

Skriptování na straně serveru a klienta

4IZ228 – tvorba webových stránek a aplikací

Jirka Kosek

Poslední modifikace: \$Date: 2012/03/29 21:34:27 \$

Copyright © 2000-2012 Jiří Kosek

Obsah

Úvod	3
Nová platforma	4
Výhody webové platformy	5
Nevýhody webové platformy	6
Přístupy k tvorbě webových aplikací	7
Klientské technologie pro tvorbu aplikací	8
Co umí každý prohlížeč	9
Rozšiřující technologie	10
Výhody použití rozšiřujících klientských technologií	11
Nevýhody použití rozšiřujících klientských technologií	12
Klientské skripty v HTML	13
Princip	14
Vložení skriptu do stránky (Skripty prováděné při načítání stránky)	15
Vložení skriptu do stránky (Volání skriptu v okamžiku, kdy nastane nějaká událost)	16
Události	17
Obsluha událostí	19
Javascript	20
Objektový model dokumentu	21
Objektový model dokumentu a kompatibilita	22
Behaviours	23
JS frameworky	24
AJAX	25
Java applety	26
Java applety	27
Bezpečnost	28
Rychlost a výkon	29
Vložení do HTML	30
Předávání parametrů Java appletu	31
ActiveX	32
ActiveX	33
Bezpečnost (či spíše nebezpečnost)	34
Vložení do stránky	35

Úvod

Nová platforma	4
Výhody webové platformy	5
Nevýhody webové platformy	6
Přístupy k tvorbě webových aplikací	7

Nová platforma

- původně byl web navržen pro sdílení statických dokumentů
- dnes v něm lze používat různé aplikace
 - elektronické obchody
 - internetové vyhledávače
 - firemní informační systémy
 - hry
 - ...
- web je nová platforma pro provozování aplikací

Výhody webové platformy

- globální dosah – aplikaci mohou využívat uživatelé z celého světa
- minimalizace nákladů na správu aplikací – vše běží na serveru, není potřeba instalovat klientskou část aplikace, tu zajistí standardní prohlížeč

Nevýhody webové platformy

- ovládání aplikací nedosahuje pohodlí a interaktivity nativních klientských aplikací určených pro konkrétní operační systém
- webové vývojové nástroje zatím většinou nedosahují kvalit prostředí (IDE) pro vývoj klasických klientských aplikací
- je potřeba pochopit a naučit se používat několik nových technologií
 - HTML/XHTML + CSS
 - JavaScript + DOM
 - HTTP
 - serverový skriptovací jazyk/framework – PHP, ASP.NET, JSP, Ruby on Rails, ...

Přístupy k tvorbě webových aplikací

- výkonná část aplikace běží přímo v prohlížeči
- výkonná část aplikace běží na serveru
- nejčastěji se používá kombinace obou technologií

Klientské technologie pro tvorbu aplikací

Co umí každý prohlížeč	9
Rozšiřující technologie	10
Výhody použití rozšiřujících klientských technologií	11
Nevýhody použití rozšiřujících klientských technologií	12

Co umí každý prohlížeč

- zobrazovat HTML stránky
- používat v HTML stránce formuláře (obdoba dialogových oken)
- odeslat data vyplněná do formuláře pomocí protokolu HTTP zpět serveru
- bez formulářů by webové aplikace nemohly vůbec existovat

Rozšiřující technologie

- klientské skripty (Javascript)
- AJAX
- Java applety
- ActiveX komponenty
- Native Client

Výhody použití rozšiřujících klientských technologií

- stránky mohou být více interaktivní a snazší na ovládání
- rychlejší odezva
 - nemusí se vždy komunikovat se serverem
 - nemusí se přenášet a překreslovat celá stránka
- odlehčení zátěže na serveru

Nevýhody použití rozšiřujících klientských technologií

- podpora těchto technologií v jednotlivých prohlížečích není navzájem 100% kompatibilní
- některé technologie nejsou nezávislé na platformě (ActiveX)
- z bezpečnostních důvodů může uživatel zakázat používání některých technologií
- některé technologie umožňují neoprávněně manipulovat s daty na počítači uživatele
- sečteno a podtrženo → na klientské technologie není 100% spolehnutí, základní funkcionality aplikace by měla být dostupná i v prohlížeči bez podpory klientských technologií

