
Stručný obsah

Předmluva	19
Úvod	23
1. Co je JavaScript	29
2. JavaScript v jazyku HTML	41
3. Základy jazyka	51
4. Proměnné, obor platnosti a paměť	103
5. Odkazované typy	121
6. Objektově orientované programování	177
7. Anonymní funkce	213
8. Objektový model prohlížeče	233
9. Detekce klienta	261
10. Objektový model dokumentu	293
11. Model DOM na úrovni 2 a 3	347
12. Události	395
13. Skriptování formulářů	467
14. Ošetření a ladění chyb	501
15. JavaScript a XML	553
16. ECMAScript pro XML	587
17. Ajax a JSON	607

18. Pokročilé techniky	631
19. Úložiště na straně klienta	663
20. Osvědčené postupy	683
21. Připravovaná aplikační rozhraní	717
22. Vývoj JavaScriptu	751
A. Knihovny JavaScriptu	809
B. Nástroje JavaScriptu	815
Rejstřík	823

Obsah

Předmluva	19
Úvod	23
Co v knize najdete?	23
Komu je kniha určena?	24
Co potřebujete ke studiu témat této knihy	24
Uspořádání knihy	24
Použité konvence	27
Zdrojový kód	27
Errata	28
Poznámka redakce českého vydání	28
Kapitola 1: Co je JavaScript	29
Stručná historie	29
Implementace jazyka JavaScript	31
ECMAScript	31
Objektový model dokumentu (DOM)	34
Objektový model prohlížeče (BOM)	38
Verze jazyka JavaScript	38
Shrnutí	39
Kapitola 2: JavaScript v jazyku HTML	41
Element script	41
Umístění značky	43

Odložené spouštění skriptů	44
Změny v jazyku XHTML	45
Zastaralá syntaxe	47
Vložený kód a externí soubory	47
Režimy dokumentu	48
Element noscript	49
Shrnutí	50
Kapitola 3: Základy jazyka	51
Syntaxe	51
Rozlišování velkých a malých písmen	51
Identifikátory	52
Komentáře	52
Příkazy	52
Klíčová a vyhrazená slova	53
Proměnné	54
Datové typy	55
Operátor typeof	56
Typ Undefined	56
Typ Null	57
Typ Boolean	58
Typ Number	59
Typ String	65
Typ Object	67
Operátory	68
Unární operátory	68
Bitové operátory	72
Logické operátory	78
Multiplikativní operátory	81
Aditivní operátory	82
Relační operátory	84
Porovnávací operátory	86
Podmínkový operátor	87
Přiřazovací operátory	88
Operátor čárka	88
Příkazy	89
Příkaz if	89
Příkaz do-while	90
Příkaz while	90
Příkaz for	91
Příkaz for-in	92
Označování příkazů	92
Příkazy break a continue	93
Příkaz with	95

Příkaz switch	95
Funkce	98
Argumenty funkce	99
Žádné přetěžování	101
Shrnutí	102
Kapitola 4: Proměnné, obor platnosti a paměť	103
Primitivní a odkazované hodnoty	103
Dynamické vlastnosti	104
Kopírování hodnot	105
Předávání argumentů	106
Zjišťování typu	107
Kontext a obor platnosti provádění kódu	108
Rozšíření řetězu oborů platnosti	111
Žádné obory platnosti na úrovni bloků	112
Úklid paměti	115
Označ a zameř	115
Počítání odkazů	116
Výkon	117
Správa paměti	118
Shrnutí	118
Kapitola 5: Odkazované typy	121
Typ Object	121
Typ Array	124
Metody pro převod	127
Metody pro práci se zásobníkem	128
Metody pro práci s frontou	129
Metody pro přeskupování prvků	130
Metody pro manipulaci s prvky	132
Datový typ Date	134
Zděděné metody	136
Metody pro formátování kalendářních dat	137
Metody pro práci se složkami data a času	137
Typ RegExp	139
Vlastnosti instancí typu RegExp	141
Metody instancí typu RegExp	142
Vlastnosti konstruktoru typu RegExp	144
Omezení vzorů	146
Typ Function	147
Žádné přetěžování (revize)	148
Deklarace funkcí vs. funkční výrazy	149
Funkce jako hodnoty	149

Obsah

Interní objekty funkce	151
Vlastnosti a metody funkce	153
Primitivní obalové typy	155
Typ Boolean	156
Typ Number	157
Typ String	159
Vestavěné objekty	168
Objekt Global	168
Objekt Math	171
Shrnutí	175
Kapitola 6: Objektově orientované programování	177
Vytváříme objekty	177
Vzor továrna	178
Vzor konstruktor	178
Vzor prototyp	182
Kombinace vzorů konstruktor a prototyp	193
Vzor dynamický prototyp	194
Vzor parazitní konstruktor	195
Vzor odolný konstruktor	196
Dědičnost	197
Řetězení prototypů	198
Kradení konstruktoru	203
Kombinovaná dědičnost	205
Prototypová dědičnost	206
Parazitní dědičnost	207
Parazitní kombinovaná dědičnost	208
Shrnutí	211
Kapitola 7: Anonymní funkce	213
Rekurze	214
Uzávěry	215
Uzávěry a proměnné	218
Objekt this	219
Neuvolněná paměť	221
Napodobování blokového oboru platnosti	222
Soukromé proměnné	224
Statické soukromé proměnné	226
Vzor modul	228
Vzor rozšíření modulu	230

