

Ďalšie vzdelávanie učiteľov základných škôl a stredných škôl v predmete informatika

Internet: princípy a tvorba webu 3

Predmet: Internet: princípy a tvorba webu

Línia: Vlastný odborový kontext informatiky a informatickej výchovy



Internet: princípy a tvorba webu 3

Identifikácia modulu

Aktivita projektu:

1.2 Vzdelávanie nekvalifikovaných učiteľov informatiky na 2. stupni ZŠ a na SŠ

Línia aktivity:

Vlastný odborový kontext informatiky a informatickej výchovy

Predmet:

Internet: princípy a tvorba webu

Zaradenie modulu



Modul je zameraný na tvorbu webových stránok pomocou jazyka XHTML a definovanie vzhľadu webových stránok pomocou kaskádových štýlov.

Abstrakt modulu

Rovnako ako modul Internet: princípy a tvorba webu 2, aj tento modul sa zaoberá tvorbou webových stránok, avšak už nie WYSIWYG spôsobom, ale editovaním zdrojového kódu stránok v jazyku XHTML a definovaním vzhľadu stránok pomocou kaskádových štýlov. Účastníci si počas modulu budú tvoriť jednoduché profesijné webové sídlo (lokalitu) s niekoľkými podstránkami.

V prvej časti sa účastníci oboznámia s editovaním webových stránok v jazyku XHTML. V tejto časti sa zameriame na vytvorenie logickej a obsahovej štruktúry stránky (rozdelenie do blokov a odsekov), jednoduché formátovanie textu, vkladanie zoznamov, obrázkov a odkazov. Vložíme tiež jednoduchú tabuľku. Vzhľad stránky v tejto časti nebude podstatný. V druhej časti sa zameriame na úpravu vzhľadu stránky pomocou kaskádových štýlov. Oboznámime sa so základnými vlastnosťami jednotlivých elementov (farby, rôzne nastavenia písma, zarovnávanie, orámovanie, priestory okolo elementov, ...), naučíme sa vytvárať vlastné triedy a pseudotriedy a používať ich. Tiež si ukážeme, ako vytvárať kaskádové štýly pre iné zariadenia.

Počas tvorby webového sídla sa budeme držať zásad tvorby stránok, s ktorými sa účastníci oboznámili v predchádzajúcom module, prípadne doplníme ďalšie.

Garant predmetu:

PaedDr. Roman Hrušecký,
PhD.
KZVI FMFI UK, Bratislava
hrusecky@fmph.uniba.sk

Autori:

PaedDr. Roman Hrušecký,
PhD.
KZVI FMFI UK, Bratislava
PaedDr. Daniela Bezáková,
PhD.
KZVI FMFI UK, Bratislava



Obsah

Internet: princípy a tvorba webu 3	1
Identifikácia modulu	1
Zaradenie modulu	1
Abstrakt modulu	1
Obsah	2
Úvod	3
Cieľ modulu	3
Vstupné vedomosti	3
Požadované prerekvizity	3
Predpokladané vstupné vedomosti, skúsenosti a zručnosti	3
Tvorba webových stránok	4
Jazyk XHTML	4
Jazyk XHTML, štandardy	4
Štruktúra dokumentu	5
Kódovanie stránky	6
Kontrola zdrojového kódu stránky	7
Štruktúrovanie obsahu stránky	7
Formátovanie textu	9
Vnáranie elementov	11
Zoznamy (s odrážkami, číslované)	12
Obrázky	13
Atribúty elementov	14
Odkazy	14
Relatívne a absolútne adresovanie	15
Špeciálne znaky	16
Tabuľky	17
Kaskádové štýly	20
Úvod do kaskádových štýlov	20
Definovanie a vkladanie štýlov	20
Pripojenie kaskádových štýlov pomocou elementu <link/>	21
Element ako selektor	22
Vlastnosti textu	23
Nastavenie pozadia	25
Zoznamy	26
Pomenovanie elementov	27
Selektor ID	28
Selektor triedy	28
Pseudotriedy	29
Selektor nasledovníka	30
Okraje	31
Orámovanie	31
Vnútorne okraje	32
Vonkajšie okraje	33
Tabuľky	33
Rozmery	35
Štýly pre iné zariadenia	36
Záver	37
Čo sme sa naučili v tomto module	39
Preverenie výstupných vedomostí	39
Literatúra a použité zdroje	39

Úvod

V predchádzajúcom module Internet: princípy a tvorba webu 2 sme tvorili webové stránky v dvoch WYSIWYG prostrediach. Prostredie Google webové stránky bolo síce jednoduché, mnohé veci za nás robilo automaticky, ale v istom zmysle nás aj obmedzovalo. Prostredie NVU nám už poskytovalo väčšiu „slobodu“ pri tvorbe stránok (napr. viac nástrojov na prácu s tabuľkami), ale aj tu sme obmedzení možnosťami prostredia.

Najväčšiu voľnosť pri tvorbe webových stránok dosiahneme, ak budeme písať zdrojový kód stránky. Je to samozrejme náročnejšie, pretože musíme poznať jazyk, v ktorom sa zdrojový kód webových stránok píše a správne ho používať.

V prvej časti tohto modulu si predstavíme jazyk XHTML a jeho základné elementy, ktoré nám pomôžu vytvoriť webovú stránku. Na stránky postupne vložíme texty štruktúrované do blokov, obrázky, zoznamy, odkazy, tabuľky a špeciálne znaky.

Aktuálnym trendom tvorby webových stránok je vytváranie dizajnu len pomocou kaskádových štýlov (CSS - Cascading Style Sheets), ktoré si predstavíme v druhej časti modulu. Oboznámime sa so základnými vlastnosťami jednotlivých elementov (farby, rôzne nastavenia písma, zarovnávanie, orámovanie, priestory okolo elementov, ...), naučíme sa vytvárať vlastné triedy a pseudotriedy a používať ich.

Cieľ modulu

Cieľom modulu je naučiť účastníkov rozumieť zdrojovému kódu webových stránok, a tiež vytvárať vlastné webové stránky písaním zdrojového kódu v jazyku XHTML, oboznámiť ich so základnými elementmi, ktoré tvoria webovú stránku.

Modul si tiež kladie za cieľ naučiť účastníkov definovať vzhľad webových stránok pomocou kaskádových štýlov.

Vstupné vedomosti

Požadované prerekvizity

Pred absolvovaním modulu musí mať účastník absolvované moduly:

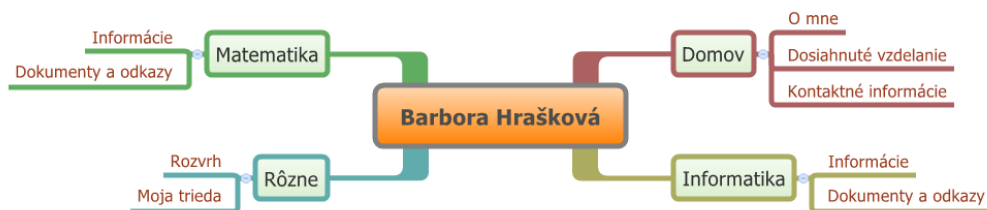
- Základná digitálna gramotnosť (2DG1),
- Multimédiá (2DG2),
- Základy hardvérového a softvérového vybavenia počítača (2DG3),
- Internet: princípy a tvorba webu 1 (2Inter1),
- Internet: princípy a tvorba webu 2 (2Inter2).

Predpokladané vstupné vedomosti, skúsenosti a zručnosti

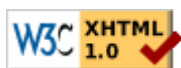
Účastník vie pracovať s prehliadačom webových stránok. Má skúsenosti s tvorbou webových stránok vo WYSIWYG editore, pozná niektoré pojmy z oblasti tvorby webových stránok (napr. čo je to titulok stránky, kódovanie, alternatívny text) a zásady tvorby webových stránok.

Tvorba webových stránok

Už vieme, že tvorbu webových stránok máme začať premyslením si obsahu, štruktúry a zvolením cieľovej skupiny čitateľov našich stránok. Rovnako ako v predchádzajúcom module budeme tvoriť profesijné webové sídlo, skladajúce sa z niekoľkých podstránok:



Z časových dôvodov nebudete v tomto module písať vlastný obsah (texty) webových stránok. Všetci budeme tvoriť rovnaké webové sídlo - sídlo imaginárnej učiteľky Barbory Hraškovej. Texty jednotlivých stránok sídla nájdete pripravené v e-learningovej podpore kurzu. Neskôr môžete informácie v nich nahradiť informáciami o Vás a Vašich predmetoch, rozšíriť ich, pridať ďalšie stránky.



Jazyk XHTML

Jazyk XHTML, štandardy

Webová stránka je v skutočnosti obyčajný textový súbor, v ktorom je pomocou špeciálneho značkovacieho jazyka - HTML - popísané, akým spôsobom má byť zobrazený štruktúrovaný obsah stránky, aké kódovanie sa má použiť pri zobrazení stránky, atď. Webový prehliadač jazyku HTML rozumie a na základe popisu nám zobrazí výslednú webovú stránku. Tento popis nazývame **zdrojový kód stránky**.

Zobrazenie zdrojového kódu stránky:

- v Internet Exploreri - v ponuke **Strana** zvolíme **Zobraziť zdrojový kód**,
- v Mozilla Firefox - v ponuke **Zobraziť** zvolíme **Zdrojový kód stránky**.

Jazyk HTML nie je programovací jazyk, ani nie je určený na spracovanie textu.

W3C (World Wide Web Consortium) je konzorcium, ktoré bolo vytvorené za účelom definovať štandardizované verzie jazyka HTML. Taktiež je zodpovedné za štandardizáciu ostatných technológií, ktoré súvisia s WWW, ako napríklad CSS (Cascading Style Sheets) a XML (Extensible Markup Language - rozšírený značkovací jazyk).

Zadanie 1

V prehliadači si zobrazte stránku <http://www.issep2011.org> a pozrite si jej zdrojový kód.

Rozumiete niečomu v kóde? Porovnajete výslednú stránku so zdrojovým kódom - nájdete v kóde texty, ktoré sa nachádzajú na stránke?

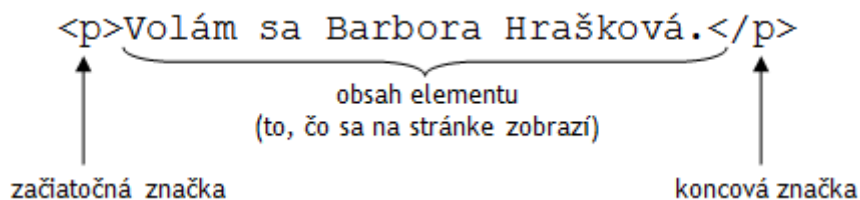
HTML (HyperText Markup Language) je značkovací jazyk, v ktorom sa definuje rozloženie dokumentu, syntax a rozmiestnenie prvkov na webovej stránke. Je určený na definovanie štruktúry a vzhľadu dokumentov.

Jazyk HTML nemá presné a striktné pravidlá, a preto stránky v tomto jazyku nevedeli niektoré zariadenia správne zobraziť. Neskôr konzorcium **W3C** definovalo jazyk (štandard) **XHTML (Extensible HyperText Markup Language)**, ktorý má striktné pravidlá. Používaním XHTML štandardu dokážeme redukovat' veľkosť zdrojového kódu stránky, oddelíme obsah od dizajnu, môžeme ľahšie ladiť zdrojový kód, zvýšime šance, že sa naša stránka zobrazí v rôznych prehliadačoch a zariadeniach (napr. mobil, PDA, rôzny pomocný softvér - napr. pre hendikepovaných), zabezpečíme spätnú kompatibilitu aj kompatibilitu do budúcnosti s prehliadačmi a inými zariadeniami.

Jazyk XHTML slúži na vytvorenie logickej a obsahovej štruktúry stránky (rozdelenie do blokov a odsekov), jednoduché formátovanie textu, vkladanie zoznamov, obrázkov, odkazov a tabuliek. Vzhľad stránky pomocou jazyka XHTML neriešime. V tomto module sa budeme zaoberať prevažne štandardom **XHTML 1.0**.

Jazyk XHTML tvorí množina **značiek**, tiež im hovoríme aj tagy. Sú to špeciálne slová uzavreté v zátvorkách <>, napr. <html>, <body>, , atď. Väčšina značiek sa v kóde nachádza v pároch, napr. a .

Zdrojový kód stránky sa skladá z (HTML) elementov. Príkladom elementu je napr. <p>Volám sa Barbora Hrašková.</p>. Na obrázku 1 si môžeme pozrieť, z čoho sa element skladá.



Obrázok 1 - Zloženie elementu

Niektoré elementy nemajú obsah, hovoríme im aj prázdne elementy, napr. element `
`. Prázdny element sa vždy začína znakom `<`, za ktorým nasleduje názov značky a končí znakmi `/>`. Elementy môžu mať aj atribúty, o tých si však povieme neskôr.

Zadanie 2

Ešte raz si pozrite zdrojový kód stránky www.issep2011.org. Aké elementy ste v ňom našli, s akými značkami?

Stránky (teda ich zdrojový kód) môžeme vytvárať v ľubovoľnom textovom editore (napr. Poznámkový blok, PSPad, 1st page a iné). Najlepšie je však používať taký editor, ktorý nám čiastočne pomáha pri písaní elementov (napr. PSPad). Ak si chceme pozrieť výsledok našej práce, musíme si stránku otvoriť vo webovom prehliadači (Mozilla Firefox, Opera, Internet Explorer a iné).

PSPad - www.pspad.com
1st Page - www.evrsoft.com

Štruktúra dokumentu

Webová stránka nemôže mať ľubovoľnú štruktúru. Presnejšie povedané štruktúru základných elementov, ktoré sa musia na stránke nachádzať.

Kostru webového dokumentu tvoria elementy `<html> </html>`, `<head> </head>`, `<title> </title>` a `<body> </body>`.

<code><html></html></code>	začiatok a koniec celého dokumentu
<code><head></head></code>	hlavička dokumentu (skrytá v prehliadači)
<code><title></title></code>	titulok dokumentu - zobrazuje sa v titulnom riadku prehliadača
<code><body></body></code>	telo dokumentu - zobrazuje sa v okne prehliadača

Elementy sa v zdrojovom kóde môžu nachádzať pod sebou, ale aj vedľa seba - nezáleží na tom (okrem špeciálnych prípadov). Dôležité je len poradie, v akom idú za sebou. Avšak vždy je dobré vytvárať zdrojový kód tak, aby bol dobre čitateľný.

Zadanie 3

Nájdite elementy `<html></html>`, `<head></head>`, `<title></title>` a `<body></body>` v zdrojovom kóde stránky z predchádzajúceho zadania. Všimnite si ich usporiadanie.

Všimli ste si, že elementy sa do seba vnášajú?

Správne usporiadanie dokumentu je nasledovné:

```

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

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

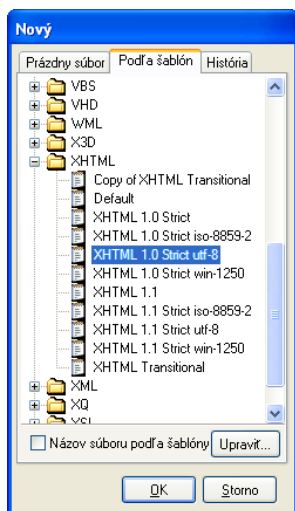
<head>

<title>Toto je titulok</title>

</head>

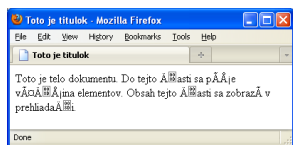
```

Prvé dva riadky uvedeného kódu treba brať ako „fakt“. Hovoria o tom, podľa akého štandardu je daná stránka vytvorená. Netreba si ich pamätať, ale treba ich skopírovať zo správneho zdroja, napr. http://www.w3schools.com/tags/tag_doctype.asp



Obrázok 3 - Výber šablóny v prostredí PSPad

Už v predchádzajúcom module sme sa naučili, že v názvoch súborov webových dokumentov nepoužívame diakritiku, ani medzery, a tiež, že domovskú (hlavnú) stránku webového sídla zvykneme pomenovávať `index.html`.



