



Jonathan Chaffer · Karl Swedberg

jQuery lernen und einsetzen

Bessere Webanwendungen mit einfachen
JavaScript-Techniken entwickeln

Learning jQuery
Deutsche Ausgabe
der 3. engl. Aufl.

dpunkt.verlag



jQuery lernen und einsetzen

Jonathan Chaffer · Karl Swedberg

jQuery lernen und einsetzen

Bessere Webanwendungen mit einfachen JavaScript-Techniken entwickeln

Übersetzung der 3. engl. Auflage



dpunkt.verlag

Lektorat: Dr. Michael Barabas
Copy Editing: Ursula Zimpfer, Herrenberg
Übersetzung und Satz: G&U Language & Publishing Services GmbH, Flensburg, (www.GundU.com)
Herstellung: Nadine Thiele
Umschlaggestaltung: Helmut Kraus, www.exclam.de
Druck und Bindung: M.P. Media-Print Informationstechnologie GmbH, 33100 Paderborn

Bibliografische Information der Deutschen Nationalbibliothek

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de> abrufbar.

ISBN:

Buch 978-3-89864-786-1
PDF 978-3-86491-150-7
ePub 978-3-86491-151-4

1. Auflage 2012

Copyright der deutschen Ausgabe © 2012 [dpunkt.verlag](http://www.dpunkt-verlag.de) GmbH

Ringstraße 19B
69115 Heidelberg

Copyright © Packt Publishing 2011.

First published in the English language under the title »Learning jQuery«, Third Edition

Die vorliegende Publikation ist urheberrechtlich geschützt. Alle Rechte vorbehalten. Die Verwendung der Texte und Abbildungen, auch auszugsweise, ist ohne die schriftliche Zustimmung des Verlags urheberrechtswidrig und daher strafbar. Dies gilt insbesondere für die Vervielfältigung, Übersetzung oder die Verwendung in elektronischen Systemen.

Es wird darauf hingewiesen, dass die im Buch verwendeten Soft- und Hardware-Bezeichnungen sowie Markennamen und Produktbezeichnungen der jeweiligen Firmen im Allgemeinen warenzeichen-, marken- oder patentrechtlichem Schutz unterliegen.

Alle Angaben und Programme in diesem Buch wurden mit größter Sorgfalt kontrolliert. Weder Autor noch Verlag können jedoch für Schäden haftbar gemacht werden, die in Zusammenhang mit der Verwendung dieses Buches stehen.

5 4 3 2 1 0

Geleitwort

Ich fühle mich geehrt, dass Karl Swedberg und Jonathan Chaffer die Mühe auf sich genommen haben, »Learning jQuery« zu schreiben. Als erstes jQuery-Buch setzte es den Standard, den andere jQuery-Bücher – und praktisch auch alle anderen JavaScript-Bücher – zu erreichen versuchen. Es ist eines der meistgekauften JavaScript-Bücher, was zu keinem geringen Teil seiner Genauigkeit und seinen vielen Details zu verdanken ist.

Er freut mich besonders, dass Karl und Jonathan dieses Buch geschrieben haben, weil ich sie bereits so gut kannte, dass ich wusste, sie wären die besten für diese Aufgabe. Als Teil des Kernteams von jQuery hatte ich Karl über mehrere Jahre intensiv kennen gelernt und natürlich besonders, während dieses Buch entstand. Wenn ich das Ergebnis betrachte, wird deutlich, dass seine Fähigkeiten als Entwickler und als ehemaliger Englischlehrer hier eine hervorragende Kombination abgegeben haben.

Beide Autoren konnte ich persönlich treffen, was in der Welt verteilter Open-Source-Projekte eher selten ist, und beide sind und bleiben herausragende Mitglieder der jQuery-Gemeinschaft.

Die jQuery-Bibliothek wird von den unterschiedlichsten Personen der jQuery-Community genutzt. Diese Gemeinschaft ist voll von Designern, Entwicklern, Personen mit Programmiererfahrung oder ohne. Auch im jQuery-Team finden sich Personen mit den unterschiedlichsten beruflichen Hintergründen, die alle ihren Beitrag zum Projekt leisten. Eine Sache ist jedoch allen jQuery-Anwendern gemein: Wir sind eine Gemeinschaft von Entwicklern und Designern, die sich wünscht, dass die JavaScript-Entwicklung einfach wird.

Es ist an dieser Stelle fast schon klischeehaft zu sagen, dass ein Open-Source-Projekt gemeinschaftsorientiert ist oder dass es sich darauf fokussiert, neuen Anwendern den Einstieg zu erleichtern. Bei jQuery ist das jedoch keine leere Floskel, sondern die treibende Kraft des Projekts. Es gibt momentan mehr Personen im jQuery-Team, die sich der Gemeinschaft, der Dokumentation und den Plugins widmen, als der Wartung der Codebasis. Obwohl eine korrekte Bibliothek

extrem wichtig ist, macht die Gemeinschaft den Unterschied zwischen einem stockenden, mittelmäßigen und einem Projekt aus, das allen Anforderungen gerecht wird oder sie noch übertrifft.

Wie wir das Projekt betreiben und wie Sie den Code einsetzen, unterscheidet sich deutlich von anderen Open-Source-Projekten und den meisten JavaScript-Bibliotheken. Das jQuery-Projekt und die Community verfügen über weitreichende Sachkunde, wir wissen, warum jQuery eine völlig andere Art zu programmieren ist, und tun unser bestes, dieses Wissen weiterzugeben.

Die jQuery-Gemeinschaft können Sie nicht durch bloßes Lesen verstehen, Sie müssen daran teilhaben, um vollständig darin einzutauchen. Ich hoffe, Sie haben die Möglichkeit dazu. Nehmen Sie an den Foren, Mailinglisten und Blogs teil, und lassen Sie uns Ihnen dabei helfen, jQuery besser kennen zu lernen.

Für mich ist jQuery viel mehr als ein Haufen Quelltext. Es ist die Summe der über die Jahre angesammelten Erfahrungen, die eine Bibliothek bilden, die beträchtlichen Höhen und Tiefen, die schwierige Entwicklung und die aufregenden Momente, das Projekt wachsen und blühen zu sehen. Und es ist ein Zusammenwachsen der Anwender und der Team-Mitglieder, das Bemühen, sie zu verstehen, und der Versuch, sich anzupassen und zu wachsen.

Als ich dieses Buch erstmals gesehen und seinen Ansatz, jQuery als einheitliches Werkzeug zu erläutern, gelesen habe, war ich verblüfft und angeregt zugleich. Zu erkennen wie andere lernen, verstehen und jQuery so anpassen, wie es für sie richtig ist, macht das Besondere dieses Projekts aus.

Ich bin nicht der einzige, der jQuery von einer anderen Warte als der normalen Werkzeug-Anwender-Beziehung betrachtet. Ich bin nicht sicher, ob ich genau sagen kann, warum, aber ich habe es wieder und wieder beobachtet – der Augenblick, in dem der Anwender zu lächeln beginnt, weil er erkennt, wie viel jQuery ihm helfen kann.

