

Nejnovější
technologie
a revoluční
techniky vývoje

Peter Lubbers, Brian Albers, Frank Salim

HTML5

Programujeme moderní webové aplikace

Využijte výkonná API
k vývoji moderních
RIA aplikací



Ke stažení zdrojové
kódy příkladu

 C PRESS

Apress®

Peter Lubbers, Brian Albers, Frank Salim

HTML5

Programujeme moderní webové aplikace

Computer Press, a.s.
Brno
2011

HTML5

Programujeme moderní webové aplikace

Peter Lubbers, Brian Albers, Frank Salim

Computer Press, a. s., 2011. Vydání první.

Překlad: Ondřej Gibl

Jazyková korektura: Alena Láníčková, Jana Otevřelová

Sazba: Martin Hubík

Rejstřík: Jakub Urban

Obálka: Martin Sodomka

Komentář na zadní straně obálky: Martin Domes

Technická spolupráce: Jiří Matoušek,

Zuzana Šindlerová, Dagmar Hajdajová

Odpovědný redaktor: Martin Domes

Technický redaktor: Jiří Matoušek

Produkce: Petr Baláš

Original edition copyright: © 2010 by Peter Lubbers, Brian Albers, Frank Salim. All rights reserved.

Czech Edition copyright: © 2011 by Computer Press. All rights reserved.

Autorizovaný překlad z originálního anglického vydání Pro HTML5 programming.

Originální copyright: © 2010 by Peter Lubbers, Brian Albers, Frank Salim.

Překlad: © Computer Press, a.s., 2011.

Computer Press, a. s.,

Holandská 3, 639 00 Brno

Objednávky knih:

<http://knihy.cpress.cz>

distribuce@cpress.cz

tel.: 800 555 513

ISBN 978-80-251-3539-6

Prodejní kód: K1939

Vydalo nakladatelství Computer Press, a. s., jako svou 4003. publikaci.

© Computer Press, a. s. Všechna práva vyhrazena. Žádná část této publikace nesmí být kopírována a rozmnožována za účelem rozšiřování v jakékoli formě či jakýmkoli způsobem bez písemného souhlasu vydavatele.

Stručný obsah

Úvodem.....	17
1. Seznámení s HTML5.....	21
2. Rozhraní Canvas.....	47
3. Práce s audiem a videem v HTML5	85
4. Geolokační rozhraní	109
5. Komunikační rozhraní	137
6. Rozhraní WebSocket.....	161
7. Rozhraní HTML5 pro formuláře	195
8. Rozhraní Web Workers.....	219
9. Rozhraní webového úložiště.....	241
10. Tvorba off-line webových aplikací HTML5	273
11. Budoucnost HTML5	291

Obsah

Předmluva	13
O autorech.....	14
O odborném korektrovi	14
Poděkování.....	15
Věnování	16
Úvodem	17
Komu je tato kniha určena	17
Co v této knize najdete.....	18
Kontakt na autory.....	19
Zpětná vazba od čtenářů.....	19
Zdrojové kódy ke knize	19
Errata	20
Kapitola 1	
Seznámení s HTML5	21
Jak se vyvíjela specifikace HTML5	21
Mýtus roku 2022 a proč na tom nezáleží.....	23
Kdo vyvíjí HTML5	23
Nová vize.....	24
Kompatibilita.....	24
Přínos a upřednostnění uživatelů.....	24
Zjednodušení interoperability	25
Obecný přístup.....	26
Paradigma bez zásuvných modulů	26
Co sem patří a co ne.....	27
Co je v HTML5 nového.....	29
Nový DOCTYPE a znaková sada	29
Nové a zastaralé elementy.....	31

Sémantické elementy	31
Zjednodušení výběru pomocí rozhraní Selector	38
Protokolování a ladění JavaScriptu	41
window.JSON	42
DOM Level 3	43
Monkeys, Squirrelish a další podivnosti.....	43
Shrnutí	45

Kapitola 2

Rozhraní Canvas 47

Seznámení s rozhraním Canvas jazyka HTML5.....	47
Historie	48
Co je to plátno?	48
Souřadnice plátna	49
Kdy plátno nepoužívat	49
Alternativní obsah.....	50
CSS a plátno.....	50
Podpora rozhraní Canvas ze strany prohlížečů	50
Použití rozhraní Canvas	51
Ověření podpory prohlížeče	51
Přidání plátna do stránky.....	52
Aplikace transformací	55
Práce s cestami	58
Vytažení cesty a jeho styl.....	60
Práce s výplněmi	61
Vyplnění obdélníkové oblasti	62
Křivky.....	63
Vložení obrázku do plátna	65
Přechody.....	66
Pozadí.....	69
Změna velikost objektů plátna.....	71
Použití transformací.....	73
Text.....	74
Stíny	76
Manipulace s pixely	78
Zabezpečení plátna	79
Vytvoření aplikace využívající rozhraní Canvas.....	80
Praktický příklad – skleněná tabule přes celé okno.....	84
Shrnutí	84

Kapitola 3

Práce s audiem a videem v HTML5 85

Seznámení s audio- a videorozhraním HTML5	86
Kontejnery videa.....	86
Kodeky audia a videa	87
Omezení audia a videa	88
Podpora audia a videa HTML5 v prohlížečích.....	89
Použití rozhraní audio a video	89
Seznámení s multimediálními elementy	91
Práce s audiem.....	96
Práce s videem.....	97
Praktické doplňky	104
Shrnutí	106

Kapitola 4

Geolokační rozhraní..... 109

Informace o poloze	110
Zeměpisná šířka a délka.....	110
Odkud pochází informace o poloze?	111
Lokalizace podle adresy IP	111
Lokalizace pomocí GPS	112
Lokalizace pomocí Wi-Fi.....	112
Lokalizace v mobilní síti	113
Informace o poloze zadané uživatelem	113
Podpora geolokačního rozhraní HTML5 ze strany prohlížečů	114
Soukromí.....	114
Ochrana soukromí.....	115
Nakládání s informacemi o poloze.....	116
Použití geolokačního rozhraní HTML5.....	117
Ověření podpory prohlížeče	117
Požadavky na polohu.....	117
Opakované aktualizace polohy	122
Praktický příklad použití geolokačního rozhraní HTML5	123
Kód HTML	125
Zpracování geolokačních dat	126
Výsledný kód	129

Praktické doplňky	132
Jaký je stav?	132
Zobrazení na mapě	134
Shrnutí	136

Kapitola 5

Komunikační rozhraní..... 137

Komunikace mezi dokumenty	138
Zabezpečení na základě původu	140
Podpora komunikace mezi dokumenty ze strany prohlížečů	141
Použití metody postMessage	142
Ukázka použití metody postMessage	143
Rozhraní XMLHttpRequest Level 2	149
Požadavky mezi zdroji různých původů	149
Stavové události	151
Podpora rozhraní XMLHttpRequest Level 2 prohlížeči	152
Použití rozhraní XMLHttpRequest	152
Ukázka použití rozhraní XMLHttpRequest	154
Praktické doplňky	159
Strukturovaná data	159
Framebusting	159
Shrnutí	160

Kapitola 6

Rozhraní WebSocket 161

Seznámení s rozhraním WebSocket	162
HTTP a komunikace v reálném čase	162
Rozhraní WebSocket	163
Podpora rozhraní WebSocket ze strany prohlížečů	170
Vytvoření jednoduchého serveru WebSocket	170
Použití rozhraní WebSocket	178
Ověření podpory prohlížeče	178
Seznámení s rozhraním	179
Vytvoření aplikace s použitím rozhraní WebSocket	183
Soubor HTML	184
Kód pro práci s rozhraním WebSocket	186
Kód pro práci s geolokačním rozhraním	187
A teď to dáme všechno dohromady	188
Výsledný kód	190
Shrnutí	193

Kapitola 7

Rozhraní HTML5 pro formuláře..... 195

Seznámení s rozhraním HTML5 pro formuláře	196
Formuláře HTML vs. XForms	196
Funkční formuláře	196
Podpora formulářů HTML5 ze strany prohlížečů	197
Nové ovládací prvky	198
Použití formulářů HTML5	202
Nové atributy a metody	202
Validace formulářů	206
Zobrazení výsledku validace	210
Vytvoření aplikace s formuláři HTML5	212
Praktické doplňky	217
Heslo zní validace	217
Shrnutí	218

Kapitola 8

Rozhraní Web Workers 219

Podpora rozhraní Web Workers prohlížeči	220
Použití rozhraní Web Workers	220
Ověření podpory prohlížeče	221
Vytvoření nové instance Web Worker	221
Načítání externích skriptů	222
Vzájemná komunikace	222
Vytvoření hlavní stránky	223
Vytvoření skriptu instance Web Worker	223
Ošetření chyb	224
Ukončení instance Web Worker	224
Použití rozhraní Web Workers uvnitř instance Web Worker	224
Časovače	225
Zdrojový kód ukázky	225
Vytvoření aplikace využívající rozhraní	
Web Workers HTML5	227
Vytvoření skriptu rozmazani.js	228
Vytvoření stránky rozmazani.html	229
Vytvoření skriptu rozmazaniWorker.js	230
Komunikace s instancemi Web Worker	232
Aplikace v akci	233
Kód ukázky	234
Shrnutí	240

Kapitola 9

Rozhraní webového úložiště 241

Seznámení s rozhraním webového úložiště.....	242
Podpora rozhraní v prohlížečích	243
Použití rozhraní webového úložiště.....	243
Ověření podpory prohlížeče	243
Nastavení a získávání hodnot.....	245
Úniky dat.....	246
Lokální vs. relační úložiště	248
Další metody a vlastnosti rozhraní webového úložiště.....	248
Upozornění na aktualizaci obsahu úložiště	250
Prozkoumání obsahu úložiště.....	252
Vytvoření aplikace využívající rozhraní webového úložiště	253
Budoucnost úložišť na straně prohlížečů	266
Praktické doplňky	269
Ukládání objektů JSON.....	269
Komunikace mezi okny s použitím úložiště	270
Shrnutí	272

Kapitola 10

Tvorba off-line webových aplikací HTML5 273

Seznámení s off-line	
webovými aplikacemi HTML5	273
Podpora off-line webových aplikací HTML5 prohlížeči	276
Použití rozhraní HTML5	
pro off-line webové aplikace	276
Ověření podpory prohlížeče	276
Vytvoření jednoduché off-line aplikace	276
Práce off-line.....	277
Soubory manifestu.....	278
Vlastnost applicationCache	279
Vytvoření aplikace využívající rozhraní	
pro off-line webové aplikace HTML5	280
Vytvoření souboru manifestu aplikace.....	282
Vytvoření kódu HTML a CSS	282
Vytvoření kódu v JavaScriptu.....	283
Ověření podpory prohlížeče	285

Přidání obsluhy aktualizací tlačítka.....	286
Přidání kódu pro sledování polohy.....	287
Přidání kódu pro lokální ukládání dat.....	287
Obsluha událostí spojených s prací v režimu off-line	288
Shrnutí	289

Kapitola 11

Budoucnost HTML5 291

Podpora HTML5 prohlížeči	291
Vývoj HTML	292
WebGL	292
Zařízení.....	295
Datová audiorozhraní	295
Vylepšení videa	296
Události a dotykové obrazovky.....	296
P2P síť	299
Kam to celé směřuje.....	299
Shrnutí	300

Rejstřík 301

Předmluva

V červnu 2004 se v San José v Kalifornii sešli zástupci komunity sémantického webu, hlavní tvůrci prohlížečů a zástupci W3C, aby probrali další postup související s nárůstem zájmu o webové aplikace. Na konci druhého dne jednání se hlasovalo o tom, zda má organizace W3C rozšířit specifikaci HTML a DOM o nové požadavky na webové aplikace. O pár minut později bylo rozhodnuto: 8 hlasů pro a 14 proti.

