

Technologie pro tvorbu webových aplikací - 2. díl (CSS, JavaScript, šablonovací systémy)

Vávra David · Informačné technológie

09.05.2012



Tento seriál je rozdelen na tri články, ktoré spoločne poskytujú pomerne široký náhľad do možností a technológií pro tvorbu webových aplikací. Z technológií byly vybrány asi nejčastěji používané - (X)HTML, CSS, JavaScript a PHP. Pomocí těchto technologií je dnes možné vytvářet velmi kvalitní dynamické webové aplikace. Vše začíná obecným rozdělením tvorby těchto aplikací.

První díl a druhý díl nesoucí podtitulky „rozdělení, HTML, XHTML,“ a „CSS, JavaScript, šablonovací systémy“ se zabývají současnými technologiemi používanými pro vývoj webových aplikací zpracovávanými na straně klienta. Poslední třetí díl s podtitulkem „PHP, PERL, ASP.NET“ se pak zabývá částmi, které jsou zpracovávány na straně serveru.

Úvod

Základní jazyk HTML je v dnešní době často doplněn o kaskádové styly - CSS, které nám umožňují oddělit formátování stránky od její vlastní struktury. JavaScript je multiplatformní, objektově orientovaný skriptovací jazyk. Jsou jím obvykle ovládány různé interaktivní prvky nebo tvořeny animace a efekty obrázků a je nejpoužívanějším skriptovacím jazykem na straně klienta. Šablonovací systémy jsou v první řadě určeny k oddělení aplikační a prezentační vrstvy.

1. CSS

CSS je zkratka pro anglický název Cascading Style Sheets, česky tabulky kaskádových stylů. Je to jazyk pro popis způsobu zobrazení stránek napsaných v jazycích HTML, XHTML nebo XML. Jazyk byl navržen standardizační organizací W3C, autorem prvotního návrhu byl Håkon Wium Lie. Byly vydány zatím dvě úrovně specifikace CSS1 a CSS2, dokončuje se revize CSS 2.1 a pracuje se na verzi CSS3.

Hlavním smyslem je umožnit návrhářům oddělit vzhled dokumentu od jeho struktury a obsahu. Původně to měl umožnit už jazyk HTML, ale v důsledku nedostatečných standardů a konkurenčního boje výrobců prohlížečů se vyvinul jinak. Starší verze HTML obsahují celou řadu elementů, které nepopisují pouze obsah a strukturu dokumentu, ale i způsob jeho zobrazení. Z hlediska zpracování dokumentů a vyhledávání informací není takový vývoj žádoucí.

2. Výhody CSS

- rozsáhlejší možnosti formátování – CSS nabízí rozsáhlejší formátovací možnosti než samotné HTML. Např. pro formátování bloku textu – tj. určení vzdálenosti od jejich elementu či okraje stránky nenabízí HTML nic. CSS má vlastnosti padding a margin. V HTML by bylo potřeba vytvořit složitou konstrukci vnořených tabulek.
- konzistentní styl – Na všech stránkách webové prezentace by měly být všechny nadpisy stejné úrovně, seznamy, zdůrazněné části textu apod. stejného stylu. S použitím formátovacích možností HTML je to obtížné – u každého objektu v každém dokumentu se vzhled objektu stále znovu nastavuje. S použitím CSS je to velmi jednoduché. Vytvoří se soubor stylu, který se připojuje k HTML dokumentu. Ve všech dokumentech jsou pak objekty stejného vzhledu.
- oddělení struktury a stylu
- dynamická práce se styly – Provést změnu stylu webu, který pro formátování vzhledu využívá jen možnosti HTML, znamená najít a nahradit všechny značky a změnit atributy mnoha dalších značek. V případě používání CSS znamená změna stylu webu přepsání jediného souboru – souboru stylů.
- formátování XML dokumentů
- větší kompatibilita alternativních webových prohlížečů
- kratší doba načítání stránky

Výhodou CSS oproti starému formátování v HTML je, že kód a obsah webu je uložen v souboru .html a veškerý design a formátování se načítá z jednoho souboru .css, který je většinou společný pro celý web. To znamená, že pokud máte v plánu změnu designu webu, stačí změnit pouze jeden soubor .css a změna se aplikuje na celý web. Také se soubor CSS uloží do mezipaměti prohlížeče a pokud není změněn, tak se načítá pouze jednou a zobrazení stránek se velmi urychlí.