Es gibt diesen besonderen Moment, bei dem es beim Anwender »Klick macht«, wenn er erkennt, dass dieses Werkzeug viel, viel mehr ist als ein einfaches Hilfsmittel. In diesem Augenblick ändert sich auch sein Verständnis völlig, wie dynamische Webanwendungen zu programmieren sind. Das ist eine faszinierende Sache und immer der schönste Augenblick in einem jQuery-Projekt.

Ich hoffe, dass Sie dieses Erlebnis ebenfalls haben werden.

John Resig
Erfinder von jQuery

Über die Autoren

Jonathan Chaffer ist Mitglied der Rapid Development Group, einer Web-Entwicklungsfirma in Grand Rapids, Michigan. Zu seiner Arbeit gehört die Projektüberwachung und Implementierung einer Vielzahl von Technologien mit einem Schwerpunkt auf PHP, MySQL und JavaScript. Er leitet außerdem jQuery-Trainingsseminare für Webentwickler.

In der Open-Source-Gemeinschaft ist Jonathan sehr aktiv beim Drupal-CMS-Projekt, das jQuery als JavaScript-Framework ausgewählt hat. Er ist der Schöpfer des Content Construction Kits, eines beliebten Moduls für die Verwaltung strukturierter Inhalte auf Drupal-Sites. Er ist für die wesentlichen Überarbeitungen im Drupal-Menüsystem und der Entwickler-API verantwortlich.

Jonathan lebt mit seiner Frau Jennifer in Grand Rapids.

Ich möchte Jenny für ihre unermüdliche Begeisterung und Unterstützung danken, Karl für die Motivation weiterzumachen, als es beim Skript langsamer voranging, und der Ars-Technica-Gemeinschaft für ihre fortwährende Inspiration bei diesem technischen Werk. Zusätzlich möchte ich Mike Henry und dem Twisted Pixel-Team für die netten Ablenkungen zwischen meinen Schreibphasen danken.

Karl Swedberg ist Webentwickler bei Fusionary Media in Grand Rapids, Michigan, wo er viel Zeit damit verbringt, tolle Sachen mit JavaScript anzustellen. Als Mitglied des jQuery-Teams ist Karl zuständig für die Pflege der jQuery-API-Site unter api.jquery.com. Er veröffentlicht außerdem Anleitungen in seinem Blog learningjquery.com und präsentiert auf Workshops und Konferenzen. Wenn er nicht gerade programmiert, verbringt Karl gern seine Zeit mit der Familie, röstet in seiner Garage Kaffee und trainiert im örtlichen Fitnessclub.

Ich danke meiner Frau Sara und meinen beiden Kindern Benjamin und Lucia für all die Freude, die sie meinem Leben geben. Dank auch an Jonathan Chaffer für seine Geduld und die gemeinsame Arbeit an diesem Buch.

Vielen Dank auch an John Resig für die beste JavaScript-Bibliothek der Welt und alle anderen, die ihren Code, ihre Zeit und Expertise in dieses Projekt investiert haben. Danke an die Mitarbeiter von Packt Publishing, die technischen Redakteure dieses Buchs, jQuery Cabal und den vielen anderen, die auf dem langen Weg Hilfe und Inspiration geleistet haben.

Die Fachgutachter

Kaiser Ahmed ist professioneller Webentwickler mit einem Bachelor-Grad von der Khula University of Engineering and Technology (KUET). Außerdem ist er Mitbegründer des vollständig auf Outsourcing gestützten Unternehmens CyberXpress.Net Inc. in Bangladesch.

Er verfügt über ein breites Spektrum an technischen Fertigkeiten, Kenntnissen über das Internet und Erfahrung über die ganze Palette der Onlineentwicklung hinweg, die er dazu einsetzt, Onlinepräsentationen für viele Kunden zu erstellen und zu verbessern. Er hat viel Freude an der Gestaltung der Architektur und Infrastruktur von Websites, an der Back-End-Entwicklung mit Open-Source-Werkzeugen (Linux, Apache, MySQL, PHP [LAMP] und andere) und an der Front-End-Entwicklung mit CSS und HTML/XHTML.

*Er möchte seiner liebenden Frau Maria Akter
für ihre Unterstützung danken.*

Kevin Boudloche ist ein Webentwickler aus Mississippi. Webseiten erstellt er als Hobby seit mehr als acht Jahren und professionell seit drei Jahren. Sein Schwerpunkt liegt auf der Entwicklung von Front-Ends und Webanwendungen.

Carlos Esteves ist der Gründer von Ehxioz (<http://ehxioz.com/>), einer jungen Softwareentwicklungsfirma aus Los Angeles, die sich auf die Entwicklung moderner Webanwendungen spezialisiert hat und die neuesten Technologien und Methoden zur Webentwicklung nutzt. Er hat über zehn Jahre Erfahrung als Webentwickler und einen Bachelor-Grad in Informatik von der California State University in Los Angeles.

Inhaltsverzeichnis

	Einleitung	1
1	Erste Schritte	7
1.1	Was bietet jQuery?	7
1.2	Warum jQuery so gut funktioniert	9
1.3	Unsere erste Webseite mit jQuery	11
	1.3.1 jQuery herunterladen	11
	1.3.2 Einrichten von jQuery in einem HTML-Dokument	11
	1.3.3 jQuery-Code hinzufügen	14
	1.3.4 Das fertige Produkt	16
1.4	Einfaches JavaScript und jQuery im Vergleich	17
1.5	Entwicklungswerkzeuge	18
	1.5.1 Firebug	19
1.6	Zusammenfassung	22
2	Elemente auswählen	23
2.1	Das Document Object Model	23
2.2	Die Funktion <code>\$()</code>	25
2.3	CSS-Selektoren	26
	2.3.1 Listenelemente formatieren	27
	2.3.2 Attributselektoren	29
	2.3.3 Links formatieren	29
2.4	jQuery-Selektoren	31
	2.4.1 Zeilen abwechselnd formatieren	32
	2.4.2 Formularselektoren	36
2.5	Methoden zum Durchlaufen des DOM	37
	2.5.1 Einzelne Zellen formatieren	38
	2.5.2 Verkettung	40

2.6	Zugriff auf DOM-Elemente	41
2.7	Zusammenfassung	42
2.7.1	Literatur	42
2.8	Übungsaufgaben	43
3	Ereignisbehandlung	45
3.1	Aufgaben beim Laden der Seite durchführen	45
3.1.1	Die Codeausführung zeitlich abstimmen	45
3.1.2	Mehrere Skripte auf einer Seite	46
3.1.3	Kurzschreibweisen	48
3.1.4	Argumente an den .ready()-Callback übergeben	48
3.2	Einfache Ereignisse	49
3.2.1	Ein einfacher Formatwechsler	49
3.2.2	Die anderen Schaltflächen aktivieren	52
3.2.3	Ereignishandler-Kontext	53
3.2.4	Weitere Konsolidierung	55
3.2.5	Kurzformen für Ereignisse	57
3.3	Zusammengesetzte Ereignisse	58
3.3.1	Erweiterte Funktionen anzeigen und ausblenden	58
3.3.2	Anklickbare Elemente hervorheben	60
3.4	Der Weg eines Ereignisses	62
3.4.1	Nebenwirkungen des Event Bubbling	64
3.5	Den Weg ändern: das Ereignisobjekt	64
3.5.1	Ereignisziele	66
3.5.2	Die Ereignisweiterleitung abbrechen	66
3.5.3	Standardaktionen	67
3.5.4	Ereignisdelegierung	68
3.5.5	Methoden für die Ereignisdelegierung	71
3.6	Ereignishandler entfernen	71
3.6.1	Namensräume für Ereignisse	72
3.6.2	Ereignisse erneut binden	73
3.7	Benutzerinteraktion simulieren	75
3.7.1	Tastaturereignisse	76
3.8	Zusammenfassung	79
3.8.1	Literatur	80
3.9	Übungsaufgaben	80