Tento rozkol názorů vedl k rozdělení snah – o dva dny později vzniklo ve snaze vyřešit vyvstávající problémy těleso WHATWG, tvořené hlavními tvůrci prohlížečů. Mezi tím organizace W3C pracovala na specifikaci XHTML2, která ale o pět let později vešla v zapomnění, a organizace W3C tak zaměřila své úsilí na HTML5, společně s WHATWG.

Nyní, o šest let později, se těšíme z úsilí vloženého do návrhu HTML5. Jeho funkce nejenom sjednocují de facto standardy používané řadu let, ale také pokládají základy pro další generaci webových aplikací. Jejich příchod znamená mnohem vstřícnější aplikace pro uživatele a naopak, velmi často, mnohem méně kódu pro vás.

V této knize na vás čeká dobře promyšlená učicí křivka, která vás rychle seznámí s funkcemi HTML5 a s nimi spojenými specifikacemi. Seznámíte se s doporučenými postupy ověření dostupnosti funkcí, patřičnými případy užití a hlavně také s řadou odpovědí na otázky, které ve specifikaci nenajdete. Ukázky kódů, které v knize najdete, nejsou prostou a triviální aplikací daných rozhraní. Naopak, provedou vás tvorbou skutečných webových aplikací. Doufám, že vám tato kniha dobře poslouží, a také, že se zájmem očekáváte novou generaci webu, stejně tak jako já.

Paul Irish

Člen vývojového týmu Googlu a jQuery a hlavní vývojář knihovny Modernizr

O autorech



Peter Lubbers je ředitelem dokumentace a školení ve společnosti Kaazing. Je velkým fandou HTML5 a rozhraní WebSockets, častým mluvčím na mezinárodních konferencích a školitelem HTML5 po celém světě. Než přišel do společnosti Kaazing, pracoval téměř deset let jako architekt pro Oracle, kde napsal cenami ověřené knižní tituly a vyvinul řadu softwarových řešení, která čekají na patentování. Narodil se v Nizozemí, kde sloužil v komandu speciálních sil RoyalDutch Green Berets. Žije u národního parku TahoeNationalForest v Kalifornii a ve svém čase rád běhá a účastní se maratonů. Najdete ho i na Twitteru (@peterlubbers).



Brian Albers je viceprezidentem vývoje ve společnosti Kaazing. Jeho kariéra v oblasti webového vývoje sahá o více než patnáct let nazpět, včetně nedávné pozice senior manažera vývoje u společnosti Oracle. Vystupuje často na konferencích, jako je Web 2.0 Expo, AJAXWorld Expo a JavaOne, kde se zaměřuje na web a technologie související s uživatelským rozhraním. Narodil se v Texasu a nyní žijící v Kalifornii; každou volnou chvíli tráví na Havaii. Když zrovna nemůže relaxovat na pláži, tráví svůj volný čas ve virtuálních světech.



Frank Salim je jedním z původních zakladatelů společnosti Kaazing, který pomohl s vývojem brány pro rozhraní WebSockets a klientskou strategií. Narodil se v San Diegu a momentálně žije v MountainView v Kalifornii. Má titul v oblasti počítačových věd z univerzity Pomona-College. Když zrovna neprogramuje, rád si čte, maluje a bruslí.

O odborném korektrovi

Paul Hain je vývojář klientských aplikací, který momentálně pracuje v Londýně pro noviny Guardian. Je autorem knihy *HTML Mastery: Semantics, Standards and Styling* (friendsof ED, 2006) a má svůj vlastní web na adrese www.joeblade.com.

Poděkování

Rád bych poděkoval své ženě Vittorii, za její lásku a trpělivost, a také svým talentovaným synům Seanu a Rockymu – vzhůru ke hvězdám, kluci.

Děkuji svým rodičům, Hermanovi a Elisabeth, své sestře Alice a bratrovi Davidovi za neutuchající důvěru v mou osobu a také své babičce Gebbechien za její hrdinské činy při okupaci Holandska nacisty, které byly pro celou naši rodinu velkou lekcí.

Děkuji také svým spoluautorům, neúnavnému Brianovi a lidskému generátoru kódu Frankovi za skvělou práci vás obou.

Dík patří také Clayovi v Apressu za jeho podporu. Nesmím zapomenout ani na Jonase a Johna z Kaazingu, kteří nás přiměli napsat skutečnou knihu – neoficiální e-kniha by byla pouhým výplodem naší fantazie, tím jsem si jist.

- Peter Lubbers

Děkuji svým rodičům, Kenovi a PattyAlbersovým, za jejich neutuchající lásku a oběti, které museli podstoupit proto, abych v životě získal tolik příležitostí a možností. Bez vaší podpory a životních hodnot, které jste mi předali, bych tuto ani žádnou jinou důležitou životní úlohu nedokončil. Doprovázeli jste mě na každém mém kroku.

Johnovi patří mé nejvřelejší díky za jeho trpělivost, vždy když se naše práce o hodinu, dvě, tři nebo i více protáhla. Nepřestával jsi mě překvapovat a inspirovat.

Děkuji Pitch, Bonnie a Penelope a slibuji, že tak pozdě už večere nikdy nebude. Děkuji také všem dalším kočkám, které tu byly před nimi, vaše vrnění a předení mě bude provázet.

Dík patří mým spolupracovníkům ze společnosti Kaazing, velmi si cením své možnosti pracovat s těmi nejlepšími.

Speciální dík patří editorům v Apressu, kteří nám hned od začátku důvěřovali v tom, že nadešel ten správný čas na knihu o HTML5. Také jim děkuji za trpělivost, kterou s námi měli, zatímco jsme se pokoušeli zdokumentovat rychle se pohybující cíl.

- Brian Albers

Rád bych poděkoval svým rodičům, Mary a Sabri, kteří jsou zodpovědní za mou existenci a bez kterých by tato kniha nebyla nikdy možná.

- Frank Salim

Věnování

Mé krásné ženě Viktorii a mým synům
Seanovi a Rockymu. Jsem na vás hrdý!

- Peter Lubbers

Johnovi. Díky tobě to stojí za to.

- Brian Albers

Lidem, kteří stále čtou knihy.

- Frank Salim

Úvodem

HTML5 je naprostou novinkou. O tom nemůže být pochyb už jen proto, že jeho vývoj ještě ani neskončil. Zlomyslné jazyky dokonce tvrdí, že HTML5 nebude připravené dříve než za deset, či dokonce více let.

Proč by tedy někoho měla zajímat tato kniha? To je přece jednoduché. Protože pro každého, kdo hledá něco navíc, co by jeho webovou aplikaci vyzdvihlo nad ostatní, je ten pravý čas pro HTML5 právě teď. Autoři této knihy se už více než dva roky podílí na vývoji a výuce technologií spojených s HTML5 a na práci s nimi a s jistotou mohou potvrdit, že rychlost, s jakou jsou tyto nové standardy přijímány, se zvyšuje závratným tempem. Dokonce i v průběhu psaní této knihy jsme museli neustále aktualizovat tabulku podporovaných prohlížečů a přehodnocovat naše domněnky o tom, co je možné použít.

Většina uživatelů si neuvědomuje skutečnou sílu dřímající v prohlížečích, které nyní používají. Ano, možná si všimnou menší změny uživatelského rozhraní poté, co dojde k automatické aktualizaci jejich prohlížeče. Nemusí však mít ani potuchy o tom, že tato nová verze umožňuje volné kreslení na stránce anebo síťovou komunikaci v reálném čase či mnohá další potencionální vylepšení.

Tato kniha se vám snaží přiblížit možnosti HTML5.

Komu je tato kniha určena

Obsah této knihy je určený zkušeným vývojářům webových aplikací znalým programování v JavaScriptu. Jinak řečeno tato kniha nepopisuje základy webového vývoje. Seznámit se se základy webového programování vám mohou pomoci mnohé z existujících zdrojů. Pokud se nacházíte v některé z následujících situací, pak vám tato kniha pravděpodobně poskytne užitečné a cenné informace, které hledáte:

- ◆ Někdy si říkáte, „Kdyby tak prohlížeč dokázal...“
- ◆ Pomocí vývojových nástrojů a zdrojového kódu stránky pitváte poměrně zajímavý web.
- ◆ Se zájmem si čtete informace o nejnovějších aktualizacích prohlížečů, abyste zjistili, co je nového.

- ◆ Hledáte nové způsoby optimalizace a modernizace vašich aplikací.
- ◆ Jste ochotní upravit váš web pro jeho návštěvníky s novými prohlížeči tak, abyste zajistili jejich maximální spokojenost.

Nacházíte-li se některé z těchto situací, neměla by tato kniha uniknout vaší pozornosti.

Přestože se budeme snažit upozorňovat na omezenou podporu prohlížečů, bude-li to na místě, není naším cílem vytvářet oklidy umožňující hladký běh vaší HTML5 aplikace v deset let starém prohlížeči. Praxe potvrdila, že se tato náhradní řešení i základní podpora ze strany prohlížečů vyvíjí tak rychle, že kniha tohoto ražení není nejlepším místem pro tyto informace. Místo toho se zaměříme na specifikaci HTML5 a jeho použití. Detailní náhradní řešení, která časem budou ztrácet na významu, najdete na Internetu.

Co v této knize najdete

Jedenáct kapitol této knihy probírá výběr populárních, užitečných a mocných rozhraní API HTML5. Některé části staví, za účelem nabídnutí co nejbohatších ukázek, na možnostech představených v předchozích kapitolách.

Kapitola 1, *Seznámení s HTML5*, začíná seznámením s předchozími a aktuální verzí specifikace HTML. Kapitola představuje nové sémantické značky vysoké úrovně a tatké se základní změny a důvody, které stojí za veškerými snahami poslední doby o vývoj HTML5. Je dobré znát terén.

Kapitola 2, *Rozhraní Canvas HTML5*, a kapitola 3, *Práce s audiem a videem v HTML5*, popisují nové vizuální a multimediální elementy. Tyto kapitoly se zaměřují na to, jak snadněji obohatit uživatelské rozhraní bez použití pluginů anebo interakcí se serverem.