Shrnutí	231
Kapitola 8: Objektový model prohlížeče	233
<hr/>	
Objekt window	233
Globální obor platnosti	233
Okno a rámy	234
Pozice okna	237
Velikost okna	238
Otevírání adres URL a oken	239
Intervaly a časové prodlevy	243
Systémová dialogová okna	246
Objekt location	248
Argumenty dotazového řetězce	249
Práce s lokací	250
Objekt navigator	252
Detekce modulů plug-in	254
Registrace obsluhy	256
Objekt screen	257
Objekt history	258
Shrnutí	260
Kapitola 9: Detekce klienta	261
<hr/>	
Detekce schopností	261
Detekce podivností	263
Detekce agenta uživatele	264
Historie	265
Pracujeme s detekcí agenta uživatele	272
Kompletní skript	288
Použití	292
Shrnutí	292
Kapitola 10: Objektový model dokumentu	293
<hr/>	
Hierarchie uzlů	293
Typ Node	295
Typ Document	300
Typ Element	310
Typ Text	320
Typ Comment	323
Typ CDATASection	324
Typ DocumentType	324
Typ DocumentFragment	325
Typ Attr	326

Rozšíření modelu DOM	327
Vykreslovací režimy	327
Posouvání	328
Vlastnost children	329
Metoda contains()	329
Manipulace s obsahem	331
Práce s modelem DOM	337
Dynamické skripty	337
Dynamické styly	340
Manipulace s tabulkami	342
Použití seznamů NodeList	344
Shrnutí	345
Kapitola 11: Model DOM na úrovni 2 a 3	347
Změny modelu DOM	348
Obory názvů jazyka XML	348
Další změny	352
Styly	356
Přístup ke stylům elementů	356
Práce s šablonami stylů	361
Rozměry elementu	366
Procházení prvků	373
NodeIterator	375
TreeWalker	378
Rozsahy	379
Rozsahy v modelu DOM	380
Rozsahy v Internet Exploreru	389
Shrnutí	393
Kapitola 12: Události	395
Tok událostí	395
Probublávání událostí	396
Zachytávání událostí	397
Tok událostí v modelu DOM	397
Obsluhy nebo posluchači událostí	398
Obsluhy událostí v jazyku HTML	398
Obsluhy událostí na úrovni DOM Level 0	399
Obsluhy událostí na úrovni DOM Level 2	401
Obsluhy událostí v Internet Exploreru	402
Obsluhy událostí fungující ve všech prohlížečích	404
Objekt event	405
Objekt event modelu DOM	405
Objekt event v Internet Exploreru	409

Objekt event fungující ve všech prohlížečích	411
Typy událostí	414
Události uživatelského rozhraní	414
Události myši	414
Událost klávesnice	423
Události jazyka HTML	428
Mutační události	434
Proprietární události	440
Události prohlížeče Safari na mobilních zařízeních	451
Paměť a výkon	455
Delegování událostí	455
Odstraňování obsluh událostí	457
Simulace událostí	459
Simulace událostí modelu DOM	459
Simulace událostí v Internet Exploreru	464
Shrnutí	465
Kapitola 13: Skriptování formulářů	467
Základy formulářů	467
Odesílání formulářů	468
Resetování formulářů	469
Pole formuláře	470
Skriptování textových polí	475
Výběr textu	476
Filtrování vstupu	480
Automatické posouvání tabulátoru	484
Skriptování výběrových polí	485
Výběr možností	487
Přidávání možností	489
Odstraňování možností	490
Přesouvání a přeskupování možností	490
Serializace formuláře	491
Úpravy textu s formátováním	493
Práce s textovým polem s formátováním	494
Výběr v textovém poli s formátováním	497
Textová pole s formátováním ve formulářích	498
Shrnutí	499
Kapitola 14: Ošetření a ladění chyb	501
Hlášení chyb v prohlížečích	501
Internet Explorer	501
Firefox	504
Safari	505

Opera	506
Chrome	508
Ošetření chyb	509
Příkaz try-catch	510
Vyvolávání chyb	513
Událost error	516
Strategie ošetřování chyb	517
Identifikace míst s potenciálním vznikem chyb	518
Rozlišování mezi fatálními a nefatálními chybami	523
Protokolování chyb na server	524
Techniky ladění	525
Protokolování zpráv do konzoly	525
Protokolování zpráv na stránku	528
Vyvolávání chyb	529
Chyby v Internet Exploreru	530
Operace přerušena (operation aborted)	530
Neplatný znak (invalid characted)	532
Člen nebyl nalezen (member not found)	532
Neznámá chyba při běhu programu (unknown runtime error)	533
Syntaktická chyba (syntax error)	533
Systém nemůže nalézt uvedený zdroj (the systém cannot locate the resource specified)	534
Ladící nástroje	534
Internet Explorer Debugger	534
Firebug	540
Drosera	544
Nástroj pro ladění JavaScriptu v prohlížeči Opera	548
Další možnosti	550
Shrnutí	551
Kapitola 15: JavaScript a XML	553
Podpora modelu XML DOM v prohlížečích	553
Modul DOM Core na úrovni DOM Level 2	553
Typ DOMParser	554
Typ XMLSerializer	555
Modul DOM Load and Save na úrovni DOM Level 3	556
Jazyk XML a Internet Explorer	562
Zpracování kódu jazyka XML fungující ve všech prohlížečích	566
Podpora jazyka XPath v prohlížečích	568
Modul DOM XPath na úrovni DOM Level 3	569
Jazyk XPath a Internet Explorer	574
Práce s jazykem XPath napříč všemi prohlížeči	575
Podpora jazyka XSLT v prohlížečích	578
Jazyk XSLT a Internet Explorer	578
Typ XSLTProcessor	583

