Marián Margorín

jQuery

bez předchozích znalostí

Průvodce pro samouky

Od prvních krůčků
až po efektní aplikace

Srozumitelnou řečí bez
zbytečných detailů

Kontrolní otázky
a závěrečný test

Začněte i vy v krátkém čase
vylepšovat svůj web

CPRESS

Marián Margorín

jQuery
bez předchozích znalostí

Computer Press
Brno
2014

jQuery bez předchozích znalostí

Marián Margorín

Obálka: Martin Sodomka
Odpovědný redaktor: Martin Domes
Technický redaktor: Jiří Matoušek

Objednávky knih:
http://knihy.cpress.cz
www.albatrosmedia.cz
eshop@albatrosmedia.cz
bezplatná linka 800 555 513
ISBN 978-80-251-3379-8

Vydalo nakladatelství Computer Press v Brně roku 2014 ve společnosti Albatros Media a. s. se sídlem Na Pankráci 30, Praha 4. Číslo publikace 18 632.

Dotisk 1. vydání

ALBATROS MEDIA a.s.

Obsah

KAPITOLA 3

KAPITOLA 4

KAPITOLA 5

KAPITOLA 6

Úvodem

Koupili jste si knihu o knihovně jQuery, která je určená každému, kdo chce vytvářet vysoce interaktivní internetové aplikace. V současném moderním světě Internetu je nevyhnutelné vytvářet aplikace tak, abyste co nejvíce zpřehlednili informace pro návštěvníky svých stránek. V této knize si společně projdeme devět kapitol, v nichž se naučíme používat knihovnu jQuery. Pokud chcete dosáhnout nejlepších výsledků, čtěte tuto knihu postupně od první až po poslední kapitolu.

Abyste se naučili používat knihovnu jQuery, musíte mít zkušenosti s tvorbou internetových aplikací. Nemusíte se ale bát – postačí vám základní znalost jazyka HTML. Je samozřejmě výhodu, jestliže jste dříve používali programovací jazyk JavaScript, na němž je tato knihovna postavená. Znalost tohoto programovacího jazyka však není podmínkou k tomu, abyste se naučili používat knihovnu jQuery. Pokud jste už četli nějakou publikaci o knihovně jQuery, můžete přeskočit úvodní kapitoly, které se věnují základnímu způsobu použití této knihovny.

Na konci této knihy najdete závěrečný test, jenž obsahuje praktické otázky týkající se témat ze všech kapitol. Na tento test byste měli odpovídat až po přečtení všech kapitol. Za uspokojivý výsledek lze považovat 75 procent správných odpovědí.

Pro dosažení co nejlepšího výsledku je vhodné číst tuto knihu jednu až dvě hodiny denně. Tímto tempem budete vstřebávat informace nejefektivněji. Celou publikaci je možné přečíst a pochopit v průběhu několika měsíců. Dále ji můžete používat jako příručku při vývoji v knihovně jQuery.

Zpětná vazba od čtenářů

Nakladatelství a vydavatelství Computer Press stojí o zpětnou vazbu ke knize a bude na vaše podněty a dotazy reagovat. Můžete se obrátit na následující adresy:

redakce PC literatury
Computer Press
Spielberk Office Centre
Holandská 3
639 00 Brno

nebo

sefredaktor.pc@cpress.cz

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

Zdrojové kódy ke knize

Z adresy http://knihy.cpress.cz/k1897 si po klepnutí na odkaz Soubory ke stažení můžete přímo stáhnout archiv s ukázkovými kódy.

Errata

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

Veškerá existující errata zobrazíte na adrese http://knihy.cpress.cz/k1897 po klepnutí na odkaz Soubory ke stažení.

Co budeme potřebovat

Než napíšeme zdrojový kód v knihovně jQuery, popíšeme si vše, co budete potřebovat při práci s touto knihovnou. Knihovna jQuery se zakládá na velmi známém programovacím jazyce JavaScript. Tato knihovna výrazně ulehčuje programátorům tvorbu JavaScriptových aplikací. Cílem knihy je naučit se psát méně kódu, ale přitom dosáhnout stejné funkčnosti, jako bychom psali kód čistě v jazyce JavaScript. Kromě tohoto tato kniha nabízí velké množství vylepšení v podobě různých efektů a zjednodušení.

Co knihovna jQuery nabízí

Knihovna jQuery je natolik propracovaná, že s ní můžeme vytvářet nejrůznější animace na velmi dobré úrovni. Ve spoustě situací nebudeme muset vytvářet animace pomocí technologie Flash. Díky tomu se budou naše stránky rychleji načítat a budou přístupnější pro internetové vyhledávače.

Tato kniha je určená pro všechny, kteří se zajímají o tvorbu interaktivních webových aplikací. V současné době jsou už internetové technologie velmi vyspělé a je nevyhnutelné vytvářet internetové aplikace interaktivně. Jestliže zapomeneme na své zákazníky anebo na své uživatele, vystavujeme se vážné hrozbě, že naše stránky nepřinesou takový užitek, jaký jsme předpokládali. Takový postup uživatele je pochopitelný, ačkoliv nikdo z nás není spokojený, jestliže musí kupříkladu procházet velké množství dat s vědomím, že se mu stránka načítá neustále dokola.

Dalším typickým příkladem je validace formulářových dat. Je velmi nepohodlné, když musíme vyplňovat formulář, aniž bychom věděli, co přesně od nás jeho autor požaduje. Drtivá většina uživatelů udělá při vyplňování formuláře chybu a opakované odesílání formuláře s chybou je může přivést k zoufalství. Samozřejmě je možné přidat k takovému formuláři statický popis toho, co chceme vyplnit, ale tato informace postačuje málokdy. Nejúčinnějším a nejpohodlnějším řešení je zobrazovat uživate-

lům aktuální stav validace. Takto uživatel okamžitě pozná, kde a kdy nastal problém, díky čemuž jej může včas opravit.

V průběhu celé této knihy budeme pracovat s knihovnou jQuery a s technologií Ajax, s níž si tato knihovna umí velmi jednoduše a pohodlně poradit. Přenos dat mezi prohlížečem a serverem bude tudíž probíhat tak, že návštěvník našich stránek nic nepozná. Jediné, čeho si všimne, budou jím požadované informace, přičemž stránka nebude nijak blikat a skákat nahoru a dolů.

Editor zdrojového kódu

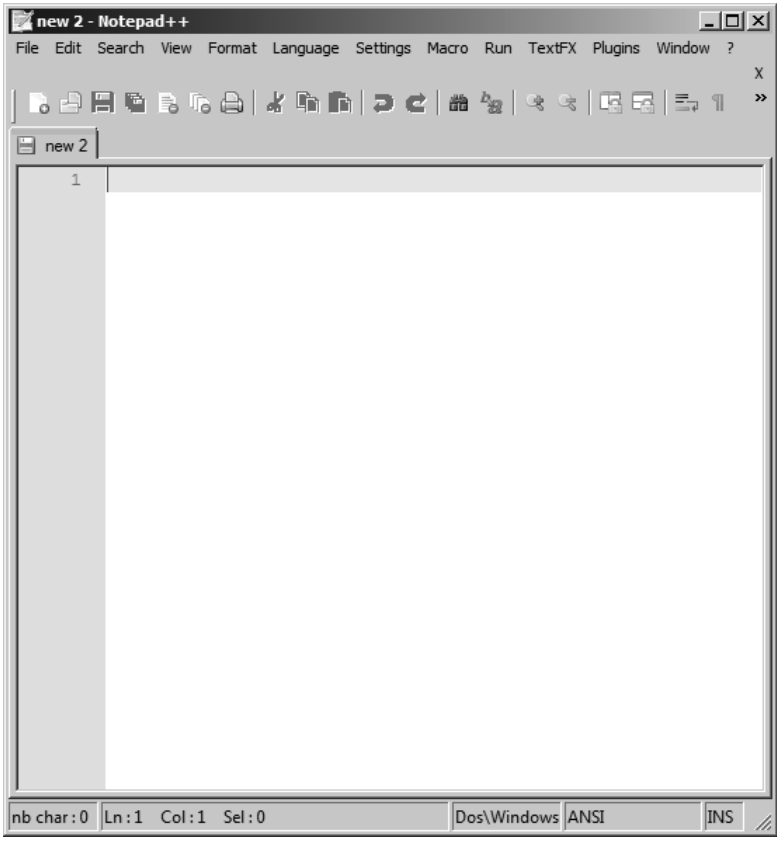
Představme si první a současně i nejdůležitější nástroj, který budeme při programování potřebovat. Jedná se o textový editor Notepad++, v němž budeme psát náš zdrojový kód. Výhodou tohoto textového editoru je, že nám zvýrazní napsaný kód, abychom se v něm snadněji vyznali. Textový editor Notepad++ můžete stáhnout zdarma z internetové adresy *http://notepad-plus-plus.org/*. Po stažení a nainstalování instalačního balíku získáte kvalitní textový editor.

Editor Notepad++ je bezplatný editor zdrojového kódu s podporou více programovacích jazyků. Můžeme v něm kupříkladu psát zdrojový kód v jazycích PHP, C++, JavaScript apod. Na obrázku 1.1 vidíte nasnímanou úvodní obrazovku této aplikace. Pevně doufám, že budete s programováním v této aplikaci nadmíru spokojeni.

Webový server

Textový editor Notepad++ nám bude při programování v knihovně jQuery stačit, dokud budeme ve svých aplikacích používat čistě jen jazyk HTML. Jestliže ale budeme své internetové aplikace vytvářet v některém skriptovacím jazyce (například PHP, ASP nebo JSP), budeme potřebovat také jejich součásti. Všechny příklady z této knihy vykonávané na straně serveru pracují s jazykem PHP. Proto se musíte seznámit s nástrojem, který je nezbytný k tomu, abyste mohli tyto příklady otestovat na svém lokálním počítači.

Velmi jednoduchý na instalaci a ovládání je program XAMPP, který si můžete stáhnout bezplatně z internetové adresy *http://www.apachefriends.org/en/index.html*. Instalací tohoto balíku si do počítače zavedete webový server Apache, který je pro programování v jazyce PHP nevyhnutelný. Zároveň s ním získáte interpreter jazyka



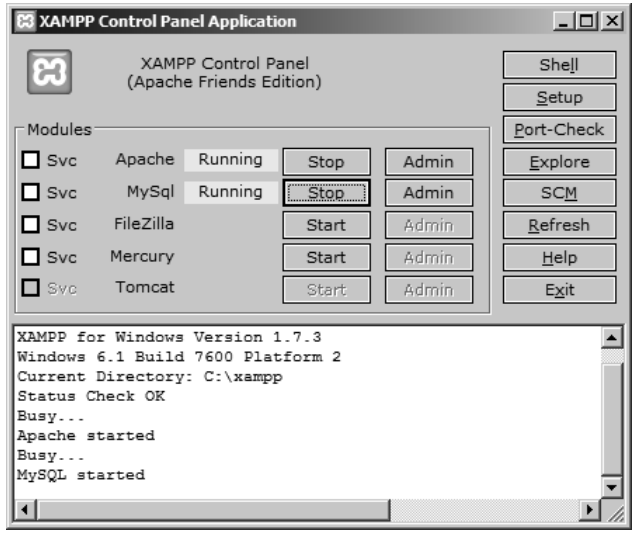
Obrázek 1.1. Úvodní obrazovka programu Notepad++

PHP, databázový systém MySQL a nástroj pro jeho správu s názvem PhpMyAdmin. Balík XAMPP nainstalujete velmi jednoduše:

1. Stáhněte si instalační balík z adresy *http://www.apachefriends.org*.
2. Spusťte instalační balík, který vás provede zbytkem instalace.
3. Tento instalační balík nainstaluje všechny potřebné součásti, které budete při práci potřebovat.