Kapitola 4, *Geolokační rozhraní*, představuje zcela novou funkci, kterou doposud nebylo snadné zastoupit, a tou je možnost identifikovat aktuální umístění uživatele, na jehož základě může aplikace pro uživatele upravit své rozhraní. Velmi důležité je z tohoto pohledu zajištění soukromí, a proto v této kapitole probereme také některá z možných úskalí.

Následující dvě kapitoly, *Komunikační rozhraní* a *Rozhraní WebSocket HTML5*, prezentují stále se zdokonalující způsoby, jakými HTML5 umožňuje komunikovat s ostatními weby a streamovat v reálném čase data aplikaci s maximální jednoduchostí a minimální režií. Techniky popisované v těchto kapitolách umožňují zjednodušení mnoha velmi komplexních architektur, se kterými se dnes na webu setkáte.

Kapitola 7, *Rozhraní HTML5 pro formuláře*, prezentuje minimální zásahy, které můžete v rámci zajištění lepší použitelnosti vaší webové aplikace provést, stejně jako zásadnější změny, které můžete v těch nejběžnějších situacích učinit pro detekci špatně zadaných vstupních údajů.

Kapitoly 8, 9 a 10, *Rozhraní Web Workers HTML5*, *Rozhraní Web Storage HTML5* a *Tvorba off-line webových aplikací HTML5*, pojednávají o interním chodu aplikací. Najdete zde způsoby optimalizace existujících funkcí vedoucí k dosažení vyššího výkonu a umožňující lepší správu dat.

V kapitole 11, *Budoucnost HTML5*, se pak můžete těšit na pořádnou ochutnávku toho, co ještě přijde.

Kontakt na autory

Děkujeme, že jste se rozhodli koupit tuto knihu. Doufáme, že vás bude bavit a že pro vás bude cenným zdrojem informací. Navzdory naší maximální snaze vyvarovat se chybám je nám jasné, že věci někdy nejdou úplně podle plánu, a předem se tak omlouváme za jakákoli podobná pochybení. Uvítáme váš osobní názor, otázky a komentáře týkající se obsahu knihy a zdrojových kódů. Můžete nás kontaktovat e-mailem na adrese *prohtml5@gmail.com*.

Zpětná vazba od čtenářů

Nakladatelství a vydavatelství ComputerPress, 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 PC literatury
ComputerPress
Spielberk Office Centre
Holandská 3
639 00 Brno

nebo

sefredaktor.pc@cpress.cz

Zdrojové kódy ke knize

Z adresy <http://knihy.cpress.cz/k1939> si po klepnutí na odkaz Soubory ke stažení můžete přímo stáhnout archiv s ukázkovými kódy.

Errata

Přestože jsme udělali maximum pro to, abychom zajistili přesnost a správnost obsahu, chybám se úplně vyhnout nedá. Pokud v některé z našich knih najdete chybu, ať už chybu v textu nebo v kódu, budeme rádi, pokud nám ji nahlásíte. Ostatní uživatelé tak můžete ušetřit frustrace a pomoci nám zlepšit následující vydání této knihy.

Veškerá existující errata najdete zobrazíte na adrese <http://knihy.cpress.cz/k1939> po klepnutí na odkaz Soubory ke stažení.

KAPITOLA 1

Seznámení s HTML5

Tato kniha se zabývá programováním v HTML5. Dříve než se ale budete moci naučit v HTML5 programovat, je třeba udělat krok opačným směrem a porozumět tomu, co HTML5 doopravdy je, jaká je jeho historie a jaké jsou rozdíly mezi HTML 4 a HTML5.

V této kapitole se pustíme rovnou do praktických otázek, na které chtějí všichni znát odpovědi. Proč HTML5 a proč je kolem něj teď takový povyk? Jaké nové vlastnosti činí HTML5 skutečně revolučním a současně vycházejícím vstříc? Jaké jsou implikace paradigmatu bez zásuvných modulů? Jaké jsou jeho výhody a nevýhody? Co je nového v HTML? A skutečně začíná pro webové vývojáře nová éra? Pojďme to zjistit.

Jak se vyvíjela specifikace HTML5

Kořeny jazyka HTML sahají hluboko do historie. Jeho návrh byl poprvé zveřejněn v roce 1993. 90. léta byla ve znamení výrazné aktivity okolo HTML a přinesla verze 2.0, 3.2 a 4.0 (ve stejném roce) a v roce 1999 pak konečně verzi 4.01. V průběhu jeho vývoje převzalo konsorciem W3C (WorldWide Web Consortium) vládu nad specifikací.

Po rychlém vzniku těchto čtyř verzí se však HTML považovalo spíše za slepou uličku. Zaměření webových standardů se přesunulo k XML a XHTML a jazyk HTML se odsunul na druhou kolej. Mezi tím však jazyk HTML odmítl vzdát se bez boje a většina obsahu na webu byla stále ve formátu HTML. V rámci podpory vzniku nových webových aplikací a vyřešení nedostatků HTML byly zapotřebí nové funkce a specifikace pro HTML.

Malá skupina lidí založila v roce 2004 ve snaze povýšit webovou platformu na novou úroveň skupinu WHATWG (Web Hypertext Application Working Group). Vytvořili specifikaci HTML5. Zahájili také práce na nových funkcích specificky cílených na webové aplikace – na oblast, která podle nich strádala nejvíce. Zhruba v této době se objevil termín Web 2.0. A skutečně to bylo jako druhý a nový web, když statické weby uvolnily místo těm více dynamickým a sociálním webům vyžadujícím více funkcí – mnohem více funkcí.

Konsorcium W3C se do vývoje HTML opět zapojilo v roce 2006. V roce 2008 zveřejnilo první pracovní návrh HTML5 a v roce 2009 pak XHTML 2. Uplynul další rok a jsme u současného stavu. Protože HTML5 řeší velmi praktické problémy (jak později sami uvidíte), tvůrci prohlížečů horečně implementují jeho nové funkce i přesto, že jeho specifikace ještě není definitivně ustanovená. Experimenty na straně prohlížečů přinášejí cennou zpětnou vazbu, a vylepšují tak specifikaci. HTML5 se rapidně vyvíjí a adresuje skutečná a praktická vylepšení webové platformy.

Historické milníky HTML

Brian říká: „Ahoj, jmenuju se Brian a z HTML jsem značně rozmrzelý.“

Svou první domovskou stránku jsem vytvářel v roce 1995. V té době byla domovská stránka něčím, co člověk vytvářel, aby o sobě ostatním něco řekl. Zpravidla obsahovala špatně naskenované obrázky, značky `<blink>`, informace o tom, kde bydlíte a co čtete a na jakém projektu, souvisejícím s počítači, právě pracujete. Já i většina mých kolegů vývojářů WorldWide Webu studovala anebo pracovala na univerzitách.

V době, o které je zde řeč, byl jazyk HTML na primitivní úrovni a nástroje pro práci s ním nebyly k dispozici. O existenci webových aplikací se stěželo hovořit, pomineme-li několik primitivních skriptů pro zpracování textu. Stránky se vytvářely ručně s použitím oblíbeného textového editoru. Aktualizovaly se jednou za pár týdnů či měsíců, pokud vůbec.

Za těch 15 let jsme ušli skutečně velký kus cesty.

Dnes není nic neobvyklého, že si uživatelé aktualizují své online profily i několikrát za den. Tento typ interakce by nebyl možný bez trvalého vývoje a zdokonalování online nástrojů stavějících na předchozí generaci.

Mějte to při čtení této knihy na paměti. Příklady, které zde ukazujeme, se mohou někdy jevit velmi jednoduché, potenciál je však neomezený. Ti z nás, kteří v polovině 90. let poprvé použili značku `<img>` nejspíše neměli ani tušení o tom, že během deseti let budou lidé ukládat a editovat své fotografie online. Měli jsme to však tušit.

Doufáme, že vás ukázky, které v této knize prezentujeme, inspirují k vytvoření nových základů webu pro další dekádu.“

Mýtus roku 2022 a proč na tom nezáleží

Specifikace HTML5, kterou dnes máme, byla zveřejněna jako pracovní verze – ještě není ve své finální podobě. Kdy tedy dojde k jejímu vytesání do kamene? Zde je několik klíčových dat, která je dobré znát. Tím v pořadí prvním je rok 2012, který je datem pro *kandidáta na doporučení*. Druhým datem je rok 2022, představující rok pro *navrhovaná doporučení*. Ještě však neodkládejte tuto knihu na deset let stranou, jinak se nedozvíte, co tato dvě data doopravdy vyjadřují.

První a nejbližší datum je tím pravděpodobně nejdůležitějším, protože poté, co dosáhneme této fáze, bude HTML5 kompletní. A to by mělo být už za dva roky. Význam navrhovaných doporučení (která, jak se všichni jistě shodneme, na nás čekají až za delší čas) spočívá v tom, že poté budou existovat dvě implementace schopné vzájemné spolupráce. To znamená dva prohlížeče vybavené plně spolupracujícími implementacemi celých specifikací – ušlechtilý cíl, který činí z roku 2022 ambiciózní termín. Tohoto stavu jsme koneckonců ještě nedosáhli ani u HTML 4.

V tuto chvíli je velmi důležité, že tvůrci webových prohlížečů aktivně rozšiřují podporu mnoha skvělých nových funkcí. V závislosti na vaší cílové skupině návštěvníků můžete mnohé z těchto funkcí začít používat už teď. Jistě, nevyhnete se nutnosti menších změn, to je však jen malá daň za výhody, které nejnovější technologie přináší. Pokud vaše cílová skupina uživatelů používá Internet Explorer 6.0, nebudou přirozeně mnohé z těchto nových funkcí pracovat a budou vyžadovat emulaci – to však ještě stále není dostatečný důvod pro odsouzení HTML5. I tito uživatelé totiž jednoho dne přejdou k novější verzi prohlížeče. Mnozí z nich pravděpodobně sáhnou rovnou po Internet Exploreru 9.0, v jehož případě se společnost Microsoft kasá, že se jedná o prohlížeč s rozšířenou podporou HTML5. V praxi znamená kombinace nových prohlížečů a zdokonalujících se technik emulace to, že mnohé z funkcí HTML5 můžete používat už nyní anebo v krátkém časovém horizontu.

Kdo vyvíjí HTML5

Všem je nám jasné, že je nutné určité rozdělení odpovědností a někdo musí být zodpovědný za specifikaci HTML5. Tento úkol leží na bedrech tří důležitých organizací:

- ◆ *WHATWG (Web Hypertext Application Technology Working Group)* – tato organizace, založená v roce 2004 několika jednotlivci pracujícími pro společnosti vyvíjející prohlížeče, jako je Apple, Mozilla, Google a Opera, vyvíjí HTML a rozhraní API pro vývoj webových aplikací a otevřeně spolupracuje s tvůrci prohlížečů a dalšími zainteresovanými stranami.
- ◆ *W3C (WorldWide Web Consortium)* – součástí konsorcia W3C je i pracovní skupina zodpovědná za HTML, která má momentálně na starosti dodání specifikace HTML5.
- ◆ *IETF (Internet Engineering Task Force)* – součástí této organizace jsou také skupiny odpovědné za Internetové protokoly, jako je HTTP. HTML5 definuje nové rozhraní API WebSocket, které staví na protokolu WebSocket, jež vyvíjí jedna z pracovních skupin IETF.