Práce s jazykem XSLT napříč všemi prohlížeči	585
Shrnutí	586
Kapitola 16: ECMAScript pro XML	587
<hr/>	
Typy E4X	587
Typ XML	587
Typ XMLList	588
Typ Namespace	590
Typ QName	591
Praktické použití rozšíření E4X	592
Přístup k atributům	593
Další typy uzlů	595
Dotazy nad strukturou XML	596
Konstrukce kódu jazyka XML a manipulace s tímto kódem	597
Možnosti analýzy a serializace	600
Obory názvů	601
Další změny	603
Aktivace plného rozšíření E4X	604
Shrnutí	605
Kapitola 17: Ajax a JSON	607
<hr/>	
Objekt XHR	608
Práce s objektem XHR	609
Hlavičky protokolu HTTP	612
Požadavky GET	613
Požadavky POST	614
Odlišnosti v jednotlivých prohlížečích	615
Bezpečnost	617
Požadavky při křížení domén	618
Objekt XDomainRequest	619
Objekt XHR a požadavky mezi doménami	621
JSON	622
JSON a Ajax	624
Bezpečnost	628
Shrnutí	628
Kapitola 18: Pokročilé techniky	631
<hr/>	
Pokročilé funkce	631
Konstruktory bezpečné vzhledem k oboru platnosti	631
Lině načítané funkce	634
Svázání funkce	636
Curryfikace funkce	639

Pokročilé časovače	641
Opakující se časovače	644
Potlačené procesy	645
Přiškrcení funkce	647
Vlastní události	650
Přetahování prvků	653
Oprava implementace přetahování prvků	656
Přidávání vlastních událostí	658
Shrnutí	660
Kapitola 19: Úložiště na straně klienta	663
Soubory cookie	663
Omezení	664
Části souboru cookie	664
Soubory cookie v JavaScriptu	666
Podsoubory cookie	669
Další pokyny pro práci se soubory cookie	673
Uživatelská data v Internet Exploreru	674
Modul DOM Storage	675
Typ Storage	676
Objekt sessionStorage	676
Objekt globalStorage	678
Objekt localStorage	679
Typ StorageItem	680
Událost storage	681
Omezení	681
Shrnutí	681
Kapitola 20: Osvědčené postupy	683
Udržovatelnost	683
Co je udržovatelný kód	684
Konvence při psaní kódu	684
Volné propojení	687
Programátorské postupy	691
Výkon	696
Myslete na obor platnosti	696
Volte správný algoritmus	698
Minimalizujte počet příkazů	704
Optimalizace práce s modelem DOM	706
Nasazení	709
Proces sestavení	709
Validace	711
Kompresie	713

Shrnutí	716
Kapitola 21: Připravovaná aplikační rozhraní	717
Rozhraní API pro práci se selektory	717
Metoda querySelector()	718
Metoda querySelectorAll()	719
Podpora a budoucnost	720
Jazyk HTML 5	720
Vlastnosti pro práci se znakovou sadou	720
Doplňky související s třídami	721
Vlastní datové atributy	723
Výměna zpráv mezi dokumenty	724
Elementy pro práci s médii	726
Element canvas	730
Podpora provozu offline	740
Změny v historii	741
Databázové úložiště	741
Přetahování prvků	744
Typ WebSocket	748
Budoucnost jazyka HTML 5	749
Shrnutí	750
Kapitola 22: Vývoj JavaScriptu	751
ECMAScript 4 neboli JavaScript 2	751
JavaScript 1.5	752
JavaScript 1.6	754
JavaScript 1.7	757
JavaScript 1.8	763
JavaScript 1.9	765
Návrhy ECMAScriptu 4	765
Přiřazení typu proměnným	765
Definování typů	771
Třídy a rozhraní	775
Rozhraní	779
Dědičnost	780
Finální třídy a metody	781
Obory názvů	782
Baličky	783
Další změny jazyka	784
Budoucnost ECMAScriptu 4	790
ECMAScript 3.1	791
Změny interních prvků objektů	791
Statické metody objektů	792
Nativní podpora formátu JSON	800

Obsah

Desetinná čísla	803
Používané podmnožiny	806
Budoucnost ECMAScriptu 3.1	807

Shrnutí	807
----------------	------------

Příloha A: Knihovny JavaScriptu **809**

Všeobecné knihovny **809**

Yahoo! User Interface (YUI)	809
Prototype	810
Dojo Toolkit	810
MooTools	810
jQuery	810
MochiKit	811
Ext JS	811

Internetové aplikace **811**

Rico	811
qooxdoo	811

Animace a efekty **812**

script.aculo.us	812
moo.fx	812
Lightbox	812

Kryptografie **813**

JavaScript MD 5	813
JavaScript	813

Příloha B: Nástroje JavaScriptu **815**

Ladící nástroje **815**

Microsoft Script Debugger	815
Microsoft Script Editor	816
Visual Studio .NET	816
Visual Web Developer Express	816
Firebug	816
Venkman	817
Drosera	817
Web Inspector	817
Ladící nástroj prostředí Aptana	817
Validátory	818
JSLint	818
JavaScript Lint	818

Kompresní nástroje **818**

JSMin	818
Dojo ShrinkSafe	819
YUI Compressor	819

Testování jednotek	819
JsUnit	819
YUI Test	819
Dojo Object Harness (DOH)	820
Generátory dokumentace	820
JsDoc Toolkit	820
YUI Doc	820
AjaxDoc	821
Zabezpečená prováděcí prostředí	821
ADsafe	821
Caja	821
Rejstřík	823

Předmluva

JavaScript byl po většinu své existence předmětem obav, urážek, pohrdání a neporozumění. V prvních letech svého života se o něm řada „seriózních“ programátorů domnívala, že není dostatečně seriózní.

Naproti tomu spousta absolventů humanitních studií naverbovaných do řad webových vývojářů během internetového boomu považovala jazyk JavaScript za záhadný a tajemný. Mnozí z těch, kteří díky vytrvalosti a trpělivosti dospěli až k plnému porozumění JavaScriptu jakožto jazyku, byli nicméně frustrováni jeho nekonzistentní implementací v jednotlivých prohlížečích. Všechny tyto faktory pak dopomohly k rozbušení neohrabaných a špatně koncipovaných skriptů. A kvůli mimořádné otevřenosti klientského kódu na webových stránkách se řada těchto zlovyků kopírovala z jednoho webu na druhý. Tím se špatná reputace JavaScriptu jakožto jazyka, která byla obecně nezasloužená, propletla se zaslouženě špatnou reputací obklopující jeho implementace.

Kolem roku 2001 (s vydáním Internet Exploreru 6) si kvalitnější implementace v prohlížečích stále častěji podávala ruku s promyšlenějšími postupy při vývoji webů. Pomalu začal být objevován objekt XMLHttpRequest ležící v srdci technologie Ajax a v prohlížečích se vynořovalo nové paradigma interakce uživatelů ve stylu desktopových aplikací. Při tom se upevnilo aplikační rozhraní pro práci s modelem DOM, které dovolovalo JavaScriptu manipulovat se strukturami a obsahem webových dokumentů. Jazyk CSS i přes veškeré překroucení, zanedbání a svéhlavou nesmyslnost implementace ze strany výrobců prohlížečů dospěl do takové podoby, že již bylo možné novou interaktivní sílu Webu kombinovat s krásou a svižností. V důsledku toho se JavaScript stal předmětem nové vlny emocí, kterou lze charakterizovat překvapením, potěšením i respektem. Pokud si vzpomenete na chvíle, kdy jste v roce 2004 poprvé pracovali s webovou aplikací Google Maps, pak si tyto pocity jistě vybavíte.

Aplikace Google Maps patřila do rozvíjející se třídy aplikací, které braly programování na straně prohlížeče stejně vážně jako programování na straně serveru a které nás přiměly změnit pohled na aplikační plochu nabízenou webovým prohlížečem. (Webová aplikace Oddpost, která již v roce 2003 umožňovala pracovat s e-maily ve webovém prohlížeči podobně jako v aplikaci Outlook, byla v tomto směru dalším pozoruhodným průkopníkem.) Zvyšující se počet těchto aplikací společně

ně s rostoucím podílem prohlížečů, které je podporují, vedl k nefalšovanému obrození v projektování webových aplikací. Zrodil se „Web 2.0“ a Ajax se v této oblasti stal dominující technologií. Celý Web tak začal být znovu zajímavý. A také JavaScript, jakožto jediný programovací jazyk webu, proto stal mnohem zajímavějším.

JavaScript se sice opět dostal do popředí, ale jeho správné použití zůstalo poněkud nesnadné. Samotný jazyk včetně doprovodných aplikačních rozhraní v modelech DOM (Document Object Model) a BOM (Browser Object Model) byl nekonzistentně implementován, což značně a zbytečně ztěžovalo psaní kódu fungujícího ve všech prohlížečích. Profese projektování klientské části webových aplikací byla v té době poměrně mladá. Do učebních plánů vysokých škol však tento druh výuky proniká i nyní jen velmi pozvolna, pokud vůbec.

JavaScript, který byl na konci roku 2004 pravděpodobně nejvýznamnějším programovacím jazykem na světě, nebyl v akademickém smyslu předmětem zájmu číslo jedna. S probuzením webu do nového dne však vyvstala závažná otázka: Bude k dispozici dostatek vzdělaných, dobře připravených inženýrů, kteří by byli schopni postavit se těmto novým výzvám?

