

Novinky v HTML5 a CSS3

1. díl seriálu

VLASTISLAV KUČERA

Pedagogická fakulta Univerzity Hradec Králové

V tomto seriálu se seznámíme s novinkami, které se objevily v jazycích HTML5 a CSS3. I když ještě nejsou oba jazyky uznány jako standard pro publikování obsahu na webu – v současné době jsou standardy jazyky HTML4.01, XHTML1 a CSS2.1 – tvůrci / výrobci prohlížečů však již delší dobu, cca 2 roky, nové prvky jazyků HTML5 a CSS3 implementují do verzí svých prohlížečů. O tom, jak je na tom s podporou HTML5 a CSS3 prohlížeč, který právě používáme, můžeme zjistit pomocí [1], [3]. Na [2] nalezneme přehled nových vlastností a jejich podporu v jednotlivých verzích prohlížečů.

Vymezení pojmů

Protože se v literatuře vyskytují rozdíly v pojmenování základních komponent jazyka HTML, považuji za potřebné uvést pojmenování, které budu v následujícím textu používat.

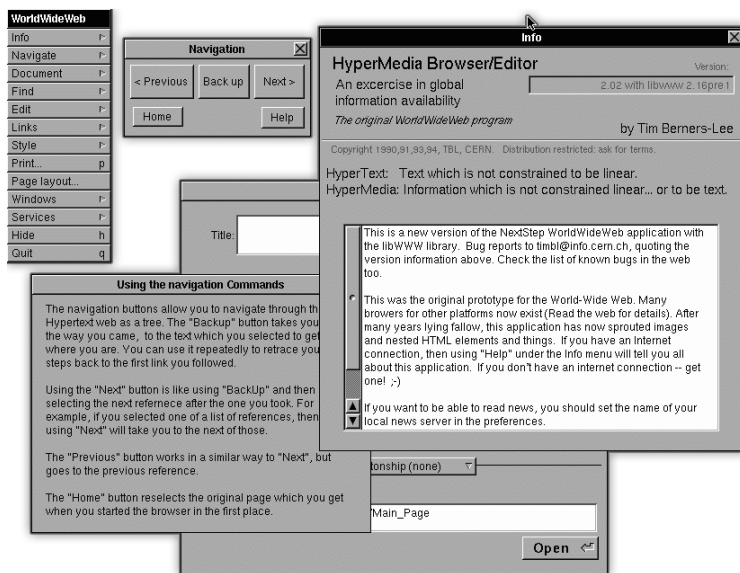
Prvek – základní komponenta jazyka HTML. Můžeme se také setkat s pojmenováním element. Prvek se skládá z počáteční značky, např. `<p>`, obsahu a koncové značky, např. `</p>`. Některé prvky nemusí mít obsah, v takovém případě mluvíme o prázdném prvku, např. ``. Co se má zobrazit pomocí prázdného prvku, je dáno parametry počáteční značky.

Značka neboli tag. Značky dělíme na párové a nepárové. Párové značky mají počáteční (`<p>`) a koncovou (`</p>`) značku, nepárové jenom počáteční (``, `
`).

Parametr – můžeme se také setkat s pojmenováním atribut. Hodnotu, kterou může parametr nabývat, budu označovat hodnotou parametru.

Stručně o HTML

Historie HTML se začala psát počátkem 90. let 20. století. V první verzi nebyl podporován grafický režim. Jinak řečeno, stránky vypadaly jako textové soubory. Na obr. 1 je vidět, jak zobrazoval www stránky první webový prohlížeč WorldWideWeb (<http://cs.wikipedia.org/wiki/WorldWideWeb>). Tento prohlížeč byl později přejmenován na Nexus.



Obr. 1

Až později, kdy rostla popularita webu, byla do nových verzí přidávána podpora grafiky, formulářů, tabulek, zarovnávání textu a prvky pro definici vzhledu stránky.

Skutečnou revoluci v tvorbě stránek znamenal konec roku 1997, kdy byla vydána 4. verze jazyka HTML. V této verzi se poprvé mluví o oddělení významu (obsahu) od vzhledu. Vzhled má být pro příště definován přípojitelnými styly. Tak se dostal do popředí jazyk CSS. Proto také ze 4. verze jazyka HTML byly vyřazeny prvky, které definovaly vzhled jednotlivých částí stránky.

V roce 2007 se začalo pracovat na nové verzi jazyka HTML, na verzi HTML5. Její specifikace byla dokončena v roce 2012. Ukončení specifikace a standardizace jazyka HTML5 se předpokládá kolem roku 2020 (některé zdroje mluví o roku 2022).

