných hodnot můžeme použít např. formulář pouze s odesílacím tlačítkem nasměrovaný na původní stránku s formulářem.

Př. `<form action="..." method="post">`  
`<input type="submit" value="zadat nové hodnoty">`  
`</form>`

Př. formulář

```
<form action="formular.php" method="get">
  <p>
    Jméno: <input type="text" name="jmeno"><br>
    Věk: <input type="text" name="vek"><br>
    <input type="submit" value="Pošli!" name="odeslano">
  </p>
</form>
```

zpracování formuláře na stejné stránce

```
<?php
if (isset($_GET["odeslano"]))
echo "<p>Jmenujete se ".$_GET["jmeno"]." a je Vám ".$_GET["vek"]."let.</p>";
?>
```

## 16. Include

Příkaz **include** umožňuje integrovat obsah jedné stránky do jiné.

Vhodné použití je např. u nabídky, kterou chceme opakovat na více stránkách. Pokud bychom tuto situaci řešili kopírováním kódu do každé stránky zvlášť, tak budeme mít problém s úpravou takové nabídky, protože budeme muset upravovat opět každou stránku zvlášť. Pokud ovšem vytvoříme samostatnou stránku s nabídkou, kterou vnoříme do všech ostatních stránek, pak stačí upravit vždy pouze tuto stránku s nabídkou.

Př. Vytvoříme stránku *menu.php* do které napíšeme zdrojový kód nabídky.

Do každé stránky, kam chceme menu vložit, napíšeme:

```
<?php
include "menu.php";
```

?>

Ve vkládaném souboru jde pouze o text (zdrojový kód), proto může být přípona souboru i jiná, např. *.html* nebo *.txt*.

## 17. Menu webové prezentace

Pro vytvoření menu webové prezentace a orientaci na jednotlivých stránkách můžeme zvolit dvě různé cesty.

1. Pro každou stránku budeme mít vlastní soubor s kompletním kódem a menu z externího souboru budeme vkládat do všech stránek stejné.

Př. kód stránky

```
<html>
  <head>
    <meta http-equiv="content-type" content="text/html; charset=UTF-8">
    <title>Stránky IT2</title>
    <link type="text/css" href="styl.css" rel="stylesheet">
  </head>
  <body>
    <div id="stred">
      <div id="stranka">
... obsah hlavní části konkrétní stránky
      </div>
      <div id="menu">
        <?php
          include "menu.php";
        ?>
      </div>
      <div id="localmenu">
        <?php
          include "localmenu.php";
        ?>
      </div>
      <div id="logo">
      </div>
    </div>
  </body>
</html>
```

definování identifikátorů *logo*, *menu*, *localmenu* je obsaženo ve stylu *styl.css*.

Kód menu

```
<a href="index.php" class="localmenu">úvod</a>
<br>
<br>
<a href="formular.php" class="localmenu">formulář</a>
<br>
<br>
<a href="semafor.php" class="localmenu">semafor</a>
<br>
<br>
<a href="obrazce.php" class="localmenu">obrazce</a>
```

Vše viz. Příklad *web – vzor1*

2. Kompletní kód má pouze stránka *index*. Soubory ostatních stránek obsahují pouze obsah hlavní části stránky, kam se vkládají pomocí příkazu *include* a řídicí konstrukce *switch*.

Adresy v menu jsou vytvořené tak, aby simulovaly odeslání formuláře s prvkem *str*. Podle jeho hodnoty se pak vkládají jednotlivé externí soubory.

Vše viz. Příklad *web – vzor2*

Př. kód stránky

```
<html>
<head>
<meta http-equiv="content-type" content="text/html; charset=UTF-8">
<title>Stránky IT2</title>
<link type="text/css" href="styl.css" rel="stylesheet">
</head>
<body>
<div id="stred">
<div id="stranka"> <p>
<?php
  if (isset($_GET["str"]))
  switch ($_GET["str"]){
  case "formular":include "formular.php";break;
  case "semafor":include "semafor.php";break;
  case "obrazce":include "obrazce.php";break;
  }
  else include "uvod.php";
?>
</div>
<div id="menu">
<a href="http://www.sous.cz" class="menu">spoje</a>
</div>
<div id="localmenu">
<br>
<br>
<a href="index.php" class="localmenu">úvod</a>
<br>
<br>
<a href="index.php?str=formular" class="localmenu">formulář</a>
<br>
<br>
<a href="index.php?str=semafor" class="localmenu">semafor</a>
<br>
<br>
<a href="index.php?str=obrazce" class="localmenu">obrazce</a>
</div>
<div id="logo">
</div>
</div>
</body>
</html>
```