Nová vize

HTML5 staví na různých principech uváděných specifikací WHATWG, které ztělesňují novou vizi možností a praktičnosti.

- ◆ Kompatibilita
- ◆ Přínos
- ◆ Interoperabilita
- ◆ Obecný přístup

Kompatibilita

Nemusíte se bát, HTML5 nepředstavuje znepokojivý typ revoluce. Jednou z jeho klíčových zásad je naopak zajistit, aby vše hladce pracovalo. Pokud nejsou funkce HTML5 podporované, musí dojít k ladné degradaci chování. A vzhledem k tomu, že tam venku čeká více jak 20 let obsahu v HTML, je velice důležitá podpora tohoto stávajícího obsahu.

Na výzkum typického chování bylo vynaloženo mnoho úsilí. Google například analýzou milionů stránek odhalil typické identifikátory elementů `DIV` a zjistil obrovské množství shod. Mnoho lidí například používá element `DIV` s identifikátorem `header` pro označení záhlaví stránky. HTML5 je především o řešení skutečných problémů, že ano? Proč tedy jednoduše nevytvořit značku `<header>?`

Přestože jsou některé funkce standardu HTML5 poměrně revoluční, jedná se o evoluci, a nikoli o revoluci. Proč znovu vynalézat kolo? (A pokud už musíte, ať je aspoň lepší.)

Přínos a upřednostnění uživatelů

Specifikace HTML5 vychází z jednoznačného *upřednostnění uživatelů*. Co se priorit týká, je uživatel králem. To znamená, že v případě pochyb specifikace upřednostňuje uživatele před autory, programátory (prohlížeči), těmi, kdo stojí za specifikací (W3C/WHATWG), a před teoretickou čistotou. Důsledkem toho je až nebývalá praktičnost HTML5, která však v některých případech má k dokonalosti ještě daleko.

Uvažte následující příklad. Následující kódy jsou v HTML5 všechny zcela platné:

```
id="prohtml5"  
id=prohtml5  
ID="prohtml5"
```

Někteří jistě budou mít proti této uvolněnější syntaxi námitky, faktem však je, že koncovému uživateli je to jedno. Neříkáme, abyste začali psát ledabylý kód, v konečném důsledku je to však koncový uživatel, kdo doplatí na případné chyby vyvolané výše uvedenými kódy, jež mají za následek nevykreslení stránky.

HTML5 také dalo podnět k vytvoření XHTML5, aby mohly nástroje pro práci s XML generovat platný kód HTML5. Serializace HTML anebo XHTML verze by měla vyprodukovat stejné stromy DOM jen s minimálními rozdíly. Syntaxe XHTML je přirozeně mnohem přísnější a poslední dva řádky z předchozího příkladu by tak nebyly legitimní.

BEZPEČNÝ NÁVRH

Již od samotného začátku byl kladen velký důraz na zabezpečení HTML5. Každá část specifikace obsahuje sekce věnované úvahám o zabezpečení, které je vždy důkladně rozvážené již v předstihu. HTML5 přichází s novým modelem zabezpečení, jehož použití je nejenom snadné, ale také konzistenční napříč různými rozhraními API. Tento model zabezpečení umožňuje provádět určité věci takovým způsobem, který dříve nebyl možný. Umožňuje zabezpečenou komunikaci mezi doménami, aniž by k tomu byly zapotřebí nejrůznější důmyslné, důvtipné a v konečném důsledku nepříliš bezpečné postupy. V tomto ohledu jistě nebudeme vzpomínat na staré dobré časy.

ODDĚLENÍ FORMY A OBSAHU

HTML5 učinilo obrovský krok kupředu k čistému oddělení formy a obsahu. HTML5 se o toto oddělení snaží všude, kde je to možné, pomocí kaskádových stylů. Většina metod prezentace obsahu z dřívějších verzí HTML už nyní není k dispozici (díky kompatibilnímu návrhu popisovanému dříve však stále budou fungovat). Tato myšlenka není žádnou novinkou – do praxe byla uvedena už v HTML 4 Transitional a XHTML1.1. Weboví návrháři používají tento doporučovaný postup už dlouhou dobu, nyní je však oddělení obsahu od formy ještě důležitější. Prezentační značky trpí následujícími problémy:

- ◆ Špatná přístupnost
- ◆ Zbytečná složitost (kód s řádkovými styly se obtížněji čte)
- ◆ Větší velikost dokumentů (z důvodu opakování stylů), která má za následek pomalejší načítání stránek

Zjednodušení interoperability

HTML5 je především o jednoduchosti a odstranění zbytečné složitosti. Slogan HTML5 by mohl znít: „Nejlepší řešení je to nejjednodušší. Zjednodušte, kdekoli je to možné.“ Zde je několik příkladů tohoto přístupu:

- ◆ Nativní podpora ze strany prohlížečů namísto komplexního kódu v JavaScriptu
- ◆ Nový, zjednodušený DOCTYPE
- ◆ Nová, zjednodušená deklarace znakové sady
- ◆ Mocná a přitom jednoduchá rozhraní API v HTML5

O některých z těchto prvků si později řekneme více.

Zajištění vši této jednoduchosti vyžadovalo značné rozšíření specifikace, která tak musí být mnohem přesnější – dokonce o poznání přesnější než v případě jakýchkoli předchozích verzí specifikace HTML. Specifikuje řadu přesně definovaných rysů ve snaze docílit do roku 2022 skutečné interoperability prohlížečů. Nepřesnosti by tento cíl mohly snadno zhatit.

Specifikace HTML5 je mnohem detailnější než všechny předchozí také proto, aby se zabránilo nesprávnému výkladu. Cílem je definovat věci důkladně, obzvláště pak webové aplikace. Není tedy divu, že má specifikace více než 900 stránek.

Návrh HTML5 počítá také s obsluhou výjimek a má řadu ambiciózních plánů a vylepšení. Z praktických důvodů preferuje ladnou obsluhu chyb před drsným chybovým hlášením. I zde je absolutní prioritou koncový uživatel a jeho zájmy. Například chyby v dokumentech nevyústí v katastrofická selhání, kdy se stránky nezobrazí. Místo toho je přesně definovaná obsluha chyb, tak aby mohl prohlížeč zobrazit chybné značky standardním způsobem.

Obecný přístup

Tento princip se dělí do třech částí:

- ◆ *Přístupnost* – v rámci podpory uživatelů s postižením pracuje HTML5 se standardem WAI (Web Accessibility Initiative) ARIA (Accessible Rich Internet Applications). Role WAI-ARIA podporované čtečkami obrazovek už nyní můžete přiřadit vašim elementům HTML.
- ◆ *Nezávislost na médiu* – jazyk HTML5 by měl fungovat napříč všemi zařízeními a platformami, je-li to vůbec možné.
- ◆ *Podpora všech světových jazyků* – například nová značka `<ruby>` podporuje anotace Ruby, které se používají v typografii Východní Asie.

Paradigma bez zásuvných modulů

HTML5 nabízí podporu mnoha funkcí dříve možných pouze s použitím zásuvných modulů anebo komplexních postupů (nativní podpora rozhraní pro kreslení, sockety atd.). Se zásuvnými moduly jdou samozřejmě ruku v ruce problémy:

- ◆ Zásuvné moduly není vždy možné nainstalovat.
- ◆ Zásuvné moduly mohou být deaktivované anebo blokové (například Apple iPad se nedodává se zásuvným modulem pro Flash).
- ◆ Zásuvné moduly představují samostatné bezpečnostní riziko.
- ◆ Zásuvné moduly je obtížné začlenit do zbytku dokumentu HTML (díky omezením zásuvných modulů a problémům s ořezáváním a transparentí).

Přestože se některé zásuvné moduly těší velké popularitě, v kontrolovaném prostředí společnosti se často blokují. Někteří uživatelé si navíc tyto zásuvné moduly záměrně deaktivují kvůli jejich převažujícímu využívání k reklamním účelům. Pokud si však uživatel deaktivuje daný zásuvný modul, deaktivuje si také jediný program, na který při zobrazení vašeho obsahu spoléháte.

Zásuvné moduly často mívají také problémy s integrací svého obsahu se zbytkem stránky v prohlížeči, což v případě některých typů webů způsobuje problémy s ořezáváním a transparentí. Protože zásuvné moduly používají vykreslovací jádro odlišné od zbytku webové stránky, potýkají se vývojáři s problémy, kdykoliv je zapotřebí, aby nabídka nebo jiný vizuální element na stránce překročil hranice zásuvného modulu. Zde přichází na pomoc HTML5 a jako mávnutím kouzelné hůlky nabízí *nativní* podporu. Elementy můžete opatřit kaskádovými styly a pracovat s nimi v JavaScriptu. Právě to je silnou stránkou HTML5, na které se jasně ukazují možnosti, které v předchozích verzích HTML neexistovaly. Není to jen o tom, že nové elementy nabízí novou funkcionalitu. Je to také přidání nativní interakce pomocí skriptů a stylů, která umožňuje provádět mnohem více, než kdy bylo možné.

Uvažte například nový element plátna. Umožňuje provádět základní věci, které předtím nebyly možné (zkuste si nakreslit diagonální čáru na webové stránce v HTML 4). Co je však nejzajímavější, jsou možnosti ukrývající se v rozhraní API a stylech, které je možné aplikovat jen pomocí několika řádků kódu jazyka CSS. Stejně jako dobře vychované děti si i elementy HTML5 vzájemně rozumí. Můžete například vzít snímek z videoelementu a zobrazit ho na plátně tak, aby se uživateli po klepnutí na plátno přehrál video od právě zobrazeného snímku. To je jen jeden z příkladů toho, co může nativní kód nabídnout v porovnání se zásuvným modulem. Ve skutečnosti je téměř vše snazší, nepracujete-li s černou skříňkou. Dohromady tak vzniká skutečně mocné médium, což je také důvod, proč jsme se rozhodli napsat knihu o *programování* v HTML5, a ne pouze o nových elementech.

Co sem patří a co ne

Co je tedy doopravdy součástí HTML5? I přesto, že si pečlivě prostudujete celou dokumentaci, nemusíte najít některé funkce popisované v této knize. Nenajdete například geolokaci HTML5 ani část Web Workers. Copak si to jen vymýšlíme? Je to všechno jen zbytečný povyk? Vůbec ne.

Mnohé z funkcí HTML5 (například Web Storage a Canvas 2D) byly původně součástí specifikace HTML5 a poté se přesunuly do samostatných standardizovaných dokumentů, tak aby se specifikace neodkláněla od svého primárního záměru. Považovalo se za rozumné probírat a upravovat některé funkce samostatně, před jejich začleněním do oficiální dokumentace. Jeden drobný problém značkovacího jazyka tak nezdrží celou specifikaci.

