

Předmluva	11
Spolupracovníci	13
Autoři kapitol	13
Odborní korektoři	15
Úvodem	17
Komu je tato kniha určena	17
Co se naučíte	17
Styl a konvence knihovny jQuery	18
Další možnosti	18
Případné problémy se zprovozněním příkladů	18
Jestliže se vám tato kniha líbí (nebo nelíbí)	19
Konvence používané v knize	20
Použití ukázkových zdrojových kódů	20
1. Základy knihovny jQuery	21
1.1 Úvod	21
1.2 Proč zvolit knihovnu jQuery	21
1.3 Filozofie knihovny jQuery	23
1.4 Uspořádání rozhraní API knihovny jQuery	26
1.5 Začlenění kódu knihovny jQuery do stránky HTML	28
1.6 Spouštění jQuery nebo JavaScriptu po načtení modelu DOM před kompletním načtením stránky	29
1.7 Výběr elementů modelu DOM pomocí selektorů a funkce jQuery	31
1.8 Výběr elementů modelu DOM v určitém kontextu	33
1.9 Filtrování obalené skupiny elementů modelu DOM	34
1.10 Hledání dceřiných elementů v aktuálně vybrané obalené skupině	36
1.11 Vrácení k předchozímu výběru před destruktivní změnou	37
1.12 Sloučení předchozího výběru s aktuálním	38
1.13 Získání nové skupiny elementů modelu DOM procházením modelu DOM podle aktuálního kontextu	39

1.14	Vytváření, manipulace a vkládání elementů modelu DOM	41
1.15	Odstraňování elementů modelu DOM	42
1.16	Nahrazování elementů modelu DOM	43
1.17	Klonování elementů modelu DOM	44
1.18	Získávání, nastavování a odstraňování atributů elementů modelu DOM	46
1.19	Získávání a nastavování obsahu stránky HTML	48
1.20	Získávání a nastavování textového obsahu	48
1.21	Používání aliasu \$ bez vzniku globálních konfliktů	49

2. Výběr elementů pomocí knihovny jQuery **51**

2.1	Úvod	51
2.2	Výběr pouze dceřiných elementů	52
2.3	Výběr určitých sourozenců	53
2.4	Výběr elementů podle indexového pořadí	55
2.5	Výběr právě animovaných elementů	56
2.6	Výběr elementů podle jejich obsahu	57
2.7	Výběr elementů podle toho, čemu neodpovídají	58
2.8	Výběr elementů na základě jejich viditelnosti	59
2.9	Výběr elementů podle atributů	59
2.10	Výběr formulářových elementů podle typu	61
2.11	Výběr elementu s určitými charakteristikami	62
2.12	Použití kontextového parametru	63
2.13	Tvorba vlastního filtrovacího selektoru	64

3. Pokročilejší techniky **67**

3.1	Úvod	67
3.2	Procházení výsledků výběru	67
3.3	Redukování vybrané skupiny na určitý element	69
3.4	Převedení objektu jQuery na holý objekt modelu DOM	72
3.5	Získání indexového pořadí elementu ve výběru	75
3.6	Vytváření jiného pole z existujícího pole	76
3.7	Provedení akce na podmnožině vybrané skupiny elementů	79
3.8	Nastavení knihovny jQuery tak, aby nekolidovala s ostatními knihovnami	81
3.9	Přidávání funkcí pomocí zásuvných modulů	84
3.10	Přesné určení použitého dotazu	86

4. Pomocné funkce knihovny jQuery **89**

4.1	Úvod	89
4.2	Detekce možností pomocí objektu jQuery.support	89
4.3	Procházení polí a objektů metodou jQuery.each	90
4.4	Filtrování polí metodou jQuery.grep	91
4.5	Procházení a úprava prvků polí metodou jQuery.map	92

4.6	Spojování dvou polí metodou jQuery.merge	93
4.7	Vyfiltrování duplicitních prvků pole metodou jQuery.unique	93
4.8	Testování funkcí zpětného volání metodou jQuery.isFunction	94
4.9	Odstranění bílých míst z textových řetězců nebo formulářových hodnot metodou jQuery.trim	95
4.10	Připojování objektů a dat elementům modelu DOM metodou jQuery.data	96
4.11	Rozšiřování objektů metodou jQuery.extend	97

5. Rychlejší, jednodušší a zábavnější

99

5.1	Úvod	99
5.2	To není knihovna jQuery, ale jazyk JavaScript	99
5.3	Proč někdy výraz \$(this) nefunguje?	100
5.4	Odstraňování přebytečného opakování	102
5.5	Formátování řetězového volání metod knihovny jQuery	104
5.6	Vypůjčení kódu z jiných knihoven	105
5.7	Vytvoření vlastního iterátoru	107
5.8	Přepínání hodnoty atributu	110
5.9	Vyhledávání úzkých míst	112
5.10	Ukládání objektů jQuery do mezipaměti	116
5.11	Psaní rychlejších selektorů	118
5.12	Rychlejší načítání tabulek	120
5.13	Programování se standardními cykly	122
5.14	Snižování počtu vyhledávání jmen	125
5.15	Rychlejší aktualizace modelu DOM pomocí vlastnosti innerHTML	128
5.16	Ladění a řetězová volání metod	129
5.17	Jde o chybu knihovny jQuery?	131
5.18	Sledování kódu v knihovně jQuery	132
5.19	Snížení počtu požadavků odeslaných na server	134
5.20	Nenarušující postupy při programování v JavaScriptu	136
5.21	Používání knihovny jQuery pro progresivní vylepšování	139
5.22	Tvorba přístupných stránek	141