Klientské skripty v HTML

Princip	14
Vložení skriptu do stránky (Skripty prováděné při načítání stránky)	15
Vložení skriptu do stránky (Volání skriptu v okamžiku, kdy nastane nějaká událost)	16
Události	17
Obsluha událostí	19
Javascript	20
Objektový model dokumentu	21
Objektový model dokumentu a kompatibilita	22
Behaviours	23
JS frameworky	24
AJAX	25

Princip

- přímo do HTML stránky lze zařadit jednoduché programy
- program může reagovat na události vyvolané uživatelem
- program může manipulovat s dokumentem a prohlížečem
- teoreticky lze použít libovolný jazyk
 - nejpodporovanější je Javascript
 - některé prohlížeče podporují VBScript, Perl, Tcl, Python, ...

Vložení skriptu do stránky

Skripty prováděné při načítání stránky

- skript vložený přímo do HTML

```
<script type="text/javascript">  
  příkazy skriptového jazyka  
</script>
```

- skript načtený z externího souboru

```
<script type="text/javascript"  
  src="URL adresa skriptu">  
</script>
```

- vzhledem k dominanci Javascriptu a chování prohlížečů je možné vynechat atribut `type`
- skripty prováděné v okamžiku načtení stránky většinou jen nadefinují funkce, které se pak vyvolávají jako odezva na reakci uživatele

Vložení skriptu do stránky

Volání skriptu v okamžiku, kdy nastane nějaká událost

- u každého elementu v HTML dokumentu můžeme použít několik atributů, které odpovídají jednotlivým událostem
- jako hodnota atributu se uvádí kód, který se má provést

```
<tag událost="obslužný kód">...</tag>
```

- pomocí událostí lze vytvářet interaktivní stránky – reagují na chování uživatele
 - změna obrázků při přejetí myši
 - kontrola formuláře před odesláním

Události

Tabulka 1. Události, které lze obsloužit skriptem

Událost	Popis
onLoad	Událost je vyvolána po natažení dokumentu do okna prohlížeče nebo do všech rámců v rámci jednoho FRAMESET. Atribut může být použit u elementů BODY a FRAMESET.
onUnLoad	Událost je vyvolána po odstranění dokumentu z okna nebo rámu. Atribut může být použit s elementy BODY a FRAMESET.
onClick	Událost je vyvolána po kliknutí myši na element. Atribut může být použit u většiny elementů.
onDbIcIck	Událost je vyvolána po dvojitém kliknutí myši na element. Atribut může být použit u většiny elementů.
onMouseDown	Událost je vyvolána po stisknutí tlačítka myši nad elementem. Atribut může být použit u většiny elementů.
onMouseUp	Událost je vyvolána po uvolnění tlačítka myši nad elementem. Atribut může být použit u většiny elementů.
onMouseOver	Událost je vyvolána při přesunutí myši nad element. Atribut může být použit u většiny elementů.
onMouseMove	Událost je vyvolána při pohybu myši nad elementem. Atribut může být použit u většiny elementů.
onMouseOut	Událost je vyvolána po odsunutí myši z elementu. Atribut může být použit u většiny elementů.
onFocus	Událost je vyvolána v okamžiku, kdy je element aktivován myší nebo pomocí tabulátoru. Atribut je možno použít u elementů LABEL, INPUT, SELECT, TEXTAREA a BUTTON.

Události (Pokračování)

Událost	Popis
onBlur	Událost je vyvolána v okamžiku, kdy element přestává být aktivní. Atribut je možno použít u elementů LABEL, INPUT, SELECT, TEXTAREA a BUTTON.
onKeyPress	Událost je vyvolána po stisku a následném uvolnění tlačítka na klávesnici. Atribut může být použit u většiny elementů.
onKeyDown	Událost je vyvolána po stisku tlačítka na klávesnici. Atribut může být použit u většiny elementů.
onKeyUp	Událost je vyvolána po uvolnění tlačítka na klávesnici. Atribut může být použit u většiny elementů.
onSubmit	Událost je vyvolána při odesílání formuláře. Atribut může být použit pouze u elementu FORM.
onReset	Událost je vyvolána po vynulování formuláře. Atribut může být použit pouze u elementu FORM.
onSelect	Událost je vyvolána po označení textu ve vstupním poli. Atribut může být použit u elementů INPUT a TEXTAREA.
onChange	Událost je vyvolána pokud se změnila hodnota vstupního pole formuláře. Atribut může být použit u elementů INPUT, SELECT a TEXTAREA.

Obsluha událostí

Příklad 1. Ukázka jednoduché obsluhy události

O rozvoj Webu se stará konsorcium

```
<a href="http://www.w3.org"
  onmouseover="alert('Tenhle odkaz nesmíte minout!')">W3C</a>
```

Javascript

- jednoduchý skriptovací jazyk se syntaxí vycházející z Javy a C
- netyповané proměnné, podmínky, cykly, funkce, prototypové objekty
- několik verzí
 - Javascript – Netscape
 - JScript – implementace Javascriptu od Microsoftu
 - ECMAScript – standard
- Javascript i JScript mají oproti ECMAScriptu rozšíření navíc

Objektový model dokumentu

- skupina objektů, které zpřístupňují objekty stránky a prohlížeč
- základní objekt, který umožňuje přístup k prohlížeči se jmenuje `window`, takže jeho jméno můžeme při zápisu cesty o objektu obvykle vynechat
- v rámci `window` jsou pak dostupné další objekty:

`location`

Informace o URL stránky zobrazené v okně

`frames`

Rámy na stránce

`history`

Historie dokumentů prohlížeče

`navigator`

Informace o prohlížeči

`document`

Tento objekt obsahuje další objekty, které se vztahují k jednomu dokumentu:

`images`

Obrázky na stránce; lze využít k různým interaktivním odezvám

`forms`

Objekt zpřístupňující všechny formuláře na stránce, jejich pole apod.

`frames`

Přístup k ráům, které dokument obsahuje

- při přístupu k nějaké části dokumentu, se objekty oddělují tečkou

`location.href`

`navigator.appName`

`window.navigator.appName`

Objektový model dokumentu a kompatibilita

- dříve popsaný objektový model vychází z první implementace Javascriptu od firmy Netscape
- Netscape a Microsoft tento model dále upravovali, každý bohužel dost odlišným způsobem
- kromě základních věcí nebyly prohlížeče dlouhou kompatibilní a kód se musí psát dvakrát
- konsorcium W3C vytvořilo standard – Document Object Model (DOM), který sjednocuje objektovou hierarchii používanou pro přístup k dokumentu
- podpora W3C DOM se v posledních letech hodně zlepšuje a lépe se píší skripty, které fungují ve většině prohlížečů
- specifikace DOM je navíc dále upřesňována a je nyní součástí specifikace HTML5

Behaviours

- metoda pro zjednodušení zápisu stránek bohatých na klientské skriptování
- kód obsluhy událostí je zcela oddělen od HTML stránky
- vazba mezi událostmi a elementy na stránce je definována pomocí CSS
- podporuje IE5+ (HTC¹) a Mozilla (XBL²)
- W3C snad časem sjednotí jako jeden standard³
- podobný přístup lze díky moderním javascriptovým frameworkům používat již dnes

¹ <http://www.w3.org/TR/1998/NOTE-HTMLComponents-19981023>

² <http://www.w3.org/TR/2001/NOTE-xbl-20010223/>

³ <http://www.w3.org/TR/xbl/>

JS frameworky

- současné aplikace vyžadují stále složitější klientské skripty
- mezi prohlížeči pořád existují drobné rozdíly a jejich detekování a obcházení je pracné
- bez XBL/HTC je správa obsluhy událostí velice nepřehledná
- řešení nabízejí tzv. javascriptové frameworky
 - nabízejí vlastní objektový model, který překlenuje případnou nekompatibilitu mezi prohlížeči
 - nabízejí hotové knihovny pro mnoho běžných operací
 - většinou nabízejí efektivní mechanismus pro hromadné definování obsluhy událostí
 - např. Dojo⁴, jQuery⁵, Prototype⁶, Yahoo User Library⁷

⁴ <http://dojotoolkit.org/>

⁵ <http://jquery.com/>

⁶ <http://www.prototypejs.org/>

⁷ <http://developer.yahoo.com/yui/>

AJAX

- AJAX = Asynchronous Javascript and XML
- nejedná se o novou technologii, ale o pojmenování skupiny technologií a jejich inovativního způsobu užití
- aplikace běží v prohlížeči a o výměnu dat se serverem se stará kód v Javascriptu
- není vyžadován reload celé stránky
- data se serverem se vyměňují v XML, jako fragmenty kódu Javascriptu nebo pomocí formát JSON
- AJAX umožňuje psát vysoce interaktivní aplikace
- některé serverové frameworky pro psaní webových aplikací obsahují komponenty uživatelského rozhraní, které automaticky generují AJAX kód
- pozor na zachování vazby mezi stavem AJAXové aplikace a URL

Java applety

Java applety	27
Bezpečnost	28
Rychlost a výkon	29
Vložení do HTML	30
Předávání parametrů Java appletu	31

Java applety

- malé kusy kódu, které běží přímo v prohlížeči
- mají vymezený určitý prostor obrazovky a do něj mohou podle potřeby kreslit, číst klávesnici, myš apod.
- ze zdrojového tvaru (.java) je potřeba zkompileovat do byte-code (.class)
- byte-code je nezávislý na platformě
- JVM (Java Virtual Machine) se stará o interpretaci kódu

Bezpečnost

- JVM odstiňuje aplikaci od operačního systému a hardware počítače
- JVM obsahuje security manager, který definuje práva appletu
- dobře napsaná JVM se správným security managerem např. nedovoluje appletu přístup k lokálnímu souborovému systému

Rychlost a výkon

- rychlost stahování – jednoduché aplikace zabírají jen několik desítek KB
- rychlost provádění – většina JVM dnes již obsahuje JIT kompilátor
- JIT (just-in-time) – před spuštěním se třídy appletu převedou z byte-code do nativního kódu

Vložení do HTML

- Java applety se do stránky vkládají pomocí elementu `applet`
- od HTML 4.0 je lze vkládat i pomocí elementu `object`
- při vkládání každého appletu bychom měli použít alespoň tři následující atributy:

CODE

jméno appletu (`.class` souboru), který chceme spustit

WIDTH

šířka plochy vyhrazené pro applet v pixelech

HEIGHT

výška plochy vyhrazené pro applet v pixelech

- můžeme používat i atributy, které známe od obrázků – `ALIGN`, `HSPACE` a `VSPACE`
- mezi `<applet>` a `</applet>` se vkládá text, který se zobrazí v prohlížečích bez podpory Javy
- pokud je soubor s appletem v jiném adresáři než stránka, která ho volá, musíme adresář, kde je applet uložen, určit pomocí atributu `CODEBASE`
- pokud máme na jedné stránce několik appletů, které spolu mají komunikovat, musíme je pojmenovat pomocí atributu `name`

Předávání parametrů Java appletu

- parametry můžeme appletům předávat přímo ze stránky
 - element `PARAM` se vkládá do elementu `APPLET`
 - hodnoty se předávají pomocí atributů `NAME` a `VALUE`
- applet může číst soubory odkudkoliv ze sítě pomocí HTTP a dalších protokolů
- applet může komunikovat s dalšími aplikacemi pomocí internetových protokolů
- spolupráce s databázemi

ActiveX

ActiveX	33
Bezpečnost (či spíše nebezpečnost)	34
Vložení do stránky	35

ActiveX

- binární komponenta vkládaná do stránky podobně jako Java applet
- binární → funguje pouze ve 32bitových Windows
- vytvořit ji lze v libovolném jazyce – VisualBasic, Visual C++, Delphi, ...

Bezpečnost

či spíše nebezpečnost

- po spuštění je již ActiveX komponenta pánem počítače
- komponenta není nijak odstíněná od OS
- komponentu lze opatřit digitálním podpisem
- uživatel se musí rozhodnout, zda tvůrci ActiveX komponenty důvěřuje
- v historii již bylo mnoho případů závažného porušení bezpečnosti a soukromí pomocí ActiveX komponent – převody pomocí homebankingového programu, změna parametrů telefonického připojení apod.

Vložení do stránky

- pomocí elementu `object`

```
<object classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000"  
  ▶  
  codebase="http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#version=5,0,0,0"  
  width="360"  
  height="360">  
  <param name="movie" value="Keyframing.swf">  
  <param name="quality" value="high">  
  <param name="bgcolor" value="#FFFFFF">  
</object>
```