Expertů na jednotlivé oblasti si mohou v diskuzích vyměňovat své názory na určité funkce bez zbytečných okolních rušivých vlivů. Průmysl pro původní funkce, jako je například geolokace, nadále používá označení HTML5. Na HTML5 tak můžete nahlížet jako na pojem zastřešující

klíčové značky i mnoho skvělých a nových rozhraní API. V době psaní této knihy jsou součástí HTML5 následující funkce:

- ◆ Canvas (2D a 3D)
- ◆ Výměna zpráv po kanálu
- ◆ Výměna zpráv mezi dokumenty
- ◆ Geolokace
- ◆ MathML
- ◆ Microdata
- ◆ Události Server-Sent
- ◆ SVG (Scalable Vector Graphics)
- ◆ Protokol a rozhraní API WebSocket
- ◆ Koncept původu webu
- ◆ Webové úložiště
- ◆ Webová SQL databáze
- ◆ Web Workers
- ◆ XMLHttpRequest Level 2

Jak sami můžete vidět, nachází se v tomto seznamu mnohá rozhraní API, se kterými se seznámíme v této knize. Na základě jakého klíče jsme vybírali rozhraní API pro tuto knihu? Zvolili jsme ta, která už jsou alespoň částečně připravená, tedy ta, která jsou v určité formě dostupná ve více než jednom prohlížeči. Ostatní (méně dopracované) funkce mohou být k dispozici pouze ve speciálních beta verzích konkrétních prohlížečů, jiné zatím existují pouze na papíře.

V této knize najdete aktuální (v době psaní knihy) přehled podpory jednotlivých probíraných funkcí HTML5 ze strany prohlížečů. Vše, co se zde dozvíte, však už brzy bude neaktuální, protože se jedná o střelbu na rychle se pohybující cíl. Ničeho se však nebojte – k dispozici je řada skvělých online zdrojů, kde můžete ověřit aktuální (a budoucí) podporu prohlížečů. Web www.caniuse.com nabízí vyčerpávající seznam funkcí a jejich podpory ze strany jednotlivých verzí prohlížečů, web www.html5test.com ověří podporu funkcí HTML5 v prohlížeči použitém pro zobrazení webu.

Tato kniha se nezabývá technikami emulace funkcí HTML5 v zastaralých prohlížečích. Primárně se zaměřuje na specifikaci HTML5 a jeho použití. U každého z popisovaných rozhraní API najdete ukázkové kódy demonstrující detekci jeho dostupnosti. Namísto detekce pomocí položky `User agent`, která je často nespolehlivá, používáme detekci pomocí položky `Feature`. K tomuto účelu můžete použít také `Modernizr` – knihovnu v JavaScriptu umožňující pokročilou detekci funkcí HTML5 a jazyka CSS3. Knihovnu `Modernizr` vřele doporučujeme, protože se jedná o zcela nejlepší nástroj pro tento účel.

Pár dalších slov o HTML

Frank říká: „Ahoj, jmenuju se Frank a někdy maluju.“

Jedna z prvních ukázek použití plátna v HTML, kterou jsem viděl, byla jednoduchá aplikace pro kreslení napodobující uživatelské rozhraní programu Malování společnosti Microsoft. Přestože značně zaostával za tehdejšími nejlepšími grafickými editory a fungoval pouze ve zlomku tehdy existujících prohlížečů, přiměl mě zamyslet se nad možnostmi, které prezentoval.

Pro grafické úpravy zpravidla používám lokálně nainstalovaný software. Přestože je řada z těchto programů na vysoké úrovni, schází jim ty charakteristické rysy, které webové aplikace činí tak výjimečnými – jednoduše to, že jsou online. Sdílení digitálních obrázků až doposud obnášelo exportování obrázků z grafického editoru a jeho nahrání na web. Aplikace HTML5 mohou pomoci změnit zavedené postupy a umožnit průnik kreativity do online světa, společně s dokončenými obrázky.

Počet aplikací, které není možné implementovat s použitím HTML5, se ztenčuje. S ohledem na text již web představuje ultimativní obousměrné komunikační médium. Textové aplikace jsou dnes k dispozici v podobně běžných webových formulářů. Jejich grafické protějšky umožňující například malování, editaci videa nebo modelování ve 3D prostředí přichází právě teď.

Nyní můžete vytvářet skvělé softwary pro vytváření a editaci obrázků, hudby, videí a dalších. A co je ještě lepší, námi vytvořený software bude na webu – platformě, jež je všudypřítomná, mocná a online.“

Co je v HTML5 nového

Ještě než začneme programovat v HTML5, pojďme se v rychlosti podívat na to, co je v HTML5 nového.

Nový DOCTYPE a znaková sada

V souladu s v pořadí třetí zásadou, kterou je jednoduchost, došlo k výraznému zjednodušení elementu DOCTYPE webových stránek. Podívejte se na následující deklaraci DOCTYPE v HTML 4:

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

Kdo si něco takového dokáže zapamatovat? My tedy určitě ne. Vždy jsme pouze zkopírovali tuto zdlouhavou deklaraci DOCTYPE a vložili ji do stránky, s obavami, zda se jedná o tu správnou. HTML5 tento problém elegantně řeší následujícím způsobem:

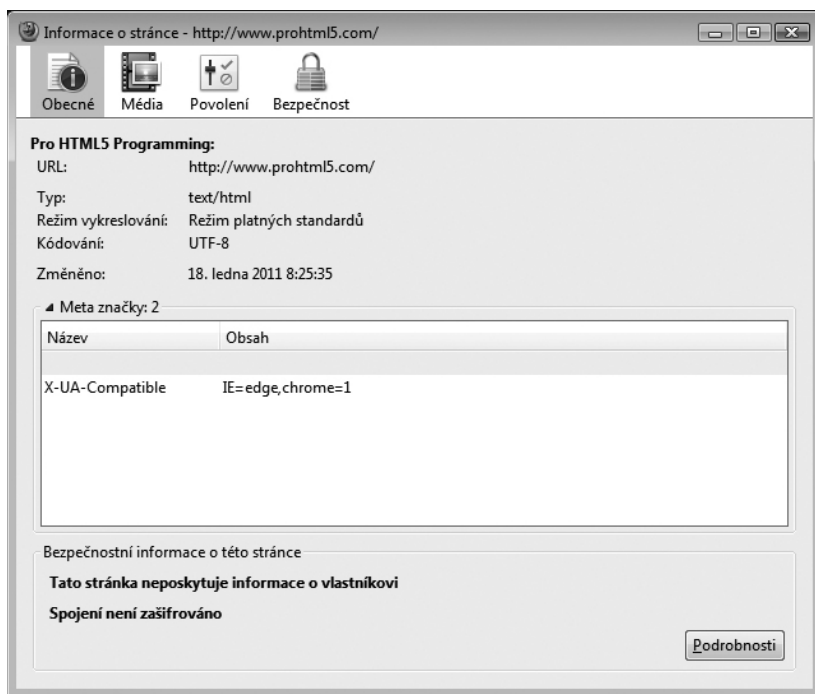
```
<!DOCTYPE html>
```

Tak *tuto* deklaraci DOCTYPE už si zapamatovat lze. Stejně jako v případě DOCTYPE došlo také k zjednodušení a zkrácení deklarace znakové sady. Z tohoto:

```
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
```

se stalo toto:

```
<meta charset="utf-8">
```



Obrázek 1.1: Stránka zobrazená ve standardním režimu

Použití nové deklarace `DOCTYPE` způsobí zobrazení stránky ve standardním režimu prohlížeče. Například obrázek 1.1 znázorňuje informace, které uvidíte, pokud otevřete HTML5 stránku ve Firefoxu, klepnete pravým tlačítkem do stránky a pak klepnete na příkaz Zobrazit informace o stránce. V tomto případě se stránka vykreslila ve standardním režimu.

Použití nové deklarace `DOCTYPE HTML5` přiměje prohlížeč k tomu, aby stránku zobrazil v režimu vyhovujícím standardům. Jak už možná víte, mohou se stránky zobrazovat v různých režimech, jako je například nestandardní režim (tzv. quirk režim), téměř standardní režim a standardní režim (ne-quirk). Deklarace `DOCTYPE` říká prohlížeči, jaký režim se má použít a jaká pravidla se mají použít pro validaci dané stránky. V nestandardním režimu se prohlížeč snaží zobrazit stránku i přesto, že nejsou úplně validní. HTML5 přichází s novými elementy a dalšími prvky (více se dozvíte v následující části). Použijete-li tedy některé ze zastaralých elementů, nebude výsledná stránka validní.

Nové a zastaralé elementy

HTML5 zavádí mnoho nových značek, které se dělí do celkem sedmi různých typů obsahu. Ty uvádí tabulka 1.1.

Tabulka 1.1: Typy obsahu v HTML5

Typ obsahu	Popis
Vložený	Obsah importující do dokumentu další zdroje, například elementy <code>audio</code> , <code>video</code> , <code>canvas</code> a <code>iframe</code> .
Tok	Elementy použité v těle dokumentu a aplikacích, například <code>form</code> , <code>h1</code> a <code>small</code> .
Nadpis	Nadpisy jako například <code>h1</code> , <code>h2</code> a <code>hgroup</code> .
Interaktivní	Obsah, se kterým uživatelé interaktivně pracují, například elementy <code>audio</code> nebo <code>video</code> , <code>button</code> nebo <code>textarea</code> .
Metadata	Elementy, které se zpravidla nacházejí v části <code>head</code> a nastavují formu anebo chování zbytku dokumentu, například <code>script</code> , <code>style</code> a <code>title</code> .
Fráze	Text a textové elementy, například <code>mark</code> , <code>kbd</code> , <code>sub</code> a <code>sup</code> .
Sekce	Elementy definující sekce dokumentu, například <code>article</code> , <code>aside</code> a <code>title</code> .

Na všechny tyto elementy je možné aplikovat kaskádové styly. Některé z nich, jako je například `canvas`, `audio` a `video`, je navíc možné používat samostatně a k tomu je ještě programově dolaďovat pomocí jejich rozhraní API. Tato rozhraní API si mnohem detailněji představíme dále v této knize.

Popis všech těchto nových elementů spadá za rámec této knihy, elementy sekcí (popisované v následující části) jsou nové. Elementy `canvas`, `audio` a `video` jsou také novinkou HTML5.

Stejně tak zde nenajdete vyčerpávající seznam všech zastaralých značek (na toto téma najdete na Internetu mnoho dobrých zdrojů). Došlo k odstranění mnohých elementů, které suplovaly funkci stylů, jako je například `big`, `center`, `font` a `basefont`, ve prospěch jazyka CSS.

Sémantické elementy

Jedním z typů obsahu, který zavádí mnoho nových elementů HTML5, jsou *sekcce*. HTML5 definuje nové *sémantické* značky popisující obsah elementu. Použití sémantického značení nepřináší žádné okamžité výhody, zjednodušuje však návrh stránek HTML a v budoucnosti budou vyhledávače jistě využívat tyto elementy při indexaci stránek.

