

Ondřej Baše

jQuery pro neprogramátory

Průvodce využitím knihovny jQuery UI

Tvorba formulářů a seznamů

Interaktivní práce
s objekty na stránce

Začlenění
multimediálního
obsahu do projektu

Přílohy s lokalizovanou
dokumentací



Ke stažení zdrojové
kódy příkladů z knihy

computer
press



Ondřej Baše

jQuery pro neprogramátory

Průvodce využitím knihovny jQuery UI

Computer Press
Brno
2012

jQuery pro neprogramátory

Průvodce využitím knihovny jQuery UI

Ondřej Baše

Obálka: Martin Sodomka

Odpovědný redaktor: Martin Herodek

Technický redaktor: Jiří Matoušek

Objednávky knih:

<http://knihy.cpress.cz>

www.albatrosmedia.cz

eshop@albatrosmedia.cz

bezplatná linka 800 555 513

ISBN 978-80-251-3750-5

Vydalo nakladatelství Computer Press v Brně roku 2012 ve společnosti Albatros Media a. s. se sídlem Na Pankráci 30, Praha 4. Číslo publikace 16 341.

© 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.

1. vydání


ALBATROS MEDIA a.s.

Obsah

ÚVOD	9
Co najdete v této knize	10
Komu je tato kniha určena	11
Co budeme potřebovat k vývoji	11
Webový prohlížeč.....	11
Vývojové prostředí.....	12
Webový server.....	14
Archiv se zdrojovými kódy příkladů.....	17
Konvence zápisu textu	18
Zpětná vazba od čtenářů	19
Errata.....	19
 KAPITOLA 1	
KNIHOVNA JQUERY UI	21
Co je knihovna jQuery UI	21
Dokumentace knihovny.....	22
Stažení knihovny	23
Výběr verze	25
Výběr komponent.....	25
Výběr některého ze standardních motivů vzhledu	26
Obsah balíku knihovny jQuery UI	27
Shrnutí.....	27
 KAPITOLA 2	
MOTIVY VZHLEDU	29
Pokročilá nastavení motivů vzhledu	29
Název složky s motivem vzhledu	30
Oblast platnosti kaskádových stylů	30
Příklad přepínání motivů vzhledu s pomocí pokročilých nastavení	31
Nástroj ThemeRoller	32
Galerie motivů vzhledu.....	33

Vlastní motiv vzhledu	34
Vývojářský nástroj pro prohlížeč Firefox	39
Zásuvný modul Theme Switcher	39
Vytvoření zásuvného modulu Theme Switcher	40
Možnosti nastavení	41
Shrnutí druhé varianty příkladu	42
Framework knihovny jQuery UI pro kaskádové styly	43
Šablony stylů motivu vzhledu	43
Obecné třídy frameworku knihovny jQuery UI pro kaskádové styly	44
Tvorba vlastního motivu vzhledu od začátku	46
Shrnutí	47

KAPITOLA 3

ZÁKLADY KNIHOVNY JQUERY 49

Co je knihovna jQuery	49
Stažení knihovny jQuery	50
Oficiální dokumentace	51
Základní stavební kameny knihovny jQuery	51
Funkce jQuery()	55
Globální objekt jQuery	57
Objekty knihovny jQuery	58
Jak používat selektory knihovny jQuery	59
Shrnutí	60

KAPITOLA 4

NAVIGACE V OBSAHU STRÁNKY 61

Náš cíl	61
Základ webových stránek	62
Knihovna jQuery UI s motivem vzhledu Start	63
Dokumenty HTML	64
Vložení stylů a skriptů	68
Tvorba hlavní nabídky	70
Ovládací prvek Tabs	70
Rozdělení informací do podsekcí	80
Rozvržení úvodní strany	80
Ovládací prvek Accordion	80
Upozornění při vstupu do části pro registrované uživatele	85
Ovládací prvek Dialog	86
Doladění detailů	93
Zápatí stránek	93
Vtipy	93
Upozornění a logo	95
Shrnutí	96

KAPITOLA 5

FORMULÁŘE	97
Náš cíl	98
Základ příkladu	99
Úprava velikosti textové oblasti	102
Interakce Resizable	102
Datum ve formuláři	106
Ovládací prvek Datepicker	107
Výběr rezervovatelných dní pomocí technologie AJAX	114
Funkce automatického dokončování	119
Ovládací prvek Autocomplete	120
Vzdálený zdroj pro funkci automatického dokončování	126
Shrnutí	130

KAPITOLA 6

PŘESOUVÁNÍ OBJEKTŮ	131
Náš cíl	132
Základ příkladu	133
Soubor index.html	133
Soubor hlavni.js	135
Přesouvání produktů do košíku a z košíku	136
Interakce Draggable	136
Interakce Droppable	143
Úprava počtu kusů produktů v košíku	161
Ověření, zda uživatel zadává jako počet kusů číslo	161
Změna počtu kusů produktu v košíku	164
Animace a efekty	166
Prosté uplatnění efektů	166
Animovaná změna viditelnosti	167
Postupná změna barvy	172
Pozvolné přechody mezi třídami	172
Pokročilé možnosti průběhu animace	174
Shrnutí	174

KAPITOLA 7

SEZNAMY	175
Náš cíl	175
Základ příkladu	177
Soubor index.html	177
Soubor hlavni.js	179
Výběr skladeb do uživatelského seznamu skladeb	181
Možnost nastavení handle interakce Draggable	182
Možnost nastavení connectToSortable interakce Draggable	182
Možnost nastavení revert interakce Draggable	183

Řazení seznamu	183
Interakce Sortable	183
Výběr položek seznamu	191
Interakce Selectable	191
Označení skladeb jako oblíbených	199
Svázání události click s odkazy	199
Obsluhující funkce označVybraneSkladbyJakoOblibene()	199
Obsluhující funkce vyradVybraneSkladbyZOblibenyh()	201
Shrnutí.	201

KAPITOLA 8

MULTIMEDIÁLNÍ OBSAH 203

Náš cíl	204
Jazyk HTML5	205
Struktura dokumentu HTML5	205
Element audio	206
Element video	207
Základ příkladu	208
Soubor index.html	208
Soubor hlavni.js	210
Přízpůsobení motivu vzhledu	212
Změna velikosti přehrávače	213
Výběr a načtení prostředku k přehrávání	215
Údaje o prostředcích	215
Výběr prostředku klepnutím na jeho položku.	216
Načítání prostředku	219
Inicializace přehrávače.	220
Uvedení přehrávače do výchozího stavu	221
Zobrazení tlačítka pro přehrávání	222
Výběr výchozího prostředku	223
Zobrazování časových údajů	224
Zobrazování aktuálního času	225
Zobrazování celkového času	226
Reakce na načtení metadat prostředku	227
Ovládací tlačítka	229
Ovládací prvek Button	229
Indikátor průběhu načítání	237
Ovládací prvek Progressbar.	237
Posouvání přehrávaného prostředku a změna hlasitosti	241
Ovládací prvek Slider.	241
Ztlumení a zesílení přehrávače	249
Shrnutí.	252

KAPITOLA 9

FOTOGALERIE	253
Náš cíl	254
Základ příkladu	255
Soubor index.html	255
Uspořádání zdrojového kódu	256
Návrhový vzor MVC	257
Seskupování zdrojového kódu do objektů	257
Uspořádání kódu do objektů podle vzoru MVC	257
Otevření fotogalerie	259
Generování ovládacích prvků	260
Načítání fotografií	264
Určení třídy skupiny fotografií	264
Načtení údajů o skupině fotografií	265
Načítání originálních fotografií ze skupiny	269
Základní rozvržení	280
Pomocný nástroj Position	280
Zobrazení aktuální fotografie	284
Události aktuální fotografie	287
Rozmístění ovládacích prvků	291
Korekce rozvržení	295
Přizpůsobení stavu tlačítek	297
Doplnění obsluhy událostí	298
Svázání událostí s tlačítky	298
Odstranění událostí aktuální fotografie	299
Zavření fotogalerie	300
Zobrazování předchozí a následující fotografie	302
Příprava na animaci	302
Animace přesunu	305
Dokončení animace	310
Shrnutí	311

PŘÍLOHA A

OVLÁDACÍ PRVKY	313
Ovládací prvek Tabs	313
Ovládací prvek Accordion	317
Ovládací prvek Dialog	319
Ovládací prvek Datepicker	323
Ovládací prvek Autocomplete	330
Ovládací prvek Button	332
Ovládací prvek Progressbar	334
Ovládací prvek Slider	335

PŘÍLOHA B

INTERAKCE	337
Interakce Resizable	337
Interakce Draggable	340
Interakce Droppable	344
Interakce Sortable	345
Interakce Selectable	350

PŘÍLOHA C

EFEKTY	353
Efekt Blind	353
Efekt Bounce	353
Efekt Clip	354
Efekt Drop	354
Efekt Explode	354
Efekt Fade	355
Efekt Fold	355
Efekt Highlight	355
Efekt Puff	356
Efekt Pulsate	356
Efekt Scale	356
Efekt Shake	357
Efekt Size	357
Efekt Slide	358
Efekt Transfer	358

PŘÍLOHA D

ZÁSUVNÉ MODULY	359
Zásuvný modul Theme Switcher	359

REJSTŘÍK	361
-----------------------	------------

Úvod

V dnešní uspěchané době jsme všichni rádi, když můžeme nějakým způsobem ušetřit čas. U webových aplikací to platí dvojnásob. Aplikace lze vyvíjet složitě, ale také jednoduše. Buď jen uspoříme nějaký čas, nebo zbývající čas využijeme k testování a zlepšování své aplikace.

Každý, kdo někdy zkoušel programovat v jazyku JavaScript, ví, že situace není až tak jednoduchá jako u programovacích jazyků pro desktopové aplikace. Webové prostředí je značně nestabilní. Jednotlivé webové prohlížeče interpretují jazyky HTML, CSS a JavaScript odlišným způsobem a obvykle ne v souladu se standardy. Poslední dobou sice začíná svítat na lepší časy, ale stále musíme počítat se skutečností, že někteří uživatelé mohou mít starší verze webových prohlížečů.

Starší verze webových prohlížečů se vyznačovaly tím, že jsme napsali zdrojový kód jazyka JavaScript (případně jazyka HTML či CSS) v jednom prohlížeči a následně jsme zjistili, že v druhém webovém prohlížeči nefunguje, ačkoliv by měl. Napsat na první pokus kód fungující ve všech webových prohlížečích bylo spíše dílem náhody. Trendem současné doby je přesvědčovat uživatele, že přejít na moderní webový prohlížeč se jim vyplatí. Důvod nespočívá pouze v tom, že vývojáři se nemusejí s novými prohlížeči tolik vztekat a posílat je ke všem čertům, ale nové prohlížeče mají oproti těm starým mnoho funkcí, které ocení právě uživatelé.

Ne vždy se nám ale podaří uživatele přesvědčit. Například někteří uživatelé nemůžou změnit svůj webový prohlížeč, protože jim to interní směrnice nebo technologická infrastruktura jejich společnosti neumožňuje. Jiní uživatelé nemusejí ani vědět, jak svůj webový prohlížeč změnit nebo aktualizovat. Zkrátka a dobře – musíme počítat s tím, že nevytváříme aplikaci šitou na míru poslední verzi svého webového prohlížeče.

Vývoj složitějších projektů jen s použitím jazyka JavaScript dnes odpovídá téměř vraždě své konkurenceschopnosti. Každý vývojář potřebuje knihovnu, která se snaží potírat (v ideálním případě eliminovat) rozdíly mezi jednotlivými webovými prohlížeči. Nemusíme se hned hrnout do psaní své vlastní knihovny, protože přesně z výše zmíněných důvodů naprogramoval John Resig v roce 2006 knihovnu jQuery, aniž by tušil, jak oblíbenou se stane. Podobných knihoven jazyka JavaScript existuje celá řada, ale tato si získala velké množství fanoušků zejména díky snadné použitelnosti.

Na knihovně jQuery je založena také knihovna jQuery UI, kterou v roce 2007 naprogramoval Paul Bakaus. Tato knihovna má usnadnit tvorbu nejrůznějších dynamických prvků uživatelského rozhraní, a to až do takové míry, že v mnoha případech dosáhneme skvělých výsledků takřka bez práce. Právě této knihovně se budeme věnovat v této knize. Na některých místech se samozřejmě nevyhneme tématům z knihovny jQuery, jelikož se přece jen jedná o rodičovskou knihovnu pro knihovny jQuery UI.

Co najdete v této knize

Knihovně jQuery UI se budeme snažit porozumět na praktických příkladech, kterými je tato kniha protkána. Na začátku kapitoly si obvykle vytyčíme cíle, kterých chceme dosáhnout, na konci kapitoly si shrneme, jak (a jestli) se nám to podařilo. Tato kniha je rozdělena do devíti kapitol:

1. **Knihovna jQuery UI** – první kapitola obsahuje základní informace o knihovně jQuery UI, které se budeme věnovat v této knize.
2. **Motivy vzhledu** – kapitola pojednávající o tom, jak přizpůsobit vzhled ovládacích prvků knihovny jQuery UI svým vlastním představám.
3. **Základy knihovny jQuery** – základní informace o knihovně jQuery nezbytné pro lepší pochopení zdrojových kódů v jednotlivých příkladech.
4. **Navigace v obsahu stránky** – v této kapitole si ukážeme první ovládací prvky knihovny jQuery UI, s nimiž můžeme uspořádat informace na svých webových stránkách a ulehčit tak uživateli navigaci v jejich obsahu.
5. **Formuláře** – rady a nápady, jak nám knihovna jQuery UI může pomoci s tvorbou „inteligentních“ formulářů.
6. **Přesouvání objektů** – podrobný popis způsobu přesouvání objektů technikou táhni a pusť, demonstrováný na fiktivním internetovém obchodu.
7. **Seznamy** – na příkladu hudební knihovny zjistíme, jak lze pracovat se seznamy (a jinými podobnými elementy) v knihovně jQuery UI.
8. **Multimediální obsah** – v této kapitole sestavíme multimediální přehrávač založený na jazyku HTML5. Uživatelské rozhraní k tomuto přehrávači se bude samozřejmě skládat z tlačítek, indikátorů průběhu a jezdců, což jsou všechno ovládací prvky dostupné v knihovně jQuery UI.
9. **Fotogalerie** – vytvoříme fotogalerii typu lightbox s pomocí nástroje Position. Tento příklad bude o něco složitější, proto se budeme rovněž zabývat problémem, jak převést své představy do kódu a neztratit se v něm.

Za poslední kapitolou se nacházejí čtyři přílohy s tabulkami, které obsahují informace z oficiální dokumentace knihovny jQuery UI. Ačkoliv se většinou jedná o přesně přeložené informace, místy jsou doplněné, nebo naopak některé informace chybějí, a to z toho důvodu, že daná věc například v době psaní této knihy nefungovala, jak měla. Myslím, že tyto informace ocení i čtenáři, kteří jsou méně zblhlí v angličtině.

Komu je tato kniha určena

Tato kniha je určena samozřejmě všem vývojářům, kteří chtějí ušetřit čas při tvorbě dynamických webových aplikací takovým způsobem, že se ponoří do světa knihoven jQuery a jQuery UI. Zkušeným programátorům bude pravděpodobně připadat až příliš jednoduchá, ale začínající programátoři, kodéři, designéři a další počítačovní nadšenci ji jistě ocení. Všechna témata v knize jsou doplněna příklady popsanými podrobně krok za krokem.

Nebudete-li si vědět rady, nebojte se na chvíli si odpočinout a následně se k danému tématu vrátit znovu. Pokud přeskočíte kapitolu o základech knihovny jQuery, protože vám bude připadat příliš složitá, zkuste se k ní vrátit, až napíšete několik příkladů a zatoužíte lépe poznat, jak fungují „pod pokličkou“.

Tato kniha předpokládá aktivní znalost jazyků HTML a CSS. Vhodné je mít alespoň nějaké povědomí o tom, jak funguje jazyk JavaScript. Zkušenost s tvorbou moderní javascriptové aplikace není nutná, ale pokud občas používáte fragmenty kódu jazyka JavaScript ve svých webových stránkách, snadněji pochopíte příklady z této knihy.

Co budeme potřebovat k vývoji

Máte rádi skládačky? Snad ano, protože nyní budeme jednu takovou skládat. Abychom mohli psát zdrojový kód pro příklady v této knize, musíme posbírat všechny nezbytné prostředky, které budeme k vývoji potřebovat. Jedná se o tyto prostředky:

- moderní webový prohlížeč,
- vývojové prostředí,
- webový server,
- archiv se zdrojovými kódy příkladů z této knihy,
- knihovnu jQuery UI (a knihovnu jQuery, která však obvykle bývá součástí staženého balíku knihovny jQuery UI) s motivem vzhledu Start.

Knihovně jQuery UI a jejím motivům vzhledu se budeme věnovat v kapitolách „Knihovna jQuery UI“ a „Motivy vzhledu“. Knihovnu jQuery si stručně popíšeme v kapitole „Základy knihovny jQuery“.

Webový prohlížeč

Webový prohlížeč je nezbytnou součástí pro vývoj webových stránek a aplikací. Jedná se o tak samozřejmou součást, že jsme se o ní ani nemuseli zmiňovat. Nicméně pro úplnost a objasnění několika věcí ho nesmíme opomenout.

Velmi pravděpodobně již máte svůj oblíbený webový prohlížeč. Pokud jste si zatím prohlížeč nevybrali, vhodné je zvolit si takový, k němuž je k dispozici nějaký ladicí nástroj pro kód jazyků HTML, CSS a JavaScript. Takový webový prohlížeč není těžké najít, protože téměř všechny moderní webové prohlížeče nabízejí vývojářské nástroje buď přímo, nebo jako doplněk.

LADĚNÍ ZDROJOVÉHO KÓDU

Ladění kódu pomocí ladicího nástroje velmi zjednodušuje práci. Věřte, že žádný člověk není neomylný, a o vývojářích to platí dvojnásob. Někteří z vás se už možná někdy s takovým nástrojem setkali, nebo jej dokonce pravidelně používají, ale asi jsou mezi vámi i lidé, kteří se s ním nikdy nesetkali nebo ho používat nechtějí.

Ladění kódu lze považovat za pokročilejší téma, proto si ho v této knize popisovat nebudeme. Pokud se však někdy přistihnete, že přidáváte do kódu příkazy vypisující nějaké hodnoty na obrazovku jen proto, abyste odhalili chybu, vzpomeňte si na tento odstavec a uvědomte si, že právě začínáte ladit kód. Tato situace nastane dříve, než si někteří z vás možná myslí. V takovém případě platí jedna rada – okamžitě zbystřete, zkuste vyhledat nějaký ladicí nástroj a naučte se s ním pracovat. Dosáhnete stejného výsledku, jako byste vkládali do kódu ladicí vypisující příkazy, ale mnohem snadněji a nehrozí, že byste po odhalení chyby zapoměli své nadbytečné příkazy odstranit.

Příklady v této knize jsou otestované v těchto webových prohlížečích:

- Firefox 11 od společnosti Mozilla,
- Google Chrome 18 od společnosti Google,
- Opera 11 od společnosti Opera,
- Internet Explorer 9, 8 a 7 od společnosti Microsoft.

Velmi pravděpodobně budou fungovat i ve velké řadě dalších moderních webových prohlížečů. Pokud se mezi vámi nachází příznivce staříckého prohlížeče Internet Explorer 6 od společnosti Microsoft, velmi se omlouvám za svůj negativní přístup k němu. Přesto si myslím, že většina vývojářů přivítala takovou typicky smutnou událost, jako je pohřeb tohoto prohlížeče, s velkým nadšením. Jestliže vás tato událost zcela minula, zkuste vyhledat ve vyhledávací Google výraz „pohřeb prohlížeče Internet Explorer 6“. Jistě jste z tohoto textu pochopili, že příklady v této knize nejsou odladěné pro prohlížeč Internet Explorer ve verzi 6 a nižší (pravděpodobně můžete narazit na problémy i s prohlížečem Internet Explorer 7, který nemá k oficiálnímu pohřbu také daleko).

Vývojové prostředí

Jestliže máte své oblíbené vývojové prostředí, s nímž jste spokojeni, můžete tuto část bez obav přeskocit.

Pro vývoj dynamických webových stránek je nevhodnější textový editor schopný pracovat s prostým textem. Do této kategorie nespadá kupříkladu program Word od společnosti Microsoft, protože tento program je textovým procesorem, jenž generuje binární soubory obsahující text se styly a jinými metadaty. Patří sem ale nástroj Poznámkový blok z operačního systému Windows od stejné společnosti.

Znamená to tedy, že nástroj Poznámkový blok je nejlepší volbou pro vývoj? Rozhodně ne. Problém spočívá v tom, že pracuje s kódem jako s obecným textem. Ve zdrojovém kódu lze ale rozlišit klíčová slova, proměnné, funkce a další prvky. Určitě oceníte, pokud váš editor

zvýrazní speciální prvky příslušného programovacího jazyka. Podívejte se na obrázek Ú.1 s kódem jazyka JavaScript bez zvýrazněné syntaxe.

```

/**
 * Zpracuje událost načtení karty Pokec.
 */
function zpracujNacteniKartyPokec() {
  $('#upozorneni').dialog({
    autoOpen: false,
    buttons: {
      'Beru na vědomí': zavriDialogoveOkno
    },
    modal: true,
    resizable: false
  });

  /* Dialogové okno s upozorněním zobrazíme jen za předpokladu, že se
  uživatel nepřihlásil a poprvé načítá obsah karty Pokec. */
  if (!uzivatelSePrihlasil() &&
    !zpracujNacteniKartyPokec.oknoUzJsmeZobrazili) {
    $('#upozorneni').dialog('open');
    zpracujNacteniKartyPokec.oknoUzJsmeZobrazili = true;
  }
}

```

Obrázek Ú.1: Kód jazyka JavaScript bez zvýrazněné syntaxe

A nyní si prohlédněte stejný kód se zapnutým zvýrazňováním syntaxe.

```

/**
 * Zpracuje událost načtení karty Pokec.
 */
function zpracujNacteniKartyPokec() {
  $('#upozorneni').dialog({
    autoOpen: false,
    buttons: {
      'Beru na vědomí': zavriDialogoveOkno
    },
    modal: true,
    resizable: false
  });

  /* Dialogové okno s upozorněním zobrazíme jen za předpokladu, že se
  uživatel nepřihlásil a poprvé načítá obsah karty Pokec. */
  if (!uzivatelSePrihlasil() &&
    !zpracujNacteniKartyPokec.oknoUzJsmeZobrazili) {
    $('#upozorneni').dialog('open');
    zpracujNacteniKartyPokec.oknoUzJsmeZobrazili = true;
  }
}

```

Obrázek Ú.2: Kód jazyka JavaScript se zvýrazněnou syntaxí

Kdybyste měli určit, kolik výskytů pravdivostních hodnot `true` a `false` se tento kód obsahuje, určitě byste to rychleji zjistili z obrázku Ú.2, protože se na něm zobrazují *kurzívou*. Zvýrazňování syntaxe však není jedinou vlastností, kterou některé vývojové nástroje nabízejí. Zajistě vám přijde vhod také zvýrazňování párů složených z počáteční a koncové závorky, a to pro kulaté, hranaté i složené závorky. Existuje spousta zajímavých funkcí, které usnadňují psaní (a také čtení) zdrojového kódu.

Vývojových prostředí je k dispozici opravdu mnoho. Zde uvedeme dvě jednoduchá a velmi oblíbená vývojová prostředí pro vývoj v jazycích HTML, CSS a JavaScript.

PSPad Editor

Český vývojový nástroj PSPad Editor je velmi oblíbeným editorem zdrojových kódů. K jeho rozšíření jistě přispělo, že jej jeho autor (Jan Fiala) nabízí zdarma. Mezi jeho funkce patří práce s více dokumenty současně, zvýrazňování syntaxe pro spoustu programovacích jazyků, konverze textu mezi různými znakovými sadami a bezpočet dalších. Dostupná je také řada rozšíření.

Tento nástroj můžete stáhnout z oficiálních webových stránek na adrese <http://www.pspad.com/cz/>. Určitě si zaslouží vyzkoušet. Pokud se vám program zalíbí a přinese vám užitek, můžete autora za jeho snahu obdarovat – na uvedených webových stránkách najdete další informace.

Notepad++

Nástroj Notepad++ nese název od zmiňovaného nástroje Poznámkový blok (v angličtině Notepad); obohacují ho však dvě znaménka plus. Je nutné poznamenat, že těch plus je mnohem více, co se týká jeho funkčnosti. Program Notepad++ nabízí zvýrazňování syntaxe pro různé programovací jazyky, převod textu mezi znakovými sadami, funkci automatického dokončování, klienta FTP v podobě zásuvného modulu a spoustu jiných funkcí.

Tento program naprogramoval a udržuje jej Don Ho. Výchozím jazykem tohoto nástroje je angličtina, ale k dispozici je spousta lokalizovaných textů (mimo jiné i pro češtinu).

Nástroj Notepad++ si můžete stáhnout z oficiálních webových stránek na adrese <http://notepad-plus-plus.org/>. Opět v případě, že by se vám program líbil, můžete autora odměnit.

Webový server

Možná se ptáte, k čemu byste potřebovali webový server, když jazyk JavaScript je skriptovacím jazykem na straně klienta a stránky HTML obohacené kaskádovými styly rovněž fungují na vašem lokálním počítači.

Webový server potřebujete, protože některé příklady používají technologii AJAX. Jestliže chcete otestovat funkčnost založenou na technologii AJAX, musíte spustit své webové stránky přímo přes webový prohlížeč a protokol HTTP. Ke kořenovému adresáři lokálního webového serveru se běžně dostanete zadáním adresy URL <http://localhost/> (nebo <http://127.0.0.1/>) do panelu adresy svého webového prohlížeče. Kdybyste totiž spustili své webové stránky například poklepáním na soubor *index.html* v programu Průzkumník Windows, funkce založené na technologii AJAX by nefungovaly.

Webový server se spoustou dalších služeb samozřejmě nabízejí všichni poskytovatelé hostingových služeb, a pokud provozujete nějaké vlastní webové stránky, jistě už jste jejich služby použili. Některé tyto služby jsou k dispozici zdarma; za jiné se platí, přičemž přidanou hodnotou bývá obvykle zvýšená dostupnost a technická podpora. Většina bezplatných řešení si rovněž vynucuje vložení reklamy. Svá data obvykle nahráváte do adresáře webového serveru pomocí klienta FTP nebo webového rozhraní nabízeného daným poskytovatelem. Jestliže nemáte vlastní firemní řešení, určitě budete tyto služby potřebovat pro produkč-

ní nasazení svých webových stránek. Pro účely vývoje je však toto řešení nevhodné – bez webového serveru na svém lokálním počítači se prostě neobejdete, jelikož tato varianta je rychlejší a pohodlnější.

Opět samozřejmě platí, že pokud máte svůj oblíbený webový server, můžete jeho výběr přeskočit.

Pro naše účely by stačilo nainstalovat velmi oblíbený webový server Apache samotný, nicméně jeho instalace a konfigurace je trochu obtížnější. Proto si popíšeme, jak nainstalovat český balík Complex Web Server, a to na operační systém Windows. Jestliže máte jiný operační systém, zkuste na Internetu vyhledat například postup instalace webového serveru Apache pro svůj operační systém.

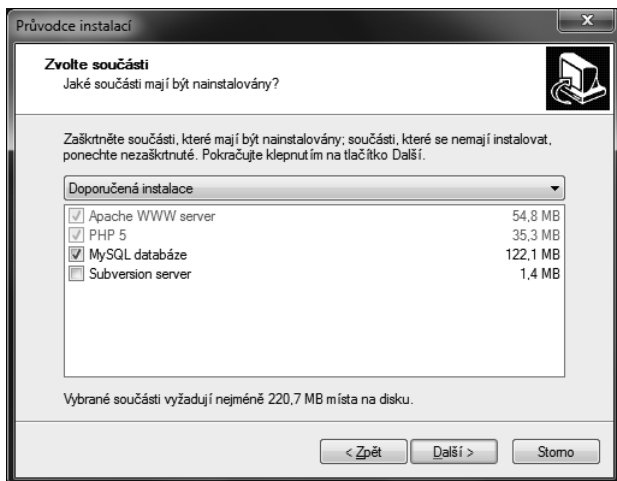
Instalace balíku Complex Web Server

Balík Complex Web Server můžete stahovat z oficiálních stránek na adrese <http://ponkrac.net/complex-web-server/cs>. Autorem tohoto balíku je ing. Miloslav Ponkrác. Cílem tohoto balíku je usnadnění konfigurace webového serveru Apache spolu s podporou jazyka PHP a databázového stroje MySQL. Navíc díky němu můžete získat i systém pro správu verzí.

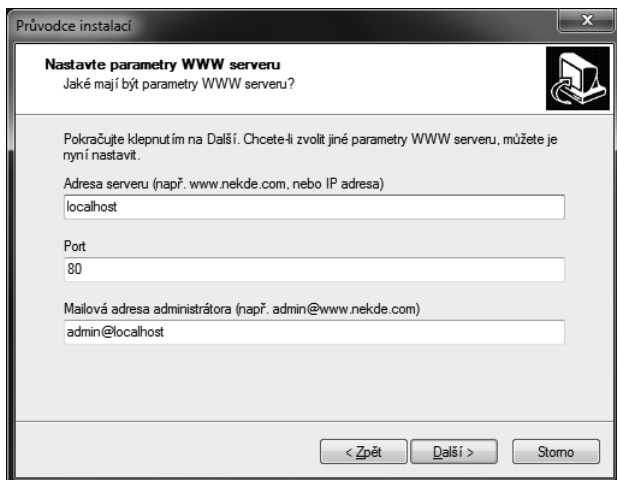
Nyní si popíšeme, jak nainstalovat balík Complex Web Server ve stabilní verzi 1.4.14, jež lze stáhnout z adresy <http://ponkrac.net/file/cws/complex-web-server-1-4-14/>:

1. Spustíte instalační program balíku Complex Web Server 1.4.14.
2. Potvrďte, že chcete pokračovat v instalaci.
3. V dialogovém okně s výběrem jazyka pro průvodce instalací ponechejte vybranou češtinu a klepnutím na tlačítko **OK** spustíte průvodce instalací.
4. Přejděte k druhému kroku instalačního průvodce – na této stránce můžete zvolit cílové umístění balíku Complex Web Server. Výchozím umístěním je složka `c:\ComplexWebServer\`. Po výběru cílové složky klepněte na tlačítko **Další**.
5. Na stránce s volbou součástí si můžete vybrat součásti, které chcete nainstalovat. Minimální konfigurace obsahuje jen webových server Apache a interpreter jazyka PHP 5. V doporučené konfiguraci se navíc nachází databázový stroj MySQL a plná instalace obsahuje navíc systém pro správu verzí. V následujících krocích si popíšeme postup pro výchozí doporučenou konfiguraci (viz obrázek Ú.3). Přejděte k dalšímu kroku instalačního průvodce.
6. V následujícím kroku si můžete zvolit název složky pro tento balík v nabídce **Start**. Výchozí název *Complex Web Server* je však nadměru výstižný. Klepnutím na tlačítko **Další** zobrazíte další stránku instalačního průvodce.
7. V kroku s nastavením parametrů webového serveru ponechejte výchozí hodnoty a klepněte na tlačítko **Další**. Tento krok můžete vidět na obrázku Ú.4.
8. V kroku s nastavením hesla pro databázi MySQL nechejte výchozí hodnotu 3306 v poli **Port**. Musíte zadat heslo pro uživatele root databázového stroje MySQL do polí **Heslo uživatele root** a **Zde zadejte heslo ještě jednou**. Toto heslo si zapamatujte nebo bezpečně poznamenejte. Obrázek Ú.5 znázorňuje tento krok.
9. V předposledním kroku instalačního průvodce můžete nechat zaškrťovací políčka s dalšími úlohami instalace zaškrtnutá tak, jak jsou.

10. V posledním kroku instalačního průvodce klepněte na tlačítko **Instalovat**, čímž nainstalujete balík Complex Web Server.

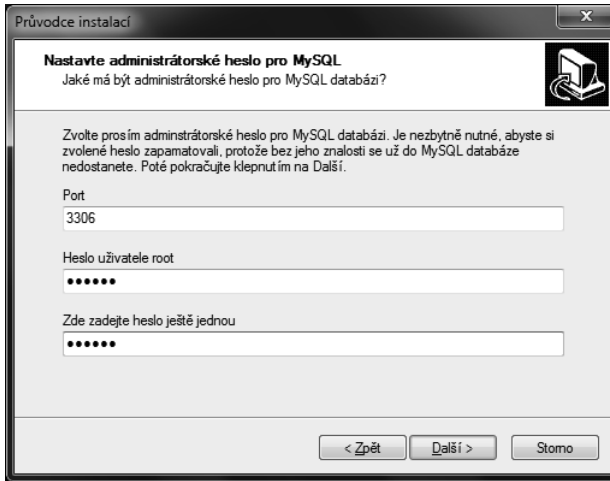


Obrázek Ú.3: Volba součástí balíku Complex Web Server



Obrázek Ú.4: Nastavení parametrů webového serveru

Zda se instalace skutečně zdařila a webový server Apache běží, můžete otestovat jednoduše. Zadejte adresu `http://localhost/` do panelu adresy svého webového prohlížeče. V případě úspěšné instalace byste měli vidět něco podobného jako na obrázku Ú.6.



Obrázek Ú.5: Nastavení hesla databázového stroje MySQL

Jméno	Změněno	Velikost	Popis
~admin/	23.10.2011 13:41:52		
~users/	23.10.2011 13:40:52		
complex_web_server_manual.http	15.05.2008 13:23:30	185 B	
complex_web_server_manual.php	15.06.2008 08:32:16	83 B	PHP skript
phpMyAdmin.php	15.06.2008 08:31:18	61 B	PHP skript

Complex Web Server 1.4.14
Copyright © 2008 Ing. Milošlav Ponkrác

Obrázek Ú.6: Obsah kořenového adresáře webového serveru

Webový server po zadání této adresy do webového prohlížeče zobrazuje obsah svého kořenového adresáře. Tento kořenový adresář webového serveru odpovídá složce `c:\Complex-WebServer\http_docs\`, pokud jste zvolili výchozí cílové umístění v instalačním průvodci.

Archiv se zdrojovými kódy příkladů

Při práci s příklady v této knize se neobejdete bez archivu, který můžete stáhnout z adresy <http://knihy.cpress.cz/K2000> po klepnutí na odkaz Soubory ke stažení. Tento archiv obsahuje složku *Základy příkladů* se základní strukturou adresářů a souborů pro příklady z jednotlivých kapitol. Obsah této složky se dělí na kapitoly, takže obsahuje složky s předponou *Kapitola* a číslem kapitoly; například *Kapitola04*.

Je vhodné si zkopírovat základy příkladů do složky, k níž máte právo zápisu. Teoreticky lze zkopírovat základy příkladů do jakékoliv takové složky; vhodným umístěním je však podsložka kořenového adresáře webového serveru. Některé příklady vyžadují ke svému fungování webový server, tuto skutečnost si u nich zdůrazníme. K základu příkladu budete postupně doplňovat soubory nebo kód, až vznikne plně funkční verze příkladu.

Abyste zkontrolovali, jestli jste si počínali při práci na příkladu správně, můžete si prohlédnout jeho kompletní verzi. Plně funkční verze příkladů s kompletními zdrojovými kódy najdete rovněž v tomto archivu, a to ve složce *Kompletní příklady*, jež má stejné uspořádání jako složka *Základy příkladů*.

V tomto archivu se nachází ještě jedna složka – *jQuery UI 1.8.18*. Tato složka obsahuje knihovnu jQuery UI ve verzi 1.8.18 se všemi komponentami a s motivem vzhledu Start. Je zde z toho důvodu, že pro tuto verzi knihovny byly otestovány všechny příklady v této knize. Nové i starší verze této knihovny jsou samozřejmě dostupné na oficiálních stránkách této knihovny, jak uvidíte v příští kapitole.

Konvence zápisu textu

Pro lepší srozumitelnost textu v knize jsou některé části odlišené na základě svého významu různými vlastnostmi písma a odstavce.

Termíny jsou označené **tučným písmem**. Příklad: „K tomuto účelu použijeme ovládací prvek, jenž se nazývá **jezdec**“.

Adresy URL jsou napsané *kurzívou*. Příklad: „Pokud se chcete dozvědět více informací o rozhraní Google Maps JavaScript API verze 3, najdete je na adrese <http://code.google.com/intl/cs/apis/maps/documentation/javascript/services.html#GeocodingRequests>“.

Ovládací prvky uživatelského rozhraní jsou zvýrazněné **tučným písmem**. Příklad: „Jak je patrné, pod názvem každého motivu vzhledu je možné najít dvě tlačítka – tlačítko **Download** a tlačítko **Edit**“.

Soubory, složky a jiné objekty systému souborů jsou označené *kurzívou*. Příklad: „Soubor *hlavni.js* ze složky *js* momentálně vypadá následovně“.

Funkce, proměnné a jiné prvky programovacího jazyka budeme psát ve větě neproporcionálním písmem. Příklad: „Z předchozí verze naší funkce `nactiNavrhyObci()` víme, že tento text získáme jako hodnotu vlastnosti `pozadavek.term`“.

Rozsáhlejší výpisy zdrojového kódu budou vždy začínat na samostatném řádku a budou rovněž psané neproporcionálním písmem. **Tučným neproporcionálním písmem** budeme zvýrazňovat důležité části kódu, nebo ty části kódu, které je nutné doplnit do již existujícího zdrojového kódu. Tři tečky (. . .) budou naopak označovat místa, která se oproti dřívějšímu stavu nezměnila, nebo nejsou z hlediska popisovaného problému podstatná. Příklad:

```
function zobrazProduktyVKosiku() {
    ...
    $('img.nahled-produktu').draggable({
        ...
        stop: function() {
            var $nahled = $(this);
            if (!$nahled.data('probihaOdstranovani')) {
                $nahled.show();
            }
        }
    });
}
```

Kromě toho rozlišujeme také níže uvedené speciální typy odstavců.



POZNÁMKA

Jakákoliv informace doplňující výklad.



TIP

Zajímavé informace k danému tématu.



UPOZORNĚNÍ

Důležité informace, které obhajují zvolené postupy a varují před potenciálními problémy.

NADPIS POZNÁMKY

Rozsáhlá poznámka na několik odstavců.

Zpětná vazba od čtenářů

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

Computer Press
Albatros Media a. s., pobočka Brno
 IBC
 Příkop 4
 602 00 Brno

nebo

sefredaktor.pc@albatrosmedia.cz

Computer Press neposkytuje rady ani jakýkoliv servis pro aplikace třetích stran. Pokud budete mít dotaz k programu, obraťte se prosím na jeho tvůrce.

Errata

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

Veškerá existující errata zobrazíte na adrese <http://knihy.cpress.cz/K2000> po klepnutí na odkaz Soubory ke stažení.

Knihovna jQuery UI

Na začátek by se slušelo uvést několik informací o knihovně jQuery UI, která je hlavním tématem této knihy. V této kapitole se tedy dozvíte:

- Co je to knihovna jQuery UI.
- Kde byste měli hledat oficiální dokumentaci ke knihovně jQuery UI a co v ní najdete.
- Kde stáhnete balík obsahující knihovnu jQuery UI.
- Jakou verzi knihovny jQuery UI byste si měli vybrat.
- Jak si můžete sestavit vlastní balík s knihovnou jQuery UI.

Co je knihovna jQuery UI

Knihovna jQuery UI je knihovna s otevřeným zdrojovým kódem, která nabízí spoustu okamžitě použitelných komponent. Její komponenty se dělí do čtyř kategorií – interakce, ovládací prvky, efekty a pomocné nástroje. Hlavní výhoda této knihovny spočívá ve snadné použitelnosti a možnostech stylování. K dispozici je celá řada motivů vzhledu, mezi nimiž můžeme vybírat, nebo si vytvářet své vlastní.

Knihovna jQuery UI spatřila poprvé světlo světa 17. září 2007. Zakládá se na velmi oblíbené javascriptové knihovně jQuery. Tvůrcem knihovny jQuery UI je Paul Bakaus, který však momentálně není jejím aktivním vývojářem, ale na jejím vývoji se podílejí (nebo dříve podílely) desítky vývojářů, a přestože svou práci dělají rádi a zdarma, řada komerčních společností a vývojářů na volné noze jim poskytla dobrovolné příspěvky. Pokud je chcete rovněž obdarovat, na oficiálních stránkách této knihovny (<http://jqueryui.com/>) najdete podrobné informace.

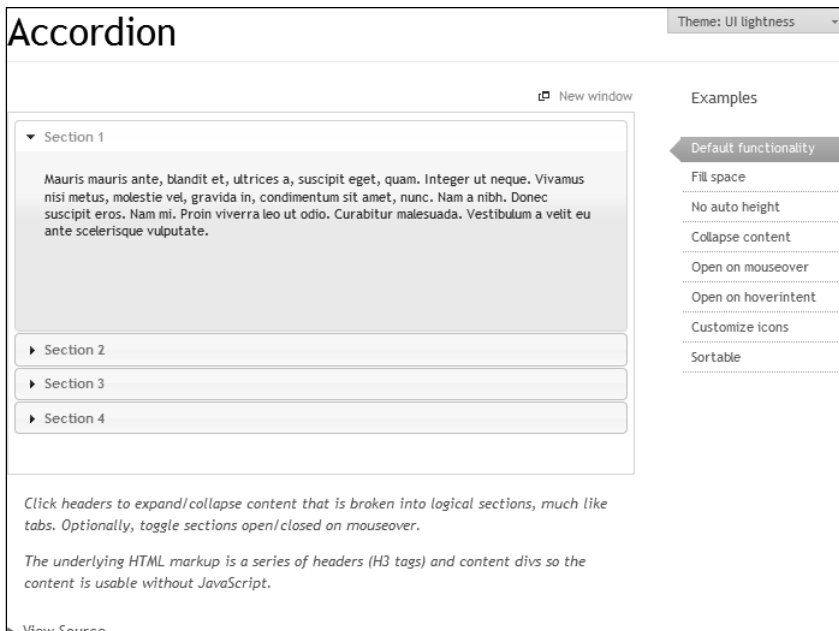
Autoři knihovny jQuery UI nabízejí tuto knihovnu zdarma pro otevřené i komerční projekty pod licencemi GPL a MIT. Jestliže nevíte, co obnáší licence GPL, můžete se to dočíst na internetové adrese http://cs.wikipedia.org/wiki/GNU_General_Public_License. Popis licence MIT se nachází na adrese http://cs.wikipedia.org/wiki/Licence_MIT. Text licence GPL pro knihovnu jQuery UI najdete na adrese <http://jquery-ui.googlecode.com/svn/tags/latest/GPL-LICENSE.txt> a text licence MIT na adrese <http://jquery-ui.googlecode.com/svn/tags/latest/MIT-LICENSE.txt>.

Dokumentace knihovny

Co všechno knihovna jQuery UI umí, můžete lehce zjistit sami, jelikož na oficiálních stránkách této knihovny se nachází rozsáhlá dokumentace všech komponent spolu s ukázkami; konkrétně na adrese <http://jqueryui.com/demos/>.

Informace o možnostech nastavení, metodách a událostech jednotlivých komponent najdete v přílohách k této knize. Informace v přílohách jsou lokalizované do češtiny, přičemž na některých místech jsou doplněné, nebo naopak ochuzené v případě, že určitá věc v době psaní této knihy (tj. ve verzi jQuery UI 1.8.18) nefungovala, jak měla.

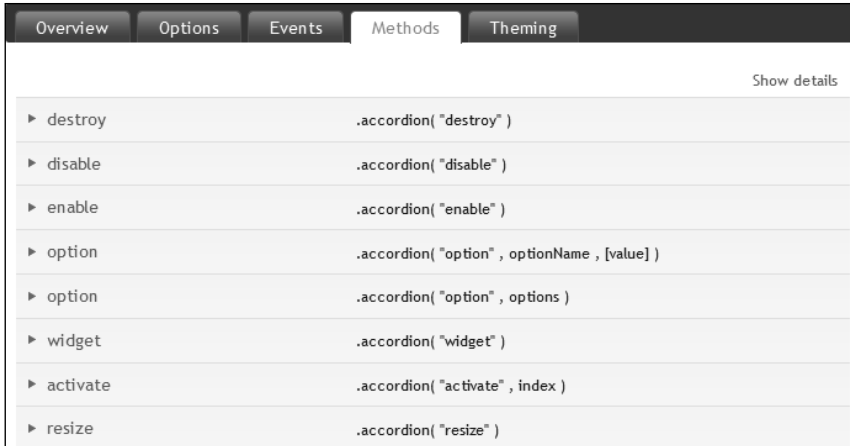
Vraťme se ale k ukázkám komponent na oficiálních webových stránkách. Popíšme si kupříkladu dokumentaci ovládacího prvku Accordion na adrese <http://jqueryui.com/demos/accordion/>. Tuto stránku si můžete prohlédnout také na obrázku 1.1.



Obrázek 1.1: Ukázka ovládacího prvku Accordion

Jak je patrné z předchozího obrázku, vpravo se nachází seznam příkladů pro ovládací prvek Accordion. Mezi jednotlivými příklady můžeme přepínat a výsledky sledovat vlevo na reálných ukázkách. Příklady komponent jsou zcela funkční a nejen to – můžeme dokonce měnit jejich vzhled. Vpravo nahoře najdeme rozevírací seznam **Theme** s dostupnými motivy vzhledu. Jakmile v něm zvolíme jiný motiv vzhledu, aplikuje se na všechny příklady komponent knihovny jQuery UI.

Pokud si chcete prostudovat, jak daný příklad funguje, můžete si přečíst jeho zdrojový kód po klepnutí na odkaz **View Source**. Všechny informace o dané komponentě knihovny jQuery UI se pak nacházejí ve spodní části této stránky, jak lze spatřit na obrázku 1.2.



Method	Signature
▶ destroy	<code>.accordion("destroy")</code>
▶ disable	<code>.accordion("disable")</code>
▶ enable	<code>.accordion("enable")</code>
▶ option	<code>.accordion("option", optionName, [value])</code>
▶ option	<code>.accordion("option", options)</code>
▶ widget	<code>.accordion("widget")</code>
▶ activate	<code>.accordion("activate", index)</code>
▶ resize	<code>.accordion("resize")</code>

Obrázek 1.2: Informace o ovládacím prvku Accordion

Zde se dozvíte obecné informace o dané komponentě knihovny jQuery UI, dále informace o možnostech nastavení, metodách, událostech a možnostech stylování.

Se studiem těchto informací však nemusíte pospíchat – vše se postupně dozvíte v průběhu této knihy na praktických příkladech a také z přehledů v přílohách. Je však vhodné, abyste kontrolovali rovněž informace z oficiální dokumentace, jestli se od vydání této knihy něco nezměnilo. Změny jsou totiž velmi pravděpodobné, protože tato knihovna se neustále vyvíjí.

Stažení knihovny

Knihovnu jQuery UI je možné stahovat ze stránky na adrese <http://jqueryui.com/download>. Na této stránce máte na výběr z několika možností. Ke stažení můžete použít takzvané rychlé odkazy v horní části stránky, jak vidíte na obrázku 1.3.

Rychlé odkazy umožňují stahovat standardní balík se stabilní verzí knihovny jQuery UI a také se starší verzí této knihovny. K dispozici jsou rovněž balíčky s plnými zdrojovými kódy této knihovny. Pokud chcete stáhnout taktéž motiv vzhledu dle vlastní volby, můžete jej stáhnout přes další rychlý odkaz.

Build Your Download

Customize your jQuery UI download by selecting the version and specific modules you need in the form below or select a quick download package. A range of current and historical jQuery UI releases are also hosted on Google's CDN.



Quick downloads: Stable (Themes) (1.8.18: for jQuery 1.3.2+) | Legacy (Themes) (1.7.3: for jQuery 1.3+)

Source (pre-build): Stable (1.8.18: for jQuery 1.3.2+) | Master (unreleased) | GitHub repo

1.8.18 Themes: base, black-tie, blitzr, cupertino, dark-hive, dot-luv, eggplant, excite-bike, flick, hot-sneaks, humanity, le-frog, mint-choc, overcast, pepper-grinder, redmond, smoothness, south-street, start, sunny, swanky-purse,trontastic, ui-darkness, ui-lightness, and vader.

Components (31 of 31 selected)		Deselect all components	
UI Core A required dependency, contains basic functions and initializers.	<input checked="" type="checkbox"/> Core	The core of jQuery UI, required for all interactions and widgets.	
	<input checked="" type="checkbox"/> Widget	The widget factory, base for all widgets	
	<input checked="" type="checkbox"/> Mouse	The mouse widget, a base class for all interactions and widgets with heavy mouse interaction.	
	<input checked="" type="checkbox"/> Position	A utility plugin for positioning elements relative to other elements.	
		Deselect all	
Interactions These add basic behaviors to	<input checked="" type="checkbox"/> Draggable	Makes any element on the page draggable.	
	<input checked="" type="checkbox"/> Droppable	Creates drop targets for draggable elements.	

Theme
Select the theme you want to include or design a custom theme

UI lightness

▶ Advanced Theme Settings

Version
Select the release version you want to download.

1.8.18
(Stable, for jQuery 1.3.2+)

1.7.3
(Legacy release, for jQuery 1.3.2)

Obrázek 1.3: Stránka pro stahování knihovny jQuery UI



POZNÁMKA

Jak už jsme si řekli v úvodu, v archivu se zdrojovými kódy k příkladům v této knize je k dispozici balík s kompletní knihovnou jQuery UI ve verzi 1.8.18 a motivem vzhledu Start. Pro tuto verzi knihovny jsou odladěné všechny příklady v této knize. Za účelem testování příkladů z této knihy tudíž nemusíte stahovat knihovnu jQuery UI. Zde uvedené postupy vám spíše přijdou vhod, až budete chtít použít tuto knihovnu ve svých vlastních projektech.

Balík s knihovnou jQuery UI můžete také stahovat ze sítě CDN společnosti Google na adrese <http://code.google.com/intl/cs/apis/libraries/devguide.html#jqueryUI>. Na těchto stránkách najdete kromě nejnovějších verzí této knihovny i její starší verze. Výhodou sítě CDN společnosti Google je, že můžete skripty načítat přímo z této sítě – nemusíte je tedy stahovat. Tuto techniku si však popisovat nebudeme – jako každá věc má svoje výhody i nevýhody. Jestliže se chcete dozvědět více informací, na právě uvedené adrese je najdete.

Rychlé odkazy a síť CDN společnosti Google mají jednu velkou nevýhodu, a tou je, že si nemůžeme sestavit vlastní balík. Výše uvedená stránka pro stahování knihovny jQuery UI (<http://jqueryui.com/download>) ve skutečnosti poskytuje nástroj, jenž nám dovoluje sestavit si vlastní balík. Vlastní balík obsahuje námi vybranou verzi knihovny jQuery UI, námi zvolený motiv vzhledu, a také nemusí obsahovat celou knihovnu jQuery UI, ale můžeme si vybrat jen určité komponenty. Ve zbytku této kapitoly si povíme jak na to.

Výběr verze

V pravé části nástroje pro sestavení balíku knihovny jQuery UI si můžeme vybrat verzi, a to v části označené **Version**. Na tuto část stránky se můžete podívat na obrázku 1.4.



Obrázek 1.4: Část s výběrem verze knihovny jQuery UI

Zde si můžete vybrat stabilní verzi knihovny jQuery UI nebo její starší verzi. Pokud nemáte důvod ke stažení starší verze, je vhodné stahovat stabilní verzi. Důvodem pro výběr starší verze knihovny jQuery UI může být například to, že ve svém projektu používáte starší verzi knihovny jQuery, pro kterou jste odladili všechny své skripty jazyka JavaScript a víte, že s novější verzí knihovny jQuery by nefungovaly.

Měli byste tedy věnovat zvýšenou pozornost tomu, s jakými verzemi knihovny jQuery vybraná knihovna jQuery UI spolupracuje. Ne všechny kombinace knihovny jQuery a jQuery UI jsou vzájemně kompatibilní. S tímto problémem se setkáte jen v situaci, kdy máte vlastní verzi knihovny jQuery, kterou se nechystáte v brzké době inovovat, jelikož balík knihovny jQuery UI obsahuje i kompatibilní verzi knihovny jQuery.

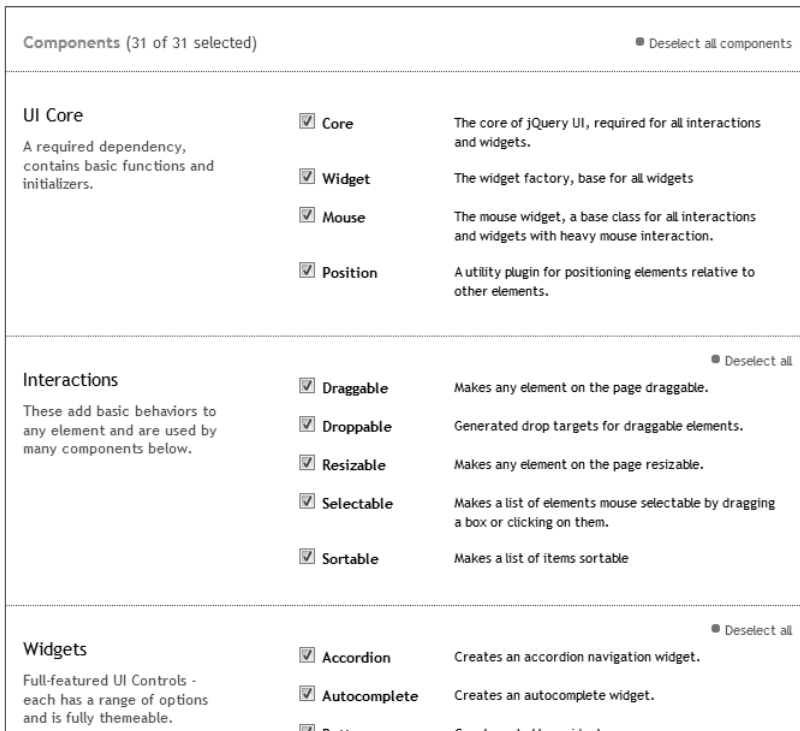
Když se podíváte na předchozí obrázek, uvidíte na něm přímo příklad toho, co jste se právě dozvěděli. Ihned totiž zjistíte, že když chcete použít knihovnu jQuery UI ve verzi 1.8.18, musíte mít knihovnu jQuery ve verzi 1.3.2 nebo novější.

Výběr komponent

V levé části nástroje pro sestavení balíku ke stažení, konkrétně v části **Components**, lze vybrat komponenty, které budou součástí balíku. Tuto část znázorňuje obrázek 1.5.

Komponenty se dělí do několika kategorií – Jádro knihovny jQuery UI (**UI Core**), Interakce (**Interactions**), Ovládací prvky (**Widgets**) a Efekty (**Effects**). Standardně bývají vybrané všechny komponenty. Pokud ale víte, že některé komponenty zcela jistě nebudete potřebovat, je rozumné zrušit jejich výběr. Čím více komponent bude výsledný skript knihovny obsahovat, tím bude větší a tím pomaleji se budou načítat vaše webové aplikace.

Výběr komponent je možné zrušit přímo u jednotlivých komponent odškrtnutím příslušných zaškrtačacích polí, nebo můžeme zrušit výběr všech komponent některé kategorie klepnutím na její odkaz **Deselect all**; případně zrušit výběr všech komponent klepnutím na odkaz **Deselect all components**. Když zrušíte výběr všech komponent, automaticky se zruší také výběr motivu vzhledu, proto vybírejte motiv vzhledu raději až po komponentách.

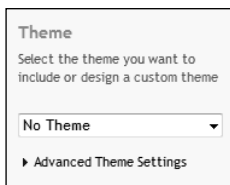


Obrázek 1.5: Část pro výběr komponent knihovny jQuery UI

Při rušení výběru komponent si pravděpodobně všimnete, že některá zaškrťovací pole není možné odškrtnout. Pokud se tak stane, vězte, že se nejedná o chybu, ale zkoušíte odebrat komponentu, na níž jsou závislé některé jiné komponenty, které jsou stále vybrány.

Výběr některého ze standardních motivů vzhledu

Do vlastního balíku si můžete vybrat také motiv vzhledu. V pravé části si jej můžete zvolit v sekci **Theme**, jak ukazuje obrázek 1.6.



Obrázek 1.6: Část pro výběr motivu vzhledu knihovny jQuery UI

Jak vypadají jednotlivé komponenty s vybraným motivem vzhledu, si můžete ověřit v oficiální dokumentaci knihovny jQuery UI. Že je možné aplikovat motiv vzhledu na ukázky jednotlivých komponent, jsme si už říkali v části „Dokumentace knihovny“ této kapitoly.

Zatím budeme vybírat jen některý ze standardních motivů a v následující kapitole „Motivy vzhledu“ se podíváme na tuto problematiku podrobněji.

Obsah balíku knihovny jQuery UI

Běžný balík s knihovnou jQuery UI obsahuje spoustu souborů a složek. Zde si popíšeme tuto strukturu pro verzi 1.8.18:

- */index.html* – tento soubor v kořenové složce obsahuje ukázky jednotlivých komponent.
- */css/{název-motivu-vzhledu}/* – tato složka obsahuje šablonu kaskádových stylů pro vybraný motiv a také nezbytné obrázky.
- */js/* – tato složka obsahuje **minifikované verze** skriptů knihoven jQuery a jQuery UI. Tyto skripty jsou vhodné pro produkční nasazení, jelikož mají mnohem menší velikost než jejich plné verze.
- */development-bundle/* – tato složka obsahuje plnou verzi skriptu knihovny jQuery, která je vhodná pro vývoj, dále pak texty licencí a další podsložky.
- */development-bundle/demos/* – podrobné ukázky jednotlivých komponent jako v online dokumentaci.
- */development-bundle/docs/* – informace o komponentách jako v online dokumentaci.
- */development-bundle/external/* – nezbytné externí zásuvné moduly sloužící pro vývoj knihovny jQuery UI, a to včetně javascriptového testovacího frameworku QUnit.
- */development-bundle/themes/* – vybrané motivy vzhledu podobně jako ve složce */css/*, avšak tentokrát jsou součástí i soubory s pravidly kaskádových stylů pro jednotlivé komponenty.
- */development-bundle/ui/* – tato složka obsahuje plnou verzi skriptu knihovny jQuery UI, soubory s kódem pro jednotlivé komponenty a také jejich minifikované verze. Kromě toho najdete v podsložce */development-bundle/ui/i18n/* lokalizace ovládacího prvku DatePicker.

Základní způsob použití souborů z tohoto balíku si popíšeme v kapitole „Navigace v obsahu stránky“.

Shrnutí

V této kapitole jsme si řekli několik základních informací o knihovně jQuery UI. Již víme, k čemu tato knihovna slouží a co umí. Dále víme, kde můžeme najít ukázky komponent a další informace o těchto komponentách. Není pro nás problém stáhnout si správnou verzi knihovny jQuery UI a sestavit si vlastní balík. V neposlední řadě známe obsah běžného balíku knihovny jQuery UI.

Motivy vzhledu

Knihovna jQuery UI nabízí spoustu předem vytvořených motivů vzhledu. Díky těmto motivům vzhledu můžeme snadno měnit vzhled celé své webové aplikace používající komponenty knihovny jQuery UI. Jestliže se nám žádný z předem vytvořených vzhledů nelíbí, můžeme si dokonce vytvářet své vlastní motivy vzhledu.

V této kapitole se seznámíte s:

- pokročilými nastaveními motivů vzhledu v nástroji pro sestavení balíku knihovny jQuery UI,
- nástrojem ThemeRoller,
- zásuvným modulem Theme Switcher,
- frameworkem knihovny jQuery UI pro kaskádové styly.

Pokročilá nastavení motivů vzhledu

Možná jste si v minulé kapitole všimli, že když vybíráte motiv vzhledu v nástroji pro sestavení balíku knihovny jQuery UI, v dolní části sekce **Theme** se nachází odkaz **Advanced Theme Settings**. Pod tímto odkazem se skrývají pokročilá nastavení motivů vzhledu. Ve skutečnosti jsou dvě a obě si popíšeme v této části kapitoly.

Obrázek 2.1: Část pro výběr motivu vzhledu s pokročilými nastaveními

Název složky s motivem vzhledu

Prvním z pokročilých nastavení je nastavení názvu složky s motivem vzhledu (pole **Theme Folder Name**). Ačkoliv se v uživatelském rozhraní nachází až na druhém místě, z hlediska pochopení významu obou zde popisovaných pokročilých nastavení je lepší začít popisem právě tohoto nastavení. Druhé pokročilé nastavení totiž ovlivňuje rovněž nastavení názvu složky s motivem vzhledu.

Jestli si vybavujete z předchozí kapitoly, že složka `/css/{název-motivu-vzhledu}/` ve staženém balíku knihovny jQuery UI obsahuje soubory motivu vzhledu, není to tak docela pravda. Ve skutečnosti platí, že soubory motivu vzhledu se nacházejí ve složce `/css/{hodnota-pole-Theme-Folder-Name}/`. Hodnota pole **Theme Folder Name** tudíž ovlivňuje cílové umístění motivu vzhledu v balíku knihovny jQuery UI. Výchozí hodnotou tohoto pole je název motivu vzhledu, ale všechna písmena jsou malá a mezery se nahrazují spojovníkem (-). Ponechat hodnotu tohoto pole prázdnou není vhodné, protože výsledná složka bude mít poměrně nesmyslný název `/css/ui+lightness/`, ať vybereme jakýkoliv motiv vzhledu.

Oblast platnosti kaskádových stylů

Dalším z pokročilých nastavení motivů vzhledu je nastavení oblasti platnosti kaskádových stylů. Stačí, když do pole **CSS Scope** napíšeme selektor jazyka CSS a námi vybraný motiv vzhledu se uplatní jen pod elementy, kterých se tento selektor týká (to znamená na všechny jejich dceřiné elementy). Jestliže toto pole necháme prázdné, motiv vzhledu se aplikuje na všechny elementy na stránce (což je výchozí stav).

Všimněte si, že dokud nezadáte vlastní hodnotu do pole **Theme Folder Name**, pak se do něj kopíruje hodnota z pole **CSS Scope** v okamžiku, kdy do něj něco píšete, přičemž se ignorují některé speciální znaky a mezery se převádějí na spojovníky.

Ptáte se, k čemu jsou obě pokročilá nastavení dobrá? Díky těmto nastavením můžete mít více motivů vzhledu ve své webové aplikaci. To může mít dva důvody – buď potřebujete kombinovat více motivů vzhledu zároveň, nebo je chcete střídat.

Příklad přepínání motivů vzhledu s pomocí pokročilých nastavení

Nyní si ukážeme jednoduchý příklad dynamického střídání motivů vzhledu – tj. něco podobného, jako umožňují oficiální ukázky knihovny jQuery UI.

Základ pro tento příklad najdete ve složce */Základy příkladů/Kapitola02/Varianta01/* v archivu se zdrojovými kódy příkladů. Základ příkladu této kapitoly už obsahuje většinu kódu, jehož princip si vysvětlíme až v následujících kapitolách; zejména v kapitole „Navigace v obsahu stránky“.

Především nás zajímá složka *css* se třemi podsložkami – *dark-hive*, *start* a *ui-lightness*. Ty obsahují soubory stejnojmenných motivů vzhledu. Jednotlivé motivy vzhledu jsme získali tak, že jsme nejprve sestavili balík knihovny jQuery UI s motivem vzhledu Dark Hive, přičemž jsme do pole **CSS Scope** vyplnili selektor `.motiv-dark-hive` a do pole **Theme Folder Name** jsme napsali hodnotu `dark-hive`. Posléze jsme sestavili balík s motivem vzhledu Start a přitom jsme do pole **CSS Scope** zapsali selektor `.motiv-start` a do pole **Theme Folder Name** hodnotu `start`. Nakonec jsme sestavili balík s motivem vzhledu UI lightness, přičemž jsme do pole **CSS Scope** napsali selektor `.motiv-ui-lightness` a do pole **Theme Folder Name** hodnotu `ui-lightness`. Ze všech těchto balíků jsme zkopírovali obsah složky *css* do složky *css* v našem příkladu.

Složka *js* obsahuje pouze skripty knihoven jQuery a jQuery UI a soubor *hlavni.js*, v němž prozatím inicializujeme ovládací prvky knihovny jQuery UI.

V kořenovém adresáři tohoto jednoduchého příkladu se nachází jen soubor *index.html*, ve kterém si důležité elementy zvýrazníme tučně:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html
  PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="cs" lang="cs">
<head>
  ...
</head>
<body>
  <div id="hlavni">
    <h1>Přepínání motivů vzhledu</h1>
    <p>
      <label>
        Aktuální motiv vzhledu:
        <select id="seznam-motivu">
          <option value="motiv-dark-hive">Dark Hive</option>
          <option value="motiv-start">Start</option>
          <option value="motiv-ui-lightness">UI lightness</option>
        </select>
      </label>
    </p>
    ...
  </div>
```



```
</body>
</html>
```

Celá myšlenka tohoto příkladu spočívá v tom, že obalujícímu elementu `div` s identifikátorem `hlavni` přiřadíme třídu podle hodnoty atributu `value` vybraného prvku (`option`) v elementu `select` s identifikátorem `seznam-motivu`. Všimněte si, že tyto prvky mají hodnoty `motiv-dark-hive`, `motiv-start` a `motiv-ui-lightness`. Přiřadíme-li některou z nich jako třídu elementu s identifikátorem `hlavni`, který obaluje celou naši stránku, aplikuje se motiv vzhledu s odpovídající oblastí platnosti (`.motiv-dark-hive`, `.motiv-start` nebo `.motiv-ui-lightness`) na všechny komponenty knihovny jQuery UI uvnitř něj.

Začneme tak, že do souboru `hlavni.js` napíšeme níže uvedený zdrojový kód bezprostředně za řádek s kódem `$('#harmonika').accordion()`:

```
document.getElementById('seznam-motivu').onchange = nastavMotivVzhledu;
nastavMotivVzhledu();
```

Tímto kódem nastavujeme, že při výběru (změně) prvku ze seznamu s identifikátorem `seznam-motivu` chceme zavolat funkci `nastavMotivVzhledu()`. Na druhém z předchozích řádků tuto funkci voláme rovněž hned po načtení stránky (jinak by po spuštění naší stránky nebyl nastaven žádný motiv vzhledu). Jak už název funkce `nastavMotivVzhledu()` napovídá, nastavíme v ní motiv vzhledu knihovny jQuery UI, a protože tato funkce prozatím neexistuje, doplníme ji na úplný konec souboru `hlavni.js`:

```
function nastavMotivVzhledu() {
    var seznamMotivu = document.getElementById('seznam-motivu'),
        prvekVybranehoMotivu =
            seznamMotivu.options[seznamMotivu.selectedIndex],
        hodnotaVybranehoMotivu = prvekVybranehoMotivu.value;

    document.getElementById('hlavni').className = hodnotaVybranehoMotivu;
}
```

V předchozím kódu nejdříve hledáme element seznamu motivů, potom element jeho vybraného prvku a nakonec ukládáme hodnotu vybraného prvku do proměnné `hodnotaVybranehoMotivu`. Řekli jsme si, že tuto hodnotu přiřadíme jako třídu elementu s identifikátorem `hlavni`, což také děláme příkazem:

```
document.getElementById('hlavni').className = hodnotaVybranehoMotivu;
```

A náš jednoduchý příklad je hotový. Otestujte jej: zkuste vybírat různé motivy vzhledu a uvidíte, že se dynamicky mění vzhled stránky. Výsledek svého snažení si můžete také prohlédnout na obrázku 2.2. Kompletní verzi tohoto příkladu najdete rovněž v archivu se zdrojovými kódy, a to ve složce `/Kompletní příklady/Kapitola02/Varianta01/`.

Nástroj ThemeRoller

Jestliže chcete upravit vzhled komponent knihovny jQuery UI, nemusíte psát vlastní šablony stylů v jazyce CSS – na oficiálních stránkách knihovny jQuery UI najdete pod adresou <http://jqueryui.com/themeroller/> velmi užitečný nástroj ThemeRoller. S jeho pomocí můžete prohlížet standardní motivy vzhledu, upravovat je a vytvářet zcela nové motivy vzhledu.



Obrázek 2.2: Dynamická změna motivu vzhledu

Galerie motivů vzhledu

Nástroj ThemeRoller umožňuje prohlížet si standardní motivy vzhledu na kartě **Galerie**. Tuto část budeme nazývat **galerie**. V době psaní této knihy se v galerii tohoto nástroje vyskytovalo 24 předdefinovaných motivů vzhledu. U každého motivu vzhledu se nachází náhled na to, jak budou vypadat komponenty knihovny jQuery UI s tímto motivem vzhledu. Snímek galerie motivů vzhledu najdete na obrázku 2.3.



Obrázek 2.3: Galerie motivů vzhledu

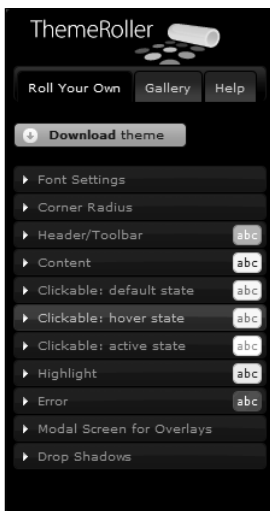
Jak je patrné, pod názvem každého motivu vzhledu je možné najít dvě tlačítka – tlačítko **Download** a tlačítko **Edit**.

Jestliže klepneme na tlačítko **Download**, přeneseme se na již známou stránku s nástrojem pro sestavení balíku knihovny jQuery UI ke stažení. Klepnutí na toto tlačítko má stejný efekt jako výběr motivu vzhledu v části **Theme** nástroje pro sestavení balíku.

Pokud klepneme na tlačítko **Edit**, přesuneme se na kartu **Roll Your Own** nástroje ThemeRoller, na které můžeme příslušný motiv vzhledu upravit. Úprava existujícího motivu vzhledu se shoduje s tvorbou vlastního motivu vzhledu. Tímto tématem se budeme zabývat v následující části kapitoly.

Vlastní motiv vzhledu

V případě, že budeme chtít vytvořit vlastní motiv vzhledu (nebo upravit některý současný), přijde nám vhod karta **Roll Your Own**. Na této kartě můžeme najít ovládací prvky pro nastavení hodnot jednotlivým vlastnostem komponent knihovny jQuery UI. Všechna nastavení na této kartě jsou rozdělená do přehledných skupin, jak můžete vidět na obrázku 2.4.



Obrázek 2.4: Uživatelské rozhraní nástroje ThemeRoller pro tvorbu vlastního motivu vzhledu

Skupiny nastavení jsou následující:

- skupina **Font Settings** – nastavení písma,
- skupina **Corner Radius** – nastavení poloměru zaoblení rohů,
- skupina **Header/Toolbar** – nastavení záhlaví a nástrojové lišty,
- skupina **Content** – nastavení obsahu,
- skupina **Clickable: default state** – nastavení prvků s možností klepnutí, které se nachází ve výchozím stavu (tj. nejsou aktivní, ani nad ně uživatel nepřesunul ukazatel myši),
- skupina **Clickable: hover state** – nastavení prvků s možností klepnutí, nad které uživatel právě přesunul ukazatel myši,
- skupina **Clickable: active state** – nastavení aktivních prvků s možností klepnutí,
- skupina **Highlight** – nastavení zvýraznění,

- skupina **Error** – nastavení chybových zpráv,
- skupina **Modal Screen for Overlays** – nastavení modálních překryvných vrstev,
- skupina **Drop Shadows** – nastavení vrženého stínu.

Velkou výhodou nástroje ThemeRoller při tvorbě vlastního motivu vzhledu je, že si nemusíme vytvářet žádnou testovací aplikaci, abychom viděli, jak naše změny vypadají. Cokoliv, co změníme na kartě **Roll Your Own**, se okamžitě projeví v pravé části nástroje Theme Roller, ve které jsou všechny viditelné komponenty knihovny jQuery UI.

Projdeme si tudíž jednotlivá nastavení a zkusíme si s nimi pohrát. Jako základ použijeme nový vlastní motiv vzhledu.

Skupina Font Settings

Do pole **Family** zadáváte rodiny písma, které jistě znáte z jazyka CSS. Můžeme zadat více rodin, přičemž jednotlivé rodiny písma oddělujeme čárkou. Webový prohlížeč pak hledá rodiny písma zleva doprava a aplikuje z nich první, kterou najde v počítači uživatele. Aby se změna projevila v náhledech komponent, musíme nakonec stisknout klávesu *Enter* nebo zaměřit jiné pole. Zkuste kupříkladu zadat řetězec *Tahoma, Arial, sans-serif* a potom stiskněte klávesu *Enter*. Za předpokladu, že máte ve svém počítači nainstalované písmo Tahoma, měli byste jej vidět v pravé části na všech textových řetězcích.

Ze seznamu **Weight** vybíráme šířku (neboli tučnost) písma. Výběr položky **bold** má za následek, že všechna záhlaví komponent získají tučné písmo. Položka **normal** odpovídá běžnému písmu. Vyberte například položku **bold**; zjistíte, že změna se projeví okamžitě (nemusíte tedy mačkat žádnou klávesu ani přecházet na jiné pole).

Do pole **Size** vyplňujeme velikost písma. Hodnotou může být jakákoliv hodnota (číslo včetně jednotky), kterou lze zapsat do definice vlastnosti `font-size` z jazyka CSS. V případě, že se rozhodnete změnit tuto hodnotu, nezapomeňte stisknout klávesu *Enter*.

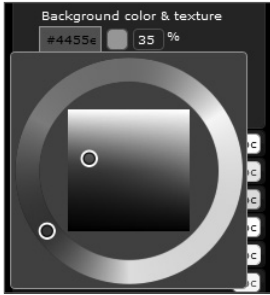
Skupina Corner Radius

Ve skupině **Corner Radius** najdete jediné pole – **Corners**. Do něj zapisujeme poloměr oblých rohů komponent knihovny jQuery UI. Podobně jako u předchozího pole **Size** musíme zadávat typickou hodnotu (s jednotkou) velikosti z jazyka CSS. Změna se opět neprojevuje ihned. Napište kupříkladu hodnotu `10px` a stiskněte klávesu *Enter*.

Skupina Header/Toolbar

Na začátku této skupiny se nachází část označená **Background color & texture**, jež se ve skutečnosti skládá ze tří samostatných ovládacích prvků.

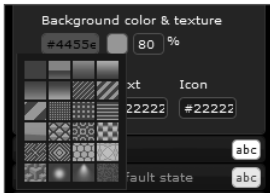
Do prvního pole vkládáme barvu jazyka CSS. Toto pole je poměrně „inteligentní“ a nedovolí nám zadat neplatnou hodnotu. Navíc – jakmile zaměříme toto pole, objeví se nástroj pro výběr barvy, z něhož můžeme vybrat barvu přímo. Jak tento nástroj vypadá, ukazuje obrázek 2.5.



Obrázek 2.5: Nástroj pro výběr barvy

Zkuste zadat například hodnotu #4455ee. Znak # nemusíte psát, jelikož se do tohoto pole doplní automaticky. Povšimněte si, že obsah daného pole se automaticky obarví vybranou barvou; pokud ale chcete, aby se změna projevila i na ukázkových komponentách, stiskněte klávesu *Enter*.

Druhým ovládacím prvkem je obrázkový rozevírací seznam. Z tohoto seznamu vybíráme texturu záhlaví. Tento ovládací prvek je rovněž vcelku unikátní, takže si jej ukážeme na obrázku 2.6.



Obrázek 2.6: Obrázkový rozevírací seznam s texturami

Vyberte kupříkladu texturu **glass**. Název textury se objeví, když chvíli ponecháte ukazatel myši nad danou texturou. V době psaní této knihy se jednalo o druhou texturu zleva v prvním řádku.

Posledním ovládacím prvkem části **Background color & texture** je textové pole, do kterého vkládáme průhlednost textury vyjádřenou procentuálně jako číslo od 0 do 100. Hodnota 0 znamená absolutní průhlednost textury (to znamená, že texturu nelze vůbec vidět) a hodnota 100 maximální neprůhlednost, která ovšem v případě textury neznamená, že by se vůbec neprojevila barva pozadí. Mohli bychom si tedy toto nastavení představit spíše jako viditelnost textury, která vzrůstá v souladu s uvedeným procentem. Vyzkoušejte si zadat výraznější texturu – například číslo 85. Změna se opět neobjeví okamžitě.

Do pole **Border** vkládáme barvu rámečku okolo záhlaví. U tohoto pole je také k dispozici nástroj pro výběr barvy. Zadejte kupříkladu barvu #4455ee, což je stejná barva, jakou jste nastavili pro pozadí záhlaví.

V poli **Text** uvádíme barvu textu záhlaví, přičemž můžeme použít i nástroj pro výběr barvy. Zkuste vyplnit kupříkladu barvu #eeeeee.

Do pole **Icon** zadáváme barvu ikon v záhlaví. Samozřejmě můžeme vybrat barvu také v nástroji pro výběr barvy. Nastavte například barvu #111111.

Skupina Content

Skupina **Content** nabízí prvky uživatelského rozhraní shodné se skupinou **Header/Toolbar**.

V části **Background color & texture** nastavujeme barvu pozadí, typ a průhlednost textury pro obsahovou část komponent knihovny jQuery UI. Zadejte kupříkladu barvu pozadí #fafafa a u textury nechejte výchozí hodnoty.

V poli **Border** nastavujeme barvu rámečku okolo obsahové části komponenty, v poli **Text** barvu textu a pomocí pole **Icon** barvu ikony v obsahové části. Do pole **Border** zkuste zadat například barvu #4455ee a v ostatních polích nechejte výchozí hodnoty.

Skupiny Clickable: default state, Clickable: hover state a Clickable: active state

Skupiny, jejichž název začíná předponou **Clickable**, poskytují stejnou sadu konfiguračních prvků jako skupina **Header/Toolbar** nebo skupina **Content**.

V části **Background color & texture** určujeme barvu pozadí, typ a průhlednost textury pro prvky s možností klepnutí. Ve skupině **Clickable: default state** se jedná o prvky v běžném stavu, ve skupině **Clickable: hover state** o prvky, nad které uživatel právě přesunul ukazatel myši, a ve skupině **Clickable: active state** o aktivní prvky. Ve skupině **Clickable: hover state** nastavte například hodnotu #4455ee jako barvu pozadí.

V poli **Border** opět nastavujeme barvu rámečku okolo prvku s možností klepnutí, v poli **Text** barvu textu a pomocí pole **Icon** barvu ikony prvku s možností klepnutí. Do pole **Text** ve skupině **Clickable: hover state** zkuste zadat kupříkladu hodnotu #eeeeee jako barvu pozadí.

Skupina Highlight

Zde se nachází opět stejná sada prvků jako u předchozích skupin. S jejich pomocí nastavujeme vzhled upozornění.

Vložte například barvu #4455ee jako barvu rámečku do pole **Border**. Všimněte si, že v kalendáři je jedno datum rovněž označené jako upozornění, a ne jako aktivní prvek s možností klepnutí (viz obrázek 2.7).

Skupina Error

Opět naprosto stejná sada prvků, s níž nastavujeme vzhled chybové zprávy.

Skupina Modal Screen for Overlays

Na začátku této skupiny se znovu nachází nám už důvěrně známá část **Background color & texture**. V této části nastavujeme barvu pozadí, typ a průhlednost textury pro modální překryvnou vrstvu. Zadejte kupříkladu barvu #4455ee jako barvu pozadí.

Konečně se po delší době setkáváme s novým typem pole – s polem **Overlay Opacity**. Hodnota tohoto pole určuje průhlednost modální překryvné vrstvy vyjádřenou procentuálně číslem od 0 do 100. Tentokrát už se jedná o běžnou průhlednost, jakou známe z jazyka CSS – hodnota 0 znamená absolutní průhlednost (překryvná vrstva je skrytá) a hodnota

100 absolutní neprůhlednost (tj. není vidět obsah pod ní). Do pole **Overlay Opacity** vložte například hodnotu 50.

Skupina Drop Shadows

Ve skupině **Drop Shadows** specifikueme vzhled vrženého stínu. Na začátku této skupiny je opět část **Background color & texture**. V ní neděláme nic jiného, než že nastavujeme barvu pozadí, typ a průhlednost textury vrženého stínu. Zkuste zadat jako barvu pozadí vrženého stínu hodnotu #666666.

Prostřednictvím pole **Shadow Opacity** definujeme průhlednost vrženého stínu. Zase je možné zadat číslo od 0 (absolutní průhlednost) do 100 (absolutní neprůhlednost). Vyzkoušejte například hodnotu 65 pro výraznější vržený stín.

Pole **Shadow Thickness** reprezentuje tloušťku vrženého stínu, kterou zadáváme jako velikost v jazyce CSS (včetně jednotky). Vyplňte kupříkladu hodnotu 6px a povšimněte si, že vržený stín nyní vypadá, jako by se světelný zdroj nacházel v pravém dolním rohu, což není moc obvyklé – běžně efekt vrženého stínu vypadá, jako by někdo umístil světelný zdroj do horního levého rohu. To můžeme ovlivnit pomocí následujících polí.

Pole **Top Offset** a **Left Offset** určují pořadí svislé a vodorovné posunutí vrženého stínu (vyjádřené opět jako velikost včetně jednotky). Jednoduše si lze toto posunutí představit jako pozici levého horního rohu vrženého stínu relativně vzhledem k samotné komponentě knihovny jQuery UI. Pokud zadáváme kladná čísla, horní levý roh vrženého stínu se bude nacházet vpravo pod horním levým rohem komponenty. Naopak – se zápornými čísly se bude horní levý roh vrženého stínu nacházet vlevo nad horním levým rohem komponenty. Zkuste zadat do obou polí například zápornou hodnotu -2px.

V poli **Corners** nastavujeme poloměr kulatých rohů vrženého stínu jako číslo s jednotkou. Je vhodné zadávat o něco větší hodnotu než ve stejnojmenném poli ve skupině **Corner Radius**, aby vržený stín vypadal skutečněji. Vložte sem kupříkladu hodnotu 14px.

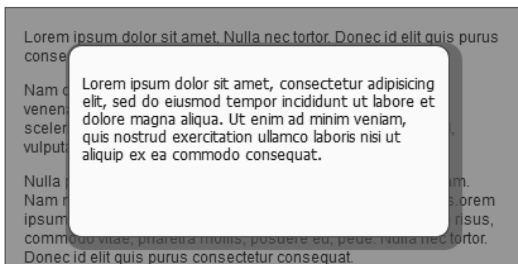
Stažení vlastního motivu vzhledu

Až dokončíte veškerá nastavení, můžete si ještě jednou prohlédnout výsledek své práce. Jestliže jste postupovali podle tipů v této části kapitoly, měl by váš motiv vzhledu vypadat podobně jako na obrázcích 2.7 a 2.8.



Obrázek 2.7: Kalendář s vlastním motivem vzhledu

Pokud jste s výsledkem spokojeni, klepnutím na tlačítko **Download theme** můžete přejít ke stažení svého motivu vzhledu. Tato akce vás přenese do nástroje pro sestavení balíku knihovny jQuery UI ke stažení, přičemž už budete mít vybraný svůj motiv vzhledu, jenž nese název Custom Theme.



Obrázek 2.8: Modální překryvná vrstva a vržený stín

Vývojářský nástroj pro prohlížeč Firefox

Jestliže používáte webový prohlížeč Firefox, na adrese <http://jqueryui.com/themeroller/developertool/> najdete vývojářskou verzi nástroje ThemeRoller pro tento prohlížeč. Tento nástroj má podobu okna, které je možné otevřít na jakémkoliv stránce. Jednoduše si jej můžete prohlédnout tak, že klepnete na tlačítko **ThemeRoller Dev Tool** na uvedené adrese. Pokud se vám tento nástroj zalíbí, můžete přesunout dané tlačítko do záložek.

Zásuvný modul Theme Switcher



UPOZORNĚNÍ

Tuto část kapitoly byste měli přeskočit a vrátit se k ní, až si přečtete kapitolu „Navigace v obsahu stránky“. Současně s tím si můžete vyzkoušet své znalosti, které v ní získáte – uvidíte, že vám po přečtení této kapitoly bude tento příklad o dost jasnější, přestože je i teď velmi intuitivní.

Nyní se vrátíme k našemu příkladu z této kapitoly. Líbí se vám galerie standardních motivů vzhledu nástroje ThemeRoller? Teď si představte, že byste mohli vzít už vytvořený rozevírací seznam s náhledy standardních motivů vzhledu (tedy podobné uživatelské rozhraní jako v galerii nástroje ThemeRoller) a vložit jej do vlastních stránek, přičemž výběr motivu vzhledu z tohoto seznamu by ovlivnil vzhled všech komponent knihovny jQuery UI na stránce. Představte si, že je to možné, a nejen to – je to dokonce jednodušší než naše první řešení dynamického přepínání motivů vzhledu. Existuje totiž zásuvný modul Theme Switcher, který však není běžnou součástí knihovny jQuery UI. Skript tohoto zásuvného modulu lze stáhnout z adresy <http://jqueryui.com/themeroller/themeswitchertool/>. Nachází se však rovněž v archivu se zdrojovými kódy příkladů k této knize – ve složce */Zásuvné moduly/Theme Switcher/* a také ve složce tohoto příkladu.

Oficiální dokumentaci k tomuto zásuvnému modulu najdete na adrese <http://jqueryui.com/docs/Theming/ThemeSwitcher>.

Nyní si tedy vysvětlíme druhý způsob řešení našeho příkladu z této kapitoly, který však neodpovídá prvnímu zadání zcela přesně, jak si ukážeme na konci této části kapitoly. Základ tohoto příkladu najdete ve složce */Základy příkladů/Kapitola02/Varianta02/*.

Oproti první variantě tohoto příkladu neobsahuje složka *css* složky s názvy motivů vzhledu, ale složku */themes/base/*. Tato složka je pouze kopií složky */development-bundle/themes/base/* z balíku knihovny jQuery UI. Složka *js* našeho příkladu obsahuje navíc soubor *theme-switcher.js*, což je zmíněný skript zásuvného modulu Theme Switcher. Soubor *index.html* se také lehce změnil. Důležitou část tohoto souboru si zvýrazníme tučně:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html
  PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="cs" lang="cs">
<head>
  ...
</head>
<body>
  <h1>Přepínání motivů vzhledu (druhá varianta)</h1>
  <p>
    <div id="seznam-motivu"></div>
  </p>
  ...
</body>
</html>
```

Jak je patrné, momentálně nás zajímá jen element *div* s identifikátorem *seznam-motivu*. Z toho totiž vytvoříme rozevírací seznam s náhledy motivů vzhledu. Začneme tak, že přidáme do souboru *index.html* šablonu stylů a skript jazyka JavaScript pro zásuvný modul Theme Switcher:

```
...
<head>
  ...
  <link type="text/css" href="css/themes/base/jquery.ui.all.css"
    rel="stylesheet" />
  <script type="text/javascript" src="js/jquery.js"></script>
  <script type="text/javascript" src="js/jquery-ui.js"></script>
  <script type="text/javascript" src="js/theme-switcher.js"></script>
  <script type="text/javascript" src="js/hlavni.js"></script>
</head>
...
```

Vytvoření zásuvného modulu Theme Switcher

Zásuvný modul inicializujeme tím způsobem, že vybereme element a zavoláme na něj metodu *themeswitcher()*. Zde je příklad:

```
$('#prepinač-motivu-vzhledu').themeswitcher();
```

V našem příkladu chceme aplikovat zásuvný modul na element s identifikátorem seznam-motivu, proto přidáme následující zdrojový kód za řádek s kódem \$('#harmonika').accordion() v souboru *hlavni.js*:

```
$('#seznam-motivu').themeswitcher();
```

Možnosti nastavení

Zásuvný modul Theme Switcher nemá žádné vlastní metody (tj. ani metodu `option`), takže jediným vhodným postupem je konfigurovat ho přímo při inicializaci. Například takto:

```
$('#prepinač-motivu-vzhledu').themeswitcher({
  height: 400
});
```

V předchozím kódu nastavujeme možnosti nastavení `height` hodnotu 400 pro zásuvný modul Theme Switcher na elementu s identifikátorem `prepinač-motivu-vzhledu`. Díky tomu získá rozevírací seznam s motivy vzhledu výšku 400 pixelů.

Přehled všech možností nastavení zásuvného modulu Theme Switcher podle dokumentace zásuvného modulu Theme Switcher najdete v tabulce D.1 v příloze „Zásuvné moduly“.

Možnost nastavení `buttonPreText`

Možnost nastavení `buttonPreText` specifikuje text otevíracího tlačítka, který předchází názvu vybraného motivu vzhledu poté, co uživatel nějaký vybere.

V předchozí variantě našeho příkladu jsme před rozevírací seznam s motivy vzhledu vkládali popisek „Aktuální motiv vzhledu“, proto nyní nastavíme stejný text. Rozšíříme volání metody `themeswitcher()` v souboru *hlavni.js* o níže uvedený kód:

```
$('#seznam-motivu').themeswitcher({
  buttonPreText: 'Aktuální motiv vzhledu: '
});
```

Možnost nastavení `initialText`

Hodnotou možnosti nastavení `initialText` je výchozí text, který se zobrazuje uvnitř tlačítka pro otevření rozevíracího seznamu, dokud uživatel nevybere nějaký motiv vzhledu.

Jestliže si uživatel zatím nevybral žádný motiv vzhledu, měli bychom mu sdělit, aby tak učinil. Doplňme tudíž inicializaci zásuvného modulu ThemeSwitcher takto:

```
$('#seznam-motivu').themeswitcher({
  buttonPreText: 'Aktuální motiv vzhledu: ',
  initialText: 'Vyberte si motiv vzhledu'
});
```

Tímto řádkem kódu jsme přiřadili hodnotu 'Vyberte si motiv vzhledu' možnosti nastavení `initialText`.

Možnost nastavení `width`

Možnost nastavení `width` reprezentuje šířku rozevíracího seznamu (a také tlačítka pro jeho otevření).

Kdybyste se teď pokusili otestovat tento příklad, zjistili byste, že kvůli delšímu textu, který se zobrazuje v tlačítku pro otevření rozevíracího seznamu, se tento text zalamuje a nevypadá to příliš dobře. Tento problém vyřešíme přidáním následujícího nastavení:

```
$('#seznam-motivu').themeswitcher({
    buttonPreText: 'Aktuální motiv vzhledu: ',
    initialText: 'Vyberte si motiv vzhledu',
    width: 230
});
```

Nastavíme možnosti nastavení `width` hodnotu 230, čímž rozšíříme otevírací tlačítko (a rozevírací seznam) tak, že se do něj náš text pohodlně vejde.

Shrnutí druhé varianty příkladu

Úspěšně jsme dokončili druhou variantu našeho příkladu. Jak už jsme si naznačili dříve, obě varianty nejsou zcela totožné. Musíme si uvědomit, že druhá varianta nám umožňuje vybírat si pouze standardní motivy vzhledu. V první variantě tohoto příkladu jsme mohli bez problémů použít vlastní motiv vzhledu, a dokonce i více motivů vzhledu současně pro různé části stránky.

Pokud jste postupovali podle pokynů v této části kapitoly, měla by vaše stránka vypadat jako na obrázku 2.9.



Obrázek 2.9: Přepínání motivů vzhledu pomocí zásuvného modulu Theme Switcher

Kompletní verzi tohoto příkladu najdete v archivu se zdrojovými kódy příkladů, a to ve složce `/Kompletní příklady/Kapitola02/Varianta02/`.

Framework knihovny jQuery UI pro kaskádové styly

V této části kapitoly se podíváme na to, jak můžeme ručně vytvořit vlastní motiv vzhledu nebo editovat některý existující. Mnohem pohodlnější je samozřejmě použít nástroj ThemeRoller, který jsme si popsali dříve v této kapitole. Problém však spočívá v tom, že se omezujeme na konfigurační schopnosti tohoto nástroje. Jazyk CSS je přece jenom mnohem flexibilnější a můžeme s ním vytvořit téměř vše, co nás napadne.

Jak už to tak bývá, plná síla jazyka CSS má své výhody, ale také nevýhody. Výhodou je jistě to, že si můžeme dělat, co se nám zlíbí. Tato skutečnost je ale současně nevýhodou. Pokud se nebudeme držet určitých zásad, budeme toho litovat při budoucích úpravách nebo aktualizaci na novější verzi knihovny jQuery UI. Nyní si tedy vysvětlíme:

- jak autoři knihovny jQuery UI uspořádali pravidla stylů v motivech vzhledu do šablon stylů,
- jaké obecné třídy frameworku kaskádových stylů jsou k dispozici,
- jakými zásadami bychom se měli držet, pokud budeme vytvářet vlastní motiv vzhledu od začátku.

Šablony stylů motivu vzhledu

Každý motiv vzhledu se skládá z několika samostatných šablon stylů (samostatných souborů), které se ve výsledku kombinují do jedné velké šablony stylů. Jistě jste si už všimli, že v balíku s knihovnou jQuery UI se nachází spousta šablon stylů ve složkách `/development-bundle/themes/base/` a `/development-bundle/themes/{název motivu vzhledu}/`. Všechny tyto šablony stylů společně obsahují definice všech tříd frameworku knihovny jQuery UI pro kaskádové styly. Některé z těchto tříd jsou obecné a nacházejí uplatnění u všech komponent knihovny jQuery UI (případně včetně našich vlastních) a jiné jsou specifické pro jednotlivé komponenty.

Teď si popíšeme jednotlivé šablony stylů, které tvoří motiv vzhledu:

- `jquery-ui-{verze knihovny}.custom.css` – sloučená šablona stylů, kterou obvykle vkládáme do svých webových aplikací.
- `jquery.ui.all.css` – tato šablona stylů pouze importuje všechny ostatní šablony stylů (ve skutečnosti importuje šablonu stylů `jquery.ui.theme.css` a šablonu stylů `jquery.ui.base.css`, která potom importuje všechny ostatní šablony stylů).
- `jquery.ui.theme.css` – tato šablona stylů obsahuje definice obecných tříd frameworku knihovny jQuery UI pro kaskádové styly. Pravidla stylů v ní uvedená definují pouze základní vzhled (barvu textu, barvu pozadí, ikony apod.), a nikoliv rozvržení komponent a jejich rozměry. Editaci této šablony stylů lze tudíž považovat za velmi bezpečnou, protože nebude mít negativní dopad na žádné komponenty knihovny jQuery UI ani na budoucí aktualizace.
- `jquery.ui.base.css` – tato šablona stylů importuje šablonu stylů `jquery.ui.core.css` a všechny šablony stylů specifické pro komponenty knihovny jQuery UI.

- *jquery.ui.core.css* – tato šablona stylů obsahuje další pomocné obecné třídy, které se nenacházejí v šabloně stylů *jquery.ui.theme.css*. Zapamatujte si, že šablony stylů *jquery.ui.theme.css* a *jquery.ui.core.css* společně obsahují definice obecných tříd frameworku knihovny jQuery UI pro kaskádové styly, avšak šablona stylů *jquery.ui.core.css* obsahuje spíše ty definice, u nichž se předpokládá, že nebudeme mít potřebu je měnit.
- *jquery.ui.{název komponenty}.css* (například *jquery.ui.accordion.css*) – jedná se o specifické šablony stylů pro jednotlivé komponenty knihovny jQuery UI. Je vcelku logické, že tyto šablony stylů obsahují specifická pravidla stylů, a to především pravidla ovlivňující umístění jednotlivých prvků, okraje apod. Úprava těchto pravidel se nedoporučuje, protože může negativně ovlivnit funkčnost těchto komponent. Pokud se chystáte vytvořit vlastní komponentu knihovny jQuery UI (zásuvný modul knihovny jQuery UI), měli byste se řídit zásadami, které si uvedeme dále v této kapitole.

Obecné třídy frameworku knihovny jQuery UI pro kaskádové styly

Jestliže chcete vytvářet vlastní zásuvné moduly pro knihovnu jQuery UI, měli byste používat obecné třídy frameworku knihovny jQuery UI pro kaskádové styly. Důvod je jednoduchý – usnadníte ostatním vývojářům (a koneckonců i sobě) tvorbu motivů vzhledu pro váš zásuvný modul. Praktický příklad si ukážeme v kapitole „Navigace v obsahu stránky“, v níž sice nebudeme vytvářet vlastní zásuvný modul, ale alespoň jednoduše nastylujeme vlastní prvky na našich webových stránkách.

Dále v této části kapitoly si uvedeme přeložený přehled obecných tříd frameworku knihovny jQuery UI pro kaskádové styly. Pokud chcete držet krok s vývojem knihovny jQuery UI, měli byste sledovat originální dokumentaci na adrese <http://jqueryui.com/docs/Theming/API>.

Třídy pro podporu rozvržení

- Třída *ui-helper-hidden* – nastaví elementům hodnotu *none* pro vlastnost *display*. Jednoduše řečeno skryje elementy.
- Třída *ui-helper-hidden-accessible* – podobně jako třída *ui-helper-hidden* skryje elementy, ale tentokrát s ohledem na přístupnost (tj. absolutním umístěním mimo viditelnou oblast stránky).
- Třída *ui-helper-reset* – vynuluje definice základních vlastností prvků uživatelského rozhraní – například vnitřní okraje, vnější okraje, styl seznamu atd.
- Třída *ui-helper-clearfix* – přiřadíme-li tuto třídu elementu, který obsahuje plovoucí elementy, automaticky přidá za tento element neviditelnou „zarážku“ pro plovoucí elementy.
- Třída *ui-helper-zfix* – aplikuje „opravné“ kaskádové styly na elementy *iframe*, pokud je potřebujeme používat v kombinaci s překryvnými vrstvami.

Třídy pro obalující elementy ovládacích prvků

- Třída *ui-widget* – tuto třídu přiřazujeme elementu, jenž obaluje celý ovládací prvek. Tato třída nastavuje rodinu a velikost písma. Kromě toho nastavuje stejnou rodinu

písma a velikost písma 1em všem dceřiným formulářovým prvkům, čímž řeší problémy s dědičností ve webových prohlížečích pro operační systém Windows.

- Třída `ui-widger-header` – tuto třídu přiřazujeme elementu, který obaluje záhlaví ovládacího prvku. Tato třída aplikuje své definice vlastností na tento obalující element a všechny jeho dceřiné textové uzly, odkazy a ikony.
- Třída `ui-widger-content` – tuto třídu přiřazujeme elementu, jenž obaluje obsahovou část ovládacího prvku. Tato třída aplikuje své definice vlastností na tento obalující element a všechny jeho dceřiné textové uzly, odkazy a ikony. Tuto třídu je možné přidělit jak rodičovskému elementu záhlaví, tak jeho sourozeneckému elementu.

Interakční třídy

- Třída `ui-state-default` – třída pro elementy s možností klepnutí ve výchozím stavu. Aplikuje své definice vlastností na daný element a jeho dceřiné textové uzly, odkazy a ikony.
- Třída `ui-state-hover` – třída pro elementy s možností klepnutí, nad které uživatel právě přesunul ukazatel myši. Aplikuje své definice vlastností na daný element a jeho dceřiné textové uzly, odkazy a ikony.
- Třída `ui-state-focus` – třída pro elementy s možností klepnutí, které získaly zaměření. Aplikuje své definice vlastností na daný element a jeho dceřiné textové uzly, odkazy a ikony.
- Třída `ui-state-active` – třída pro aktivní elementy s možností klepnutí. Aplikuje své definice vlastností na daný element a jeho dceřiné textové uzly, odkazy a ikony.
- Třída `ui-state-highlight` – třída pro elementy upozornění a vybrané elementy. Aplikuje své definice vlastností na daný element a jeho dceřiné textové uzly, odkazy a ikony.
- Třída `ui-state-error` – třída pro obalující elementy chybových zpráv. Aplikuje své definice vlastností na daný element a jeho dceřiné textové uzly, odkazy a ikony.
- Třída `ui-state-error-text` – dodatečná třída, která aplikuje jen barvu textu, ale ne pozadí chybové zprávy. Tuto třídu můžeme použít kupříkladu na popisky formulářových prvků. Dále tato třída aplikuje barvu ikony na dceřiné ikony.
- Třída `ui-state-disabled` – tato třída nastavuje částečnou průhlednost na zakázané elementy. Tyto elementy už musejí mít vlastní vzhled. Navíc je nutné poznamenat, že tato třída nemá žádný vliv na funkčnost – mění pouze vzhled elementu, ale pokud chceme zakázat například klepnutí na něj, musíme si poradit jinak.
- Třída `ui-priority-primary` – pokud potřebujeme hierarchii tlačítek, prostřednictvím této třídy označíme tlačítko, které upřednostňujeme více. Aplikuje tučné písmo.
- Třída `ui-priority-secondary` – jestliže potřebujeme hierarchii tlačítek, pomocí této třídy označíme tlačítko, které upřednostňujeme méně. Aplikuje normální šířku (tučnost) písma a částečnou průhlednost.

Třídy ikon

- Třída `ui-icon` – základní třída pro element ikony. Dělá z elementu čtvercový blok o délce strany 16 pixelů, skrývá jeho vnitřní text a nastavuje vhodný sprajtový obrázek jako obrázek na pozadí. Tento element může získat různé sprajtové obrázky v závis-

losti na stavu svého rodičovského elementu. Například element s třídou `ui-icon` uvnitř elementu s třídou `ui-state-default` se obarví podle barvy ikony definované v třídě `ui-state-default`.

- Třída `ui-icon-{typ}-{podtyp}-{směr}` – pokud přiřadíme elementu třídu `ui-icon`, můžeme mu přidat také tuto dodatečnou třídu. Podtyp a směr ikony jsou k dispozici jen u některých ikon. Směr ikony specifikujeme pomocí anglických zkratk světových stran – `n` (sever), `ne` (severovýchod), `e` (východ), `se` (jihovýchod), `s` (jih), `sw` (jihozápad), `w` (západ) a `nw` (severozápad). Například třídou `ui-icon-triangle-1-e` nastavíme ikonu s trojúhelníkem směřujícím doprava. Nástroj ThemeRoller obsahuje náhled všech ikon frameworku knihovny jQuery UI pro kaskádové styly – když přesunete ukazatel myši nad ikonu, objeví se její úplný název.

Třídy oblých rohů

- Třída `ui-corner-all` – zaoblí všechny rohy elementu.
- Třída `ui-corner-tl` – zaoblí levý horní roh elementu.
- Třída `ui-corner-tr` – zaoblí pravý horní roh elementu.
- Třída `ui-corner-bl` – zaoblí levý dolní roh elementu.
- Třída `ui-corner-br` – zaoblí pravý dolní roh elementu.
- Třída `ui-corner-top` – zaoblí oba horní rohy elementu.
- Třída `ui-corner-bottom` – zaoblí oba dolní rohy elementu.
- Třída `ui-corner-left` – zaoblí oba levé rohy elementu.
- Třída `ui-corner-right` – zaoblí oba pravé rohy elementu.

Třídy pro vržené stíny a překryvné vrstvy

- Třída `ui-widget-overlay` – udělá z elementu překryvnou vrstvu; to znamená, že ho roztáhne přes celou obrazovku, nastaví barvu a texturu pozadí a průhlednost.
- Třída `ui-widget-shadow` – třída pro vržené stíny elementů. Aplikuje barvu a texturu pozadí, vlastní poloměr oblých rohů, průhlednost, vodorovný a svislý posun a „tloušťku“ stínu. Nastavení tloušťky se dosahuje nastavením vnitřních okrajů pro všechny strany vrženého stínu, jehož rozměry původně odpovídají elementu, který jej vrhá. Posunu vrženého stínu se dosahuje nastavením horního a levého vnějšího okraje (které mohou mít kladnou i zápornou hodnotu).

Tvorba vlastního motivu vzhledu od začátku

Pokud chcete vytvořit vlastní motiv vzhledu ručně zcela od začátku, nebo dokonce vytvořit vzhled pro svůj nový ovládací prvek, měli byste se řídit následujícími pravidly:

- dodržujte konvenci pojmenování šablon stylů knihovny jQuery UI,
- před všechny názvy tříd nezapomeňte uvést předponu `ui-`,
- nevytvářejte globální pravidla stylů – vždy před obecný element vložte nějakou specifickou třídu komponenty; například `.ui-widget-accordion input`,
- nepoužívejte v selektorech identifikátory elementů,
- jako oddělovače slov používejte pomlčky, ne podtržítka,
- vše pište malými písmeny.

Shrnutí

V této kapitole jsme se nejprve seznámili s pokročilými nastaveními motivů vzhledu v nástroji pro sestavení balíku ke stažení. Dozvěděli jsme se, že můžeme měnit umístění motivů vzhledu v balíku a omezovat platnost motivu vzhledu na určité elementy uvnitř svých webových stránek. Ukázali jsme si velmi jednoduchý příklad dynamického přepínání motivů vzhledu pomocí pokročilých nastavení.

Následně jsme si popsali, jak lze snadno vytvářet a upravovat motivy vzhledu v nástroji ThemeRoller, a také že existuje jeho zjednodušená verze pro webový prohlížeč Firefox.

Dále tato kapitola obsahuje pokročilejší řešení příkladu s přepínáním motivů vzhledu, v němž začleňujeme galerii motivů vzhledu nástroje ThemeRoller do svých stránek pomocí zásuvného modulu Theme Switcher. K tomuto příkladu byste se měli vrátit až po přečtení kapitoly „Navigace v obsahu stránky“.

Nakonec jsme se seznámili s frameworkem knihovny jQuery UI pro kaskádové styly. Už víme, jaké šablony stylů tvoří motiv vzhledu; z kterých tříd se skládá a jak postupovat při psaní vlastního motivu vzhledu od začátku.

Základy knihovny jQuery

Dříve v této knize jsme si řekli, že knihovna jQuery UI se zakládá na knihovně jQuery. Rozhodně bychom tedy neměli na knihovnu jQuery zapomenout, protože základní teoretické znalosti této knihovny bychom měli mít, abychom mohli nerušeně pracovat s knihovnou jQuery UI. V této kapitole se proto dozvíte:

- Co to je knihovna jQuery.
- Kde lze sehnat tuto knihovnu.
- Kde se nachází její oficiální dokumentace.
- Z čeho se knihovna jQuery skládá.
- Jak používat selektory knihovny jQuery.

Co je knihovna jQuery

Knihovna jQuery je velmi oblíbenou javascriptovou knihovnou. Základním cílem této knihovny je smazat rozdíly mezi implementacemi jazyka JavaScript v různých webových prohlížečích. Pokud jste někdy psali rozsáhlejší skript v jazyce JavaScript, zajisté víte, jak úmorné může být dosáhnout toho, aby tento skript fungoval ve všech hojně používaných webových prohlížečích stejně. Někdy se jedná až o téměř nadlidský úkol.

Knihovna jQuery se nesnaží zavést objektově-orientovaný způsob programování v jazyce JavaScript, ale místo toho se drží programování řízeného událostmi. Současně s tím se snaží o minimalizaci délky kódu, proto zavádí princip řetězení volání svých metod. Brzy zjistíte, že to, co byste normálně napsali v jazyce JavaScript na deset řádků, můžete s knihovnou

jQuery napsat na jeden řádek kódu. Co je však důležitější – přestože se vám zápis kódu takto zkrátí, bude fungovat (většinou) ve všech moderních webových prohlížečích stejně.

První verzi knihovny jQuery vydal známý programátor John Resig 26. srpna 2006. Tehdy ji vydal především proto, aby usnadnil práci sám sobě. Stejně jako nikoho jiného ho totiž nebavilo bojovat s rozdíly mezi webovými prohlížeči a také si chtěl ulehčit hledání a manipulaci s elementy v dokumentech HTML. Od té doby se rozrostla nejen samotná knihovna, ale rovněž její vývojový tým. K vývoji knihovny jQuery, jakožto knihovny s otevřeným zdrojovým kódem, se přidala řada dobrovolníků. Opět platí, že ačkoliv svou práci dělají rádi a zdarma, dobrovolným příspěvkem jistě nepohrdnou (<http://jquery.org/donate/>).

Autoři knihovny jQuery nabízejí tuto knihovnu pod licencemi MIT a GPL. Jedná se tedy o stejné licence jako u knihovny jQuery UI. Více informací a plné texty těchto licencí najdete na adrese <http://jquery.org/license/>. Nedělitelnou součástí knihovny jQuery je i selektorové jádro Sizzle, jehož autorem je společnost Dojo Foundation a nabízí jej po licencemi MIT, GPL a BSD. Chcete-li se dozvědět více informací o licenci BSD, najdete je kupříkladu zde: http://cs.wikipedia.org/wiki/BSD_licence.



UPOZORNĚNÍ

Je důležité si uvědomit, že pokud použijete ve své webové aplikaci knihovnu jQuery UI, používáte ve skutečnosti tři samostatné produkty a u každého z nich musíte přijmout některé licenční ujednání a řídit se jím. Krokem správným směrem je tedy nemazat z příslušných skriptů těchto produktů komentářové bloky s autorskými právy na jejich začátku, a to ani z produkčních (minifikovaných) verzí.

Stažení knihovny jQuery

Již víme, že každý balík s knihovnou jQuery UI obsahuje také knihovnu jQuery, a to jak v produkční, tak ve vývojové verzi. Knihovnu jQuery však můžete stáhnout a používat také samostatně. Všechny nezbytné informace a odkazy ke stažení knihovny jQuery včetně archivu se staršími verzemi najdete na adrese http://docs.jquery.com/Downloading_jQuery.

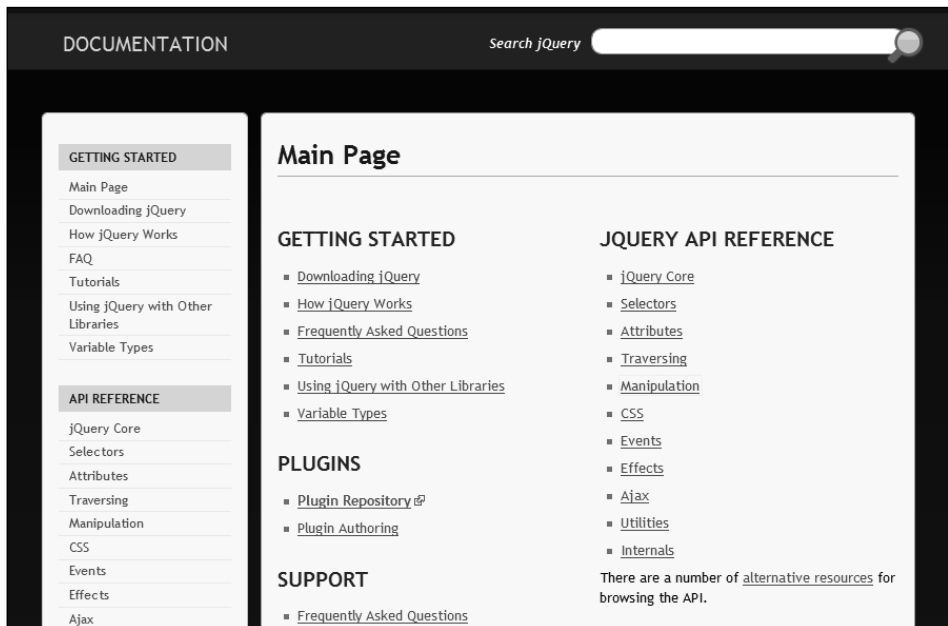
Jestliže nechcete stahovat tuto knihovnu, můžete použít některou z dostupných sítí CDN – síť Google Ajax API CDN, síť Microsoft CDN nebo síť jQuery CDN (jednoduše se pomocí elementu `script` odkážete na soubor přímo na síti CDN). Drobnou nevýhodou sítí CDN je, že jste vázáni na dostupnost připojení k Internetu a dostupnost těchto sítí, což vzhledem ke stabilitě těchto sítí a dostupnosti připojení není až tak zlé (u produkčních verzí webových aplikací nelze dostupnost připojení považovat za nevýhodu vůbec, protože musí být k dispozici stejně).

Další nevýhodou sítí CDN je o něco větší zpoždění při požadavku na příslušný skript, protože ten se přece jen nachází na jiném serveru než webová aplikace. Tato nevýhoda se však stává výhodou (a dosáhnete tedy menšího zpoždění požadavku), pokud nasazujete svou webovou aplikaci pro mezinárodní použití. Síť CDN se totiž skládá z mnoha serverů v různých částech světa a váš skript se vždy načte ze zdroje, který má fyzicky nejbližší ke klientovi.

K dispozici je rovněž celá řada zásuvných modulů pro knihovnu jQuery. Tyto zásuvné moduly jsou přehledně rozdělené do různých kategorií, jak se můžete sami přesvědčit na adrese <http://plugins.jquery.com/>. Nechybí zde ani vyhledávání zásuvných modulů.

Oficiální dokumentace

Oficiální dokumentaci knihovny jQuery najdete na adrese http://docs.jquery.com/Main_Page. Tato dokumentace je opravdu dobře zpracovaná, jak už jsme si zvykli z knihovny jQuery UI. Všechny informace jsou rozdělené do kategorií a u každé dokumentované části knihovny jQuery se kromě popisu nacházejí ukázky kódu, náhledy výsledků a komentáře uživatelů. Jestliže potřebujete rychle najít konkrétní informaci, můžete použít vyhledávání. Jak vypadá úvodní strana dokumentace knihovny jQuery, ukazuje obrázek 3.1.



Obrázek 3.1: Úvodní strana dokumentace knihovny jQuery

Základní stavební kameny knihovny jQuery

Nejdůležitější součástí knihovny jQuery, bez které se neobejdeme, je funkce se stejnojmenným názvem `jQuery()`. Tato funkce současně vyvolává nejvíce zmatení jak mezi začínajícími, tak mezi pokročilými programátory. Vzhledem k tomu, že má více druhů využití a různé pojmenování, někteří programátoři považují kód napsaný v knihovně jQuery za nový programovací jazyk. Není tomu tak – stále se jedná o kód jazyka JavaScript; jen to chce dobrou orientaci v terminologii a schopnostech knihovny jQuery a jazyka JavaScript.

Nesprávně tuto funkci popisuje i řada knih věnovaných přímo knihovně jQuery, takže si v tom pojdme udělat jasno.



POZNÁMKA

Pokud vám v této části kapitoly nebude všechno zcela jasné, nezdoufejte – v klidu si projděte příklady v následujících kapitolách, a až podlehnete své zvědavosti, jak to celé vlastně funguje, vraťte se sem znovu.

OBJEKTY V JAZYCE JAVASCRIPT

Pokud jste nikdy nepracovali s objekty v jazyce JavaScript, zastavte se na chvíli a přečtěte si tento popis – velmi vám to usnadní pochopení následujících termínů a hlavně se vám bude lépe pracovat s příklady v této knize.

Ve většině objektově orientovaných programovacích jazyků se setkáváme s pojmy jako **třída** a **objekt** (nebo také **instance třídy**). Jazyk JavaScript je trochu zvláštní programovací jazyk – třídy v něm nenajdeme, zato se hemží objekty. Pomineme fakt, že v tomto jazyce lze simulovat rovněž třídy, jelikož toto téma je už nad rámec této knihy, a zaměříme se čistě na objekty.

Jako příklad objektu můžeme vzít jakýkoliv objekt z reálného světa – například představme si, že jsme si nedávno koupili automobil Opel Corsa (nemůžeme říct prostě jen „automobil“, protože toto slovo označuje celou třídu objektů, ale my máme na mysli jen náš jediný konkrétní objekt). Naše smyšlené auto má mnoho vlastností se specifickými hodnotami – jeho výrobcem je společnost Opel, název modelu je Corsa A, kód VIN je WOL00009994022569, motor má výkon 59 kW atd. Dále má auto řadu funkcí – za všechny jmenujme kupříkladu, že se umí rozjet a zastavit.

Takový automobil by vypadal v jazyce JavaScript například následovně:

```
var nasAutomobil = {
  vyrobce: 'Opel',
  model: 'Corsa A',
  VIN: 'WOL00009994022569',
  vykon: 59,
  rozjedSe: function() {
    ...
  },
  zastavSe: function() {
    ...
  },
  ...
};
```

K **vlastnostem** a **metodám** objektu přistupujeme tak, že jejich název napíšeme za název objektu následovaný tečkou. Kdybychom chtěli kupříkladu vypsat hodnotu vlastnosti `model` objektu `nasAutomobil` do standardního výstražného okna webového prohlížeče, udělali bychom to takto:

```
window.alert(nasAutomobil.model);
```

Vlastnost je tudíž proměnná (nebo konstanta), která náleží objektu. Alternativně lze k vlastnostem objektu přistupovat stejně jako k prvkům pole (tento postup je vhodný zejména tehdy, když neznáme název vlastnosti předem):

```
window.alert(nasAutomobil[model]);
```

Stejným způsobem můžeme změnit hodnotu vlastnosti – stačí uvést daný výraz na levé straně příkazu přiřazení:

```
nasAutomobil.model = 'Insignia';
```

nebo také:

```
nasAutomobil[model] = 'Insignia';
```

Takto můžeme dokonce přidávat nové vlastnosti objektu:

```
nasAutomobil.rokVyroby = 2009;
```

Předchozím řádkem jsme rozšířili definici našeho objektu o vlastnost `rokVyroby` s hodnotou 2009.

Metoda je funkce, která patří k nějakému objektu. Od funkce se liší tedy jen tak, že ji nevoláme samostatně, ale uvedeme před ní název objektu s tečkou. V našem objektu jsme si ukázali dvě metody – metodu `rozjedSe()` a metodu `zastavSe()`. Kdybychom chtěli zavolat například metodu `rozjedSe()`, napsali bychom:

```
nasAutomobil.rozjedSe();
```

Jistou zvláštností jazyka JavaScript je, že všechny funkce a metody jsou současně také objekty. Co to znamená? Znamená to, že kromě toho, že je můžeme volat, můžeme jim přiřazovat vlastnosti a metody a pak s nimi pracovat. Přepíšeme si tedy náš ukázkový objekt `nasAutomobil` jako funkci:

```
function nasAutomobil() {
  window.alert(nasAutomobil.vyrobce + ' ' + nasAutomobil.model);
}
```

```
nasAutomobil.vyrobce = 'Ope1';
nasAutomobil.model = 'Corsa A';
nasAutomobil.VIN = 'WOL00009994022569';
nasAutomobil.vykon = 59;
nasAutomobil.rozjedSe = function() {
  ...
};
nasAutomobil.zastavSe = function() {
  ...
};
...
```

Jak je patrné, s takto vytvořeným objektem `nasAutomobil` pracujeme stejně jako s původním objektem, ale navíc ho můžeme volat jako funkci `nasAutomobil()`. Tato funkce vypíše text s výrobcem a modelem našeho smyšleného auta do výstražného okna.

PARAMETRY A ARGUMENTY FUNKCÍ A METOD

Přestože některé odborné články a knihy nerozlišují termíny **argument** a **parametr**, v této knize je rozlišovat budeme. Pokud definujeme funkci nebo metodu, mluvíme zásadně o parametrech. Mějme kupříkladu následující definici funkce:

```
function mojeFunkce(parametr1, parametr2) { ... }
```

V definici funkce/metody říkáme, že daná funkce/metoda má parametry, takže v tomto případě řekneme, že funkce `mojeFunkce()` má dva parametry – parametr s názvem `parametr1` a parametr s názvem `parametr2`. **Parametr** je tudíž lokální proměnná dané funkce/metody, jejíž hodnotu můžeme nastavit zvenku.

Hodnotou parametru je právě **argument**. Ten nastavujeme při volání dané funkce/metody. Například takto:

```
mojeFunkce(5, 'ahoj');
```

Jak je patrné, argument je už konkrétní hodnota, kterou předáváme při volání funkce/metody (případně hodnota některé proměnné/konstanty). V tomto případě bychom mohli tedy prohlásit, že předáváme číslo 5 jako první argument a textový řetězec 'ahoj' jako druhý argument funkci `mojeFunkce()`. Občas nebudeme označovat argument pořadím, ale názvem parametru, ke kterému patří. Nyní bychom mohli tudíž říct, že předáváme číslo 5 jako argument `parametr1` a textový řetězec 'ahoj' jako argument `parametr2` funkci `mojeFunkce()`.

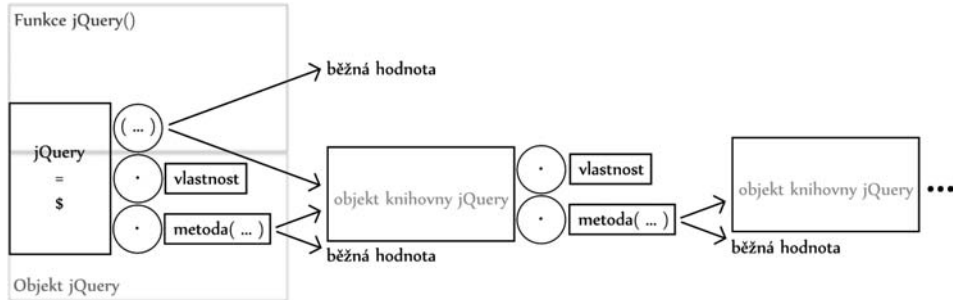
Začneme tak, že si uvedeme tři termíny, jež tvoří základ knihovny jQuery a které si lidé často pletou:

- **Funkce jQuery()** – hlavní funkce knihovny jQuery. Pracujeme s ní jako s jakoukoliv běžnou funkcí – tj. zavoláme ji a předáme jí nějaké argumenty, přičemž očekáváme, že něco provede a případně nám vrátí nějakou hodnotu. Tato funkce má více variant; to znamená, že podle počtu a typu argumentů, které jí předáváme při volání, provede různé akce. Různé varianty funkce `jQuery()` si popíšeme za malý okamžik.
- **Globální objekt jQuery** (někdy také jen **objekt jQuery**) – ve skutečnosti se jedná opět o funkci `jQuery()`, ale vzhledem k tomu, že v jazyce JavaScript je každá funkce také objektem, tentokrát na ni budeme pohlížet jako na objekt (toto je obdoba našeho objektu `nasAutomobil`, jež jsme definovali jako funkci). S jakýmkoliv objektem můžeme dělat v základu tři věci – načítat hodnoty jeho vlastností, ukládat hodnoty jeho vlastností a volat jeho metody. Objekt jQuery se nijak neliší. Ukázkou práce s tímto objektem si předvedeme dále v této části kapitoly.
- **Objekt knihovny jQuery** – je velmi důležité si všimnout, že v předchozím bodu tohoto seznamu (globální objekt jQuery) označujeme slovo „jQuery“ jako zdrojový kód ve větě. Není to náhoda – jednalo se totiž o jeden konkrétní objekt s názvem jQuery. Tentokrát však mluvíme o typu objektu. Většina variant funkce `jQuery()` a celá řada metod globálního objektu jQuery vrací objekt knihovny jQuery jako svou návratovou hodnotu. Objekt knihovny jQuery je opět jen obyčejný objekt, takže můžeme přistupovat k jeho vlastnostem a volat jeho metody. Převážná část jeho metod vrací

rovněž další objekt knihovny jQuery – z toho vychází princip řetězení volání metod, o kterém jsme se již bavili.

Funkce jQuery() má také svůj kratší alias \$(). Stejně tak můžeme objekt jQuery zapisovat jako \$ (jelikož víme, že objekt jQuery je ve skutečnosti funkce jQuery(), na kterou nahlédneme z jiného úhlu pohledu).

Obrázek 3.2 znázorňuje právě popsané termíny.



Obrázek 3.2: Základní stavební kameny knihovny jQuery

Teorie už bylo až nad hlavu – pojďme si ukázat nějaké příklady.

Funkce jQuery()

Základní způsob použití (a také nejčastější způsob) funkce jQuery() spočívá ve výběru elementů dokumentu, na které potom můžeme aplikovat metody knihovny jQuery. Kdybychom kupříkladu chtěli zjistit, jaké jméno uživatel vyplnil do formuláře na stránce, provedli bychom to takto:

```
var jmenoUzivatele = jQuery('input[name=jmeno]').attr('value');
```

Nebo bychom tento řádek mohli napsat stručněji:

```
var jmenoUzivatele = $('input[name=jmeno]').attr('value');
```

Přestože je tento kód velmi krátký, odehrává se na něm hned několik věcí. Nejprve v části \$('input[name=jmeno]') voláme funkci jQuery() a předáváme jí selektor jazyka CSS, což je obyčejný textový řetězec. O selektorech jazyka CSS, které můžeme používat v knihovně jQuery, si povíme více informací později v této kapitole. V tomto případě vybíráme pomocí selektoru 'input[name=jmeno]' element input, jehož atribut name obsahuje hodnotu jmeno.

Volání \$('input[name=jmeno]') nám vrátí bezejmenný objekt knihovny jQuery. Tento objekt může obalovat element input se jménem jmeno, pokud se daný element na stránce skutečně nachází, nebo žádný element v případě, že takový element neexistuje (proto se objekt knihovny jQuery někdy označuje také jako **obalená skupina knihovny jQuery**, přičemž může obalovat i více elementů). Následně voláme metodu attr() tohoto objektu knihovny jQuery, která nám vrátí hodnotu uvedeného atributu – v tomto případě atributu value, a to pro první element obsažený v daném objektu, je-li nějaký. Získanou hodnotu ukládáme do proměnné jmenoUzivatele.

Většina volání funkce `jQuery()` vrací objekt knihovny jQuery. Výsledek se však liší v tom, co tento objekt obaluje. Tato funkce se totiž chová jinak podle toho, kolik argumentů jí předáme a jakého jsou typu.

Jeden způsob využití už jsme viděli – v následující tabulce se nachází seznam všech možných způsobů použití. Tento seznam se může v průběhu vývoje knihovny jQuery měnit, proto je vhodné sledovat informace na adrese <http://api.jquery.com/jquery/>. Kromě toho, že jsou tyto informace aktuální, jsou také mnohem podrobnější a doplněné o příklady.

Tabulka 3.1: Varianty funkce `jQuery()`

Účel	Varianta funkce	Popis parametrů
Hledání skupiny elementů v dokumentu	<code>jQuery(selektor[, kontext])</code>	<ul style="list-style-type: none"> ■ <code>selektor</code> – textový řetězec, který reprezentuje selektor ■ <code>kontext</code> – element, dokument nebo objekt knihovny jQuery
	<code>jQuery(element)</code>	<ul style="list-style-type: none"> ■ <code>element</code> – element, jež chceme zabalit do objektu knihovny jQuery
	<code>jQuery(objekt)</code>	<ul style="list-style-type: none"> ■ <code>objekt</code> – obyčejný objekt, který chceme zabalit do objektu knihovny jQuery
	<code>jQuery(poleElementu)</code>	<ul style="list-style-type: none"> ■ <code>poleElementu</code> – pole obsahující skupinu elementů, které chceme zabalit do objektu knihovny jQuery
	<code>jQuery(objektKnihovnyjQuery)</code>	<ul style="list-style-type: none"> ■ <code>objektKnihovnyjQuery</code> – objekt knihovny jQuery, který chceme klonovat
	<code>jQuery()</code>	Tato varianta žádný parametr nemá a vrací prázdný objekt knihovny jQuery
Tvorba elementů dokumentu z textového řetězce s kódem jazyka HTML	<code>jQuery(kodHTML[, dokument])</code>	<ul style="list-style-type: none"> ■ <code>kodHTML</code> – textový řetězec obsahující kód jazyka HTML, z něž chceme vytvořit elementy za běhu ■ <code>dokument</code> – dokument, v němž nové elementy vzniknou
	<code>jQuery(kodHTML, vlastnosti)</code>	<ul style="list-style-type: none"> ■ <code>kodHTML</code> – textový řetězec s kódem jazyka HTML, ve kterém definujeme jediný element; například <code>'<div/>'</code> nebo <code>'<div></div>'</code> ■ <code>vlastnosti</code> – objekt, s jehož vlastnostmi specifikujeme atributy, události a metody, které by se měly zavolat na nově vzniklý element.
Spuštění funkce po načtení modelu DOM	<code>jQuery(funkceZpetnehoVolani)</code>	<ul style="list-style-type: none"> ■ <code>funkceZpetnehoVolani</code> – funkce zpětného volání, kterou chceme zavolat po načtení modelu DOM

Velmi důležitým způsobem volání funkce `jQuery()`, který budeme používat v této knize, je poslední uvedený způsob (`jQuery(funkceZpetnehoVolani)`), jehož účelem je spuštění funkce po načtení modelu DOM. Model DOM je objektový model dokumentu, který webový prohlížeč skládá za účelem práce s dokumentem jazyka HTML v kódu jazyka JavaScript,

a to postupně, když načítá jednotlivé jeho elementy. Jakmile webový prohlížeč takto zpracuje celý dokument, můžeme v jazyce JavaScript hledat všechny elementy daného dokumentu a pracovat s nimi.

```
$(function() {
    ...
});
```

Do těla výše uvedené anonymní funkce můžeme tudíž napsat místo tří teček jakýkoliv kód, v němž chceme pracovat s elementy dokumentu – například vyhledat element, ze kterého bychom chtěli vytvořit dialogové okno pomocí ovládacího prvku Dialog, jak zjistíme v kapitole „Navigace v obsahu stránky“. Díky tomu, že počkáme na načtení modelu DOM, máme jistotu, že element najdeme, pokud v dokumentu existuje. Jednoduše řečeno – jedná se o vhodný vstupní bod do programu (zvláště do toho, v němž používáme komponenty knihovny jQuery UI), který tímto způsobem budeme používat napříč celou touto knihou.

Globální objekt jQuery

Globální objekt jQuery nabízí celou řadu vlastností a metod. Přehled všech jeho vlastností a metod lze najít na adrese <http://api.jquery.com/>. Snadno je poznáme tak, že před jejich názvem se nachází předpona jQuery., přičemž za metodami následuje dvojice kulatých závorek, ale za vlastnostmi nikoliv. Nebudeme se zabývat tím, k čemu slouží všechny tyto vlastnosti a metody, ale začneme kupříkladu stručným popisem vlastnosti jQuery.fx.off.

Vlastnost jQuery.fx.off globálně zakazuje všechny animace:

```
jQuery.fx.off = true;
```

Předchozím příkazem přiřazení nastavujeme vlastnosti fx.off globálního objektu jQuery hodnotu true, čímž zakážeme všechny animace – to znamená, že animované elementy se okamžitě dostanou do svého cílového stavu, aniž by se animace jakkoliv projevila. Animace je možné zase zapnout přidělením hodnoty false. Znak tečky v názvu vlastnosti fx.off značí, že vlastnost fx je také objektem se svou vlastností off. Tento objekt nás však nezajímá – důležité je, že jsme si ukázali, jak změnit hodnotu vlastnosti globálního objektu jQuery.

Stejně tak bychom neměli zapomínat, že je možné napsat tento kód stručněji:

```
$.fx.off = true;
```

Jako ukázkovou metodu globálního objektu jQuery si zvolíme například metodu jQuery.isArray(). Tato metoda vrací hodnotu true, jestliže je jí předaný argument skutečným polem jazyka JavaScript. Pokud tedy spustíme tento kód:

```
var jeToPole = jQuery.isArray(['jablko', 'hruška']);
```

proměnná jeToPole bude obsahovat hodnotu true. A zde je kratší zápis tohoto kódu s naprosto stejným výsledkem:

```
var jeToPole = $.isArray(['jablko', 'hruška']);
```

Objekty knihovny jQuery

Už víme, že pod pojmem objekt knihovny jQuery si nemáme představovat žádný konkrétní objekt, ale celou skupinu (typ) objektů. Jak takový objekt získáme, již rovněž víme – většina variant funkce jQuery() a některé metody globálního objektu jQuery nám vracejí objekt knihovny jQuery jako svou návratovou hodnotu.

Objekt knihovny jQuery má stejně jako globální objekt jQuery spoustu vlastností a metod, které najdeme na adrese <http://api.jquery.com/>. Opět je velmi jednoduše rozlišíme – jejich název začíná tečkou (bez předpony jQuery), přičemž za metodami následuje dvojice kulatých závorek, ale za vlastnostmi nikoliv. S jednou metodou objektu knihovny jQuery jsme se už setkali při popisu funkce jQuery() – jednalo se o metodu .attr().

```
var jmenoUzivatele = jQuery('input[name=jmeno]').attr('value');
```

Na tomto řádku jsme volali metodu attr() bezejmenného objektu knihovny jQuery, který jsme získali voláním jQuery('input[name=jmeno]'). Tato metoda vrací pro vybraný element input konkrétní textovou hodnotu (existuje-li) atributu value, kterou ukládáme do proměnné jmenoUzivatele. Na takovou hodnotu už nemůžeme volat další metody a řetězit volání metod, jak jsme si slibovali dříve. Jaká metoda objektu knihovny jQuery tedy kupříkladu vrací jiný objekt knihovny jQuery, na který můžeme volat další metodu, a tak stále dokola?

Uvedeme si například hned dvě takové metody – metodu addClass() a metodu removeClass(). První jmenovaná metoda (addClass()) přidá k vybrané skupině elementů třídy předané jako argument (více tříd oddělujeme mezerou) a metoda removeClass() je naopak odebrává. Obě tyto metody vracejí objekt knihovny jQuery; tj. stejnou skupinu elementů, které však mají upravený atribut class. Dejme tomu, že máme dokument s mnoha elementy p, které mají třídu cernyOdstavec. Budeme chtít všem těmto odstavcům v dokumentu odebrat třídu cernyOdstavec a přidat k nim třídu modryOdstavec:

```
$('.p').removeClass('cernyOdstavec').addClass('modryOdstavec');
```

Jak je patrné, díky řetězení volání metod je řešení opravdu jednoduché. Nejprve vybíráme všechny elementy p v dokumentu voláním \$('.p'). Tím získáme objekt knihovny jQuery, na nějž voláme metodu removeClass() s argumentem 'cernyOdstavec' a ta nám vrátí další objekt knihovny jQuery, jenž reprezentuje všechny elementy p v dokumentu, ale bez třídy cernyOdstavec.

Jelikož se opět jedná o objekt knihovny jQuery, můžeme na něho zavolat metodu addClass s argumentem modryOdstavec. Takto vznikne konečně náš finální objekt knihovny jQuery, který představuje všechny elementy p v dokumentu bez třídy cernyOdstavec, avšak doplněné o třídu modryOdstavec.

Kontrolní otázka: „Kolik funkcí jsme v tomto případě volali?“

Správná odpověď: „Volali jsme funkci jQuery() a dvě metody objektu knihovny jQuery, takže celkem tři funkce“.

Mezi nejznámější vlastnost objektu knihovny jQuery patří vlastnost length. Tato vlastnost nám sděluje, kolik elementů daný objekt obsahuje. Zde je příklad:

```
var pocetOdstavcu = $('.p').length;
```

V předchozím kódu ukládáme počet elementů `p` v dokumentu do proměnné `pocetOdstavcu`, a to tak, že vybereme tyto elementy do objektu knihovny jQuery a načteme hodnotu jeho vlastnosti `length`.

Jak používat selektory knihovny jQuery

V minulé části této kapitoly jsme zjistili, že funkci `jQuery()` můžeme předat jako její první argument selektor v podobě textového řetězce. Tyto selektory vycházejí z jazyka CSS, přičemž knihovna jQuery navíc přidává spoustu vlastních selektorů. To je velmi výhodné, protože se nemusíme učit novou syntaxi selektorů od začátku.

Co to vlastně je selektor? **Selektor** je textový výraz, s jehož pomocí vyhledáváme elementy v dokumentu. V jazyce CSS obvykle na vyhledané elementy aplikujeme nějaká pravidla stylů, ale v knihovně jQuery si s nimi můžeme dělat, co se nám zlíbí – dokonce je můžeme z dokumentu zcela odstranit.

Zde jsou příklady standardních selektorů z jazyka CSS (do verze 3, a to včetně):

- `#identifikator` – odpovídá jedinému elementu s identifikátorem `identifikator`.
- `znacka` – vyhovuje všem elementům se značkou `znacka` (například `div`).
- `.trida` – reprezentuje všechny elementy s třídou `trida`.
- `[atribut="hodnota"]` – odpovídá všem elementům, jejichž atribut `atribut` má hodnotu `hodnota`, přičemž uvozovky musíme doplňovat pouze k víceslovným hodnotám nebo hodnotám se speciálními znaky.
- `znacka.trida` – vyhovuje všem elementům `znacka` s třídou `trida`.
- `znacka1, znacka2` – reprezentuje všechny elementy, které mají značku `znacka1` nebo `znacka2`.
- `znacka1 znacka2[atribut="hodnota"]` – odpovídá všem elementům `znacka2`, jejichž atribut `atribut` má hodnotu `hodnota`, a zároveň jsou potomky elementů `znacka1`.
- `:disabled` – reprezentuje všechny zakázané elementy.
- Atd.

A tady jsou ukázky nových selektorů knihovny jQuery:

- `:submit` – odpovídá všem elementům `input` s typem `submit` (odesílací tlačítko formuláře).
- `:password` – reprezentuje všechny elementy `input` typu `password`.
- `:file` – vyhovuje všem elementům `input` s typem `file`.
- Atd.

Kompletní seznam všech selektorů knihovny jQuery spolu s popisy a příklady lze najít na adrese <http://api.jquery.com/category/selectors/>.

Jakmile předáme selektor funkci `jQuery()`, vybere elementy z dokumentu a zabalí je do objektu knihovny jQuery. Co například dělá níže uvedené volání?

```
var pocetPriloh = $('#zprava :file.priloha').length;
```

Na tomto řádku vybíráme všechny elementy `input` s typem `file` a třídou `pri_loha`, které se nacházejí ve struktuře dokumentu pod elementem s identifikátorem `zprava` (tj. jsou jeho potomky). Vybrané elementy se obalují do objektu knihovny jQuery, jehož hodnotu vlastnosti `length` ukládáme do proměnné `pocetPri_loh`.

Shrnutí

V této kapitole jsme se dozvěděli, co to je knihovna jQuery, kdo ji vytvořil a odkud ji můžeme stáhnout. Potom jsme si ukázali, kde se nachází její oficiální dokumentace, v níž najdeme všechny nezbytné informace rozdělené do kategorií.

Popsali jsme si, z jakých základních stavebních kamenů se knihovna jQuery skládá. Kromě toho, že umíme volat funkci `jQuery()`, víme také, jak používat globální objekt jQuery a objekty knihovny jQuery. Mimo jiné jsme zjistili, kde se bude nacházet vstupní bod do programů, které si popíšeme v této knize.

Nakonec jsme si představili selektory, s nimiž můžeme pracovat v knihovně jQuery. Tyto základní znalosti stačí pro důkladné pochopení zdrojových kódů v této knize.

Knihovna jQuery však velmi ulehčuje práci v jazyce JavaScript, a pokud vás tato kapitola zaujala, určitě si toto téma nastudujte podrobněji.

Navigace v obsahu stránky

Nyní se budeme zabývat vcelku běžnou situací – chceme prezentovat informace na svých webových stránkách. Nejprve si musíme rozmyslet, jak své informace uspořádáme. Jakmile je rozdělíme do skupin a podskupin, začneme přemýšlet, jak dosáhnout toho, aby se k nim uživatel snadno dostal. Obohatíme tudíž naše webové stránky o různé navigační prvky – nabídky, panely, okna apod.

Jak vám s tímto úkolem může pomoci knihovna jQuery UI, se dozvíte v této kapitole. Naučíte se:

- vkládat nezbytné skripty a šablony stylů do dokumentu jazyka HTML,
- používat třídy frameworku knihovny jQuery UI pro kaskádové styly v praxi,
- rozdělovat obsah webových stránek do karet,
- načítat obsah jednotlivých částí webových stránek prostřednictvím technologie AJAX,
- uspořádat informace s ohledem na úsporu výšky stránky pomocí harmoniky,
- upozornit uživatele na nějakou důležitou skutečnost s použitím modálního dialogového okna.

Náš cíl

Budeme vytvářet zábavný web pro všechny uživatele, kteří si chtějí ukrátit volný čas. Konkrétně jim nabídneme různé vtipy a možnost popovídat si s ostatními návštěvníky našich webových stránek. Nezapomínejme ale, že se jedná pouze o příklad, proto nebudeme implementovat veškerou funkčnost, kterou by tyto stránky měly mít. Na následujícím obrázku můžete vidět, jak by měly tyto webové stránky vypadat na konci této kapitoly.



Obrázek 4.1: Zábavné webové stránky

Rozvržení stránky bude statické a dynamicky budeme měnit jen obsah jednotlivých karet. Jejich obsah budeme načítat pomocí technologie AJAX (asynchronní JavaScript a XML). Na výše uvedeném obrázku můžete rozpoznat tři hlavní sekce tohoto webu.

První sekci je úvodní strana, která vítá návštěvníky a zobrazuje nejnovější a nejlepší vtipy v pravém panelu. Abychom ušetřili místo na stránce, zobrazíme vždy jen nejnovější vtipy, nebo pouze nejlepší vtipy, ale nikdy obě kategorie vtipů současně.

Druhá sekce obsahuje samotné vtipy. Každý vtip má jistý náznak uživatelského hodnocení a datum svého vložení.

Třetí sekce bude o něco zajímavější z hlediska vývoje. Ačkoliv samotný obsah dané karty bude pro zachování jednoduchosti statický, pokud jako nepřihlášení uživatelé poprvé zobrazíme její obsah, objeví se okno s upozorněním. Toto okno bude informovat o tom, že jestliže si chce uživatel popovídat s ostatními návštěvníky našich webových stránek, musí se přihlásit.

Základ webových stránek

Jelikož tento příklad používá technologii AJAX, vyžaduje ke své funkci webový server. Ujistěte se tedy, že kopírujete základ příkladu do podsložky adresáře webového serveru. O této

cílové složce budeme dále v této kapitole mluvit jako o složce příkladu. Základ tohoto příkladu se skládá ze čtyř složek – *css*, *fonty*, *js* a *obrazky*.

Složka *css* je prozatím prázdná a je určena k ukládání šablon kaskádových stylů. Stejný případ je u složky *js* – ta je rovněž prázdná a budeme do ní ukládat skripty jazyka JavaScript.

Ve složce *fonty* se nachází soubor *akaDylan Open.ttf*. Jedná se o volně dostupné písmo *akaDylan* ve variantě *Open* od Jamese Milligana. Toto písmo si nemusíme instalovat do operačního systému. S pomocí jazyka CSS ho použijeme pro hlavní nadpis našich webových stránek.

Jak už název napovídá, složka *obrazky* obsahuje obrázky ve formátu PNG. Obrázek *logo-smajlik.png* použijeme pro hlavní nadpis webových stránek. Obrázek *ikona-info.png* najde uplatnění v okně s upozorněním pro uživatele a obrázky *hodnoceni-0.png* až *hodnoceni-5.png* použijeme k napodobení uživatelského hodnocení vtípů.

Knihovna jQuery UI s motivem vzhledu Start

Z oficiálních stránek knihovny jQuery UI si můžete stáhnout tuto knihovnu s motivem vzhledu Start. K tomuto příkladu vám stačí ovládací prvky *Accordion*, *Dialog* a *Tabs*; můžete však stáhnout celou knihovnu, protože stejně budete pracovat jen na ukázkovém příkladu, a ne na produkční verzi webových stránek.



POZNÁMKA

Jak už jsme si řekli v úvodu k této knize, v archivu se zdrojovými kódy příkladů najdeme ve složce *jQuery UI 1.8.18* kompletní knihovnu jQuery UI ve verzi 1.8.18 s motivem vzhledu Start.

Z balíku staženého z oficiálních stránek knihovny jQuery UI nebo ze složky archivu se zdrojovými kódy, která obsahuje knihovnu jQuery UI, zkopírujte soubory *jquery-*{číslo verze}*.min.js* a *jquery-ui-*{číslo verze}*.custom.min.js* ze složky *js* do podsložky *js* ve složce příkladu. Je vhodné přejmenovat soubory v cílové složce tak, aby neobsahovaly čísla verzí a označení variant. Důvod je ten, že na tyto soubory se budete odkazovat v kódu, a kdybyste chtěli v budoucnu aktualizovat knihovny jQuery a jQuery UI, museli byste přepisovat i zdrojový kód. Proto tyto soubory pojmenujte *jquery.js* a *jquery-ui.js*.



UPOZORNĚNÍ

Zvolený postup není nejlepší pro vývojovou verzi webových stránek. Pokud už tedy umíte ladit kód jazyka JavaScript ve svém webovém prohlížeči, zvolte raději plné verze těchto dvou souborů. Kompletní verzi skriptu knihovny jQuery najdete v podsložce *development-bundle* v souboru *jquery-*{číslo verze}*.js* a kompletní verzi skriptu knihovny jQuery UI v podsložce *development-bundle/ui/* v souboru *jquery-ui-*{číslo verze}*.custom.js*. Po zkopírování do cílové složky oba soubory přejmenujte stejně, jako byste pracovali s jejich zmenšenými verzemi.

Složka knihovny jQuery UI obsahuje také podsložku *css*. Z této podsložky zkopírujte celou složku *start* do podsložky *css* ve složce příkladu. V této zkopírované složce přejme-

nujte soubor `jquery-ui-{číslo verze}.custom.css` na soubor `jquery-ui.css` ze stejného důvodu, z jakého jste přejmenovávali skripty.

Dokumenty HTML

Nyní máme důvod k oslavě – už máme vše, co potřebujeme, abychom mohli začít psát kód. Než se dostaneme k psaní kódu JavaScriptu s pomocí knihovny jQuery UI, tak to sice ještě chvílku potrvá, ale alespoň si napíšeme základ našich webových stránek v jazyce HTML.



UPOZORNĚNÍ

Veškerým svým souborům (dokumentům HTML, šablonám stylů a skriptům) nastavte znakovou sadu UTF-8. Umožňuje-li vaše vývojové prostředí takové rozlišení, volte znakovou sadu UTF-8 bez značky BOM (Byte Order Mark).

Hlavní dokument HTML

Ve složce příkladu vytvoříme soubor `index.html`. Do něj vložíme níže uvedený zdrojový kód:

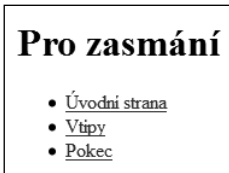
```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html
  PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="cs" lang="cs">
<head>
  <meta http-equiv="content-type" content="text/html; charset=utf-8" />
  <meta http-equiv="content-language" content="cs" />
  <title>Kapitola 4 – příklad Pro zasmání</title>
</head>
<body>
  <h1>Pro zasmání</h1>
  <div id="nabidka">
    <ul>
      <li><a href="uvodni-strana.html">Úvodní strana</a></li>
      <li><a href="vtipy.html">Vtipy</a></li>
      <li><a href="pokec.html">Pokec</a></li>
    </ul>
  </div>
</body>
</html>
```

V tomto kódu vytváříme běžnou českou stránku ve značkovacím jazyce XHTML se znakovou sadou UTF-8. Tato stránka obsahuje jen hlavní nadpis a neuspořádaný seznam s několika odkazy na doposud neexistující stránky, jak je patrné na obrázku 4.2.

Důležitou částí tohoto kódu je tento element `div`:

```
<div id="nabidka">
  <ul>
    <li><a href="uvodni-strana.html">Úvodní strana</a></li>
    <li><a href="vtipy.html">Vtipy</a></li>
    <li><a href="pokec.html">Pokec</a></li>
```

```
</ul>
</div>
```



Obrázek 4.2: Běžná stránka v jazyce XHTML

Označujeme si ho identifikátorem nabídka. Díky tomu jej budeme moci později snadno najít a přetvořit na hlavní nabídku ve formě karet. Uspořádání dat v tomto elementu do neuspořádaného seznamu s odkazy je rovněž důležité, jak později zjistíte při popisu ovládacího prvku Tabs.

Úvodní strana

Vytvoříme soubor *uvodni-strana.html* ve složce příkladu. Potom do něj vložíme tento kód:

```
<div id="postranni-panel">
  <h2><a href="#">Nejnovější vtipy</a></h2>
  <div>
    <p class="vtip hodnoceni-1">
      Farář se ptá Pepíčka: "Modlíte se před jídlem?"<br />
      Pepíček odpoví: "No, jen když máme jíst houby, co nasbíral táta".
      <span>10. 2. 2012 09:55</span>
    </p>
    <p class="vtip hodnoceni-3">
      Chuck Norris se narodil v dřevěném srubu, který
      si sám postavil.
      <span>29. 1. 2012 16:03</span>
    </p>
  </div>
  <h2><a href="#">Nejlepší vtipy</a></h2>
  <div>
    <p class="vtip hodnoceni-5">
      Sejdou se zvířátka v pralese.<br />
      Lev povídá: "Zítřej pojedeme na výlet".<br />
      Žába po něm opakuje: "Zítřej pojedeme na výlet".<br />
      Lev pokračuje: "Vezměte si batůžky a do nich svačinky".<br />
      Žába po něm zase opakuje: "Vezměte si batůžky a do nich svačinky".
      <br />
      Lev povídá: "A ta zelená tlama nikam nepojede".<br />
      A žába si povzdechne: "Chudák krokodýl, ten se tak těšil".
      <span>10. 9. 2011 23:30</span>
    </p>
    <p class="vtip hodnoceni-4">
      Jaký je rozdíl mezi Windows a naší vládou?<br />
      Žádný – všichni doufají, že příští verze bude stabilní.
      <span>4. 6. 2011 12:30</span>
    </p>
  </div>
</div>
```

```

    </p>
  </div>
</div>
<h2>Vítejte,</h2>
<p>
  pokud se chcete pobavit, jste na správném místě. Pokusí se vás
  rozesmát velké množství vtípů z nejrůznějších kategorií a ve volné
  chvílce si můžete popovídat s ostatními návštěvníky těchto webových
  stránek.
</p>
<div class="zapati">&copy; Ondřej Baše, 2011</div>

```

Všimněte si něčeho zvláštního – tento soubor není kompletním dokumentem HTML; neobsahuje totiž definici typu dokumentu ani elementy `html`, `head` a `body`. Obsah tohoto souboru budeme načítat přímo do těla dokumentu `index.html` prostřednictvím technologie AJAX, takže se ve skutečnosti jedná jen o fragment kódu, jež dynamicky vložíme do skutečné stránky.

Začneme elementem s identifikátorem `postranni-pane1`. Tento element později přeměníme na „harmoniku“ s pomocí ovládacího prvku `Accordion`. Strukturu kódu uvnitř tohoto elementu jsme rovněž napsali tak, jak ji vyžaduje ovládací prvek `Accordion`, což si popíšeme dále v této kapitole. Uvnitř tohoto elementu je však jeden typ elementu, který jsme si definovali sami, aniž by jej po nás vyžadovala knihovna `jQuery UI`, a to kupříkladu tento:

```

<p class="vtip hodnoceni-1">
  Farář se ptá Pepíčka: "Modlíte se před jídlem?"<br />
  Pepíček odpoví: "No, jen když máme jíst houby, co nasbíral táta".
  <span>10. 2. 2012 09:55</span>
</p>

```

Tento text jsme napsali do elementu odstavce, protože je pro vtip nejvhodnější a navíc nám vtipy vizuálně oddělí mezerami. Elementu pro vtip jsme přiřadili třídu `vtip`, abychom mohli upravit vzhled vtipů, a také třídu `hodnoceni-1`. Třídy `hodnoceni-0` až `hodnoceni-5` jsou součástí simulovaného systému hodnocení. V našem příkladu však bude systém hodnocení spočívat v tom, že jen nastavíme jednotlivým jeho třídám mírně odlišné grafické prvky (počet smajlíků podle čísla třídy). Vnitřní element `span` obsahuje čas vložení daného vtipu a tento samostatný element volíme jen proto, abychom mohli čas vložení graficky odlišit od textu vtipu.



POZNÁMKA

Ve skutečném systému by se samozřejmě tento kód generoval nejspíše tak, že data vtipů (text, hodnocení a datum vložení) by byla uložena v databázi, z níž by je načítal skript na straně serveru, a potom byste z nich pomocí nějakého šablonového systému vytvářeli tento kód jazyka HTML. V dnešní době už pravděpodobně nikdo nevytváří zcela statické webové stránky a nepřidává nové příspěvky editací zdrojového kódu. Proto berte tento kód jen jako vodítko, jak by měl vypadat výsledný kód jazyka HTML po mnoha transformacích dat z databáze. To samé samozřejmě platí o většině kódu jazyka HTML v této knize.

Za elementem s identifikátorem `post-ranni-panel` následuje nadpis druhé úrovně a odstavec s uvítáním. Na konci stránky se nachází zápatí – element `div` s třídou `zapatí`. Zápatí vkládáme do naší stránky z více důvodů, jak si vysvětlíme později.

Vtipy

Vytvoříme nový soubor `vtipy.html` ve složce příkladu. Napíšeme do něj tento zdrojový kód:

```
<p class="vtip hodnoceni-1">
  Farář se ptá Pepíčka: "Modlíte se před jídlem?"<br />
  Pepíček odpoví: "No, jen když máme jíst houby, co nasbíral táta".
  <span>10. 2. 2012 09:55</span>
</p>
<p class="vtip hodnoceni-3">
  Chuck Norris se narodil v dřevěném srubu, který
  si sám postavil.
  <span>29. 1. 2012 16:03</span>
</p>
<p class="vtip hodnoceni-5">
  Sejdou se zvířátka v pralese.<br />
  Lev povídá: "Zítra pojedeme na výlet".<br />
  Žába po něm opakuje: "Zítra pojedeme na výlet".<br />
  Lev pokračuje: "Vezměte si batůžky a do nich svačinky".<br />
  Žába po něm zase opakuje: "Vezměte si batůžky a do nich svačinky".<br />
  Lev povídá: "A ta zelená tlama nikam nepojede".<br />
  A žába si povzdechne: "Chudák krokodýl, ten se tak těšil".
  <span>10. 9. 2011 23:30</span></p>
<p class="vtip hodnoceni-4">
  Jaký je rozdíl mezi Windows a naší vládou?<br />
  Žádný – všichni doufají, že příští verze bude stabilní.
  <span>4. 6. 2011 12:30</span>
</p>
<div class="zapatí">&copy; Ondřej Baše, 2011</div>
```

Předchozí kód je asi nejméně záživným kódem, který v této knize najdete. Pouze jsme zkopírovali všechny elementy `p` s třídou `vtip` z harmoniky v minulém dokumentu a vložili je do tohoto souboru. Dále jsme přidali zápatí, které obsahovala i poslední stránka.

V praxi by tato stránka obsahovala daleko větší množství vtipů. Rozhodně by si zasloužila také filtrování vtipů podle kategorií a řazení vtipů například podle uživatelského hodnocení, data vložení apod.

Pokec

Ve složce příkladu vytvoříme soubor `pokec.html`. Vložíme do něj níže uvedený kód:

```
<p>
  12. 2. 2012 08:01 (legolas): Čau všichni, co tak po ránu děláte?
</p>
<p>
  12. 2. 2012 08:02 (dm19): Zrovna jsem si uvařil kávu. Koukám, že sem už
  dlouho nikdo nevložil nic nového.
</p>
```

```

<p>
  12. 2. 2012 08:04 (legolas): Nj, taky koukám. Je tu nuda, budeme s tím
  muset něco udělat. :-)
</p>
<p>...</p>
<div class="zapati">&copy; Ondřej Baše, 2011</div>

<div id="upozorneni" title="Pokec pouze pro čtení">
  <p>
    
    Bez přihlášení můžete pouze číst příspěvky ostatních registrovaných
    uživatelů tohoto webu.
  </p>
  <p>
    Pokud jste se dosud neregistrovali, můžete to provést nyní – získáte
    řadu výhod. Kromě možnosti pokecat si s ostatními uživateli v této
    části se můžete přihlásit k odběru novinek pomocí e-mailu a vybírat
    své oblíbené vtipy.
  </p>
</div>

```

Na začátku tohoto kódu vkládáme několik statických odstavců s fiktivním rozhovorem dvou návštěvníků našich webových stránek. Za nimi se nachází nám už dobře známé zápatí.

Zajímavou částí předchozího kódu je následující element `div`:

```

<div id="upozorneni" title="Pokec pouze pro čtení">
  ...
</div>

```

Přiřadili jsme mu identifikátor `upozorneni`, abychom jej mohli snadno vyhledat a udělat z něj modální dialogové okno s upozorněním pro uživatele. Obsah atributu `title` poslouží jako titulek nového okna, což si popíšeme později v této kapitole. Tento element samozřejmě skryjeme a celý jeho obsah zobrazíme jen tehdy, když se uživatel nepřihlásí (to ani v tomto ukázkovém příkladu nemůže) a poprvé načte obsah karty **Pokec**.

Vložení stylů a skriptů

Ačkoliv to nemusí být na první pohled jasné, vytvořili jsme už veškerý obsah našich webových stránek. To je samozřejmě dobře, protože jsme doposud pracovali jen s jazykem HTML, a to je přesně jeho účel.

Máme ale jeden problém – vzhledem k tomu, že tři předchozí soubory nejsou kompletní dokumenty HTML, nemají ani správně nastavenou znakovou sadu. Nastal tedy čas zapojit jazyk CSS a knihovnu jQuery UI a obléknout naše webové stránky do nějakého slušivého kabátu.

Kde bychom měli začít? Bezesporu vložení šablon kaskádových stylů a skriptů. Vrátime se tedy k souboru *index.html* a rozšíříme element `head` o následující řádky zvýrazněné tučným písmem:

```

...
<head>

```

```

<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<meta http-equiv="content-language" content="cs" />
<title>Kapitola 4 - příklad Pro zasmání</title>
<link type="text/css" href="css/start/jquery-ui.css" rel="stylesheet" />
<link type="text/css" href="css/hlavni.css" rel="stylesheet" />
<script type="text/javascript" src="js/jquery.js"></script>
<script type="text/javascript" src="js/jquery-ui.js"></script>
<script type="text/javascript" src="js/hlavni.js"></script>
</head>
...

```

Povšimněte si, že šablony kaskádových stylů vkládáte před skripty jazyka JavaScript. Přestože se to může zdát jako náhoda, opak je pravdou. Vždy vkládejte šablony kaskádových stylů před skripty. Důvodem je způsob, jakým webové prohlížeče zpracovávají dokumenty HTML, a externí prostředky, jako jsou kupříkladu právě šablony kaskádových stylů a skripty. Webový prohlížeč totiž postupuje (dle očekávání) shora dolů, proto nejprve načítá výše položené řádky. Výsledkem tudíž je, že šablony kaskádových stylů zpracuje před skripty jazyka JavaScript, což je žádoucí. Ve skriptech jazyka JavaScript se totiž můžeme spoléhat na některé vlastnosti elementů, které by mohly kaskádové styly změnit.

```
<link type="text/css" href="css/start/jquery-ui.css" rel="stylesheet" />
```

Vkládáme dříve přichystanou šablonu kaskádových stylů pro motiv vzhledu Start knihovny jQuery UI.

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

Jedná se o šablonu kaskádových stylů pro naše webové stránky. Říkáte si, že soubor *hlavni.css* neexistuje? Máte pravdu. Hned to napravíme a vytvoříme nový soubor *hlavni.css* v podsložce *css* složky příkladu.

```
<script type="text/javascript" src="js/jquery.js"></script>
```

Předchozím řádkem vkládáme skript knihovny jQuery.

```
<script type="text/javascript" src="js/jquery-ui.js"></script>
```

Zde je konečně skript knihovny jQuery UI. Na pořadí opět záleží – skript knihovny jQuery UI musíme vložit až po skriptu knihovny jQuery, jelikož základem knihovny jQuery UI je knihovna jQuery. Stejně tak musíme vložit níže uvedený hlavní skript našich webových stránek až po skriptu knihovny jQuery UI, protože se jí chystáme v našem skriptu používat:

```
<script type="text/javascript" src="js/hlavni.js"></script>
```

Tento skript také ještě neexistuje. Proto vytvoříme nový soubor *hlavni.js* v podsložce *js* složky příkladu.

Nyní se vysvětlíme, jak s pomocí knihovny jQuery UI vytvořit pěknou hlavní nabídku na svých webových stránkách.

Tvorba hlavní nabídky

Chceme z neuspořádaného seznamu s odkazy udělat hlavní nabídku, a ne jen tak ledajakou. Jednotlivé části webu rozdělíme do karet, přičemž každou kartu budeme načítat technologií AJAX. K tomuto účelu nabízí knihovna jQuery UI ovládací prvek Tabs.

Ovládací prvek Tabs

Ovládací prvek Tabs slouží k rozčlenění obsahu na logické části (karty), čímž uspoří místo v designu stránky a zvyšuje přehlednost informací. Svým zaměřením se tak tento ovládací prvek nejvíce podobá ovládacímu prvku Accordion, který si popíšeme později v této kapitole.

Uživatel většinou otevírá další kartu klepnutím tlačítkem myši. Alternativní řešením je však otevřít kartu, pokud nad ni uživatel přesune ukazatel myši.

Obsah jednotlivých karet se může nacházet buď přímo ve stránce, nebo ho můžeme načítat dynamicky z externích souborů pomocí technologie AJAX.

Základní struktura elementů jazyka HTML

Abychom mohli aplikovat ovládací prvek Tabs, musíme si uspořádat obsah do elementů jazyka HTML tak, aby tomuto prvku vyhovoval. Kód jazyka HTML byste měli napsat přibližně takto:

```
<div id="karty">
  <ul>
    <li><a href="#karta1">První karta</a></li>
    <li><a href="#karta2">Druhá karta</a></li>
    <li><a href="#karta3">Třetí karta</a></li>
  </ul>
  <div id="karta1">
    <p>Obsah první karty.</p>
  </div>
  <div id="karta2">
    <p>První odstavec druhé karty.</p>
    <p>Druhý odstavec druhé karty.</p>
  </div>
  <div id="karta3">
    <p>Obsah třetí karty.</p>
  </div>
</div>
```

Veškerý obsah jsme obalili elementem `div` s identifikátorem karty. Právě na něj budeme aplikovat ovládací prvek Tabs. Identifikátory můžeme volit dle vlastního uvážení – snažíme se volit výstižné názvy.

Prvním dceřiným elementem je element `ul` (neuspořádaný seznam) se svými elementy `li` (položkami). Každá položka seznamu obsahuje element `a` (odkaz). Z obsahu tohoto elementu se stane popis karet. Hodnota atributu `href` tohoto elementu a by však rovněž neměla ujít naší pozornosti. Obsahuje-li tento atribut hodnotu začínající znakem `#` (kotvu), z elementu `div` s identifikátorem shodujícím se s touto kotvou (avšak bez znaku `#`) se stane obsah

dané karty. Jestliže má kupříkladu tento atribut href hodnotu #karta1, spojí se s obsahem elementu div s identifikátorem karta1.

Element div, z něž se má stát obsah karty, může obsahovat jakékoliv elementy. Ve výše uvedeném kódu jsme do těchto elementů (s identifikátory karta1, karta2 a karta3) vložili jeden nebo dva elementy p (odstavce), ale nic nám nebrání vložit obrázek, formulář, tabulku nebo jakýkoliv jiný element jazyka HTML.

NAČÍTÁNÍ OBSAHU KARTY POMOCÍ TECHNOLOGIE AJAX

Hodnota atributu href u odkazu, jenž reprezentuje popis karta, by mohla vypadat i jinak. Kdybychom do něj místo kotvy napsali název souboru, knihovna jQuery UI by jako obsah příslušné karty načetla obsah tohoto souboru. V této kapitole už jsme mnohokrát narazili na zmínku o technologii AJAX a nyní se s ní konečně setkáme v praxi. Prohlédněte si následující řádek:

```
<li><a href="karta1.html">První karta</a></li>
```

Kdyby uživatel klepnul na kartu **První karta**, knihovna jQuery UI by načetla obsah souboru *karta1.html* v aktuální složce pomocí technologie AJAX a zobrazila jej jako obsah této karty. Tento postup se nevylučuje s předchozím postupem s kotvami – v rámci jedné aplikace ovládacího prvku Tabs je možné kombinovat oba přístupy.

V našem příkladu už jsme vytvořili element s vhodným uspořádáním pro ovládací prvek Tabs, a to element div s identifikátorem nabídka v souboru *index.html*.

Vytvoření ovládacího prvku Tabs

Ovládací prvek Tabs vytvoříme tak, že ve skriptu jazyka JavaScript vybereme obalující element pro karty pomocí funkce jQuery() a zavoláme na něj metodu tabs(). Například takto:

```
$(function() {
    $('#karty').tabs();
});
```

Abychom mohli vyhledat element, musíme počkat na načtení modelu DOM, proto v tomto kódu píšeme:

```
$(function() {
    ...
});
```

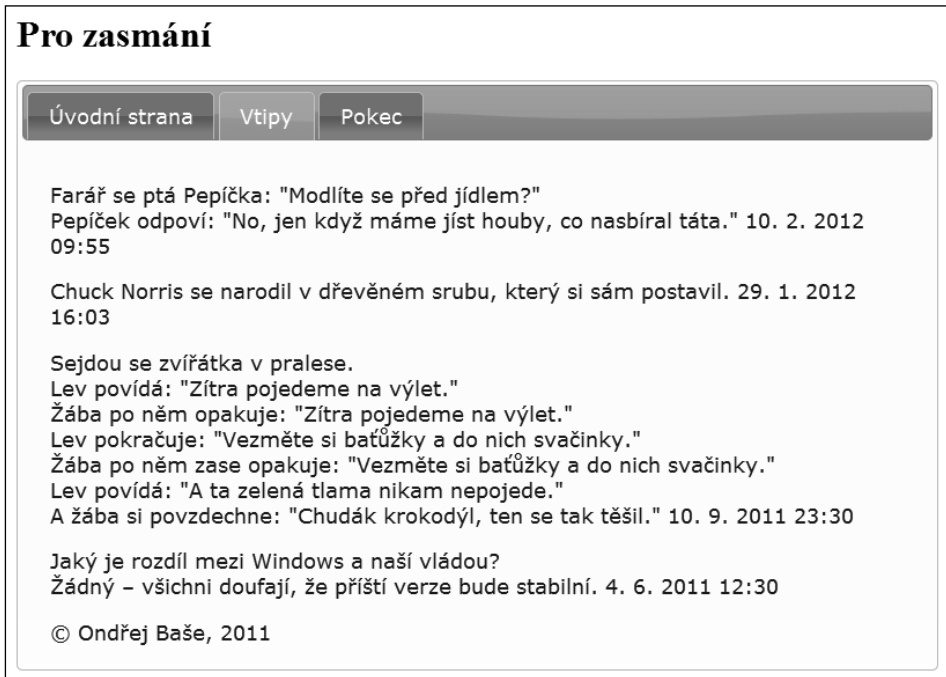
Ve skutečnosti děláme to, že voláme funkci jQuery() a předáváme jí anonymní (bezejmennou) funkci. Kód uvnitř této funkce se tudíž provede až po kompletním načtení modelu DOM, a proto můžeme začít hledat elementy.

V tomto případě hledáme element s identifikátorem karty. Z tohoto důvodu zavoláme funkci jQuery() a předáme jí selektor #karty. Na tento vybraný element posléze voláme metodu tabs(), a tím jej přeměníme na skupinu karet.

V našem příkladu aplikujeme ovládací prvek Tabs na element s identifikátorem nabídka. Otevřeme si tudíž soubor *hlavni.js* a vložíme do něj níže uvedený kód:


```
$(function() {
    $('#nabidka').tabs();
});
```

Výsledek přidání tohoto kódu je poměrně ohromující, jak je patrné na obrázku 4.3.



Obrázek 4.3: Hlavní nabídka webových stránek ve formě karet

Co se vlastně stalo? Z jednoduchých položek neuspořádaného seznamu se staly plně funkční karty. Kromě toho začala knihovna jQuery UI načítat kód ze souborů *uvodni-strana.html*, *vtipy.html* a *pokec.html* pomocí technologie AJAX a zobrazuje ho do stránky jako obsah jednotlivých karet. Na předchozím obrázku lze spatřit načtený obsah souboru *vtipy.html* v záložce **Vtipy**.

Zkuste klepat na popisky jednotlivých karet a zjistíte, že vše funguje, jak má.

Možnosti nastavení

Všechny komponenty knihovny jQuery UI lze konfigurovat a ovládací prvek Tabs není žádnou výjimkou. Ovládací prvek můžeme nastavit hned při jeho aplikaci na element tak, že jeho hlavní metodě (v tomto případě `tabs()`) předáme objekt složený z dvojic obsahujících názvy nastavení a jejich hodnoty. Následuje příklad konfigurace ovládacího prvku Tabs při jeho vytváření:

```
$('#karty').tabs({
    selected: 1,
    event: 'mouseover'
});
```

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