4	Formatierung und Animation	81
4.1	Inline-Bearbeitung mit CSS	81
4.2	Anzeigen und Verbergen	86
4.3	Effekte und Speed	89
4.3.1	Anzeigen mit »Geschwindigkeit«	89
4.3.2	Ein- und ausblenden	90
4.3.3	Auseinander- und zusammenfallen	91
4.3.4	Zusammengesetzte Effekte	92
4.4	Benutzerdefinierte Animationen erstellen	93
4.4.1	Effekte manuell erstellen	94
4.4.2	Mehrere Eigenschaften gleichzeitig animieren	95
4.5	Gleichzeitige und aneinandergereihte Effekte	99
4.5.1	Mit einem einzelnen Satz von Elementen arbeiten	99
4.5.2	Mit mehreren Sätzen von Elementen arbeiten	103
4.5.3	Kurz und bündig	107
4.6	Zusammenfassung	108
4.6.1	Literatur	108
4.7	Übungsaufgaben	108
5	DOM-Bearbeitung	109
5.1	Attribute bearbeiten	109
5.1.1	Nicht-Klassenattribute	110
5.1.2	Eigenschaften von DOM-Elementen	113
5.2	Bearbeitung des DOM-Baums	114
5.2.1	Neues zur Funktion <code>\$()</code>	114
5.2.2	Neue Elemente erstellen	115
5.2.3	Neue Elemente einfügen	116
5.2.4	Elemente verschieben	117
5.2.5	Elemente verschachteln	119
5.2.6	Umgekehrte Einfügemethoden	121
5.3	Elemente kopieren	125
5.3.1	Klonen für interne Zitate	126
5.4	Get- und Set-Methoden für Inhalte	128
5.4.1	Weitere Formatanpassungen	130
5.5	Methoden zur DOM-Bearbeitung – kurz und bündig	132
5.6	Zusammenfassung	133
5.6.1	Literatur	133
5.7	Übungsaufgaben	133

6	Daten mit Ajax senden	135
6.1	Daten bei Bedarf laden	135
6.1.1	HTML anhängen	137
6.1.2	Mit JavaScript-Objekten arbeiten	140
6.1.3	XML-Dokumente laden	146
6.2	Ein Datenformat auswählen	149
6.3	Daten an den Server übergeben	150
6.3.1	GET-Requests durchführen	151
6.3.2	POST-Requests durchführen	155
6.3.3	Formulare serialisieren	156
6.4	Unterschiedliche Inhalte liefern	158
6.5	Die Anforderung im Auge behalten	160
6.6	Fehlerbehandlung	162
6.7	Ereignisse in Ajax	164
6.8	Sicherheitseinschränkungen	165
6.8.1	JSONP für fremde Daten verwenden	166
6.9	Zusätzliche Optionen	168
6.9.1	Die grundlegende Methode ajax	168
6.9.2	Standardoptionen ändern	169
6.9.3	Teile einer HTML-Seite laden	170
6.10	Zusammenfassung	172
6.10.1	Literatur	172
6.11	Übungsaufgaben	173
7	Plug-ins verwenden	175
7.1	Plug-ins finden und Unterstützung bekommen	175
7.2	Ein Plug-in verwenden	176
7.2.1	Das Cycle-Plug-in herunterladen und einbinden	176
7.2.2	Einfache Plug-in-Anwendungen	176
7.2.3	Parameter an Plug-in-Methoden übergeben	178
7.2.4	Voreingestellte Parameter	179
7.2.5	Andere Arten von Plug-ins	180
7.3	Die UI-Plug-in-Bibliothek von jQuery	182
7.3.1	Effekte	182
7.3.2	Interaktionskomponenten	186
7.3.3	Widgets	187
7.3.4	jQuery-UI-ThemeRoller	190
7.4	Zusammenfassung	191
7.5	Übungsaufgaben	191

8	Plug-ins entwickeln	193
8.1	Das Alias \$ innerhalb von Plug-ins verwenden	193
8.2	Neue globale Funktionen hinzufügen	194
8.2.1	Mehrere Funktionen hinzufügen	196
8.3	jQuery Objektmethoden hinzufügen	199
8.3.1	Kontext von Objektmethoden	200
8.3.2	Implizite Iteration	201
8.3.3	Verkettete Methoden	202
8.4	Methodenparameter	203
8.4.1	Parameter-Maps	204
8.4.2	Voreinstellungen für Parameterwerte	205
8.4.3	Callback-Funktionen	206
8.4.4	Anpassbare Voreinstellungen	208
8.5	Die Widget-Factory von jQuery UI	209
8.5.1	Ein Widget erstellen	210
8.5.2	Widgets entfernen	212
8.5.3	Widgets aktivieren und deaktivieren	213
8.5.4	Widget-Optionen übernehmen	213
8.5.5	Untermethoden hinzufügen	214
8.5.6	Widget-Ereignisse auslösen	215
8.6	Designempfehlungen für Plug-ins	216
8.6.1	Plug-ins veröffentlichen	217
8.7	Zusammenfassung	217
8.8	Übungsaufgaben	218
9	Komplexe Selektoren und Durchlaufen des DOM	219
9.1	Auswahl und Durchlaufen – Teil 2	219
9.1.1	Dynamisches Filtern von Tabellen	221
9.1.2	Streifenmuster für Tabellenzeilen	223
9.1.3	Filter und Streifenmuster kombinieren	225
9.1.4	Weitere Selektoren und Traversierungsmethoden	226
9.2	Selektoren anpassen und optimieren	226
9.2.1	Ein eigenes Selektor-Plug-in schreiben	226
9.2.2	Selektor-Performance	228
9.3	Durchlaufen des DOM – Hinter den Kulissen	231
9.3.1	jQuery-Objekteigenschaften	232
9.3.2	Der DOM-Elementstack	234
9.3.3	Ein Plug-in für DOM-Traversierungsmethoden schreiben	235
9.3.4	Performance von DOM-Traversierungsmethoden	237
9.4	Zusammenfassung	239
9.4.1	Literatur	239
9.5	Übungsaufgaben	239

10	Komplexe Ereignisse	241
10.1	Ereignisse – Teil 2	241
10.1.1	Zusätzliche Datenseiten laden	243
10.1.2	Daten beim Darüberfahren mit der Maus anzeigen	244
10.2	Ereignisdelegation	245
10.2.1	Die jQuery-Delegationsmethoden verwenden	246
10.2.2	Eine Delegationsmethode wählen	247
10.2.3	Frühe Delegation	248
10.2.4	Ein Kontextargument verwenden	249
10.3	Benutzerdefinierte Ereignisse	249
10.3.1	Unendliches Scrollen	251
10.3.2	Benutzerdefinierte Ereignisparameter	252
10.4	Ereignisse drosseln	253
10.4.1	Andere Arten der Drosselung	254
10.5	Spezielle Ereignisse	255
10.5.1	Weitere Informationen zu speziellen Ereignissen	257
10.6	Zusammenfassung	257
10.6.1	Literatur	257
10.7	Übungsaufgaben	258
11	Anspruchsvolle Effekte	259
11.1	Animation – Teil 2	259
11.2	Animationen beobachten und unterbrechen	261
11.2.1	Den Animationsstatus bestimmen	262
11.2.2	Eine laufende Animation anhalten	263
11.3	Globale Effekteigenschaften	264
11.3.1	Globales Deaktivieren aller Effekte	264
11.3.2	Feineinstellung der Animationsübergänge	265
11.3.3	Die Effektdauer festlegen	265
11.4	Easing mit mehreren Eigenschaften	268
11.5	Verzögerte Objekte	268
11.5.1	Animations-Promises	270
11.6	Zusammenfassung	273
11.6.1	Literatur	273
11.7	Übungsaufgaben	273

12	DOM-Manipulation für Fortgeschrittene	275
12.1	Tabellenzeilen sortieren	275
12.1.1	Serverseitiges Sortieren	275
12.1.2	Sortierung mit Ajax	276
12.1.3	Sortierung mit JavaScript	277
12.2	Elemente verschieben und einfügen – Teil 2	278
12.2.1	Links um bestehenden Text herum einfügen	278
12.2.2	Einfache JavaScript-Arrays sortieren	279
12.2.3	DOM-Elemente sortieren	280
12.3	Daten zusammen mit DOM-Elementen ablegen	282
12.3.1	Zusätzliche Vorberechnungen	283
12.3.2	Nicht-String-Daten speichern	284
12.3.3	Umkehren der Sortierrichtung	286
12.4	HTML5 mit eigenen Datenattributen einsetzen	288
12.5	Zeilen mit JSON sortieren und erzeugen	290
12.5.1	Das JSON-Objekt modifizieren	292
12.5.2	Inhalte bei Bedarf wiederherstellen	293
12.6	Attributmanipulation für Fortgeschrittene	295
12.6.1	Elementerstellung per Kurzschrift	295
12.6.2	DOM-Manipulation mit Hooks	296
12.7	Zusammenfassung	298
12.7.1	Literatur	299
12.8	Übungsaufgaben	299
13	Ajax für Fortgeschrittene	301
13.1	Fortschreitende Verbesserung mit Ajax	301
13.1.1	JSONP-Daten einsammeln	303
13.2	Ajax-Fehlerbehandlung	306
13.3	Das jqXHR-Objekt	308
13.3.1	Ajax-Promises	309
13.3.2	Antworten cachen	310
13.4	Ajax-Anfragen drosseln	312
13.5	Ajax-Funktionen erweitern	313
13.5.1	Konverter für Datentypen	313
13.5.2	Ajax-Prefilter	318
13.5.3	Alternative Transporte	318
13.6	Zusammenfassung	322
13.6.1	Literatur	322
13.7	Übungsaufgaben	322

A	JavaScript-Closures	323
A.1	Innere Funktionen	323
A.1.1	Gesprengte Ketten	324
A.1.2	Gültigkeitsbereiche von Variablen	326
A.2	Interaktion zwischen Closures	328
A.3	Closures in jQuery	329
A.3.1	Argumente für <code>\$(document).ready()</code>	329
A.3.2	Ereignishandler	330
A.3.3	Handler in Schleifen binden	331
A.3.4	Benannte und anonyme Funktionen	333
A.4	Gefahren durch Speicherlecks	334
A.4.1	Unerwünschte Verweisschleifen	335
A.4.2	Internet Explorer und sein Speicherleck-Problem	336
A.5	Zusammenfassung	338
B	JavaScript mit QUnit testen	339
B.1	QUnit herunterladen	339
B.2	Das Dokument einrichten	340
B.3	Tests organisieren	341
B.4	Tests hinzufügen und ausführen	342
B.4.1	Asynchrones Testen	344
B.5	Andere Testarten	345
B.6	Praktische Erwägungen	346
B.6.1	Literatur	347
B.7	Zusammenfassung	347
C	Kurzreferenz	349
C.1	Selektorausdrücke	349
C.2	Methoden zum Durchlaufen des DOM	352
C.3	Ereignismethoden	354
C.4	Effektmethoden	357
C.5	DOM-Manipulationsmethoden	359
C.6	Ajax-Methoden	362
C.7	Verzögerte Objekte	364
C.8	Verschiedene Eigenschaften und Funktionen	365
	Index	367

Einleitung

Angeregt von Pionieren auf diesem Gebiet wie Dean Edwards und Simon Willison, stellte John Resig 2005 einen Satz von Funktionen zusammen, um den Vorgang zu vereinfachen, durch Programme Elemente auf einer Webseite zu finden und ihnen Verhalten zuzuweisen. Als er sein Projekt im Januar 2006 erstmals der Öffentlichkeit vorstellte, hatte er ihm DOM-Bearbeitungsmöglichkeiten und einfache Animationen hinzugefügt. Er nannte es jQuery, um die zentrale Rolle hervorzuheben, die das Abfragen (*to query*) von Webseiten und ihre Bearbeitung mit JavaScript spielten. In den wenigen Jahren, die seither vergangen sind, wurde der Funktionsumfang von jQuery erweitert und die Leistung verbessert. Viele der beliebtesten Websites im Internet greifen inzwischen auf jQuery zurück. Resig bleibt zwar der leitende Entwickler, doch jQuery ist in echter Open-Source-Manier zu einem Projekt gewachsen, das sich eines Kernteams von JavaScript-Spitzenentwicklern und einer lebendigen Community von Tausenden von Entwicklern rühmen kann.

Die JavaScript-Bibliothek jQuery kann auch Ihre Websites unabhängig von Ihrem Hintergrund aufwerten. Sie bietet eine breite Palette von Funktionen, eine leicht zu erlernende Syntax und eine solide plattformübergreifende Kompatibilität in einer einzigen, kompakten Datei. Überdies wurden Hunderte von Plug-ins entwickelt, um den Funktionsumfang von jQuery zu erweitern und es zu einem unverzichtbaren Werkzeug für praktisch jede clientseitige Skriptaufgabe zu machen.

Dieses Buch gibt eine behutsame Einführung in die Prinzipien von jQuery, damit Sie Ihren Seiten Interaktion und Animationen hinzufügen können – auch wenn frühere Versuche, JavaScript zu schreiben, Sie nur in Verwirrung gestürzt haben. Dieses Buch hilft Ihnen, die Klippen zu umschiffen, die bei Ajax, Ereignissen, Effekten und anspruchsvolleren Merkmalen der Sprache JavaScript lauern. Außerdem fungiert es als kurzes Nachschlagewerk zur Bibliothek jQuery, die sie immer wieder benutzen können.

Der Inhalt dieses Buches

In Kapitel 1, *Erste Schritte*, lernen Sie die JavaScript-Bibliothek jQuery kennen. Das Kapitel beginnt mit einer Beschreibung von jQuery und dem Nutzen für Sie. Anschließend erfahren Sie, wie Sie die Bibliothek herunterladen und einrichten und wie Sie Ihr erstes Skript schreiben.

In Kapitel 2, *Elemente auswählen*, lernen Sie, wie Sie die Selektorausdrücke und DOM-Durchquerungsmethoden von jQuery nutzen, um Elemente auf einer Seite zu finden, wo auch immer sie stecken mögen. Sie verwenden jQuery, um unterschiedliche Seitenelemente zu formatieren, teilweise sogar auf eine Weise, die mit reinem CSS nicht möglich ist.

In Kapitel 3, *Ereignisbehandlung*, nutzen Sie den Ereignisbehandlungsmechanismus von jQuery, um beim Auftreten bestimmter Browserereignisse Verhaltensweisen auszulösen. Sie erfahren, wie Sie mit jQuery auf unaufdringliche und einfache Weise Ereignisse an Elemente anhängen können, selbst wenn die Seite noch nicht vollständig geladen ist. Außerdem erhalten Sie einen Überblick über anspruchsvollere Themen wie Event Bubbling, Delegation und die Verwendung von Namensräumen.

In Kapitel 4, *Formatierung und Animation*, werden die Animationstechniken von jQuery eingeführt. Sie erfahren, wie Sie Seitenelemente mit sowohl nützlichen als auch ästhetischen Effekten ein- und ausblenden und verschieben können.

In Kapitel 5, *DOM-Bearbeitung*, lernen Sie, wie Sie Ihre Seite auf Befehl umgestalten können. Sie erfahren, wie Sie sowohl den Inhalt als auch die Struktur eines HTML-Dokuments im laufenden Betrieb ändern können.

In Kapitel 6, *Daten mit Ajax senden*, lernen Sie die verschiedenen Möglichkeiten kennen, mit denen jQuery den Zugriff auf serverseitige Funktionen ohne hinderliche Seitenaktualisierungen vereinfacht. Nachdem Sie nun die grundlegenden Bestandteile der Bibliothek beherrschen, können Sie sich genauer ansehen, wie Sie sie erweitern können, um sie an Ihre Bedürfnisse anzupassen.

Kapitel 7, *Plug-ins verwenden*, zeigt Ihnen, wie Sie Plug-ins finden, installieren und verwenden, u.a. die leistungsfähige Plug-in-Bibliothek jQuery UI.

In Kapitel 8, *Plug-ins entwickeln*, lernen Sie, wie Sie die eindrucksvollen Erweiterungsfähigkeiten von jQuery nutzen können, um eigene Plug-ins von Grund auf zu erstellen. Sie legen hier eigene Hilfsfunktionen an, fügen jQuery-Objektmethoden hinzu und sehen sich die Widget-Factory von jQuery UI näher an. Danach beschäftigen wir uns erneut mit den Grundbausteinen von jQuery, um einige anspruchsvollere Techniken zu erlernen.

In Kapitel 9, *Komplexe Selektoren und Durchlaufen des DOM*, erweitern Sie Ihre Kenntnisse über Selektoren und das Durchlaufen des DOM. Hier erwerben Sie die Fähigkeit, die Leistung von Selektoren zu optimieren, den DOM-Elementstack zu bearbeiten und Plug-ins zu schreiben, die die Auswahl- und Durchlaufmöglichkeiten erweitern.

In Kapitel 10, *Komplexe Ereignisse*, beschäftigen Sie sich eingehender mit Techniken wie Delegation und Drosselung, mit denen sich die Leistung der Ereignisbehandlung erheblich verbessern lässt. Außerdem erstellen Sie eigene und besondere Ereignisse, die die Möglichkeiten von jQuery noch erweitern.

In Kapitel 11, *Anspruchsvolle Effekte*, optimieren Sie die grafischen Effekte, die jQuery bietet, indem Sie benutzerdefinierte Easing-Funktionen erstellen und auf jeden Schritt einer Animation reagieren. Hier erhalten Sie die Möglichkeit, Animationen zu bearbeiten, während sie auftreten, und mit benutzerdefinierten Warteschleifen Aktionen nach Zeitplan ablaufen zu lassen.

In Kapitel 12, *DOM-Manipulation für Fortgeschrittene*, vertiefen Sie Ihre praktischen Kenntnisse in der Bearbeitung des DOM mit Techniken wie dem Anhängen beliebiger Daten an Elemente. Außerdem lernen Sie, wie Sie die Verarbeitung der CSS-Eigenschaften von Elementen durch jQuery erweitern.

Kapitel 13, *Ajax für Fortgeschrittene*, gibt Ihnen tiefere Einblicke in Ajax-Transaktionen, unter anderem in das jQuery-System für verzögerte Objekte, mit dem Daten gehandhabt werden, die erst später verfügbar werden.

In Anhang A, *JavaScript-Closures*, erhalten Sie solide Grundkenntnisse von Closures in JavaScript. Sie erfahren, worum es sich dabei handelt und wie Sie sie zu Ihrem Vorteil einsetzen können.

In Anhang B, *JavaScript mit QUnit testen*, lernen Sie die Bibliothek QUnit kennen, die für Unit-Tests von JavaScript-Programmen da ist. Diese Bibliothek ist eine wichtige Ergänzung Ihres Werkzeugkastens zur Entwicklung und Wartung anspruchsvoller Webanwendungen.

Anhang C, *Kurzreferenz*, gibt einen Überblick über die gesamte Bibliothek jQuery und führt dabei sämtliche Methoden und Selektorausdrücke auf. Die übersichtliche Gestaltung ist praktisch, wenn Sie wissen, was Sie tun wollen, aber nicht auf den richtigen Namen der gewünschten Methode bzw. des Selektors kommen.

Voraussetzungen für dieses Buch

Um den Beispielcode auszuführen, der in diesem Buch vorgestellt wird, benötigen Sie einen modernen Webbrowser wie Mozilla Firefox, Apple Safari, Google Chrome oder Microsoft Internet Explorer.

Um mit den Beispielen herumzuspielen und die am Kapitelende aufgeführten Übungsaufgaben zu bearbeiten, benötigen Sie außerdem Folgendes:

- Einen einfachen Texteditor
- Webentwicklungstools für Ihren Browser wie z.B. Firebug (siehe Kapitel 1 im Abschnitt »Entwicklungswerkzeuge«)
- Das komplette Codepaket für jedes Kapitel. Darin ist auch eine Kopie der Bibliothek jQuery enthalten (siehe den Abschnitt »Den Beispielcode herunterladen« weiter hinten).

Um einige der Ajax-Beispiele auszuführen, die ab Kapitel 6 vorgestellt werden, brauchen Sie einen Webserver mit aktiviertem PHP.

Zielgruppe dieses Buches

Dieses Buch ist für Webdesigner gedacht, die interaktive Elemente erstellen möchten, und für Entwickler, die die bestmöglichen Benutzerschnittstellen für ihre Webanwendungen gestalten wollen. Grundkenntnisse in JavaScript-Programmierung werden vorausgesetzt. Sie müssen die Grundlagen von HTML und CSS kennen und sollten mit der Syntax von JavaScript vertraut sein. Vorkenntnisse in jQuery oder mit anderen JavaScript-Bibliotheken werden nicht benötigt.

Dieses Buch macht Sie mit dem Funktionsumfang und der Syntax von jQuery 1.6.x vertraut, der zurzeit neuesten Version.

Die Geschichte des Projekts jQuery

Dieses Buch deckt den Funktionsumfang und die Syntax von jQuery 1.6.x ab, der zurzeit neuesten Version. Der Tenor der Bibliothek – eine einfache Möglichkeit zu bieten, um Elemente auf einer Webseite zu finden und sie zu bearbeiten – hat sich im Verlauf der Entwicklung nicht geändert, aber die Einzelheiten der Syntax und der Funktionsmerkmale. Diese kurze Übersicht über die Geschichte des Projekts beschreibt die wichtigsten Änderungen von einer Version zur nächsten, was für Leser, mit älteren Versionen der Bibliothek arbeiten, hilfreich sein mag.

- **Phase der öffentlichen Entwicklung:** Im August 2005 erwähnte John Resig erstmals eine Verbesserung der Prototype-Bibliothek Behavior. Dieses neue Framework wurde am 14. Januar 2006 formal als jQuery veröffentlicht.
- **jQuery 1.0** (August 2006): Dieses erste stabile Release der Bibliothek bot bereits eine solide Unterstützung für CSS-Selektoren, Ereignisbehandlung und AJAX-Interaktion.
- **jQuery 1.1** (Januar 2007): Mit diesem Release wurde die API erheblich verschlankt. Viele selten genutzte Methoden wurden kombiniert, sodass weniger Methoden zu lernen und zu dokumentieren waren.
- **jQuery 1.1.3** (Juli 2007): Dieses untergeordnete Release bot erhebliche Geschwindigkeitssteigerungen der Selektor-Engine von jQuery. Seit dieser Option übertraf die Leistung von jQuery die anderer JavaScript-Bibliotheken wie Prototype, Mootools und Dojo.
- **jQuery 1.2** (September 2007): In diesem Release wurde die XPath-Syntax für die Elementauswahl entfernt, da sie zur CSS-Syntax redundant geworden

war. Die Gestaltung von Effekten wurde in diesem Release flexibler, und die Entwicklung Plug-ins wurde durch die Ergänzung von Ereignis-Namensräumen vereinfacht.

- **jQuery UI** (September 2007): Die neue Plug-in-Suite wurde als Ersatz für das beliebte, aber veraltete Plug-in Interface angekündigt. Es enthielt eine reichhaltige Sammlung vorgefertigter Widgets sowie einen Satz von Werkzeugen zur Gestaltung anspruchsvoller Elemente wie Drag-&-Drop-Oberflächen.
- **jQuery 1.2.6** (Mai 2008): Die Funktionsmerkmale des beliebten Plug-ins Dimensions von Brandon Aaron wurden in die Hauptbibliothek aufgenommen.
- **jQuery 1.3** (Januar 2009): Eine Generalüberholung der Selektor-Engine (Sizzle) bot einen gewaltigen Schub für die Leistung der Bibliothek. Die Ereignisdelegierung wurde jetzt formal unterstützt.
- **jQuery 1.4** (Januar 2010): Diese Version, die wahrscheinlich ehrgeizigste Aktualisierung seit 1.0, brachte viele Leistungsverbesserungen für die DOM-Bearbeitung sowie eine große Menge neuer oder verbesserter Methoden für fast jeden Aspekt der Bibliothek. Das Erscheinen Version 1.4 wurde vierzehn Tage lang mit Ankündigungen und Videos auf einer eigens dazu angelegten Website (<http://jquery14.com/>) begleitet.
- **jQuery 1.4.2** (Februar 2010): Zwei neue Methoden zur Ereignisdelegierung wurden hinzugefügt (`.delegate()` und `.undelegate()`). Das gesamte Ereignissystem von jQuery wurde generalüberholt, um eine flexiblere Benutzung und eine größere browserübergreifende Konsistenz zu erreichen.
- **jQuery Mobile** (August 2010): Das jQuery-Projekt skizzierte öffentlich seine Strategie, seine Forschung und seine Benutzerschnittstellenentwürfe für die mobile Webelemente mit jQuery und einem neuen Mobilframework unter <http://jquerymobile.com/>.
- **jQuery 1.5** (Januar 2011): Die Ajax-Komponenten wurden erheblich umgeschrieben, um die Erweiterbarkeit und Leistung zu verbessern. Außerdem enthielt jQuery 1.5 eine Implementierung des Promise-Musters zur Handhabung von Abfragen von sowohl synchronen als auch asynchronen Funktionen.
- **jQuery 1.6** (Mai 2011): Die Komponente Attribute wurde umgeschrieben, um die Unterscheidung zwischen HTML-Attributen und DOM-Eigenschaften genauer widerzuspiegeln. Außerdem erhielt das in jQuery 1.5 eingeführte Objekt Deferred die beiden neuen Methoden `.always()` und `.pipe()`.

Historische Einzelheiten

Release Notes für ältere jQuery-Versionen sind auf der Projektwebsite unter <http://jquery.org/history> zu finden.

Schreibweisen

In diesem Buch werden verschiedene Arten von Informationen durch unterschiedliche Formatierung des Textes gekennzeichnet. Im Folgenden finden Sie einige Beispiele für diese Formate und eine Erklärung ihrer Bedeutung.

Einzelne Codebegriffe werden im Text wie folgt dargestellt: »Dieser Code zeigt, dass wir der Methode `console.log()` jede Art von Ausdruck übergeben können.«

Ein ganzer Block von Code sieht folgendermaßen aus:

```
$('#button.show-details').click(function() {  
    $('#div.details').show();  
});
```

Um die Aufmerksamkeit auf einen bestimmten Teil eines Codeblocks zu lenken, sind die entsprechenden Zeilen oder Elemente fett hervorgehoben:

```
$('#switcher-narrow').bind('click', function() {  
    $('#body').removeClass().addClass('narrow');  
});
```

Neue Begriffe und *wichtige Wörter* werden kursiv hervorgehoben. Begriffe, die Sie auf dem Bildschirm sehen, also z.B. in Menüs oder Dialogfeldern, werden folgendermaßen gekennzeichnet: »Die Registerkarte `CONSOLE`, die Sie im folgenden Screenshot sehen, werden wir am häufigsten verwenden, während wir den Umgang mit jQuery erlernen.«

Warnungen, wichtige Hinweise sowie Tipps und Tricks erscheinen in einem Kasten wie diesem.

Herunterladen des Beispielcodes

Die Dateien mit dem Beispielcode können Sie für dieses Buch unter <http://www.dpunkt.de/jquery> herunterladen.

1 Erste Schritte

Das World Wide Web von heute stellt ein dynamisches Umfeld dar und seine Benutzer haben hohe Ansprüche sowohl an die Gestaltung als auch an die Funktion von Websites. Entwickler, die bemerkenswerte, interaktive Sites erstellen wollen, nutzen JavaScript-Bibliotheken wie jQuery, um häufig vorkommende Aufgaben zu automatisieren und komplizierte zu vereinfachen. Einer der Gründe für die Beliebtheit von jQuery besteht darin, dass diese Bibliothek bei einer breiten Palette von Aufgaben Hilfestellung geben kann.

Da jQuery so viele verschiedene Funktionen bietet, erscheint es schwierig, einen Anfang zu finden. Die Bibliothek ist jedoch logisch und konsistent aufgebaut, und viele ihrer Konzepte entstammen den Strukturen von HTML und CSS (Cascading Style Sheets). Die Gestaltung der Bibliothek ermöglicht auch Designern mit geringen Programmierkenntnissen einen schnellen Einstieg, da viele von ihnen mit diesen Technologien mehr Erfahrung haben als mit JavaScript. In diesem ersten Kapitel werden wir tatsächlich ein funktionierendes jQuery-Programm schreiben, das aus lediglich drei Zeilen Code besteht. Auch erfahrenen Programmierern hilft die konzeptuelle Klarheit, wie wir in späteren, anspruchsvolleren Kapiteln noch sehen werden.

Schauen wir uns also an, was jQuery für uns tun kann.

1.1 Was bietet jQuery?

Die Bibliothek jQuery bietet eine Allzweck-Abstraktionsschicht für die gängige Skripterstellung im Web und eignet sich daher für fast jede Situation, in der Sie Skripte einsetzen müssen. Da sie erweiterbar ist und ständig Plug-ins entwickelt werden, um neue Fähigkeiten hinzuzufügen, können wir niemals alle möglichen Verwendungsmöglichkeiten und Funktionen in einem Buch beschreiben. Die Kernfunktionen jedoch helfen bei der Erledigung der folgenden Aufgaben:

- Zugriff auf Elemente in einem Dokument: Ohne JavaScript-Bibliothek müssen Entwickler häufig viele Zeilen Code schreiben, um den *DOM*-Baum (*Document Object Model*) zu durchlaufen und einzelne Teile der Struktur

eines HTML-Dokuments zu finden. Mit jQuery steht den Entwicklern ein robuster und effizienter Selektionsmechanismus zur Verfügung, der es einfach macht, genau den Teil eines Dokuments abzurufen, den Sie untersuchen oder bearbeiten wollen.

```
$('#div.content').find('p');
```

- **Ändern des Erscheinungsbilds einer Webseite:** CSS bietet sich als leistungsfähige Methode an, die Darstellung eines Dokuments zu beeinflussen, leidet jedoch darunter, dass Webbrowser die Standards unterschiedlich oder nur unvollständig unterstützen. Mit jQuery können Entwickler diese Lücke schließen und auf einer browserunabhängigen Abstraktion aufsetzen. Außerdem kann jQuery Klassen und einzelne Formateigenschaften, die einem Teil des Dokuments zugewiesen sind, auch dann noch ändern, wenn dieses bereits vom Browser dargestellt wird.

```
$('#ul > li:first').addClass('active');
```

- **Ändern des Dokumentinhalts:** jQuery ist nicht auf die Modifikation der reinen Darstellung beschränkt, sondern kann auch den Inhalt eines Dokuments ändern, wozu nur wenige Tastendrucke erforderlich sind. So ist es möglich, Text zu ändern, Bilder einzufügen oder auszutauschen, Listen umzusortieren oder die gesamte HTML-Struktur umzustellen und zu erweitern, und das alles mit einer einzigen, einfach zu verwendenden *Anwendungsprogrammierschnittstelle* (*Application Programming Interface, API*).

```
$('#container').append('<a href="more.html">more</a>');
```

- **Reagieren auf die Aktionen des Benutzers:** Selbst die raffiniertesten und leistungsfähigsten Verhaltensweisen nützen nichts, wenn wir nicht steuern können, wann sie stattfinden. Die Bibliothek jQuery bietet eine elegante Möglichkeit, ein breites Spektrum an Ereignissen, z.B. den Klick auf einen Link, abzufangen, ohne den HTML-Code dabei übermäßig mit deren Behandlung zu durchsetzen. Gleichzeitig überbrückt die API zur Ereignisbehandlung Browserunterschiede, die Webentwicklern häufig das Leben schwer machen.

```
$('#button.show-details').click(function() {
    $('#div.details').show();
});
```

- **Animierte Darstellung von Änderungen an einem Dokument:** Um solche interaktiven Änderungen wirkungsvoll umzusetzen, muss ein Designer dem Benutzer auch eine optische Rückmeldung geben. Die Bibliothek jQuery ermöglicht dies durch einen reichen Fundus an Effekten wie Fade (Ausblenden) und Wipe (Wegwischen) sowie durch einen Werkzeugkasten zum Gestalten eigener grafischer Effekte.

```
$('#div.details').slideDown();
```

- Abrufen von Informationen von einem Server, ohne die ganze Seite aktualisieren zu müssen: Dieses Codemuster ist als Ajax bekannt geworden, was ursprünglich für »asynchronous JavaScript and XML« stand, inzwischen aber eine viel breitere Palette von Technologien für die Kommunikation zwischen Client und Server umfasst. Die Bibliothek jQuery nimmt diesem aufwendigen und sehr vom darstellenden Endgerät abhängigen Vorgang die Komplexität browserspezifischer Aspekte, sodass sich Entwickler auf die eigentliche Serverfunktionalität konzentrieren können.

```
$('#div.details').load('more.html #content');
```

- Vereinfachen häufig vorkommender JavaScript-Aufgaben: Neben all den dokumentenspezifischen Merkmalen bietet jQuery auch Erweiterungen zu grundlegenden JavaScript-Konstrukten wie Iteration und Array-Manipulation.

```
$.each(obj, function(key, value) {  
    total += value;  
});
```

Den Beispielcode herunterladen

Die Dateien mit dem Beispielcode können Sie unter <http://www.dpunkt.de/jquery> herunterladen.

1.2 Warum jQuery so gut funktioniert

Mit dem Wiederaufleben des Interesses an dynamischem HTML kam es zu einer starken Vermehrung der JavaScript-Frameworks. Manche sind spezialisiert und konzentrieren sich nur auf ein oder zwei Aufgaben, während andere versuchen, alle möglichen Verhaltensweisen und Animationen zu katalogisieren und sie vorgefertigt anzubieten. Um das oben angedeutete breite Spektrum an Funktionen zu bieten und gleichzeitig relativ kompakt zu bleiben, werden bei jQuery die folgenden Strategien eingesetzt:

- Nutzen von CSS-Kenntnissen: Indem der Mechanismus zum Auffinden von Seitenelementen auf CSS-Selektoren basiert, erbt jQuery eine knappe und doch lesbare Art, um die Struktur eines Dokuments auszudrücken. Die Bibliothek jQuery ist ein Einstiegspunkt für Designer, die ihren Seiten Verhaltensweisen hinzufügen möchten, da Kenntnisse der CSS-Syntax eine Voraussetzung für die professionelle Webentwicklung sind.
- Unterstützung von Erweiterungen: Um einen schleichenden Funktionszuwachs zu vermeiden, werden bei jQuery Sonderfälle in *Plug-ins* verbannt. Die Methode zum Erstellen neuer Plug-ins ist einfach und gut dokumentiert, was die Entwicklung einer großen Vielfalt origineller und nützlicher Module gefördert hat. Die meisten Funktionen im grundlegenden jQuery-Download sind intern durch die Plug-in-Architektur realisiert und können falls gewünscht entfernt werden, was zu einer noch kleineren Bibliothek führt.

- Wegabstrahieren von Eigenheiten der Browser: Es ist eine bedauernde Tatsache in der Webentwicklung, dass jeder Browser seine eigenen Bereiche hat, in denen er von den öffentlichen Standards abweicht. Ein erheblicher Anteil jeder Webanwendung kann ausgelagert werden, um sich um die Merkmale zu kümmern, die auf jeder Plattform anders sind. Während die im ständigen Wandel begriffene Browserlandschaft es unmöglich macht, für bestimmte anspruchsvolle Funktionen vollständig browserneutralen Code zu schreiben, fügt jQuery eine *Abstraktionsschicht* hinzu, die häufig vorkommende Aufgaben normalisiert, den Code erheblich vereinfacht und seinen Umfang verringert.
- Konsequente Verwendung von Mengen: Wenn wir jQuery anweisen: »Finde alle Elemente der Klasse `collapsible` und blende sie aus (`»hide them«`)«, besteht keine Notwendigkeit, jedes zurückgegebene Element in einer Schleife durchzugehen. Stattdessen sind Methoden wie `.hide()` dazu da, automatisch Mengen von Objekten zu verarbeiten statt Einzelobjekte. Diese Technik, die *implizite Iteration*, bedeutet, dass viele Schleifenkonstrukte unnötig werden, was den Code erheblich verkürzt.
- Zulassen mehrerer Aktionen in einer Zeile: Um die übermäßige Verwendung temporärer Variablen und verschwenderische Wiederholungen zu vermeiden, nutzt jQuery für die meisten seiner Methoden das Programmiermuster der *Verkettung*. Das bedeutet, dass das Ergebnis der meisten Operationen an einem Objekt das Objekt selbst ist, das gleich wieder für die nächste Aktion bereitsteht.

Diese Strategien haben dafür gesorgt, dass das jQuery-Paket schlank bleibt – komprimiert ungefähr 30 KB –, während es gleichzeitig Techniken bereitstellt, die unseren selbst geschriebenen Code, der die Bibliothek nutzt, kompakt halten.

Die Eleganz dieser Bibliothek gründet sich zum Teil auf das Design, zum Teil auf den evolutionären Prozess, den die rund um das Projekt aufgekommene, lebendige Community fördert. Benutzer von jQuery diskutieren nicht nur die Entwicklung von Plug-ins, sondern auch Verbesserungen an der Kernbibliothek. Außerdem tragen sowohl Benutzer als auch Entwickler zu der ständigen Verbesserung der offiziellen Projektdokumentation bei, die Sie unter <http://api.jquery.com> finden können.

Obwohl zur Konstruktion eines so flexiblen und stabilen Systems viel Arbeit erforderlich ist, kann das Endprodukt kostenlos genutzt werden. Dieses Open-Source-Projekt steht unter den beiden Lizenzen *MIT Licence* (zur freien Verwendung von jQuery auf jeder Site und zu seiner Nutzung in proprietärer Software) und *GNU Public Licence* (für die Einbindung in andere Open-Source-Projekte mit GNU-Lizenz) zur Verfügung.

1.3 Unsere erste Webseite mit jQuery

Nachdem wir uns damit befasst haben, welcher Umfang an Funktionalität in jQuery zur Verfügung steht, sehen wir uns nun an, wie wir die Bibliothek tatsächlich nutzen. Dazu benötigen wir zunächst eine Kopie von jQuery.

1.3.1 jQuery herunterladen

Eine Installation ist nicht erforderlich. Um jQuery verwenden zu können, brauchen wir lediglich eine öffentlich verfügbare Kopie der Datei, die sich auf einer externen Site oder auf unserer eigenen befinden kann. Da JavaScript eine interpretierte Sprache ist, gibt es keine Kompilierung und keine Build-Phase, über die Sie sich Sorgen machen müssten. Wann immer wir auf einer Seite jQuery zur Verfügung haben müssen, verweisen wir einfach in einem `<script>`-Element des HTML-Dokuments auf den Speicherort der Datei.

Die offizielle jQuery-Website (<http://jquery.com/>) beinhaltet stets die aktuellste stabile Version der Bibliothek, die Sie gleich von der Homepage herunterladen können. Es kann sein, dass gleichzeitig mehrere Versionen von jQuery verfügbar sind. Für uns als Websiteentwickler ist die neueste unkomprimierte Version geeignet. In Produktionsumgebungen kann sie durch eine komprimierte Version ersetzt werden.

Angesichts der steigenden Beliebtheit von jQuery haben manche Unternehmen die Datei in ihren CDNs (*Content Delivery Networks*) kostenlos zur Verfügung gestellt. Vor allem Google (<http://code.google.com/apis/ajaxlibs/documentation/>) und Microsoft (<http://www.asp.net/ajax/cdn>) bieten die Datei auf leistungsfähigen, verzögerungsarmen Servern überall auf der Welt an, um einen schnellen Download unabhängig vom Standort des Benutzers sicherzustellen. Die CDN-Kopien von jQuery bieten aufgrund der Serververteilung und Zwischenspeicherung zwar Geschwindigkeitsvorteile, doch kann während der Entwicklung die Verwendung einer lokalen Kopie praktischer sein. In den Beispielen in diesem Buch nutzen wir eine Kopie der Datei in unserem eigenen System, sodass wir den Code ausführen können, unabhängig davon, ob wir mit dem Internet verbunden sind oder nicht.

1.3.2 Einrichten von jQuery in einem HTML-Dokument

Die meisten Beispiele für die Nutzung von jQuery bestehen aus drei Teilen: dem HTML-Dokument, der CSS-Datei zu seiner Formatierung und den JavaScript-Dateien, die die Interaktionen definieren. Als erstes Beispiel verwenden wir eine Seite mit dem Auszug aus einem Buch, dessen einzelnen Teilen eine Reihe von Klassen zugewiesen sind. Diese Seite enthält einen Verweis auf die neueste Version der Bibliothek jQuery, die wir heruntergeladen, in `jquery.js` umbenannt und in unserem lokalen Projektverzeichnis platziert haben. Das HTML-Dokument sieht wie folgt aus: