

Kniha  
obsahuje  
**26**  
výukových  
projektů

Brian P. Hogan



# HTML5 a CSS3

**Výukový kurz  
webového vývojáře**

Rychlé přehledy  
revolučních novinek

Výuka na konkrétních projektech

Nouzová řešení  
pro starší prohlížeče

© PRESS

**Brian P. Hogan**

# **HTML5 a CSS3**

## **Výukový kurz webového vývojáře**

---

**Computer Press**  
**Brno**  
**2012**

# HTML5 a CSS3

## Výukový kurz webového vývoje

**Brian P. Hogan**

**Překlad:** Jakub Urban

**Obálka:** Martin Sodomka

**Odpovědný redaktor:** Martin Domes

**Technický redaktor:** Jiří Matoušek

Authorized translation from English language edition Pragmatic Programmers, LLC. All rights reserved.

Original copyright: © 2011 The Pragmatic Programmers, LLC. All rights reserved.

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

Autorizovaný překlad z originálního anglického vydání Pragmatic Programmers, LLC. All rights reserved.

Originální copyright: © 2011 The Pragmatic Programmers, LLC. All rights reserved.

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

Objednávky knih:

<http://knihy.cpress.cz>

[www.albatrosmedia.cz](http://www.albatrosmedia.cz)

[eshop@albatrosmedia.cz](mailto:eshop@albatrosmedia.cz)

bezplatná linka 800 555 513

ISBN 978-80-251-3576-1

Vydalo nakladatelství Computer Press v Brně roku 2012 ve společnosti Albatros Media a. s. se sídlem Na Pankráci 30, Praha 4. Číslo publikace 16075.

© Albatros Media 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.

Dotisk 1. vydání

 **ALBATROS** MEDIA a.s.

# Obsah

<b>O KNIZE</b> .....	<b>11</b>
<b>PODĚKOVÁNÍ</b> .....	<b>12</b>
<b>ÚVODEM</b> .....	<b>14</b>
HTML5: platforma vs. specifikace .....	15
Jak pracovat s touto knihou. ....	15
Co v této knize najdete .....	16
Co kniha vyžaduje .....	16
On-line zdroje. ....	17
 <b>KAPITOLA 1</b>	
<b>PŘEHLED HTML5 A CSS3</b> .....	<b>19</b>
Platforma pro webový vývoj .....	19
Zpětná kompatibilita. ....	22
Budoucí cesta je trnitá .....	23
HTML5 a CSS3 jsou stále ve vývoji .....	27

## ČÁST I

### ZLEPŠOVÁNÍ UŽIVATELSKÉHO ROZHRANÍ

<b>KAPITOLA 2</b>	
<b>NOVÉ STRUKTURÁLNÍ ZNAČKY A ATRIBUTY</b> .....	<b>31</b>
Tip 1 Předefinování blogu pomocí sémantického zdrojového kódu. ....	34
Vše začíná se správnou deklarací typu dokumentu .....	35
Záhlaví .....	36
Zápatí .....	37
Navigace. ....	37

Sekce . . . . .	38
Články . . . . .	38
Poznámky po straně . . . . .	40
Postranní panel . . . . .	41
Stylování . . . . .	42
Nouzové řešení . . . . .	45
Tip 2 Tvorba automaticky otevíraných oken s vlastními datovými atributy . . . . .	46
Oddělení chování od obsahu, neboli proč je onClick špatná volba . . . . .	46
Vylepšení přístupnosti . . . . .	47
Zrušení metody onclick . . . . .	47
Zachrání nás vlastní datové atributy! . . . . .	48
Nouzové řešení . . . . .	49
Budoucnost . . . . .	49

### KAPITOLA 3

## TVORBA UŽIVATELSKY PŘÍVĚTIVÝCH WEBOVÝCH

### FORMULÁŘŮ . . . . .51

Tip 3 Popisování dat pomocí nových vstupních polí . . . . .	54
Vylepšení vytvářeného formuláře . . . . .	54
Nouzové řešení . . . . .	58
Tip 4 Přeskočení na první pole pomocí automatického zaměření . . . . .	61
Nouzové řešení . . . . .	61
Tip 5 Poskytnutí nápovědy pomocí výplňového textu . . . . .	62
Jednoduchý přihlašovací formulář . . . . .	62
Zabránění automatickému dokončování . . . . .	63
Nouzové řešení . . . . .	64
Tip 6 Místní úpravy pomocí atributu contenteditable . . . . .	69
Profilový formulář . . . . .	70
Uchování dat . . . . .	71
Nouzové řešení . . . . .	71
Budoucnost . . . . .	74

### KAPITOLA 4

## TVORBA LEPŠÍCH UŽIVATELSKÝCH ROZHRAŇÍ

### POMOCÍ CSS3 . . . . .77

Tip 7 Stylování tabulek pomocí pseudotříd . . . . .	80
Vylepšení faktury . . . . .	80

Pruhované řádky pomocí pseudotřídy :nth-of-type . . . . .	82
Zarovnání textu sloupců	
pomocí pseudotřídy :nth-child . . . . .	83
Přidělení tučnosti poslednímu řádku	
pomocí pseudotřídy :last-child . . . . .	84
Odečítání pomocí pseudotřídy :nth-last-child . . . . .	86
Nouzové řešení . . . . .	87
Tip 8 Tvorba tisknutelných odkazů pomocí :after a content . . . . .	89
CSS . . . . .	90
Nouzové řešení . . . . .	90
Tip 9 Tvorba vícesloupcového layoutu . . . . .	93
Rozdělení na sloupce . . . . .	93
Nouzové řešení . . . . .	97
Tip 10 Tvorba mobilních rozhraní pomocí Media Queries . . . . .	99
Nouzové řešení . . . . .	100
Budoucnost . . . . .	101

## KAPITOLA 5

### **ZLEPŠENÍ PŘÍSTUPNOSTI. . . . .103**

Tip 11 Poskytnutí navigačních rad pomocí ARIA rolí . . . . .	105
Role orientačních bodů . . . . .	106
Role dokumentové struktury . . . . .	108
Nouzové řešení . . . . .	109
Tip 12 Tvorba přístupné aktualizující se oblasti . . . . .	110
Tvorba stránky . . . . .	111
Nouzové řešení . . . . .	114
Budoucnost . . . . .	115

## ČÁST II

### **KRESLENÍ, VIDEO A AUDIO A NOVÝ VZHLED**

## KAPITOLA 6

### **KRESLENÍ S ELEMENTEM CANVAS . . . . .119**

Tip 13 Kreslení loga. . . . .	121
Kreslení loga . . . . .	123
Přidání textu . . . . .	124
Kreslení čar . . . . .	124

Posunutí počátku . . . . .	125
Přidání barev. . . . .	126
Nouzové řešení. . . . .	126
Tip 14 Statistické grafy pomocí RGraph. . . . .	128
Popisování dat pomocí HTML . . . . .	129
Proměňujeme HTML v sloupcový graf . . . . .	130
Zobrazení alternativního obsahu . . . . .	131
Nouzové řešení. . . . .	132
Budoucnost . . . . .	135

## KAPITOLA 7

### **VKLÁDÁNÍ AUDIA A VIDEO . . . . . 137**

Stručná historie . . . . .	138
Kontejnery a kodeky . . . . .	139
Kodeky videa . . . . .	139
Kodeky pro audio . . . . .	141
Kontejnery a kodeky společně . . . . .	142
Tip 15 Pracujeme s audiem. . . . .	143
Vytvoření základního seznamu . . . . .	143
Nouzové řešení. . . . .	145
Tip 16 Vkládání videa . . . . .	147
Nouzové řešení. . . . .	148
Omezení videa v HTML5 . . . . .	152
Audio, video a přístupnost . . . . .	153
Budoucnost . . . . .	153

## KAPITOLA 8

### **PŮSOBIVÉ UŽIVATELSKÉ ROZHRAŇÍ . . . . . 155**

Tip 17 Zaohlení ostrých rohů . . . . .	157
Zjemnění přihlašovacího formuláře. . . . .	157
Selektory podle prohlížeče . . . . .	159
Nouzové řešení. . . . .	159
Tip 18 Pracujeme se stíny, přechody a transformacemi. . . . .	164
Základní struktura. . . . .	164
Přidání přechodu . . . . .	166
Přidání stínu za štítek . . . . .	167

Rotace štítku . . . . .	168
Průhledná pozadí . . . . .	169
Nouzové řešení . . . . .	170
Tip 19 Použití skutečných fontů . . . . .	175
@font-face . . . . .	176
Formáty fontů . . . . .	177
Změna fontu . . . . .	177
Nouzové řešení . . . . .	179
Budoucnost . . . . .	179

## ČÁST III

### ZA HRANICÍ HTML5

#### KAPITOLA 9

#### **PRACUJEME S DATY NA STRANĚ KLIENTA . . . . .183**

Tip 20 Uložení předvoleb pomocí localStorage . . . . .	186
Tvorba formuláře předvoleb . . . . .	186
Uložení a načtení nastavení . . . . .	187
Použití nastavení na stránku . . . . .	188
Nouzové řešení . . . . .	189
Tip 21 Ukládání dat v relační databázi na straně klienta . . . . .	192
CRUD ve vašem prohlížeči . . . . .	192
Rozhraní systému poznámek . . . . .	193
Připojení k databázi . . . . .	195
Vytvoření tabulky poznámek . . . . .	196
Načítání poznámek . . . . .	197
Načtení konkrétního záznamu . . . . .	198
Vkládání, aktualizace a mazání záznamů . . . . .	199
Finální úpravy . . . . .	201
Nouzové řešení . . . . .	202
Tip 22 Pracujeme off-line . . . . .	204
Definování vyrovnávací paměti pro manifest . . . . .	204
Manifest a ukládání do vyrovnávací paměti . . . . .	205
Budoucnost . . . . .	206



## KAPITOLA 10

**VYUŽITÍ DALŠÍCH API . . . . .207**

Tip 23 Uchování historie . . . . .	209
Ukládání aktuálního stavu. . . . .	209
Získání předchozího stavu . . . . .	210
Vrácení do výchozího stavu . . . . .	211
Nouzové řešení. . . . .	211
Tip 24 Komunikace mezi doménami. . . . .	212
Seznam kontaktů . . . . .	213
Zaslání zprávy. . . . .	214
Web podpory . . . . .	215
Přijímání zpráv . . . . .	217
Nouzové řešení. . . . .	217
Tip 25 Komunikace pomocí webových socketů . . . . .	219
Rozhraní chatu . . . . .	219
Komunikace se serverem . . . . .	221
Nouzové řešení. . . . .	223
Servery . . . . .	225
Tip 26 Jak se najít: geolokace. . . . .	226
Lokalizování naší společnosti. . . . .	226
Jak být nalezen. . . . .	227
Nouzové řešení. . . . .	228
Budoucnost . . . . .	229