Stručně o XHTML

Jazyk XHTML vznikl v roce 2000 jako nástupce jazyka HTML4.01. XHTML je vlastně kombinací jazyka HTML a obecného značkovacího jazyka XML. XHTML byl navržen tak, aby vyhovoval podmínkám tvorby XML dokumentů, ale aby přitom byla zachována zpětná kompatibilita s jazykem HTML.

XML je obecný značkovací jazyk, umožňující snadné vytváření konkrétních značkovacích jazyků. XML je hlavně určen pro výměnu dat mezi aplikacemi. Dále pak pro publikování dokumentů, kde je třeba popsat strukturu z hlediska obsahu jednotlivých částí. Při tvorbě dokumentu pomocí jazyka XML si můžeme definovat vlastní značky, např. `<auto>`, `<značka>`. To, jak se zobrazí v prohlížeči, pak definujeme pomocí kaskádových stylů.

Při vytváření XML dokumentu ale musíme dodržovat jistá pravidla: XML v názvech značek rozlišuje malá a velká písmena (v XML `<Auto>` není totéž co `<auto>`) všechny prvky musí mít počáteční a koncovou značku (`<auto>` `</auto>`, případně prázdná značka `<auto />`), hodnoty parametrů musí být v uvozovkách.

Jazyk XHTML je vlastně aplikací výše uvedených pravidel do jazyka HTML: značky musí být uvedeny malým písmenem (`<H1>` je v XHTML nepřipustné, správný zápis je `<h1>`), všechny značky musí být ukončeny (např. značka `` v HTML je v XHTML definována ``), hodnoty parametrů musí být v uvozovkách.

Stručně o CSS

Jazyk CSS oficiálně vznikl koncem roku 1996. Jeho rozšíření mezi tvůrce však bránila nepříliš velká podpora v prohlížečích. Opravdový význam jazyka CSS nastal až po vydání 4. verze jazyka HTML, ve kterém, jak je uvedeno výše, nahrazuje formátovací elementy jazyka HTML. Pomocí jazyka CSS můžeme definovat např. barvu textu, velikost textu, styl textu, barvu pozadí, rozestupy mezi jednotlivými prvky, rozmístění prvků apod.

Druhá verze jazyka CSS byla vydána v roce 1998, která je po malé aktualizaci platná ve verzi 2.1. Tato verze je, stejně jak je to u verzí jazyka HTML, rozšířením předchozí verze.

HTML5

Jazyk HTML5 v sobě kombinuje, stejně jako u předchozích verzí, stávající prvky, které jsou obsaženy ve verzi 4, případně jsou stávající prvky předefinovány, a přidává k nim nové prvky, např. `article`, `section`, `header` a další.

CSS3

CSS3 je také, jako u jazyka HTML5, v podstatě rozšíření stávajícího jazyka CSS2.1. CSS3 přidává k již definovaným prvkům nové, které výrazně ulehčí práci vývojářům. Ve verzi 3 jsou nově definovány např. vlastnosti pro definici zaoblených rohů, přechodů, více obrázků na pozadí apod.

Podpora HTML5 a CSS3 v prohlížečích

I když jsem v úvodu poznamenal, že tvůrci prohlížečů do posledních verzí implementují nové prvky jazyků HTML5 a CSS3, nemusíme mít obavu, že starší prohlížeče stránky nezobrazí nebo že by prohlížeče přestaly pracovat. Prohlížeč zobrazí prakticky jakoukoliv značku. Pokud použijeme vlastní značku v HTML dokumentu a pomocí jazyka CSS ji nastavíme – nastavíme vlastnosti – prohlížeč na novou značku aplikuje definované styly, které „umí“ zpracovat, a zobrazí ji. Podobné je i při použití HTML5. Starší verze prohlížečů, které nemají implementovanou podporu HTML, nové značky jazyka HTML5 „berou“ jako značku, kterou vytvořil tvůrce stránky a pokud taková značka má definované vlastnosti pomocí CSS, nemá prohlížeč problém s jeho zobrazením.

Stejně jako v reálném životě, tak i mezi prohlížeči existuje výjimka. Tou výjimkou je Internet Explorer do verze 9, tj. Internet Explorer verze 5–8. Tyto verze byly naprogramovány, přesněji řečeno jejich vykreslovací jádra, tak, aby prvky jazyka HTML, které neznají, nezobrazovaly a aby takovým prvkům nebylo možné přiřadit styly. V těchto verzích se tedy bez „cizí“ pomoci nové prvky jazyka HTML5 nezobrazí. Proto byl vyvinut kód ve skriptovacím jazyce JavaScript, který „naučí“ i tyto verze poznávat nové prvky jazyka HTML5. Tento skript se jmenuje HTML5shiv a je k dispozici na adrese [4]. Tento skript si můžeme stáhnout a přiložit našemu webu. Ve zdrojovém kódu hmtl dokumentu se použije tzv. podmínkový komentář, který umí Internet Explorer zpracovat. Pomocí tohoto komentáře řekneme, kdy se má jeho obsah zpracovat. Např. zápis:

```
<!--[if lt IE 9]>
<script src="html5shiv.js"> </script>
<![endif] -->
```

zpracuje pouze Internet Explorer starší než 9.

HTML5shiv není jediný skript, který zapíná nové prvky jazyka HTML ve starších verzích Internet Exploreru. Dalším je např. Modernizr [5]. Při použití Modernizr již nemusíme použít HTML5shiv.

Nyní se již podíváme na nové vlastnosti a prvky jazyka HTML5, odlišný zápis stávajících prvků, jejich rozdělení do kategorií.

Dělení prvků

V jazyce HTML4 byly prvky členěny na blokové a řádkové. Mezi blokové prvky patří např. prvek pro definici nadpisu **h1**, odstavce **p**. Řádkové prvky jsou např. prvek pro vložení obrázku ****, odkazu **<a>**. Blokové prvky se zobrazují vždy na novém řádku (pod sebou), řádkové na stejném řádku (vedle sebe). Toto dělení je spjato s jejich výchozím zobrazením podle specifikace HTML4. Při změně zobrazení pomocí kaskádových stylů však toto rozdělení ztrácelo smysl.

V jazyce HTML5 bylo toto členění přepracováno. Prvky nově dělíme do kategorií formulující obsah, rozdělující obsah a nadpisový obsah. Toto rozdělení je přesnější, protože na výsledném zobrazení mají vliv kaskádové styly.

Do kategorie nadpisový obsah spadají prvky **h1**, **h2**, **h3**, **h4**, **h5**, **h6**. Do kategorie rozdělující obsah patří nové prvky jazyka HTML5: **article**, **aside**, **nav** a **section**. Kategorie formulační obsah spadají prvky, které označují text, obrázky, odkazy a další.

DOCTYPE

Pomocí DOCTYPE řekneme prohlížeči, jaký typ dokumentu zpracovává. V případě HTML se jedná o verzi a typ. DOCTYPE se musí vyskytovat na začátku html dokumentu. Jeho definice v HTML4 byla poněkud složitější, např. pro HTML4 striktní typ bylo potřeba doplnit následující:

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

Vidíme, že uvedenou definici není snadno si zapamatovat. Proto tvůrci stránek začali využívat takové programy pro tvorbu webových stránek,

kde jsou obsaženy šablony, obsahující tyto definice, nebo programy, kde je možné si vytvořit a uložit šablony dokumentů.

V jazyce HTML5 je definice DOCTYPE mnohem jednodušší:

```
<doctype html>
```

Z definice zmizela číslovka určující verzi a také typ verze. Od HTML5 již existuje jenom jeden typ. Číslovku již také nemusíme uvádět, protože se předpokládá, že se nové verze jazyka HTML budou vyvíjet z předchozích.

Kódování stránky

Dalším prvkem, jehož zápis byl v jazyce HTML5 zjednodušen, je prvek pro uvedení kódování znaků dokumentu, např. Windows-1250, UTF-8. Do verze HTML4 včetně mělo uvedení kódování tento tvar:

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

V jazyce HTML5 je prvek zjednodušen jak je to jenom možné:

```
<meta charset="Windows-1250">
```

Připojení stylového předpisu

Ve verzi 4 se externí stylový předpis připojoval pomocí prvku `<link>`, který kromě parametru `href` pro zadání adresy souboru se stylovým předpisem a parametru `rel`, pomocí něhož uvádíme, zda se jedná o preferovanou (`stylesheet`) nebo alternativní verzi (`alternate stylesheet`) stylu, obsahoval i parametr `type`, pro uvedení typu připojovaného dokumentu (`type/css`):

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

Při použití jazyka HTML5 nemusíme používat parametr `type`, všechny prohlížeče totiž obsahový typ stylového předpisu rozeznají:

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

Připojení skriptů k html dokumentu

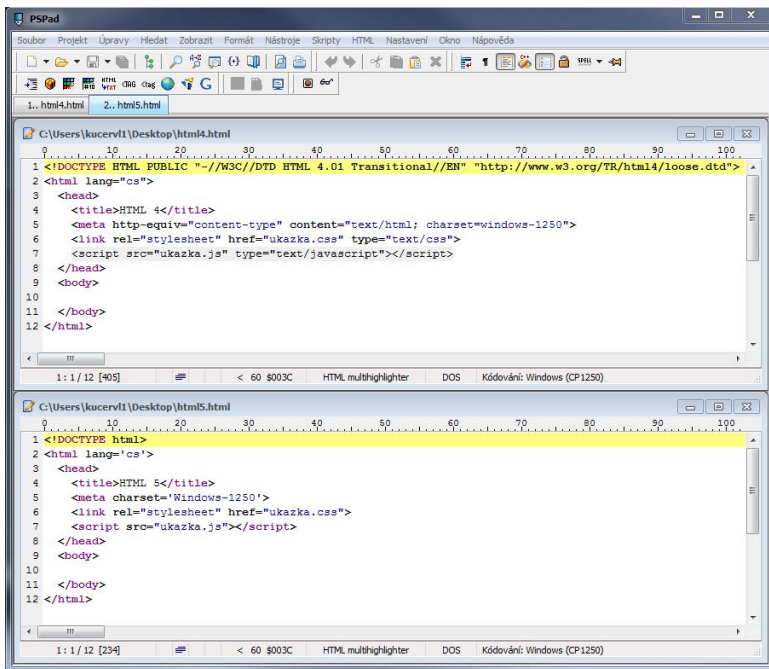
Podobně jako u připojení stylového předpisu až do verze HTML4 včetně jsme museli uvádět typ připojovaného skriptu, neboli, v jakém jazyce je skript napsán. Když např. v jazyce HTML4 připojujeme skript napsán v jazyce JavaScript, museli jsme uvést následující deklaraci:

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

JavaScript je v současné době jediným skriptovacím jazykem, který se používá na webu, a také prohlížeče předpokládají, že používáme JavaScript, není v HTML5 nutné uvádět parametr type:

```
<script src="script.js"> </script>
```

Na obr. 2 jsou názorně ukázány rozdíly ve zdrojovém kódu jazyka HTML4 a HTML5



Obr. 2

Závěr

V této části jsme se zabývali historií jazyků HTML5 a CSS. Podívali jsme se na změny, které dotýkají prvků, které se definují v hlavičce html dokumentu. V příští části se podíváme na nové prvky jazyka HTML, které nám pomáhají lépe strukturovat/členit html dokument.

Literatura

- [1] The HTML5 test – how well does your browser support HTML5? [online] [cit. 2013-05-21] Dostupné z: <http://html5test.com>
- [2] Can I use... Support tables for HTML5, CSS3, etc. [online] [cit. 2013-05-21] Dostupné z: <http://caniuse.com>
- [3] The CSS3 Test. [online] Dostupné z: <http://css3test.com> [cit. 2013-05-21].
- [4] Html5shiv – HTML5 IE enabling script. [online] [cit. 2013-05-21] Dostupné z: <http://code.google.com/p/html5shiv/>
- [5] Modernizr: The feature detection library for HTML5/CSS3. [online] [cit. 2013-05-21] Dostupné z: <http://modernizr.com/>
- [6] *Goldstein, A., Lazaris, L., Weyl, E.*: HTML5 a CSS3 pro webové designéry. Vyd. 1. Zoner Press, Brno, 2011.
- [7] *Castro, E.*: HTML5 a CSS3. Computer Press, 2012.

Bobřík učí informatiku

2. díl – Procházení grafů

DANIEL LESSNER – JIŘÍ VANÍČEK

Matematicko-fyzikální fakulta, UK Praha

Pedagogická fakulta, Jihočeská univerzita v Českých Budějovicích

V tomto dílu jsme vybrali sadu úloh o grafech a jejich procházení (resp. o stavovém prostoru). Jak z úloh zjistíme, nejedná se o znázornění tabulkových hodnot, známé z hromadného zpracování dat. Stejně tak se nejedná o grafy funkcí, známé z matematiky. Grafy v informatice jsou nástrojem k zachycení vztahů mezi objekty. Mezi příklady grafu patří rodokmen (vztahy mezi členy rodiny), schéma dopravních spojení mezi městy pro hledání nejkratší cesty, grafické znázornění výrobního procesu apod. Grafem je i vývojový diagram, ukazující postup kroků při vykonávání počítačového programu.

Objekty můžeme v grafu vyznačit jako body (tzv. vrcholy) a vztahy mezi těmito objekty pak jako spojnice těchto bodů (tzv. hrany – názvosloví je odvozeno od sítí mnohostěnnů). Grafy nám pomohou s řešením mnoha