Mohou také existovat různé styly pro různá výstupní zařízení. Webdesigner má tak možnost prostřednictvím CSS stylů dokumentu určit, jak bude vypadat na papíře, při projekci či na PDA apod. Specifikace CSS nezapomínají dokonce ani na zrakově postižené – je možno napsat styly pro hlasový syntetizátor nebo hmatovou čtečku Braillova písma.

Je také možnost upravit formátování podle prohlížeče, kterým si uživatel danou stránku zobrazuje. Jednoduše se vytvoří více souborů CSS (např. styl1.css a styl2.css) a podle prohlížeče, který si o stránku požádá, je připojen daný soubor. Tím se dá eliminovat problém různé interpretace kódu jednotlivými prohlížeči.

3. Nevýhody CSS

Hlavní nevýhodou CSS je ne vždy dostatečná podpora v majoritních prohlížečích. Prohlížeče obsahují v implementaci CSS chyby a je někdy nesnadné napsat kód tak, aby se ve všech (resp. v běžných) prohlížečích výsledek zobrazil stejně. Situace se poslední dobou značně zlepšuje, mj. opravami chyb v prohlížečích a nástupem nových verzí Internet Exploreru. Existuje ovšem jednoduchý kód, který se může pokusit zlepšit zobrazování generovaných stránek. Stačí do souboru, jenž generuje stránku vložit:

```
<!-- [if IE]>
```

```
<style type="text/css">
#left {
left: 150px;
}
</style>
<![endif]->
```

Vysvětlení:

<![if IE]> = Podmínka - pokud prohlížeč stránky je Internet Explorer, vloží se do stránky kód, který se nachází mezi <![if IE]> a <![endif]->, v našem případě #left.....

<![endif]-> syntaxe ukončující kód, který se bude vkládat do webu při splnění podmínky typu prohlížeče.

4. Syntaxe CSS

Stylový předpis se skládá z posloupnosti pravidel. Každé pravidlo určuje vzhled některého elementu dokumentu, nebo skupiny elementů. Pravidlo začíná tzv. selektorem, který specifikuje („adresuje“) skupinu elementů. Selektor je následován seznamem deklarácí, které určují vzhled vybrané skupiny elementů. Celý seznam je uzavřen ve složených závorkách a jednotlivé deklarace jsou odděleny středníkem (tj. za poslední deklarací středník už být nemusí).

Příklad stylu:

```
h1 { /* vzhled nadpisu první úrovně */
margin: 5px; /* okraj šířky 5 pixelů */
font-size: 12pt /* velikost fontu 12 bodů */
}
p { /* styl odstavce */
text-align: center; /* text centrovat */
line-height: 10pt; /* výška řádku 10 bodů */
}
```

Selektory se dají také seskupovat tzn. dva a více selektorů, jeden seznam deklarácí, oddělují se čárkou (.). Následující příklad by způsobil, že prvky h1 i h2 by byly zelené:

```
h1, h2 {color: green;}
```

Se selektory se dají dělat divy např. obarvit modře jen prvek b nacházející se v prvku p (ostatní nenacházející se v prvku p ne). Zápis by vypadal takto:

```
p b { color: blue; }
```

CSS se dá definovat i v těle stránky, například:

```
<h1 style="color:red; background-color:white;">nadpis</h1>
```

kde: h1- nadpis první úrovně, style je styl, který se píše normálně do hlavičky např.:

```
<style>
h1 {
color:red;
background-color:white;
}
</style>
```

tento způsob se ovšem nedoporučuje [1],[2],[3]

5. JavaScript

JavaScript je multiplatformní, objektově orientovaný skriptovací jazyk, jehož autorem je Brendan Eich z tehdejší společnosti Netscape. Nyní se zpravidla používá jako interpretovaný programovací jazyk pro WWW stránky, často vkládaný přímo do HTML kódu stránky. Jsou jím obvykle ovládány různé interaktivní prvky GUI (tlačítka, textová políčka) nebo tvořeny animace a efekty obrázků.