Do věci se vložili autoři odborné literatury a vzniklou mezeru vyplnili knihami o JavaScriptu. Za rok jich vycházely desítky, většina z nich však byla spíše zklamáním. Některé prosazovaly techniky, které se týkaly pouze starších prohlížečů, jiné pak propagovaly postupy, které se snadno zkopírují, ale hůře rozšiřují a udržují. Je s podivem, že u mnoha knih o JavaScriptu se zdálo, jako by je napsali lidé, kteří neměli jazyk JavaScript doopravdy rádi, kteří si nemysleli, že byste jej měli mít rádi také vy, a kteří ani nedoufali, že byste byli schopni si jej plně osvojit.

Jedna ze skutečně kvalitních knih z oblasti návrhu klientských aplikací se objevila v okamžiku, kdy Nicholas C. Zakas vydal v roce 2005 první edici knihy „Professional JavaScript for Web Developers“. V té době jsem společně se svými kolegy pracoval ve společnosti Yahoo! na vytvoření knihovny YUI (Yahoo! User Interface), představující základ pro návrh klientských aplikací a propagující osvědčené postupy v naší rodící se disciplíně. Každý pátek jsme se shromáždili v posluchárně, kde jsme diskutovali o návrhu klientských aplikací a vyučovali jazyk JavaScript, jazyk CSS a tvorbu webových aplikací v prohlížečích. Pečlivě jsme přehodnotili nabídku titulů, jež by pomohly novým inženýrům v osvojení způsobu budování robustních, na standardech založených a snadno udržovatelných webových aplikací za použití pokročilých technik JavaScriptu a modelu DOM. Ihned po svém vydání se Zakasova kniha stala naší učebnicí JavaScriptu.

A od té doby ji používáme. Ta kniha na nás tak zapůsobila, že jsme pana Zakase požádali, aby nám přišel do společnosti Yahoo! pomoci formovat komunitu kolem návrhu klientských aplikací.

To, co pan Zakas s knihou „Professional JavaScript for Web Developers“ dokázal, je naprosto fascinující: chopil se JavaScriptu jakožto látky, která je seriózní i přístupná. Jste-li programátor, pak se dozvíte, jak JavaScript zapadá do širšího spektra jazyků a paradigmat, které již znáte. Dozvíte se, jak je jeho systém dědičnosti a pravý dynamismus vskutku nekonvenční, ale také osvobozující a výkonný. Naučíte se cenit si JavaScriptu jakožto jazyka od kolegy programátora, který jej respektuje a rozumí mu.

Pokud patříte mezi absolventy humanitních studií vtažené do této profese v letech internetového boomu a pokud si chcete vyplnit mezery ve znalostech JavaScriptu, pak zjistíte, že pan Zakas je učitel, kterého jste si vždy přáli. Dokáže vám totiž pomoci v posunu od „jak to provést, aby věci fungovaly“ k „vytváření věcí, které fungují správně“. Nakonec budete mít seriózní znalosti o seri-

ózní látce. A co je nejdůležitější: zjistíte, že nijak nebazíruje na tom, jak hluboko byste měli do jazyka proniknout. Vše bere seriózně a trpělivým, přístupným způsobem vám pomáhá dělat to též.

V této druhé edici knihy, která prošla rozšířením, aktualizací a opravami, jsou vypuštěna některá témata, která již tolik nesouvisejí se současnou podobou profese, a zbytek knihy je doplněn o poznatky získané v letech 2005 až 2008. Tato léta byla velice důležitá a pan Zakas se v procesu výuky nachází v přední linii. Toto období věnoval výstavbě současné generace nejpoblárnějšího osobního portálu na webu (My Yahoo!) a další verze nejnavštěvovanější stránky tohoto webu (úvodní stránka portálu Yahoo!). Zkušenosti získané v těchto složitých, vysoce objemných aplikacích jsou vidět na každé stránce tohoto nového titulu, protože každá z nich prošla jedinečným filtrem pana Zakase ve formě učitele i autora.

V důsledku toho překračují jeho řešení hranice knižních dovedností a obsahují jistý druh praktických dovedností, které lze získat pouze každodenním programováním.

A to je opravdu dobrá zpráva pro všechny čtenáře, protože kniha, kterou právě držíte v ruce, je mnohem lepší, věcnější a důležitější než první vydání, a proto si právem zaslouží místo na vaší polici.

Eric Miraglia, Ph.D.
Sr. Engineering Manager (knihovna YUI)
Sunnyvale, California

Úvod

Někteří tvrdí, že JavaScript je v současnosti nejpobulárnějším programovacím jazykem na světě, který využívá bezpočet komplexních webových aplikací, na které se svět spoléhá v oblastech, jako je obchod, nákupy nebo správy procesů.

JavaScript je velmi volně založen na jazyku Java, což je objektově orientovaný jazyk zpopularizovaný pro použití na webu prostřednictvím vkládaných apletů. I když JavaScript využívá podobnou syntaxi a metody programování, nejedná se o „odlehčenou“ verzi Javy. JavaScript je totiž zcela samostatný dynamický jazyk, který si našel své místo ve webových prohlížečích po celém světě a který umožňuje bohatší interakci uživatelů na webových stránkách i webových aplikacích.

V této knize se budeme věnovat JavaScriptu od jeho prvopočátků v prvních prohlížečích Netscape až k současným verzím podporujícím model DOM a technologii Ajax. Naučíte se, jak si rozšířit jazyk JavaScript takovým způsobem, aby vyhovoval vašim specifickým potřebám, a jak vytvořit bezproblémovou komunikaci klienta a serveru bez prostředníků, jakou jsou aplety Java nebo skryté rámce. Zkrátka naučíte se, jak aplikovat řešení JavaScriptu na obchodní problémy, kterým musí čelit weboví vývojáři na celém světě.

Co v knize najdete?

Tato kniha nabízí úvod určený pro vývojáře spolu s pokročilejšími a užitečnými prvky JavaScriptu.

Na začátku knihy se podíváme na to, jak JavaScript vznikl a jak se vyvinul do současné podoby. Poté si podrobněji probereme ty části, které tvoří implementaci JavaScriptu, přičemž se zaměříme na standardy, jako je ECMAScript a model DOM (Document Object Model – objektový model dokumentu). Všimneme si též rozdílů mezi implementacemi JavaScriptu v jednotlivých oblíbených prohlížečích.

S těmito znalostmi se pak přesuneme k základním principům JavaScriptu, mezi něž patří jeho verze objektově orientovaného programování, dědičnosti a jeho použití nejrůznějších značkovacích

jazyků, jako je HTML. Po podrobném prostudování událostí a obsluh událostí následuje průzkum technik pro detekci prohlížečů a průvodce regulárními výrazy v JavaScriptu. Veškeré nabyté znalosti pak aplikujeme na tvorbu dynamického uživatelského rozhraní.

V poslední části knihy se soustředíme na pokročilá témata, mezi něž patří optimalizace výkonu a množství použité paměti, praxí osvědčené postupy a pohled na budoucnost JavaScriptu.

Komu je kniha určena?

Tato kniha je určena pro následující tři skupiny čtenářů:

- Zkušení vývojáři zběhlí v objektově orientovaném programování, kteří se chtějí naučit JavaScript v jeho souvislosti s tradičními objektově orientovanými jazyky, jako je Java a C++.
- Vývojáři webových aplikací, kteří se pokoušejí rozšířit použitelnost svých webových stránek a webových aplikací.
- Začínající vývojáři v jazyku JavaScript, kteří se snaží tomuto jazyku lépe porozumět.

Jasným signálem, že tato kniha je určena právě vám, je znalost některé z následujících souvisejících technologií:

- Java,
- PHP,
- ASP.NET,
- HTML,
- CSS,
- XML.

Tato kniha není určena pro začátečníky, kterým chybí základy informatiky nebo kteří chtějí na své webové stránky přidat nějaké prvky pro jednoduchou interakci s uživateli. Tito čtenáři by měli raději sáhnout po knize „JavaScript: Krok za krokem“ (<http://knihy.cpress.cz/K1615>).

Co potřebujete ke studiu témat této knihy

Ke spuštění ukázek v této knize potřebujete následující:

- Operační systém Windows 2000, Windows Server 2003, Windows XP, Windows Vista nebo Mac OS X.
- Prohlížeč Internet Explorer 6 nebo novější, Firefox 2 nebo novější, Opera 9 nebo novější, Chrome 0.2 nebo novější nebo Safari 2 nebo novější.

Kompletní zdrojové kódy pro ukázky jsou k dispozici ke stažení na adrese <http://knihy.cpress.cz/K1705>.

Uspořádání knihy

Knihu tvoří následující kapitoly:

- **Kapitola 1, Co je JavaScript:** Vysvětluje vznik JavaScriptu: odkud se vzal, jak se vyvíjel a jaký je dnes. Mezi představené koncepty patří vztah mezi JavaScriptem a ECMAScriptem,

model DOM (Document Object Model) a model BOM (Browser Object Model). Kapitola se věnuje též souvisejícím standardům sdružení ECMA (European Computer Manufacturers Association) a organizace W3C (World Wide Web Consortium).

- **Kapitola 2, JavaScript v jazyku HTML:** Zkoumá použití JavaScriptu spolu s jazykem HTML pro vytváření dynamických webových stránek. V této kapitole si představíme rozličné způsoby vkládání kódu JavaScriptu na stránku, přičemž se podíváme také na typ obsahu s kódem JavaScriptu a s ním související element `script`.
- **Kapitola 3, Základy jazyka:** Představuje základní jazykové koncepty, včetně syntaxe a příkazů pro řízení běhu programu. V této kapitole se podíváme na syntaktické podobnosti JavaScriptu s dalšími jazyky založeným na jazyku C a také na to, v čem se od nich liší. Představíme si nucené přetypování a jeho souvislost s vestavěnými operátory.
- **Kapitola 4, Proměnné, obor platnosti a paměť:** Zkoumá, jakým způsobem zachází JavaScript s proměnnými vzhledem k jejich volně typované povaze. Probereme si rozdíly mezi primitivními a odkazovanými hodnotami a souvislost prováděcího kontextu s proměnnými. V části o úklidu paměti v JavaScriptu si vysvětlíme, jak se uvolňuje paměť, když proměnné opouštějí obor platnosti.
- **Kapitola 5, Odkazované typy:** Věnuje se podrobnostem ohledně vestavěných odkazovaných typů JavaScriptu, mezi něž patří typy `Object` a `Array`. Každý odkazovaný typ popsany ve standardu ECMA-262 si probereme jak teoreticky, tak i po stránce jeho implementace v prohlížečích.
- **Kapitola 6, Objektově orientované programování:** Vysvětluje, jak se v JavaScriptu používá objektově orientované programování. JavaScript nezná pojem tříd, a proto si probereme několik technik pro tvorbu a dědičnost objektů. Kromě toho se podíváme na prototypy funkcí a na to, jak zapadají do celkového objektově orientovaného rámce jazyka.
- **Kapitola 7, Anonymní funkce:** Zkoumá jednu z nejsilnějších stránek JavaScriptu: anonymní funkce. Budeme se věnovat uzávěrům, způsobu fungování objektu `this`, vzoru modul a tvorbě soukromých členů objektu.
- **Kapitola 8, Objektový model prohlížeče:** Představuje model BOM (Browser Object Model), který je odpovědný za objekty umožňující interakci se samotným prohlížečem. Budeme se věnovat každému objektu modelu BOM, včetně objektů `window`, `document`, `location`, `navigator` a `screen`.
- **Kapitola 9, Detekce klienta:** Vysvětluje nejrůznější přístupy k detekci stroje klienta a jeho schopností. Mezi probírané techniky patří detekce schopností a detekce řetězce s údaji o agentu uživatele. V této kapitole se podíváme na klady a zápory a také na situační přiměřenost každého z přístupů.
- **Kapitola 10, Objektový model dokumentu:** Představuje objekty modelu DOM (Document Object Model) dostupné v JavaScriptu dle definice na úrovni DOM Level 1. Po stručném úvodu do jazyka XML a jeho vztahu k modelu DOM si podrobně prostudujeme celý model DOM a možnosti, které nabízí vývojářům pro manipulaci se stránkou.
- **Kapitola 11, Model DOM na úrovni 2 a 3:** Staví na předchozí kapitole a vysvětluje, jak úrovně DOM Level 2 a 3 rozšiřují model DOM o další vlastnosti, metody a objekty. Podíváme se také na problémy s kompatibilitou mezi Internet Explorerem a dalšími prohlížeči.

- **Kapitola 12, Události:** Vysvětluje povahu událostí v JavaScriptu, odkud se vzaly, podporu pro starší typy a jak model DOM redefinoval způsob fungování událostí. Budeme se věnovat rozličným zařízením, včetně zařízení Wii a iPhone.
- **Kapitola 13, Skriptování formulářů:** Zaměřuje se na použití JavaScriptu pro rozšíření možností interakce s formuláři a obcházení omezení v prohlížečích. Diskuse v této kapitole se zaměřuje na jednotlivé prvky formuláře, jako jsou textová a výběrová pole, na validaci dat a na manipulaci s daty.
- **Kapitola 14, Ošetření a ladění chyb:** Studuje, jak prohlížeče reagují na chyby v kódu JavaScriptu, a nabízí několik způsobů jejich ošetřování. Pro každý prohlížeč si představíme nástroje a techniky pro ladění, včetně doporučení pro zjednodušení ladicího procesu.
- **Kapitola 15, JavaScript a XML:** Prezentuje prvky JavaScriptu používané pro manipulaci s kódem jazyka XML (eXtensible Markup Language). V této kapitole si vysvětlíme rozdíly v podpoře a objektech v jednotlivých webových prohlížečích a představíme si několik návrhů pro snazší programování kódu fungujícího ve všech prohlížečích. Podíváme se také na jazyk XSLT (eXtensible Stylesheet Language Transformations) pro transformaci dat XML na straně klienta.
- **Kapitola 16, ECMAScript pro XML:** Probírá rozšíření JavaScriptu s názvem E4X (ECMAScript for XML), které je navrženo pro zjednodušení práce s jazykem XML. V této kapitole si vysvětlíme výhody rozšíření E4X oproti modelu DOM při manipulaci s kódem jazyka XML.
- **Kapitola 17, Ajax a JSON:** Zaměřuje se na běžné techniky Ajaxu, včetně použití objektu XMLHttpRequest a objektu Internet Exploreru s názvem XMLHttpRequest pro realizaci technologie Ajax fungující ve všech prohlížečích. V této kapitole se podíváme na rozdíly v implementaci a podpoře v jednotlivých prohlížečích a seznámíme se s několika doporučeními pro praktické použití.
- **Kapitola 18, Pokročilé techniky:** Noří se do některých z komplexnějších vzorů jazyka JavaScript, mezi něž patří curryfikace funkce, částečná aplikace funkce a dynamické funkce. V této kapitole se budeme též věnovat tvorbě vlastního rámce pro práci s událostmi, který poskytuje jednoduchou podporu událostí pro vlastní objekty.
- **Kapitola 19, Úložiště na straně klienta:** Probírá nejrůznější techniky pro ukládání dat na počítači klienta. V této kapitole se nejdříve zaměříme na nejčastěji podporovaný prvek, kterým jsou soubory cookie, a poté se budeme věnovat novějšímu funkčnímu prvku – úložišti modelu DOM.
- **Kapitola 20, Osvědčené postupy:** Zkoumá přístupy k práci s JavaScriptem v podnikovém prostředí. Podíváme se na techniky pro lepší udržitelnost, mezi něž patří techniky pro psaní kódu, formátování a obecné postupy při programování. Budeme se též věnovat výkonu při provádění kódu a představíme si několik technik pro optimalizaci rychlosti. Na závěr si projdeme problémy při nasazení, včetně způsobu vytvoření procesu sestavení.
- **Kapitola 21, Přípravovaná aplikační rozhraní:** Představuje aplikační rozhraní, která se vytvářejí, aby rozšířila JavaScript v prohlížečích. I když tato aplikační rozhraní dosud nejsou hotová nebo plně implementována, jsou již na obzoru a prohlížeče je již začaly částečně implementovat. V této kapitole se budeme věnovat také aplikačnímu rozhraní pro práci se selektory a jazyku HTML 5.