## KAPITOLA 11

**DALŠÍ NOVINKY HTML5 A CSS3. . . . .231**

Přechody CSS3. . . . .	233
Časové funkce . . . . .	234
Web Workers. . . . .	235
Nativní podpora techniky táhni a pusť . . . . .	236
Události přetahování objektů myši . . . . .	238
Puštění elementu po přetažení. . . . .	240
Změna stylů . . . . .	241
Přetažení souboru. . . . .	241
Vše není v pořádku. . . . .	241
WebGL . . . . .	242
API pro indexované databáze . . . . .	242

Ověření formuláře na straně klienta. . . . .	.243
Kupředu! . . . . .	.244

## PŘÍLOHA A

<b>RYCHLÝ PŘEHLED FUNKCÍ. . . . .</b>	<b>.245</b>
Nové elementy . . . . .	.245
Atributy . . . . .	.246
Formuláře . . . . .	.247
Atributy formulářových polí. . . . .	.248
Přístupnost . . . . .	.248
Multimédia. . . . .	.249
CSS3 . . . . .	.249
Úložiště na straně klienta . . . . .	.251
Další API . . . . .	.252

## PŘÍLOHA B

<b>PRŮVODCE ZÁKLADY JQUERY . . . . .</b>	<b>.253</b>
Načtení jQuery. . . . .	.253
Základy jQuery . . . . .	.254
Metody ke změně obsahu. . . . .	.255
Hide a Show . . . . .	.255
html, val a attr. . . . .	.255
append, prepend a wrap . . . . .	.255
CSS a třídy . . . . .	.256
Řetězení . . . . .	.257
Tvorba elementů . . . . .	.257
Události . . . . .	.258
bind . . . . .	.258
Původní událost . . . . .	.258
Dokument připraven . . . . .	.259

**PŘÍLOHA C**

**KÓDOVÁNÍ AUDIA A VIDEO . . . . .261**

Kódování audia . . . . .261

Kódování videa pro web . . . . .262

**PŘÍLOHA D**

**ZDROJE . . . . .265**

Zdroje na Internetu . . . . .265

**REJSTŘÍK . . . . .267**

# O knize

Pokud se poohlížíte po tom, jak využít nově vznikající standard HTML5, pak jste narazili na tu správnou knihu. Brianovy praktické zkušenosti a příklady ukazují, jak vyvíjet robustní webové aplikace v prostředí s velkými rozdíly v podpoře ze strany dnešních prohlížečů.

- **Mark Nichols**

Senior konzultant společnosti Microsoft a spolumoderátor podcastu DeveloperSmackDown.com

HTML5 a CSS3 zlepšila mé schopnosti práce na revolučních projektech. Právě jsem začal pracovat na projektu vytvářeném v HTML5 a bez této knihy bych se ani zdaleka necítil tak jistý.

- **Noel Rappin**

Senior konzultant společnosti Obtiva a autor knihy *Rails Test Prescriptions*

Brianova kniha vás bez námahy provede přetvořením vašeho webu v jazycích HTML5 a CSS3 tak, aby vše fungovalo ve všech prohlížečích. V knize popisuje jen to, co funguje již dnes, co naopak zatím nefunguje, a co sledovat během vývoje prohlížečů a standardů samotných.

- **Doug Rhoten**

Senior softwarový vývojář společnosti InterFlow.

# Poděkování

Tuto knihu jsem začal psát, ještě než jsem dokončil knihu předchozí. A ačkoliv si o mně většina mých přátel, rodina a pravděpodobně i vydavatel mysleli, že jsem se zbláznil, když jsem si nevzal žádnou přestávku, všichni mě podporovali. Tato kniha je tedy výsledkem podpory mnoha skvělých a nápomocných lidí.

Prvně nemohu ani dostatečně poděkovat Daveovi Thomasovi a Andy Huntovi za to, že mi dali příležitost s nimi podruhé pracovat. Díky jejich připomínkám v průběhu celého procesu psaní získala kniha skutečné obrysy a je mi ctí, že mohu být autorem vydavatelství Pragmatic Bookshelf.

Daniel Steinberg mi pomohl s knihou v začátcích a jsem mu za jeho podporu a za to, co mě všechno naučil o samotném psaní, velmi vděčný. Kdykoliv píšu, stále slyším jeho hlas, jak mě vede tím správným směrem.

Daniel nemohl dále se mnou na knize pracovat, ale ponechal mě v dobrých rukou. Susannah Pfalzer byla neskutečně nápomocná v průběhu celého procesu, dávala na mě pozor, tlačila mě k lepší práci a vždy mi položila tu správnou otázku v ten správný čas. Bez Susannah by tato kniha nebyla ani zdaleka tak dobrá.

Mí odborní korektoři byli extrémně nápomocní v případě tvarování většiny obsahu a jeho podoby. Děkuji Aaronovi Godinovi, Ali Razové, Charlesi Leffingwellovi, Danielu Steinbergovi, Davidu Kulbergovi, Donu Hentonovi, Dougu Rhotenovi, Ediemu Chlechtingerovi, Jonu Mischovi, Jonu Oebserovi, Kevinu Gisimu, Marcu Harterovi, Marku Nicholsovi, Noelu Rappinovi, Paulu Neibargerovi, Samu Elliotovi, Seanovi Cantonovi, Srdjanu Pejicovi, Stephenu Wolffovi, Toddu Dahlovi a Eriku Watsonovi.

Speciální díky patří všem dobrým lidem v ZenCoder za jejich asistenci při kódování videa pro vzorový příklad a ulehčení práce producentům obsahu při přípravě videa pro HTML5.

Děkuji mým obchodním partnerům Chrise Johnsonovi, Chrisu Warrenovi, Miku Weberovi, Jonu Kinneymu, Adamu Ludwigovi, Garymu Crabtreeovi, Carlu Hooverovi, Joshu Andersonovi, Austen Ottové a Nicku Lamurovi za jejich podporu v tomto a jiných projektech. Zvláštní díky patří Erichu Teskyovi za jeho skvělé přátelství i ve chvílích, kdy se nedaří.

Také bych rád poděkoval mému tátovi za to, že ode mě vždy očekává to nejlepší a za to, že mě nutí, abych to nevzdával, i když věci vypadají beznadějně. Díky tomu je všechno možné.

A nakonec, moje nekonečná vděčnost a láska patří mé nádherné ženě, Carisse, a mým dcerám, Aně a Lise. Vzdaly se spousty víkendů a večerů, abych je mohl trávit psáním v kanceláři. Vždy, když jsem se zadrhl, Carissa se vždy znovu ujišťovala, že vždy věci vyřeším tak, aby byly lepší. Mám neskutečné štěstí, že je mám při sobě.

# Úvodem

Tři měsíce na Internetu je jako rok ve skutečném životě.

Webovní vývojáři uvažují zhruba tímto způsobem, a to protože neustále slýcháme o něčem novém. Před rokem to vypadalo, že HTML5 a CSS3 jsou ještě hudba vzdálené budoucnosti, ale dnes už tyto technologie společnosti používají, protože prohlížeče jako Google Chrome, Safari, Firefox a Opera začínají postupně implementovat jejich specifikace.

HTML5 a CSS3 pomáhají rozvrhnout základ pro příští generaci webových aplikací. Umožňují nám tvořit weby, které se snadněji vyvíjí a spravují, a jsou přívětivější k uživatelům. HTML5 obsahuje nové elementy definující strukturu a vkládání obsahu, což znamená, že nemusíme používat zdrojové kódy navíc nebo zásuvné moduly a další doplňky. CSS3 poskytuje pokročilé selektory, grafická vylepšení a lepší podporu fontů, díky čemuž jsou naše weby vizuálně atraktivnější, aniž by bylo potřeba použít techniku nahrazování obrázků, složitý JavaScript nebo grafické nástroje. Vylepšená podpora přístupnosti zlepšuje ajaxové aplikace pro lidi s omezeními a off-line podpora nám umožňuje začít budovat fungující aplikace, které nevyžadují internetové připojení.

V této knize najdete všechny způsoby, jimiž lze HTML5 a CSS3 použít již nyní, a to i pokud vaši uživatelé nevládní prohlížeče podporující všechny dostupné funkce. Ještě než začneme, zastavme se na chvíli a povykládejme si o HTML5 a o horkých novinkách.

## HTML5: platforma vs. specifikace

HTML5 je specifikace popisující některé nové značky a zdrojový kód, stejně jako báječné javascriptové API, ale zabřednula do víru humbuku a slibů. Naneštěstí se standard HTML5 vyvinul do platformy HTML5 vedoucí k obrovskému zmatení mezi vývojáři, zákazníky, a dokonce i tvůrci. V některých případech se dokonce části specifikace CSS3, jako jsou stíny, přechody a transformace, nazývají jako „HTML“. Prohlížeče se jeden před druhým snaží předvést, kolik z „HTML5“ podporují. A lidé začínají mít podivné požadavky, jako třeba: „Můj web bude napsán v HTML5, že?“

Většinu knihy se zaměříme na samotnou specifikaci HTML5 a CSS3 a na to, jak můžete využít techniky, které popisují. V poslední části knihy se podíváme na sadu příbuzných specifikací, jež jednou byly částí HTML5, ale nyní se používají v rámci více platforem. To zahrnuje webové SQL databáze, geolokaci a webové sockety. Ačkoliv tyto věci nejsou technicky součástí HTML5, mohou vám pomoci vytvořit neskutečné věci, pokud je zkombinujete právě s HTML5 a CSS3.

## Jak pracovat s touto knihou

Každá kapitola v této knize se zaměřuje na specifický okruh problémů, které lze řešit pomocí HTML5 a CSS3. Každá kapitola obsahuje přehled a tabulku shrnující značky, funkce nebo koncepty, jež kapitola pokrývá. Hlavní obsah každé kapitoly je rozdělen do „tipů“, které vám představí určitý koncept a provedou vás za pomoci tohoto konceptu tvorbou jednoduchého příkladu. Kapitoly jsou v této knize seskupeny podle témat. Spíše než abychom věci seskupili na část o HTML5 a část o CSS3, dávalo větší smysl je seskupit podle řešených problémů.

Každý tip obsahuje část nazvanou „Nouzové řešení“, která ukazuje metody vypořádávající se s uživateli používající prohlížeče, jež nenabízejí podporu HTML5 a CSS3. Budeme používat různé techniky k fungování této rezervy, a to od knihoven třetích stran až po vlastní zásuvné moduly knihovny jQuery. Tyto tipy lze číst v jakémkoli pořadí budete chtít.

A nakonec každá kapitola obsahuje část nazvanou „Budoucnost“, v níž probíráme, jak lze koncept použít, až bude široce přijímaný.

Tato kniha se zaměřuje na to, co lze použít již dnes. Existuje řada dalších možností HTML5 a CSS3, které zatím nejsou příliš rozšířené. Více se o nich dozvíte v poslední kapitole.



## Co v této knize najdete

Začneme letným přehledem HTML5 a CSS3 a podíváme se na některé nové strukturální značky, jež lze použít k popsání obsahu vaší stránky. Poté budeme pracovat s formuláři a dostanete šanci použít některá formulářová pole a funkce, jako je automatické zaměření a výplně. Odtud se dostanete k části, v níž si budete hrát s novými selektory CSS3, takže se naučíte, jak styly přidělit elementům bez přidávání zdrojového kódu navíc do vašeho obsahu.

Poté prozkoumáme podporu jazyka HTML v oblasti audia a videa a naučíte se, jak využít element canvas ke kreslení tvarů. Také uvidíte, jak využít CSS3 k tvorbě stínů, přechodů a transformací, a stejně tak se naučíte pracovat s fonty.

V poslední části použijeme funkce HTML5 fungující na straně klienta, jako je webové úložiště, webové SQL databáze a off-line podporu pro tvorbu aplikací na straně klienta. Použijeme webové sockety k vytvoření malé chatovací služby, a uvidíte také, jak HTML5 umožňuje posílat zprávy a data mezi doménami. Také dostanete šanci pohrát si s Geolocation API a naučit se, jak manipulovat s historií prohlížeče. Poté vše shrneme tím, že se podíváme na několik věcí, které nejsou aktuálně užitečné, ale budou důležité v budoucnosti.

V příloze A najdete seznam všech možností popsaných v knize s rychlým odkazem na kapitoly, jež se těmto možnostem věnují. V této knize budeme často využívat knihovnu jQuery, takže příloha B poskytuje rychlý vhled do jejich možností. Dále zde také najdete malou přílohu vysvětlující, jak zakódovat soubory audia a videa pro použití s HTML5.

## Co kniha vyžaduje

Tato kniha je primárně zaměřena na webové vývojáře s dobrou znalostí HTML a CSS. I když zrovna začínáte, kniha pro vás bude stále užitečná, ale spíše bych vám pro začátek doporučil například knihu *Tvorba WWW stránek pro úplné začátečníky* (vydal Computer Press).

Také předpokládám, že máte základní povědomí o JavaScriptu a jQuery<sup>1</sup>, které budeme používat k implementování našich rezervních řešení. Příloha B je rychlým kurzem základů jQuery pokrývajícím základní metody, které budeme používat.

---

<sup>1</sup> [www.jquery.com](http://www.jquery.com)

Abyste mohli kód z knihy otestovat, budete potřebovat některý z těchto prohlížečů nebo jeho novější verzi: Firefox 3.6, Google Chrome 5, Opera 10.6 nebo Safari 5. Pravděpodobně budete potřebovat všechny uvedené prohlížeče, abyste byli schopni otestovat vše, co budeme tvořit, neboť každý prohlížeč dělá to či ono odlišně.

Budete také potřebovat způsob, jak váš web otestovat v Internet Exploreru, abyste se ujistili, že vytvořené rezervní řešení funguje. Pokud potřebujete výsledek testovat ve více verzích prohlížeče Internet Explorer, můžete si stáhnout IETester pro Windows, která podporuje Internet Explorer 6, 7 a 8 v rámci jediné aplikace. Pokud nepoužíváte systém Windows, pak byste měli zvážit využití virtuálního stroje, například VirtualBoxu nebo VMWare, případně použít službu jako CrossBrowserTesting<sup>2</sup> nebo MogoTest<sup>3</sup>.

## Zdrojové kódy ke knize

Z adresy <http://knihy.cpress.cz/k1948> si po klepnutí na odkaz Soubory ke stažení můžete přímo stáhnout archiv s ukázkovými kódy z knihy.

## Zpětná vazba od čtenářů

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 PC literatury  
Computer Press  
Spielberk Office Centre  
Holandská 3  
639 00 Brno  
nebo  
[sefredaktor.pc@cpress.cz](mailto:sefredaktor.pc@cpress.cz)

**Computer Press neposkytuje rady ani živá školení pro značkovací a programovací jazyky.**

<sup>2</sup> <http://crossbrowsertesting.com/>

<sup>3</sup> <http://www.mogotest.com/>

## 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 zobrazíte na adrese <http://knihy.cpress.cz/k1948> po klepnutí na odkaz Soubory ke stažení.

# Přehled HTML5 a CSS3

HTML5<sup>1</sup> a CSS3<sup>2</sup> jsou více než jen dva nové standardy představené Word Wide Web Consortiumem (W3C) a jeho pracovními skupinami. Jsou novou generací každodenně používaných technologií a jsou od toho, aby vám pomohly tvořit lepší moderní webové aplikace. Ještě než se ponoříme do detailů HTML5 a CSS3, popovídejme si o některých výhodách HTML5 a CSS3 a také o určitých výzvách, jimž budeme čelit.

## Platforma pro webový vývoj

Mnoho nových možností HTML se točí kolem tvorby lepší platformy pro webové aplikace. Vývojáři používající HTML5 dostávají do rukou nové nástroje k tvorbě lepšího uživatelského rozhraní – a to od popisnějších značek a lepší komunikace mezi weby nebo okny až k animacím a vylepšené podpoře médií.

---

<sup>1</sup> Specifikaci HTML5 najdete na adrese <http://www.w3.org/TR/html5/>.

<sup>2</sup> Specifikace CSS3 je rozdělena do více modulů a postup prací můžete vidět na adrese <http://www.w3.org/Style/CSS/current-work>.

## Popisnější zdrojový kód

Každá verze HTML uvedla nějaký nový zdrojový kód, ale nikdy předtím zde nebylo tolik nového, co přímo ovlivňuje popisování obsahu. V kapitole 2 se dozvíte o elementech definujících záhlaví, zápatí, části s navigací, postranní panely a články. Také se dozvíte o měřidlech, ukazatelích průběhu a o tom, jak vám vlastní atributy dat mohou pomoci data popsat.

## Multimédia s menší závislostí na zásuvných modulech

Pro video, audio nebo vektorovou grafiku už nepotřebujete Flash nebo Silverlight. Ačkoliv se na Flashi založené přehrávače videa relativně snadno používají, nefungují na zařízeních společnosti Apple. A to je významný trh, takže se budete muset naučit používat alternativy pro flashové video. V kapitole 7 uvidíte, jak audio a video v HTML5 použít včetně efektivního rezervního řešení.

## Lepší aplikace

Vývojáři vyzkoušeli všechno možné, aby vytvořili bohatší, interaktivnější webové aplikace – od ovládacích prvků ActiveX až po Flash. HTML5 nabízí neuvěřitelné možnosti, které v některých případech zcela vyloučí potřebu použití cizích technologií.

## Posílání zpráv mezi dokumenty

Webové prohlížeče nám brání používat skripty na jedné doméně ovlivňující nebo reagující na skripty v rámci jiné domény. Díky tomuto zákazu jsou koncoví uživatelé chráněni před skriptováním mezi weby, které se používalo k provedení různých nechutností nic netušícímu návštěvníkovi webu.

Nicméně tím se *všem* skriptům brání jejich fungování, a to i když si je píšeme sami a víme, že obsahu můžeme věřit. HTML5 obsahuje řešení, které je bezpečné a snadno se implementuje. Jak to funguje, uvidíte v kapitole 10, tipu 24 *Komunikace mezi doménami*.

## Webové sockety

HTML5 nabízí podporu webových socketů, díky nimž získáte trvalé připojení k serveru. Namísto neustálého otravování back-endu kvůli aktualizacím se váš web může přihlásit k socketu a back-end může sdělení posílat vašemu uživateli. S tím si trochu pohrajeme v kapitole 10, tipu 25 *Komunikace pomocí webových socketů*.

## Úložiště na straně klienta

Máme tendenci uvažovat o HTML5 jako o webové technologii, ale po přidání API pro webové úložiště a webové SQL databáze je teprve možné vytvářet aplikace v prohlížeči, které dokážou uchovat data zcela v zařízení na straně klienta. Jak tyto API používat, se dozvíte v kapitole 9, *Pracujeme s daty na straně klienta*.

## Lepší uživatelské rozhraní

Uživatelské rozhraní je důležitá část webové aplikace a každý den se snažíme prohlížeč přemluvit, aby udělal to, co chceme. Ke stylování tabulek nebo kulatých rohů buď použijeme knihovny JavaScriptu, či přidáme tony kódu navíc, jen abychom mohli přidělit správné styly. HTML5 a CSS3 posílá tyto způsoby do zapomnění.

## Lepší formuláře

HTML5 slibuje lepší ovládací prvky uživatelského rozhraní. Celou věčnost jsme museli používat JavaScript a CSS k vytvoření posuvných jezdců, kalendáře pro výběr data a kapátka pro výběr barvy. Všechny tyto prvky jsou nyní reálně definovanými elementy HTML5, podobně jako roletové nabídky, zatrhávací políčka a přepínače. Dozvíte se o nich v kapitole 3. Ačkoliv tuto záležitost ještě nelze použít v každém prohlížeči, je to něco, na co byste měli dávat pozor, zvláště pokud vyvíjíte webové aplikace. Navíc k vylepšení použitelnosti bez nutnosti spoléhat se na knihovny JavaScriptu zde existuje ještě jiná výhoda – zlepšená přístupnost. Čtečky obrazovky a jiné prohlížeče tyto ovládací prvky implementují určitými způsoby, takže je snadno budou ovládat i lidé s omezeními.