Jak jsme si řekli už dříve, HTML5 se snaží proměnit zaprášené polní cesty v silnice první třídy. Google analýzou milionů stránek určil běžné identifikátory elementů `DIV` a odhalil velkou míru opakování některých z nich. Mnoho lidí například používá element `DIV` s identifikátorem `footer` pro označení zápatí stránky. HTML5 proto nabízí řadu nových elementů sekcí, které můžete v moderních prohlížečích použít už nyní. Tabulka 1.2 ukazuje rozličné sémantické elementy.

Tabulka 1.2: Nové elementy sekcí HTML5

Element	Popis
header	Obsah záhlaví (stránky anebo sekce stránky)
footer	Obsah zápatí (stránky anebo sekce stránky)
section	Sekce webové stránky
article	Nezávislý obsah článku
aside	Související obsah anebo citace
nav	Navigace

Na všechny tyto elementy je možné aplikovat kaskádové styly. Jak jsme si řekli již v části o návrhu podřízeném použitelnosti, snaží se HTML5 prosazovat oddělení obsahu a formy, a proto je v HTML5 k aplikaci stylů ve stránce nutné použít jazyk CSS. Výpis 1.1 ukazuje, jak taková stránka v HTML5 může vypadat. Používá novou deklaraci DOCTYPE i znakové sady a také sémantické elementy (obsah rozdělený na sekce). Soubor s kódem je dostupný ve složce *kody/uvod*.

Výpis 1.1: Ukázka HTML5 stránky

```
<!DOCTYPE html>
<html>

<head>
  <meta charset="utf-8" />
  <title>HTML5</title>
  <link rel="stylesheet" href="html5.css">
</head>

<body>

  <header>
    <h1>Nadpis</h1>
    <h2>Podnadpis</h2>
    <h4>HTML5 je nejlepší!</h4>
  </header>

  <div id="container">

    <nav>
      <h3>Navigace</h3>
      <a href="">Odkaz 1</a>
      <a href="">Odkaz 2</a>
      <a href="">Odkaz 3</a>
```

```
</nav>

<section>
  <article>
    <header>
      <h1>Záhlaví článku</h1>
    </header>
    <p>Lorem ipsum dolor HTML5 nunc aut nunquam sit amet, con-
sectetur adipiscing elit. Vivamus at est eros, vel fringilla
urna.</p>
    <p>Per inceptos himenaeos. Quisque feugiat, justo at vehicu-
la pellentesque, turpis lorem dictum nunc.</p>
    <footer>
      <h2>Zápatí článku</h2>
    </footer>
  </article>
  <article>
    <header>
      <h1>Záhlaví článku</h1>
    </header>
    <p>HTML5: "Lorem ipsum dolor nunc aut nunquam sit amet, con-
sectetur adipiscing elit. Vivamus at est eros, vel fringilla
urna. Pellentesque odio."</p>
    <footer>
      <h2>Zápatí článku</h2>
    </footer>
  </article>
</section>

<aside>
  <h3>Související</h3>
  <p>HTML5: "Lorem ipsum dolor nunc aut nunquam sit amet, consec-
tetur adipiscing elit. Vivamus at est eros, vel fringilla urna.
Pellentesque odio rhoncus."</p>
</aside>
<footer>
  <h2>Zápatí</h2>
</footer>
</div>
</body>

</html>
```

Bez použití stylů by stránka vypadala poměrně nudně. Výpis 1.2 ukazuje kód jazyka CSS, který aplikuje styly na obsah stránky. Všimněte si, že tato šablona stylů používá některé z nových funkcí jazyka CSS3, jako jsou například zakulacené hrany (`border-radius`) a rotační transformace (`transform: rotate()`). Je třeba poznamenat, že se CSS3 stejně jako HTML5 stále vyvíjí a jeho specifikace se dělí na několik samostatných modulů (jako jsou například transformace, animace a přechody) z důvodu snazší implementace ze strany prohlížečů.

Experimentální funkce jazyka CSS3 nesou prefixy jejich tvůrců, aby se zabránilo konfliktům ve jmenném prostoru, dojde-li ke změnám specifikace. Chcete-li použít zakulacené hrany, gradienty, stíny anebo transformace, neobejdete se nyní bez prefixů, jako je `-moz-` (pro Mozillu), `-o-` (pro Operu) a `-webkit-` (pro prohlížeče založené na WebKitu, jako je například Safari anebo Chrome).

Výpis 1.2: Šablona kaskádových stylů ukázkové stránky HTML5

```
body {  
    background-color:#CCCCCC;  
    font-family:Geneva,Arial,Helvetica,sans-serif;  
    margin: 0px auto;  
    max-width:900px;  
    border:solid;  
    border-color:#FFFFFF;  
}  
  
header {  
    background-color: #F47D31;  
    display:block;  
    color:#FFFFFF;  
    text-align:center;  
}  
  
header h2 {  
    margin: 0px;  
}  
  
h1 {  
    font-size: 72px;  
    margin: 0px;  
}  
  
h2 {  
    font-size: 24px;  
    margin: 0px;  
    text-align:center;  
    color: #F47D31;
```

```
}

h3 {
    font-size: 18px;
    margin: 0px;
    text-align:center;
    color: #F47D31;
}

h4 {
    color: #F47D31;
    background-color: #fff;
    -webkit-box-shadow: 2px 2px 20px #888;
    -webkit-transform: rotate(-45deg);
    -moz-box-shadow: 2px 2px 20px #888;
    -moz-transform: rotate(-45deg);
    position: absolute;
    padding: 0px 150px;
    top: 50px;
    left: -120px;
    text-align:center;
}

nav {
    display:block;
    width:25%;
    float:left;
}

nav a:link, nav a:visited {
    display: block;
    border-bottom: 3px solid #fff;
    padding: 10px;
    text-decoration: none;
    font-weight: bold;
    margin: 5px;
}

nav a:hover {
    color: white;
    background-color: #F47D31;
}
```

```
nav h3 {
    margin: 15px;
    color: white;
}

#container {
    background-color: #888;
}

section {
    display: block;
    width: 50%;
    float: left;
}

article {
    background-color: #eee;
    display: block;
    margin: 10px;
    padding: 10px;
    -webkit-border-radius: 10px;
    -moz-border-radius: 10px;
    border-radius: 10px;
}

articleheader {
    -webkit-border-radius: 10px;
    -moz-border-radius: 10px;
    border-radius: 10px;
    padding: 5px;
}

articlefooter {
    -webkit-border-radius: 10px;
    -moz-border-radius: 10px;
    border-radius: 10px;
    padding: 5px;
}

article h1 {
    font-size: 18px;
}
```

```
aside {
    display: block;
    width: 25%;
    float: left;
}

aside h3 {
    margin: 15px;
    color: white;
}

aside p {
    margin: 15px;
    color: white;
    font-weight: bold;
    font-style: italic;
}

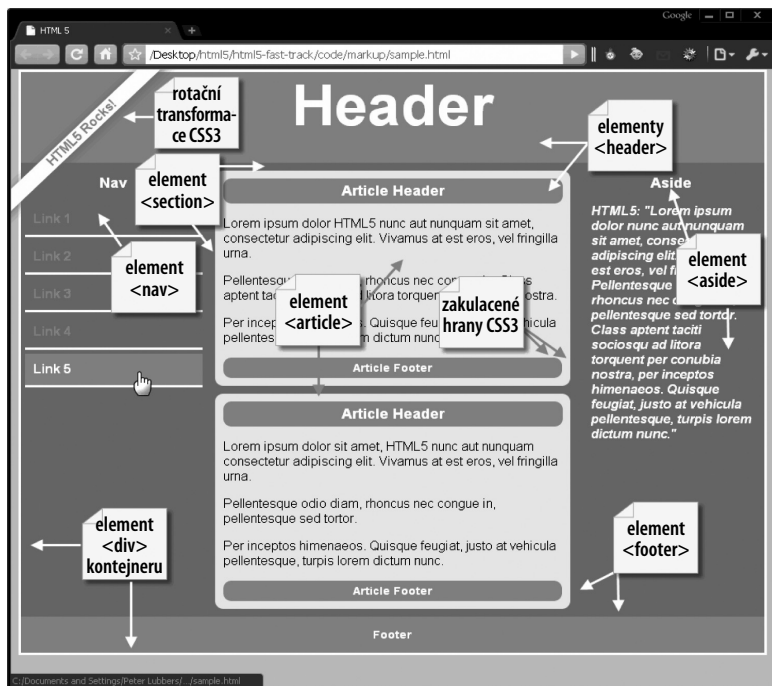
footer {
    clear: both;
    display: block;
    background-color: #F47D31;
    color: #FFFFFF;
    text-align: center;
    padding: 15px;
}

footer h2 {
    font-size: 14px;
    color: white;
}

/* odkazy */
a {
    color: #F47D31;
}

a:hover {
    text-decoration: underline;
}
```

Na obrázku 1.2 můžete vidět stránku z výpisu 1.1, s aplikovanými styly (CSS a CSS3). Mějte však na paměti, že nic takového jako typická stránka HTML5 neexistuje. Možné je doslova cokoli a tento příklad používá mnohé z nových značek především pro demonstrační účely.



Obrázek 1.2: Stránka HTML5 se všemi novými sémantickými elementy

Vzhledem k vykreslení obsahu v prohlížeči se může jevit, jako by tento všechny tyto nové elementy znal. Pravdou však je, že mohlo dojít k přejmenování těchto elementů například na `foo` a `bar` a k následné aplikaci stylů, tak aby se vykreslily totožným způsobem (v takovém případě však samozřejmě nepřinesou z pohledu optimalizace pro vyhledávače žádné výhody). Jedinou výjimkou je v tomto hledu Internet Explorer, který vyžaduje, aby byly elementy součástí DOM. Chcete-li tedy tyto elementy vidět v IE, musíte je programově vložit do DOM a zobrazit je jako blokové elementy.

Zjednodušení výběru pomocí rozhraní Selector

Společně s novými sémantickými elementy přichází HTML5 také s novými způsoby, jak v rámci dokumentu vyhledávat elementy. Tabulka 1.3 ukazuje metody objektu `document`, které vývojářům původně umožňovaly vyhledávání konkrétních elementů stránky.

Tabulka 1.3: Předchozí metody JavaScriptu pro vyhledávání elementů

Metoda	Popis	Příklad
getElementById	Vrací element se zadanou hodnotou atributu id.	<code><div id="foo"></code> <code>getElementById("foo");</code>
getElementsByName	Vrací všechny elementy, jejichž atribut name má zadanou hodnotu.	<code><input type="text" name="foo"></code> <code>getElementsByName("foo");</code>
getElementsByTagName	Vrací všechny elementy, jejichž název značky odpovídá zadané hodnotě.	<code><input type="text"></code> <code>getElementsByTagName("input");</code>

Nové rozhraní API Selector nabízí mnohem sofistikovanější způsoby, jak specifikovat elementy, které chcete získat, aniž by bylo nutné procházet v cyklu dokumentem s použitím standardního modelu DOM. Rozhraní Selectoru umožňuje používat selektory jazyka CSS k vyhledání jednoho či více elementů ve stránce. Pomocí selektorů lze vybírat elementy v závislosti na jejich zanoření, příbuznosti či vztahu rodič-potomek. Nejnovější verze jazyka CSS přidává podporu většího počtu pseudotříd (například je-li objekt aktivní, neaktivní anebo pole zaškrtnuté) a umožňuje určení téměř jakékoli kombinace vlastností a hierarchie, jakou si lze vůbec představit. K výběru elementů pomocí selektorů můžete jednoduše použít některou z metod uvedených v tabulce 1.4.

Tabulka 1.4: Nové metody QuerySelector

Metoda	Popis	Příklad	Výsledek
querySelector	Vrací první element ve stránce odpovídající zadanému selektoru či selektorům.	<code>querySelector("input.error");</code>	Vrátí element prvního vstupního pole s třídou <code>error</code> .
querySelectorAll	Vrací všechny elementy odpovídající zadanému selektoru či selektorům.	<code>querySelectorAll("#resulttd");</code>	Vrátí jakékoli buňky tabulky nacházející se uvnitř elementu s identifikátorem <code>results</code> .

Metodám rozhraní Selector je možné zadávat také více selektorů současně, například takto:

```
// vybere první element v dokumentu s třídou
// highClass anebo lowClass
var x = document.querySelector(".highClass", ".lowClass");
```

V případě metody `querySelector` se vrátí první element odpovídající zadanému selektoru. V případě metody `querySelectorAll` se vrátí všechny elementy odpovídající zadanému selektoru. Jednotlivé selektory je možné oddělovat čárkou.

Nové rozhraní Selector usnadňuje výběr těch sekcí dokumentu, které bylo dříve obtížné vyhledat. Řekněme, že si například přejete vyhledat buňku tabulky, nad kterou se právě nachází kurzor myši. Výpis 1.3 ukazuje, jak snadno je tento problém s použitím selektoru řešitelný. Tento soubor najdete také ve složce `kody/uvod`.

Výpis 1.3: Použití rozhraní Selector

```
<!DOCTYPE html>
<html>

<head>
  <meta charset="utf-8" />
  <title>Ukázka použití metody querySelector</title>

  <style type="text/css">
    td {
      border-style: solid;
      border-width: 1px;
      font-size: 300%;
    }

    td:hover {
      background-color: cyan;
    }

    #hoverResult {
      color: green;
      font-size: 200%;
    }
  </style>
</head>

<body>
  <section>
    <!-- vytvoření tabulky s třemi řádky i sloupci -->
    <table>
      <tr>
        <td>A1</td> <td>A2</td> <td>A3</td>
      </tr>
      <tr>
        <td>B1</td> <td>B2</td> <td>B3</td>
      </tr>
      <tr>
        <td>C1</td> <td>C2</td> <td>C3</td>
      </tr>
    </table>

    <div>Označte tlačítko, najedte kurzorem myši na buňku tabulky a
      stiskněte klávesu Enter. Buňka se identifikuje voláním
      querySelector('td:hover').</div>
```

```
<button type="button" id="findHover" autofocus>Vyhledat 'td: hover'  
</button>  
<div id="hoverResult"></div>  
  
<script type="text/javascript">  
  document.getElementById("findHover").onclick = function() {  
    // určení buňky tabulky, nad kterou se právě nachází kurzor myši  
    var hovered = document.querySelector("td: hover");  
    if (hovered)  
      document.getElementById("hoverResult").innerHTML = hovered.  
        innerHTML;  
  }  
</script>  
</section>  
  
</body>  
</html>
```

Jak je z tohoto příkladu patrné, je vyhledání elementu, nad kterým se právě nachází kurzor myši, záležitostí jediného řádku kódu:

```
var hovered = document.querySelector("td: hover");
```



Poznámka

Rozhraní Selector je nejenom šikovné, ale často také rychlejší než procházení modelem DOM v případě zastaralých metod vyhledávání elementů. Prohlížeče jsou s ohledem na kaskádové styly velmi dobře optimalizované na vyhledávání elementů podle selektorů.

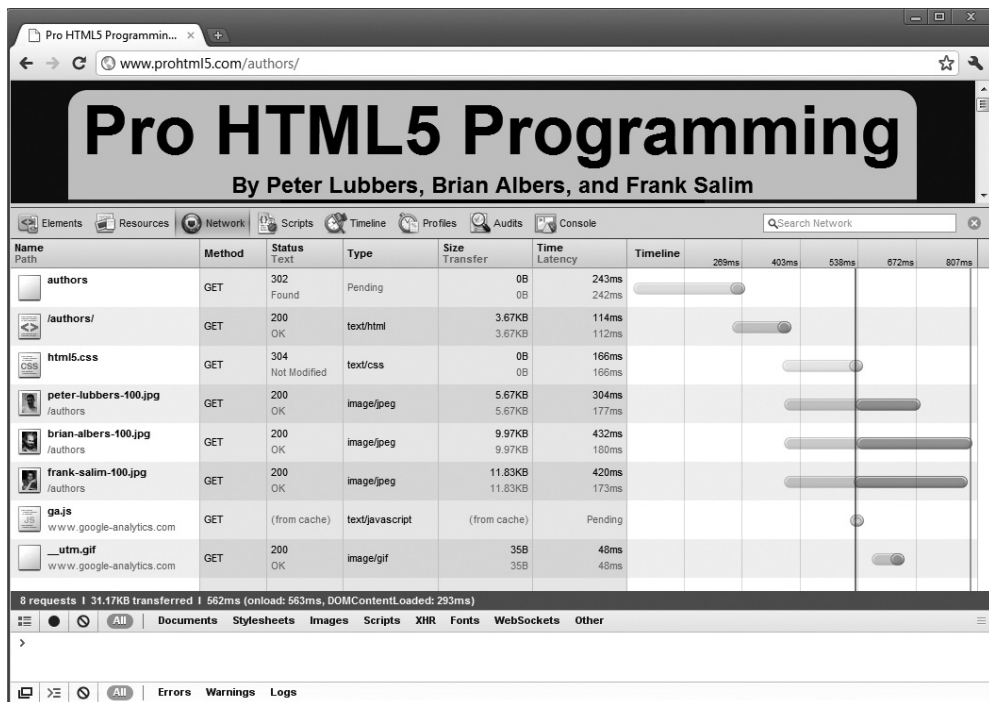
Není žádným překvapením, že konsorcium W3C oddělilo formální specifikaci selektorů od specifikace jazyka CSS. Jak jste se sami mohli přesvědčit, jsou selektory užitečné i mimo oblast kaskádových stylů. Detailní informace o nových selektorech spadají za rámec této knihy. Pokud ale jako vývojář hledáte optimální způsob manipulace s modelem DOM, lze rozhraní Selector pro rychlou navigaci strukturou dokumentu jediné doporučit.

Protokolování a ladění JavaScriptu

Přestože se technicky vzato nejedná o funkci HTML5, protokolování JavaScriptu a podpora ladění na straně prohlížeče za posledních několik let významně zlepšily. Prvním skvělým nástrojem pro analýzu webových stránek a kódu v nich obsaženého byl Firebug, doplněk pro Firefox.

Podobné funkce nyní najdete i ve vývojových nástrojích integrovaných do všech ostatních prohlížečů – v Safari a jeho Web Inspector, v Google Chrome a jeho Developer Tools, v Internet Exploreru a jeho Developer Tools a v Opeře a jejím Dragonfly. Obrázek 1.3 ukazuje nástroj Developer Tools prohlížeče Google Chrome (pro přístup k němu můžete použít klávesovou zkratku

Ctrl + Shift + J) s bezpočtem informací o webových stránkách. Mimo jiné zde najdete i konzolu pro ladění, prohlížeč zdrojů a úložišť.



Obrázek 1.3: Nástroj Nástroje pro vývojáře prohlížeče Google Chrome

Mnohé z ladicích nástrojů umožňují nastavit ladicí body (tzv. breakpointy), které pozastaví provádění kódu a umožní analyzovat aktuální stav programu a proměnných. Rozhraní `console.log` se pro vývojáře v JavaScriptu stalo defacto standardem pro protokolování. Mnohé z prohlížečů nabízí panel uvádějící zprávy zaznamenané do konzoly. Použití rozhraní `console.log` je mnohem lepší volbou než volání funkce `alert`, protože nepozastavuje běh programu.

window.JSON

JSON je relativně novým a stále populárnějším způsobem reprezentace dat. Jedná se o podmnožinu syntaxe JavaScriptu, která reprezentuje datové objekty v textové podobě. Díky své jednoduchosti a přirozenému použití v JavaScriptu se JSON stal defacto standardem pro výměnu dat HTML5 aplikací. Kanonické rozhraní API pro JSON má dvě metody: `parse` a `stringify` (serializace neboli převedení na řetězec).

Použití JSON ve starších prohlížečích vyžaduje specializovanou knihovnu JavaScriptu (několik najdete na adrese <http://json.org>). Zpracování a serializace nejsou v JavaScriptu vždy tak rychlé, jak by se slušelo, a právě z tohoto důvodu nabízí novější prohlížeče vlastní implementaci JSON, kterou je v JavaScriptu možné volat. Nativní podpora JSON je součástí specifikace standardu ECMAScript 5, představujícího další generaci jazyka JavaScript. Jedná se o jednu z prvních částí ECMAScript 5, která se dočkala široké implementace. Každý moderní prohlížeč nabízí objekt `window.JSON` a v aplikacích HTML5 lze očekávat hojně využití JSON.

DOM Level 3

Jednou z nejvíce odsuzovaných částí vývoje webových aplikací vždy byla obsluha událostí. Ačkoli většina prohlížečů podporuje standardní rozhraní API pro události a elementy, Internet Explorer je jiný. Již v dávných dobách implementoval model událostí lišící se od konečného standardu. Internet Explorer 9 bude podporovat funkce DOM Level 2 a 3, takže konečně můžete pro manipulaci s modelem DOM a obsluhu událostí použít ve všech prohlížečích HTML5 stejný kód. To zahrnuje i dvě velmi důležité metody: `addEventListener` a `dispatchEvent`.

Monkeys, Squirrelfish a další podivnosti

Poslední inovace prohlížečů se netýká pouze nových značek a rozhraní. Jednou z nejvýznamnějších změn je významná evoluce jádra JavaScriptu/ECMAScriptu v čelních prohlížečích. Podobně jako nová rozhraní API otevírají nové možnosti, o kterých se v předchozí generaci prohlížečů nikomu ani nesnilo, zvyšují rychlost provádění kódu jak stávající webové aplikace, tak ty, které používají nejnovější funkce HTML5. Myslíte, že prohlížeč nezvládne komplexní práci s obrázky a daty anebo editaci zdlouhavých textů? Zamyslete se znovu.

Posledních několik let soutěžili vývojáři prohlížečů o pomyslný titul nejrychlejšího jádra JavaScriptu. Zatímco ve svých prvních verzích byl JavaScript čistě interpretovaným jazykem, nejnovější inkarnace kompilují programový kód přímo do nativního strojového kódu a v porovnání s prohlížeči poloviny prvního desetiletí 21. století nabízí zrychlení několika řádů.

Celé toto snažení začalo, když v roce 2006 věnovala společnost Adobe své kompilační jádro JIT (Just-In-Time) a virtuální stroj pro ECMAScript, pod kódovým označením Tamarin, projektu Mozilla. Přestože v nejnovějších verzích Mozilly zůstávají už pouhé zbytky technologie Tamarin, jeho darování napomohlo zrodu nových skriptovacích jader v jednotlivých prohlížečích s názvy podobně neobvyklými a fascinujícími, jako jsou jejich deklarované výkony.

Tabulka 1.5: Jádra JavaScriptu jednotlivých webových prohlížečů

Prohlížeč	Název jádra	Poznámky
Apple Safari 5	Nitro (jinak známý jako SquirrelFishExtreme)	Toto jádro, které spatřilo světlo světa v Safari 4 a ve verzi 5 se dočkalo vylepšení, nabízí optimalizaci byte-kódu a nativní context-threaded překladač.
Google Chrome 5	V8	Dostupné od Google Chrome 2, používá generační garbage-collection pro zajištění vysoké paměťové škálovatelnosti bez přerušení.
Microsoft Internet Explorer 9	Chakra	Zaměřuje se na kompilaci na pozadí a efektivní systém typů a v porovnání s IE8 představuje výrazný krok kupředu.
MozillaFirefox 4	JägerMonkey	Toto jádro, zdokonalené ve verzi 3.5, kombinuje rychlou interpretaci s nativní kompilací z trasovacích stromů.
Opera 10.60	Carakan	Používá byte-kód založený na registrech a selektivní nativní kompilaci. Hlásá zlepšení až 75% v porovnání s verzí 10.50.

Tato zdravá soutěživost mezi jednotlivými výrobci prohlížečů v konečném důsledku přibližuje výkon JavaScriptu ještě blíže běžným desktopovým aplikacím.

A ještě pár slov k HTML

Peter říká: „Když je řeč o soutěžení a rychlosti, běh je mou velkou vášní – hodně běhání.

Maraton představuje skvělý sport, při kterém potkáte mnoho úžasných lidí. Když dobíháte poslední míle závodu na 100 nebo 165 mil, poznáte vskutku zajímavé lidi ve zcela novém světle. V tomto okamžiku je vaše duše úplně odhalena až na svou podstatu, na místě, kde může začít velké přátelství. Stále je zde soutěživý duch, prim však hraje kamarádství. Na tomto místě však odbočím.

Abych sledoval, jak se mým kamarádům daří v závodech, kterých se nemůžu účastnit (například proto, že píšu knihu o HTML5), zpravidla navštěvuji weby daných závodů. Není žádným překvapením, že možnost živého sledování je často nespolehlivá.

Před několika lety jsem narazil na web jednoho evropského závodu, který měl vskutku dobrý nápad. Přední závodníci dostali přijímače GPS a jejich poloha se zobrazovala na mapě (něco podobného si s použitím rozhraní Geolocation a WebSocket ukážeme i v této knize). Přestože se jednalo o vcelku primitivní implementaci (uživatelé museli obnovovat stránku, aby viděli aktuální stav), okamžitě jsem si všiml velkého potenciálu.

Nyní, jen o několik let později, nám HTML5 dává k dispozici nástroje v podobě rozhraní Geolocation pro lokalizaci a WebSockets pro aktualizace v reálném čase, pomocí nichž lze podobné sledovací weby snadno vytvářet. Není o tom žádných pochyb – HTML5 proběhlo cílovou páskou jako vítěz.“

Shrnutí

V této kapitole jste získali obecný přehled o základech jazyka HTML5.

Seznámili jsme se s historií jeho vývoje a některými důležitými daty, která se ho týkají. Představili jsme si také čtyři nové zásady éry HTML5, která právě nadchází: kompatibilitu, použitelnost, interoperabilitu a obecný přístup. Každá z těchto zásad otevírá nové možnosti a staví se proti principům, které se nyní už považují za zastaralé. Poté jsme se seznámili s novým paradigmatem HTML5, který nevyužívá zásuvné moduly, a odpověděli na otázku, která napadla snad každého: co patří do HTML5 a co ne. Seznámili jsme se s novinkami HTML5, jako je deklarace `DOCTYPE` a znakové sady, řadou nových elementů a také se závodem v oblasti výkonnosti JavaScriptu.

V následující kapitole se podíváme na HTML5 z pohledu programátora a prozkoumáme HTML5 rozhraní Canvas.

KAPITOLA 2

Rozhraní Canvas

V této kapitole prozkoumáme, co přináší rozhraní HTML5 s názvem Canvas, tedy rozhraní umožňující dynamicky generovat a vykreslovat grafiku, grafy, obrázky i animace. Provedeme vás základy použití tohoto vykreslovacího rozhraní a ukážeme si obrázek, který může měnit svou velikost a upravovat se na základě prostředí prohlížeče. Dozvíte se, jak vytvořit dynamické obrázky na základě uživatelského vstupu. Seznámíte se samozřejmě také s úskalími použití Canvasu HTML5 a s postupy, jak se jim vyhnout.

Tato kapitola předpokládá pouze minimální množství znalostí práce s grafikou, nebojte se tedy přidat a vyzkoušet některé z nejmočnějších funkcí HTML5.

Seznámení s rozhraním Canvas jazyka HTML5

O rozhraní Canvas jazyka HTML5 by bylo možné napsat celou knihu (a nebyla by malá). Protože máme pouze jedinou kapitolu, probereme v ní to, co považujeme za nejčastěji používané funkce tohoto velmi rozsáhlého rozhraní.

Historie

Princip plátna původně představil Apple ve svém WebKitu pro Mac OS X, za účelem vytvoření widgetů pracovní plochy. Před příchodem plátna bylo možné v prohlížeči realizovat malování pouze skrze zásuvné moduly, jako jsou například ty pro Internet Explorer od Adobe pro Flash, SVG (ScalableVectorGraphics) a VML (VectorMarkupLanguage) anebo jiné, čistě javascriptové techniky.

Zkuste bez elementu `canvas` například nakreslit obyčejnou diagonální čáru. Zní to jednoduše, ale jedná se o poměrně komplexní problém, nemáte-li k dispozici rozhraní pro snadné malování ve 2D. HTML5 rozhraní Canvas právě tohle umožňuje, a protože se jedná o velmi užitečnou funkci prohlížeče, našla si cestu do specifikace HTML5.

Apple původně s ohledem na návrh specifikace plátna organizace WHATWG poukazoval na svá vlastnická práva, což v té době vyvolalo obavy mezi některými stoupenci standardizace webu. Nakonec však Apple souhlasil s použitím v rámci volné licence W3C.

SVG vs. Canvas

Peter říká: „Canvas je v zásadě *bitmapové* plátno a obrázky na něm nakreslené jsou tím pádem finální a není možné měnit jejich velikost tak, jak je to možné u obrázku SVG (ScalableVectorGraphic). Objekty nakreslené na plátno navíc nejsou součástí modelu DOM stránky ani žádného jmenného prostoru, což se považuje za slabinu. Naproti tomu velikost obrázků SVG se může v závislosti na rozlišení snadno měnit a detekovat klepnutí myši na ně (a přesně tak vědět, kam uživatelé uživatel klepnul).

Proč by tedy specifikace HTML5 od WHATWG nepoužívala právě SVG? Navzdory zjevným nedostatkům má rozhraní Canvas dvě velké výhody: nabízí slušný výkon, protože nemusí ukládat objekty popisující jednotlivé nakreslené křivky, a implementace rozhraní Canvas na základě mnoha populárních dvourozměrných rozhraní pro kreslení z jiných programovacích jazyků je relativně snadná. V konečném důsledku je lepší mít vrabce v hrsti než holuba na střeše.“

Co je to plátno?

Když ve stránce použijete element `canvas`, vytvoří se v ní obdélníková oblast. Ve výchozím stavu je tato oblast 300 pixelů široká a 150 pixelů vysoká, můžete však zadat přesnou požadovanou velikost, stejně tak jako jiné atributy elementu `canvas`. Výpis 2.1 ukazuje nejzákladnější element `canvas`, který je možné do stránky HTML5 přidat.

Výpis 2.1: Základní použití elementu `canvas`

```
<canvas></canvas>
```

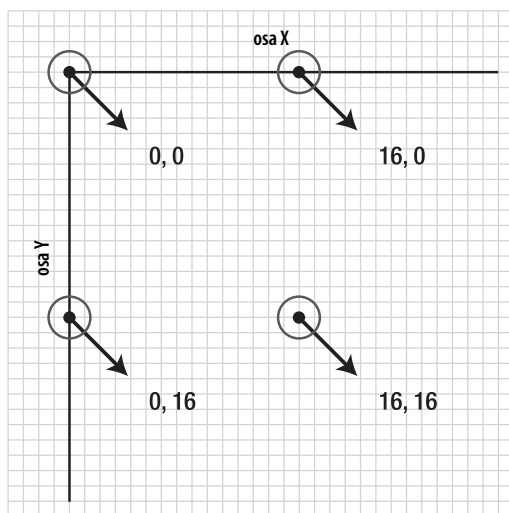
Když element `canvas` přidáte do stránky, můžete s ním pomocí JavaScriptu jakkoli pracovat. Můžete do něj přidávat grafiku, čáry či texty, můžete v něm malovat, a můžete do něj dokonce i vkládat pokročilé animace.

Rozhraní Canvas podporuje stejné dvourozměrné grafické operace, jaké najdete ve většině moderních operačních systémů a frameworků. Pokud jste někdy v posledních letech programovali dvourozměrnou grafiku, budete si v rozhraní Canvas, které se svým návrhem podobá obdobným existujícím systémům, nejspíše připadat jako doma. V opačném případě se sami přesvědčíte o tom, o kolik dokáže být vykreslovací systém mocnější než dřívější triky s obrázky a kaskádovými styly, které vývojáři řadu let používali při tvorbě webové grafiky.

Abyste mohli programově použít plátno, musíte nejdříve získat jeho *kontext*. S kontextem následně může provádět různé akce a v závěru je pak na kontext aplikovat. Na úpravy obsahu plátna je možné pohlížet jako na databázové transakce – transakce se zahájí, provedou se požadované akce a následně se transakce dokončí (provede se tzv. *commit*).

Souřadnice plátna

Jak ukazuje obrázek 2.1, souřadnice plátna začínají na hodnotách $x = 0$ a $y = 0$ v levém horním rohu, který budeme označovat jako *počátek*, a zvyšují se (po pixelech) horizontálně po ose X a vertikálně po ose Y.



Obrázek 2.1: Souřadnice plátna na osách X a Y

Kdy plátno nepoužívat

Přestože je element `canvas` skvělým pomocníkem, který jistě najde uplatnění, nepoužívejte ho v případech, kdy zcela postačí jiný element. Není například dobrý nápad dynamicky kreslit všechny různé nadpisy dokumentu HTML pomocí plátna, když stačí použít běžné nadpisové elementy (`h1`, `h2` atd.), které jsou pro tento účel stvořené.