- **Kapitola 22, Vývoj JavaScriptu:** Dívá se do budoucnosti JavaScriptu na směr, kterým se bude tento jazyk ubírat. Budeme se věnovat verzím ECMAScript 3.1, ECMAScript 4 a ECMAScript Harmony.

Použité konvence

Pro snazší čitelnost a lepší přehlednost textu dodržujeme v celé knize několik zásad.

Rámečky obsahují důležité informace, které přímo souvisejí s okolním textem.

Rady, tipy, triky a odbočky od probíraného tématu jsou posunuty a vysázeny odlišným stylem písma.

Co se typografických konvencí týče:

- Nové pojmy a důležitá slova jsou při jejich prvním uvedení *zvýrazněna*.
- Kombinace klávesové jsou zapsány takto: Ctrl+A.
- Názvy souborů, adresy URL a kód uvnitř textu je uveden takto: `persistence.properties`.

Kód je uváděn dvěma způsoby:

Pro většinu kódu v příkladech používáme font s fixní šířkou.

Zvýraznění používáme pro zdůraznění kódu, který má v aktuálním kontextu zvláštní důležitost.

Zdrojový kód

Při práci na příkladech v této knize můžete buď veškerý kód psát ručně, nebo sáhnout po souborech se zdrojovým kódem, které doprovázejí tuto knihu. Všechny zdrojové soubory k této knize (v angličtině) najdete na adrese www.wrox.com. Na webu jednoduše vyhledejte název knihy (zadejte název do vyhledávacího pole nebo se podívejte do seznamu knih) a na stránce knihy klepněte na odkaz Download Code, čímž zahájíte stahování zdrojových kódů pro tuto knihu.

Protože může existovat více knih s podobným názvem, nejjednodušší je hledat knihu podle čísla ISBN. Originál této knihy má číslo ISBN 978-0-470-22780-0.

Po stažení archivu stačí zdrojové soubory jen rozbalit pomocí vašeho oblíbeného kompresního nástroje. Můžete také navštívit stránku www.wrox.com/dynamic/books/download.aspx, kde najdete zdrojové soubory nejen pro tuto knihu, ale také pro všechny knihy z nakladatelství Wrox.

Také nakladatelství Computer Press pro vás připravilo lokalizované zdrojové kódy použité v knize. Archiv s kódy a dalšími soubory je dostupný na adrese <http://knihy.cpress.cz/K1705>.

Errata

Vynakládáme veškeré úsilí, aby se v textu a v kódu neobjevovaly žádné chyby. Nikdo ovšem není dokonalý a člověk je tvor chybující. Najdete-li v některé z našich knih chybu, překlep či nefunkční část kódu, budeme vám velmi vděční za upozornění. Zašlete-li nám upozornění na takovou chybu, můžete ostatním čtenářům ušetřit hodiny frustrace a nám zase pomůžete zlepšit kvalitu předkládaných informací.

Stránku tiskových chyb této knihy najdete na adrese www.wrox.com. Zadejte název knihy do vyhledávacího pole nebo se podívejte do seznamu knih. Poté na stránce knihy klepněte na odkaz Book Errata. Na této stránce najdete všechny tiskové chyby odhalené čtenáři nebo redaktory nakladatelství Wrox. Úplný seznam knih obsahující odkazy na stránky tiskových chyb najdete na adrese www.wrox.com/misc-pages/booklist.shtml.

Pokud vámi nalezenou chybu na stránce tiskových chyb nenajdete, podívejte se na stránku www.wrox.com/contact/techsupport.shtml a vyplňte formulář, jehož prostřednictvím nám pošlete informaci o nalezené chybě. My ji zkontrolujeme, a bude-li oprávněná, umístíme zprávu na stránku tiskových chyb. V následujících vydáních knihy pak chybu automaticky opravíme.

Poznámka redakce českého vydání

Nakladatelství CPress, které pro vás tuto knihu přeložilo, stojí o zpětnou vazbu a bude na vaše podněty a dotazy reagovat. Můžete se obrátit na následující adresy:

Computer Press, redakce PC literatury
Spielberk Office Centre
Holandská 8
639 00 Brno

nebo

knihy@cpress.cz

Další informace a případné opravy českého vydání knihy najdete na internetové adrese <http://knihy.cpress.cz/k1705>. Prostřednictvím uvedené adresy můžete též naší redakci zaslat komentář nebo dotaz týkající se knihy. Na vaše reakce se srdečně těšíme.