## Vylepšená přístupnost

Použitím nových elementů HTML5 k jasnému popsání našeho obsahu programům, jako jsou čtečky obrazovky, usnadňujeme zpracování obsahu. Například navigaci webu lze najít snadněji, pokud se poohlédnete po elementu `nav` namísto po konkrétním elementu `div` nebo neuspořádaném seznamu. Zápatí, postranní panely a jiný obsah lze snadno přeuspořádat nebo vynechat. Zpracování stránek se obecně stává mnohem méně bolestivým procesem, jenž může vést k lepší zkušenosti lidí závislých na asistenčních technologiích. Navíc nové atributy elementů dokáží specifikovat roli elementů, takže čtečky obrazovky s nimi mohou snadněji pracovat. V kapitole 5 se naučíte, jak použít nové atributy tak, aby je mohly využít současné čtečky obrazovky.

### Pokročilé selektory

CSS3 obsahuje selektory, jež vám umožňují identifikovat liché a sudé řádky tabulky, všechna vybraná zaškrťovací políčka, nebo dokonce poslední odstavec ve skupině. Více toho můžete dosáhnout s menším množstvím zdrojového kódu. Díky tomu je také snadnější stylovat kód HTML, který nemůžete upravovat. V kapitole 4 uvidíte, jak tyto selektory efektivně používat.

### Vizuální efekty

Stíny za obrázky a texty pomáhají dostat na webovou stránku hloubku a přechody také mohou přidat na efektu prostoru. CSS3 vám umožňuje přidávat stíny a přechody elementům bez nutnosti spoléhat se na obrázky na pozadí nebo zdrojový kód navíc. Navíc můžete použít transformace k vytvoření kulatých rohů nebo ke zkosení či otočení elementů. Jak tyto věci fungují, uvidíte v kapitole 8.

### Zpětná kompatibilita

Jedním z nejlepších důvodů, proč přejít již dnes k HTML5, je fakt, že to funguje ve všech prohlížečích. Ihned, dokonce i v Internet Exploreru 6, můžete začít používat HTML5 a pomalu přetvářet váš zdrojový kód. Dokonce vše projde i validací ve validátoru W3C (samozřejmě podmíněčně, neboť standardy se stále ještě vyvíjejí).

Pokud jste již pracovali s HTML nebo XML, pak jste přišli do styku s deklarací typu dokumentu. Používá se k tomu, aby sdělila validátorům a editorům, jaký typ značek a atributů můžete použít a jaké by měl dokument mít náležitosti. Mnoho webových prohlížečů deklaraci také používají k určení toho, jak prohlížeč stránku vykreslí. Validní DOCTYPE často způsobí, že prohlížeč stránku vykreslí ve „standardním režimu“.

V porovnání s rozvláchnou deklarací *XHTML 1.0 Transitional* používanou na mnoha webových stránkách:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd" >
```

...vypadá deklarace typu dokumentu v HTML5 až směšně jednoduše:

```
<!DOCTYPE html>
```

Tuto deklaraci umístíte do dokumentu a používáte HTML5.

Samozřejmě nevidíte žádné nové elementy HTML5, které váš cílový prohlížeč nepodporuje, ale váš dokument se bude validovat podle standardu HTML5.

## Budoucí cesta je trnitá

Na cestě za širším přijetím HTML5 a CSS3 existuje ještě řada překážek. Některé jsou zřejmé více a některé méně.

### Otázky a odpovědi

#### **Ale já mám rád samouzavírací značky XHTML. Můžu je i nadále používat?**

Samozřejmě, že ano! Mnoho vývojářů se do XHTML zamilovalo kvůli striktnějším požadavkům na zdrojový kód. Dokumenty XHTML si vynucují umístění hodnot atributů do uvozovek, uzavírání nepárových značek, používání malých písmen pro názvy atributů a dobře tvořený zdrojový kód. Přesun k HTML5 neznámá, že musíte hned měnit vaše způsoby. Dokumenty HTML5 budou validní, ať už použijete styl syntaxe HTML5 nebo XHTML, ale musíte rozumět důsledkům používání samouzavíracích značek.

Většina webových serverů dodává HTML stránky v MIME typu text/html kvůli neschopnosti Internet Exploreru správně zacházet s MIME typem application/xml+xhtml spojeném s XHTML stránkami. Kvůli tomu mají prohlížeče tendenci samouzavírací značky svléknout, protože nebyly v HTML považovány před HTML5 za validní. Například pokud jsme měli samouzavírací značku `script` v elementu `div`:

```
<script language="javascript" src="application.js" />
<h2>Pomoc</h2>
```

Prohlížeč by odstranil lomítko samouzavírací značce a při vykreslení by si myslel, že značka `h2` je umístěná ve značce `script`, která není uzavřená! A to je důvod, proč vidíte tyto značky zapsané vždy v páru s uzavírací značkou, i když je samouzavírací značka v kódu dle standardu XHTML povolená.

Takže si dejte pozor na možné problémy jako je tento, pokud používáte samouzavírací značky v dokumentech HTML5, protože ty se dodávají s MIME typem text/html. Více se o tomto problému dozvíte na adrese <http://www.webdevout.net/articles/beware-of-xhtml#myths>.



### Dort a poleva

Mám rád dort. A ještě raději mám koláč, ale dort je docela dobré zboží. Dávám přednost dortu s polevou.

Jestliže vyvíjíte webové aplikace, musíte myslet na to, že všechna ta hezká uživatelská rozhraní a fantastický JavaScript je poleva na dortu. Váš web může být skutečně dobrý i bez polevy, a jako u koláče platí, že potřebujete základ, na který dáte polevu.

Potkal jsem se s lidmi, kteří nemají rádi polevy. Z dortu je seškrábnou. A také jsem potkal lidi používající webové aplikace bez JavaScriptu, a to z různých důvodů.

Upečte těmto lidem skutečně skvělý dort. A pak na něj přidejte polevu.

## Internet Explorer

Internet Explorer má v současnosti největší základnu uživatelů a verze 8 a nižší podporují HTML5 a CSS3 velmi slabě. Verze 9 tuto situaci zlepšuje, ale ještě není široce používána. To ale neznamená, že na svém nemůžete HTML5 a CSS3 použít. Náš web zprovozníme i v Internet Exploreru, ale nebude v něm fungovat stejně jako v případě verze vyvíjené pro Chrome nebo Firefox. Poskytneme tedy pouze nouzové řešení, abychom nenaštvali uživatele a neztratili zákazníky.

## Přístupnost

Naši uživatelé musí být schopni s webovou stránkou interagovat, i když mají hendikep, starší prohlížeč, pomalé připojení k Internetu nebo mobilní zařízení. HTML5 představuje některé nové elementy, jako jsou audio, video a canvas. V případě zvuku a obrazu byly vždy potíže s přístupností, ale element canvas představuje novou výzvu. Element canvas nám umožňuje vytvářet vektorové obrázky přímo v dokumentu HTML za pomoci JavaScriptu. A to přináší problémy pro uživatele s omezeními a způsobuje problémy pěti procentům webových uživatelů, kteří mají vypnutou podporu JavaScriptu.<sup>3</sup>

Pokud jdeme cestou nových technologií, musíme dbát na přístupnost a poskytnout odpovídající za tyto elementy HTML5 nouzové řešení, stejně tak jak to uděláme pro uživatele Internet Exploreru.

<sup>3</sup> <http://visualrevenue.com/blog/2007/08/eu-and-us-javascript-disabled-index.html>

## Zavržené značky

HTML5 představuje mnoho nových elementů, ale specifikace také zavrhuje docela běžné elementy, které se možná nacházejí i na vašich webových stránkách.<sup>4</sup> Těch se tedy budete chtít zbavit.

Nejprve bylo zavrženo několik prezentačních elementů. Pokud některý z nich najdete ve svém kódu, zbavte se jej! Nahraďte jej sémanticky správnými elementy a použijte CSS k jejich zkrášení. Jde o:

- `basefont`
- `big`
- `center`
- `font`
- `s`
- `strike`
- `tt`
- `u`

Některé z těchto značek jsou pěkně zastaralé, ale stále najdete hodně stránek, jež se spravují pomocí editorů, jako je Dreamweaver, a které stále obsahují hodně značek `font` a `center`.

Mimo prezentačních elementů byla odstraněna také podpora rámců. Rámce byly velmi populární u podnikových webových aplikací, jako jsou PeopleSoft, Microsoft Outlook Web Access a na míru vytvořených portálů. Navzdory širokému použití rámce způsobovaly tolik problémů v použitelnosti a přístupnosti, že prostě musely zmizet ze scény. To znamená, že tyto elementy jsou pryč:

- `frame`
- `frameset`
- `noframes`

Měli byste se poohlédnout po jiných způsobech rozvržení webové stránky bez rámců za použití běžného CSS nebo nějakého JavaScriptu. Pokud používáte rámce, abyste zajistili stejné zobrazení záhlaví a navigace na každé stránce vaší aplikace, budete schopni totéž provést s nástroji, které poskytuje váš Framework pro vývoj webů. Pryč je několik dalších elementů, protože existují lepší volby:

- `acronym` nahradil `abbr`
- `applet` nahradil `object`
- `dir` nahradil `ul`

<sup>4</sup> <http://www.w3.org/TR/html5-diff/>

Mimo zavržených elementů je zde spousta atributů, jež nadále neplatí. Mezi ně patří například následující prezentační atributy:

- `align`
- `link`, `vlink`, `alink` a textové atributy ve značce `body`
- `bgcolor`
- `height` a `width`
- `scrolling` ve značce `iframe`
- `valign`
- `hspace` a `vspace`
- `cellpadding`, `cellspacing` a `border` u značky `table`

Pokud používáte atribut `target` u svých odkazů, například takto:

```
<a href="http://www.google.com" target="_blank">
```

... pak se namísto toho budete muset poohlédnout po JavaScriptu, protože atribut `target` je zavržený.

Atribut `profile` ve značce `head` již také není podporovaný, a to je něco, co uvidíte v kdejaké šabloně WordPressu.

A nakonec zmizel atribut `longdesc` ve značce `img` a `iframe`, což je pro ochránce přístupnosti trochu zklamání, neboť `longdesc` byl akceptovaný způsob poskytování další popisné informace uživatelům se čtečkami obrazovky.

Pokud plánujete používat HTML5 na svém současném webu, pak se budete muset po těchto elementech podívat, odstranit je a nahradit je těmi sémanticky korektními. Ujistěte se, že vaši stránku validujete v oficiálním validátoru W3C,<sup>5</sup> protože vám pomůže najít zavržené značky a atributy.

## Soutěžení firemních zájmů

Internet Explorer není jediný prohlížeč zpomalující přijetí HTML5 a CSS3. Google, Apple a Mozilla Foundation mají také své zásluhy, zatímco bojují o nadvládu. Hádají se o podporu zvukových a obrazových kodeků a svůj názor na věc promítají do implementace v prohlížeči. Například Safari bude pomocí elementu `audio` přehrávat hudbu v MP3, ale formát ve formátu OGG už nikoliv. Firefox zase podporuje formát OGG, ale MP3 ne.

Nakonec se tyto rozdíly vyřeší. A mezitím můžeme udělat chytrou volbu ohledně toho, co budeme podporovat, na základě toho, jaký prohlížeč používá naše

<sup>5</sup> <http://validator.w3.org/>

cílové publikum. Nebo můžeme stejnou věc implementovat vícekrát pro každý prohlížeč zvlášť, dokud nebude standard dokončený. Není to tak hrozné, jak to zní. Této problematice se budeme věnovat v kapitole 7.

## HTML5 a CSS3 jsou stále ve vývoji

Stále nejde o finální specifikaci, a to znamená, že se může ještě cokoliv změnit. Zatímco Firefox, Chrome a Safari podporují HTML5 výrazně. A pokud se specifikace změní, změní ji i prohlížeče. A to by mělo vést k narušení některých webů. Během psaní této knihy byla podpora stínování bloků v CSS ze specifikace odebrána a znovu přidána a protokol webových socketů byl pozměněn, čímž zcela zničil komunikaci mezi serverem a klientem.

Pokud budete sledovat postupující práce na HTML5 a CSS3, pak budete v pohodě. Slušná část toho, co probereme v této knize, bude fungovat dlouho.

Pokud se setkáte s něčím, co nefunguje v některém z cílových prohlížečů, prostě vyplňte mezery za pochodu využitím JavaScriptu a Flashe dle vaší libosti. Vytvoříte solidní řešení, které bude fungovat všem uživatelům, a jak bude čas plynout, budete schopni JavaScript a další nouzová řešení odstranit, aniž byste měnili vaši implementaci.

Ale ještě než začnete přemýšlet o vzdálené budoucnosti, začněte pracovat s HTML5. Je zde řada nových strukturálních značek, s nimiž se seznámíte v příští kapitole. Takže není na co čekat, ne?



ČÁST I

# **Zlepšování uživatelského rozhraní**



# Nové strukturální značky a atributy

V několika prvních kapitolách této knihy se budeme bavit o tom, jak můžeme využít možností HTML5 a CSS3 ke zlepšení rozhraní prezentovaných uživatelům. Uvidíme, jak se vytvářejí lepší formuláře, jak se snadno stylují tabulky a jak zlepšit přístupnost našich stránek pro asistenční zařízení. Také uvidíme, jak lze využít generování obsahu k vylepšení použitelnosti našich tiskových stylů, a prozkoumáme způsob místních úprav s novým atributem `contented` `table`. Nejprve se ale podívejme na to, jak nám nové elementy HTML5 mohou pomoci lépe strukturovat stránky.

Rád bych si s vámi promluvil o závažném problému ovlivňujícím mnoho webových vývojářů. *Předivování* (anglicky *Divitis*) – je to chronický syndrom způsobující, že weboví designéři obalují elementy do extra značek `div` s identifikátory jako `banner`, `panel`, `clanek` nebo `zapat`. Tento syndrom je velmi rozbujelý a také vysoce nakažlivý. Vývojáři si *předivování* předávají extrémně rychle, a protože jsou elementy `div` oku neviditelné, slabé případy *předivování* mohou povšimnutí unikát roky.



Zde je případ symptomu *předivování*:

```
<div id="navigace_obal" >
  <div id="navigace" >
    <ul>
      <li><a href="/" >Úvod</a></li>
      <li><a href="/" >Služby</a></li>
    </ul>
  </div>
</div>
```

Zde máme neuspořádaný seznam, který je sám o sobě blokovým elementem,<sup>6</sup> zabaleným do dvou značek `div`, které jsou také blokovými elementy. Atributy `id` obalových elementů nám říkají, co onen element dělá, ale minimálně jeden z obalových elementů můžete odstranit a výsledek bude stejný. Nadužívání zdrojového kódu vede k jeho nafouknutí a stránkám, jež lze složitě stylovat a udržovat.

Ale naštěstí je zde naděje. Specifikace HTML5 poskytuje léčbu ve formě nových sémantických značek popisujících obsah, jež označují. Protože tolik vývojářů ve svých designech vytvořilo postranní panely, záhlaví, zápatí a části stránky, specifikace HTML5 uvádí nové značky specificky navržených k rozdělení stránky do logických celků. Vrhněme se tedy do práce s těmito elementy. Společně s HTML5 můžeme pomoci konečně vymýtit *předivování*.

Mimo těchto strukturálních značek si povíme o elementu `meter` a probereme, jak můžeme využít možnosti nových vlastních atributů v HTML5, takže se můžeme věnovat vkládání obsahu do elementů místo toho, abychom naháněli třídy nebo existující atributy. Shrnuto v kostce, zjistíme, jak použít ty pravé značky vykonávající tu správnou práci.

V této kapitole prozkoumáme tyto nové elementy a možnosti:<sup>7</sup>

```
<header>
```

Definuje oblast záhlaví stránky nebo její části. [Ch5, FF3.6, IE8, S4, O10]

```
<footer>
```

Definuje oblast zápatí stránky nebo její části. [Ch5, FF3.6, IE8, S4, O10]

<sup>6</sup> Pamatujte si, že blokové elementy fungují minimálně na celém jednom řádku, zatímco řádkové elementy nevedou k zalomení.

<sup>7</sup> V následujících popisech se podpora prohlížečů zobrazuje v hranatých závorkách pomocí zkratky a nejnižší podporující verze. Použité kódy jsou: Ch pro Google Chrome, FF pro Firefox, IE pro Internet Explorer, O pro Operu, S pro Safari a IOS pro zařízení iOS s mobilním Safari, A pro Android.

<nav>

Definuje oblast s navigací stránky nebo její části. [Ch5, FF3.6, IE8, S4, O10]

<section>

Definuje logickou oblast stránky nebo její části. [Ch5, FF3.6, IE8, S4, O10]

<article>

Definuje článek nebo kompletní část obsahu. [Ch5, FF3.6, IE8, S4, O10]

<aside>

Definuje sekundární nebo příbuzný obsah. [Ch5, FF3.6, IE8, S4, O10]

*Vlastní datové atributy*

Umožňují přidání vlastních atributů jakýmkoli elementům pomocí vzoru `data-`. [Všechny prohlížeče podporující jejich čtení přes javascriptovou metodu `getAttribute()`.]

<meter>

Popisuje míru v rámci rozsahu. [Ch5, FF3.5, S4, O10]

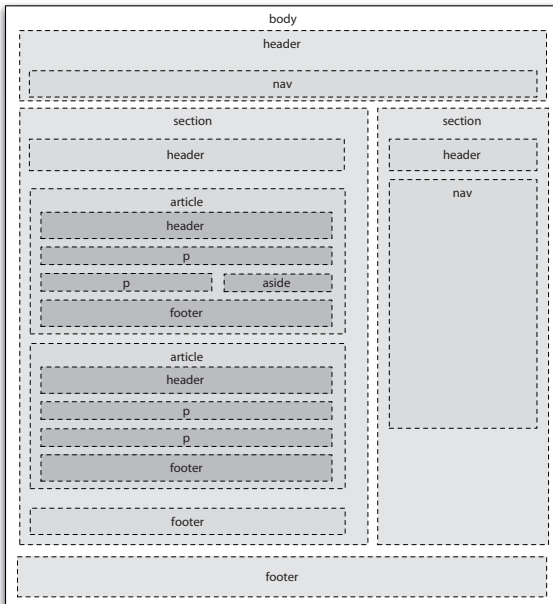
<progress>

Ovládá zobrazení reálného časového průběhu vzhledem k cíli. [Ch5, FF3.5, S4, O10]

## 1 Předefinování blogu pomocí sémantického zdrojového kódu

Jedním z míst, kde dozajisté najdete hodně obsahu s potřebou strukturovaného zdrojového kódu, je blog. Máte zde záhlaví, zápatí, více typů navigace (archiv, anketu a interní odkazy) a samozřejmě jednotlivé příspěvky ve formě článků. Použijme tedy zdrojový kód HTML5 k modelování titulní stránky blogu *AwesomeCo*, společnosti s něčím opravdu převratným.

Abyste si udělali představu, do čeho se to pouštíte, podívejte se na obrázek 2.1. Jde o běžnou strukturu blogu s hlavním záhlavím a níže s vodorovnou navigační nabídkou. V hlavní části má každý článek své záhlaví a zápatí. Článek může také obsahovat anketu nebo boční panel. Celá stránka obsahuje postranní panel a další navigační prvky. A nakonec má celá stránka zápatí s kontaktními informacemi a copyrightem. Na této struktuře není nic nového až na to, že tentokrát namísto kódování pomocí množství značek `div` použijeme specifické značky popisující tyto oblasti.



**Obrázek 2.1:** Struktura blogu pomocí sémantického zdrojového kódu HTML5

Až budeme hotovi, budeme mít něco podobného jako na obrázku 2.2.



Obrázek 2.2: Dokončený layout

## Vše začíná se správnou deklarací typu dokumentu

Protože budeme chtít použít nové elementy HTML5, budeme potřebovat dát prohlížečům a validátorům vědět, že tyto značky použijeme. Vytvořte novou stránku s názvem `index.html` a umístěte do něj tuto základní šablonu HTML5:

```

řádek 1  <!DOCTYPE html>
-        <html lang="cs-CZ">
-        <head>
-            <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
5         <title>Blog společnost AwesomeCo</title>
-        </head>
-
-        <body>
-        </body>
10       </html>
-    
```

Podívejte se na DOCTYPE ve výpisu příkladu na řádku 1. To je vše, co deklarace typu dokumentu v případě HTML5 vyžaduje. Pokud jste již vytvářeli webové stránky, pravděpodobně budete znát dlouhou a těžko zapamatovatelnou deklaraci typu dokumentu v případě XHTML, například:

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd" >
    
```

A nyní se podívejte na deklaraci typu dokumentu v případě HTML5:

```
<!DOCTYPE HTML>
```

Tato deklarace je jednodušší a snadno zapamatovatelná.

Cíl deklarace typu dokumentu je dvojitý. Jednak pomáhá validátorům určit, jaký typ validace je třeba při ověřování kódu zvolit. A za druhé deklarace typu dokumentu prohlížečům Internet Explorer 6, 7 a 8 sděluje, aby se přepnuly do „standardního režimu“. Ten je zcela nezbytný, pokud se snažíte tvořit stránky fungující stejně ve všech prohlížečích. Deklarace typu dokumentu v HTML5 uspokojuje obě tyto potřeby a *dokonce ji rozezná i Internet Explorer 6*.

## Záhlaví

Záhlaví může obsahovat veškeré možné typy obsahu, od loga společnosti po vyhledávací pole. Záhlaví našeho blogu bude nyní obsahovat pouze název blogu.

```
řádek 1 <header id="stranka_zahlaví">  
2 <h1>Blog AwesomeCo</h1>  
3 </header>
```

Na stránce nejste omezeni na jediné záhlaví. Každá jednotlivá sekce stránky nebo článek může obsahovat své záhlaví, a proto není od věci používat atribut `id`, jako jsem to udělal já, abyste snadno jednotlivé elementy odlišili. Jediný identifikátor usnadňuje stylování elementů pomocí CSS nebo nalezení elementu pomocí JavaScriptu.

### Sémantický zdrojový kód

Sémantický zdrojový kód je jen o popisování obsahu. Pokud už máte za sebou pár let vývoje webových stránek, pravděpodobně jste své stránky rozdělili do různých oblastí jako je záhlaví, zápatí či postranní panel, abyste snadněji rozeznali oblasti stránky při přidělování stylů a jiného formátování.

Sémantický zdrojový kód usnadňuje strojům a lidem pochopit smysl a kontext obsahu. Nový zdrojový kód HTML5 se značkami jako `section`, `header` a `nav` tomu napomáhá.

## Zápatí

Element `footer` definuje zápatí dokumentu nebo dané části. Na webových stránkách jste již viděli zápatí. Obvykle obsahují informaci o autorských právech a informace o majiteli webu. Specifikace praví, že v dokumentu můžeme mít vícero zápatí, takže to znamená, že bychom mohli zápatí použít také v článcích našeho blogu.

Nyní definujme jen zápatí celé stránky. A protože můžeme mít více než jedno zápatí, přidělíme mu identifikátor, jako jsme to udělali v případě záhlaví. Toto jedinečné označení zápatí nám pomůže, když budete chtít tomuto elementu a jeho potomkům přidělit styly.

```
<footer id="stranka_zapati">
  <p>&copy; 2011 AwesomeCo</p>
</footer>
```

Toto zápatí jednoduše obsahuje datum copyrightu. Nicméně, podobně jako v případě záhlaví, i zápatí na stránkách často obsahují jiné elementy včetně těch navigačních.

## Navigace

Navigace je pro úspěch webu zcela nezbytná. Lidé jednoduše nezůstanou, pokud je pro ně příliš těžké najít to, co hledají, takže dává smysl věnovat navigační nabídce její vlastní značku.

Do záhlaví našeho dokumentu přidejme část s navigací. Vložíme sem odkazy na domovskou stránku blogu, archiv, stránku se seznamem přispěvatelů blogu a odkaz na kontaktní informace.

```
řádek 1  <header id="stranka_zahlavi">
-         <h1>Blog AwesomeCo</h1>
-         <nav>
-           <ul>
5             <li><a href="/">Poslední příspěvky</a></li>
-             <li><a href="archiv">Archiv</a></li>
-             <li><a href="prispevatele">Přispěvatelé</a></li>
-             <li><a href="kontakt">Kontaktujte nás</a></li>
-           </ul>
10        </nav>
-      </header>
```

Jako v případě záhlaví a zápatí, může vaše stránka i v tomto případě obsahovat více navigačních nabídek. Navigaci často najdete v záhlaví a v zápatí, takže nyní můžete obě navigace označit odlišně. Zápatí našeho blogu vyžaduje odkazy na domovskou stránku webu společnosti, na stránku o společnosti a odkazy na obchodní podmínky a úmluvu o dodržování soukromí. Takže tyto odkazy přidáme do elementu footer, tedy zápatí stránky.

```
<footer id="stranka_zapatí">
  <p>&copy; 2011 AwesomeCo</p>
  <nav>
    <ul>
      <li><a href="http://awesomeco.cz/">AwesomeCo</a></li>
      <li><a href="onas">O nás</a></li>
      <li><a href="podminky.html">Obchodní podmínky</a></li>
      <li><a href="soukromi.html">Soukromí</a></li>
    </ul>
  </nav>
</footer>
```

Vzhled obou těchto panelů změníme později pomocí CSS, takže se nyní vzhledem nemusíte zabývat. Cílem těchto nových elementů je popsat obsah, nikoliv vysvětlovat, jak obsah vypadá.

## Sekce

Sekce jsou logickými oblastmi stránky a element `section` zde slouží jako náhrada značek `div` zneužívaných k popisování logických částí stránky.

```
<section id="prispevky">
</section>
```

Nenechte se ale sekcemi příliš unést. Používejte je k logickému seskupování obsahu! Zde jsme vytvořili sekci, do níž umístíme příspěvky blogu. Nicméně každý příspěvek by se neměl nacházet ve své vlastní sekci. K tomu je určená mnohem odpovědnější značka.

## Články

Značka `article` je skvělým elementem popisujícím vlastní obsah webové stránky. S tolika elementy na stránce, včetně záhlaví, zápatí, navigačních elementů, inzerce, widgetů, anket a záložek sociálních sítí, je možné snadno zapomenout, že lidé na váš web přišli kvůli zájmu o vámi poskytovaný obsah. Značka `article` vám pomáhá takový obsah popsat.

```
<article class="prispevek">
  <header>
    <h2>Kolik bychom na vás měli vsadit?</h2>
    <p>Publikoval Tomáš,
      <time datetimes="2011-02-24T14:39:01:00">
        24. února 2011 v 14:39</time>
      </p>
  </header>
  <p>
    První velké pravidlo v obchodu zní: pokud člověka necháte s prázdnými rukama, nejspíš se již nikdy nevrátí. To je také důvod, proč musíte být poněkud agresivnější, pokud pracujete se zákazníky, ale musíte se ujistit, že to nepřeháníte a nevystrašíte je.
  </p>
  <p>
    Jedním ze způsobů, jak udržet konverzaci, je vyhnout se otázce s odpovědí ano/ne. Například pokud prodáváte plán služeb, nikdy se neptejte: &quot;Líbí se vám náš tří nebo pětiletý plán?&quot; Namísto toho se zeptejte: &quot;Zajímáte se o náš tří nebo pětiletý plán, jenž má větší hodnotu?&quot; Na první pohled se zdá, že se ptáte na tu samou věc, ale i když zákazník může konverzaci dobrovolně ukončit, v případě druhé otázky je to pro něj těžší, protože nemůže prostě odpovědět: &quot;Ne.&quot;
  </p>
  <footer>
    <p><a href="komentare"><i>25 komentářů</i></a> ...</p>
  </footer>
</article>
```

## Otázky a odpovědi

### Jaký je rozdíl mezi články a sekcemi?

O sekci přemýšlejte jako o části dokumentu. O článku přemýšlejte jako o vlastním obsahu, například časopisovém článku, příspěvku na blogu nebo položce aktualit.

Tyto nové značky popisují obsah, který obsahují. Sekce může mít mnoho článků a články mohou mít také mnoho sekcí. Sekce je jako sportovní sekce novin. Sportovní sekce obsahuje řadu článků a každý z těchto článků se může opět dělit na své vlastní sekce. Některé sekce, jako třeba záhlaví a zápatí mají své vlastní značky. Sekce je obecnější element, který můžete použít k logickému seskupení jiných elementů.

Sémantický zdrojový kód je o podřízení se významu vašeho obsahu.



Elementy header a footer můžeme použít uvnitř článků, díky čemuž bude snazší tyto specifické sekce popsat. Náš článek také můžeme rozdělit na více sekcí pomocí elementu `section`.

### Poznámky po straně

Někdy máte obsah navíc doplňující váš hlavní obsah, jako je třeba citace, digram, vedlejší poznámky nebo příbuzné odkazy. Umístit je v takovém případě můžete jako *poznámku stranou* pomocí značky `aside`, abyste takový element identifikovali.

```
<aside>
  <p>
    &quot;Nikdy nedejte nikomu šanci odpovědět ne, když se jej ptáte
    na nějaký výrobek.&quot;
  </p>
</aside>
```

Tuto citaci jsme umístili stranou hlavního obsahu pomocí elementu `aside`. Tento element `aside` vnoříme do článku – elementu `article`, díky čemuž bude blízko hlavnímu obsahu.

Celá sekce včetně citace umístěné stranou vypadá takto:

```
<section id="prispevky">
  <article class="prispevek">
    <header>
      <h2>Kolik bychom na vás měli vsadit?</h2>
      <p>Publikoval Tomáš,
        <time datetime="2011-02-24T14:39+01:00">
          24. února 2011 v 14:39</time>
        </p>
    </header>
    <aside>
      <p>
        &quot;Nikdy nedejte nikomu šanci odpovědět ne, když se jej
        ptáte na nějaký výrobek.&quot;
      </p>
    </aside>
    <p>
      První velké pravidlo v obchodu zní: pokud člověka necháte s
      prázdnýma rukama, nejspíš se již nikdy nevrátí. To je také důvod,
      proč musíte být poněkud agresivnější, pokud pracujete se
```

zákazníky, ale musíte se ujistit, že to nepřezene a nevystrašíte je.

```
</p>
```

```
<p>
```

Jedním ze způsobů, jak udržet konverzaci, je vyhnout se otázce s odpovědí ano/ne. Například pokud prodáváte plán služeb, nikdy se neptejte: „Líbí se vám náš tří nebo pětiletý plán?“. Namísto toho se zeptejte: „Zajímáte se o náš tří nebo pětiletý plán, jenž má větší hodnotu?“. Na první pohled se zdá, že se ptáte na tu samou věc, ale i když zákazník může konverzaci dobrovolně ukončit, v případě druhé otázky je to pro něj těžší, protože nemůže prostě odpovědět: „Ne.“

```
</p>
```

```
<footer>
```

```
<p><a href="komentare"><i>25 komentářů</i></a> ...</p>
```

```
</footer>
```

```
</article>
```

```
</section>
```

Nyní již stačí přidat jen část s postranním panelem.

## Postranní panel

Náš blog obsahuje po pravé straně postranní panel obsahující odkazy na archiv blogu. Pokud přemýšlíte nad použitím elementu `aside` k definování postranního panelu našeho blogu, pak přemýšlíte špatně. *Mohli* byste to takto provést, ale jde to proti smyslu specifikace. Element `aside` je navržený k zobrazení postranní poznámky, tedy obsahu spjatého s článkem. Je to dobré místo pro příbuzné odkazy, glosář nebo citaci.

K označení postranního panelu obsahujícího náš seznam odkazů do archivu prostě použijeme další značku `section` a také značku `nav`.

```
<section id="panel">
```

```
<nav>
```

```
<h3>Archivy</h3>
```

```
<ul>
```

```
<li><a href="2011/01">leden 2011</a></li>
```

```
<li><a href="2010/12">prosinec 2010</a></li>
```

```
<li><a href="2010/11">listopad 2010</a></li>
```

```
<li><a href="2010/10">říjen 2010</a></li>
```

```
<li><a href="2010/09">září 2010</a></li>
```

```
<li><a href="2010/08">srpen 2010</a></li>
<li><a href="2010/07">červenec 2010</a></li>
<li><a href="2010/06">červen 2010</a></li>
<li><a href="2010/05">květen 2010</a></li>
</ul>
</nav>
</section>
```

A to je ke struktuře webu vše. Nyní můžeme začít našim novým elementům přidělovat styly.

## Stylování

Styly těmto elementům přidělíme úplně stejně jako bychom přidělovali styly elementům div. Nejprve vytvoříme novou šablonu stylů `styly.css` a připojíme ji k našemu dokumentu HTML vložením odkazu do záhlaví dokumentu. Takto:

```
<link rel="stylesheet" href="styly.css" type="text/css">
```

Nejdříve se zaměříme na obsah stránky a nastavme několik základních stylů písma:

```
body{
  width:960px;
  margin:15px auto;
  font-family: Arial, "MS Trebuchet", sans-serif;
}
p{
  margin:0 0 20px 0;
}
p, li{
  line-height:20px;
}
```

Poté definujeme záhlaví stránky:

```
header#stranka_zahlaví{
  width:100%;
}
```

Navigační odkazy budeme stylovat tak, abychom přeměnili odrážkový seznam na vodorovný panel s navigační nabídkou.

```
header#stranka_zahlaví nav ul, #stranka_zapati nav ul{
  list-style: none;
  margin: 0;
```

```
padding: 0;
}
#stranka_zahlavı nav ul li, footer#stranka_zapati nav ul li{
padding:0;
margin: 0 20px 0 0;
display:inline;
}
```

Sekce příspěvky potřebujeme vložit do plovoucího umístění se zarovnáním doleva a dodat šířku, a stejně musíme umístit poznámku stranou s citací uvnitř článku. A poté trochu zvětšíme velikost písma poznámky stranou.

```
section#prispevky {
float: left;
width: 74%;
}
section#prispevky aside{
float: right;
width: 35%;
margin-left: 5%;
font-size: 20px;
line-height: 40px;
}
```

Postranní panel také umístíme v rámci plovoucího umístění a definujeme jeho šířku.

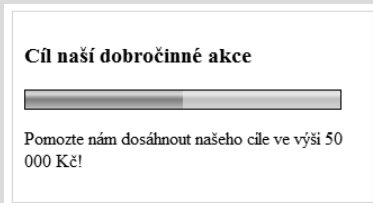
```
section#panel {
float: left;
width: 25%;
}
```

### Měřidlo a indikátor průběhu

Jestliže potřebujete implementovat zaručené měřidlo nebo indikátor průběhu ve své webové aplikaci, pak byste měli prozkoumat možnosti elementů `meter` a `progress` v HTML5.

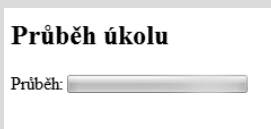
Element `meter` nám umožňuje sémanticky popsat aktuálně určený bod na měřidle s minimální a maximální hodnotou. Aby bylo vaše měřidlo v souladu se specifikací, neměli byste jej používat pro věci s náhodnými minimálními nebo maximálními hodnotami jako je váha nebo výška. Pokud ovšem nemluvíte o něčem konkrétním, u něhož jste nastavili určité hranice. Například pokud máme web dobročinné organizace a chceme ukázat, jak blízko jsme cílové částce 50 000 Kč, můžeme to provést následovně:

```
<section id="dar">
  <header>
    <h3>Cíl naší dobročinné akce </h3>
  </header>
  <meter title="CZK" id="dar_cil" value="25000" min="0" max="50000" >
    25 000Kč
  </meter>
  <p>Pomozte nám dosáhnout našeho cíle ve výši 50 000 Kč!</p>
</section>
```



Element `progress` je velmi podobný měřidlu, ale je navržený tak, aby zobrazoval aktivní průběh, který můžete vidět například při nahrávání souboru. V porovnání s ním je měřidlo navrženo tak, aby ukazovalo míru, která se nemění, jako snímek dostupného prostoru na serveru daného uživatele. Zdrojový kód indikátoru průběhu je velmi podobný elementu `meter`:

```
<progress id="prubeh" max=100><span>0</span> %</progress>
```



S elementy `meter` a `progress` si v době tvorby knihy poradily prohlížeče Chrome a Opera. U ostatních prohlížečů můžete k vytažení hodnot v měřidle použít JavaScript a vytvořit vaše vlastní ztvárnění, zatímco k sémantickému popisu dat použijete elementy `meter` a `progress`.

A nakonec musíme definovat zápatí stránky. Zrušíme veškeré obtékání elementů v plovoucím umístění, aby se zápatí umístilo na spodek stránky.

```
footer#stranka_zapati {
  clear: both;
  width: 100%;
  display: block;
  text-align: center;
}
```

To jsme si uvedli jen velmi základní styly. Jsem si jistý, že odtud to zvládnete sami a vzhled podstatně vylepšíte.

## Nouzové řešení

Ačkoliv tohle všechno funguje skvěle ve Firefoxu, Chrome a Safari, lidé z vedení firmy nebudou nadšeni, když uvidí ten nepořádek, jaký na stránce nastane při zobrazení v Internet Exploreru. Obsah se zobrazí v pořádku, ale jelikož Internet Explorer těmto elementům nerozumí, nemůže na ně použít styly a celá stránka se podobá něčemu z poloviny 90. let minulého století.

Jediným způsobem, jak Internet Explorer přinutit stylovat tyto elementy, je použití JavaScriptu k definování elementů jako součásti dokumentu. To je velmi jednoduché. Níže uvedený kód přidáme do části značky head, takže se spustí ještě před tím, než prohlížeč jakýkoli element vykreslí. Umístíme jej do podmíněného komentáře, což je speciální typ komentáře, který přečte pouze Internet Explorer.

```
<!--[if lt IE 9]>
<script type="text/javascript" >
    document.createElement("nav" );
    document.createElement("header" );
    document.createElement("footer" );
    document.createElement("section" );
    document.createElement("aside" );
    document.createElement("article" );
</script>
<![endif]-->
```

Tento jednotlivý komentář cílí na jakoukoli verzi Internet Exploreru starší než 9.0. Jakmile stránku necháme znovu načíst, vše je v pořádku.

Vytváříme tím závislost na JavaScriptu, takže to je samozřejmě na zvážení. Vylepšené uspořádání a čitelnost dokumentu za to může stát a vzhledem k tomu, že zde nejsou žádná zvažování ohledně přístupnosti, protože obsah se stále zobrazuje a čtečky obrazovky jej mohou přečíst, výsledek bude jediný: maximálně bude některým uživatelům se záměrně vypnutou podporou JavaScriptu prezentace připadat trochu zastaralá.

Tento přístup je hodný k přidávání podpory užitečných elementů, nebo kvůli pochopení toho, jak podporu můžete sami přidat. Skvělý web GTML-Shiv<sup>8</sup> Remyho Sharpa posouvá tento přístup mnohem dále a při přidávání nouzového řešení bude vhodnější, pokud hledáte podporu pro mnohem více elementů.

<sup>8</sup> <http://code.google.com/p/html5shiv/>

## 2 Tvorba automaticky otevíraných oken s vlastními datovými atributy

Pokud jste někdy vytvořili nějakou aplikaci používající JavaScript k dolování informací z dokumentu, pak víte, že občas je třeba k tomu, aby vše fungovalo, trocha *hackování* a *parsování*. Skončíte pak buď u vkládání dodatečné informace do obsluhy události, nebo u zneužití atributů `rel` a `class` k vložení chování. Tyto časy jsou už našťastí minulostí, a to díky uvedení vlastních datových atributů.

Všechny vlastní datové atributy začínají předponou `data-` a validátory dokumentů HTML5 je ignorují. Vlastní datový atribut můžete přidělit jakémukoli elementu chcete, ať už jde o metadata popisující fotografii, souřadnice zeměpisné šířky a délky, nebo, jak uvidíte v tomto tipu, půjde o rozměry automaticky otevíraného okna. Nejlepší ze všeho je, že vlastní datové atributy můžete použít okamžitě ve všech webových prohlížečích, neboť je z dokumentu dokáže snadno vydolovat JavaScript.

### Oddělení chování od obsahu, neboli proč je `onClick` špatná volba

Za uplynulé roky získala automaticky otevíraná okna špatnou pověst a často oprávněně. Často se totiž používala k umístění reklamy takovým způsobem, aby webové uživatele přesvědčila k instalaci spywaru, nebo hůře, aby získala osobní informace k pozdějšímu prodeji. Není žádným překvapením, že většina prohlížečů obsahuje určitý typ blokace automaticky otevíraných oken.

Automaticky otevíraná okna ale nejsou všechna špatná. Vývojáři webových aplikací na nich často spoléhají, pokud potřebují zobrazit on-line nápovědu, další volby, nebo jiné funkce uživatelského rozhraní. Aby byla automaticky otevíraná okna méně obtěžující, budeme je muset implementovat diskrétněji. Když se podíváte na stránku personálního oddělení společnosti AwesomeCo, uvidíte několik odkazů zobrazujících ve vyskakovacích oknech výhody. Většina z nich vypadá následovně:

```
<a href='#'  
  onClick="window.open('dovolená.html',navezOkna,'width=300,height=300);"  
>  
Dovolená  
</a>
```

To je celkem běžný způsob, jak tvořit odkazy vedoucí do automaticky otevíraných oken. Vlastně jde o způsob, jenž se nováčky v JavaScriptu učí při tvorbě automaticky otevíraných oken.

## Vylepšení přístupnosti

Cíl odkazu není zadán! Pokud je podpora JavaScriptu vypnutá, odkaz uživatele na stránku nevezme. A to je obrovský problém, s nímž se musíme ihned vypořádat. *Nikdy* nevynechejte atribut `href`, nebo mu přiřadíte hodnotu jako v tomto případě, a to za *jakýchkoli* okolností. Sdělte mu adresu zdroje, který by se za normálních okolností zobrazil v automaticky otevíraném okně.

```
<a href='dovolená.html'  
  onclick="window.open(this.href,nazevOkna,'width=300,height=300');">  
Dovolená  
</a>
```

Kód JavaScriptu poté přečte atribut `href` připojený k elementu, jenž sděluje cíl odkazu.

Prvním krokem při tvorbě přístupných stránek je zajistit, že všechna funkcionality funguje také bez JavaScriptu.

## Zrušení metody `onclick`

Chování ponechte oddělané od samotného obsahu, stejně jako udržujete oddělenou prezentační informaci používáním odkazu na šablonu kaskádových stylů. Používání metody `onclick` je na první pohled snadné, ale představte si stránku s padesáti odkazy a uvidíte, jak se vám použití metody `onclick` vymyká kontrole. A pokud tento kód generujete na straně serveru, pouze tím zvyšujete množství chování JavaScriptu a zvětšujete výslednou HTML stránku více, než je nutné.

Namísto toho přiřadíte každému odkazu na stránce třídu, která je bude identifikovat:

```
<a href="dovolená" class="popup">Dovolená</a>
```

A nyní kód JavaScriptu:

```
var links = $("a.popup" );  
  
links.click(function(udalost){  
  udalost.preventDefault();  
  window.open($(this).attr('href'));  
});
```



My používáme selektor jQuery k načtení elementu s třídou `popup` a poté přidáme posluchače každé události `click` v rámci elementu. Kód přidělený metodě `click` se spustí, když někdo na odkaz klepne. Metoda `preventDefault` zabraňuje provedení výchozího chování události `click`. V tomto případě zabraňuje prohlížeči, aby následoval odkaz a zobrazil novou stránku.

Jednou z věcí, kterou nemáme pod kontrolou je ale rozměr a umístění okna, což je něco, co jsme měli v původním příkladu. Chceme, aby designér stránky, který nemá takové povědomí o JavaScriptu byl stále schopen nastavit rozměry okna, a to u každého odkazu.

## Zachrání nás vlastní datové atributy!

Podobné situace jsou běžné kdykoliv tvoříte jakoukoli aplikaci využívající JavaScript. Jak jsme viděli, vyžadujeme vkládání požadované výšky a šířky okna do kódu, ale přístup s `onclick` má řadu nedostatků. Namísto toho můžeme tyto atributy vložit jako atributy elementu. Vše, co musíme udělat, je vytvořit odkaz podobným způsobem jako zde:

```
<a href="napoveda/dovolená.html"
  data-sirka="600"
  data-vyska="400"
  title="Dovolená"
  class="popup" >Dovolená</a>
```

A nyní stačí změnit napsanou událost `click` tak, aby vydolovala volby z vlastních datových atributů odkazu a předala je metodě `window.open`.

```
$(function(){
  $(".popup").click(function(udalost){
    udalost.preventDefault();
    var href = $(this).attr("href");
    var sirka = $(this).attr("data-sirka");
    var vyska = $(this).attr("data-vyska");
    var popup = window.open (href, "popup",
      "height=" + vyska + ",width=" + sirka + "");
  });
});
```

A to je vše! Odkaz se nyní otevírá do nového okna.

### Upozornění

V tomto příkladu jsme použili vlastní datové atributy k poskytnutí dodatečné informace skriptu na straně klienta. Je to chytrý přístup k určitému problému a ilustruje jeden ze způsobů, jak tyto atributy použít. Vede sice k míchání prezentační informace se zdrojovým kódem, ale je to jednoduchý způsob ukázání, jak jednoduše lze JavaScript použít ke čtení hodnot vložených do vaší stránky.

## Nouzové řešení

Tyto atributy fungují také ve starých prohlížečích, pokud prohlížeč podporuje JavaScript. Vlastní datové atributy prohlížeči nepodrazí nohy a váš dokument bude stále validní, neboť používáte deklaraci typu dokumentu HTML5 a všechny atributy začínající předponou `data-` se budou při validaci ignorovat.

## Budoucnost

Jakmile budou široce podporovány, bude možné s těmito novými značkami a atributy provádět zajímavé věci. Například je možné při použití tiskových stylů snadno rozeznat a vypnout navigaci.

```
nav, article>footer{display:none}
```

K rychlému určení všech článků na stránce nebo webu můžeme použít skriptovací jazyk. Ale co je nejdůležitější, označujeme obsah těmi správnými popisnými značkami, takže lze vytvářet lepší šablonu stylů a lepší skripty v JavaScriptu.


Vlastní datové atributy vývojářům poskytují flexibilitu k tomu, aby vkládali všechny druhy informace přímo ve zdrojovém kódu. Vlastně je opět použijeme v kapitole 6.

Použít je můžeme s JavaScriptem k určení, zda by se měl formulář zpracovat pomocí Ajaxu, jednoduše nalezením jakékoli značky `form` s atributem `data-remote=true`, což je něco, co se provádí ve frameworku Ruby on Rails.

Použít je také můžete k zobrazení data a času v časovém pásmu uživatele, zatímco se stránka stále načítá do vyrovnávací paměti. Jednoduše datum vložte na HTML stránku ve formátu koordinovaného světového času (UTC) a poté jej na straně klienta převedte do časového pásma uživatele. Tyto atributy vám umožňují vkládat reálná, použitelná data do vlastních stránek, a můžete očekávat, že uvidíte stále více frameworků a knihoven, jež je využívají. Jsem si jistý, že pro ně najdete hodně skvělých způsobů použití v rámci vaší vlastní práce.



# Tvorba uživatelsky přívětivých webových formulářů

Pokud jste již někdy navrhovali složité uživatelské rozhraní, pak víte, jak omezující mohou být výchozí ovládací prvky formuláře v HTML. Jste ponechání na pospas použití textových polí, výběrových nabídek, přepínačů, zaškrťovacích políček a někdy dokonce seznamů s možností více voleb, jejichž použití musíte stále dokola uživatelům vysvětlovat. (Podržte klávesu  a klepněte na hodnoty, které chcete vybrat. Pokud používáte Mac, pak použijte klávesu Cmd.)

Takže uděláte to, co každý dobrý vývojář – využijete Prototype nebo jQuery, nebo vymodelujete vlastní ovládací prvky a funkce za pomoci kombinace HTML, CSS a JavaScriptu. Ale když se podíváte na formulář, který obsahuje posouvací jezdce, kalendáře, počítadla, automaticky dokončovaná pole a vizuální editory, rychle si uvědomíte, že jste sami sobě stvořili noční múru. Musíte se ujistit, že ovládací prvky přidané na stránku nejsou v konfliktu s jakýmkoli jinými vloženými ovládacími prvky nebo jinými knihovnamy JavaScriptu, jejichž skripty na stránce používáte. Můžete strávit hodiny implementací výběru z kalendáře, abyste později zjistili, že knihovna Prototype má problém kvůli tomu, že knihovna jQue-

ry převzala kontrolu nad funkcí `$()`. Takže použijete metodu jQuery s názvem `noConflict()`, ale poté zjistíte, že použité kapátko pro výběr barvy již nefunguje, neboť daný zásuvný modul nebyl dostatečně pečlivě napsaný.

Pokud se usmíváte, je to z důvodu, že jste to nezažili. Jestliže se vztekáte, hádám, že z dobrého důvodu. Ale stále existuje naděje. V této kapitole budeme vytvářet dvojici formulářů za použití některých nových typů polí a také budeme implementovat automatické zaměřování a výplňový text.

Nakonec probereme, jak použít nový atribut `contenteditable` ke změně jakéhokoli pole HTML v uživatelem ovládaný prvek.

Zvláště se pak budeme věnovat následujícím možnostem:<sup>9</sup>

*E-mailové pole* [`<input type="email">`]

Zobrazuje formulářové pole určené pro e-mailovou adresu. [O10.1, IOS]

*Pole pro adresu URL* [`<input type="url">`]

Zobrazuje formulářové pole určené adrese URL. [O10.1, IOS]

*Telefonní pole* [`<input type="tel">`]

Zobrazuje formulářové pole určené telefonnímu číslu. [O10.1, IOS]

*Vyhledávací pole* [`<input type="search">`]

Zobrazuje formulářové pole pro vyhledávání pomocí frází. [Ch5, S4, O10.1, IOS]

*Posuvník (rozsahem)* [`<input type="range">`]

Zobrazuje ovládací prvek posuvník, někdy také označovaný jako jezdec. [Ch5, S4, O10.1]

*Číslo* [`<input type="number">`]

Zobrazuje formulářové pole určené číslu, často jde o počítadlo. [Ch5, S4, O10.1, IOS]

<sup>9</sup>V následujících popisech se podpora prohlížečů zobrazuje v hranatých závorkách pomocí zkratk a nejnižší podporující verze. Použité kódy jsou: Ch pro Google Chrome, FF pro Firefox, IE pro Internet Explorer, O pro Operu, S pro Safari a IOS pro zařízení iOS s mobilním Safari, A pro prohlížeč Androidu.

*Pole pro datum* [`<input type="date">`]

Zobrazuje formulářové pole určené pro datum. [Ch5, S4, O10.1]

*Datum s časem* [`<input type="datetime">`]

Zobrazuje formulářové pole určené pro datum s časem. Podporovány jsou hodnoty `datetime`, `datetime-local` nebo `time`. [Ch5, S4, O10.1]

*Barva* [`<input type="color">`]

Zobrazuje formulářové pole pro určení barvy. [Ch5, S5] (Chrome 5 a Safari 5 rozumějí poli pro barvu, ale nezobrazují jej jako specifický prvek.)

*Podpora pro automatické zaměření* [`<input type="text" autofocus>`]

Podpora pro umístění zaměření na specifický formulářový element. [Ch5, S4]

*Podpora pro výplň* [`<input type="email" placeholder="ja@priklad.cz">`]

Podpora zobrazení výplňového textu uvnitř formulářového pole. [Ch5, S4, F4]

*Podpora pro místní úpravy* [`<p contenteditable>lorem ipsum</p>`]

Podpora pro místní editaci obsahu přímo v prohlížeči. [Ch4, S3.2, IE6, O10.1]

Začněme nyní tím, že se naučíme něco o velmi užitečných typech formulářových polí.

Toto je pouze náhled elektronické knihy. Zakoupení její plné verze je možné v elektronickém obchodě společnosti eReading.