6. Rozměry

145

6.1	Úvod	145
6.2	Zjišťování rozměrů okna a dokumentu	145
6.3	Zjišťování rozměrů elementu	146
6.4	Zjišťování posunu elementu	148
6.5	Posun elementu do zobrazované části	151
6.6	Určení, zda se element nachází v zobrazované části	153
6.7	Centrování elementu v zobrazené ploše	156
6.8	Absolutní umístění elementu na jeho aktuální pozici	156

6.9	Umístění elementu relativně k jinému elementu	157
6.10	Změna šablony kaskádových stylů podle šířky okna prohlížeče	158
7.	Efekty	161
7.1	Úvod	161
7.2	Posouvání, objevování a mizení elementů	163
7.3	Zobrazení elementů jejich posouváním nahoru	165
7.4	Tvorba vodorovné harmoniky	167
7.5	Současné posouvání a prosvítání elementů	170
7.6	Postupné provádění efektů	171
7.7	Zjištění, zda se elementy právě animují	173
7.8	Zastavování a obnovování animací	174
7.9	Používání vlastních metod průběhu pro efekty	175
7.10	Zakázání všech efektů	176
7.11	Používání pokročilých efektů knihovny jQuery UI	177
8.	Události	179
8.1	Úvod	179
8.2	Připojování obsluhující funkce k více událostem	180
8.3	Opětovné používání obsluhující funkce s různými daty	181
8.4	Uvolnění celé sady obsluhujících funkcí	183
8.5	Spouštění určitých obsluhujících funkcí	184
8.6	Předávání dynamických dat obsluhujícím funkcím	184
8.7	Co nejrychlejší přístup k elementu (před spuštěním události document.ready)	187
8.8	Zastavení cyklu, v němž se spouštějí obsluhující funkce	189
8.9	Získání správného elementu při použití vlastnosti event.target	191
8.10	Zabránění paralelnímu spuštění několika animací současně při události hover	192
8.11	Zprovoznění obsluhujících funkcí pro nově přidané elementy	194
9.	Pokročilé události	197
9.1	Úvod	197
9.2	Zprovoznění knihovny jQuery při dynamickém načítání	197
9.3	Zrychlení globálního spouštění událostí	198
9.4	Vytváření vlastních událostí	201
9.5	Umožnění obsluhujícím funkcím poskytnout potřebná data	203
9.6	Vytváření zásuvných modulů řízených událostmi	206
9.7	Přijímání oznámení o volání metod knihovny jQuery	210
9.8	Používání metod objektu pro naslouchání událostem	212

10. Vylepšování formulářů jazyka HTML bez pomocného kódu **215**

10.1	Úvod	215
10.2	Zaměření vstupu na textové pole při načtení stránky	216
10.3	Zakazování a povolování formulářových elementů	217
10.4	Automatický výběr přepínačů	220
10.5	Výběr a zrušení výběru všech zaškrtávacích polí pomocí speciálních odkazů	222
10.6	Výběr a zrušení výběru všech zaškrtávacích polí pomocí jediného přepínače	223
10.7	Přidávání a odstraňování položek v seznamu	225
10.8	Automatické přecházení mezi elementy podle počtu zadaných znaků	227
10.9	Zobrazení zbývajících počtu znaků	229
10.10	Omezení textového pole na určité znaky	231
10.11	Odeslání formuláře technologií Ajax	232
10.12	Validace formulářů	234

11. Vylepšování formulářů jazyka HTML pomocí zásuvných modulů **241**

11.1	Úvod	241
11.2	Validace formulářů	242
11.3	Vytváření maskovaných vstupních polí	251
11.4	Automatické dokončování textových polí	252
11.5	Výběr rozsahu hodnot	254
11.6	Zadávání hodnoty z určitého rozsahu	256
11.7	Nahrávání souborů na server na pozadí	258
11.8	Omezení délky textových polí	260
11.9	Zobrazení popisků nad vstupními poli	261
11.10	Zvětšování vstupního pole podle jeho obsahu	262
11.11	Výběr data	263

12. Zásuvné moduly knihovny jQuery **267**

12.1	Úvod	267
12.2	Kde se nacházejí zásuvné moduly knihovny jQuery	267
12.3	Kdy napsat vlastní zásuvný modul knihovny jQuery	269
12.4	Vytvoření našeho prvního zásuvného modulu knihovny jQuery	270
12.5	Předávání možností nastavení našemu zásuvnému modulu	272
12.6	Používání zkráceného zápisu \$ v našem zásuvném modulu	273
12.7	Vkládání soukromých funkcí do našeho zásuvného modulu	275
12.8	Podpora zásuvného modulu Metadata	276
12.9	Přidání statické funkce k našemu zásuvnému modulu	278
12.10	Testování jednotek našeho modulu pomocí systému QUnit	280

13.Vytváření vlastních komponent rozhraní **283**

13.1	Úvod	283
13.2	Vytváření vlastní bublinové nápovědy	284
13.3	Navigace pomocí stromové nabídky	289
13.4	Rozbalování harmoniky	293
13.5	Procházení dokumentu pomocí záložek	297
13.6	Zobrazení jednoduchého modálního okna	300
13.7	Vytváření rozevíracích nabídek	307
13.8	Prolínání obrázků	310
13.9	Posouvací panely	314

14.Uživatelská rozhraní s knihovnou jQuery UI **319**

14.1	Úvod	319
14.2	Začlenění celé knihovny jQuery UI	321
14.3	Začlenění jednoho nebo dvou zásuvných modulů knihovny jQuery UI	322
14.4	Inicializace zásuvného modulu knihovny jQuery UI s výchozími možnostmi nastavení	323
14.5	Inicializace zásuvného modulu knihovny jQuery UI s vlastními možnostmi nastavení	324
14.6	Vytváření vlastních výchozích možností nastavení pro zásuvný modul knihovny jQuery UI	325
14.7	Získávání a nastavování možností nastavení pro zásuvný modul knihovny jQuery UI	327
14.8	Volání metod zásuvného modulu knihovny jQuery UI	327
14.9	Obsluha událostí zásuvného modulu knihovny jQuery UI	328
14.10	Zrušení zásuvného modulu knihovny jQuery UI	330
14.11	Vytvoření hudebního přehrávače pomocí knihovny jQuery UI	331

15.Motivy knihovny jQuery UI **343**

15.1	Úvod	343
15.2	Úprava vzhledu ovládacích prvků knihovny jQuery UI pomocí nástroje ThemeRoller	347
15.3	Přepisování rozvržení a pravidel stylů v motivech knihovny jQuery UI	359
15.4	Aplikace motivu z nástroje ThemeRoller na ovládací prvek GUI, jenž nepochází z knihovny jQuery UI	369
15.5	Odkazování na více motivů v jediné stránce	376
15.6	Příloha – další zdroje informací o jazyce CSS	385

16.Knihovna jQuery, technologie Ajax a datový formát HTML, XML, JSON a JSONP **387**

16.1	Úvod	387
16.2	Knihovna jQuery a technologie Ajax	387
16.3	Používání technologie Ajax na celém webu	390
16.4	Používání jednoduchých ajaxových požadavků se zpětnou odezvou uživateli	392
16.5	Používání zkrácených ajaxových metod a datových typů	396
16.6	Použití úryvků kódu jazyka HTML v knihovně jQuery	398

16.7	Převod textového řetězce s kódem jazyka XML na objekt modelu DOM	399
16.8	Vytváření dat ve formátu JSON	400
16.9	Analýza dat ve formátu JSON	401
16.10	Používání dat ve formátu JSONP v knihovně jQuery	402
17. Používání knihovny jQuery ve velkých projektech		405
17.1	Úvod	405
17.2	Použití úložiště na straně klienta	405
17.3	Uložení stavu aplikace pro jedinou relaci	408
17.4	Uložení stavu aplikace mezi relacemi	410
17.5	Použití šablonového subsystému jazyka JavaScript	411
17.6	Řazení ajaxových požadavků do fronty	414
17.7	Ajax a tlačítko Zpět	416
17.8	Vložení kódu jazyka JavaScript na konec stránky	418
18. Testování jednotek		421
18.1	Úvod	421
18.2	Automatické testování jednotek	421
18.3	Výsledky tvrzení	423
18.4	Testování synchronních funkcí zpětného volání	424
18.5	Testování asynchronních funkcí zpětného volání	425
18.6	Testy uživatelských akcí	426
18.7	Zachování atomičnosti testů	427
18.8	Seskupování testů	428
18.9	Výběr testů ke spuštění	429
Rejstřík		431

Předmluva

Když jsem v roce 2005 začal pracovat na knihovně jQuery, měl jsem jednoduchý cíl – chtěl jsem umět psát webové aplikace, které by fungovaly ve všech moderních webových prohlížečích bez dalšího ladění a oprav chyb. O několik měsíců později jsem dokončil sadu nástrojů, se kterými jsem mohl tohoto cíle dosáhnout pro osobní účely. Myslel jsem si, že už jsem téměř u konce, vůbec jsem ale netušil, že moje práce teprve začíná.

Postupně se knihovna jQuery rozrůstala a začalo ji ve svých projektech používat stále více programátorů. Díky tomu se stal vývoj JavaScriptové knihovny o dost složitější – zatímco je poměrně jednoduché vytvořit knihovnu pro osobní účely nebo určitou aplikaci, je neuvěřitelně těžké vyvíjet knihovnu, která bude fungovat v co největším množství prostředí (staré webové prohlížeče, zastaralé webové stránky a hojně se vyskytující podivné značky). Přestože se knihovna jQuery naučila s těmito případy vypořádat, zůstala překvapivě většina původního rozhraní API nezměněna.

Jednou věcí, která je pro mě zajímavá, je sledovat vývojáře, jak používají jQuery svým vlastním způsobem. Trochu mě překvapilo, kolik návrhářů a jiných lidí, kteří neprogramují, shledalo knihovnu jQuery zajímavou. Když jsem viděl, jak s touto knihovnou komunikují, uvědomil jsem si, jak je její rozhraní API jednoduše navrženo. O tom mě přesvědčilo i to, když jsem zjistil, kolik zkušených programátorů používá jQuery k vývoji velkých a složitých aplikací. Nejlepší na tom však je, že se můžu učit od všech, co knihovnu používají.

Další výhodou knihovny jQuery je její rozšiřitelná struktura zásuvných modulů. Když jsem začínal s jejím vývojem, byl jsem si jistý, že musím najít jednoduchá řešení umožňující vývojářům rozšiřovat její aplikační rozhraní. Nakonec z toho vznikla velmi rozsáhlá a rozmanitá komunita – různí vývojáři vydávají rozličné zásuvné moduly pro všechny možné účely. Za růst knihovny jQuery může právě tato komunita, bez níž by knihovna jistě nebyla tím, čím je dnes. Velmi mne proto těší, že tato kniha obsahuje kapitoly věnované některým nejzajímavějším zásuvným modulům. Nejlepším způsobem, jak rozšířit své povědomí o tom, co lze s knihovnou jQuery vytvořit, je učit se od této komunity a používat její zdrojové kódy.

Právě díky tomu jsou kuchařky tohoto typu tak zajímavé. Ty nejzajímavější věci, které se lidé naučili každodenní praxí, totiž uvádějí srozumitelným způsobem, takže se z nich dá později těžit. Myslím si, že tyto kuchařky představují nejlepší způsob, jak ověřit své znalosti jazyka nebo knihovny. Mám rád, když vidím, že někdo vzal aplikační rozhraní, o kterém jsem si myslel, že jej znám dokonale, a začal jej používat zcela novým způsobem. Doufám, že vám bude tato kniha dobře sloužit a naučí vás, jak používat knihovnu jQuery novými a zajímavými způsoby.

John Resig

tvůrce a vedoucí vývojář knihovny jQuery

Autoři kapitol

Jonathan Sharp se začal zabývat Internetem a vývojem webových aplikací okolo roku 1996. V následujících letech pracoval pro začínající firmy a pro klienty společnosti Fortune 500. Jonathan založil společnost Out West Media, která poskytuje služby v oblasti návrhu a vývoje aplikací se zaměřením na technologie XHTML, CSS a jQuery. Jonathan je členem týmu pro vývoj knihovny jQuery, a když zrovna neprogramuje, tak píše publikace nebo prezentuje. Jonathan děkuje především své ženě Erin, dceři Noel, dvěma psům a dvěma koním.

Rob Burns vyvíjí interaktivní webové aplikace ve firmě A Mountain Top. V uplynulých 12 letech studoval vývoj webových stránek pomocí široké škály nástrojů a technologií. Ve volném čase se věnuje zpracování přirozeného jazyka a řadě softwarových projektů s otevřeným zdrojovým kódem.

Rebecca Murphey je nezávislá konzultantka prezentačních architektur, vytváří vlastní prezentační řešení, která slouží jako pojítka mezi serverem a prohlížečem. Rovněž poskytuje školení v oblasti vývoje prezentací s důrazem na knihovnu jQuery. Žije se svým přítelem, dvěma psy a dvěma kočkami ve městě Durham, ve státě Severní Karolína.

Ariel Flesler je webový programátor a programátor videoher. Ke knihovně jQuery přispívá od ledna 2007 a do týmu pro vývoj jQuery se přidal v květnu 2008. Má 23 let a narodil se v Buenos Aires, hlavním městě Argentiny. Studuje na univerzitě National Technological University (Argentina) a doufá, že od roku 2010 bude pracovat jako systémový analytik a v roce 2012 se stane systémovým inženýrem. Začal pracovat jako programátor na platformě ASP.NET (C#) a potom se přeorientoval na vývoj klientských částí XHTML stránek a Ajaxových aplikací. Momentálně pracuje ve společnosti QB9, kde vyvíjí hry pro volný čas a online hry postavené na jazyku AS3.

Cody Lindley je křesťan, manžel, syn, otec, bratr, nadšený turista a profesionální programátor klientských částí webových aplikací. Od roku 1997 se zajímá o HTML, CSS, JavaScript, Flash, interaktivní design, návrh rozhraní a interakci člověka s počítačem (HCI). V komunitě knihovny jQuery je známý především díky vytvoření ThickBox, což je řešení pro dialogová a modální okna. V roce 2008 se oficiálně připojil k týmu pro vývoj knihovny jQuery. V současné době se zaměřuje na techniky optimalizace na straně klienta a také na psaní a přednášení o knihovně jQuery. Jeho webové stránky najdete na adrese <http://www.codylindley.com>.

Remy Sharp je programátor, autor, řečník a blogger. Remy zahájil svou profesionální kariéru webového programátora v roce 1999 jako samostatný vývojář s orientací na finanční webové stránky, díky čemuž se seznámil se všemi aspekty provozu webových stránek během internetové-

ho boomu a také dlouho po něm. Dnes řídí vlastní firmu Left Logic v Brightonu v Anglii, programuje v JavaScriptu, jQuery, HTML 5, CSS, PHP, Perl a ve všem ostatním, co se mu dostane pod ruku.

Mike Hostetler je vynálezce, podnikatel, programátor a hrdý otec. Pracuje s webovými technologiemi od poloviny 90. let, má bohaté zkušenosti s vývojem webových aplikací v PHP a JavaScriptu. Momentálně pracuje ve vedení společnosti A Mountain Top, webové konzultační společnosti ve městě Denver (stát Colorado). Mike se intenzivně věnuje open-source projektům, je členem týmu pro vývoj knihovny jQuery, řídí projekt QCubed PHP5 Framework a podílí se na projektu Drupal. Když zrovna nesedí před obrazovkou počítače, chodí na výlety, rybaří, jezdí na snowboardu nebo tráví volný čas se svou rodinou.

Ralph Whitbeck je absolventem univerzity Rochester Institute of Technology a v současnosti pracuje jako senior programátor ve společnosti BrandLogic ve městě Rochester ve státě New York. Ve firmě BrandLogic má na starost návrh rozhraní, testování použitelnosti a vývoj webových a desktopových aplikací. Ralph umí naprogramovat komplexní webové aplikace s využitím technologií ASP.NET, C#, SQL Server na straně serveru a s technologiemi XHTML, CSS a JavaScript/jQuery na straně klienta. Ralph se oficiálně přidal k týmu pro vývoj knihovny jQuery v říjnu roku 2009. Ralph rád tráví čas se svou manželkou Hope a svými třemi syny – Brandonem, Jordanem a Ralphiem. Více se o Ralphovi dočtete na jeho osobním blogu na adrese <http://ralphwhitbeck.com/>.

Nathan Smith je veselý člověk, který programuje webové stránky od konce minulého století. Rád programuje v HTML, CSS a JavaScriptu. Také fušuje do designu a informační architektury. Psal pro elektronické i tištěné publikace, jako je například Adobe Developer Center, Digital Web nebo .NET Magazine. Přednášel na mnoha konferencích, mezi něž patří Adobe MAX, BibleTech, Drupal Camp, Echo Conference, Ministry 2.0, Refresh Dallas a Webmaster Jam Session. Nathan pracuje pro společnost Fellowship Technologies jako vývojář v oblasti uživatelských zkušeností. Je držitelem titulu Master of Divinity ze semináře Asbury Theological Seminary. Stál u zrodu projektu Godbit.com, což je komunita, která se zaměřuje na pomoc kostelům a klášterům a pomáhá jim zlepšovat použitelnost webových stránek. Vytvořil systém 960 Grid System, což je framework pro náčrt, návrh a programování rozvržení stránek.

Brian Cherne je softwarový vývojář s více než desetiletou zkušeností v oblasti navrhování a sestavování webových aplikací, informačních kiosků a e-shopů s vysokou návštěvností. Rovněž je autorem zásuvného modulu hoverInvent pro knihovnu jQuery. Když neprogramuje, tak jej můžete najít tančícího v tanečním sále, jak cvičí bojová umění nebo jak studuje ruskou kulturu a jazyk.

Jörn Zaefferer je profesionální softwarový vývojář z Cologne v Německu. Vytváří aplikační programová rozhraní (API), grafická uživatelská rozhraní (GUI), softwarové architektury a databáze pro webové i desktopové aplikace. Při práci se zaměřuje na platformu Java, přičemž skriptování na straně klienta se točí kolem knihovny jQuery. Ke knihovně jQuery začal přispívat v polovině roku 2006 a od té doby spolupracoval na projektu QUnit, což je framework pro testování jednotek knihovny jQuery. Dále vydal a udržuje půl tuctu oblíbených zásuvných modulů pro knihovnu jQuery a rovněž přispíval do knih o jQuery – ať už jako autor, nebo jako odborný korektor. Je též vedoucím programátorem knihovny jQuery UI.

James Padolsey je nadšeným webovým vývojářem a blogerem z Londýna (Anglie). Šíleně obdivuje knihovnu jQuery od první chvíle, co jí spatřil – napsal pro ni spoustu tutoriálů, článků, příspěvků a zásuvných modulů. James plánuje získat titul na univerzitě University of Kent a věnovat se kariéře, která bude stále posouvat hranice jeho schopností. Jeho webové stránky najdete na adrese <http://james.padolsey.com>.

Scott González je vývojář webových aplikací žijící v Raleigh (Severní Karolína). Zajímá se o vytváření velmi dynamických systémů a flexibilních, škálovatelných frameworků. Ke knihovně jQuery přispívá od roku 2007 a v současnosti řídí vývoj jQuery UI, což je oficiální knihovna uživatelského rozhraní postavená na knihovně jQuery. Scott rovněž píše tutoriály o jQuery a jQuery UI na webových stránkách *nemikor.com* a přednáší o knihovně jQuery na konferencích.

Michael Gray začal programovat, když úprava kódu obnášela děrování štítků a dodržování standardů znamenalo řídit se standardem ECMA-10 pro výměnu dat na děrných štítcích. Dnes je Michael webovým programátorem a programátorem pro systém Android, který rád píše rychlý, čistý a jednoduchý kód a rád odpovídá ostatním programátorům na jejich e-mailové dotazy ohledně knihovny jQuery. Mezi jeho nedávné projekty patří výsledky voleb 2008 a informační mapy o voličích pro společnost Google, dále pak StrataLogic, což je kombinace běžných nástěnných map a atlasů založená na Google Earth. Jeho webové stránky najdete na adrese <http://mg.to>.

Maggie Wachs, Scott Jehl, Todd Parker a Patty Toland společně tvoří skupinu Filament Group. Dohromady navrhují a vyvíjejí funkčně bohatá uživatelská rozhraní pro spotřebitelské i firemní webové stránky, bezdrátová zařízení a desktopové i webové aplikace, přičemž se zaměřují na intuitivnost, použitelnost a přístupnost. Sponzorují a řídí návrh knihovny jQuery, pro kterou navrhli a naprogramovali aplikaci ThemRoller.com. Rovněž aktivně přispívají k neustálému vývoji oficiální knihovny jQuery UI a frameworku CSS Framework.

Richard D. Worth je vývojářem webových uživatelských rozhraní. Zastává funkci release manažera pro knihovnu jQuery UI a je také jedním z nejdéle přispívajících vývojářů. Je autorem nebo spoluautorem zásuvných modulů Dialog, Progressbar, Selectable a Slide. Richard rád hovoří o jQuery a jQuery UI po celém světě. Richard si užívá chvíle se svou rozrůstající se rodinou v Severní Virginii. Se svou milovanou ženou Nancy mají tři nádherné děti – Naomi, Aster a Isaiah. Richardovy webové stránky najdete na adrese <http://rdworth.org/>.

Odborní korektoři

Karl Swedberg po vystudování střední školy upravoval materiály pro reklamní agenturu, vlastnil kavárnu a před čtyřmi roky zahájil svou kariéru webového vývojáře. Nyní pracuje pro společnost Fusionary Media ve městě Grand Rapids ve státě Michigan, kde se specializuje na skriptování na straně klienta a na návrh interakce s uživatelem. Karl je členem týmu pro vývoj knihovny jQuery a spoluautor knih Learning jQuery 1.3 a jQuery Reference Guide. Některé jeho tipy a tutoriály najdete na adrese <http://www.learningjquery.com>.

Dave Methvin je chief technology officer ve společnosti PC Pitstop a je jedním ze zakládajících členů této společnosti. Knihovnu jQuery používá od roku 2006, aktivně pomáhá odpovídat na otázky ohledně jQuery a přispěl několika oblíbenými zásuvnými moduly, mezi něž patří moduly Corner a Splitter. Než začal pracovat pro společnost PC Pitstop, pracoval jako odpovědný redaktor v časopisech PC Tech Journal a Windows Magazine, kde psal články o jazyku JavaScript. Stále píše články pro několik webů věnujících se osobním počítačům, jako je například InformationWeek. Dave je držitelem bakalářského a magisterského titulu v oboru informačních technologií z univerzity University of Virginia.

David Serduke je programátorem prezentací, který poslední dobou tráví příliš času programováním na straně serveru. Po několikaleté zkušenosti s programováním začal na konci roku 2007 používat knihovnu jQuery a krátce na to se připojil k týmu pro její vývoj. David v současnosti vytváří webové stránky pro finanční instituce a přináší výhody knihovny jQuery do podnikových aplikací postavených na technologii ASP.NET. David žije v Severní Karolině, kde získal

bakalářský titul na univerzitě University of Carolina v oboru elektrotechnika a titul MBA na univerzitě St. Mary's College.

Scott Mark je architektem podnikových aplikací ve firmě Medtronic. Pracuje na webových informačních portálech a transakčních aplikacích, přičemž dbá na udržení vysoké míry použitelnosti v regulovaných prostředích. Momentálně se zajímá především o RIA aplikace a technologie více-dotykového uživatelského rozhraní. Scott žije v Minnesotě se svou milovanou ženou, dvěma syny a černým labradorem. O počítačích bloguje na webu <http://scottmark.wordpress.com> a o běhání na dlouhé tratě píše na adrese <http://runlikemonkey.com>.

Knihovna jQuery zasáhla svět vývoje prezentací jako blesk z čistého nebe. Její jednoduchý zápis dělá i z těch nejsložitějších úloh úlohy triviální, dokonce až zábavné. Spoustu programátorů rychle oslovila její elegance a čistota. Pokud používáte tuto knihovnu, jistě již k vašim projektům přidáváte bohatý a interaktivní obsah.

Začít s knihovnou jQuery je velmi snadné, ale jak už to s řadou nástrojů pro vývoj webových stránek bývá, jejich plnou sílu pochopíte až po měsících či letech jejich používání. Knihovna je plná funkcí, o kterých jste dříve možná neměli nejmenší tušení. Jakmile je poznáte, může se radikálně změnit způsob, jakým řešíte problémy.

Cílem této knihy je zprostředkovat vám ukázky práce předních programátorů, kteří používají knihovnu jQuery každý den ve svých projektech. V průběhu 18 kapitol vás provedou řešeními nejrůznějších problémů, od jednoduchých až po ty složité. Ať už jste začátečník, nebo zkušený veterán v JavaScriptu, určitě oceníte pohled na plnou sílu knihovny jQuery k tvorbě působivých, robustních a efektivních uživatelských rozhraní.

Komu je tato kniha určena

Možná jste návrhář, kterého zaujala interaktivita, jakou může knihovna jQuery nabídnout. Možná jste vývojář webových prezentací, který se již s knihovnou jQuery setkal a chce vidět, jak s ní jiní lidé řeší běžné úlohy. Možná píšete kód na straně serveru, který musí často psát skripty na straně klienta.

Po pravdě řečeno, tato kniha bude užitečná pro každého, kdo pracuje (nebo chce pracovat) s knihovnou jQuery. Jestliže s knihovnou jQuery začínáte, měli byste si také přečíst knihu „Learning jQuery 1.3“ (vydavatelství Packt) nebo „jQuery in Action“ (vydavatelství Manning). Pokud už jQuery ve svých projektech používáte, tato kniha by měla rozšířit vaše znalosti funkcí této knihovny a objevit její skryté klenoty a výstřednosti.

Co se naučíte

Začnete základy a obecnými, praxí ověřenými postupy, mezi něž patří začlenění kódu knihovny jQuery do stránky, provádění výběrů a procházení a manipulace. Dokonce i zkušení uživatelé knihovny jQuery zde jistě najdou nejednu inspiraci. Od základů se přesunete k praktickým případům užití, uvidíte osvědčená řešení na časté problémy, například s událostmi, efekty, rozměry, formuláři a prvky uživatelského rozhraní (s i bez pomoci knihovny jQuery UI). Nakonec zjistíte, jak testovat jQuery aplikace a jak začlenit jQuery do komplexních webových stránek.

Mimo jiné se také naučíte, jak s jQuery řešit složité problémy. Objevíte, jak vytěžit maximum ze systému správy událostí knihovny jQuery, včetně vlastních událostí a vlastních dat událostí, jak progresivně vylepšovat formuláře, jako pozicovat a přemísťovat elementy na stránce, jak vytvářet elementy uživatelského rozhraní, jako jsou například záložky, harmoniky a modální okna, jak psát čitelný a udržitelný kód, jak optimalizovat kód pro snadné testování, jak eliminovat úzká místa, jak zajistit maximální efektivitu, a tak dále.

Protože tato kniha není manuálem, můžete si sami vybrat, jaké recepty chcete číst. Celkově však kniha nabízí letmý pohled na jedny z nejlepších řešení problémů od komunity kolem knihovny jQuery. Z tohoto důvodu doporučujeme knihu alespoň prolétnout od začátku do konce, protože nikdy nevíte, kdy vás nějaký řádek kódu překvapí a přivede vás tak k rozhodnutí, že nastal čas, abyste své znalosti pozvedli na další úroveň.

Styl a konvence knihovny jQuery

Knihovna jQuery klade důraz na *řetězení*, což je postupné volání metod, kde každá metoda vrací nějaký výběr elementů, se kterými můžete dále pracovat. Tento princip popisuje podrobněji kapitola 1. Pokud s knihovnou jQuery začínáte, musíte nejprve pochopit tento princip, protože se v následujících kapitolách hojně používá.

Funkce knihovny jQuery jsou rozděleny do jednoduchých kategorií – hlavní funkce, výběr, manipulace, procházení, kaskádové styly, atributy, události, efekty, Ajax a nástroje. Jestliže budete znát tyto kategorie a s nimi související metody, snáze pochopíte látku probíranou v této knize.

V této knize si popíšeme jeden z nejdůležitějších, praxí osvědčených postupů, který spočívá v ukládání výběrů elementů do proměnné, což je výhodnější, než opakovat stejný výběr vícekrát. Když ukládáte výběr do proměnné, je vhodné, aby proměnná začínala znakem \$, který značí, že jde o objekt knihovny jQuery. Díky tomu bude kód čitelnější a lépe udržitelný. Na druhou stranu byste si ale měli uvědomit, že název proměnné začínající znakem \$ je jen konvence, která nemá žádný speciální význam, jako tomu je v třeba jazycích typu PHP.

V příkladech zdrojových kódů v této knize upřednostňujeme čitelnost a srozumitelnost před stručností, proto se mohou některé příklady zdát upovídanější, než je nutné. Pokud zjistíte, že některý kód jde optimalizovat, neváhejte a udělejte to. Samozřejmě byste rovněž měli dbát na srozumitelnost a čitelnost svého kódu a pro ostrý kód používat nástroje pro redukování velikosti kódu.

Další možnosti

Pokud hledáte další zdroje o knihovně jQuery, můžete nahlédnout do následujících publikací:

- *Learning jQuery 1.3* od Jonathana Chaffera, Karla Swedberga a Johna Resiga (vydavatelství Packt).
- *jQuery in Action* od Beara Bibeaulta, Yehuda Katze a Johna Resiga (vydavatelství Manning).
- *jQuery UI 1.6: The User Interface Library for jQuery* od Dana Wellmana (vydavatelství Packt).

Případné problémy se zprovozněním příkladů

Než začnete dělat cokoliv jiného, podívejte se, jestli na stránce načítáte knihovnu jQuery – divili byste se, kolikrát to pomůže, když něco nefunguje. Jestliže spolu s jQuery používáte další knihovny, budete možná muset použít metodu `jQuery.noConflict()`. Pokud používáte skripty založené na knihovně jQuery, ujistěte se, že je načítáte až po knihovně jQuery.

Spousta příkladů v této knize vyžaduje, aby byl dokument před svou interakcí s JavaScriptem tzv. připraven. Jestliže jste svůj kód vložili do záhlaví dokumentu, ujistěte se, že je uzavřen do funkce `$(document).ready(function() { ... });`, díky které skript počká, až bude dokument připraven.

Některé funkce v této knize jsou k dispozici jen ve verzi jQuery 1.3 nebo starší. Jestliže jste aktualizovali starší verzi knihovny jQuery, ujistěte se, že jste aktualizovali i případné zásuvné moduly – zastaralé zásuvné moduly se mohou chovat nepředvídatelně.

Jestliže nějaký příklad ve vaší stávající aplikaci nefunguje, zkuste před jeho začleněním do svého zdrojového kódu, zda funguje samostatně. Pokud příklad funguje samostatně, zkuste nějakým ladicím nástrojem, jako je například Firebug v prohlížeči Firefox, odhalit příčinu problému.

Pokud vkládáte redukovanou verzi knihovny jQuery a objeví se chyba, která poukazuje na to, že ji způsobila samotná knihovna, měli byste pro odladění této chyby použít její neredukovanou verzi. Snadněji tak odhalíte, jaký řádek způsobil chybu, což vás často navede směrem k jejímu řešení.

Jestliže si stále nevíte rady, zkuste poslat otázku do diskusního fóra na adrese <http://forum.jquery.com/>. Řada autorů této knihy zde pravidelně přispívá a vždy se najde někdo, kdo vám bude schopen poskytnout užitečnou radu. Dalším cenným zdrojem pro řešení problémů může být IRC kanál #jquery na Freenode.

Pokud nic z toho nepomůže, je možné, že jsme udělali chybu. Veškerý zdrojový kód v této knize jsme poctivě testovali, ale je možné, že nám nějaké chyby unikly. Zkontrolujte errata (popsaná v následující části) a stáhněte si zdrojový kód s příklady, který bude aktualizován dle chyb, které objevíme.

Jestliže se vám tato kniha líbí (nebo nelíbí)

Podělte se s námi, zda se vám tato kniha líbí. Recenze knihy na Amazonu je dobrým způsobem, jak vyjádřit své nadšení (nebo zklamání). Dále můžete využít možnosti ohodnotit knihu přímo na její webové stránce (<http://knihy.cpress.cz/K1840/>). Zde také najdete odkaz na errata.

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

Redakce počítačové literatury

Computer Press, a. s.

Spielberk Office Centre

Holandská 8

639 00 Brno

Nebo na e-mail:

sefredaktor.pc@cpress.cz

Konvence používané v knize

Kniha používá následující typografické konvence:

Kurzíva

Označuje internetové adresy, například doménová jména nebo URL adresy a nové termíny v místě jejich definice.

Písmo s konstantní šířkou

Označuje příkazy a výrazy, které by měly být zapsány doslovně; názvy a klíčová slova programů, včetně názvů metod, proměnných a tříd; značky elementů jazyka HTML, přepínače, atributy, klíče, funkce, typy, jmenné prostory, moduly, vlastnosti, parametry, hodnoty, objekty, události, obsluhy událostí, makra, obsahy souborů a výstupy příkazů.

Tučné písmo s konstantní šířkou

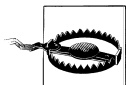
Označuje zvýrazněnou část ve zdrojovém kódu.

Skloněné písmo s konstantní šířkou

Označuje text, který by měl být před použitím nahrazen konkrétními hodnotami.



Tato ikona označuje tip, nápad nebo obecnou poznámku.



Tato ikona označuje varování nebo upozornění.

Použití ukázkových zdrojových kódů

Veškeré zdrojové kódy použité v příkladech na stránkách této knihy jsou k dispozici ke stažení na webové stránce knihy (<http://knihy.cpress.cz/K1840>). Stačí stisknout tlačítko Soubory ke stažení a stáhnout nabízený soubor.

Tato kniha má ulehčit vaši práci. Kód v této knize můžete používat ve svých programech a dokumentaci. Nemusíte nás žádat o povolení, dokud nekopírujete výraznou část zdrojového kódu této knihy. Například pro vytvoření programu s několika úryvky kódu z této knihy není potřeba žádné povolení. Ani odpověď na nějakou otázku citací z knihy s uvedením některých příkladů nevyžaduje žádné povolení. K prodeji nebo distribuci nosičů s příklady z knih vydavatelství Computer Press je však povolení *nutné*. Začlenění výrazného množství kódu z této knihy do dokumentace vašeho produktu také *vyžaduje* povolení.

Při citacích z této knihy oceníme uvedení zdroje. Obvykle stačí uvést název, vydavatele a webovou stránku knihy: *jQuery*, *Kuchařka programátora* (Computer Press, <http://knihy.cpress.cz/K1840>). Pokud si myslíte, že použití kódu z této knihy vyžaduje naše povolení, nebojte se nás kontaktovat na e-mailové adrese knihy@cpress.cz.