Jeho syntaxe patří do rodiny jazyků C/C++/Java. Slovo Java je však součástí jeho názvu pouze z marketingových důvodů a s programovacím jazykem Java jej vedle názvu spojuje jen podobná syntaxe. JavaScript byl v červenci 1997 standardizován asociací ECMA (European Computer Manufacturers Association) a v srpnu 1998 ISO (International Organization for Standardization). Standardizovaná verze JavaScriptu je pojmenována jako ECMAScript a z ní byly odvozeny i další implementace, jako je například ActionScript.

JavaScript byl původně obchodní název implementace společnosti Netscape, kde byl vyvíjen nejprve pod názvem Mocha, později LiveScript, ohlášen byl společně se společností Sun Microsystems v prosinci 1995 jako doplněk k jazykům HTML a Java. Pro verzi firmy Microsoft je použit název JScript. JScript je podporován platformou .NET. Program v JavaScriptu se obvykle spouští až po stažení WWW stránky z Internetu (tzv. na straně klienta), na rozdíl od ostatních jiných interpretovaných programovacích jazyků (např. PHP a ASP), které se spouštějí na straně serveru ještě před stažením z Internetu. Z toho plynou jistá bezpečnostní omezení, JavaScript např. nemůže pracovat se soubory, aby tím neohrozil soukromí uživatele.

JavaScript je možné použít i na straně serveru. První implementací JavaScriptu na straně serveru byl LiveWire firmy Netscape vypuštěný roku 1996, dnes existuje několik možností včetně opensource implementace Rhinola založená na Rhino, gcj a Apache.

Kromě DHTML se JavaScript používá k psaní rozšíření pro mnohé aplikace, například Adobe Acrobat. JavaScript je také možno spustět v operačních systémech Windows pomocí programu Windows Script Host a nahradit tak dávkové soubory MS-DOS. [3],[4],[5]

6. Šablonovací systémy

Výhody použití šablonovacího systému

- Oddělení aplikační a prezentační vrstvy (snadná změna vzhledu změnou šablony).
- Vymezení rolí při vývoji a správě.

- Zamezení duplicit v kódu - snížení nákladů při správě.
- Nezávislost na použitém programovacím jazyce.

Nevýhody použití šablonovacího systému

- Nutnost naučit se pseudojazyk vkládaný do (X)HTML šablon.
- Pomalejší generování výstupu (lze zrychlit cachováním).

Specializace (vymezení rolí)

Najít odborníky, kteří jsou schopni produkovat kvalitní XHTML + CSS a současně jsou schopnými programátory, je velice obtížné. Šablonovací systémy tento problém řeší.

Oddělení (X)HTML od aplikačního kódu

Pouhé oddělení (X)HTML od kódu aplikace však nestačí, situaci nám komplikuje prezentační logika (požadavky typu "každý sudý řádek výpisu podbarvit"). Různé šablonovací systémy toto řeší různě, každé z řešení má však své výhody i nevýhody.

Prezentační logika v aplikaci

V případě, kdy máme prezentační logiku uvnitř aplikace, je vhodné ji oddělit od zbytku aplikace například vlastním API, komponentami nebo pomocnými objekty.

Prezentační logika v šablonách

Prezentační logiku lze řešit také přímo v šablonách (pokud to daný systém umožňuje). Například systém Smarty obsahuje značky, kterými lze provádět podmíněný výpis atp. [3]

Zdroje

1. CASTRO, Elizabeth. HTML, XHTML a CSS - Názorný průvodce tvorbou WWW stránek. Computer Press, 2007. ISBN 978-80-251-1531-2.
2. Wikipedie: otevřená encyklopedie [online]. [cit. 2012-06-26]. Dostupný z WWW: http://cs.wikipedia.org/wiki/Cascading_Style_Sheets
3. VÁVRA, David. WWW prezentace a návrh e-obchodu soukromého subjektu - živnosti [bakalářská práce]. FAI UTB Zlín, 2009.
4. STEJSKAL, J. Vytváříme WWW stránky pomocí HTML, CSS, a JavaScript. Computer Press, 2006, ISBN 80-251-0176-3.
5. Wikipedie: otevřená encyklopedie [online]. [cit. 2012-06-26]. Dostupný z WWW: <http://cs.wikipedia.org/wiki/Javascript>
6. VÁVRA, David. Tvorba e-obchodu a vývoj linuxového skriptu pro automatické generování nových e-obchodů [diplomová práce]. FAI UTB Zlín, 2011.