Jakmile se balík XAMPP nainstaluje, nainstaluje také svůj kontrolní panel, s nímž můžete podle potřeby zapínat a vypínat jednotlivé služby. Tento kontrolní panel si můžete prohlédnout na obrázku 1.2.

Upozornění: Při spouštění webového serveru si dejte pozor na aplikace spuštěné na svém počítači. Některé programy (například program Skype) mohou blokovat spuštění webového serveru, protože používají stejný komunikační port.

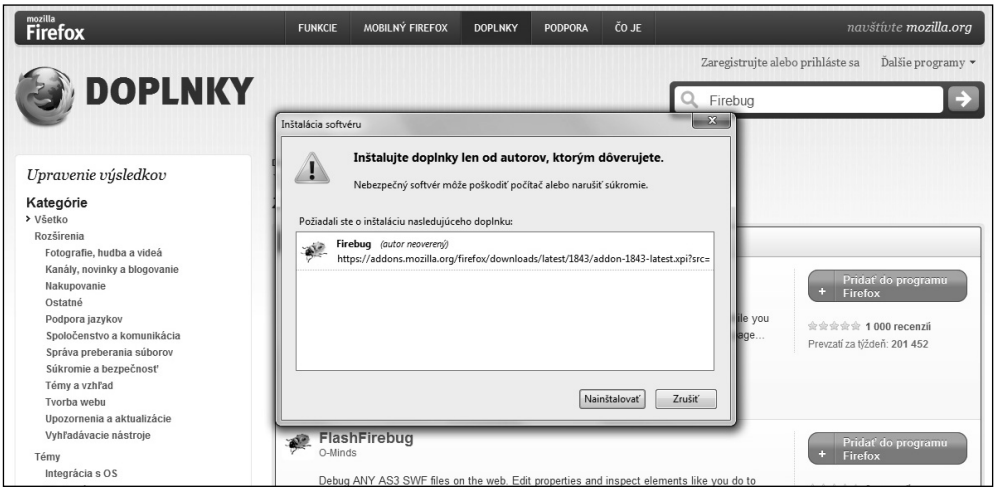


Obrázek 1.2. Kontrolní panel programu XAMPP

Software pro ladění kódu

Už máme k dispozici nástroje pro programování na straně serveru. Můžeme tudíž začít plně využívat veškerou sílu, kterou nám knihovna jQuery nabízí. Při vytváření internetových aplikací ale zajisté uděláme mnoho chyb, které budeme muset odladit. Rovněž budeme muset odladit různé části programu, abychom se na ně mohli spolehat. Pro usnadnění trápení použijeme ladicí nástroje určené přesně k tomuto účelu.

Vzhledem k tomu, že základem knihovny jQuery je jazyk JavaScript, budeme svůj zdrojový kód provádět ve webovém prohlížeči. V této části kapitoly si popíšeme ladicí nástroje pro dva nejrozšířenější webové prohlížeče, a to prohlížeč Mozilla Firefox a prohlížeč Internet Explorer. Začneme s ladicím nástrojem pro prohlížeč Mozilla Firefox s názvem Firebug.



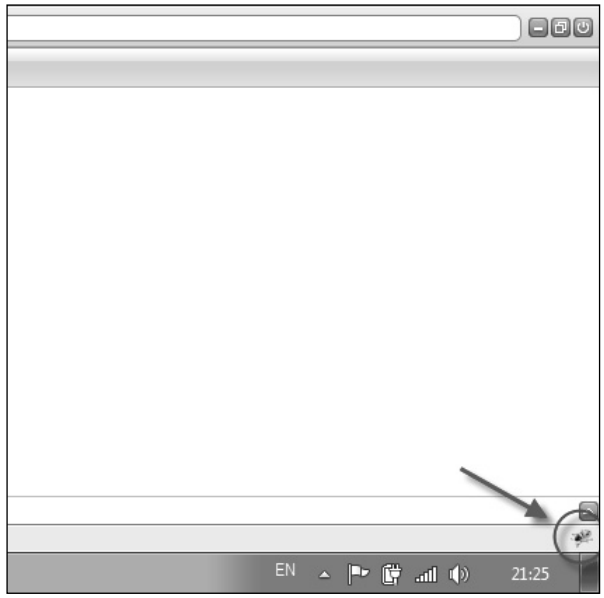
Obrázek 1.3. Přidání doplňku Firebug

Firebug

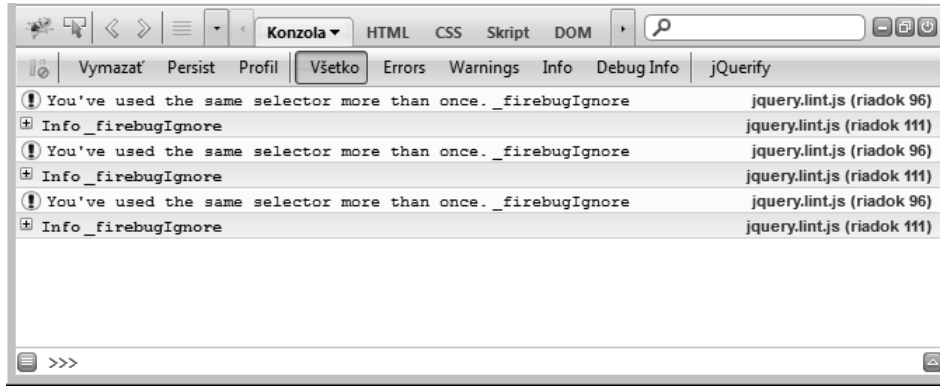
Ladicí nástroj Firebug nainstalujete do prohlížeče Mozilla Firefox ve formě jeho doplňku. Navštivte internetovou adresu *https://addons.mozilla.org/cs/firefox/?browse=featured*, na níž se nacházejí doplňky tohoto webového prohlížeče. Pomocí vyhledávacího pole vyhledejte rozšíření Firebug a přidejte si jej do prohlížeče Firefox. Tentokrát nemusíte stahovat žádný instalační balík, jelikož instalace proběhne v prohlížeči. Když přidáváte doplněk, měli byste si ale dát pozor, abyste měli povolená vyskakovací okna – jinak by se instalace nespustila. Po nainstalování doplňku se prohlížeč restartuje. Po restartu už bude váš doplněk plně k dispozici.

V pracovním prostředí prohlížeče Firefox nám po nainstalování doplňku Firebug přibude tlačítko v pravém dolním rohu okna. S jeho pomocí lze snadno zapínat tento kvalitní nástroj. Z obrázku 1.4 je patrné, kde se toto tlačítko nachází.

Velmi silným nástrojem aplikace Firebug je jeho konzola. Po spuštění této aplikace hned vidíme, kde nastala chyba a můžeme ji opravit. Možnosti konzoly si můžete prohlédnout na obrázku 1.5.



Obrázek 1.4. Spuštění aplikace Firebug



Obrázek 1.5. Konzola aplikace Firebug

Dalším mocným nástrojem tohoto doplňku je ladicí nástroj, s jehož pomocí můžeme nastavovat kontrolní body. Posléze je možné provádění zdrojového kódu pozastavit na tomto bodu, potom jej znovu spustit a sledovat, kde vzniká chyba. Můžeme taktéž sledovat hodnotu vybrané proměnné po dobu běhu skriptu.

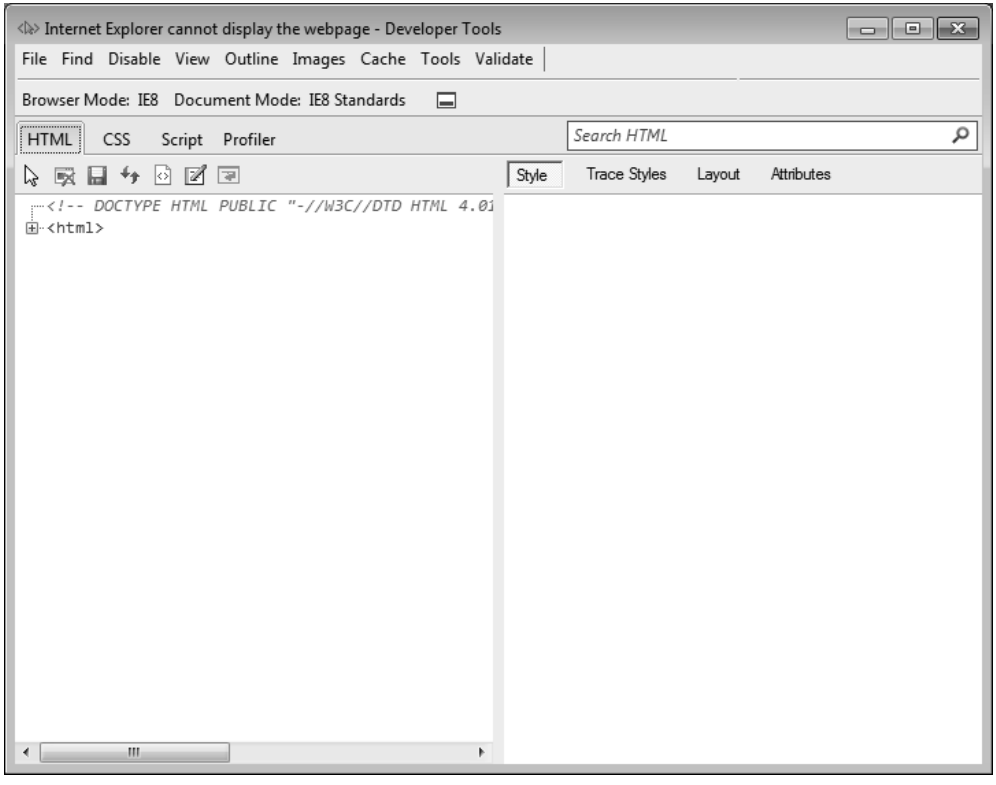
FireQuery

Jakmile úspěšně nainstalujete doplněk Firebug, vraťte se na výše uvedenou internetovou adresu s rozšířeními. Tentokrát vyhledejte doplněk pojmenovaný FireQuery, který rozšiřuje funkčnost doplňku Firebug pro zdrojové kódy psané v knihovně jQuery.

Developer Toolbar

Uživatelé webového prohlížeče Internet Explorer musí ladit svou aplikaci v jiném nástroji než je doplněk Firebug. V této části kapitoly si krátce představíme nástroj Internet Explorer Developer Toolbar od společnosti Microsoft, který se nachází na internetové adrese *http://www.microsoft.com/downloads/*. Tento nástroj je v prohlížeči Internet Explorer dostupný hned po instalaci.

Poznámka: Nástroj Internet Explorer Developer Toolbar se dodává s prohlížečem Internet Explorer od verze 8. V případě starší verze tohoto prohlížeče je nutné jej nejprve stáhnout a nainstalovat.



Obrázek 1.6. Nástroj Internet Explorer Developer Toolbar

Tento ladicí nástroj spustíme stiskem klávesy *F12*; případně jej najdeme v nabídce prohlížeče Internet Explorer. Nástroj se otevře v novém okně a my se můžeme pustit do práce. Internet Explorer Developer Toolbar lze vidět na obrázku 1.6.

Funkce tohoto nástroje se velmi podobají schopnostem nástroje Firebug, který jsme si popsali výše. Mezi jeho nejdůležitější funkce patří:

- vyhledávání a zobrazování jednotlivých elementů modelu DOM (objektový model dokumentu),
- rozšířená práce se souborem kaskádových stylů,
- ladění jednotlivých částí zdrojového kódu.

Připravili jsme si všechny nástroje, které potřebujeme, abychom mohli začít pracovat s knihovnou jQuery. Aby byl náš seznam nástrojů kompletní, musíme si obstarat samotnou knihovnu jQuery. Shánění této knihovny si ale necháme na později – konkrétně na kapitolu 3, v níž se budeme věnovat prvnímu zdrojovému kódu napsanému v této knihovně.

Test

1. Jaké výhody má textový editor Notepad++?
2. Co všechno získáme instalací balíku XAMPP?
3. Jaký ladicí nástroj můžeme použít při práci s prohlížečem Mozilla Firefox?
4. Jak se nazývá doplněk, kterým rozšíříme funkčnost nástroje Firebug o knihovnu jQuery?
5. Jaký ladicí nástroj použijeme při práci s prohlížečem Internet Explorer?

Úvod do jQuery

Cílem této kapitoly je představit knihovnu jQuery. Povíme si, co všechno tato knihovna dokáže a samozřejmě si uvedeme, jaké má výhody a nevýhody. Abychom se o knihovně jQuery nebavili pouze teoreticky, ukážeme si některé situace, s nimiž se během vývoje v knihovně jQuery můžeme setkat. Naučíme se předcházet konfliktům, pokud chceme ve své aplikaci pracovat s více knihovnami. Ukážeme si také, jak používat nástroj Firebug – konkrétně jeho konzolu a způsob, jak můžeme vypisovat své proměnné.

V této kapitole si rovněž povíme, které významné společnosti používají knihovnu jQuery pro vývoj svých webových aplikací. Knihovna jQuery se zakládá na programovacím jazyku JavaScript, přičemž v době psaní této publikace se knihovna jQuery pyšnila prvenstvím v používání. Neztrácejme tedy čas a pojďme se s touto oblíbenou knihovnou seznámit.

Knihovna jQuery – seznamte se

Knihovna jQuery klade důraz na interakci mezi značkovacím jazykem HTML a programovacím jazykem JavaScript. Tato knihovna je rychlá a přesná, přičemž zjednodušuje tvorbu a správu událostí, animací a spousty dalších komponent, které s sebou přináší nejmodernější technologie. Poprvé ji představil John Resig v roce 2006. Knihovna jQuery se šíří pod licencemi GPL a MIT, což znamená, že ji můžeme volně používat ve svých projektech. Rovněž se můžeme díky těmto licencím zapojit do vývoje knihovny jQuery.

Poznámka: Licence GNU – GPL (General Public License) je licence pro šíření svobodného softwaru. Díky této licenci můžeme používat knihovnu pro jakýkoliv účel. Můžeme také šířit její kopie, měnit její zdrojový kód a vylepšovat ji. Mezi nejvýznamnější produkty šířené pod licencí GPL patří programovací jazyk Perl a operační systém Linux.

Kromě toho existuje velké množství zásuvných modulů. Zkušenější programátoři si můžou vytvářet a distribuovat své vlastní zásuvné moduly. Knihovna jQuery obsahuje spoustu funkcí, které zajisté při vývoji svých aplikací použijeme. Síla knihovny jQuery spočívá převážně ve výběru jednotlivých elementů v rámci modelu DOM. Knihovna jQuery vybírá tyto elementy s pomocí selektorového jádra Sizzle, které si můžete prohlédnout na internetové adrese *http://sizzlejs.com*.

Další silnou zbraní knihovny jQuery je schopnost modifikovat jednotlivé elementy modelu DOM. Můžeme tak kupříkladu jednoduše a rychle měnit styly jazyka CSS pro jednotlivé elementy.

Knihovnu jQuery bychom nemohli ale plnohodnotně používat, kdybychom s ní nemohli zpracovávat jednotlivé události. Představme si například, že budeme chtít zobrazit uživateli informaci, jakmile přesune ukazatel myši nad předem stanovený objekt. Tuto akci se nám nepodaří provést, aniž bychom zpracovali událost `onMouseOver`; případně by bylo toto zpracování velmi složité. Je však jasné, že knihovna jQuery nám tuto možnost nabízí, a navíc přináší řadu vylepšení, jež při své práci oceníme.

Pravděpodobně nejzajímavější částí knihovny jQuery je široká škála efektů. Jejich použití je jednoduché, a přitom dosáhneme skvělého výsledku. Nemusíme už psát složité kusy zdrojového kódu v jazyku JavaScript a trápit se jejich laděním tak, aby fungovaly správně ve všech webových prohlížečích. V knihovně jQuery jednoduše zavoláme požadovaný efekt a tato knihovna provede veškerou práci za nás. Kromě toho v ní lze vytvářet různé animace. Vyhneme se tak často zavedení technologie Flash, jejíž použití rovněž vyžaduje znalosti o tvorbě animací.

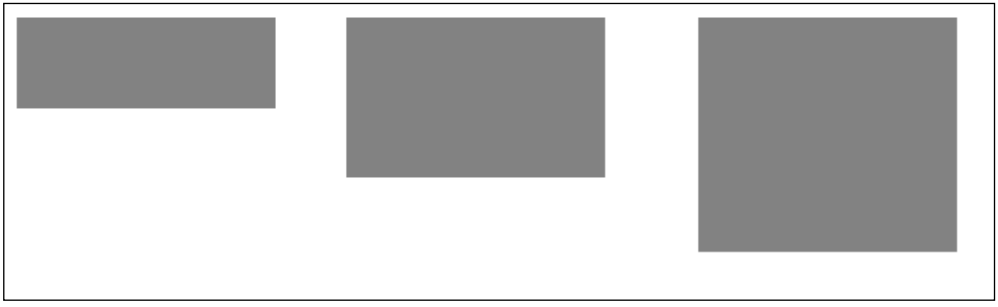
Poznámka: Technologie Flash je multimediální platforma založená na vektorové grafice, která slouží k tvorbě interaktivních aplikací. V této technologii je také možné vytvářet plnohodnotné internetové prezentace.

Příklad jednoduchého skriptu knihovny jQuery

Náš zdrojový kód bude tedy přehlednější a dostupnější pro vyhledávače. Prohlédněte si krátký příklad, kterému nejspíš nebudete ještě rozumět. V této chvíli se nezabývejte tím, jak kód funguje – na to bude čas v následujících kapitolách. Podívejte se na efekt vytvořený prostřednictvím knihovny jQuery, jehož kód najdete ve výpise 2.1, a výslednou animaci na obrázku 2.1.

Výpis 2.1. Animace v knihovně jQuery

```
$("#priklad").animate({
   left: '500px',
   height: '150px'
}, 500);
```



Obrázek 2.1. Animace v knihovně jQuery

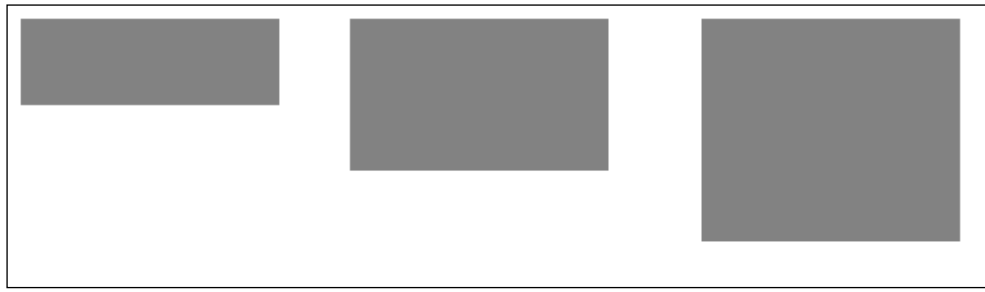
Pro srovnání si prohlédněte stejnou animaci vytvořenou v jazyce JavaScript, a to ve výpise 2.2. Rozdíl je dozajista patrný na první pohled. Kromě toho tento příklad nebere v úvahu různé prohlížeče, proto by mohl být výsledný kód v praxi ještě delší. Nemluvě ani o tom, že na tvorbu složitějších animací v jazyce JavaScript potřebuje programátor velmi slušné znalosti. Na tuto animaci se můžete podívat na obrázku 2.2.

Výpis 2.2. Animace v JavaScriptu

```
var test = null;
function pohyb() {
   test.style.left = parseInt(test.style.left)+500+'px';
   test.style.height = parseInt(test.style.height)+150+'px';
   setTimeout(pohyb,20);
}

function init() {
   test = document.getElementById('priklad');
   test.style.top = '0px';
   test.style.left = '0px';
   pohyb();
}

window.onload = init;
```



Obrázek 2.2. Animace v JavaScriptu

Knihovna jQuery a technologie AJAX

Moderní Internet požaduje, aby webové aplikace byly interaktivní. Abychom toho dosáhli, musíme do svých aplikací zavést technologii AJAX. K jejímu použití ve svých aplikacích nepotřebujeme knihovnu jQuery, protože se jedná o samostatnou programátorskou techniku. Tento způsob by byl ale zbytečně komplikovaný, jelikož knihovna jQuery spolupracuje přímo s technologií AJAX a výrazně zjednodušuje práci s ní.

Nyní si předvedeme, jak vypadá kód napsaný čistě pomocí technologie AJAX a jak vypadá odpovídající kód napsaný v knihovně jQuery. Začneme zdrojovým kódem napsaným pouze prostřednictvím technologie AJAX. Tento zdrojový kód se nachází ve výpise 2.3. Ani tentokrát není nutné pochopit daný zdrojový kód. Důležité je porovnat dva zdrojové kódy, které vykonávají stejnou činnost, ale jsou napsané jinými technologiemi.

Výpis 2.3. Zdrojový kód napsaný výhradně technologií AJAX

```
// Ošetření požadavků v prohlížeči Mozilla Firefox
if (window.XMLHttpRequest) {
   xmlHttpReq = new XMLHttpRequest();
}
// Ošetření požadavků v prohlížeči Internet Explorer
else if (window.ActiveXObject) {
   xmlHttpReq = new ActiveXObject("Microsoft.XMLHTTP");
}
xmlHttpReq.open('POST', ajax.php, true);
xmlHttpReq.setRequestHeader('Content-Type', 'application/x-www-form-urlen-
coded');
xmlHttpReq.onreadystatechange = function() {
   if(xmlHttpReq.readyState == 4) {
      // akce
   }
}
xmlHttpReq.send("data");
```

Na první pohled je jasné, že kód napsaný pouze pomocí technologie AJAX je poměrně rozsáhlý a složitý. Samozřejmě nás nutí přemýšlet nad jeho kompatibilitou, a to podle toho, jestli chceme, aby fungoval ve všech prohlížečích stejně. Každý takový problém znepříjemňuje naši práci a prodlužuje náš projekt.

Výpis 2.4. Odpovídající zdrojový kód jako ve výpisu 2.3, ale napsaný v knihovně jQuery

```
$(document).ready(function(){
   $('#hodnota').keyup(function(){
      $.post("ajax.php",{
         $(this).val() }, function(data) {
            $('#element').html(data.returnValue);
      }, "json");
   });
});
```

Zdrojový kód psaný v knihovně jQuery je mnohem jednodušší a výrazně kratší. Podrobněji si technologii AJAX v knihovně jQuery vysvětlíme v kapitole 7, která se věnuje této problematice. Poslední důležitou a velmi silnou stránkou knihovny jQuery je používání zásuvných modulů.

Zásuvné moduly knihovny jQuery

Zásuvné moduly knihovny jQuery nám umožňují používat ucelené balíky hotových řešení. Můžeme tudíž ve svých aplikacích uplatnit kupříkladu hotovou validaci formulářů, galerii, přepínací nabídku a spoustu dalších věcí. Jak už víme ze začátku této kapitoly, v knihovně jQuery můžeme vytvářet vlastní zásuvné moduly podle svých představ a potřeb. Zásuvné moduly pro tuto knihovnu si popíšeme podrobněji v kapitole 8.

Na závěr našeho seznamování s knihovnou jQuery si ukažme některé příklady, protože je lepší vidět koncepci jedenkrát v praxi, než o ní stokrát slyšet. Získáte tak lepší představu o tom, co tato knihovna dokáže, a rovněž budete mít ještě větší chuť ke studiu. Prvním příkladem je automatický posuvník nejnovějších zpráv (viz obrázek 2.3).

Obrázek 2.3. Automatický posuvník zpráv

Tento příklad je velmi užitečný, pokud chceme zobrazovat na malém místě velké množství důležitých informací. Jestliže uživatele zaujme některý název článku, jednoduše na něj klepne a zobrazí se mu jeho obsah. V knihovně jQuery není vytvoření takového efektu vůbec složité, přičemž na Internetu lze najít celou řadu řešení.

Velmi často někdo požaduje, abychom vytvořili interaktivní galerii. Pro knihovnu jQuery existuje nesmírné množství různých galerií. Předvedeme si jednu z nich, a to galerii s názvem Fancy Box. Ta umožňuje ukazovat obrázky, elementy dokumentu HTML, videa a taktéž ajaxové požadavky. Na obrázku 2.4 vidíte tuto galerii v akci.

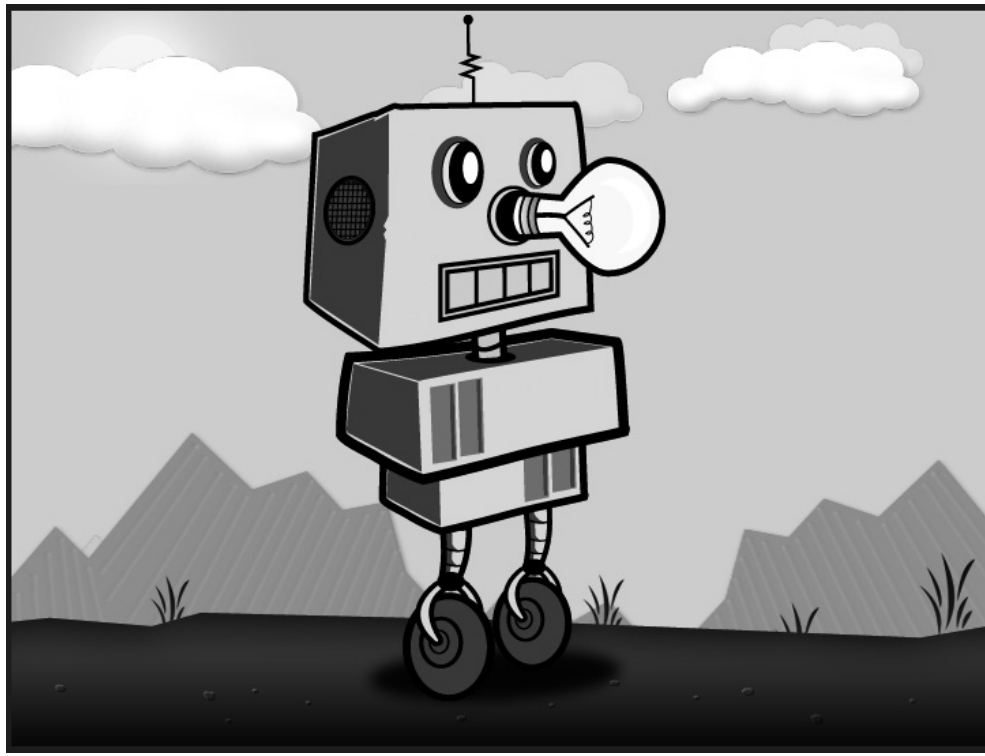
Ve svých aplikacích můžeme také používat takové působivé galerie. Tyto zásuvné moduly se totiž téměř vždy šíří pod licencí GPL, a tedy úplně zadarmo. Jejich používání je opravdu velmi jednoduché a rychlé. Navíc si můžeme libovolně upravit soubor kaskádových stylů, a přizpůsobit tak galerii svým představám.

Obrázek 2.4. Galerie Fancy Box v akci

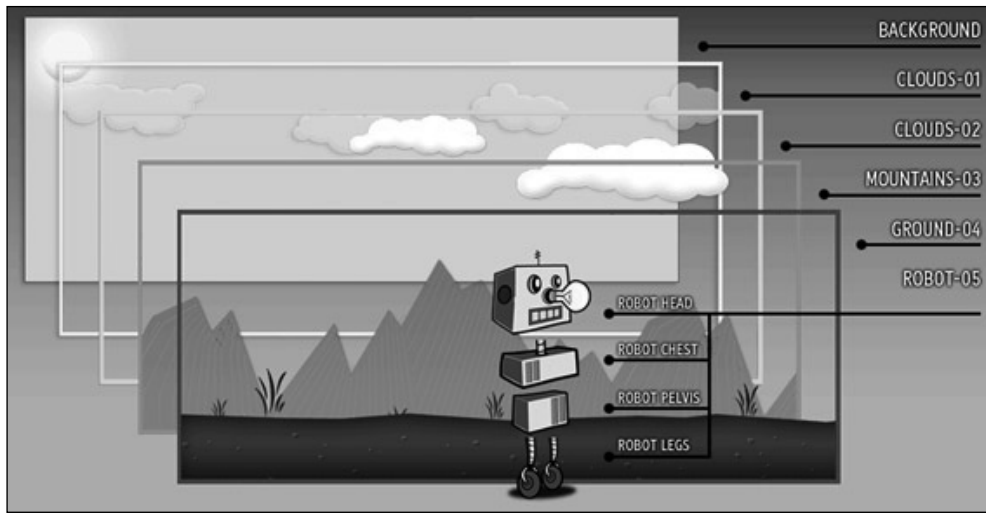
Dalším velmi zajímavým příkladem je zpracování animace v podobě pohybujícího se robota. Tato animace je velmi zábavná a současně demonstruje sílu animací, které lze vytvářet v knihovně jQuery. Můžete se tudíž inspirovat a posunout hranice svého vnímání webového designu. Jak tato animace vypadá, prozradí obrázek 2.5.

Tento příklad se skládá z více neprůhledných vrstev. Ty se skládají na sebe tak, aby tvořili souvislou vrstvu, kterou vidí návštěvník stránky. Jednotlivé vrstvy se animují samostatně, díky čemuž vzniká velmi působivý efekt pohybu. Robot se skládá podobně jako pozadí z více vrstev. Pozadí i samotného robota animuje zásuvný modul knihovny jQuery. Na dalším obrázku vidíte jednotlivé vrstvy s popisem. Obrázek 2.6 pochází z domovské stránky projektu.

Takto rozvržené průhledné vrstvy se zapisují do zdrojového kódu jazyka HTML tak, jak je patrné na výpise 2.5. Tento výpis je rovněž převzatý z domovské stránky animace. Uvádíme si ho především proto, aby bylo jasnější, jak se interpretují vrstvy ukázané na obrázku 2.6.



Obrázek 2.5. Animace pohybu robota

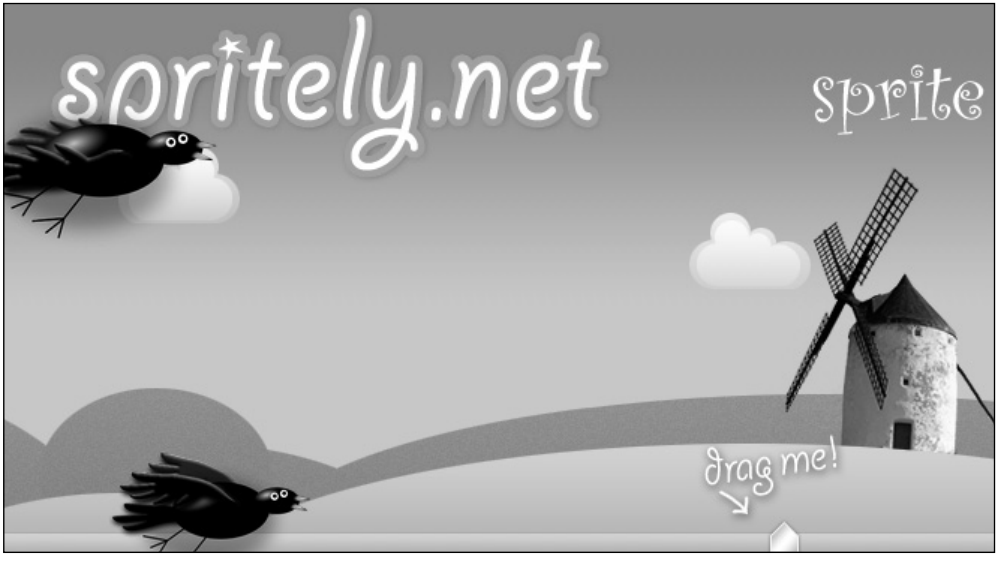


Obrázek 2.6. Rozvržení vrstev v animaci

Výpis 2.5. Rozvržení vrstev ve zdrojovém kódu

```
<div id="obal">
   <div id="obloha-01">
      <div id="obloha-02">
         <div id="hory-03">
            <div id="krajina">
            <div id="robot">
            <div id="hlava"><h1>Hlava</h1></div>
            <div id="hrud"><p>Hruď</p></div>
            <div id="panea"><p>Pánev</p></div>
            <div id="nohy"><p>Nohy</p></div>
            </div>
            </div>
         </div>
      </div>
   </div>
</div>
```

Podobným velmi působivým projektem je zásuvný modul Spritely. Jedná se o velmi jednoduchý zásuvný modul poskytující několik metod knihovny jQuery, s jejichž pomocí lze vytvářet výborné animace. Podstatou je opět jako u předchozího příkladu animace pozadí a vybraných prvků. Kromě toho tento modul skvěle spolupracuje i s mobilními technologiemi – například s iPhone, iPod apod.



Obrázek 2.7. Použití zásuvného modulu Spritely

Tento zásuvný modul je výbornou alternativou k technologii Flash a je možné jej používat i na platformě, která technologii Flash nepodporuje. Můžeme tak vytvářet stránky pouze s jazyky HTML a JavaScript. Na obrázku 2.7 vidíte ukázku použití tohoto zásuvného modulu pro tvorbu animací.

Knihovna jQuery nenabízí pouze snadnou tvorbu animací a galerií, ale také spoustu možností, jak ulehčit práci uživatelům našich stránek. Častým problémem, s nímž se při vývoji webových aplikací setkáváme, je validace formulářových dat, kterou si detailněji popíšeme v dalších kapitolách.

Obdobně se budeme věnovat situacím, v nichž uživatel musí procházet velké množství informací. Vysvětlíme si, jak zavést do svých aplikací stránkování a řazení dat.

Jako příklad si ukážeme způsob řazení dat v knihovně jQuery. Pro tuto důležitou funkčnost nám obvykle postačí správné nastavení naší tabulky a použití vhodného zásuvného modulu. Příklad řazení v knihovně jQuery si můžete prohlédnout na obrázku 2.8, který zachycuje zásuvný modul tablesorter v akci. Díky němu lze snadno a rychle řadit více sloupců tabulky, přičemž jednotlivé sloupce můžou obsahovat data jiného typu.

First Name	Last Name	Age	Total
Peter	Parker	28	$9.99
John	Hood	33	$19.99
Clark	Kent	18	$15.89
Bruce	Almighty	45	$153.19
Bruce	Evans	22	$13.19

Obrázek 2.8. Řazení v knihovně jQuery

V knihovně jQuery je rovněž možné vytvářet velmi působivé prezentace našich fotografií. Jedná se o automatické prezentace, v nichž se námi vybrané fotografie automaticky střídají, přičemž při jejich výměně se vždy používá jiný efekt. Fotografie se tudíž mění podobně jako v některých spořičích obrazovek. S pomocí knihovny jQuery můžeme vytvářet spoustu různorodých efektů, přičemž nás prakticky nic neomezuje. Taktéž je pouze na našem rozhodnutí, jak rychle budeme daný efekt provádět. Samozřejmě, lze jej spojit s fotogalerií, jestliže budeme chtít. Ukázku takového efektu najdete na obrázku 2.9.

Obrázek 2.9. Efekt automatické prezentace vytvořený knihovnou jQuery

Díky knihovně jQuery máme k dispozici opravdu spoustu možností, jak vylepšit své internetové prezentace. Kromě toho jsme v této kapitole odkryli jen malou část toho, co knihovna jQuery dokáže. I z těchto důvodů používá knihovnu jQuery neuvěřitelné množství významných firem. Uvedeme si některé z nich. Takto vypadá implementace knihovny jQuery v praxi:

- Stránka společnosti IBM, která se zabývá výpočetní technikou. Tuto stránku si můžete prohlédnout na internetové adrese *http://www.ibm.com/*.
- Stránka společnosti Dell, kterou najdete na internetové adrese *http://www.dell.com/*.
- Oficiální stránka společnosti NBC nacházející se na internetové adrese *http://www.nbc.com/*.
- Stránka redakčního systému WordPress, kterou najdete na internetové adrese *http://wordpress.org/*.

- Skvělý e-mailový klient RoundCube, který najdete na internetové adrese *http://www.roundcube.net/*.
- Stránka internetové společnosti Amazon skrývající se pod adresou *http://www.amazon.com/*.
- Knihovnu jQuery používá i vyhledávač Google, který je v současnosti považovaný za nejúspěšnější vyhledávač na Internetu.

Kombinace knihovny jQuery s jinými knihovnami

Knihovnu jQuery lze samozřejmě používat také v kombinaci s jinými knihovnami. Musíme si ale dávat pozor na některé situace, které mohou při používání více knihoven nastat. V třetí a čtvrté kapitole se naučíme používat základní funkci knihovny jQuery, kterou rovněž nazýváme funkce `jQuery()`. Její zkrácený zápis začíná znakem dolaru, jak je patrné z výpisu 2.6. Pokud používáme další knihovny, pak nám právě tento způsob zápisu může způsobit konflikt, protože znak dolaru používá spousta jiných knihoven.

Výpis 2.6. Základní funkce knihovny jQuery

```
$(document).ready(function () {
   // akce knihovny jQuery
});
```

Knihovna jQuery nám nabízí několik možností, jak tento problém vyřešit. Jednou z nich je zavolat funkci `jQuery.noConflict()`. S její pomocí jednoznačně určíme pracovní kontext, který patří knihovně jQuery. Výpis 2.7 obsahuje krátký příklad demonstrující, jak vypadá použití této funkce. V tomto příkladu používáme kromě knihovny jQuery také knihovnu Prototype, o níž si v této kapitole povíme ještě pár slov.

Výpis 2.7. Použití funkce jQuery.noConflict()

```
<script src="jquery.js"></script>
<script src="prototype.js"></script>
<script>
   jQuery.noConflict();

   jQuery(document).ready(function(){
      jQuery("element").html();
   });

   // použití knihovny Prototype
   $('element').hide();
</script>
```

Ladění kódu s Firebugem

V předchozí kapitole jsme si popsali nástroje, které budeme potřebovat pro vývoj webových aplikací v knihovně jQuery. Mezi tyto nástroje patří i ladicí nástroj Firebug. Pojďme si povědět něco o jeho konzole, která nám pomůže s řešením našich problémů. Při ladění zdrojového kódu potřebujeme velmi často zjistit hodnotu některé z proměnných. Právě pro vypsání proměnných nám velmi dobře poslouží tento nástroj. Proměnnou vypíšeme do konzole nástroje Firebug pomocí funkce `console.log()`. Příklad takové proměnné najdete ve výpisu 2.9.

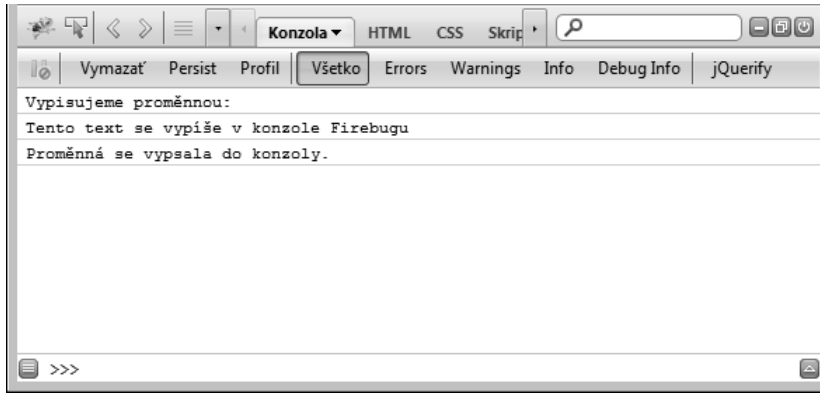
Výpis 2.9. Vypsání proměnné do konzoly nástroje Firebug

```
<script language="text/javascript">
$(document).ready(function () {
   var promenna = "Tento text se vypíše v konzole Firebugu";

   console.log("Vypisujeme proměnnou:");
   console.log(promenna);
   console.log("Proměnná se vypsala do konzoly.");
});
</script>
```

Na obrázku 2.10 se zobrazuje stránka se zapnutým nástrojem Firebug a v její konzole vidíte vypsané zprávy z předchozího překladu. Tento příklad lze vyzkoušet následujícím způsobem:

1. Vytvoříme si jednoduchou internetovou stránku, která nemusí mít ve svém těle žádné informace.
2. Následně do této jednoduché stránky opíšeme zdrojový kód z výpisu 2.9.
3. Otevřeme si připravenou stránku a zapneme nástroj Firebug tak, jak jsme si ukázali v první kapitole.
4. V konzole tohoto nástroje se zobrazí námi definované zprávy.



Obrázek 2.10. Použití konzoly nástroje Firebug

Proč knihovna jQuery

Před příchodem knihoven postavených na skriptovacím jazyce JavaScript neměli programátoři jinou možnost jak vytvářet interaktivní aplikace, než v samotném jazyce JavaScript. Tento vývoj byl několikanásobně složitější než vývoj v knihovně typu jQuery. Samotný jazyk JavaScript implementoval v různých prohlížečích objektový model a kaskádové styly jinak, což způsobovalo velké problémy s laděním zdrojového kódu. Knihovna jQuery používá ale jednotné aplikační rozhraní, díky němuž se nemusíme starat o kompatibilitu mezi jednotlivými webovými prohlížeči. Navíc je zdrojový kód psaný v knihovně jQuery mnohem čitelnější, jednodušší a nesrovnatelně kratší. Klíčové vlastnosti knihovny jQuery můžeme shrnout následovně:

- Knihovnu jQuery podporují všechny současné webové prohlížeče. Výrazně ji podporují také moderní mobilní technologie.
- Knihovna jQuery nabízí výborné možnosti pro práci s modelem DOM. Stejně dobře podporuje kaskádové styly, které umí jednoduše zpracovávat.
- Knihovna jQuery zpracovává různé typy událostí při minimálním rozsahu zdrojového kódu. Rovněž můžeme velmi jednoduše vytvářet nejrůznější animace.
- Knihovna jQuery poskytuje propracovanou podporu technologie AJAX, a tím výrazně ulehčuje práci s ní.

Při naší práci se ale zajisté nesetkáme pouze s knihovnou jQuery. Kromě ní je možné narazit na knihovnu Dojo. Knihovna Dojo vznikla roku 2004 a šíří se zdarma pod licencemi AFD a BSD. Oproti knihovně jQuery má ale jisté nevýhody, a to přede-

vším v podobě robustního kódu. Další její nevýhodou je velmi špatná dokumentace, protože neobsahuje ucelený popis všech funkcí. Její výhodou je propracovaný systém komponent, a navíc lze vytvářet vlastní komponenty podle našich představ. Programátoři používají knihovnu Dojo velmi často v kombinaci s frameworkem jazyka PHP Zend Framework, přestože tento framework podporuje také jiné knihovny.

Poznámka: Framework Zend Framework je objektově orientovaný framework implementovaný v jazyce PHP 5. Používá modulární architekturu, což znamená, že programátor používá pouze ty komponenty, které potřebuje k vývoji. Dále patří mezi volně šířené frameworky a má kolem sebe obsáhlou komunitu vývojářů.

Prohlédněte si jednoduchý příklad, jenž porovnává zápis kódu v knihovně Dojo a v knihovně jQuery. Ve výpise 2.10 najdete kód napsaný v knihovně jQuery, zatímco ve výpisu 2.11 uvidíte stejný příklad zapsaný v knihovně Dojo.

Výpis 2.10. Zápis kódu v knihovně jQuery

```
$('#element').click(function() {
   // zpracování metody click()
});
```

Výpis 2.11. Zápis kódu v knihovně Dojo

```
test = dojo.byId("element");
konektor = [];
konektor.push(dojo.connect(test, 'onclick', element));
```

Velmi známou a často používanou knihovnou je také knihovna YUI. Tato knihovna vznikla roku 2005 a šíří se volně pod licencí BSD. Její velkou nevýhodou je mírně zastaralý vzhled jednotlivých komponent, přičemž převod na vlastní vzhled je náročný. Rovněž kód psaný pro tuto knihovnu je o trošku rozsáhlejší než kód pro knihovnu jQuery. Další hojně používanou knihovnou JavaScriptu je knihovna Prototype. Jedná se o známou knihovnu jazyka JavaScript, jež obsahuje velké množství užitečných funkcí pro zjednodušení programování v jazyce JavaScript, a to včetně vlastních ajaxových rutin.

Knihovna Prototype se šíří pod licencí MIT. Ačkoliv má rozsáhlou funkčnost, je poměrně malá. Celá tato knihovna se nachází v jediném souboru *prototype.js*. Tvůrci knihovny Prototype se inspirovali jazykem Ruby, proto má s tímto jazykem hodně společného. Spousta rozšíření této knihovny má syntaxi shodnou s jazykem Ruby, což značně ulehčuje práci programátorům jazyka Ruby. Knihovna Prototype má následující funkce:

- manipulace s modelem DOM,
- správa událostí,
- spolupráce s technologií AJAX,
- různé efekty,
- správa formulářů.

Poslední známou knihovnou, s níž se můžeme při své práci setkat, je knihovna MooTools. Jedná se o čistě objektově orientovanou knihovnu, která vznikla z funkce pojmenované `moo.fx`.

Knihovna MooTools se řadí k mladším knihovnám JavaScriptu – konkrétně spatřila světlo světa v roce 2006. Její velkou výhodou je modularita. Aplikace vytvořené v této knihovně jsou obvykle malé a optimalizované pro výkon. Druhou její výhodou je výborně propracovaná dokumentace. Nakonec bychom neměli zapomenout, že autor této knihovny se inspiroval knihovnou Prototype. Tudíž i tato knihovna se značně podobá jazyku Ruby.

Hlavní výhody a nevýhody knihovny jQuery

Současný trend internetových technologií vyžaduje dynamické stránky HTML. Pro dosažení potřebné dynamiky musíme použít jazyk JavaScript. Samotný jazyk JavaScript není však zcela triviální a jeho studium zabere určitý čas. Z tohoto důvodu se před nějakou dobou začaly v programátorské komunitě objevovat různé balíky a vylepšení, které zjednodušovaly práci s jazykem JavaScript. Spousta z nich se specializovala na konkrétní úlohu. S nástupem knihovny jQuery jsme získali balík, s nímž můžeme vyvíjet dynamické stránky s pomocí jazyka JavaScript rychle a jednoduše.

Knihovna jQuery vděčí za svou eleganci hlavně široké komunitě programátorů, která okolo tohoto projektu vyrostla. Vznikl tak evoluční proces, v němž programátoři vylepšují jádro knihovny. Jak už víme, konečný produkt se šíří pod licencemi GPL a MIT. Proto je zdarma pro veškeré typy použití. Pojďme se společně podívat na hlavní výhody knihovny jQuery:

- Knihovna jQuery plnohodnotně využívá kaskádových stylů. Současně se jedná o jednu z předních zbraní, kterou tato knihovna disponuje. Její mechanismus pro práci s jazykem CSS je jednoduchý, rychle se jej naučíme a takto zapsaný zdrojový kód je snadno čitelný. Díky těmto přednostem si knihovnu jQuery nesmírně oblíbili návrháři webových stránek.
- Knihovna jQuery funguje téměř ve všech webových prohlížečích. Každý prohlížeč má svůj vlastní systém zobrazování zdrojového kódu, který napíšeme. Mluvíme o odchylkách v prohlížečích, které neodpovídají vydaným standardům. Toto je opravdu velký problém, jestliže píšeme kód čistě jen v jazyce JavaScript. Velmi často se nám stane, že kód, jejž jsme odladili v jednom prohlížeči, nefunguje správně v jiném prohlížeči. Knihovna jQuery řeší však tento problém tak, že přidává vlastní abstraktní vrstvu. Ta normalizuje námi psaný zdrojový kód. Kromě toho je takový kód výrazně kratší a mnohem jednodušší.
- Knihovna jQuery velmi dobře spolupracuje s technologií AJAX. Díky této spolupráci můžeme jednoduše zavolat funkci, kterou potřebujeme a všechny nezbytné akce pro vznik ajaxového požadavku se provedou za nás. Rovněž se můžeme spolehnout, že takto vytvořené požadavky budou fungovat ve všech prohlížečích. Stejně jako u jiných součástí této knihovny musíme vyzdvihnout hlavně ulehčení a značnou úsporu zdrojového kódu.
- Knihovna jQuery nabízí různé zásuvné moduly. K dispozici máme obrovské množství volně použitelných rozšíření. Navíc lze snadno vytvářet nové moduly, přičemž tato činnost je opravdu jednoduchá a velmi dobře zdokumentovaná. Proto existuje široká škála velmi užitečných a propracovaných rozšíření. Dokonce i samotná knihovna jQuery obsahuje velké množství funkcí, které vznikly prostřednictvím této architektury.
- Knihovna jQuery vždy pracuje s konkrétním blokem. Nemusíme tedy procházet úplně všechny prvky, díky čemuž je tato knihovna o hodně rychlejší. Kupříkladu metody, které mají za úkol schovat nebo zobrazit element, jsou navržené tak, že berou v úvahu pouze konkrétní skupinu elementů. Tuto techniku nazýváme implicitní iterace a přináší nám to, že nemusíme provádět všechny cykly.
- Knihovna jQuery umožňuje vykonat více akcí za sebou. Toto řetězení je velmi výhodné, jelikož předcházíme zbytečnému používání dočasných proměnných. Výsledkem je přehlednější a kratší kód, který ve svých aplikacích určitě oceníme. Výsledkem většiny operací je samostatný objekt připravený na další akci.

Všechny tyto výhody, které nám knihovna jQuery nabízí, jsou uložené v komprimovaném balíku o velikosti přibližně 30 KB. Současně nám tento balík poskytuje metody pro udržovatelnost našeho vlastního zdrojového kódu.

Knihovna jQuery má velmi málo nevýhod. Nesmíme ale opomenout, že hlavní nevýhodou knihovny jQuery je chybějící komerční podpora. Knihovna jQuery sice disponuje značnou komunitou, ale nemáme záruku, že se naše případné problémy vyřeší v požadovaném čase. Další nevýhodou je upřednostňování formátu JSON nad formátem XML.

Test

1. Co je knihovna jQuery a na co slouží?
2. Kolik stojí používání knihovny jQuery?
3. Jaké další knihovny JavaScriptu znáte?
4. Na co všechno lze použít knihovnu jQuery?
5. Je možné používat knihovnu jQuery dohromady s jinými knihovnami?
6. K čemu slouží funkce `jQuery.noConflict()`?
7. Jak můžete vypsat obsah proměnné do konzole nástroje Firebug?
8. Jaké jsou hlavní výhody knihovny jQuery?
9. Jaké má knihovna jQuery nevýhody?

Náš první kód v knihovně jQuery

Než začneme naplno využívat sílu knihovny jQuery, seznámíme se s jejím základním způsobem použití. Nejdříve napíšeme a spustíme notoricky známý kód „Ahoj knihovno jQuery." Jedná se o základní zdrojový kód, kterým začíná téměř každá učebnice programování. Na tomto jednoduchém příkladu si vysvětlíme základní strukturu kódu napsaného v knihovně jQuery. Budeme jej považovat za základní kámen svých znalostí, na němž můžeme stavět. Ukážeme si, jak knihovna jQuery pracuje s dokumentem HTML a popíšeme si, jak můžeme s touto knihovnou pracovat. Nakonec si ukážeme použití základních metod této knihovny.

Stahujeme knihovnu jQuery

Abychom mohli začít psát první zdrojový kód v knihovně jQuery, musíme si ji nejprve stáhnout kupříkladu z oficiálních stránek na adrese *http://jquery.com/*. K dispozici máme komprimovanou i nekomprimovanou verzi knihovny jQuery. Pro nás jako vývojáře je nejvhodnější nekomprimovaná verze. Nekomprimovaná verze knihovny jQuery nám usnadňuje ladění zdrojového kódu. V produkčním prostředí ale použijeme komprimovanou (efektivnější) verzi knihovny jQuery. Na uvedené stránce najdete i podrobnou dokumentaci, různé návody a užitečné odkazy.

Knihovnu jQuery nemusíme vůbec instalovat. Pokud ji chceme používat, jednoduše ji uložíme do veřejně dostupné složky, kam ukládáme i náš webový projekt. Většinou se knihovna jQuery kopíruje do složky *js* – například *js/jquery.js*. Samozřejmě je pouze na našem uvážení, kam knihovnu jQuery uložíme. Při vkládání této knihovny do dokumentu HTML musíme ale nastavit správnou cestu. V této knize budeme volit složku *js*, protože kromě knihovny jQuery do ní budeme kopírovat i jednotlivé příklady. Tento postup je logický a přehledný.

Připojení knihovny jQuery

Knihovna jQuery nám nabízí několik možností, jak ji připojit ke svému projektu. Jestliže ji nechceme stahovat z její domovské stránky, můžeme ji připojit ke svému projektu ze zdroje CDN. Tento způsob ale může způsobit problém, o němž si řekneme později v této kapitole. K dispozici máme výběr ze tří zdrojů:

- jQuery CDN – *http://code.jquery.com/jquery-1.6.1.min.js*,
- Google CDN – *http://ajax.googleapis.com/ajax/libs/jquery/1.6.1/jquery.min.js*,
- Microsoft CDN – *http://ajax.microsoft.com/ajax/jquery/jquery-1.6.1.min.js*.

Poznámka: Síť CDN je síť pro poskytování obsahu. Jedná se o počítačový systém obsahující kopie dat, která umísťuje na různá místa v síti tak, aby se maximalizovala šířka pásma pro klienty v celé síti. Klienti mají tedy přístup ke kopii dat v jejich blízkosti, čímž se zvyšuje výkon.

Distribuce CDN dodává soubory z mezipaměti. Tím umožňuje načítání dat z nedalekého místa, čímž zrychluje jejich nahrávání. Hlavní výhody používání zdroje CDN lze shrnout následujícím způsobem:

- Používáním zdroje CDN nezatěžujeme náš hosting, a tím samozřejmě zvyšujeme náš výkon.
- Při použití zdroje CDN získáme soubory rychleji než z vlastního hostingu, což neodmyslitelně zrychluje naši aplikaci.
- Je pravděpodobnější, že soubor knihovny jQuery je už uložený v mezipaměti webového prohlížeče.
- K dispozici máme vždy aktuální verzi knihovny jQuery.

Popsali jsme si jednotlivé způsoby použití knihovny jQuery a zůstává jen na našem uvážení, pro jaký způsob se rozhodneme. Každý z nich má totiž své výhody i nevýhody. Musíme si ale uvědomit, že použití externího zdroje CDN nám přináší mnohem větší výkon, než když knihovnu zkopírujeme do složky našeho projektu.

Jak jsme si právě naznačili, použití externího zdroje má rovněž své nevýhody. Může totiž nastat velmi nepříjemný problém, pokud není dostupný externí server, na němž je naše knihovna jQuery uložena. Na takovou situaci musíme být bezesporu připraveni, jinak naše aplikace nebude fungovat správně. Nejlepším řešením je připojit zdroj CDN knihovny jQuery a přitom mít v záloze její lokální kopii. Ve výpise 3.1 si předvedeme, jak tuto nepříjemnou situaci jednoduše vyřešit.

Výpis 3.1. Zabezpečení dostupnosti knihovny jQuery

```
<script type="text/javascript" src="http://ajax.googleapis.com/ajax/libs/
jquery/1.6.1/jquery.min.js"></script>
<script type="text/javascript">
   !window.jQuery && document.write(
      '<script src="js/jquery.js"><\/script>')
</script>
```

Na prvním řádku tohoto zdrojového kódu načítáme knihovnu jQuery ze zdroje CDN společnosti Google. V případě, že se nám to z nějakého důvodu nepodaří, přidáme knihovnu jQuery, kterou jsme uložili na lokální úložiště. Vytvořili jsme jednoduchou podmínku, s jejíž pomocí kontrolujeme, zda knihovna jQuery existuje. Pro tuto kontrolu používáme jednoduchý příkaz zobrazený ve výpisu 3.2.

Výpis 3.2. Kontrola existence knihovny jQuery

```
<script type="text/javascript">
   !window.jQuery && document.write(
      '<script src="js/jquery.js"><\/script>')
</script>
```

Díky tomuto zápisu připojíme lokální kopii knihovny jQuery jen tehdy, když platí výraz zapsaný před operátorem `&&`, a to výraz `window.jQuery`. Tímto výrazem ověřujeme, že v naší aplikaci existuje instance knihovny jQuery. My ve skutečnosti kontrolujeme, že v naší aplikaci neexistuje knihovna jQuery, protože před uvedený výraz vkládáme ještě vykřičník. Víme tedy, že připojení knihovny jQuery z externího zdroje CDN selhalo, takže můžeme provést příkaz napsaný za operátorem `&&`, k němuž připojíme lokální kopii knihovny jQuery.

Poznámka: Operátor `&&` nazýváme také logický součin a jedná se o logický operátor typu `Boolean`, který vrací hodnotu `true` (pravda), nebo hodnotu `false` (nepravda). Pokud jsou podmínky na obou stranách operátoru `&&` pravdivé, tak výsledná hodnota bude pravda. V ostatních případech nám tento operátor vrátí hodnotu nepravda.

Ahoj knihovno jQuery!

Konečně nastal správný čas, abychom napsali svůj první skript v knihovně jQuery, na němž si vysvětlíme základy používání této knihovny. Náš první příklad bude samozřejmě velmi jednoduchý. Jediná věc, kterou uvidíte na obrazovce po jeho spuštění, bude text „Ahoj knihovno jQuery!“. Tento příklad není použitelný v praxi, ale je to základní zdrojový kód, který byste měli zvládnout. Mnohokrát se totiž při učení některého programovacího jazyka setkáte s názorem, že po zvládnutí tohoto příkladu

se z žáka stává programátor. Následně je nutné tyto znalosti prohlubovat a získávat zkušenosti.

Píšeme svůj první skript

Pojďme se ale vrhnout na náš první příklad. Vytvoříme nový soubor a pojmenujeme jej *priklad_1.js*. Do tohoto souboru zapíšeme zdrojový kód z výpisu 3.3.

Výpis 3.3. Náš první kód v knihovně jQuery

```
$(document).ready(function () {
   $("body").html("<h1>Ahoj knihovno jQuery!</h1>");
});
```

Takto vytvořený soubor připojíme do souboru HTML (v našem případě soubor *index.html*), jehož obsah se zobrazuje ve výpisu 3.4. Samotné připojení daného souboru zvýrazníme tučným písmem. Nezapomínejme však, že všechny příklady a externí soubory, které chceme přidat ke svému projektu, bychom měli umístit do elementu `head`. Element `head` je hlavička dokumentu, která obsahuje informace o našem projektu. Tato hlavička nám dovoluje nastavovat různé vlastnosti daného dokumentu. Je důležité vědět, že obsah hlavičky neovlivňuje vzhled našeho projektu.

Výpis 3.4. Připojení skriptu v knihovně jQuery

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html>
<head>
   <meta http-equiv="content-type" content="text/html; charset=UTF-8" />
   <title>Ahoj knihovno jQuery</title>
   <link rel="stylesheet" href="styl.css" type="text/css" />

   <script src="js/jquery.js" type="text/javascript"></script>
   <script src="js/priklad_1.js" type="text/javascript"></script>
</head>
<body>
   <!-- vlastní kód stránky -->
</body>
</html>
```

Připojujeme kaskádové styly

Zároveň si vytvoříme soubor obsahující pravidla kaskádových stylů. Cílem kaskádových stylů je upravovat vzhled předem definovaného dokumentu HTML. Jejich

použití je velmi jednoduché a tvůrcům webových stránek výrazně ulehčují práci. Správným zápisem těchto stylů můžeme získat velmi užitečný dokument, který lze v případě potřeby používat v kterémkoliv jiném projektu.

S pomocí kaskádových stylů můžeme nastavovat velikost text, barvu textu, pozici elementu, pozadí daného elementu, šířku a výšku daného elementu a spoustu dalších vlastností. Struktura moderní stránky by měla být působivá a hlavně přehledná. Takového výsledku dosáhneme prostřednictvím kaskádových stylů velmi snadno.

Tip: Při vytváření souboru s pravidly kaskádových stylů můžou vzniknout drobné syntaktické chyby. Abychom zkontrolovali, že jsou naše pravidla v pořádku, můžeme použít kontrolní nástroj od konzorcia W3C. Tento validační nástroj najdete na internetové adrese *http://validator.w3.org/*.

Jednotlivá pravidla kaskádových stylů bychom měli definovat výhradně v externím souboru, jejž připojujeme k dokumentu HTML. Pravidla stylů je totiž možné zapisovat přímo do hlavičky dokumentu HTML, nebo dokonce do elementu, jehož vzhled chceme upravit.

Když definujeme kaskádové styly, můžeme tedy používat i jiné způsoby, než je definice v externím souboru. Ztratíme ale výhody, které nám definice v externím souboru nabízí. Jedině tak dosáhneme ideálního stavu, kvůli kterému tato technologie vznikla. Tímto způsobem totiž jednoznačně oddělíme obsah dokumentu HTML od jeho vzhledu. Námi vytvořený soubor můžeme případně průběžně doplňovat. Tento soubor uložíme do stejného adresáře, v němž máme soubor *index.html*. Pojmenujeme jej *styl.css* a otevřeme ho v editoru Notepad++. Následně do něj vložíme pravidla kaskádových stylů z výpisu 3.5.

Výpis 3.5. Základní definice stylů

```
body {
   margin: 50px;
   background-color: #FFFFFF;
   font-size: 11px;
}

h1 {
   text-align: center;
   font-family: Verdana;
   font-size: 100px;
   color: #333300;
}

p {
   text-align: justify;
   font-family: Verdana;
```

```
    font-size: 12px;
}
```

Tento soubor jazyka CSS připojíme ke svému projektu podobně jako jsme připojili náš skript knihovny jQuery. Místo elementu `script` ale použijeme element `link`. Stejně tak musíme dbát na správnou cestu k danému souboru, což je v našem případě *styl.css*. Dokument HTML s připojenými kaskádovými styly najdete na výpise 3.6, přičemž připojení souboru je zvýrazněné tučným písmem. Samotná stránka HTML neobsahuje v části `body` žádné informace. Kdybychom tuto jednoduchou stránku zobrazili bez připojeného souboru *priklad_1. js*, objevila by se pouze čistá stránka, což znamená, že v prohlížeči se nezobrazila žádná informace.

Výpis 3.6. Připojení souboru CSS

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html>
<head>
   <meta http-equiv="content-type" content="text/html; charset=UTF-8" />
   <title>Ahoj knihovno jQuery</title>
   <link rel="stylesheet" href="styl.css" type="text/css" />

   <script src="js/jquery.js" type="text/javascript"></script>
   <script src="js/priklad_1.js" type="text/javascript"></script>
</head>
<body>
   <!-- vlastní kód stránky -->
</body>
</html>
```

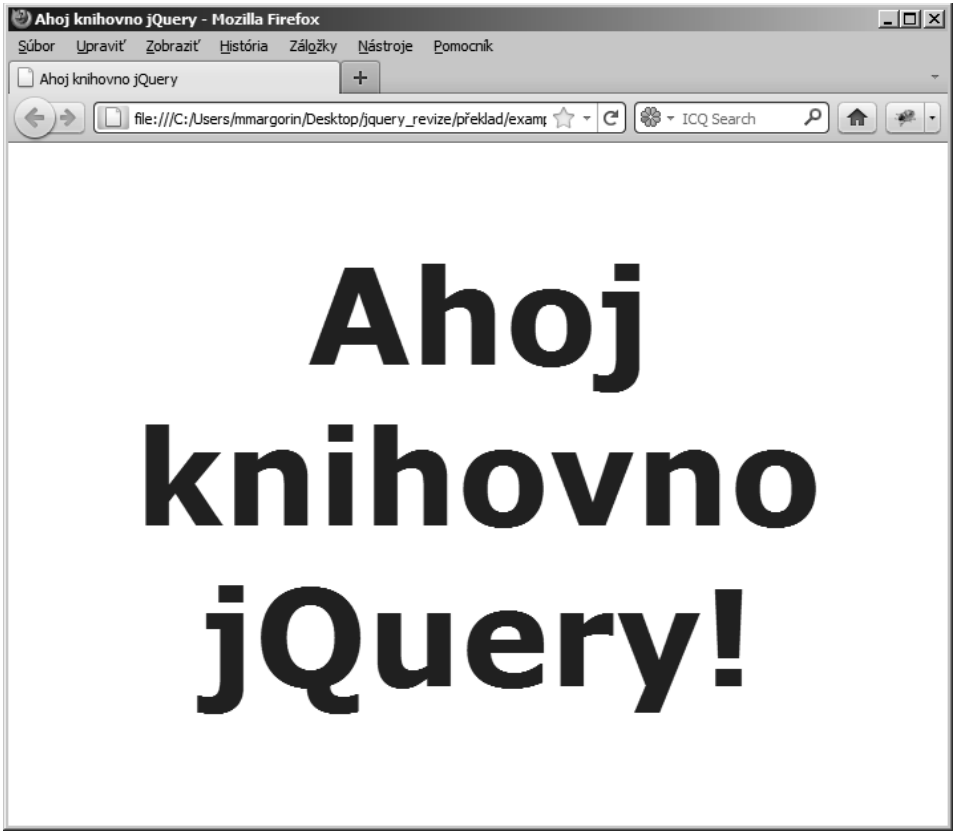
Soubor *styl.css* s pravidly stylů jsme úspěšně vložili do své testovací stránky. Od této chvíle můžeme používat jednotlivá pravidla a upravovat vzhled příslušných elementů.

Výsledek spuštění skriptu

Po spuštění našeho kódu bychom měli získat stejný výstup, jaký lze spatřit na obrázku 3.1, na němž se zobrazuje velký text „Ahoj knihovno jQuery!“. Je nutné zdůraznit, že v běžných situacích bychom určitě nenastavovali velikost písma nadpisu na 100 pixelů. Ve skutečné aplikaci, kterou prezentujeme na Internetu, bychom s největší pravděpodobností použili nadpis o velikosti maximálně 16 nebo 22 pixelů. V tomto případě však chceme, aby byl náš pozdrav „Ahoj knihovno jQuery!“ opravdu výrazný, proto jsme výjimečně nastavili nadpisu tak velké písmo.

Úspěšně jsme napsali a spustili svůj první a nesmírně důležitý zdrojový kód napsaný pomocí knihovny jQuery. Odteď můžete tvrdit, že víte, jak používat knihovnu jQuery. V další části této publikace se naučíte ještě spoustu dalších věcí, avšak teď jste udělali největší krok na cestě ke zvládnutí knihovny jQuery.

Postupné získávání nových poznatků bude v případě této knihovny dozajista zábavné. Vůbec se nemusíte bát složitých podmínek a jiných nástrah, jak tomu bývá zvykem u jiných programovacích jazyků. Tato knihovna vám pomůže natolik, že se budete moci soustředit na svůj projekt. Všechny abnormální jevy uvedené v kapitole 2 vyřeší knihovna jQuery za nás, aniž bychom se jimi museli zabývat.



Obrázek 3.1. Výstup skriptu Ahoj knihovno jQuery!

Knihovna jQuery ve spolupráci s jazykem HTML

Hypertextový značkovací jazyk HTML je určený na tvorbu webových stránek a jiných informací, které chceme zobrazit ve webovém prohlížeči. Jeho úlohou je tedy prezentace informací – například textu, tabulek, odstavců apod. Abychom mohli zpracovávat různé události a jinak manipulovat se samotným dokumentem HTML, potřebujeme skriptovací jazyk. Nejčastěji zpracováváme dokumenty HTML ve webovém prohlížeči pomocí jazyka JavaScript; v našem případě pomocí knihovny jQuery, která je nad ním postavená. Pokud chceme používat tuto knihovnu, musíme ji připojit ke svému dokumentu HTML. Pojďme si tedy ukázat, jak na to.

Knihovnu jQuery vložíme do kódu jazyka HTML stejným způsobem jako jakýkoliv jiný skript jazyka JavaScript. Pochopitelně nesmíme zapomenout uvést správnou cestu ke knihovně jQuery. Když uděláme chybu a zadáme cestu špatně, nebude náš příklad fungovat. Na připojování používáme element `script`. Tento element má atribut `src`, který obsahuje cestu k připojovanému souboru. Tento atribut nastavíme tudíž tak, aby jeho hodnota ukazovala na místo, kam jsme uložili náš soubor jazyka JavaScript. Druhým důležitým atributem tohoto elementu je atribut `type`, jenž určuje typ připojovaného souboru. V této situaci nastavujeme danému atributu hodnotu `text/javascript`, protože chceme pracovat se souborem jazyka JavaScript. Knihovnu jQuery proto připojíme stejně jako na výpisu 3.7.

Výpis 3.7. Připojení knihovny jQuery

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

Připojení knihovny jQuery

Ukážeme si další příklad, na němž otestujeme připojování knihovny jQuery. Tento příklad zahrnuje jednoduchý test připojení knihovny jQuery. Svým způsobem se jedná o stejný příklad jako první, v němž jsme zobrazovali text „Ahoj knihovno jQuery!“. Na tomto příkladu si ale podrobněji vysvětlíme připojování knihovny jQuery k dokumentu HTML.

Vytvoříme si proto nový soubor jazyka HTML a pojmenujeme jej *index.html*. V tomto souboru načteme knihovnu jQuery stejně jako ve výpisu 3.7. Samotné připojování této knihovny zvýrazníme tučným písmem. Samozřejmě i tentokrát umístíme připojování knihovny jQuery do elementu `head` tak, jak je patrné na výpise 3.8.

Výpis 3.8. Načítání knihovny jQuery

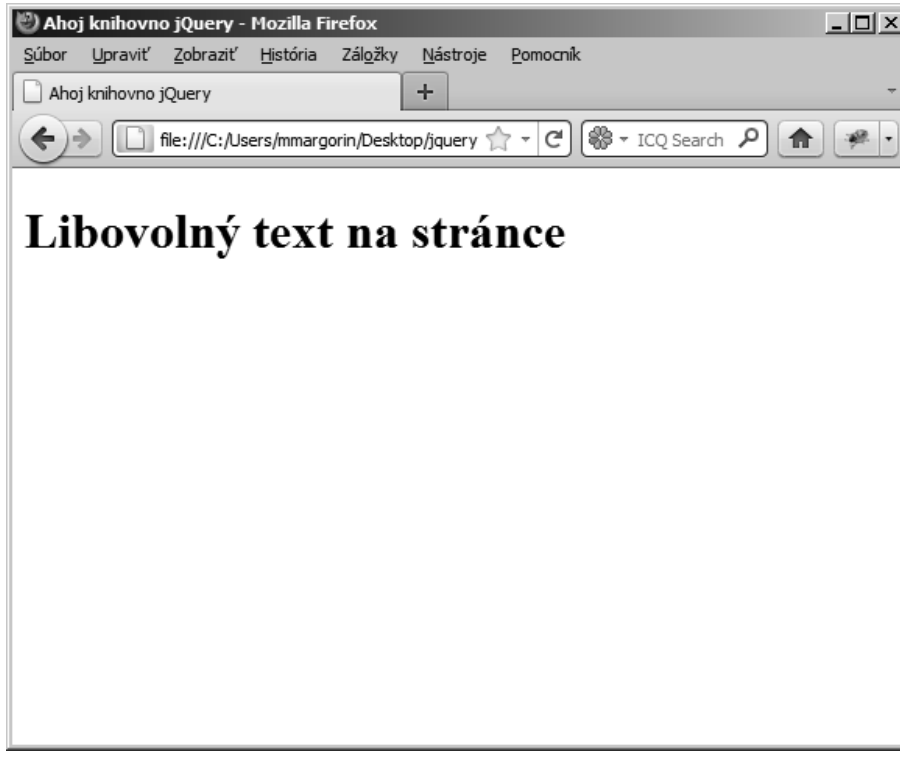
```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html>
<head>
   <meta http-equiv="content-type" content="text/html; charset=UTF-8" />
   <title>Ahoj knihovno jQuery</title>
   <script src="js/jquery.js" type="text/javascript"></script>
</head>
<body>
   <!-- vlastní kód stránky -->
</body>
</html>
```

Základní práce s knihovnou jQuery

Na následujícím příkladu si popíšeme základní práci s knihovnou jQuery. V tomto velmi jednoduchém příkladu otestujeme, jestli se nahrála knihovna jQuery. Nejprve načteme samotnou knihovnu jQuery, aniž bychom ji nějakým způsobem použili. Na stránce se proto zobrazí jen samotný text „Libovolný text na stránce," jak vidíte na obrázku 3.2. Zdrojový kód tohoto příkladu se zobrazuje ve výpise 3.9.

Výpis 3.9. Načítání knihovny jQuery bez použití jejích funkcí

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html>
<head>
   <meta http-equiv="content-type" content="text/html; charset=UTF-8" />
   <title>Ahoj knihovno jQuery</title>
   <script src="js/jquery.js" type="text/javascript"></script>
</head>
<body>
   <h1>
      Libovolný text na stránce
   </h1>
</body>
</html>
```



Obrázek 3.2. Výstup výpisu 3.9

Posléze k tomuto příkladu připojíme kód knihovny jQuery, s nímž nahradíme text „Libovolný text na stránce“ textem „Knihovna jQuery pracuje správně“. Kdybychom nenačetli knihovnu jQuery správně nebo bychom udělali nějakou chybu v zápise kódu, zobrazil by se stejný text jako na obrázku 3.2. Pokud je všechno v pořádku, kód se provede korektně a my získáme stejný text jako na obrázku 3.3. Jak je patrné z tohoto příkladu, pomocí knihovny jQuery nahradíme obsah elementu `h1`.

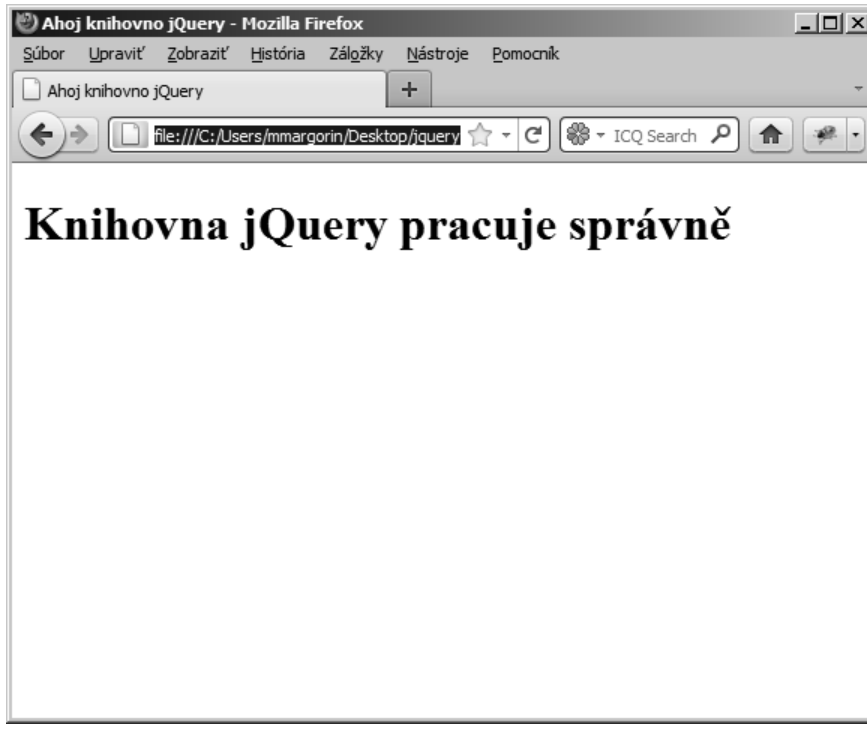
Poznámka: Základní jednotkou značkovacího jazyka HTML je element. Elementy se v dokumentu HTML uspořádávají do stromové struktury. Každý element jazyka HTML má svůj význam. Například element `h1` představuje hlavní nadpis dokumentu.

Kód knihovny jQuery je v tomto příkladu stejný jako ten, s nímž jsme zobrazovali text „Ahoj knihovno jQuery!“, pouze jsme použili jiný element a zobrazili jiný text. Jak kód funguje si popíšeme zanedlouho v této kapitole. Kód upravený tímto způsobem se nachází ve výpise 3.10. V tomto příkladu jsme kód napsaný v knihovně jQue-

ry nepřipojovali z externího soboru, ale vložili jsme jej přímo do hlavičky dokumentu HTML, abychom si ukázali i tuto možnost. Ve všech dalších příkladech už budeme zdrojový kód psaný v knihovně jQuery připojovat z externího souboru, protože je to rozumnější volba. Určitě si dokážete představit, jak by vypadal dokument HTML se stovkami řádků kódu v knihovně jQuery.

Výpis 3.10. Načtení knihovny jQuery s použitím jejích funkcí

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html>
<head>
   <meta http-equiv="content-type" content="text/html; charset=UTF-8" />
   <title>Knihovna jQuery</title>
   <script src="js/jquery.js" type="text/javascript"></script>
   <script type="text/javascript">
      $(document).ready(
         function() {
            $("h1").text("Knihovna jQuery pracuje správně");
         }
      );
   </script>
</head>
<body>
   <h1>
      Libovolný text na stránce
   </h1>
</body>
</html>
```



Obrázek 3.3. Výstup výpisu 3.10

Úspěšně jsme načetli knihovnu jQuery a můžeme ji s chutí začít používat. Jak už jsme si řekli, samotný zdrojový kód napsaný v knihovně jQuery můžeme vkládat do dokumentu HTML dvěma způsoby. Můžeme psát kód přímo do dokumentu HTML, nebo načítat skript z externího souboru. Tento externí soubor potom připojujeme stejně jako jsme vložili knihovnu jQuery.

Tip: Externí soubory je vhodné rozdělovat na menší logické celky, aby byly přehlednější. V případě, že budeme chtít později změnit část kódu, změníme pouze nezbytný soubor a ostatní zůstanou netknuté. Ve svých příkladech budeme používat pouze externí soubory, protože tento způsobe je daleko přehlednější.

V další části této kapitoly si rozebereme výše uvedený zdrojový kód krok po kroku a naučíme se základům práce s knihovnou jQuery.

Spouštíme příkazy pro načítání stránky

První a neuvěřitelně důležitou funkcí, kterou musíme znát, je funkce `$(document).ready(function(){})`. Jejím zápisem sdělujeme, že uvnitř dokumentu jsou funkce připravené ke zpracování. Dovnitř této funkce (mezi složené závorky `{}`) budeme zapisovat všechny akce knihovny jQuery. Zápis této funkce najdete ve výpise 3.11.

Výpis 3.11. Základní funkce knihovny jQuery

```
$(document).ready(function () {
   // akce knihovny jQuery
});
```

Toto řešení je mnohem výhodnější než používání čistého kódu JavaScriptu. Příkazy, které zapíšeme do této funkce, se provedou hned jak se kompletně načte model DOM (objektový model dokumentu), a to ještě před samotným načítáním obsahu stránky (musí se načíst všechny obrázky, bannery atd.). Model DOM si popíšeme podrobněji v kapitole o selektorech. Náš skript se na rozdíl od klasického skriptu JavaScript vykoná dřív a efekty pro elementy se spustí okamžitě, jak se daný element načte.

Je možné říct, že funkce `$(document).ready()` nahrazuje funkci `onload()` jazyka JavaScript. Nevýhodou funkce `onload()` je možnost volání jediné funkce a také příkazy volané tímto způsobem se vykonávají až po načtení kompletního obsahu stránky. Funkce `$(document).ready()` tedy přináší spoustu vylepšení oproti běžným možnostem jazyka JavaScript.

Poznámka: Událost `onLoad` nastává ve chvíli, kdy se načítá objekt. Kupříkladu kód `<body onload="alert('test')">` vypíše po načtení stránky na obrazovku text „test." Událost `onLoad` se spustí až po načtení všech prvků dané stránky.

Metoda html()

V této části kapitoly si vysvětlíme, jak pracuje metoda `html()` knihovny jQuery. Tato velmi jednoduchá metoda slouží pro zápis textové hodnoty do námi vybraného elementu. Ještě předtím si ale stručně popíšeme, jak funguje nejdůležitější funkce knihovny jQuery. Ta se řadí mezi základní operace, které provádíme v knihovně jQuery, a vybírá konkrétní části dokumentu. Tohoto výběru dosáhneme pomocí funkce `jQuery()`, kterou zkráceně zapisujeme jako `$()`.