Obrázok 4 - Ukážka stránky s nenastaveným kódovaním

Element `<meta />` si nebudeme podrobnejšie vysvetľovať. V stručnosti: Pomocou neho môžeme definovať viacero skrytých informácií pre webový prehliadač, okrem iného aj typ kódovania stránky. Podrobnejšie informácie o elemente `<meta />` nájdete napr. na stránke http://www.w3schools.com/tags/tag_meta.asp.

```
<body>
```

Toto je telo dokumentu. Do tejto časti sa píše väčšina elementov. Obsah tejto časti sa zobrazí v prehliadači.

```
</body>
```

```
</html>
```

Zadanie 4

V prostredí PSPad vytvorte dokument popísaný vyššie.

Pomôcka

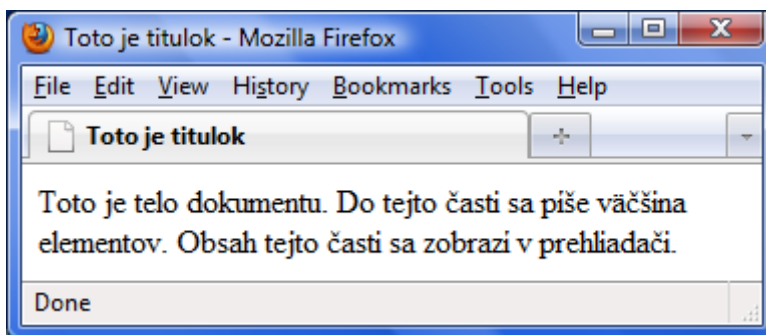
Kód nemusíte odpisovať ani kopírovať. Otvorte prostredie PSPad, zvolte ponuku **Súbor -> Nový** a prepnite sa do záložky **Podľa šablón**. Vyhľadajte šablónu **XHTML** a v rámci nej zvolte **XHTML 1.0 Strict utf-8**. Potvrďte výber tlačidlom **OK**.

Otvorí sa nový súbor s čiastočne preddefinovaným XHTML kódom. Porovnajta ho s kódom na predchádzajúcej strane.

Ako titulok uveďte svoje meno a priezvisko. Ako obsah elementu `<body></body>` doplňte zatiaľ ľubovoľný text.

Zadanie 5

Uložte vytvorený textový dokument ako `index.html` a pozrite si stránku vo webovom prehliadači.



Obrázok 2- Ukážka stránky vo webovom prehliadači

Kódovanie stránky

Aby sa nám správne zobrazovala stránka so slovenským textom vo webovom prehliadači, musíme jej nastaviť tzv. kódovanie. V prípade, že toto kódovanie nevedieme, nemusia sa správne zobrazit' slovenské znaky. V tomto module budeme používať univerzálne kódovanie **UTF-8**.

Kódovanie stránky nastavíme pomocou elementu `<meta />`, ktorý musí byť vložený do hlavičky stránky - čiže elementu `<head></head>`. Zdrojový kód pre nastavenie kódovania UTF-8 je nasledujúci:

```
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
```

Element `<head></head>` nášho dokumentu by teraz mal obsahovať minimálne nastavenie titulku a kódovania:

```
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<title>Barbora Hrašková</title>
</head>
```

Zadanie 6

Ak ste predchádzajúcu stránku vytvorili podľa danej šablóny, riadok definujúci kódovanie už bude vložený. Nájdite ho.

V prípade, že tento riadok kódu vo Vašom zdrojovom súbore nemáte, doplňte ho, uložte súbor a pozrite si stránku v prehliadači.

Kontrola zdrojového kódu stránky

V textových editoroch existujú nástroje na kontrolu pravopisu, ktoré nám „povedia“, či je náš text gramaticky správny. Podobne existuje nástroj, tzv. validátor, ktorý dokáže skontrolovať zdrojový kód stránky a povie nám, kde a aké chyby máme. Validátor nájdeme na stránke <http://validator.w3.org>.

Zadanie 7

Zvalidujte si svoju stránku.

Pomôcka

Prejdime na stránku <http://validator.w3.org>. Keďže vytvárame stránku na lokálnom počítači, zvolíme záložku **Validate by File Upload**, tlačidlo **Browse**, nájdeme náš súbor **index.html** a napokon stlačíme tlačidlo **Check**.

Pokiaľ je náš zdrojový kód správny, zobrazí sa správa v zelenom riadku **This document was successfully checked ...**, pokiaľ nie, zobrazí sa správa o tom, že v dokumente máme chyby **Error found while checking this document...** a nižšie nájdeme rozbor chýb.

Náš zdrojový kód je zatiaľ chybný. Chybou je, že samotné texty stránky nemôžeme mať priamo v elemente `<body></body>`, ale musíme ich vložiť do niektorého z blokových elementov. Čo je to blokový element a ako odstránime uvedenú chybu si ukážeme v nasledujúcej časti.

Prehliadače často dokážu zobrazit' aj stránky, ktorých zdrojový kód nie je validný, a dokonca často správne (rovnako, ako my dokážeme prečítať aj text s pravopisnými chybami). Ale napriek tomu by sme sa mali snažiť vytvárať validné zdrojové kódy.

Štruktúrovanie obsahu stránky

Po vytvorení základnej (povinnnej) štruktúry zdrojového kódu stránky môžeme pristúpiť k vytváraniu obsahu stránky a jeho štruktúrovania.

Zadanie 8

Skopírujte do elementu `<body></body>` text patriaci na úvodnú stránku (pôvodný obsah elementu `<body></body>` zrušte) - nájdete ho v e-learningovej podpore kurzu. Dokument uložte a stránku si pozrite v prehliadači.

Čím sa líši text vložený do kódu stránky a zobrazený text v prehliadači?

Ak chceme, aby prehliadač zobrazil text štruktúrovaný, musíme v zdrojovom kóde stránky určiť, ktorý text má byť odsek, ktorý nadpis a pod. - samozrejme, pomocou elementov.

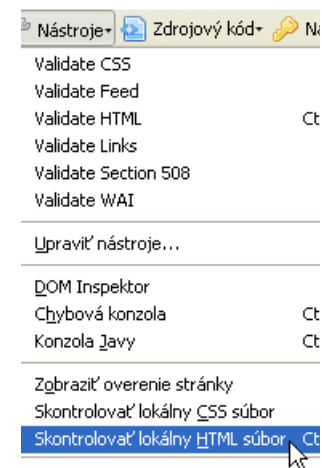
Základnými elementmi na štruktúrovanie obsahu stránky sú nadpisy (elementy `<h1></h1>`, `<h2></h2>`, ..., `<h6></h6>`) a odseky (element `<p></p>`). Tieto

Všimnite si, že názov znakovkej sady, v našom prípade utf-8, sa nachádza v doplnenom elemente na konci.

Ak napriek nastavenému kódovaniu zostávajú na stránke niektoré znaky „pokazené“, skúste v prostredí PSPad nastaviť formát dokumentu na UTF-8 - zvolte ponuku **Formát** a zvolte **UTF-8**. Znova dokument uložte a zobrazte stránku v prehliadači.

Pri tvorbe a ladení stránok môžeme využiť rôzne pomôcky, napr. doplnok do prehliadača Mozilla Firefox - *Web Developer* (<https://addons.mozilla.org/en-US/firefox/addon/60/>)

Validovanie stránky môžeme robiť priamo zo stránky www.w3.org, alebo môžeme využiť aj doplnok Web Developer v časti *Nástroje*:



Obrázok 5 - Validovanie stránky z doplnku Web Developer

Všetky súbory (texty, obrázky, zdrojové súbory) použité v materiáli sú v e-learningovej podpore kurzu.

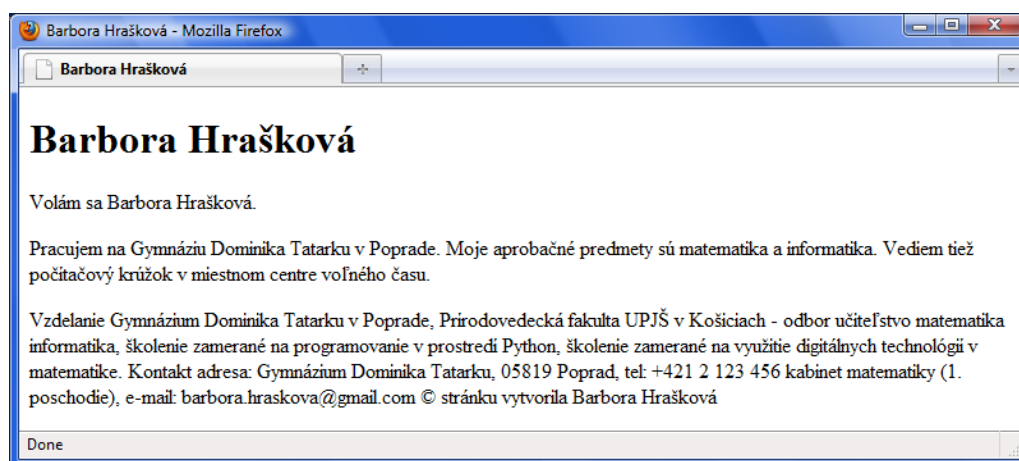
Blokové elementy za normálnych okolností nemôžu byť vedľa seba, ale len pod sebou. Zaberajú celú šírku priestoru, v ktorom sa nachádzajú. Napr. nadpis alebo odsek na našej stránke zaberá celú šírku okna prehliadača.

elementy nazývame blokové. Neskôr sa ešte stretneme s ďalšími blokovými elementmi napr. pre tabuľky, zoznamy.

Upravme časť zdrojového kódu našej stránky takto:

```
<body>
<h1>Barbora Hrašková</h1>
<p>Volám sa Barbora Hrašková.</p>
<p>Pracujem na Gymnáziu Dominika Tataruku v Poprade. Moje
aprobačné predmety sú matematika a informatika. Vediem tiež
počítačový krúžok v miestnom centre voľného času.</p>
...
</body>
```

Všimnite si, že nadpisy sa štandardne zobrazia tučné.

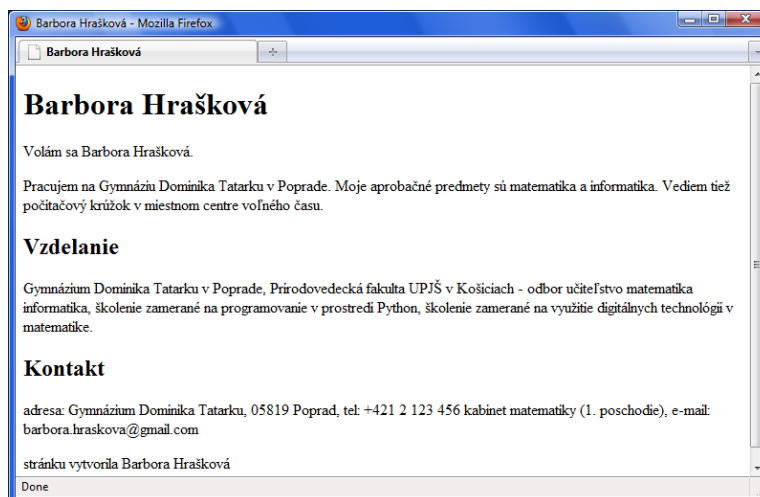


Obrázok 6 - Čiastočne štruktúrovaný obsah stránky

Zadanie 9

Pokračujte v štruktúrovaní hlavnej stránky. *Vzdelanie* a *Kontakt* označte ako nadpisy druhej úrovne (element `<h2></h2>`), ostatné časti ako odseky. Uložte stránku a pozrite si ju v prehliadači. Skontrolujte korektnosť zdrojového kódu validátorom.

Teraz by už stránka mala byť validná.



Obrázok 7 - Štruktúrovaná stránka

Okrem elementov pre nadpisy a odsek môžeme využiť špeciálny blokový element `<div></div>` (sekcia), do ktorého môžeme vložiť viacero blokových elementov (nadpisy, odseky, tabuľky, sekcie,...).

Element sekcie (`<div></div>`) môžeme na stránke využiť na logické rozčlenenie stránky na hlavičku, navigáciu, samotný obsah stránky (ktorý bude zvyčajne tvoriť viacero odsekov), pätičku a pod. Vďaka takémuto zoskupeniu, resp. rozčleneniu, elementov môžeme lepšie vytvoriť dizajn stránky pomocou kaskádových štýlov (ukážeme si to v druhej časti modulu).

Obsahom elementu `<div></div>` môžu byť takmer ľubovoľné elementy, vrátane ďalších div-ov. To však neplatí pre odseky a nadpisy. Vnútri elementov `<p></p>`, `<h1></h1>`, ..., `<h6></h6>` sa už žiadne ďalšie blokové elementy nachádzať nemôžu.

```

<body>
<div><h1>Barbora Hrašková</h1></div> ← hľavička
<div>
<p>Volám sa Barbora Hrašková.</p>
<p>Pracujem na Gymnáziu Dominika Tatarku v Poprade. Moje
apobačné predmety sú matematika a informatika. Viedem tiež
počítačový krúžok v miestnom centre voľného času.</p>
<h2>Vzdelanie</h2>
<p>Gymnázium Dominika Tatarku v Poprade, Prírodovedecká fakulta
UPJŠ v Košiciach - odbor učiteľstvo matematika informatika,
školenie zamerané na programovanie v prostredí Python, školenie
zamerané na využitie digitálnych technológií v matematike.</p>
<h2>Kontakt</h2>
<p>adresa: Gymnázium Dominika Tatarku, 05819 Poprad, tel: +421 2
123 456 kabinet matematiky (1. poschodie), e-mail:
barbora.hraskova@gmail.com</p>
</div>
<div><p>stránku vytvorila Barbora Hrašková</p></div> ← päta
</body>

```

obsah

Zadanie 10

Zmeňte zdrojový kód stránky podľa predchádzajúceho návodu. Pozrite si stránku v prehliadači.

Upravte kód stránky tak, aby text v pätičke bol v sekcii, ale nebol v odseku.

Formátovanie textu

Text vnútri rôznych blokov (nadpisov, sekcií, odsekov, tabuliek) môžeme formátovať rôznymi elementmi, napr. ``, ``, `<code></code>`, `<cite></cite>`, `
` atď. Význam najpoužívanejších uvádzame v nasledujúcej tabuľke:

<code>
</code>	Nový riadok v rámci sekcie, odseku alebo bloku textu
<code></code>	Zvýraznené písmo (zvyčajne šikmé)
<code></code>	Veľmi zvýraznené písmo (zvyčajne tučné)
<code><sub></sub></code>	Dolný index
<code><sup></sup></code>	Horný index
<code><big></big></code>	Veľké písmo

Všimnite si, že element `
` nemá koncovú značku, ani žiaden obsah - je to prázdny element. Tiež si všimnite, že začiatočná značka takéhoto prázdneho elementu je ukončená znakmi `/>`.

Zobrazenie jednotlivých elementov závisí od konkrétneho prehliadača. V druhej polovici modulu si ukážeme definovanie veľkosti pomocou kaskádových štýlov.

<code><small></small></code>	Malé písmo
<code><code></code></code>	Zdrojový kód
<code><cite></cite></code>	Citácia

Napr. nasledujúci zdrojový kód

```
<div>
Normálny text,
<em>zvýraznený text </em>,
<strong>veľmi zvýraznený text</strong>,<br/>
<sub>dolný index</sub>,
normálny text,
<sup>horný index</sup>,<br/>
<big>veľké písmo</big>,
<small>malé písmo</small>,<br/>
<code>kód</code>,
<cite>citácia</cite>.
</div>
```

zobrazí prehliadač takto:

Normálny text, *zvýraznený text*, **veľmi zvýraznený text**,
dolný index. normálny text, horný index,
veľké písmo, malé písmo,
kód, citácia.

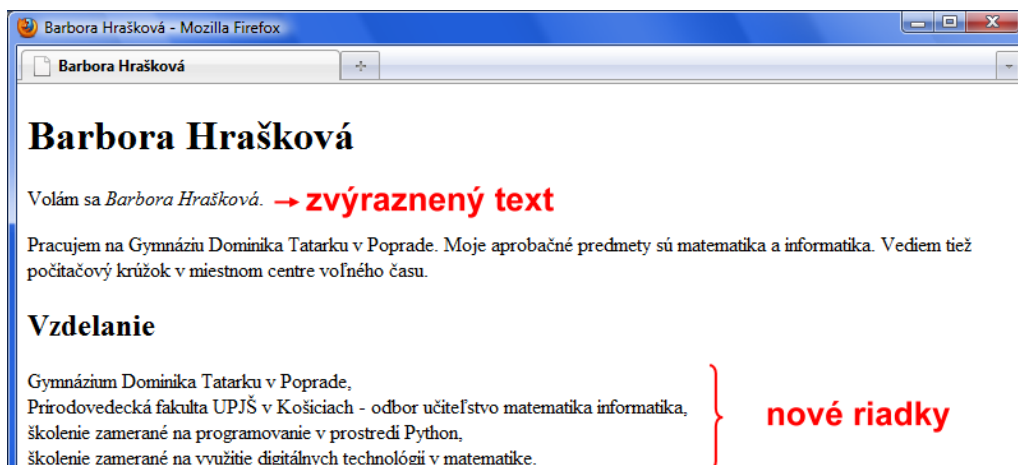
Obrázok 8 - Ukážky formátovania textu

Využime vhodné formátovanie pre niektoré časti našej stránky. V odseku *Volám sa Barbora Hrašková* zvýraznime meno a v odseku s informáciami o vzdelaní dajme každú školu, či školenie do nového riadka.

V zdrojovom kóde nemusíme za `
` písať do nového riadka, kód začiatok `
` nový riadok a kód začiatok `
` nový riadok zobrazí prehliadač rovnako. Robíme to skôr preto, aby aj zdrojový kód bol čitateľný.

```
<p>Volám sa <em>Barbora Hrašková</em>.</p>
<p>Pracujem na Gymnáziu Dominika Tatarku v Poprade. Moje aprobačné predmety sú matematika a informatika. Vedím tiež počítačový krúžok v miestnom centre voľného času.</p>
<h2>Vzdelanie</h2>
<p>Gymnázium Dominika Tatarku v Poprade,<br />
Prírodovedecká fakulta UPJŠ v Košiciach - odbor učiteľstvo matematika informatika,<br />
školenie zamerané na programovanie v prostredí Python,<br/>
školenie zamerané na využitie digitálnych technológií v matematike.
</p>
```

Uložíme zdrojový kód stránky a pozrieme si stránku v prehliadači (Obrázok 9).



Obrázok 9 - Ukážka stránky s formátovaním textu

Zadanie 11

Využite niektoré z elementov na formátovanie textu vo svojej stránke, napr. veľmi zvýraznite názov školy, na ktorej učíte.

V odseku s kontaktnými informáciami dajte každý druh kontaktu do nového riadka.

Stránku uložte, zobrazte v prehliadači a zvalidujte.

Vnárание elementov

Jednotlivé elementy v zdrojom kóde stránky nemôžeme vkladať len tak. Je dôležité, v akom poradí sú začiatkové značky a tiež koncové značky. Poradie koncových značiek musí byť v opačnom poradí začiatkových značiek.

Ak spojíme čiarami začiatkové a koncové značky elementov, tak pri správnom poradí elementov sa čiary nepretínajú (Obrázok 10). Správne vnárание elementov nám skontroluje validátor.

```
<div><h1>nadpis stránky</h1></div>
```

Obrázok 10 - Správne poradie značiek

```
<div><h1>nadpis stránky</div></h1>
```

Obrázok 11 - Nesprávne poradie značiek

Poradie začiatkových a koncových značiek je analogické poradiu zátvoriek v matematike - ktorú zátvorku *skôr* „otvorím“, tú neskôr „zavriem“, napr. $(\{[]\}[])$

Zadanie 12

Aké značky a kam by ste umiestnili, aby sa nasledujúci text zobrazil takto?

- **Malá morská víla**
- *Malá morská víla*
- **Malá morská víla**
- *Malá morská víla*

Zoznamy (s odrážkami, číslované)

Informácie o dosiahnutom vzdelaní sme na našej stránke formátovali ako odsek (element `<p></p>`), v ktorom sme využili element `
` na „zariadkovanie“. Keďže však ide o zoznam, je vhodnejšie použiť elementy určené na tvorbu zoznamov: číslovaných či s odrážkami. Oba typy zoznamov sa vytvárajú veľmi podobne.

Na vytvorenie zoznamu s odrážkami slúži element ``. Obsahom tohto elementu musia byť všetky prvky zoznamu, ktoré definujeme pomocou elementu ``.

```
<ul>
  <li>prvý prvok zoznamu</li>
  <li>druhý prvok zoznamu</li>
  <li>tretí prvok zoznamu</li>
</ul>
```

Zmeňme časť kódu stránky s informáciami o vzdelaní takto:

```
<h2>Vzdelanie</h2>
<ul>
  <li>Gymnázium Dominika Tatarku v Poprade,</li>
  <li>Prírodovedecká fakulta UPJŠ v Košiciach - odbor učiteľstvo matematika informatika,</li>
  <li>školenie zamerané na programovanie v prostredí Python,</li>
  <li>školenie zamerané na využitie digitálnych technológií v matematike.</li>
</ul>
```

Po uložení zdrojového kódu si pozrieme stránku v prehliadači (Obrázok 12).

Vzdelanie

- Gymnázium Dominika Tatarku v Poprade,
- Prírodovedecká fakulta UPJŠ v Košiciach - odbor učiteľstvo matematika informatika,
- školenie zamerané na programovanie v prostredí Python,
- školenie zamerané na využitie digitálnych technológií v matematike.

Obrázok 12 - Ukážka zoznamu s odrážkami

V druhej časti modulu si ukážeme, ako môžeme zmeniť vizualizáciu zoznamov pomocou kaskádových štýlov.

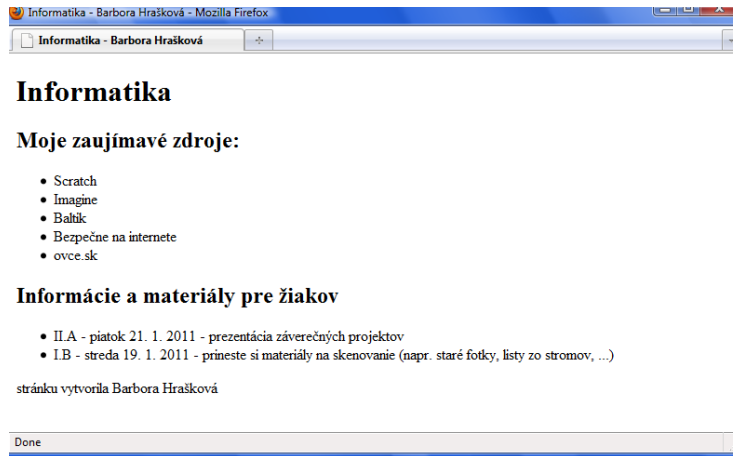
Ak chceme namiesto zoznamu s odrážkami použiť číslovaný zoznam, použijeme element ``. Jednotlivé prvky zoznamu potom definujeme opäť pomocou elementu ``.

Zadanie 13

Zmeňte odsek s kontaktnými údajmi na zoznam.

Zadanie 14

Vytvorte dve kópie stránky `index.html`: jednu pomenujte `informatika.html`, druhú `matematika.html`. Zmeňte obsahovú časť (sekciiu) oboch stránok - nahraďte aktuálny text textami zo súborov `informatika.txt` a `matematika.txt` z e-learningovej podpory kurzu. Túto sekciu ďalej štruktúrujte použitím elementov pre nadpisy, zoznamy, odseky. Vhodne upravte titulky oboch stránok.



Obrázok 13 - Ukážka stránky informatika

Obrázky

K textom, ktoré máme na stránke, vložíme obrázok. Na vloženie obrázka do dokumentu použijeme element ``. Element `` je, podobne ako element `
`, prázdny element, tvorí ho len začiatková značka (prípadne ju môžeme nazvať začiatko-koncová značka). Samotný element `` nám však nestačí. Potrebujeme mu nejakú povedať, odkiaľ má načítať zdrojový obrázok. K tomuto nám slúži **povinný atribút** `src`. Napríklad obrázok zo súboru `fotka.jpg` vložíme do stránky kódom:

```

```

Hodnotou atribútu `src` je „cesta k obrázku“, t.j. jeho umiestnenie. Pokiaľ uvedieme len názov súboru, znamená to, že prehliadač bude obrázok `fotka.jpg` hľadať na tom istom mieste (v tom istom priečinku), v ktorom sa nachádza naša stránka (mali by sme ho tam teda mať uložený!). Viac o cestách k obrázkom či iným súborom sa dozvieme v kapitole *Relatívne a absolútne adresovanie*.

Ďalším povinným atribútom elementu `` je atribút `alt`. Jeho hodnotou je alternatívny text k obrázku.

```

```

Dôležitými atribútmi sú aj `width` a `height`, ktorými určujeme, v akom veľkom priestore sa má obrázok zobrazit'. Hodnoty atribútov môžu byť v obrazových bodoch (pixeloch) alebo v percentách. Uvedením rozmerov obrázka uľahčíme jeho zobrazenie vo webovom prehliadači. Ten si môže vyhradiť priestor na stránke ešte pred načítaním obrázka zo servera.

```

```

Zadanie 15

Doplňte do zdrojového kódu stránky kód na vloženie Vašej fotky, napr. pred informácie o vzdelaní.

Prvkami zoznamu môžu byť ďalšie zoznamy, t.j. do elementu `` môžeme vnoriť element ``, resp. ``. Tak môžeme vytvárať aj zložené zoznamy. Napr. kódom

```
<ul>
  <li>Programovanie
    <ul>
      <li>Scratch</li>
      <li>Imagine</li>
      <li>Baltik</li>
    </ul>
  </li>
  <li>Bezpečnosť na
    internete
    <ul>
      <li>Bezpečne na
        internete</li>
      <li>ovce.sk</li>
    </ul>
  </li>
</ul>
```

vytvoríme takýto zoznam:

- Programovanie
 - Scratch
 - Imagine
 - Baltik
- Bezpečnosť na internete
 - Bezpečne na internete
 - ovce.sk

Význam alternatívneho textu pre obrázok už poznáme z modulu Internet: princípy a tvorba webu 2.

Nezabudnime, že najlepšie je najskôr upraviť obrázok v grafickom editore na takú veľkosť, v akej ho chceme mať na stránke, až potom ho na stránku vložiť. Hodnoty parametrov `width` a `height` by teda mali zodpovedať aktuálnej veľkosti obrázka, nemali by sme pomocou nich obrázok ani zmenšovať, ani zväčšovať.

Zopakujme si: Na stránkach môžeme používať len obrázky formátu jpg (vhodný pre fotky), gif alebo png (vhodné pre kresby, kliparty).

V prípade, že chceme k obrázku pridať ďalšiu doplňujúcu informáciu, ktorá by sa používateľovi zobrazila v bublinovej nápovede, použijeme atribút `title`. Napr.:

```

```

Atribúty elementov

Atribúty elementov sú akési vlastnosti (parametre), ktoré slúžia na doplnenie informácií pre element. Rôzne elementy majú rôzne atribúty. **Atribúty sa vždy uvádzajú v začiatkovej značke** v tvare `atribút="hodnota"`. Jeden element môže mať aj viacero atribútov. Atribúty elementu môžeme uvádzať v ľubovoľnom poradí. **Hodnoty atribútov musia byť uvedené v úvodzovkách alebo apostrofoch.** Vo všeobecnosti element s atribútmi zapíšeme takto:

```
<značka atribút1="hodnota1" atribútN="hodnotaN">toto je obsah elementu</značka>
```

alebo prázdny element takto:

```
<značka atribut1="hodnota1" atributN="hodnotaN" />
```

Podľa štandardu XHTML sú isté atribúty pre niektoré elementy povinné. Ak nám v zdrojovom kóde chýba povinný atribút pre niektorý element, validátor nebude takúto stránku považovať za správnu a označí v danom elemente chybu (vypíše, ktorý atribút chýba).

Zadanie 16

V zdrojovom kóde našej stránky sme sa už skôr stretli s jedným elementom s atribútmi. Pozrite si element `<meta/>`, ktorým sme definovali kódovanie - zistite, aké má atribúty a aké sú ich hodnoty.

Odkazy

V prípade, že chceme prepojiť nejaké dve webové stránky, vložíme do nich tzv. **hypertextový odkaz**. K tomu slúži element `<a>`. Obsahom elementu je text odkazu, ktorý sa zobrazí na stránke.

```
<a>text odkazu</a>
```

Podobne ako pri obrázkoch, aj pri odkaze musíme definovať atribút, ktorým zadáme, kam má odkaz smerovať. Ide o povinný atribút `href`. Jeho hodnotou je

- adresa stránky na internete, napr.

```
<a href="http://infovekacik.infovek.sk">Infovekáčik</a>
```
- alebo cesta k lokálnemu webovému dokumentu (prípadne ľubovoľnému súboru) - relatívna vzhľadom na stránku, v ktorej vytvárame odkaz, napr.

```
<a href="informatika.html">informatika</a>
```

Na stránke **Informatika** vytvoríme odkaz na stránku programovacieho prostredia Scratch:

1. V zdrojovom kóde dokumentu `informatika.html` nájdeme prvok zoznamu s textom Scratch:

```
<li>Scratch</li>
```
2. Slovo Scratch bude text nového odkazu, preto ho obalíme do značiek `<a>`, t.j. prvok zoznamu teraz bude:

```
<li><a>Scratch</a></li>
```
3. Do začiatkovej značky `<a>` pridáme atribút `href`, ktorého hodnotou bude adresa stránky prostredia Scratch, napr. <http://scratch.mit.edu/>:

```
<li><a href="http://scratch.mit.edu/">Scratch</a></li>
```

Podobne vytvoríme odkaz na stránku k prostrediu Imagine:

```
<li><a href="http://imagine.infovek.sk/">Imagine</a></li>
```

Zadanie 17

Vytvorte odkazy na stránky k prostrediu Baltík, projektu Bezpečne na internete a Ovce.sk. Adresy príslušných stránok nájdete v e-learningovej podpore kurzu.

Na domovskej stránke (**index.html**) vytvoríme odkazy na stránky informatika a matematika. Texty odkazov budú názvy predmetov v tomto odseku:

```
<p>Pracujem na Gymnáziu Dominika Tatarku v Poprade. Moje aprobačné predmety sú matematika a informatika. Vediem tiež počítačový krúžok v miestnom centre voľného času.</p>
```

Obalme text *matematika* značkami `<a>` a doplníme do začiatkovej značky atribút `href`. Keďže tentoraz robíme odkaz na lokálny dokument - súbor **matematika.html**, ktorý sa nachádza na rovnakom mieste ako stránka, kde tvoríme odkaz, bude hodnotou atribútu `href` len názov súboru. Kód odkazu na stránku matematiky teda bude: `matematika`

Podobne kód odkazu na stránku informatiky bude:
`informatika`

Odsek s hlavnými informáciami sa po vytvorení odkazov zmení takto:

```
<p>Pracujem na Gymnáziu Dominika Tatarku v Poprade. Moje aprobačné predmety sú <a href="matematika.html">matematika</a> a <a href="informatika.html">informatika</a>. Vediem tiež počítačový krúžok v miestnom centre voľného času.</p>
```

Zadanie 18

Medzi sekciu pre hlavičku a sekciu pre obsah vložte ďalšiu sekciu - pre navigáciu. Vložte od nej tri odkazy:

- odkaz na domovskú stránku,
- odkaz na stránku informatika,
- odkaz na stránku matematika.

Barbora Hrašková

[Domov](#) | [Informatika](#) | [Matematika](#)

Volám sa Barbora Hrašková.

Obrázok 14 - Ukážka navigácie

Skopírujte celú sekciu do stránok informatika a matematika.

Ďalším atribútom, ktorým môžeme doplniť odkaz, je nepovinný atribút `title`. Text, ktorý uvedieme v tomto atribúte, sa zobrazí v bublinovej nápovede pri nadínení myšou ponad odkaz (podobne, ako je to pri obrázku). **Upozornenie:** Neplet'me si tento atribút s **elementom** `<title></title>`.

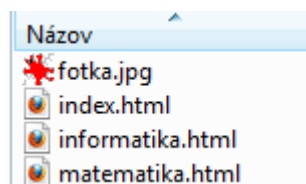
Atribút `title` je akýsi univerzálny atribút, ktorý môžeme dať ľubovoľnému elementu.

Relatívne a absolútne adresovanie

Relatívna adresa je adresa k zdrojovému súboru vzhľadom k aktuálnemu webovému dokumentu napr. `fotka.jpg`, `obrazky/pozadie.gif`, `informatika.html`, `matematika/pisomky/rovnice.html`.

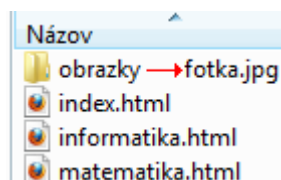
Pre odkazy na stránky a súbory v rámci nášho webového sídla je vhodnejšie používať lokálne adresy. Výhoda je v tom, že v prípade potreby môžeme naše sídlo premiestniť na inú adresu a nemusíme zmeniť žiadne (lokálne) adresy.

Ak je štruktúra nášho webového sídla ako na obrázku 15, potom zo stránky **index.html** sa na obrázok **fotka.jpg** odkazujeme cestou `fotka.jpg`.



Obrázok 15 - Štruktúra dokumentov 1

Pri štruktúre dokumentov z obrázka 16 by relatívna cesta zo stránky **index.html** k obrázku **fotka.jpg** bola `obrazky/fotka.jpg`.



Obrázok 16 - štruktúra dokumentov 2

Absolútnu adresu na dokument umiestnený na lokálnom počítači **nepoužívame!** Ak stránky s takouto adresou premiestnime na iný počítač, alebo čo len na iné miesto v našom počítači, odkazy prestanú fungovať.

V prípade, že sa chceme odkázať v hierarchii priečinkov o úroveň vyššie, použijeme znaky `../`, napr. ``. Ak sa potrebujeme odkázať o viac úrovní vyššie, použijeme toľkokrát znaky `../`.

Absolútna adresa je úplná adresa k zdrojovému súboru na serveri (alebo lokálnom počítači) bez ohľadu na umiestnenie nášho dokumentu, napr. `http://www.nejakyserver.sk/obrazky/logo.jpg` alebo pri umiestnení na lokálnom počítači `c://DVUI/2Inter3/stranka/fotka.jpg`.

Špeciálne znaky

Ako vyplýva z definícií elementov, znaky `<` a `>` majú špeciálne postavenie. Ak však chceme tieto znaky vložiť na stránku (napríklad pri vzorci $3 < 5$), nemôžeme ich len tak napísať do zdrojového kódu stránky. Webové prehliadače by to totiž interpretovali ako začiatok, resp. koniec značky. Namiesto nich vložíme špeciálne znaky `<` resp. `>`. Pre uvedený vzorec by teda zdrojový kód bol `3 < 5`.

Podobne je to aj s inými špeciálnymi znakmi. Špeciálny znak vždy začína znakom `&` a končí bodkočiarkou. Takýto zápis sa nazýva **entita**.

Niekoľko najpoužívanejších špeciálnych znakov a ich entít uvádzame v tabuľke:

špeciálny znak	entita
nedeliteľná medzera	<code>&nbsp;</code>
<code><</code>	<code>&lt;</code>
<code>></code>	<code>&gt;</code>
<code>&</code>	<code>&amp;</code>
©	<code>&copy;</code>
×	<code>&times;</code>
÷	<code>&divide;</code>
€	<code>&euro;</code>

Nedeliteľná medzera je medzera medzi slovami, ktorá zabezpečí, že slová vľavo a vpravo od nej musia byť spolu na jednom riadku.

Podrobný zoznam špeciálnych znakov je uvedený na stránke <http://www.w3.org/MarkUp/Guide/Advanced.html>.

Zadanie 19

Doplňte do päty všetkých stránok znak ©.

Tabuľky

Tabuľka je pomerne zložitá štruktúra, preto na jej definovanie potrebujeme viac elementov. Každá tabuľka sa skladá z riadkov, v ktorých je jedna alebo viacero buniek. Zdrojový kód jednoduchej tabuľky s dvoma riadkami a dvoma stĺpcami vyzerá takto:

```
<table>
  <tr>
    <th>meno</th>
    <th>priezvisko</th>
  </tr>
  <tr>
    <td>Barbora</td>
    <td>Hrašková</td>
  </tr>
</table>
```

meno	priezvisko
Barbora	Hrašková

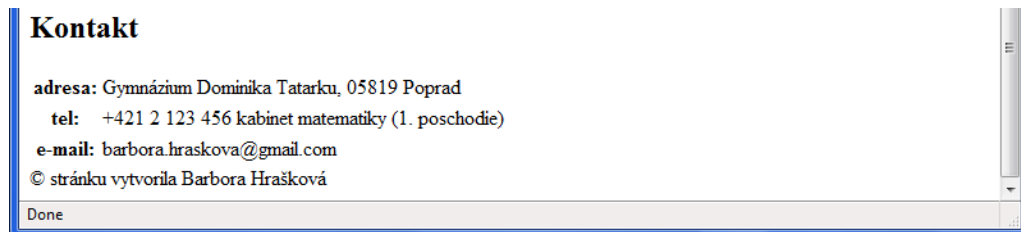
Obrázok 17 - Tabuľka vytvorená predchádzajúcim zdrojovým kódom

Popis použitých elementov, ako aj ďalších elementov používaných pri definovaní tabuliek, nájdeme v nasledujúcej tabuľke.

<code><table></table></code>	začiatok a koniec celej tabuľky (vnútri musí byť všetko, čo sa týka tabuľky)
<code><tr></tr></code>	definuje celý riadok (vnútri sú bunky)
<code><td></td></code>	definuje bunku v rámci riadka
<code><th></th></code>	definuje bunku v rámci riadka, ktorá je hlavičkou (zvyčajne zvýraznená tučným písmom)
<code><caption></caption></code>	legenda celej tabuľky, musí byť definovaná v rámci elementu <code><table></table></code> ; zobrazuje sa nad okrajom tabuľky (všetkými riadkami)

V prípade, že tabuľka nemá žiadne hlavičky riadkov alebo stĺpcov, element `<th></th>` neuvádzame.

Na našej stránke môžeme tabuľku použiť na prehľadnejšie uvedenie kontaktných informácií: v ľavom stĺpci budeme mať hlavičky riadkov (adresa, tel., e-mail), v pravom stĺpci príslušné údaje. Tabuľkou nahradíme pôvodný zoznam kontaktných údajov.



Obrázok 18 - Tabuľka s kontaktnými údajmi

Pri tvorbe zdrojového kódu tabuľky musíme **vždy postupovať po riadkoch**.

```
<table>
  <tr>
    <th>adresa:</th>
    <td>Gymnázium Dominika Tatarku, 05819 Poprad</td>
  </tr>
  <tr>
    <th>tel:</th>
    <td>+421 2 123 456 kabinet matematiky (1. poschodie)</td>
  </tr>
  <tr>
    <th>e-mail:</th>
    <td>barbora.hraskova@gmail.com</td>
  </tr>
</table>
```

Pokiaľ chceme mať tabuľku orámovanú, použijeme atribút `border` elementu `<table></table>`. Jeho hodnotou je číslo – šírka okraja tabuľky v bodoch.

Zadanie 20

Vyskúšajte zmeniť začiatočnú značku tabuľky na `<table border="1">`.

Na spájanie buniek v tabuľke vyžívame atribúty `colspan` a `rowspan` elementu `<td></td>`, resp. `<th></th>`. Ich použitie ukážeme na dvoch príkladoch.

V prvom príklade spojíme dve susedné bunky v jednom riadku. Ukážka stránky k príslušnému kódu je na obrázku 19.

```
<table border="1">
  <tr>
    <th>deň</th>
    <th>1. hodina</th>
    <th>2. hodina</th>
  </tr>
  <tr>
    <th>pondelok:</th>
    <td colspan="2">informatika</td>
  </tr>
</table>
```

Pri použití atribútov `colspan` a `rowspan` musíme dať pozor na počty buniek v riadkoch. Ak použijeme napr. `<td colspan="2">`, musíme mať v danom riadku o jednu bunku menej. Súčet hodnôt z atribútov `colspan` + počet buniek bez tohto atribútu (`td`, `th`) je počet stĺpcov v danom riadku, ktorý by mal byť zhodný s počtom stĺpcov celej tabuľky. Podobne, ak použijeme napr. `<td rowspan="2">`, musíme v nasledujúcom riadku mať o jednu bunku menej. Tieto pravidlá musia platiť pre všetky riadky. Ak by sme v niektorom riadku mali iný počet buniek, bude nám nejaká bunka chýbať (ostatné bunky v riadku sa posunú vľavo), alebo budeme mať bunku navyše, ktorá naruší štruktúru tabuľky.

deň	1. hodina	2. hodina
pondelok:	informatika	

Obrázok 19 - Spojenie buniek v riadku

V druhom príklade spojíme dve bunky nad sebou v jednom stĺpci. Ukážka stránky k príslušnému kódu je na obrázku 20.

```
<table border="1">
  <tr>
    <th rowspan="2">tel:</th>
    <td>+421 2 123 456</td>
  </tr>
  <tr>
    <td>+421 2 123 457</td>
  </tr>
  <tr>
    <th>e-mail:</th>
    <td>barbora.hraskova@gmail.com</td>
  </tr>
</table>
```

sem nedávame „prvú bunku“, pretože sem preteká bunka z predchádzajúceho riadka, ktorá má definovaný rowspan

tel:	+421 2 123 456
	+421 2 123 457
e-mail:	barbora.hraskova@gmail.com

Obrázok 20 - Spojenie buniek v stĺpci

Ďalšie vlastnosti tabuľky sa naučíme definovať pomocou kaskádových štýlov v 2. časti modulu.

Zadanie 21

Vytvorte novú stránku **rozvrh.html** ako kópiu stránky matematika. Ako obsah stránky vytvorte rozvrh hodín (stačia dva riadky).

Kaskádové štýly



V predchádzajúcej časti sme vytvorili základnú štruktúru stránky, ale bez akéhokoľvek dizajnu. Aktuálnym trendom tvorby webových stránok je vytváranie dizajnu len pomocou **kaskádových štýlov (CSS - Cascading Style Sheets)**, ktoré si predstavíme v tejto časti. Oboznámime sa so základnými vlastnosťami jednotlivých elementov (farby, rôzne nastavenia písma, zarovnávanie, orámovanie, priestory okolo elementov, ...), naučíme sa vytvárať vlastné triedy a pseudotriedy a používať ich.

Úvod do kaskádových štýlov

Kaskádové štýly poskytujú tvorcom webových stránok širšie možnosti formátovania, oddelenie štruktúry a štýlu, ľahkú tvorbu a údržbu „pevného“ štýlu a mnoho ďalšieho. Výhodou kaskádových štýlov je to, že zmenu stačí spraviť len v kaskádovom štýle a niekedy nemusíme zasahovať ani do zdrojového kódu stránky. Pomocou kaskádových štýlov tak vieme dosiahnuť pri minimálnej námahe napr. rovnaký nadpis, jednotné odrážky či rovnako vyzerajúce poznámky na všetkých stránkach.

Kaskádové štýly môžeme použiť tromi spôsobmi:

1. doplnok formátovacích možností jazyka XHTML (na úrovni nastavenia farieb a veľkosti písma),
2. formátovanie dokumentu bez rozmiestnenia prvkov na stránke,
3. formátovanie dokumentu vrátane rozmiestnenia prvkov na stránke.

Žiadnu z týchto možností nemôžeme označiť za tú najlepšiu, či najvhodnejšiu. Všetko závisí od rozsahu vytváraných stránok, či aktualizujeme už existujúce stránky alebo vytvárame všetko od začiatku.

Definovanie a vkladanie štýlov

V prípade, že kaskádové štýly máme uložené v súbore, tento súbor je obyčajný textový, ktorý môžeme vytvárať v rôznych editoroch. Jedným z editorov určených najmä na vytváranie kaskádových štýlov je napr. TopStyle. K nemu existuje aj voľne dostupná verzia TopStyle Lite (verzia 3.5).

TopStyle Lite (verzia 3.5):
<http://www.softpedia.com/get/Internet/WEB-Design/HTML-Editors/TopStyle-Lite.shtml>

Jednoduchá definícia kaskádového štýlu môže vyzerat' nasledovne:

```
p {
  color: blue;
  font-weight: bold;
}
```

Tento zápis znamená, že písmo vo všetkých odsekoch sa zmení na modré a tučné.

Vo všeobecnosti definujeme kaskádový štýl nasledovne:

```
selektor {
  vlastnosť1: hodnota1;
  vlastnosť2: hodnota2 hodnota3;
  ...
}
```

- **selektor** je napr. názov elementu, „vzorec“ alebo skupina, pre ktorú chceme štýl definovať.
- Samotný obsah štýlu definujeme **vnútri zložených zátvoriek**.
- Vnútri sú dvojice **vlastnosť: hodnoty**. Tieto dvojice od seba oddeľujeme bodkočiarkou.
- Viacero hodnôt pre jednu vlastnosť oddeľujeme medzerou.

Existuje niekoľko spôsobov, ako vložiť kaskádové štýly do webovej stránky:

- pomocou elementu `<link/>`, (štýly sú definované v samostatnom súbore, mimo zdrojového kódu stránky),
- pomocou elementu `<style></style>`, (štýly sú definované priamo v zdrojovom kóde stránky),
- pomocou atribútu `style` pre príslušný element.

My sa budeme zaoberať prevažne prvou možnosťou - budeme kaskádové štýly vkladať pomocou elementu `<link/>`. Tento spôsob je pravdepodobne najpoužívanejší a plne podporuje oddelenie štruktúry a obsahu stránky od dizajnu.

Pripojenie kaskádových štýlov pomocou elementu `<link/>`

Pomocou tohto elementu môžeme k súboru s webovou stránkou pripojiť externý súbor s kaskádovými štýlmi. Je to asi najpoužívanejší spôsob. Využíva sa vtedy, ak chceme na viacerých stránkach používať rovnaké štýly. Štýly si definujeme v externom súbore a do jednotlivých stránok ho len pripojíme. Pri zmene štýlu stačí urobiť zmenu v externom súbore a zmeny sa prejaví na všetkých stránkach, kde tento súbor so štýlmi používame. Práve tento spôsob úplne splňa ideu oddelenia štruktúry a dizajnu dokumentu.

Element `<link/>` musí byť použitý vnútri hlavičky dokumentu, teda vnútri elementu `<head></head>`.

Element `<link/>` má niekoľko parametrov, ktorých význam a popis je v nasledujúcej tabuľke:

atribút	hodnoty	popis
href	adresa súboru	cieľová adresa súboru s kaskádovým štýlom
type	v našom prípade musí byť <code>text/css</code>	typ obsahu súboru
rel	v našom prípade <code>stylesheet</code>	vzťah k aktuálnemu dokumentu
media	all aural braille embossed handheld print projection screen tty tv	médium, pre ktoré je kaskádový štýl určený; v súčasnosti sú podporované média <code>all</code> (ľubovoľné médium), <code>print</code> (tlač), <code>screen</code>

Definícia štýlu pomocou elementu

```
<style></style>
<head>
<style type="text/css">
p {
  color: #003399;
}
</style>
</head>
```

Definícia štýlu pomocou atribútu `style`:

```
<p style="color: #FFFF00;">Toto je obsah odseku - elementu p</p>
```

Znakom | oddeľujeme hodnoty atribútu, ktoré môže atribút nadobúdať. Túto konvenciu budeme používať aj ďalej v materiáli.

V nasledujúcom výpise je časť zdrojového kódu webovej stránky (`index.html`), ku ktorej chceme pripojiť kaskádové štýly uložené v súbore `style.css`. Štýly sa budú aplikovať na ľubovoľné médium. Hlavička súboru `index.html`:

K jednej webovej stránke môžeme pripojiť viacero kaskádových štýlov. Vtedy použijeme viacero elementov `<link/>`.

V tomto prípade prehliadač použije postupne všetky štýly pre dané médiá v poradí, ako sú definované.

V súbore s kaskádovými štýlmi už neuvádzame žiadne iné značky. Uvedený zdrojový kód je kompletný výpis súboru `styly.css`.

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

Súbor s kaskádovými štýlmi `styly.css` by mohol vyzerat' napríklad takto:

```
body {
    background-color: #ECF2FB;
}
p {
    color: #003399;
}
```

Vlastnosti v uvedenom zdrojovom kóde v stručnosti znamenajú to, že pre element `<body></body>` sa zmení farba pozadia a pre všetky elementy `<p></p>` sa zmení farba textu. Podrobne si uvedené vlastnosti vysvetlíme neskôr.

Zadanie 22

Vytvorte textový dokument s kaskádovými štýlmi podľa predchádzajúcej ukážky a pripojte ho k zdrojovému kódu webovej stránky. Po uložení oboch súborov si pozrite webovú stránku v prehliadači.

Element ako selektor

V úvode kapitoly sme pri definícii kaskádových štýlov uviedli, že selektor môže mať viacero podôb, jednou z nich je názov elementu. V takomto prípade sa štýl viaže len k tomuto konkrétnemu elementu - bude sa aplikovať na všetky tieto elementy v dokumente, kde platí kaskádový štýl.

Príklad: `p { color: #003399; }`

Tento štýl (farba písma) sa aplikuje na každý odsek (element `<p></p>`) v dokumente.

V nasledujúcej tabuľke je zoznam akýchsi premenných, ktoré budeme ďalej používať pri popise vlastností štýlov a budú zastupovať niektoré hodnoty.

%	Veľkosť/vzdialenosť vyjadrená v percentách, ktorá sa vzťahuje k nadradenému elementu.
color	Takmer každému elementu môžeme nastaviť farbu textu. Farbu môžeme zapísať viacerými spôsobmi: <ul style="list-style-type: none">• <code>#RRGGBB</code> - zložky <code>r,g,b</code> vyjadrené v hexadecimálnom tvare (napr. <code>#FFCC00</code>)• <code>rgb(r,g,b)</code> - zložky <code>r,g,b</code> vyjadrené v decimálnom tvare (napr. <code>rgb(255,104,0)</code>)• <code>meno_farby</code> - meno farby podľa štandardu (napr. <code>Black</code>, <code>CadetBlue</code>)
url	Pri niektorých elementoch môžeme použiť obrázok (napr. pozadie, odrážka). URL je teda adresa k súboru, priečinku alebo serveru. Adresa môže byť absolútna alebo relatívna (vzhľadom k CSS súboru, nie k XHTML súboru), napr. (napr. <code>url('obrazky/foto.jpg')</code>).

Mená niektorých farieb nájdeme napr. na stránke http://www.w3schools.com/tags/ref_colornames.asp.

length (veľkosť, rozmer)

Takmer každému elementu môžeme nastaviť nejakú vlastnosť, ktorá určuje veľkosť, napr. veľkosť písma, šírka/výška blokového elementu, veľkosť okrajov atď. Veľkosť môžeme definovať v rôznych jednotkách:

- **em** - rozmer znaku/písmena (napr. 2.5em)
- **ex** - rozmer písmena x, zvyčajne polovica rozmeru znaku (napr. 3ex)
- **px** - body obrazovky, hodnoty sú relatívne k aktuálnemu rozlíšeniu obrazovky (napr. 800px)
- **pt** - point (1pt = 1/72 inch), typografické body, pri tlači (napr. 800pt)
- **in** - palce
- **cm** - centimetre
- **mm** - milimetre

Pri všetkých jednotkách musíme hodnotu spolu s jednotkou uvádzať ako jedno slovo (bez medzery), napr. 2.5em, 3ex, 800px, 75%.

Kompletnú špecifikáciu CSS nájdete na adrese <http://www.w3.org/Style/CSS/>.

Na stránke <http://www.w3schools.com/css/default.asp> si môžete pozrieť a vyskúšať všetky vlastnosti kaskádových štýlov.

Vlastnosti textu

Po rôznych všeobecných informáciách si podme ukázať nejaké konkrétne kaskádové štýly, ktoré by sme mohli využiť na našich webových stránkach. V tejto časti si predstavíme vlastnosti, ktoré nejakým spôsobom nastavujú text, napr. farbu, veľkosť a typ písma, sklon, rôzne transformácie, medzery atď.

vlastnosť	hodnoty	popis
color	color	farba textu
font-family	napr. Arial, Helvetica, sans-serif, "Times New Roman", Times, serif, "Courier New", Courier, mono	určuje tzv. rodinu písma viacslovné názvy musia byť v úvodzokách alebo v apostrofoch
font-size	xx-small x-small small medium large x-large xx-large smaller larger length %	nastaví výšku písma preddefinované konštanty a hodnoty v jednotkách em, ex, % sa vzťahujú k rodičovskému elementu
font-style	normal italic oblique	nastaví štýl (šikmosť) písma
font-variant	normal small-caps	zobrazí text normálne, alebo ako malé kapitálky (veľkosť písmen je zachovaná, ale všetky vyzerajú ako veľké)

Hodnotou atribútu `font-family` môže byť viacero písiem oddelených čiarkou. Dôvod je ten, že nie každý používateľ musí mať nainštalované uvedené písmo, a preto ak ho nemá, použije sa ďalšie písmo v poradí. Zoznam (sada) písiem by mal byť ukončený tzv. typovým písmom, napr. `sans-serif`, `serif`, `monospace`, podľa ktorého sa riadi prehliadač.

Odporúčanie pri použití konštant veľkosti:

Niektoré konštanty (`x-small`, `small` atď.) si niektoré prehliadače interpretujú po svojom, a preto ich zobrazenie nemusí byť vždy rovnaké. Napr. prehliadač Internet Explorer 6 neinterpretuje konštantu `small`. Nie vždy sa tomu dá vyhnúť, ale často pomôže definovanie veľkosti v percentách alebo v jednotkách `em` alebo `ex`.

Texty v zdrojovom kóde stránky (malé a veľké písmená) by sme mali písať tak, ako by sme ich písali do normálneho dokumentu. Ak chceme niečo zobrazit' inak (napr. všetko veľkými alebo malými písmenami), mali by sme to riešiť pomocou kaskádových štýlov a nie ručne prepisovať texty.

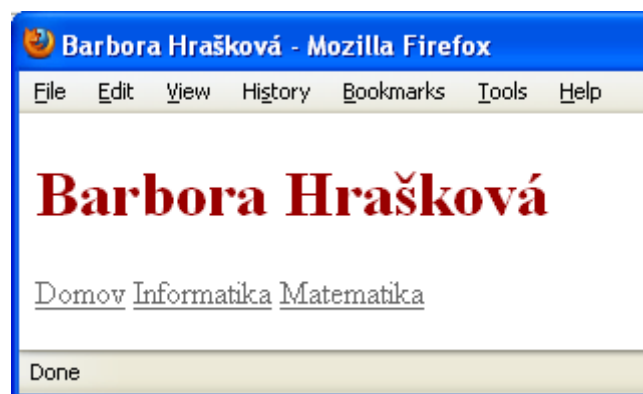
font-weight	normal bold bolder lighter 100 200 300 ... 800 900	nastaví hrúbku písma
letter-spacing	normal <i>length</i>	nastavuje medzery medzi písmenami
text-align	left right center justify	vodorovné zarovnanie textu v rámci blokového elementu (napr. nadpis, odsek, sekcia)
text-decoration	none underline overline line-through blink	nastavenie podčiarknutia, nadčiarknutia, prečiarknutia a blikania textu
text-transform	none capitalize uppercase lowercase	mení veľké a malé písmená
vertical-align	baseline top text-top middle bottom text-bottom <i>length</i> %	zvislé zarovnanie textu v rámci blokového elementu (napr. nadpis, odsek, sekcia)
word-spacing	normal <i>length</i>	nastavuje medzery medzi slovami

Podme niektoré z vlastností vyskúšať.

Vlastnosť `color` sme už spomenuli v úvodnej časti. Nastavujeme ňou farbu textu v odsekoch, nadpisoch, zoznamoch, tabuľkách atď. Najčastejšie sa zadáva v hexadecimálnom tvare (`#RRGGBB`), prípadne ako názov farby (pre základné farby). Túto vlastnosť môžeme využiť pri farbe nadpisu `h1` (`#990000`), a tiež pri farbe odkazov (`#666666`).

Súbor s kaskádovými štýlmi by mohol vyzerat' nasledovne:

```
h1 { color: #990000; }
a { color: #666666; }
```



Obrázok 21 - Stránka s upraveným hlavným nadpisom a odkazmi

Ak si pozrieme našu stránku vo webovom prehliadači (Obrázok 21), zistíme, že nadpisy sú väčšie ako iné písmo, aj keď sme im žiadnu veľkosť nenastavili. Je to

tým, že ak používateľ nenastaví žiadnu veľkosť nadpisu, prehliadač použije akúsi ním predvolenú veľkosť. Na toto sa však netreba spoliehať, lebo každý prehliadač môže nastaviť rôznu veľkosť. Preto je lepšie, ak si veľkosť nadpisov nastavíme sami pomocou kaskádových štýlov. Na to využijeme vlastnosť `font-size`. Veľkosť môžeme nastaviť v rôznych jednotkách, najčastejšie sa pre obrazovky používajú percentá (% , napr. 250%) a veľkosť znaku (`em`, napr. 2.5em). Tieto jednotky znamenajú relatívne zväčšenie/zmenšenie oproti normálnej veľkosti písma. Majú výhodu v tom, že ak si používateľ z nejakého dôvodu zmenší alebo zväčší písmo v prehliadači (napr. z dôvodu zrkovného postihnutia), zachová sa mu proporionalita veľkostí písma textu a nadpisov. V našom prípade môžeme pre nadpisy `h1` nastaviť 250%.

```
h1 {
  color: #990000;
  font-size: 250%;
}
```

Aby sme odkazy viac odlišili od textu (okrem farby), môžeme ich zvýrazniť napr. tučným písmom. K tomu využijeme vlastnosť `font-weight`. Z uvedených vlastností sa najčastejšie používa hodnota `bold`, ktorá funguje vo všetkých prehliadačoch. Fungovanie ostatných hodnôt nemusí byť vo všetkých prehliadačoch zaručené.

```
a {
  color: #666666;
  font-weight: bold;
}
```

Nikde na stránke ani v kaskádových štýloch sme nenastavili žiadny typ písma, no prehliadače nám stránku zobrazia bez akýchkoľvek varovaní. Prehliadače použijú, podobne ako pri veľkosti nadpisov, nejaké svoje štandardné nastavenie typu písma. To sa môže líšiť v jednotlivých prehliadačoch, preto aj pri tejto vlastnosti odporúčame nastaviť ju pomocou kaskádových štýlov. Ak chceme nastaviť typ písma pre celú stránku, nastavíme vlastnosť `font-family` elementu `body`. V našom prípade môžeme nastaviť `font-family: Arial, Helvetica, sans-serif;`.

```
body {
  font-family: Arial, Helvetica, sans-serif;
}
```

Zadanie 23

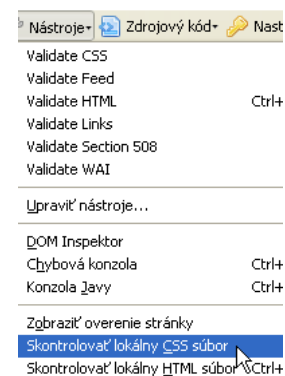
Pomocou kaskádových štýlov upravte niektoré nadpisy na stránke - použite aspoň tri rôzne vlastnosti textu (napr. veľkosť písma pre nadpisy `h2`, zarovnanie textu, medzery medzi písmenami/slovami, zmena malých/veľkých písmen).

Nastavenie pozadia

V predchádzajúcej časti sme nastavovali rôzne vlastnosti textu, okrem iného aj farbu. Avšak nenastavovali sme farbu ani iné vlastnosti pozadia. To si predstavíme v tejto časti.

Pozadie môžeme nastaviť buď celej stránke, alebo ľubovoľnej jej časti, napr. niektorému odseku, nadpisu, zoznamu. Ako pozadie môžeme nastaviť buď len farbu alebo aj obrázok. Pri obrázku na pozadí môžeme definovať pozíciu, prípadne automatické opakovanie obrázka na pozadí daného elementu.

Podobne ako pri XHTML, aj pri kaskádových štýloch môžeme využiť validátor, ktorý nám skontroluje syntaktickú správnosť kódu. Nájdeme ho na stránke <http://jigsaw.w3.org/css-validator/>. Môžeme tiež využiť doplnok Web Developer:



Obrázok 22 - Validovanie kaskádových štýlov z doplnku Web Developer

Nastavme hlavnému nadpisu `h1` farbu pozadia (vlastnosť `background-color`) na `#DCDCDC`. Obrázok na pozadí môžeme dočasne nastaviť napr. pre celú stránku (element `body`) - `background-image: url('pozadie.gif');`.

Ak v štýle použijeme relatívnu adresu k nejakému súboru (napr. k obrázku), tak táto relatívna adresa sa vzťahuje k súboru s kaskádovým štýlom, nie k XHTML súboru, ku ktorému kaskádové štýly pripájame.

```
body {
    font-family: Arial, Helvetica, sans-serif;
    background-image: url('pozadie.gif');
}
```

V nasledujúcej tabuľke sú uvedené vlastnosti pozadia, ktoré môžeme nastaviť ľubovoľnému elementu:

Nezabudnime, že farbu pozadia či obrázok na pozadí volíme tak, aby text na ňom bol dobre čitateľný.

Niekoľko stránok, kde môžeme nájsť pozadia: <http://www.grsites.com/archive/textures/>, <http://www.graphics-4free.com/backs.html>, <http://www.free-graphics.com/>

vlastnosť	hodnoty	popis
<code>background-color</code>	<i>color</i> transparent	farba pozadia
<code>background-image</code>	<i>url</i> none	obrázok ako pozadie
<code>background-position</code>	top left top center top right center left center center center right bottom left bottom center bottom right <i>x-% y-% x-pos y-pos</i>	umiestnenie obrázka na pozadí pozícia je definovaná ako dvojica pre vodorovnú a zvislú pozíciu
<code>background-repeat</code>	repeat repeat-x repeat-y no-repeat	ak je definovaný obrázok ako pozadie, tak táto vlastnosť definuje opakovanie obrázka v niektorom smere
<code>background-attachment</code>	scroll fixed	určuje, či sa obrázok na pozadí bude pri posúvaní stránky pohybovať, alebo nie

Zadanie 24

V kaskádových štýloch definujte pre celú stránku pozadie z vhodne zvoleného súboru tak, aby sa opakovalo len zvislo.

Zoznamy

V prvej časti modulu (XHTML) sme na stránku vložili zoznam dosiahnutého vzdelania, ktorý sme ďalej nijako neupravovali. Tvar odrážky, ktorá sa nám zobrazí, závisí od nastavení webového prehliadača. Rovnako ako pri veľkostiach textu, aj pri zoznamoch odporúčame nastaviť konkrétny štýl zoznamu pomocou kaskádových štýlov.

Jazyk XHTML ponúka len veľmi málo možností, akú odrážku či číslo nastaviť zoznamu (v najnovších špecifikáciách XHTML sa už vôbec neodporúča nastavovať štýl zoznamu v jazyku XHTML). Pomocou kaskádových štýlov môžeme nastaviť viacero typov odrážok a číslovania a navyše pomocou vlastnosti `list-style-image` môžeme zoznamom nastaviť ľubovoľný obrázok ako odrážku.

Na našej stránke môžeme využiť vlastné odrážky napr. pri zozname dosiahnutého vzdelania. V zdrojovom XHTML kóde nemusíme nič meniť, stačí, ak pre zoznam vytvoríme vhodný kaskádový štýl.

Keďže chceme zmeniť obrázok odrážky, nastavíme vlastnosť celému zoznamu s odrážkami `` a nie jednotlivým odrážkam (element ``).

```
ul {
  list-style-type: disc;
  list-style-image: url('odrazka.gif');
}
```

V tabuľke sú uvedené CSS vlastnosti, ktorými nastavujeme vlastnosti zoznamov.

vlastnosť	hodnoty	popis
<code>list-style-type</code>	<code>none</code> <code>disc</code> <code>circle</code> <code>square</code> <code>decimal</code> <code>decimal-leading-zero</code> <code>lower-roman</code> <code>upper-roman</code> <code>lower-alpha</code> <code>upper-alpha</code> <code>lower-greek</code> <code>lower-latin</code> <code>upper-latin</code> <code>hebrew</code> <code>armenian</code> <code>georgian</code> <code>ckj-ideographic</code> <code>hiragana</code> <code>katakana</code> <code>hiragana-iroha</code> <code>katakana-iroha</code>	typ zoznamu (odrážka, číslo)
<code>list-style-image</code>	<code>none</code> <i>url</i>	obrázok ako odrážka ak obrázok neexistuje, alebo sa nedá načítať, použije sa vlastnosť <code>list-style-type</code>
<code>list-style-position</code>	<code>inside</code> <code>outside</code>	umiestnenie textu vzhľadom na odrážku <i>inside</i> : v prípade, že text odrážky je vo viacerých riadkoch, tak nový riadok začína na úrovni odrážky a nie na úrovni začiatku textu

- ♦ Gymnázium Dominika T
- ♦ Prírodovedecká fakulta
- ♦ školenie zamerané na p
- ♦ školenie zamerané na v

Obrázok 23 - obrázok ako odrážka zoznamu

Pri nastavovaní obrázka ako odrážky je vhodné nastaviť aj typ odrážky pomocou vlastnosti `list-style-type`. Ak by obrázok v odrážke nebol dostupný, zobrazí sa aspoň odrážka.

Nevýhodou obrázka ako odrážky je to, že pri zmene veľkosti písma textu zoznamu sa veľkosť odrážky nezmení.

- a. Gymnázium Dominika T
- b. Prírodovedecká fakulta
- c. školenie zamerané na p
- d. školenie zamerané na v

Obrázok 24 - typ zoznamu *lower-alpha*

- i. Gymnázium Dominika T
- ii. Prírodovedecká fakulta
- iii. školenie zamerané na p
- iv. školenie zamerané na v

Obrázok 25 - typ zoznamu *lower-roman*

- i. Gymnázium Dominika T
- ii. Prírodovedecká fakulta
- iii. školenie zamerané na p
- iv. školenie zamerané na v

Obrázok 26 - typ zoznamu *lower-roman* (pozícia *inside*)

Zadanie 25

Zmeňte odrážky na úvodnej stránke na štvorčekové.

Pomenovanie elementov

Doteraz sme mohli pomocou kaskádových štýlov meniť vzhľad len pre konkrétne elementy (všetky odkazy, odseky atď.), ale nemohli sme odlišiť napr. hlavičku, navigáciu, pätičku a iné časti stránky. V nasledujúcich častiach si postupne ukážeme, ako odlišiť niektoré časti stránok, a tiež ako využiť nejaké kaskádové štýly na viacerých miestach. Využijeme na to tzv. **selektor ID** a **selektor triedy**.

Pomocou atribútu `id` môžeme pomenovať ľubovoľný element stránky. Hodnoty atribútov `id` musia byť jedinečné v rámci celej stránky. Ak by dva elementy mali rovnakú hodnotu atribútu `id`, chybu by odhalil XHTML validátor.

Doteraz sme mali definované pozadie pre celú stránku (element `<body></body>`), čo môžeme zrušiť a presunieme pozadie iba do navigácie.

Selektor ID

Predstavme si, že chceme sekciu s navigáciou zmeniť pozadie. Keby sme vytvorili štýl pre element `<div></div>`, zmenili by sme tým pozadie všetkých sekcií na stránke, čo ale nechceme. Aby sme nejako odlišili sekciu s navigáciou od ostatných sekcií, pomenujeme ju. Využijeme na to atribút `id`.

```
<div id="navigacia"> ... </div>
```

V súbore s kaskádovými štýlmi potom pridáme štýl, ktorého selektorom nebude žiaden element, ale tzv. **selektor ID**.

```
#navigacia {  
    background-image: url('pozadie.gif');  
}
```

Selektor ID v kaskádových štýloch sa používa pre prvky stránky, ktoré sa na nej budú nachádzať **len raz**, napr. hlavička, navigácia, pätička atď. Vztahuje sa len na prvky dokumentu, ktoré majú atribút `id` v zdrojovom kóde stránky zhodný s názvom selektora v kaskádových štýloch. Selektor ID obsahuje znak `#`, za ktorým je hodnota atribútu `id` (bez medzery). Napríklad `#navigacia`.

Zadanie 26

Vytvorte selektor ID pre pätičku stránky tak, aby bola pätička písaná trochu menším písmom ako zvyšný text (napr. 80%), zarovnaná v strede a farba textu aby bola `#CCCCCC`.

Selektor triedy môžeme na stránke použiť viackrát, čo je výhoda oproti selektoru ID.

Mohlo by sa zdať, že stačí používať selektory tried a nemusíme používať selektory ID. Na jednej strane to tak je, no na druhej strane nám selektory ID môžu pomôcť pri odhaľovaní chýb a sprehladnia kód. Ak totiž má byť nejaký prvok na stránke len raz (napr. hlavička, pätička), ale my by sme rovnaké meno (atribút `id`) použili viackrát, chybu nám odhalí XHTML validátor.

Selektor triedy

Pri vytváraní stránky sme už určite narazili na prípady, keď by sa nám hodilo odlišiť rovnaké elementy, ak sa na stránke vyskytujú viackrát. Napríklad by sme chceli odlišiť odkazy v navigácii a odkazy inde na stránke, alebo zvýrazniť niektoré slová v texte. Môžeme ich samozrejme všetky pomenovať pomocou atribútov `id`, ale pri viacerých odkazoch alebo iných elementoch, by sme mali príliš veľa pomenovaných elementov, pričom pre každý z nich by sme museli definovať samostatný kaskádový štýl. A to by bolo pomerne neprehľadné, zdĺhavé a hlavne neflexibilné.

V tomto nám pomôžu tzv. **selektory tried**. Takto vytvorené triedy môžeme priradovať elementom cez atribút `class`. Tento atribút môžeme definovať ľubovoľnému elementu, napr. `<p class="dolezite">Text</p>`.

V definícii kaskádového štýlu musíme meno triedy uvádzať **s bodkou pred menom triedy**, napr. `.dolezite`.

Selektory tried môžeme rozdeliť do dvoch kategórií:

1. **všeobecné** - môžeme ich priradiť ľubovoľným elementom (pred bodkou pri názve triedy už nie je žiadny selektor),
2. **pre elementy** - môžeme ich priradiť len konkrétnemu elementu (pre ktorý je trieda definovaná - názov elementu je hneď pred bodkou pred názvom triedy).

Všeobecné triedy

Výhodou všeobecných tried je to, že ich môžeme použiť pri ľubovoľnom elemente (odkazy, odseky, bunky, ...). Všeobecnú triedu `.dolezite` (tučné červené písmo) definujeme nasledovne:

```
.dolezite {
  color: #FF0000;
  font-weight: bold;
}
```

Túto všeobecnú triedu môžeme použiť napr. pri položke zoznamu (`<li class="dolezite">Text`), ale môžeme ju použiť napr. aj vnútri položky zoznamu na zvýraznenie slov (`Text`).

Element `` "zoskupí" znaky (text), a tak im môžeme nastaviť vlastnosti. Element `` môžeme použiť len vnútri blokového elementu.

Zadanie 27

Definujte triedu „upozornenie“, ktorú budete môcť použiť pre ľubovoľný element. Zvoľte červené (alebo iné výrazné) tučné písmo.

Použite triedu upozornenie na stránkach aspoň v dvoch rôznych elementoch.

Triedy pre elementy

V niektorých situáciách sa nám môže hodiť, ak dokážeme prepojiť triedu kaskádového štýlu len s konkrétnym typom elementu, napr. odsek alebo odkaz. Vtedy sa dá písať, resp. bude fungovať, len v elemente tohto typu. V inom type elementu nebude fungovať.

Definícia takejto triedy vyzerá nasledovne: `nazov_elementu.nazov_triedy`, napr. `a.navig`. (Pred názov všeobecnej triedy pridáme názov elementu, v ktorom má trieda platiť.)

Definujme triedu `a.navig` pre odkazy v navigácii, ktoré budú písané tučným písmom červenej farby a nebudú podčiarknuté.

```
a.navig {
  color: #990000;
  font-weight: bold;
  text-decoration: none;
}
```

Ak chceme takto vytvorenú triedu použiť pri odkazoch, zapíšeme v zdrojovom kóde stránky `Text`. Ak by sme chceli takto definovanú triedu použiť napr. aj pri odseku (`<p class="navig">Text</p>`), nebude to fungovať. Text odseku zostane nezmenený. Nikto nám nemôže zakázať takýto kód napísať, ale fungovať nebude.

Triedy kaskádových štýlov pre elementy majú vyššiu prioritu ako všeobecné triedy. Ak by existovali triedy `.navig` a `a.navig` a v nich by boli rovnaké vlastnosti, tak by sa použili vlastnosti definované v triede `a.navig`.

Pseudotriedy

Pseudotriedy sú špeciálne vlastnosti, ktoré rozširujú možnosti štýlov pre niektoré elementy. Uvádzame ich s dvojbodkou (napr. `:hover`). Nedefinujeme ich však samostatne, ale s nejakým selektorom (element, ID...), napr. `a:hover`.

Môžeme ich rozdeliť do dvoch skupín:

1. Pseudotriedy používané len pri hypertextových odkazoch (`:link`, `:visited`),
2. Pseudotriedy používané pri ľubovoľných elementoch (`:hover`, `:active`, `:focus`).

Pseudotriedy používané len pri hypertextových odkazoch

- `:link` - vzťahuje sa k ešte nenavštvíveným odkazom,
- `:visited` - vzťahuje sa k už navštvíveným odkazom.

Odkazom pri uvedených pseudotriedach môžeme nastavovať ľubovoľné vlastnosti kaskádových štýlov, ktoré sa vzťahujú k textu, písmu atď. Nemôžeme nastavovať vlastnosti, ktoré sa vzťahujú len na blokové elementy (napr. zarovnávanie textu).

Zmeňme vlastnosti všetkých nenavštvívených odkazov na sivú farbu a aby boli tučné:

```
a:link {
    color: #990000;
    font-weight: bold;
}
```

Podobne by sme definovali aj vlastnosti pre navštvívené odkazy (`a:visited`).

Pseudotriedy používané pri ľubovoľných elementoch

Nasledujúce pseudotriedy môžeme použiť pri ľubovoľných elementoch (aj odkazoch).

Upozornenie:

Pseudotriedy pri ľubovoľných elementoch nefungujú vo všetkých prehliadačoch. Napríklad `:hover` v Internet Exploreri 6 funguje len pre niektoré elementy. Napríklad pri nadpisoch, odsekoch a ďalších blokových elementoch nefunguje. Funguje prakticky len pri odkazoch.

- `:hover` - vzťahuje sa na element, ponad ktorý prechádzame myšou,
- `:active` - vzťahuje sa na element, ktorý je aktívny (napr. držíme stlačené ľavé tlačidlo myši),
- `:focus` - vzťahuje sa na element, ktorý je pripravený reagovať na vstup z klávesnice.

Ak by sme chceli nastaviť všetkým odkazom, že pri nadínení ponad ne sa zmení farba pozadia, vytvorili by sme definíciu kaskádového štýlu:

```
a:hover {
    background-color: #DCDCDC;
}
```

Z uvedeného výpisu môžeme vidieť, že ostatné vlastnosti nemusíme definovať, lebo sa zachovávajú. Ak by sme chceli niektoré vlastnosti zmeniť, musíme ich uviesť.

Upozornenie: Pseudotrieda `:hover` musí byť definovaná až po triedach `:link` a `:visited`. Pseudotrieda `:active` musí byť definovaná po triede `:hover`. V prípade, že pseudotriedy neuvedieme v správnom poradí, niektoré z nich nebudú fungovať.

Zadanie 28

Zabezpečte, aby sa pri nadínení myšou nad upozornenia (elementy naformátované triedou `upozornenie` zo zadania 27) tieto zobrazili veľkými písmenami.

Selektor nasledovníka

Pri použití selektora nasledovníka sa zjednoduší XHTML kód a pri zmene zdrojového kódu nemusíme nastavovať a myslieť na niektoré vlastnosti, lebo to zabezpečia kaskádové štýly.

Odkazy v navigácii sme doteraz nastavili pomocou atribútu `class="navig"`. Toto riešenie nie je zlé, ale má nevýhodu v tom, že každému odkazu v časti s navigáciou musíme tento atribút nastaviť. Ak by sme ho nenastavili, odkaz by sa zobrazil iným spôsobom. Pomocou selektora nasledovníka už toto dokážeme vyriešiť. **Selektor nasledovníka** definuje, ako sa majú správať vnorené elementy (do ľubovoľnej hĺbky).

Selektor nasledovníka definujeme: `selektor1 selektor2` (oddelené medzerou), kde `selektor2` je nasledovníkom `selektor1` (vnorený element).

Upravme odkazy v navigácii pomocou selektora nasledovnika. V súbore `styly.css` nahradime triedu `a.navig` selektorom `#navigacia a`, v zdrojovom XHTML kóde zrušime atribút `class="navig"`.

Zdrojový kód stránky časti s navigáciou sa zmení na:

```
<div id="navigacia">
  <a href="index.html">Domov</a>
  <a href="informatika.html">Informatika</a>
  <a href="matematika.html">Matematika</a>
</div>
```

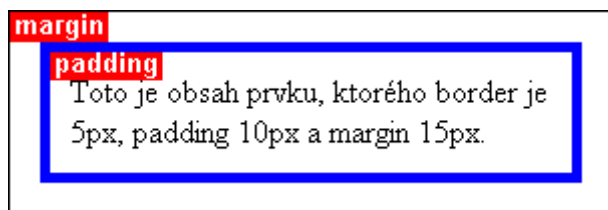
V súbore `styly.css` bude:

```
#navigacia a {
  color: #990000;
  font-weight: bold;
  font-size: 80%;
  text-decoration: none;
}
```

Okraje

Pre každý element sa automaticky generuje tzv. **box**. Box sa skladá z obsahu, vnútorného okraja (**padding**), orámovania (**border**) a vonkajšieho okraja (**margin**).

Postupne si ich všetky vysvetlíme a vyskúšame.



Obrázok 27 - Vzt'ah častí boxu

- **Border** na obrázku je modrý pásik. Je to viditeľný okraj (orámovanie) elementu.
- **Padding** je vzdialenosť obsahu elementu od orámovania elementu (vnútorný okraj).
- **Margin** je vonkajší okraj elementu, čiže ako blízko môžu ísť okolité elementy. Na obrázku je to vzdialenosť orámovania elementu od čierneho rámy (ľubovoľný element okolo).

Orámovanie

Orámovanie elementu je viditeľný okraj. Definujeme ho pomocou vlastnosti `border`, ktorej môžeme nastaviť **hrúbku**, **štýl orámovania** a **farbu** elementu (tzv. združená vlastnosť). Nastavíme ňou orámovanie pre všetky štyri okraje súčasne.

Existujú však aj vlastnosti `border-top`, `border-bottom`, `border-left`, `border-right`. Pomocou týchto vlastností môžeme samostatne definovať farbu, šírku a štýl orámovania pre horný, dolný, ľavý a pravý okraj elementu.

V prípade, že použijeme niektorú združenú vlastnosť (napr. `border-top`), tak poradie hodnôt musí byť: **hrúbka štýl farba** (oddelené medzerou).

Selektory a pseudotriedy

Selektory (ID, tried) a pseudotriedy môžeme vzájomne kombinovať. Platí však pravidlo, že najprv musí ísť definícia triedy a až potom definícia pseudotriedy.

Napríklad môžeme definovať štýly

```
p.zvyrazni:hover,
.zvyrazni:hover, a:link,
a.navig:link, a tiež aj
#pata:hover.
```

Podľa špecifikácie CSS sa **šírka elementu rovná šírke obsahu elementu**. Niektoré prehliadače (najmä Internet Explorer) však šírku elementu počítajú takto: **šírka elementu = ľavý margin + ľavý border + ľavý padding + šírka obsahu + pravý padding + pravý border + pravý margin**. Preto je niekedy ťažké zosúladiť správne zobrazovanie stránok v jednotlivých prehliadačoch (ak používame šírky elementov).

Okrem týchto vlastností existujú aj vlastnosti, ktorými môžeme definovať konkrétnu zložku, napr. len farbu okraja. Napr. pre dolný okraj môžeme použiť vlastnosti `border-bottom-color`, `border-bottom-style`, `border-bottom-width` a pre všetky ostatné okraje použijeme ekvivalentné vlastnosti.

Orámovanie môžeme na našej stránke využiť napríklad na optické oddelenie pätičky od ostatných častí - definujeme horný okraj pätičky:

```
#pata {
  border-top: 1px dashed #CCCCCC;
}
```

Jednotlivé zložky orámovania si vysvetlíme v nasledujúcej tabuľke:

Vlastnosť `border` a všetky ostatné príbuzné vlastnosti môžeme aplikovať na ľubovoľný viditeľný element. Aby sa zobrazilo orámovanie elementu, musia byť nastavené všetky jeho zložky (hrúbka, štýl, farba).

zložka orámovania	hodnoty	popis
farba	<i>color</i>	farba orámovania
štýl	none hidden dotted dashed solid double groove ridge inset outset	štýl orámovania
hrúbka	thin medium thick <i>length</i>	šírka orámovania

Na našej stránke môžeme orámovanie využiť ešte aj na oddelenie jednotlivých odkazov v navigácii.

Zadanie 29

Odkazy v navigácii oddelíte zvislým orámovaním farby #CCCCCC. Hrúbku aj štýl orámovania si zvolíte.

Vnútorne okraje

`padding: 0.5em;` (všetky vnútorné okraje budú veľkosti polovice znaku)

`padding: 2em 20px;` (horný a dolný okraj budú 2 znaky, okraje vľavo a vpravo budú 20 bodov)

`padding: 0 1em 5px 1em;` (horný: 0, vpravo: 1 znak, dolu: 5 bodov, vľavo: 1 znak)

Vzdialenosť obsahu elementu od jeho orámovania (vnútorný okraj) definujeme pomocou vlastnosti `padding`. Nastavíme ňou vnútorný okraj pre všetky štyri okraje súčasne. V prípade, že chceme nastaviť rovnakú vzdialenosť od všetkých štyroch okrajov, môžeme zadať len jednu hodnotu. V prípade, že všetky vzdialenosti nemajú byť rovnaké, potom poradie hodnôt musí byť v poradí: **hore vpravo dolu vľavo** (hodnoty oddeľujeme medzerou).

V prípade, že element má definovanú farbu pozadia, tak voľný priestor definovaný vlastnosťou `padding` alebo odvodených vlastností bude tiež podfarbený.

Vlastnosť `padding` a všetky ostatné odvodené vlastnosti môžeme aplikovať na ľubovoľný viditeľný element.

Veľkosť vnútorného okraja definujeme v rovnakých jednotkách ako napr. veľkosť textu (*length* | %).

Na našej stránke môžeme využiť vnútorný okraj napr. v navigácii:

```
#navigacia {
  background-image: url('pozadie.gif');
  padding: 0.5em;
}
```

V prípade, že chceme nastaviť rovnaké medzery vo vodorovnom a zvislom smere, môžeme vlastnosti `padding` zadať dve hodnoty, napr. `padding: 2em 20px`. Táto vlastnosť nastaví veľkosť vnútorného okraja vo zvislom smere na 2 znaky (2em) a vo vodorovnom smere to bude 20 bodov (20px).



Obrázok 28 - padding pre odkazy v navigácii (vidieť sivé pozadie)

Okrem vlastnosti `padding` existujú aj vlastnosti `padding-top`, `padding-bottom`, `padding-left`, `padding-right`. Pomocou týchto vlastností môžeme samostatne definovať vnútorný okraj pre horný, dolný, ľavý a pravý okraj elementu.

Vonkajšie okraje

Vzdialenosť orámovania elementu od okolitých elementov na stránke definujeme pomocou vlastnosti `margin`. Nastavíme ňou vonkajší okraj pre všetky štyri okraje súčasne. V prípade, že chceme nastaviť rovnakú vzdialenosť od všetkých štyroch okrajov, môžeme zadať len jednu hodnotu. V prípade, že všetky vzdialenosti nemajú byť rovnaké, potom poradie hodnôt musí byť v poradí: **hore vpravo dolu vľavo** (hodnoty oddelujeme medzerou).

Vlastnosť `margin` a všetky ostatné odvodené vlastnosti môžeme aplikovať na ľubovoľný viditeľný element.

Veľkosť vonkajšieho okraja definujeme v rovnakých jednotkách ako napr. veľkosť textu (`length` | %).

Vonkajšie okraje môžeme využiť napr. pri nadpisoch, ktoré si webové prehliadače prispôsobujú (ak nie sú zadané).

```
h1 {
  color: #990000;
  font-size: 250%;
  margin: 0.5em 0;
}
```

V prípade, že chceme nastaviť rovnaké medzery vo vodorovnom a zvislom smere, môžeme vlastnosti `margin` zadať dve hodnoty, napr. `margin: 2em 20px`. Táto vlastnosť nastaví veľkosť vonkajšieho okraja vo zvislom smere na 2 znaky (2em) a vo vodorovnom smere to bude 20 bodov (20px).

Okrem vlastnosti `margin` existujú aj vlastnosti `margin-top`, `margin-bottom`, `margin-left`, `margin-right`. Pomocou týchto vlastností môžeme samostatne definovať vonkajší okraj pre horný, dolný, ľavý a pravý okraj elementu.

`margin: 0.5em;` (všetky vonkajšie okraje budú veľkosti polovice znaku)

`margin: 2em 20px;` (horný a dolný okraj budú 2 znaky, okraje vľavo a vpravo budú 20 bodov)

`margin: 0 1em 5px 1em;` (horný: 0, vpravo: 1 znak, dolu: 5 bodov, vľavo: 1 znak)

Vlastnosť `margin` môžeme využiť na prisunutie celého obsahu stránky k okrajom okna prehliadača. Aj keď by k tomuto logicky mala slúžiť vlastnosť `padding`, v tomto prípade prehliadače reagujú väčšinou na vlastnosť `margin`. Môžeme pre istotu nastaviť obe vlastnosti:

```
body { margin: 0; padding: 0; }
```

Bunkám tabuľky nemá zmysel nastavovať vlastnosť `margin`, lebo okolie buniek určuje tabuľka a jej vlastnosti.

Pozor na rozdiel pri definovaní orámovania tabuľky len pomocou XHTML (`border="5"`) a pomocou CSS.

Zadanie 30

Zabezpečte, aby odkazy v navigácii mali medzi sebou „väčšie rozostupy“ (viac priestoru okolo seba zľava a sprava - aspoň 1 znak). Vyskúšajte použiť vlastnosť `padding` aj `margin` a porovnajte rozdiely.

Zadanie 31

Skúšajte nastavenia vonkajších a vnútorných okrajov pre rôzne elementy či selektory.

Tabuľky

Na úvodnej stránke máme vloženú tabuľku s kontaktnými údajmi, ktorú sme však zatiaľ nijako neupravovali. V kaskádových štýloch existujú aj vlastnosti, ktorými môžeme upraviť vzhľad tabuliek. Zo všetkých vlastností pre tabuľky sú pre nás najzaujímavejšie: `border-collapse`, `border-spacing`, `empty-cells`. Podme si ich postupne vysvetliť a vyskúšať.

V prvej polovici modulu sme si ukázali vlastnosť `border`, ktorou môžeme orámovat ľubovoľný viditeľný element - okrem iného aj tabuľku. Pomocou kaskádových štýlov nestačí definovať orámovanie len pre tabuľku (element `<table></table>`), ale

musíme orámovanie definovať aj pre jednotlivé bunky (element `<td></td>`, prípadne aj `<th></th>`).

vlastnosť	hodnoty	popis
<code>border-collapse</code>	<code>collapse</code> <code>separate</code>	<p>Definuje správanie orámovania buniek tabuľky.</p> <ul style="list-style-type: none"> <code>collapse</code> - Orámovania buniek, riadkov a stĺpcov budú splyvať. Čiary orámovania budú vycentrované v priestore medzi bunkami. <code>separate</code> - Každá bunka tabuľky má svoje vlastné orámovanie a vzdialenosť medzi bunkami je riadená vlastnosťou <code>border-spacing</code>. V tomto režime nemôžeme orámoviť riadky a stĺpce.
<code>border-spacing</code>	<i>length length</i> <i>length</i>	<p>Určuje vzdialenosť medzi orámovaním susedných buniek. Priestor medzi orámovaním je vyplnený pozadím tabuľky.</p> <p>Ak zadáme dve hodnoty, prvá určuje vzdialenosť vo vodorovnom a druhá v zvislom smere. Ak zadáme len jednu hodnotu, určuje vzdialenosť pre oba smery.</p> <p>Táto vlastnosť nefunguje v Internet Exploreri 6.</p>
<code>empty-cells</code>	<code>show</code> <code>hide</code>	<p>Určuje, či sa orámovanie buniek s neviditeľným obsahom má zobrazit'/skryť.</p> <p>Za neviditeľný obsah považujeme napr. klasické medzery. Tzv. nedeliteľná medzera sa nepovažuje za neviditeľný obsah.</p> <p>Táto vlastnosť má zmysel len v prípade oddeleného orámovania buniek tabuľky (<code>border-collapse: separate</code>).</p>

Upozornenie: vlastnosť `border-spacing` nefunguje v Internet Exploreri 6. Pred použitím vlastností v tabuľke sa presvedčte o ich podpore v aktuálne najpoužívanejších prehliadačoch.

Správanie niektorých vlastností tabuľky si ukážeme na dočasnej stránke s kontaktnými údajmi (len pre ilustráciu). Upravíme si úvodnú stránku, na ktorej zduplikujeme niekoľkokrát tabuľku s kontaktnými údajmi, pričom každej z tabuliek nastavíme iné vlastnosti. Pre jednotlivé tabuľky si definujeme triedy kaskádových štýlov `tab2`, `tab3` a `tab4`. Postupne si vysvetlíme a ukážeme jednotlivé triedy.

Prvej tabuľke sme nenastavili žiadny kaskádový štýl - je to naša základná tabuľka, s ktorou budeme ostatné porovnávať. Tabuľka má nastavené iba orámovanie pomocou atribútu `border="1"` v zdrojovom kóde stránky.

```
.tab2 {
border-collapse:
collapse;
}
```

Druhej tabuľke sme priradili štýl `.tab2`. Nastavili sme, aby orámovanie tabuľky a buniek splyvalo. Ak by sme nastavili medzery medzi bunkami (vlastnosť `border-spacing`), nemalo by to vplyv na zobrazenie tabuľky.

```
.tab3 {
border-collapse:
separate;
}
```

Tretej tabuľke sme priradili štýl `.tab3`. V tomto štýle bude orámovanie buniek oddelené. Toto nastavenie zodpovedá pôvodnému zobrazeniu tabuľky. Ak by sme

nastavili medzery medzi bunkami (vlastnosť `border-spacing`), orámovania buniek by boli od seba viac vzdialené - táto zmena je v štýle `.tab4`.

adresa: Gymnázium Dominika Tatarku, 05819 Poprad	adresa: Gymnázium Dominika Tatarku, 05819 Poprad
tel: +421 2 123 456 kabinet matematiky (1. poschodie)	tel: +421 2 123 456 kabinet matematiky (1. poschodie)
e-mail: barbora.hraskova@gmail.com	e-mail: barbora.hraskova@gmail.com

.tab2

.tab3

Obrázok 29 - Tabuľka kontaktov (štýl `.tab2` a `.tab3`)

Posledné dve tabuľky používajú identický kaskádový štýl `tab4`. Líšia sa len nastavením zdrojového XHTML kódu. Štvrtá tabuľka má v XHTML nastavené orámovanie `border="1"`, ale posledná tabuľka tento atribút nemá nastavený. Na obrázku 30 je vidieť rozdiel. Kým v tabuľke vľavo (`.tab4`) sú orámované všetky bunky tabuľky, v tabuľke vpravo (`.tab5`) je orámovaná len tabuľka. Ak chceme orámovat' pomocou kaskádových štýlov všetky bunky, musíme orámovanie nastaviť každej bunke tabuľky (pozor na samostatné definovanie štýlov pre `<td>` a `<th>`).

adresa: Gymnázium Dominika Tatarku, 05819 Poprad	adresa: Gymnázium Dominika Tatarku, 05819 Poprad
tel: +421 2 123 456 kabinet matematiky (1. poschodie)	tel: +421 2 123 456 kabinet matematiky (1. poschodie)
e-mail: barbora.hraskova@gmail.com	e-mail: barbora.hraskova@gmail.com

.tab4

.tab5

Obrázok 30 - Tabuľka kontaktov (štýl `.tab4` a `.tab5`)

Rozmery

V tejto kapitole popíšeme vlastnosti, ktorými môžeme nastaviť rozmery a prislúchajúce vlastnosti rôznym elementom. Zoznam vlastností je v nasledujúcej tabuľke:

vlastnosť	hodnoty	popis
<code>height</code>	<code>auto</code> <i>length</i> %	definuje výšku elementu
<code>overflow</code>	<code>visible</code> <code>hidden</code> <code>scroll</code> <code>auto</code>	Definuje, čo sa udeje s obsahom elementu, ktorý presahuje jeho obsah. <ul style="list-style-type: none"> <code>visible</code> - Obsah sa neoreže a môže presahovať hranice elementu. <code>hidden</code> - Obsah sa oreže a nie je možné si orezaný obsah inak pozrieť. Rozmer a tvar prekrytej oblasti definujeme vlastnosťou <code>clip</code>. <code>scroll</code> - Obsah sa oreže, ale môžeme si ho pozrieť vďaka posúvačom, ktoré by mali byť k dispozícii. <code>auto</code> - Orežanie či neorežanie závisí od webového prehliadača. V prípade, že by mal byť obsah elementu orezaný, malo by byť umožnené posúvanie v ňom.
<code>width</code>	<code>auto</code> <i>length</i> %	definuje šírku obsahu elementu

Všimnite si, že v pôvodnej tabuľke neboli orámovania čisto čiernou farbou, ale smerom do niektorých rohov sa menili na sivú farbu. Pôsobí to ako plastické tlačidlo. Nie je to žiaden trik. Takýto "plastický" vzhľad docielime pomocou štýlov `inset` a `outset`.

```
.tab4 {
border-collapse:
separate;
border-spacing: 1em;
border: 1px solid
#000000;
}
```

Ak definujeme šírku elementu, môže nastať problém s centrovaním objektu v okne prehliadača. Nemôžeme použiť vlastnosť `text-align`, lebo tá je určená len na vnútro elementu a nie na samotný element. Môžeme však použiť vlastnosti `margin-left: auto;` `margin-right: auto;;`, pomocou ktorých element vycentrujeme.

Vlastnosť overflow

Blokový element je štandardne ohraničený svojím obsahom. Niekedy však môže obsah elementu presiahnuť jeho hranice, teda môže byť celý alebo čiastočne mimo hranice elementu. Táto situácia môže nastať napríklad vtedy, ak má element nastavenú pevnú šírku, ale vo vnútri je text, ktorý sa nedá rozdeliť (zalomiť) a je širší ako šírka elementu. Podobne to môže byť pre výšku elementu. Práve pomocou vlastnosti `overflow` môžeme nastaviť, čo sa má udiť s obsahom elementu, ktorý presiahne jeho hranice.

Zadanie 32	Nastavte výšku bloku s navigáciou na rôzne hodnoty, napr. 30 bodov alebo 2 znaky.
Zadanie 33	Vyskúšajte si vlastnosť <code>overflow</code> napr. pri zozname dosiahnutého vzdelania. Nezapomnite si nastaviť presnú šírku a výšku elementu.

Štýly pre iné zariadenia

Pri vkladaní kaskádových štýlov môžeme zvoliť médium, pre ktoré je štýl platný (pomocou atribútu `media`). Vyššie sme napísali, že v súčasnosti podporovanými hodnotami vlastnosti `media` sú: `all`, `print`, `screen`.

Výhoda tvorby štýlov pre rôzne zariadenia je v tom, že nemusíme vytvárať novú stránku prispôbenú pre dané médium, ale používame rovnaký zdrojový kód ako pre obrazovku počítača, resp. iné zariadenia. A až nastavením kaskádových štýlov nastavíme „správanie“ stránky pre dané médium.

Takisto môžeme skryť rôzne objekty, napr. navigáciu, reklamu, ankety a pod.

Štýly pre tlač

Vytvorme si štýl pre tlač. Kaskádový štýl pre tlač vložíme nasledovne:

```
<link href="styley-tlac.css" rel="stylesheet" type="text/css"
media="print" />
```

Pri definovaní štýlu pre tlač musíme presne nastaviť, ako sa majú jednotlivé elementy na stránke zobrazovať a správať.

Pre našu stránku môžeme skryť navigáciu (pomocou vlastnosti `display: none`) a všetkým textom nastaviť čiernu farbu písma. Stránka by potom mohla vyzeráť podobne ako to ukazuje Obrázok 31.

```
body {
    font-family: Arial, Helvetica, sans-serif;
    background-color: #FFFFFF;
    color: #000000;
    font-size: 12pt;
}
h1 {
    color: #000000;
    font-size: 250%;
    margin: 0.5em 0;
}
```

Pri tvorbe kaskádových štýlov pre iné médiá si musíme uvedomiť ich špecifiká. Napr. ak budeme vytvárať štýl pre tlač, namiesto jednotiek `px` použijeme radšej `pt` (pointy, typografické body) alebo `mm`, `cm`.

Upozornenie:

Pri vkladaní štýlu pre tlač by sme nemali v elemente `<link/>` definovať atribút `title`, nakoľko v niektorých prehliadačoch (vrátane Mozilla Firefox) to znefunkční daný štýl.

Vlastnosť `display` určuje, či a ako sa má element zobrazovať. Pomocou tejto vlastnosti dokážeme zmeniť spôsob zobrazenia elementu, a tak zmeniť jeho pôvodné správanie. Musíme pritom zvážiť dôsledky. Ak totiž zmeníme spôsob správania elementu, napr. z riadkového na blokový, alebo naopak, neovplyvní to len samotný element, ale aj ďalšie elementy a v konečnom dôsledku to môže mať vplyv na celkový vzhľad a rozloženie stránky.

```
#navigacia {
  display: none;
}
```

Podrobnejšie informácie o jednotlivých hodnotách vlastnosti `display` nájdete na stránke www.w3schools.com/css/pr_class_display.asp.

Barbora Hrašková

Barbora Hrašková

Volám sa Barbora Hrašková.

Pracujem na **Gymnázium Dominika Tatarku** v Poprade. moje aprobačné predmety sú **matematika** a **informatika**. Viedem tiež počítačový krúžok v miestnom centre voľného času.

Vzdelanie

- Gymnázium Dominika Tatarku v Poprade
- Prírodovedecká fakulta UPJŠ v Košiciach - odbor učiteľstvo matematika informatika
- školenie zamerané na programovanie v prostredí Python
- školenie zamerané na využitie digitálnych technológií v matematike

Kontakt

adresa: Gymnázium Dominika Tatarku , 05819 Poprad
tel: +421 2 123 456 kabin et matematiky (1. poschodie)
e-mail: barbora.hraskova@gmail.com

© stránku vytvorila Barbora Hrašková



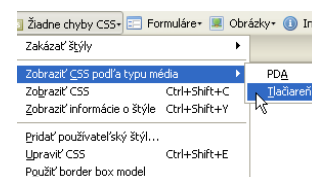
Obrázok 31 - Náhľad stránky pred tlačou

Ako zobrazit' štýly pre tlač

Možností je niekoľko:

- Zvolíme náhľad pred tlačou.
- Ak máme nainštalovaný doplnok *Web Developer* do prehliadača Mozilla Firefox, tak v ňom zvolíme *CSS => Display CSS By Media Type => Print*
- Použijeme ho dočasne ako primárny štýl.

Ďalšie informácie o vytváraní štýlov pre tlač nájdete na stránke **CSS Design: Going to Print** (www.alistapart.com/articles/goingtoprint/).



Obrázok 32 - Zobrazenie štýlu pre tlač v doplnku *Web Developer*

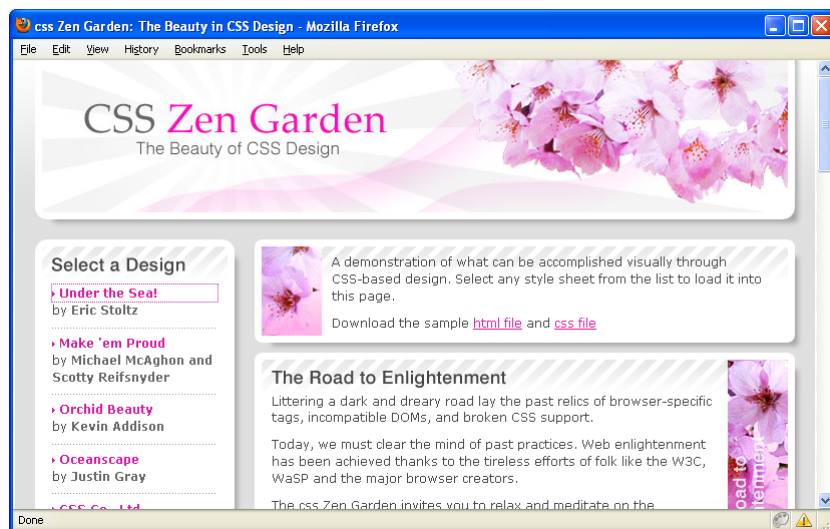
Záver

Problematika kaskádových štýlov je oveľa rozsiahlejšia, ako sme si ukázali v tomto module. Je to však veľmi dobrý základ na pokračovanie v tejto oblasti. Predstavili sme si nielen základné vlastnosti, ale aj niektoré z pokročilejších. Nezostal nám však priestor napr. na umiestňovanie objektov. Práve vytváranie dizajnu stránok len pomocou kaskádových štýlov s využitím umiestňovania objektov je aktuálnym trendom tvorby webových stránok. Toto použitie kaskádových štýlov je najnáročnejšie.

Majstrovstvo použitia kaskádových štýlov s jedným zdrojovým XHTML kódom a viacerými štýlmi (úplné oddelenie obsahu od dizajnu) nájdeme na stránke www.csszengarden.com. Na obrázkoch 32 a 34 sú zobrazené dve rôzne webové stránky, ktoré používajú identický zdrojový XHTML kód, no líšia sa použitými kaskádovými štýlmi. Môžeme na nich vidieť, že sa zmenil nielen dizajn, ale aj celé rozloženie objektov.



Obrázok 33 - Stránka s dizajnom vytvoreným pomocou kaskádových štýlov



Obrázok 34 - Stránka s identickým XHTML kódom ako Obrázok 33, ale iným kaskádovým štýlom

Tvorba pokročilých webových stránok

V tomto module sme vytvárali tzv. statické webové stránky, to znamená, že na serveri existuje finálna „verzia“ stránky, teda to, čo je uložené na serveri, sa presne zobrazí používateľovi vo webovom prehliadači. Tento prístup však dnešnému dynamickému svetu nestačí - informácie sa neustále menia, pribúdajú. Pri dynamicky sa meniacich informáciách treba vytvárať tzv. dynamické stránky.

Dynamická stránka neexistuje vo finálnej podobe na serveri, ale vzniká (generuje sa) až pri požiadavke používateľa o jej zobrazenie. Vtedy sa vykonajú všetky potrebné činnosti a vygeneruje sa výsledná stránka, ktorá sa zobrazí vo webovom prehliadači. Takto sa vytvárajú stránky rôznych spravodajských serverov, internetové bankovníctva, diskusné fóra atď. Takéto dynamické stránky už treba programovať v niektorom programovacom jazyku, napr. PHP, ASP, C#.

Čo sme sa naučili v tomto module

Cieľom tohto modulu bolo naučiť sa rozumieť zdrojovému kódu webových stránok, vytvárať webové stránky písaním zdrojového kódu v jazyku XHTML a definovať dizajn stránok pomocou kaskádových štýlov.

Oboznámili sme sa so základnými elementmi jazyka XHTML, pomocou ktorých môžeme štruktúrovať obsah webových stránok, vkladať obrázky, zoznamy, odkazy, tabuľky a špeciálne znaky. V druhej časti modulu sme si predstavili kaskádové štýly, pomocou ktorých sa v súčasnosti vytvára dizajn webových stránok. Pomocou kaskádových štýlov sme zmenili farby elementov, rôzne nastavenia písma, zarovnanie textov, orámovanie a priestory okolo elementov. Definovali sme tiež vlastné triedy, pseudotriedy a aj štýly pre iné zariadenia.

Zhrnutie

Účastník vzdelávania po úspešnom absolvovaní tohto modulu dokáže vytvoriť niekoľko webových stránok v jazyku XHTML s využitím kaskádových štýlov na definovanie dizajnu. Stránky budú vhodne naformátované a budú mať jednotný dizajn definovaný v spoločnom kaskádovom štýle. Účastník dokáže vložiť obrázky, odkazy, vhodne štruktúrovať obsah pomocou blokov, prípadne tabuliek. Dokáže vytvoriť kaskádové štýly pre rôzne zariadenia. Všetky stránky budú mať korektný zdrojový kód podľa štandardu XHTML 1.0.

Preverenie výstupných vedomostí

Preverovanie vedomostí prebieha priebežne. Účastníci riešia úlohy, ktoré preukážu ich zručnosti, a tiež sa zapájajú do diskusií.

Literatúra a použité zdroje

- [1] Cederholm D.: *Flexibilní webdesign*, Computer Press, 2006
- [2] Cederholm D.: *Webdesign s webovými standardy*, Zoner Press, 2004
- [3] Chuck M., Kennedy B.: *HTML a XHTML kompletní průvodce*, Computer Press, 2000
- [4] Meyer E.: *Eric Meyer o CSS - ovládněte kaskádové styly!*, Zoner Press, 2004
- [5] Prokop M.: *CSS pro webdesignéry*, Mobil Media a.s., 2003
- [6] Zeldman J.: *Tvorba webů podle standardů*, Computer Press, 2005
- [7] *World Wide Web Consortium (W3C)* [online]. Dostupné na internete: <www.w3.org>
- [8] *W3Schools Online Web Tutorials* [online]. Dostupné na internete: <www.w3schools.com>

Tento študijný materiál vznikol ako súčasť národného projektu Ďalšie vzdelávanie učiteľov základných škôl a stredných škôl v predmete informatika v rámci Aktivity „Vzdelávanie nekvalifikovaných učiteľov informatiky na 2. stupni ZŠ a na SŠ“.

Autori © PaedDr. Roman Hrušecký, PhD.
PaedDr. Daniela Bezáková, PhD.

Názov Ďalšie vzdelávanie učiteľov základných škôl a stredných škôl v predmete informatika

Podnázov Internet: princípy a tvorba webu 3

Študijný materiál prešiel recenzným pokračovaním.

Recenzenti RNDr. František Galčík, PhD.
Ing. Radoslav Gargalík

Počet strán 40

Náklad 300 ks

Prvé vydanie, Bratislava 2011

Všetky práva vyhradené.

Toto dielo ani žiadnu jeho časť nemožno reprodukovat' bez súhlasu majiteľa práv.

Vydal Štátny pedagogický ústav, Pluhová 8, 830 00 Bratislava, v súčinnosti s Univerzitou Pavla Jozefa Šafárika v Košiciach, Univerzitou Komenského v Bratislave, Univerzitou Konštantína Filozofa v Nitre, Univerzitou Mateja Bela v Banskej Bystrici a Žilinskou univerzitou v Žiline

Vytlačil BRATIA SABOVCI, s r.o., Zvolen

ISBN 978-80-8118-078-1