

Jednoduchý návod, jak psát v XHTML validní webové stránky

Účel této stránky:

Tento spis si nedělá žádné ambice na odbornost a striktnost, je určen pro naprosté začátečníky v tvorbě webu a měl by je nasměrovat tak, aby se od začátku učili pracovat v prostředí, odpovídajícím roku 2004 a nezabíjeli drahocenný čas výukou archaického HTML 3.2 a 4.0 nebo aby se nenaučili zlovykům tabulkových rozmístění obsahu stránky nebo psaní formátu stránky (myšleno barvy, align, bordery, odstupy) do kódu. Naučit se základům CSS.

Co bychom měli znát úvodem:

1. Co je to HTML a XHTML

Obecně a zjednodušeně lze říci, že internetové prezentace jsou ve většině případů psány v tzv. značkovacích jazycích. Značky neboli tagy jsou ve skutečnosti příkazy, které dávají webovému prohlížeči najevo, jak má tento prohlížeč zobrazit danou textovou, grafickou nebo jinou (objekty) pasáž prezentace nebo kam (na jaký prohlížený web, soubor, objekt, místo) má prohlížeč přejít. HTML je zkratka, vytvořená z prvních písmen HYPER TEXT MARKUP LANGUAGE – - hypertextový značkový jazyk.

hyper text je technologie, která umožňuje rychlé odskakování. Obsahuje start a cíl. Hyper textový odkaz poznáme v dokumentu (jakémkoli dokumentu, tedy i v internetové prezentaci) podle aktivního chování. Při akci mouse over - přejedeme-li myší nad takový odkaz – pointer myši se změní zpravidla v symbol ručičky. Hypertextový odkaz je nejčastěji barevně odlišený od jiného textu a bývá podtržený). Při akci mouse-click (jedno kliknutí levým tlačítkem myši) dojde k uskutečnění hypertextového odskoku do cíle, kam hypertextový odskok míří. To může být kamkoli do Internetu, nebo do jiných souborů naší internetové prezentace.

Ukázka hypertextu: tento odskok přeskočí na první řádek tohoto učebního textu. Jiný hypertextový odkaz umí například spustit e-mailového klienta:

2. HTML - jazyky

Tento jazyk se vyvíjí s nejméně dvacetiletou historií a jeho vývoj prodělal mnoho zvrátů.

Dodnes je několik verzí, ještě běžné verze jsou HTML 3.2, HTML 4.0., poslední verze HTML jazyka je HTML 4.01. Touto verzí skončil vývoj jazyka HTML a další vývoj již patří XHTML.

Během vývoje do jazyka (od verze 1.0 do verze 4.0) byly dávány různé doplňky a nové= sributy tagů. Jejich používáním postupem času vznikla džungle, takže ne každý prohlížeč si rozumí s každou formou jazyka od verze 1.0 do 4.0. Rovněž je třeba rozlišovat deklaraci jazyka.

Jestliže se snažíme o první stránku, je třeba si alespoň hrubě ujasnit, v jakém jazyce píšeme. Pamatujme, že pokud námi napsanou stránku vyvěsíme na Internet, její kód by měl být napsán správně, odpovídat deklaraci jazyka

HTML 4.0 a HTML 4.01 známe již jako STRICT, Transitional, Frameset

3. XHTML 1.0

XHTML znamená „Extensible Hypertext Markup Language“. Je nejnovější jazyk, v roce 2004 je běžně nasazen ve třech deklaracích. První XHTML 1.0 je opět jako STRICT, další jsou Transitional, a Frameset. Co jednotlivé deklarace znamenají ?

1. Deklarace STRICT – striktní jazyk je nejpřísnější deklarací. Neobsahuje např. rámy, neobsahuje tagové atributy <target>, což dosti omezuje webmastery, neboť se to musí obejít javascripty.

2. Deklarace TRANSITIONAL – přechodný jazyk je naopak nejméně přísná.

3. Deklarace FRAMESET je pro stránky, obsahující rámy (frames). Ovšem začátečníci, vezměte si na srdce, že technologie ráků je dnes již překonaná a rámy by se neměly využívat, zejména s ohledem na to, že dělají těžkosti u vyhledávačů, u wapu.

XHTML 1.1 je již jednou z forem platformy XML. V podstatě lze zjednodušeně říci, že je totožný s deklarací jazyka XHTML1.0 Strict. Jde o velmi přísný přístup, který se zaměřuje na zobrazení obsahu webu. Forma zobrazení je ve vlastním dokumentu XHTML druhotná. Přesto lze bez ráků a bez tabulkového rozmístování (začátečníci – ani se o to nezajímejte) vytvořit velmi kvalitní, transparentní a vzhledově jednotné stránky.

Začínáme s vytvářením jednoduché webové prezentace.

PROČ JE NUTNO ZNÁT ZÁKLADY HTML resp. XHTML ?

Námítka, že webovou prezentaci lze vytvořit pouhým exportem dokumentu z wordu, open-office (asi nejlepší) nebo jiného editoru, či použít nějaký klikací wisiwyg editor je od začátku mylná. V 99% případů tyto editory vytvoří velmi složitý kód, která je v 99% nevalidní, většinou dodnes je zde použit starý HTML 4.0. Stránky potom obsahují „více kódu než obsahu“, zadržávají ve vyhledávačích a velmi dlouho se natahují i v běžných prohlížečích. Proto doporučuji postupovat cestou použití textového editoru. Mě se nejlépe osvědčil FirstPage 2000 (freeware). Ten sice umí jen HTML 4.0, ale jeho text je velmi přehledný a dobře strukturovaný. Stejnou službu vykoná i ACE HTML 5.0 (freeware), který již umí i XHTML 1.1. Stejně dobře lze k tvorbě použít i freewarovou verzi W3C HTML Validatoru 3.0

NEŽ ZAČNEME

Než se pustíte do první stránky, stáhněte si off-line verzi internetové učebnice Jiřího Koska „Jak psát web“ na www.jakpsatweb.cz. Prostudujte si stránky www.sovavsiti.cz a prohlédněte si odborné články www.vitdlouhy.cz a rovněž si alespoň přečtěte prohlášení webmasterů – tzv dogma W4D. Rovněž si stáhněte off-line učební text z www.jsi.cz/stranky. Po přečtení úvodní filosofie doporučuji několika denní přestávku na setřesení přečteného obsahu, neboť začátečníkovi se toto může zdát jako nedostižné. Je však věci zkušenosti, že během kurzu (nebo řádné výuce) pod vedením lektora lze znalosti velmi rychle zvládnout.

JAK TEDY POSTUPOVAT CESTOU VALIDNÍHO KÓDU ?

Doporučuji držet se plně prohlášení W4D. Z prohlášení W4D vyplývají zásady, které lze shrnout do několika vět:

Citace českého překladu zásad W4D –

(podotýkám, že nejsem autorem následujících 13 zásad. že je cituji z webu, který uvádím jako zdroj:)

Prohlášení W4D zdroj:

<http://www.pixy.cz/dogma/dogmaw41/cs/index.html>

O pravidlech tvorby hodnotného a všeobecně přístupného webu bylo napsáno mnoho článků a publikací. Jako autoři jsme si vědomi, že jeden příklad je lepší než tisíce knih. Presentace, která se hlásí k dodržení Dogmatu W4, proto respektuje [Zásady W4D](#).

Tyto zásady nejsou pro kvalitní web nezbytné, jejich dodržování ale snižuje rizika omezení jeho přístupnosti a použitelnosti.

Zásady W4D

Pozn.: U každého bodu je uveden odkaz na detailní popis, vysvětlení důvodů, proč byla uvedena pravidla použita a jaké výhody jejich dodržení přináší.

1. [Kód dokumentu](#)

Kód dokumentů bezchybně vyhovuje striktnímu HTML 4.01 nebo XHTML. Současně zohledňuje zpětnou kompatibilitu i upřednostňuje dopředně kompatibilní syntaxi.

2. [Záhlaví dokumentu](#)

Záhlaví obsahuje stručný a výstižný titulek, jméno a e-mail autorů kódu a kódování dokumentu, pokud se liší od výchozího kódování.

3. [Struktura dokumentu](#)

Autor musí zajistit, že dokument a jeho kód věrně odrážejí sémantiku a strukturu obsahu. Hlavnímu

sdělení dokumentu smí předcházet pouze informace, které se čtenář má dozvědět, než toto sdělení začne číst. Tělo dokumentu nesmí obsahovat značky, které nenesou sémantickou nebo strukturální informaci, ani obsah, který má pouze prezentační charakter.

4. Navigace

Každý dokument v prezentaci vyjma úvodní strany musí obsahovat odkaz na úvodní stranu. Všechny další části dokumentu určené k navigaci musí být prezentovány a umístěny ve všech dokumentech shodně. Delší dokumenty musí být rozděleny do sekcí, odkazovaných z prologu dokumentu.

5. Úvodní strana

Úvodní strana celé prezentace musí zřetelně popisovat cíl a sdělení této prezentace. Viditelně odkazuje kontakt na administrátora.

6. Sdělení dokumentu

Každá informace sdělovaná dokumentem musí být dostupná v textovém formátu.

7. Odkazy

Všechny odkazy a další aktivní prvky musí jasně popisovat svůj cíl a akci, kterou jejich aktivace způsobí. Odkazy musí zůstat podtrženy, pokud jejich podtržení nezakáže uživatel. Přeformátování dokumentu smí způsobit pouze aktivace nějakého prvku.

8. Tabulky

Tabulkové prvky mohou být v dokumentu použity k rozmístění a formátování tabulkových dat, nikdy k definování vzhledu dokumentu. Tabulky musí být přístupné bez ohledu na použitý způsob prezentace (vyjma případů kdy koncové zařízení tabulky vůbec nepodporuje), čitelné a jejich obsahu musí předcházet výstižný nadpis a nebo shrnutí.

9. Použití dokumentu

Obsah ani kód dokumentu nesmí předjímat, doporučovat ani vyžadovat konkrétní způsob použití. Přístupnost obsahu dokumentu nesmí žádnému způsobu prezentace dávat přednost.

10. Doplňky dokumentu

Kód, ovlivňující prezentaci dokumentu, musí být umístěn v externích souborech, připojených k dokumentu v jeho záhlaví. Korektní prezentace dokumentu nesmí záviset na nějakém doplňku (nebo jeho části), který může být uživatelem zakázán či odinstalován, nebo který po uživateli vyžaduje instalaci softwaru, který není v jeho prostředí právě dostupný. Kód dokumentu ani jeho doplňky nesmí měnit nebo nebrat ohled na uživatelské předvolby.

11. Barvy

Barvy v dokumentu musí být definovány s ohledem na dostatečný kontrast jasu a odstínu popředí a pozadí. Na pozadí nesmí být použit vzorek, který jakkoli snižuje čitelnost obsahu.

12. Text

Kolem každého textu musí být zřetelný prostor. Velikost písma může být definována pouze absolutně nebo relativními jednotkami, žádné písmo nesmí být definováno menší než `xx-small`. Definice typu písma musí obsahovat položku, která je zaručeně dostupná na každém zařízení (to je obvykle zajištěno obecnou rodinou písma).

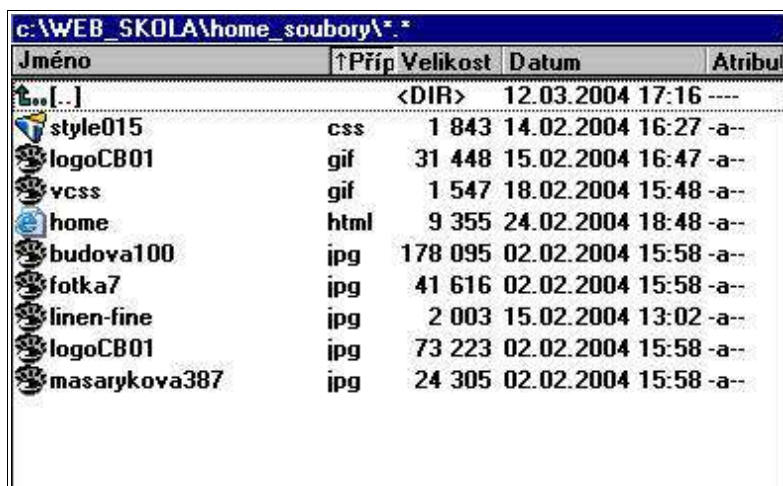
13. Autoři

Autoři dodržující tyto zásady je mohou uvést a/nebo odkázat ve svých dokumentech.

STRUKTURA DOKUMENTU HTML A RELATIVNÍ ADRESOVÁNÍ SOUBORŮ.

PRO STRUKTURU SOUBORŮ DBÁME:

- všechny soubory pojmenováváme malými písmeny bez diakritiky
- všechny adresáře pojmenováváme malými písmeny bez diakritiky a bez přípon – viz obr.:



Jméno	Příp	Velikost	Datum	Atribul
[..]	<DIR>		12.03.2004 17:16	---
style015	css	1 843	14.02.2004 16:27	-a--
logoCB01	gif	31 448	15.02.2004 16:47	-a--
vcss	gif	1 547	18.02.2004 15:48	-a--
home	html	9 355	24.02.2004 18:48	-a--
budova100	jpg	178 095	02.02.2004 15:58	-a--
fotka7	jpg	41 616	02.02.2004 15:58	-a--
linen-fine	jpg	2 003	15.02.2004 13:02	-a--
logoCB01	jpg	73 223	02.02.2004 15:58	-a--
masarykova387	jpg	24 305	02.02.2004 15:58	-a--

- abychom se v HTML prezentaci vyznali, vytvoříme si strukturu adresářů, do každého adresáře si ukládáme soubory daných témat a skupin k jednotlivým odkazům včetně obrázků – viz obrázek výpisu vnitřku jednoho z adresářů.

Jméno	↑ Příp	Velikost	Datum	Atribul
⋮			12.03.2004 17:16	---
[aktual]			12.03.2004 17:23	---
[css]			12.03.2004 17:23	---
[css1]			12.03.2004 17:23	---
[download]			12.03.2004 17:23	---
[galerie]			12.03.2004 17:23	---
[historie]			12.03.2004 17:23	---
[home_soubory]			12.03.2004 17:23	---
[mailform]			12.03.2004 17:23	---
[maturity]			12.03.2004 17:23	---
[menu]			12.03.2004 17:23	---
[obory]			12.03.2004 17:23	---
[obrazky]			12.03.2004 17:23	---
[praxe]			12.03.2004 17:23	---
[prijim]			12.03.2004 17:23	---
[profesori]			12.03.2004 17:23	---
[rada_rod]			12.03.2004 17:23	---
[sipvz]			12.03.2004 17:23	---
[vyrocka]			12.03.2004 17:23	---
index	html	698	16.02.2004 18:02	-a-

d) v kořenovém adresáři webu ponecháme jenom soubor index. html

e) soubory doporučuji pojmenovávat příponou *.html, nepoužívat příponu *.htm.

f) jako příklad sturktury adresářů je možno uvést takovouto konkrétní strukturu:

g) k hypertextovým vazbám používáme výhradně relativní adresování. To nám přinese výhodu, že celou prezentaci webu můžeme nahrát na jakýkoli server bez úprav. Rovněž můžeme prezentaci komukoli dát např. na CD ROM jako off-line verzi našeho webu.

Relativní adresování - základy

1. příklad: Soubor index obsahuje obrázky, které jsou umístěny v adresáři [album]. Jestliže vkládáme obrázek do index.html, musíme je adresovat takto:

```

```

Jméno	Příp	Velikost	Datum	Atribul
⋮			12.03.2004 17:16	---
[album]			12.03.2004 17:19	---
[clanky]			12.03.2004 17:19	---
[disertace]			12.03.2004 17:19	---
[home]			12.03.2004 17:19	---
[kaskada]			12.03.2004 17:19	---
[menu]			12.03.2004 17:19	---
[prace]			12.03.2004 17:19	---
pass	html	4 241	11.02.2004 14:43	-a-
index	html	435	15.02.2004 11:18	-a-

2. příklad: Soubor home.html je umístěn v adresáři [home]. Tento soubor obsahuje obrázek, který je opět umístěn v adresáři [album]. Nani jej adresujeme následovně: ../album/ tedy jednoduše řečeno: skoč do kořenového adresáře, najdi adreář album a vezmi z něho soubor obraz 1.html. Příklad:

```

```

3. příklad: Soubor album.html je umístěn v adresáři [album]. Tento soubor obsahuje obrázek obraz1.jpg, umístěný ve vlastním adresáři. Nyní na obrázek odkazujeme následovně:

```

```

4. a tak dále, jestliže bychom odkazovali na obrázek o dva podadresáře níže, zápis by vypadal asi takto:

```

```

HTML DOKUMENT A JEHO PODOBA

a) Abychom vyhověli modernímu pojetí, vystačíme si i s menším množstvím základních tagů než tomu bylo dříve. Základní dokument totiž strukturujeme jen do nadpisů různé úrovně – H1 až H6, do odstavců <p>, do bloků <div>, případně některé pasáže zvýrazníme .

b) Prvním krokem k webu je analýza. Bedlivě si rozmyslíme, jak obsah webu rozdělíme do dílčích bloků, oddělení, jakou strukturu bude mít nabídka (menu) webu. Je vhodné si graficky naši stránku vyprojektovat na kus papíru !! Vždy myslíme na budoucnost a možnost rozšíření našeho webu, myslíme i na to, že v budoucnu změnou css budeme chtít změnit design webové prezentace. V jednoduchosti a stručnosti je síla.

c) Ve druhé fázi píšeme vlastní html dokument. Píšeme jej bez jakýchkoli stylů, jen „černo-bílé zobrazení“, umístíme obrázky a tabulky na svá místa, počítáme předem s budoucím rozvržením stránek Dokument, který vytvoříme ve striktním jazyce na první pohled v grafickém prohlížeči nevypadá hezky, je jen černobílý, umístění obrázků nám zpravidla nesedí tak, jak chceme. To je ale zcela v pořádku, protože dokument je strukturován podle svého obsahu, má obsahovou vzpovídacé hodnotu. Vzhled dokumentu je také důležitý, ale ten vytvoříme takzvaným stylopisem pomocí kaskádového stylu. Kaskádový styl – CSS- je nejčastěji (v moderním pojetí) zvláštní soubor, který obsahuje příkazy, jak má prohlížeč zobrazovat jednotlivé tagy (zde jim říkáme selektory).

d) Tabulky dle W4D používáme jenom na zobrazení tabelárních údajů, nikdy ne k rozmístění obsahu stránky. Pro ty, kteří jsou „zkaženi tímto zlozvykem“ ze zkušenosti říkám, že to jde !!!

e) Nejlepší je během psaní dokumentu stále prověřovat, zda je prostý syntaktických chyb. např. projetím v HTML validátoru

JAK UMÍSTIT NA WEBU HOTOVÉ DOKUMENTY, KTERÉ BY ŠLY JEN OBTÍŽNĚ PŘEVÉST DO HTML ?

U grafických dokumentů nebo rozsáhlejších textů, plných obrázků a jiných objektů je nejlepší takové dokumenty převést do formátu *. pdf (PORTABLE DOKUMENT FILE). PDF soubor pak dát na hypertextový odkaz ke stažení formou např. zazipovaného pdf souboru. To je asi nejuniverzálnější řešení dnešní doby. Je možno použít i postscriptové soubory, ale s těmi si běžný uživatel nemusí vždy dobře poradit. PDF vytvoříme velmi jednoduše např. v OpenOffice 1.1.0, který má nativní pdf převaděč. Pak na takový dokument odkážeme asi takto:

První způsob:

```
<a href="../../pdf/muj_tutorial.zip">soubor ke stažení v zazipovaném formátu pdf <a>
```

Tento první způsob se mi jeví lepší než druhý.

Druhý způsob:

U druhého způsobu totiž u naprosté většiny počítačů s windows a nainstalovaným Acrobat Readerem se při stahování automaticky spustí AcrobatReader a uživatel musí čekat na stažení tak velké části souboru, než dojde k zobrazení. Proto je lepší pdf-kový soubor zazipovat. Při aktivaci odkazu se pak nabídne dialog na stažení, uživatel si stanoví cílové umístění ukládaného souboru a po odpojení si jej může v klidu rozbalit a otevřít. Maje na paměti, že zip formát je celosvětově běžná záležitost.

příklad:

```
<a href "../pdf/muj_tutorial.pdf">soubor ke stažení ve formátu pdf <a>
```

Je ještě třetí, pracnější způsob, zato jeho výsledek bývá asi nejlepší z uvedených – viz dále.

VLASTNÍ STRUKTURA DOKUMENTU

Každý (X)HTML dokument, musí být jasně definován, aby bylo jasné, podle jakého standardu byl vytvořen a musí být deklarován. Níže uvádím deklarace HTML 4.01 a XHTML1.0 Strict a XHTML : 1.1:

Příklad dokumentu HTML 4.01 Strict

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
```

```
<html>
```

```
<head>
```

```
<title>titulek zobrazený v titulkovém pruhu prohlížeče</title>
```

```
<meta http-equiv="Content-Type" content="text/html; charset= "windows-1250" >
```

```
<title>Moje prezentace</title>
```

```
<meta name="Description" content="Výukový text zásad psaní webových stránek" >
```

```
<meta name="Keywords" content="klíčová slova web jazyk html xhtml struktura dokumentu ">
```

```
<meta name="author" content="Vladimír Kebert mailto:analytic@seznam.cz" >
```

```
<link rel="stylesheet" type="text/css" href="style01.css" >
```

```
</head>
```

```
<body>
```

tělo dokumentu

```
</body>
```

```
</html>
```

Příklad dokumentu 4.01 Transitional

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>

<title>titulek zobrazený v titulkovém pruhu prohlížeče</title>
<meta http-equiv="Content-Type" content="text/html; charset= "windows-1250" >
<title>Moje prezentace</title>
<meta name="Description" content="Výukový text zásad psaní webových stránek" >
<meta name="Keywords" content="klíčová slova web jazyk html xhtml struktura dokumentu " >
<meta name="author" content="Vladimír Kebert, mailto:analytic@seznam.cz" >
<link rel="stylesheet" type="text/css" href="style01.css" >

</head>

<body>

tělo dokumentu

</body>
</html>
```

Příklad dokumentu HTML 4.01 FRAMESET

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN">
<html>
<head>

<title>titulek zobrazený v titulkovém pruhu prohlížeče</title>
<meta http-equiv="Content-Type" content="text/html; charset= "windows-1250" >
<title>Moje prezentace</title>
<meta name="author" content="Vladimír Kebert mailto:vk@seznam.cz" >
<link rel="stylesheet" type="text/css" href="style01.css" >

</head>

<frameset cols="125,*" >
  <frame src="menu/menu.html" name="menu" scrolling="auto" >
  <frame src="home/home.html" name="value" scrolling="auto" marginheight="30"
marginwidth="10" >
</frameset>

dokument nemá body !!!!

</html>
```

Příklad dokumentu XHTML1.0 STRICT

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

<html>
<head>
  <title>titulek zobrazený v titulkovém pruhu prohlížeče</title>
  <meta http-equiv="Content-Type" content="text/html; charset= "windows-1250" />
  <meta name="author" content="Vladimír Kebert mailto:vk@seznam.cz" />
  <meta name="Keywords" content="klíčová slova web jazyk html xhtml struktura dokumentu " />
  <link rel="stylesheet" type="text/css" href="sytle01.css" />
</head>

<body>

tělo dokumentu

</body>
</html>
```

Příklad dokumentu XHTML1.1

```
<?xml version="1.0" encoding="windows-1250"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
  "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">

<head>
<meta http-equiv="Content-Type" content="text/html; charset=windows-1250" />
<title>titulek zobrazený v titulkovém pruhu prohlížeče</title>
<meta name="author" content="Vladimír Kebert mailto:analytic@seznam.cz" />
<meta name="Keywords" content="klíčová slova web jazyk html xhtml struktura dokumentu " >
<link rel="stylesheet" type="text/css" href="sytle01.css" />
</head>
<body>
tělo dokumentu
</body>
</html>
```

PRAVIDLA PSANÍ V XHTML:

Asi jste si všimli, že u HTML 4.01 končí nepárové tagy `<meta>` jenom bez lomítka. Zásadní rozdíl mezi HTML a XHTML je v tom, že v XHTML musí být :

a) všechny tagy vypisovány malými písmeny

b) všechny nepárové tagy musí být ukončeny, takže jestliže zalomení a odskok na další řádek vypadá v HTML takto, `
` v XHTML jej musíme ukončit `
`.

c) všechny hodnoty atributů tagů musí být v uvozovkách – tedy co bylo v HTML např.:
`` musí být v XHTML zapsáno takto:
``

d) kdybychom v XHTML psali `
</br >`, je z hlediska validity tento zápis ekvivalentní `
`, ale v prohlížečích by to mohlo dělat neplechtu, tak se tento postup nedoporučuje, (ostatně není pro něj žádný důvod)

JAK PŘEVÉST NĚKOLIKASTRÁNKOVÝ DOKUMENT Z FORMÁTU TEXTOVÉHO PROCESORU (*.doc , *.sxw, *.rtf a jakékoli jiné) DO (X)HTML ?

a) jak jsem výše uvedl, je možné takový soubor exportovat rovnou do HTML pomocí vnitřního převaděče našeho textového procesoru. V 90 % při kontrole validátorem vyjde, že kód není validní. Kód vygenerovaný takovýmto způsobem je obvykle velmi složitý s nadbytkem formátovacích atributů.

b) Proto dle osobní zkušenosti doporučuji následující postup, který se mi nejlépe osvědčil a jeho zdánlivá pracnost vyváží jeho kvalita.

1. Naš dokument uložíme jako prostý text (pozor na kódování češtiny – používejte WINDOWS 1250 nebo ISO 8859-2 v Linuxu)

2. Napíšeme si kaskádový stylpis jako samostatný soubor – doporučuji TopStyle Light (freeware) nebo též ACE-HTML5.0 (také freeware)

3. Vytvoříme si (dle popisu výše) adresářovou strukturu, vypreúarujeme si obrázky, v klidu je upravíme na webové parametry (obvykle 320*240 i méně a 256 barev)

4. Založíme si (X)HTML strukturu – deklarace, metatagy, link, head, body a do body si nakopírujeme pomocí schránky celý náš prostý text.

5. Naš dokument postupně „otágujeme“, znovu vložíme upravené obrázky na správná místa.

6. Dokument s největší pravděpodobností bude vzhledově odlišný od originálu, ale bude validní a opatřený jednotným vzhledem.



pokračování příště: