

2. Auflage

ralph STEYER

jQuery

Das universelle
JavaScript-Framework
für das interaktive Web
und mobile Apps

HANSER



Im Internet: alle Beispielfcodes zum
Download

Bleiben Sie auf dem Laufenden!



Unser **Computerbuch-Newsletter** informiert Sie monatlich über neue Bücher und Termine. Profitieren Sie auch von Gewinnspielen und exklusiven Leseproben. Gleich anmelden unter



www.hanser-fachbuch.de/newsletter



Hanser Update ist der IT-Blog des Hanser Verlags mit Beiträgen und Praxistipps von unseren Autoren rund um die Themen Online Marketing, Webentwicklung, Programmierung, Softwareentwicklung sowie IT- und Projektmanagement. Lesen Sie mit und abonnieren Sie unsere News unter



www.hanser-fachbuch.de/update



Ralph Steyer

jQuery

Das universelle JavaScript-Framework
für das interaktive Web
und mobile Apps

2., aktualisierte Auflage

HANSER

Der Autor:

Ralph Steyer, Bodenheim

Kontakt: ralph.steyer@rjs.de

Alle in diesem Buch enthaltenen Informationen, Verfahren und Darstellungen wurden nach bestem Wissen zusammengestellt und mit Sorgfalt getestet. Dennoch sind Fehler nicht ganz auszuschließen. Aus diesem Grund sind die im vorliegenden Buch enthaltenen Informationen mit keiner Verpflichtung oder Garantie irgendeiner Art verbunden. Autor und Verlag übernehmen infolgedessen keine juristische Verantwortung und werden keine daraus folgende oder sonstige Haftung übernehmen, die auf irgendeine Art aus der Benutzung dieser Informationen – oder Teilen davon – entsteht.

Ebenso übernehmen Autor und Verlag keine Gewähr dafür, dass beschriebene Verfahren usw. frei von Schutzrechten Dritter sind. Die Wiedergabe von Gebrauchsnamen, Handelsnamen, Warenbezeichnungen usw. in diesem Buch berechtigt deshalb auch ohne besondere Kennzeichnung nicht zu der Annahme, dass solche Namen im Sinne der Warenzeichen- und Markenschutz-Gesetzgebung als frei zu betrachten wären und daher von jedermann benutzt werden dürften.

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.

Dieses Werk ist urheberrechtlich geschützt.

Alle Rechte, auch die der Übersetzung, des Nachdruckes und der Vervielfältigung des Buches, oder Teilen daraus, vorbehalten. Kein Teil des Werkes darf ohne schriftliche Genehmigung des Verlages in irgendeiner Form (Fotokopie, Mikrofilm oder ein anderes Verfahren) – auch nicht für Zwecke der Unterrichtsgestaltung – reproduziert oder unter Verwendung elektronischer Systeme verarbeitet, vervielfältigt oder verbreitet werden.

© 2018 Carl Hanser Verlag München, www.hanser-fachbuch.de

Lektorat: Brigitte Bauer-Schiewek

Copy editing: Petra Kienle, Fürstenfeldbruck

Umschlagdesign: Marc Müller-Bremer, München, www.rebranding.de

Umschlagrealisation: Stephan Rönigk

Gesamtherstellung: Kösel, Krugzell

Ausstattung patentrechtlich geschützt. Kösel FD 351, Patent-Nr. 0748702

Printed in Germany

Print-ISBN: 978-3-446-45558-0

E-Book-ISBN: 978-3-446-45651-8

Inhalt

Vorwort	XV
1 Einleitung	1
1.1 Das Umfeld	1
1.2 Frameworks und Toolkits	2
1.3 Was behandeln wir in diesem Buch?	3
1.3.1 Das Kern-Framework jQuery	4
1.3.2 Plugins, UI, Mobile & Co	4
1.3.3 Wen sehe ich als Zielgruppe für das Buch?	5
1.4 Schreibkonventionen	6
1.4.1 Listings	6
1.5 Was benötigen Sie beziehungsweise was nutzt Ihnen?	6
1.5.1 Hardware und Betriebssystem	6
1.5.2 jQuery, jQuery Mobile, jQuery UI & mehr	7
1.6 Die Browser	14
1.6.1 Verschiedene Betriebssysteme und virtuelle Maschinen zum Testen	15
1.7 Die Entwicklungsumgebungen und nützliche Helferlein	16
1.7.1 Editoren – oft bereits mit gewisser Hilfestellung	17
1.7.2 Integrierte Entwicklungsumgebungen	17
1.7.3 Kleine Helferlein	22
1.7.4 Node.js und Git	23
1.7.5 Integrierte Entwicklungstools in Browser und Browser-Add-ons	24
1.7.6 Der Webserver zum realistischen Testen	25
1.8 Zusammenfassung	26
2 Sprung ins kalte Wasser	27
2.1 Zugriff auf Elemente und Schutz des DOM	27
2.2 Veränderung der Webseite mit DHTML	33
2.3 Animiertes Verkleinern und Vergrößern eines Elements	36
2.4 Attribute dynamisch verändern	41
2.5 Zusammenfassung	43

3	Grundlagenwissen	45
3.1	Das WWW, das Web 2.0 und das Client-Server-Prinzip im Internet	45
3.1.1	Programmierung im WWW	46
3.1.2	Das Web 2.0 und Ajax	46
3.2	JavaScript und das Verhältnis zu jQuery	47
3.2.1	Die allgemeine Einbindung von JavaScript in Webseiten	48
3.2.2	JSON	50
3.3	DOM und Objekte	51
3.3.1	DOM und der Zugriff auf Elemente einer Webseite	52
3.4	Style Sheets und DHTML	53
3.4.1	CSS – die Standardsprache des Webs	53
3.4.2	Die konkrete Syntax von CSS-Deklarationen	54
3.4.3	Selektoren	55
3.5	Zusammenfassung	55
4	Wie jQuery grundsätzlich arbeitet	57
4.1	Grundsätzliches zum Zugriff auf Elemente der Webseite	58
4.1.1	Ein Beispiel für unterschiedliche Knoten	59
4.2	Der jQuery-Namensraum und das jQuery-Objekt	62
4.3	Spezielle Datentypen und Strukturen in Query	63
4.3.1	Methoden	63
4.3.2	Optionen	64
4.3.3	jqXHR	64
4.4	Die Funktion jQuery() und der Alias \$()	64
4.4.1	Der Kontext	66
4.4.2	Verketteten von Methoden und die jQuery-Warteschlange	67
4.4.3	Funktionsaufrufe nacheinander ausführen – die jQuery-Warteschlange	67
4.4.4	jQuery.then() und jQuery.when()	68
4.5	Funktionen nach Aufbau des DOM ausführen	69
4.5.1	Callback oder anonyme Funktion als Parameter für jQuery()	69
4.5.2	Das ready-Event	71
4.5.3	document.ready() in eine externe JavaScript-Datei auslagern	72
4.5.4	Mit jQuery.holdReady() das ready-Event beeinflussen	73
4.6	Ein Element mit jQuery() erstellen und in die Webseite einfügen	73
4.6.1	Ein trickreiches kleines Problem – können Sie es lösen?	75
4.6.2	Optionen zum Initialisieren von Attributen	79
4.7	Direkter Zugriff auf DOM-Elemente mit get()	81
4.8	Gemeinsame Verwendung von jQuery und anderen Frameworks	81
4.8.1	Die Funktion jQuery.noConflict()	82
4.8.2	Einen alternativen Alias verwenden	83
4.9	Datenspeicherung im DOM	84

4.10 Mehr zum Kontext und jQuery-Utilities	90
4.11 Zusammenfassung	92
5 Umgang mit Selektoren und Filtern	93
5.1 Klassische DOM-Zugriffsmethoden	93
5.1.1 Objektfelder	94
5.1.2 Zugriff über einen Namen	94
5.1.3 Verwandtschaftsbeziehungen	94
5.1.4 Elementnamen, IDs und Klassennamen	94
5.1.5 DOM-Zugriffsmöglichkeiten versus jQuery	95
5.2 Was versteht man unter Selektoren?	95
5.2.1 Was ist ein Selektor?	95
5.2.2 Was sind Filter?	96
5.2.3 CSS3, SQL und XPath als Grundlagen und Hintergrund	96
5.3 Basisselektoren und hierarchische Selektoren	97
5.3.1 Beispiele zur Verdeutlichung	99
5.3.2 Potenzielle Fallen	110
5.4 Filterausdrücke	110
5.4.1 Basisfilter	111
5.4.2 Inhaltsfilter	118
5.4.3 Sichtbarkeitsfilter	121
5.4.4 Kindfilter	125
5.4.5 Attributfilter	128
5.4.6 Filter für Formularelemente und Formularfilter	133
5.4.7 Formularfilter zur Auswahl aufgrund des Zustands	137
5.5 Filtermethoden	139
5.5.1 Praktische Beispiele von Filtermethoden	140
5.6 Zusammenfassung	149
6 Zugriff auf die Elemente einer Webseite	151
6.1 Inhalte von Knoten abfragen und verändern – html() und text()	151
6.1.1 Ein Beispiel zu html() und text()	152
6.2 Inhalt von Formularfeldern – val()	156
6.2.1 Ein Beispiel für den Zugriff auf Formularfelder	156
6.3 Zugriff auf Attribute und Eigenschaften mit attr() und prop()	158
6.3.1 Zugriff auf Attribute	158
6.3.2 Zugriff auf Eigenschaften	158
6.3.3 Unterschied zwischen Attributen und Eigenschaften	159
6.3.4 Das Beispiel zum Zugriff auf Eigenschaften und Attribute	160
6.4 Einfügen von Knoten in eine Webseite	166
6.4.1 append() und prepend()	166
6.4.2 appendTo() und prependTo()	171

6.5	Knoten nachher oder vorher einfügen	176
6.5.1	after() und before()	177
6.5.2	insertAfter() und insertBefore()	177
6.6	Ummanteln	177
6.6.1	Einzelnen mit wrap() ummanteln	177
6.6.2	Alles ummanteln mit wrapAll()	180
6.6.3	Innere Bereiche ummanteln mit wrapInner()	181
6.6.4	Den Mantel ablegen - unwrap()	182
6.7	Ersetzen mit replaceWith() und replaceAll()	182
6.7.1	Ersetzen mit replaceWith()	182
6.7.2	Alles ersetzen mit replaceAll()	186
6.8	Überflüssige Knoten entfernen	188
6.8.1	Die Methoden empty() und remove()	188
6.8.2	Die Alternative zu remove() - detach()	194
6.8.3	Löschen von Attributen	195
6.9	Vervielfachen mit clone()	195
6.10	Suchen & Finden	195
6.10.1	Von Kindern und Eltern	196
6.10.2	Von Geschwistern	199
6.10.3	Nachfolger mit has() suchen	201
6.11	Finden mit find() und contents()	201
6.12	Mit each() und map() über Arrays und Objekte iterieren	203
6.12.1	jQuery.each() und jQuery.map()	204
6.12.2	Die Methoden each() und map()	209
6.13	Die Methode add()	210
6.14	Die Methoden end() und andSelf()	212
6.15	Zusammenfassung	213
7	Layout und Formatieren mit Style Sheets unter jQuery	215
7.1	Hintergrundinformationen	215
7.1.1	CSS in jQuery - Vermischung von Layout und Funktionalität?	216
7.2	Die Methode css()	217
7.2.1	Abfragen von Stileigenschaften	217
7.2.2	Setzen von Eigenschaften	217
7.3	Klassen von Elementen verändern	218
7.3.1	Klassen hinzufügen - addClass()	218
7.3.2	Klassen wegnehmen - removeClass()	227
7.3.3	Klassen umschalten mit toggleClass()	227
7.3.4	Test auf eine Klasse -hasClass()	228
7.4	Methoden zur Positionierung	228
7.4.1	Bestimmen der Position mit position()	229
7.4.2	Position relativ zum Dokument - offset()	234
7.4.3	Methoden zum Scrollen	238

7.5	Höhe und Breite	241
7.5.1	height() und width()	241
7.6	Innere und äußere Maße	245
7.7	jQuery.cssHooks	248
7.8	Zusammenfassung	250
8	Ein Praxisbeispiel – eine Datumskomponente	251
8.1	Das Ziel	251
8.1.1	Die Basiswebseite	252
8.2	Die CSS-Datei – Templates	253
8.3	Die JavaScript-Datei	254
8.4	Zusammenfassung	259
9	Ereignisbehandlung unter jQuery	261
9.1	Grundlagen zu Ereignissen, Eventhandlern, Triggern und Datenbindung ..	262
9.1.1	Ereignisse	262
9.1.2	Allgemeines zu Eventhandlern	262
9.1.3	HTML-Eventhandler	263
9.1.4	JavaScript/DOM-Eventhandler	263
9.1.5	Das Ereignisobjekt	264
9.1.6	Blubbern und die Bubble-Phase	265
9.1.7	Datenbindung	267
9.1.8	Trigger	267
9.1.9	Delegation	268
9.1.10	Versprechen (Promises)	268
9.2	Das Ereignisobjekt in jQuery	269
9.2.1	Der Konstruktor von jQuery.Event	269
9.2.2	Die Eigenschaften des Ereignisobjekts jQuery.Event	270
9.2.3	Die Methoden eines Objekts vom Typ jQuery.Event	274
9.2.4	Die Ausführung des bereitstehenden Ereignisses anhalten	277
9.3	Ich habe fertig – \$(document).ready()	278
9.4	Event-Helper	278
9.5	Erweiterte Methoden für das Eventhandling	283
9.5.1	Datenbindung	283
9.5.2	Triggern	288
9.6	Live Events	292
9.6.1	Die veraltete Methode live()	292
9.6.2	Die veraltete Methode delegate() und die delegate-Variante von on() ..	293
9.6.3	Die Methoden die() und undelegate()	297
9.7	jQuery.proxy()	297
9.8	Weiterentwicklung der Datumskomponente	300
9.9	Zusammenfassung	302

10	Effekte und Animationen	303
10.1	Grundsätzliche Anwendung	303
10.1.1	Speed is all you need	303
10.1.2	Die Angabe eines Callback	305
10.1.3	Verkettung	305
10.1.4	Warteschlangen	306
10.1.5	Beendigung mit stop(), finish() und jQuery.fx.off	306
10.1.6	Endlosanimationen	307
10.1.7	Art der Animationen – Easing	307
10.2	Konkrete Effekte mit Standardmethoden	309
10.2.1	Anzeigen und Wegblenden – die Methoden show() und hide()	309
10.2.2	Gleiteffekte – slideDown(), slideUp() und slideToggle()	309
10.2.3	Transparenzeffekte – fadeIn(), fadeOut() und fadeTo() sowie toggle()	312
10.3	Individuelle Animationen mit animate()	316
10.4	Zusammenfassung	324
11	Asynchrone Programmierung	325
11.1	Ajax und XMLHttpRequest (XHR) – Grundlagen	326
11.1.1	Ein XMLHttpRequest-Objekt manuell erzeugen	326
11.1.2	Die Methoden eines XHR-Objekts	327
11.1.3	Die Eigenschaften eines XHR-Objekts	328
11.1.4	Das Datenformat bei einer Ajax-Kommunikation	328
11.1.5	Exemplarischer Ablauf einer Ajax-Anfrage	329
11.2	Spezialitäten bei der Ajax-Unterstützung in jQuery	330
11.2.1	JSONP und Remote Requests	330
11.2.2	Das jqXHR-Objekt	331
11.2.3	Grundsätzliches zu Methoden in jQuery für Ajax-Anfragen	331
11.2.4	Vermeidung von Caching	333
11.3	Anwendung der Standardmethoden für Ajax	333
11.3.1	\$.get() und \$.post()	333
11.3.2	JSON-Daten anfordern und verarbeiten – getJSON() und parseJSON()	342
11.4	Ein Skript per Ajax nachladen – jQuery.getScript()	344
11.5	Die allgemeine Variante zum Laden von Daten – load()	346
11.5.1	Filter angeben	347
11.6	Serialisieren von Daten	351
11.6.1	Die Methode serialize()	351
11.6.2	Die Methode serializeArray()	353
11.6.3	Die allgemeine Version zum Serialisieren – \$.param()	353
11.7	Vorgabewerte für Ajax – jQuery.ajaxSetup()	354

11.8	Ajax Events und Ajax-Eventhandler	354
11.8.1	Lokale Ereignisse	354
11.8.2	Globale Ereignisse	356
11.9	Die vollständige Kontrolle	357
11.9.1	jQuery.ajax()	357
11.9.2	Erweiterte Techniken für \$.ajax()	365
11.10	Web Worker	367
11.10.1	Was ist ein Web Worker?	367
11.10.2	Erzeugen von Web Workern	368
11.10.3	Die Kommunikation mit einem Web Worker	369
11.10.4	Einen Worker mit terminate() beenden	369
11.10.5	Ein Beispiel zu einem klassischen Web Worker	370
11.11	Deferred Object und Promises	372
11.11.1	Das Umfeld – Promises	373
11.11.2	Die speziellen jQuery-Methoden zum Umgang mit einem Deferred Object	376
11.11.3	Ein konkretes Beispiel mit Deferred Objects	378
11.12	Ajax mit Deferred Objects	379
11.12.1	Generische Ajax-Methoden mit then() verketteten	380
11.12.2	Das Laden von Skripten mit Deferred Objects	380
11.12.3	JSONP und Deferred Objects	381
11.12.4	Mehrere Ajax-Anfragen ausführen und synchronisieren	382
11.13	Das Callbacks Object	383
11.14	Zusammenfassung	387
12	jQuery UI	389
12.1	Was versteht man unter jQuery UI?	390
12.1.1	Komponenten zur Unterstützung der Interaktion	390
12.1.2	Widgets	390
12.1.3	Erweiterte Effekte	391
12.1.4	Das Themen-Framework samt ThemeRoller	391
12.2	Der Einstieg in jQuery UI	392
12.3	Wie nutzt man jQuery UI grundsätzlich?	393
12.3.1	Download und der ThemeRoller	393
12.3.2	Die Bereitstellung und Einbindung lokaler Ressourcen	397
12.3.3	Einbindung über ein CDN	398
12.4	Verwenden der Komponenten in jQuery UI	399
12.4.1	Die Defaulteinstellung	400
12.4.2	Einige grundsätzliche Regeln zu Komponenten und Widgets	403
12.4.3	Eigenschaften/Optionen von Komponenten	403
12.4.4	Methoden von Komponenten	406
12.4.5	Events bei Komponenten und Widgets	410

12.5	Ein Überblick über die Komponenten und Widgets	410
12.5.1	Die Interaktionskomponenten	410
12.5.2	Die Widgets	411
12.6	Effekte	423
12.6.1	Die Methode effekt()	423
12.6.2	Farbanimationen mit animate()	423
12.7	Zusammenfassung	424
13	jQuery Mobile	425
13.1	Grundlagen	426
13.1.1	Die Plattformen	427
13.1.2	Widgets und Features	429
13.1.3	Download und Bereitstellung	429
13.1.4	Themes und der ThemeRoller	431
13.2	Das Rollensystem und data-role	432
13.3	Der grundsätzliche Aufbau einer mobilen Seite	432
13.3.1	Ein erstes vollständiges Beispiel	433
13.4	Verknüpfen von Seiten	434
13.4.1	Externe Links mit Hijax	435
13.4.2	Interne Links und das spezielle Verständnis einer Seite	435
13.4.3	Zurück in der Historie	436
13.5	Die Übergänge	438
13.6	Dialoge	439
13.7	Schaltflächen	440
13.7.1	Schaltflächen mit Icons	440
13.7.2	Blockelement oder Inline-Element	441
13.7.3	Gruppierung	442
13.8	Toolbars und Navigationsbars	443
13.9	Listen	445
13.10	Formularelemente	446
13.10.1	Feldcontainer	447
13.10.2	Die verschiedenen Formularelemente	447
13.10.3	Deaktivieren von Elementen	448
13.10.4	Plugin-Methoden bei Formularelementen	448
13.10.5	Abschicken der Formulardaten	448
13.11	Spezielle Ereignisse	448
13.11.1	Berührungseignisse	449
13.11.2	Änderung der Orientierung	449
13.11.3	Verschiebeeignisse	449
13.11.4	Seitenereignisse	450
13.11.5	Ein Beispiel mit Reaktionen auf Events	450
13.12	Kollabierte und expandierte Inhalte	452
13.13	Zusammenfassung	454

14	Plugins	455
14.1	Die Plugin-Seiten von jQuery	455
14.2	Ein vorhandenes Plugin suchen und verwenden	458
14.3	Eigene Plugins erstellen	465
14.3.1	Warum eigene Plugins erstellen?	465
14.3.2	Erstellen eines ersten Plugins	466
14.3.3	Die wesentlichen Regeln zur Erstellung eines einfachen Plugins	468
14.3.4	Regeln zur Erstellung komplexerer Plugins	469
14.3.5	Ein Beispiel für ein Plugin mit Optionen	470
14.3.6	Ein weiteres Beispiel für ein Plugin mit Optionen	472
14.3.7	Ein Plugin veröffentlichen	473
14.4	Zusammenfassung	477
15	Das Habitat rund um jQuery	479
15.1	Sizzle	479
15.2	QUnit	482
15.2.1	xUnit-Testing	482
15.3	Bootstrap	488
15.3.1	Responsive Design	488
15.3.2	Bootstrap zur Umsetzung von RWD	489
15.3.3	Herunterladen von Bootstrap	490
15.3.4	Eine Basis-Vorlage	490
15.3.5	Ein Kontaktbeispiel	491
15.4	Zusammenfassung	493
16	Anhang	495
16.1	Grundlagen zu JavaScript	495
16.1.1	Case-Sensitivität	495
16.1.2	Variablen, Literale und Datentypen	495
16.1.3	Funktionen und Methoden	497
16.1.4	Objekte in JavaScript	499
16.2	Die Webseite zum Buch	501
	Index	503

Vorwort

Das World Wide Web ist schon viele Jahre unglaublich erfolgreich und hat sich gerade die letzten Jahre bezüglich seiner zukünftigen technischen Basis als auch der Form der Darstellung von Inhalten sowie der Interaktion mit dem Anwender und der Dynamik stark weiterentwickelt. Das Stichwort *Dynamik* deutet schon an, dass statische, passive Webseiten überholt sind und immer seltener werden. Bereits die Gegenwart und insbesondere die Zukunft gehören sogenannten Rich Internet Applications (RIAs). Gerade aber diese interaktiven Seiten und Applikationen im Web sind ohne geeignete Frameworks kaum noch effektiv zu erstellen und zu warten, nicht zuletzt auch deswegen, weil viele optische sowie funktionale Features wie animierte Inhaltsaufbereitung oder komfortable Benutzereingabemöglichkeiten mittlerweile verbreitet sind und vom verwöhnten Anwender ebenfalls erwartet werden. Dementsprechend wird jedoch der Aufwand zur Erstellung von solchen Webangeboten immer größer.

Nun war es einige Jahre nicht wirklich deutlich zu erkennen, in welche Richtung sich das World Wide Web zur Umsetzung solcher anspruchsvoller Applikationen wirklich entwickelt. Es gab längere Zeit verschiedene technische Ansätze, die als verschiedene Optionen für die Zukunft des Webs offen waren. Aber wenn Sie aktuell die populären und halbwegs modern gemachten interaktiven Angebote im World Wide Web betrachten, werden Sie eigentlich nur noch konservativ programmierte Applikationen auf Basis von Ajax (Asynchronous JavaScript and XML) sowie klassisches DHTML (Dynamic HTML) vorfinden. Ein paar Seiten setzen vielleicht noch auf das veraltete, proprietäre Flash, aber die verschwinden mehr und mehr. Daneben gab es aber über eine geraume Zeit Versuche, neuere proprietäre Techniken wie JavaFX, Silverlight oder AIR/Flex einzusetzen und teils sogar vollkommen auf HTML, CSS und JavaScript zu verzichten. Aber die Aktivitäten der Hersteller in Hinsicht auf die Weiterentwicklung mit proprietären Ansätzen ist mittlerweile fast vollkommen zum Erliegen gekommen.

Dementsprechend setzen aktuell für interaktive anspruchsvolle Webapplikationen die meisten Firmen, Organisationen sowie auch Privatanwender weiter ganz konservativ auf dynamisches HTML und Ajax, zumal sich mit HTML 5 und CSS 3 offene Standards etablieren, die zudem auch von den Anbietern der proprietären Techniken ganz offiziell unterstützt werden. Und nicht zuletzt setzen die großen – in der Hinsicht unabhängigen – Hersteller wie Google oder Apple ebenso explizit auf HTML 5 und CSS 3 für die Zukunft.

Ihnen sollte nun etwas aufstoßen, dass ich im Zusammenhang mit Ajax und HTML5/CCS3 von **konservativ** spreche. Es ist noch nicht ganz so lange her, da war Ajax das Buzzword schlechthin im World Wide Web. Ajax ist die programmiertechnische Basis dessen, was um das Jahr 2005/2006 als Web 2.0 in aller Munde war. Immerhin hat erst Ajax es möglich gemacht, bei Bedarf nur die Inhalte einer Webseite auszutauschen, die tatsächlich neu vom Webserver angefordert werden müssen. Die bereits geladene Webseite bleibt bei einer Datennachforderung per Ajax im Browser vorhanden und mittels DHTML wird gezielt an einer bestimmten Stelle in der Webseite ein Austausch bestehenden Inhalts durch die neu nachgeladene Information vorgenommen. Dabei kann die nachgeladene Information entweder aus Klartext (inklusive HTML-Fragmenten) oder aus strukturiertem XML oder JSON (JavaScript Object Notation) bestehen. Die Vorteile dieser Vorgehensweise sind bei stark interaktiven Applikationen mit häufigem Serverkontakt sofort offensichtlich und mittlerweile voll etabliert.

Dennoch bedeutet die Verwendung von Ajax respektive DHTML keinen Einsatz von modernen Webtechniken, denn die Grundlagen dieser damit zusammengefassten Technologien gibt es alle bereits seit 1997, was meine Bezeichnung als **konservativ** verdeutlicht. Und dass sich Ajax erst fast zehn Jahre später wie eine Explosion über das World Wide Web verbreitet hat, zeigt aus meiner Sicht ganz deutlich, dass das Internet und das World Wide Web in der Entwicklung recht träge, konservativ und langsam sind. Diese von mir provokant formulierte These soll nun das Internet und das World Wide Web nicht diskreditieren! Es ist nur so, dass sich im Internet Technologien nur sehr langsam durchsetzen können, weil sich alle Beteiligten an diesem komplexen, sehr sensiblen Gebilde World Wide Web auf eine gemeinsame Basis einigen müssen. Und das dauert eben! In der Regel viele Jahre. Und auf Dauer setzt sich scheinbar im Web nur das durch, was gut und einfach ist.

Wenn Sie nun aber eine moderne Webapplikation auf Basis von Ajax und DHTML erstellen wollen, ist eine Programmierung von Hand wie erwähnt sehr aufwendig und fehlerträchtig. Zwar ist das grundsätzliche Erstellen von DHTML- bzw. Ajax-Applikationen nicht sonderlich schwierig, wenn man die Grundlagentechniken HTML bzw. XHTML, CSS und JavaScript beherrscht. Das Zusammenspiel dieser – einzeln gesehen – in der Tat recht einfachen Webtechnologien im Client kann jedoch äußerst diffizil sein, was nicht zuletzt ein Resultat der Browserkriege des letzten Jahrtausends ist. Dazu kommen im Fall von Ajax oft noch der permanente Austausch von Daten zwischen Client und Webserver sowie die sehr feingliedrige Verteilung von Geschäftslogik zwischen Client und Server hinzu.

Zudem erzwingt die eingeschränkte Leistungsfähigkeit von JavaScript oft eine nicht ganz triviale Programmierung von Strukturen, die in leistungsfähigeren (insbesondere objekt-orientierten) Programmieretechniken auf dem Silbertablett serviert werden. So gesehen ist die Erstellung einer interaktiven Applikation für das Web heutzutage durchaus mit der Komplexität der Erstellung einer Desktop-Applikation bzw. einer verteilten Netzwerkapplikation zu vergleichen, wenn sie sich an den aktuellen Ansprüchen der Konkurrenz messen will. Das erkenne ich auch daran, dass in meinen Schulungen zu JavaScript, Ajax oder CSS mehr und mehr Programmierer sitzen, die aus mächtigen Sprachen wie Java oder C# kommen (früher war das Vorwissen im Programmierumfeld eher geringer). Für mich ist das ein deutliches Zeichen, dass die Ansprüche an eine moderne RIA steigen und von reinen Designern nicht mehr erfüllt werden können. Anders ausgedrückt – mit modernen RIAs ist das Web endgültig den Kinderschuhen entwachsen.

Nicht zuletzt bringt die Erstellung von modernen Webseiten und insbesondere Ajax-RIAs ein hohes Maß an Tests und Anpassung an verschiedene Webbrowser und Plattformen mit sich. An den unterschiedlichsten Stellen warten tückische Fallstricke. Natürlich ist die manuelle Erstellung von komplexen DHTML-Aktionen wie Drag & Drop oder animierten Menüs nicht jedermanns Sache. Ihnen sind sicher ebenfalls die oft extrem diffizilen Abhängigkeiten von den verschiedenen Browsern, Browserversionen und Betriebssystemplattformen bekannt. Diese Probleme nehmen zwar in modernen Browsern ab, sind aber immer noch vorhanden und zudem nutzen gerade viele Firmen auch noch alte Browser.

Sogenannte Frameworks und Toolkits für Ajax bzw. JavaScript versprechen nun für viele Aufgabenstellungen und Probleme im Umfeld von modernen Webseiten Abhilfe. Sie stellen vielfach vor allem JavaScript-Funktionsbibliotheken mit getesteten und hochfunktionellen Lösungen sowie ausgereifte Style Sheets bereit, damit Sie nicht jedes Mal das Rad neu erfinden und dessen einwandfreie Funktionalität umfangreich testen müssen. Dazu gibt es gelegentlich auch spezielle Tools und Programme, die eine Arbeit mit diesen Bibliotheken unterstützen oder gar erst möglich machen. Auch bringen einige mächtige HTML-Editoren mittlerweile sogar eigene Frameworks mit.

Wir werden uns in diesem Buch nun – wie der Titel unzweifelhaft aussagt – jQuery widmen und schauen, wie Sie dieses geniale Framework einsetzen können, um Ihre Webapplikationen zu verbessern bzw. die Erstellung zu vereinfachen oder bestimmte Features gar erst möglich zu machen. Wenn Sie die Möglichkeiten von jQuery nicht schon kennen, lassen Sie sich positiv überraschen, wie einfach Ihnen dieses mächtige Werkzeug Webseiten ermöglicht, die alle moderne Effekte und Features enthalten.

Zu diesem Einstieg in jQuery wünsche ich Ihnen viel Spaß und viel Erfolg. Doch vorher möchte ich ein paar abschließende Bemerkungen zu meiner Person machen. Meinen Namen werden Sie auf dem Buchumschlag oder am Ende des Vorworts gelesen haben – Ralph Steyer. Ich habe in Frankfurt/Main an der Goethe-Universität Mathematik studiert (Diplom) und danach anfangs einen recht typischen Werdegang für Mathematiker genommen – ich bin erst einmal bei einer großen Versicherung gelandet, aber schon da mit EDV-Schwerpunkt. Zunächst arbeitete ich einige Jahre als Programmierer mit Turbo Pascal und später mit C und C++. Nach vier Jahren wechselte ich in die fachliche Konzeption für eine Großrechnerdatenbank unter MVS. Die Erfahrung war für meinen Schritt in die Selbstständigkeit sehr motivationsfördernd, denn mir wurde klar, dass ich das nicht auf Dauer machen wollte. Seit 1996 verdiene ich daher meinen Lebensunterhalt als Freelancer, wobei ich fliegend zwischen der Arbeit als Fachautor, Fachjournalist, EDV-Dozent, Consultant und Programmierer wechsele. Daneben referiere ich gelegentlich auf Webkongressen, unterrichte an verschiedenen Akademien und Fachhochschulen, übersetze gelegentlich Fachbücher oder nehme Videotraining auf. Das macht aus meiner Sicht einen guten Mix aus, bewahrt vor beruflicher Langeweile und hält mich sowohl in der Praxis als auch am Puls der Entwicklung. Insbesondere habe ich das Vergnügen und gleichzeitig die Last, mich permanent über neue Entwicklungen auf dem Laufenden zu halten, denn die Halbwertszeit von Computerwissen ist ziemlich kurz. Dementsprechend ist mein Job zwar anstrengend, aber vor allem immer wieder spannend. Doch nun ab in die Welt von jQuery!

Ralph Steyer

Frühjahr 2018

1

Einleitung

In diesem einleitenden Kapitel beschäftigen wir uns mit dem Umfeld von jQuery im Spiel- und Webprogrammierung im Allgemeinen. Sie erfahren, was alles im Buch behandelt wird und was Sie bei der Arbeit mit dem Buch beachten sollten. Dazu betrachten wir vor allen Dingen auch wichtige Voraussetzungen, die Sie für die Programmierung mit jQuery schaffen sollten oder die Ihnen die Arbeit erleichtern.

■ 1.1 Das Umfeld

RIAs mit ihren – etwas schwammig formulierten – *reichhaltigen* Möglichkeiten verändern seit ein paar Jahren die Art der Nutzung des WWW. Ebenso wird die Bedeutung von klassischen Desktop-Applikationen neu positioniert. Viele früher nur als Desktop-Applikation genutzte Programmtypen finden sich plötzlich im Web und werden mit dem Browser ausgeführt, seien es persönliche Kalender, vollständige Office-Programme, Spiele, Routenplaner, komplett integrierte Entwicklungsumgebungen oder Kommunikationsprogramme.

Aber auch mobile Webseiten oder Anwendungen für Handys oder Smartphones basieren mehr und mehr auf Webtechnologie. Dies verändert nicht zuletzt das Anwenderverhalten und die Anwendererwartung bei Internetapplikationen im Allgemeinen sowie der Verfügbarkeit von Leistungen. RIAs stehen auf der einen Seite als klassische Webapplikationen (aber mit einem gewissen Mehrwert) immer zur Verfügung, wenn man einen halbwegs schnellen Internetzugang und einen modernen Browser hat. Und sie sind auf der anderen Seite von der Bedienung, der Performance sowie der Optik mittlerweile fast gar nicht von klassischen Desktop- oder Mobilapplikationen zu unterscheiden.

■ 1.2 Frameworks und Toolkits

Um diese reichhaltigen Möglichkeiten von RIAs zu gewährleisten, stellt die Verwendung eines geeigneten Web-Frameworks oder -Toolkits meist die effektivste, wenn nicht einzig sinnvolle Lösung dar. Allerdings sollte Ihnen bewusst sein, dass Sie sich beim Einsatz eines Frameworks oder Toolkits in erhebliche Abhängigkeit von einem Hersteller respektive einem Projekt begeben und die vollständige Kontrolle über den Quellcode Ihrer Applikationen verlieren. In jedem Fall erfordert der Einsatz von Frameworks/Toolkits die Einarbeitung in die jeweiligen Funktionsbibliotheken und Arbeitsweisen dieses Systems. Und im Gegensatz zu vollmundigen Werbeversprechen mancher Frameworks (sowie Tools) können Sie diese meist nur dann wirklich effektiv nutzen, wenn Sie die Idee und Arbeitsweise von Webprogrammierung verstanden haben und alle Grundtechnologien zumindest im Ansatz beherrschen. Sie profitieren am meisten von Frameworks/Toolkits, je weniger Sie diese eigentlich benötigen und je fitter Sie in den Grundlagen sind.



Hintergrundinformation

Nun war bereits sowohl von Frameworks als auch von Toolkits die Rede. Was genau ein **Framework** ist und wie es sich von einem **Toolkit** unterscheidet, ist nicht standardisiert. Sowohl eine verbindliche Definition als auch eine Abgrenzung sind gar nicht so einfach. Allgemein versteht man jedoch unter einem Framework ein Programmiergerüst, das bestimmte Funktionalitäten bereitstellt. Ein Framework ist selbst jedoch kein fertiges Programm, sondern stellt den Rahmen (Frame) zur Verfügung, innerhalb dessen ein oder mehrere Programmierer eine Anwendung erstellen. Ein Framework beinhaltet in der Regel eine Bibliothek mit nützlichen vorgegebenen Codestrukturen (meist auf Basis einer spezifischen Sprache wie JavaScript), legt aber – im Gegensatz zu einer reinen Bibliothek – auch eine Steuerung von Verhaltensweisen bei der Verwendung fest (also zum Beispiel eine Syntax beziehungsweise Grammatik).

Bei einem Toolkit steht dagegen mehr die Sammlung an Programmen (Tools) im Fokus, die aber durchaus auch auf spezifischen Bibliotheken und einem Syntaxkonzept aufsetzen können. Sowohl ein Framework, aber vor allem Toolkits stellen vielfach sogenannte **Widgets** beziehungsweise **Komponenten** zur Verfügung. Dabei handelt es sich in der Regel um Elemente, aus denen eine grafische Benutzerschnittstelle (UI – User Interface oder GUI – Graphical User Interface) zusammengesetzt wird. Wir werden im Buch den Begriff Framework als Vereinheitlichung verwenden, wenn nicht besondere Gründe dagegen sprechen.

■ 1.3 Was behandeln wir in diesem Buch?

Das vorliegende Buch soll Ihnen den Einstieg in die Webprogrammierung mit jQuery und den darauf aufbauenden Erweiterungen jQuery UI sowie jQuery Mobile und ganz kompakt auch noch Sizzle, QUnit und Bootstrap ermöglichen. Der Kern von jQuery ist ein freies, umfangreiches JavaScript-Framework, das im Fokus des Buchs steht und ursprünglich von einem gewissen John Resig entwickelt und erstmals bereits im Januar 2006 auf dem Bar-Camp (NYC) in New York veröffentlicht wurde. Mittlerweile wird es als OpenSource-Projekt der jQuery Foundation (<https://jquery.org/>) weiterentwickelt und ist eines der populärsten Web-Frameworks überhaupt.



Bild 1.1 Die Webseite der jQuery Foundation

Das liegt sicher auch daran, dass jQuery von zahlreichen großen Organisationen und Firmen¹ in unterschiedlichem Rahmen, aber ganz offiziell unterstützt wird. Zahlreiche integrierte Entwicklungsumgebungen beziehungsweise Webentwicklung-Tools stellen mittlerweile jQuery direkt bereit oder unterstützen die spezielle Syntax von jQuery.

¹ Etwa Microsoft, Wikipedia, WordPress, Adobe, Intel, Mozilla, GitHub, ...

1.3.1 Das Kern-Framework jQuery

Das Kern-Framework selbst stellt eine ganze Menge komfortabler Funktionalitäten zur Verfügung, zum Beispiel zur DOM-Manipulation und -Navigation sowie zum grundlegenden Ajax-Support. Darüber hinaus bietet jQuery Unterstützung von CSS, ein erweitertes Event-System, eindrucksvolle Effekte und Animationen, diverse Hilfsfunktionen und zahlreiche Plugins.

Sie lernen nun in diesem Buch, wie Sie jQuery für Ihre eigenen Webapplikationen nutzen können – von einfachen Webseiten, die nur um einzelne Effekte aufgewertet werden sollen bis hin zu komplexen RIAs. Dabei haben Sie auf der einen Seite ein Einsteigerbuch vorliegen, in dem wir zwar nicht ganz bei null beginnen, aber auch nicht Programmierprofis als Leser voraussetzen werden. Insbesondere soll keine große Erfahrung mit einem Framework vorausgesetzt werden. Auf der anderen Seite sollte ein Leser jedoch schon Erfahrung in Webtechnologien haben, was ich gleich bei der Definition der Zielgruppe noch etwas genauer ausführen möchte.

Das Buch selbst ist in den einzelnen Kapiteln im Wesentlichen gleich aufgebaut. Einer kurzen Einleitung folgen die detaillierten Themen und das Kapitel wird mit einer Zusammenfassung abgeschlossen.

Konkret gehen wir im Laufe des Buchs so vor: Nach diesem Kapitel, in dem bereits alle Voraussetzungen für die Arbeit mit jQuery geschaffen werden, springen wir ins kalte Wasser und spielen ohne große Vorbereitung einige Beispiele durch. Sie sollen ein Gefühl für das bekommen, was man mit jQuery machen kann.

Danach schaffen wir etwas Grundlagenwissen zum Web, JavaScript, Ajax, XML, JSON etc. Darauf aufbauend soll die grundsätzliche Arbeit von jQuery genauer betrachtet werden. Richtig intensiv werden wir uns anschließend Selektoren und Filtern widmen. Diese Möglichkeiten zur Auswahl von Objekten im Kontext einer Webseite sind meines Erachtens eines der größten Highlights des Frameworks überhaupt und bilden die Basis für den Zugriff auf die Elemente einer Webseite. Diesem widmen wir uns in der Folge ebenfalls mit vielen Beispielen. Im Anschluss behandeln wir das dynamische Beeinflussen von CSS-Eigenschaften, Ereignisbehandlung, Effekte und Animationen bis hin zum Web 2.0 und Ajax.

1.3.2 Plugins, UI, Mobile & Co

Das war es zu jQuery eigentlich. Aber da war doch noch was. Beispielsweise jQuery UI! Das wurde bisher stiefmütterlich behandelt. Sie könnten nun meinen, dass das so etwas wie das hässliche Entlein im jQuery-Universum ist. Oder uninteressant. Weit gefehlt. jQuery UI ist der stolze Schwan. Rein optisch macht jQuery UI viel mehr her als jQuery selbst, obwohl die Anwendung viel einfacher ist – wenn man jQuery vorher verstanden hat. jQuery ist die Basis, die eher im Bereich des Quellcodes und bei der Programmierung Erleichterungen verschafft, während jQuery UI darauf als eigenständiges Framework aufbaut und mit optisch ausgereiften Oberflächenkomponenten und einem CSS-Theme-Framework beeindruckt. Wir werden uns in dem Buch natürlich auch jQuery UI intensiv ansehen und Beispiele mit den verschiedenen Widgets von jQuery UI durchspielen. Aber auch die Anpassung über

Optionen, Events, Methods und Theming soll erklärt werden. Das Theme-Framework und der ThemeRoller von jQuery UI werden ebenso Gegenstand unserer Betrachtungen sein.

Und dann gibt es bei jQuery noch sogenannte Plugins als Erweiterung des Frameworks. Sie sehen in dem Buch sowohl, wie Sie fremde Plugins verwenden können, wenn Sie eine bestimmte Funktionalität in den Kernbibliotheken von jQuery, jQuery UI etc. nicht finden, als auch wie Sie eigene Plugins erstellen.

Ebenfalls wollen wir uns der Erstellung von mobilen Webseiten auf Basis von jQuery widmen. Dazu verwenden wir das Mobile Framework, das wie jQuery UI auf nativem jQuery unmittelbar aufsetzt. Mit Bootstrap werden wir zudem noch ganz kompakt eine weitere Erweiterung von jQuery als Alternative anreißen, die zwar nicht zum offiziellen Projekt-Habitat zählt, aber wegen der universellen Verwendbarkeit extrem beliebt ist und sowohl zur Erstellung von mobilen als auch „normalen“ Webseiten verwendet werden kann. Ein kurzer Blick auf die Projekte Sizzle und QUnit runden das Thema ab.



Praxistipp

Im Buch arbeiten wir zur Verdeutlichung natürlich immer wieder mit Codebeispielen. Ich empfehle ausdrücklich, dass Sie die Codebeispiele vollständig selbst abtippen (und natürlich auch bei Bedarf modifizieren und damit experimentieren). Aber Sie finden die Listings auch auf der Webseite zum Buch.

1.3.3 Wen sehe ich als Zielgruppe für das Buch?

Es ist immer schwer vorherzusehen, wer sich für ein bestimmtes Buchthema interessieren könnte. Aber ich habe gewisse Vorstellungen und mittlerweile recht umfangreiche Erfahrungen aus jQuery-Schulungen. Von daher kann ich erahnen, wer sich warum mit der Erstellung von RIAs respektive jQuery beschäftigen wird und dieses Buch lesen könnte. Ich gehe davon aus, dass Sie bereits HTML kennen und Webseiten erstellt sowie schon in irgendeiner Form programmiert haben (zumindest ein bisschen). JavaScript wäre eine gute Grundlage, aber auch andere Programmiertechniken sind als Basis willkommen, wobei Sie sich dann parallel tiefer in JavaScript einarbeiten sollten. Style Sheets sollte für Sie ebenfalls kein unbekannter Begriff sein. Haben Sie weder mit Webseitenerstellung beziehungsweise HTML noch Programmierung Erfahrung, wird das Buch für Sie wahrscheinlich eine Herausforderung (was Sie aber nicht entmutigen soll). Ich vermute weiter, Sie sind den Einschränkungen einer statischen HTML-Seite überdrüssig. Vielleicht haben Sie auch schon Erfahrungen mit dynamischen Webseiten. Als Anwender ganz sicher, nehme ich an. Und ich vermute, Sie wollen auf einfache und komfortable Weise solche interaktiven modernen Seiten erstellen. Und jQuery ist ein super Mittel dazu.

Nun habe ich schon angedeutet, dass immer mehr Programmierer von mächtigen Techniken und Umfeldern wie Java oder .NET in den Bereich der Webprogrammierung drängen. Dementsprechend möchte ich auch Leser mit diesem Hintergrundwissen berücksichtigen.

■ 1.4 Schreibkonventionen

In diesem Buch werden verschiedene Schreibkonventionen eingehalten, die Ihnen helfen sollen, die Übersicht zu bewahren. Wichtige Begriffe werden *hervorgehoben*. Vor allem sollten Sie erkennen können, ob es sich um normalen Text oder Programmcode handelt. Wenn bestimmte Codepassagen mir besonders wichtig erscheinen, werde ich sie wie gesagt **hervorheben**. Und ebenso werden Sie in dem Buch Bereiche vorfinden, die über die Markierung mit verschiedenen Symbolen besondere Aufmerksamkeit erzeugen sollen.

1.4.1 Listings

Eine besondere Bemerkung soll noch zu Quelltextpassagen von vollständigen Listings erfolgen. In seltenen Fällen kann es vorkommen, dass es aus drucktechnischen Gründen notwendig ist, eine Quelltextzeile auf mehrere Buchzeilen zu verteilen, die eigentlich nicht in mehrere Zeilen gehört. In diesem Fall können Sie diese Passagen im Editor noch in eine Zeile notieren, wobei man in der Regel aber auch mit Zeilenumbrüchen arbeiten kann. Nur nicht bei längeren Strings (Texten in Hochkommata), die nicht über mehrere Zeilen verteilt werden dürfen!

■ 1.5 Was benötigen Sie beziehungsweise was nutzt Ihnen?

Beschäftigen wir uns nun mit dem, was Sie für den Umgang mit dem Buch und der Arbeit mit jQuery als Voraussetzungen haben sollten und was Ihnen zusätzlich an verschiedenen Stellen der Webseitenerstellung sowie Programmierung im Web – mit oder ohne jQuery – allgemein nützlich sein kann.

1.5.1 Hardware und Betriebssystem

Wir bewegen uns beim Umgang mit jQuery im Internet. Sie benötigen natürlich einen Computer mit Internetzugang. Zwar werden an den Computer keine besonderen Anforderungen gestellt, aber es wäre nicht schlecht, wenn Ihre Hardware halbwegs auf einem vernünftigen Stand ist, wobei bereits die Anforderungen moderner Betriebssysteme die notwendige Hardware nach unten festlegen. Empfehlenswerte Betriebssysteme sind alle grafischen Systeme wie Linux, Windows oder MacOS, die halbwegs auf dem Stand der Zeit sind. Das spielt – wie meist im Web – für uns keine Rolle.

1.5.2 jQuery, jQuery Mobile, jQuery UI & mehr

Sie benötigen natürlich für das Nachvollziehen der Ausführungen jQuery selbst sowie – im späteren Teil des Buchs – jQuery UI, Sizzle, QUnit, Bootstrap und jQuery Mobile.

1.5.2.1 Download des Kern-Frameworks jQuery

Den Download der jeweils aktuellsten Versionen als auch ältere Versionen finden Sie über die Webseite <http://jquery.com/>.

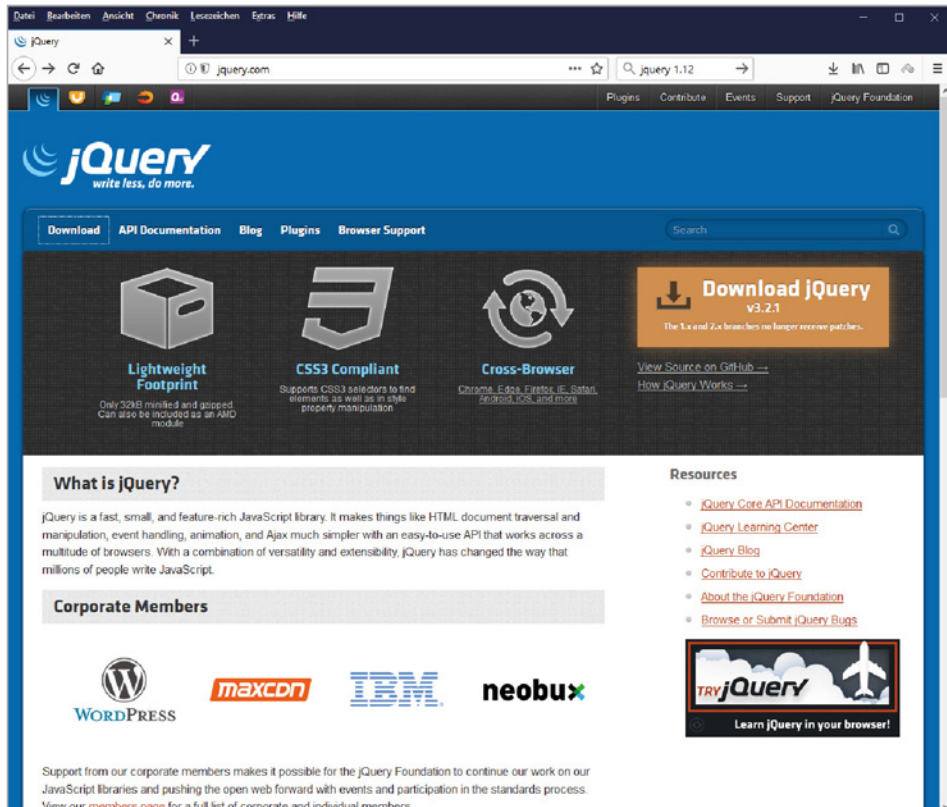


Bild 1.2 Hier gibt es das jQuery-Framework.

Die jeweils neueste Standardversion können Sie bereits von der Einstiegsseite in das Projekt laden. Da gibt es einen Button, der direkt zum Download führt. Aber es existieren verschiedene Varianten der Bibliothek zum Download: im Wesentlichen eine minimierte Version mit der Kennung *min*, die um Kommentare und überflüssige Leerzeichen sowie Zeilenumbrüche reduziert ist und die man in der Produktion einsetzt, und eine unkomprimierte Version, die im Quellcode kommentiert und gut lesbar, aber dementsprechend größer ist. Funktional sind sie aber identisch und enthalten eine JavaScript-Datei, die im Kern

² Das betrifft nicht alte Browser anderer Hersteller – diese werden meist explizit auch ab der 2.x-Schiene unterstützt.

die Bezeichnung *jquery.js* führt. Diese Datei, die in der Regel zusätzlich Versionsnummern im Namen und je nach Typ eine Beschreibung der konkreten Variante aufweisen wird, ist die zentrale Bibliothek des Frameworks, die Sie in Ihre Webseiten einbinden. Sollten Sie eine Variante mit einem ZIP-Archiv aus dem Internet laden, extrahieren Sie es einfach. Sie müssen die JavaScript-Datei zu einer Verwendung bloß entsprechend der üblichen Regeln später in Ihrer Webseite referenzieren (das besprechen wir aber noch genauer).



Hinweis

Zum Zeitpunkt der Bucherstellung gibt es gleich **zwei aktuelle** Versionen von jQuery (1.12.4 und 3.2.1), die (mit Einschränkungen) parallel gepflegt werden. Und dazu gibt es etwas mehr zu sagen.

Die Versionsschiene **1** ist abwärtskompatibel zu alten Versionen des Internet Explorers, während alle Versionen ab der Hauptnummer **2** explizit den Internet Explorer² unterhalb der Version 9 **nicht mehr unterstützen**³. Dementsprechend ist das Framework ab der Version 2 viel schlanker und schneller als die Varianten der Versionsschiene 1, die für alte Internet Explorer noch zahlreiche Workarounds beinhalten, um deren proprietäre Verhaltensweisen zu kompensieren. Dennoch muss man davon ausgehen, dass die Versionsschiene 1 nur noch eingeschränkt weiter gepflegt wird und nicht der gleiche Funktionsumfang wie in der neueren Schiene vorhanden ist. Grundsätzlich sind also die Versionen von jQuery mit mindestens der Hauptnummer 2 moderner, schlanker, schneller und leistungsfähiger, aber wenn man als Webprogrammierer für alte Internet Explorer Seiten bereitstellen muss, sollte man immer die neueste Version aus der 1.x-Schiene verwenden. Beachten Sie natürlich auch, dass dies eine Momentaufnahme zum Zeitpunkt der Bucherstellung ist.

Für Umsteiger von älteren Versionen von jQuery gilt es noch zu beachten, dass alte Codes nicht so ohne Weiteres mit neuen Framework-Versionen laufen. In der Version jQuery 1.9.0 gab es massive Veränderungen bei zahlreichen Methoden des Frameworks. Diese Auswirkungen betreffen damit auch die noch neueren Varianten.

Deshalb wird vom jQuery-Team ein **jQuery-Migrate-Plugin** für einen Umstieg von diesen ganz alten Codes angeboten und dessen Einsatz nachdrücklich empfohlen (<http://github.com/jquery/jquery-migrate/#readme>). Dazu muss gegebenenfalls eine weitere JavaScript-Datei in eine Webseite eingebunden werden, wenn Sie die alten Quellcodes nicht neu programmieren wollen.

Es gibt fernerhin einen **jQuery Upgrade Guide** mit mehr Information (<https://jquery.com/upgrade-guide/3.0/>), wobei da beachtet werden muss, von welcher ursprünglichen Version Sie auf welche Zielversion upgraden wollen. Aber auch dazu finden Sie in dem Guide alle notwendigen Informationen.

³ Besonders bemerkenswert ist, dass Microsoft dessen ungeachtet jQuery massiv unterstützt. Es ist vermutlich auch im Interesse von Microsoft, dass diese für sie kostenintensiven Altlasten irgendwann entsorgt werden. Es wird auch bei einem Umstieg der Anwender auf neuere Versionen vom Internet Explorer für Umsatz gesorgt, denn der Internet Explorer läuft ja ab der Version 9 nicht mehr unter Windows XP. Von Edge ganz zu schweigen.

Die Beispiele im Buch werden die Version 3.2.1 als Referenz verwenden, aber die konkrete Version sollte sich nur manchmal auswirken. Für viele Beispiele und Situationen können Sie jede Version ab 1.9 verwenden, aber für die Verwendung älterer Varianten gibt es kaum sinnvolle Gründe und wir werden Besonderheiten nur hin und wieder erörtern. Bei einigen Projekten, die auf jQuery aufsetzen, kommt hingegen die Version 1.11 zum Einsatz, um eine breitere Browserunterstützung zu gewährleisten.

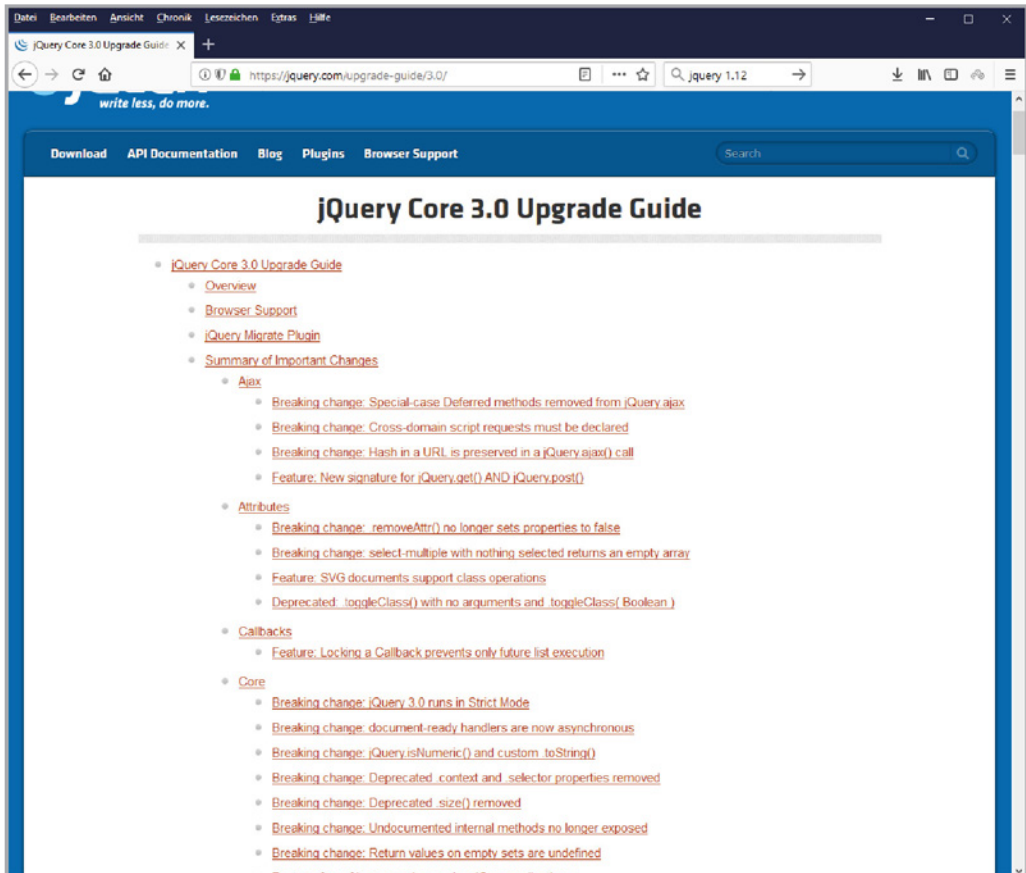


Bild 1.3 Der Upgrade Guide

Installieren über Paketmanager

Obwohl es sehr einfach ist, jQuery über den Browser zu laden und dann in ein Projekt zu integrieren, gibt es mittlerweile ebenfalls die Möglichkeit zur Installation des Frameworks über plattformneutrale **Paketmanager**. Sowohl **npm** als auch **Bower** werden offiziell unterstützt. Das vereinfacht die Sache noch einmal, setzt aber voraus, dass die entsprechenden Paketmanager auch installiert sind und es keine Probleme bei der Abhängigkeit von Ressourcen oder dem Paketmanager etc. gibt. Ich persönlich ziehe das direkte Laden von jQuery vor, da die Installation meines Erachtens keine wesentlichen Vorteile bringt (es handelt sich ja bei jQuery eben um ein Framework und kein Programm) und einige Probleme bei der Installation über Paketmanager bekannt sind. Aber wir wollen das Verfahren dennoch besprechen, zumal Sie gegebenenfalls beim Veröffentlichen von eigenen jQuery-Plugins nicht um den Einsatz von npm herumkommen.

Kommen wir zuerst zu **npm** – dem Node Package Manager (<http://www.npmjs.com>) von **Node.js**. Allgemein ist Node.js (oft auch vollständig klein als `node.js` geschrieben) eine serverseitige Plattform auf Basis von JavaScript zum Betrieb von Netzwerkanwendungen. Die Software steht für Linux, Mac OS, Windows und Solaris zur Verfügung und kann über einen einfachen Assistenten auf einem Computer installiert werden. Node.js wird in der JavaScript-Laufzeitumgebung V8 ausgeführt, die ursprünglich für Google Chrome entwickelt wurde. Die Plattform ist besonders wegen einer ressourcensparenden Architektur beliebt. Und sie ist erweiterbar, was mit einem eigenen Paketmanager erledigt wird – eben npm. Unter <https://nodejs.org/en/> finden Sie node.js zum Download und da ist dieser npm inkludiert, wenn Sie ihn nicht bei der Installation abwählen. Laden Sie einfach die passende Version von node.js für Ihr Betriebssystem und führen Sie den Installationsassistenten aus. Gegebenenfalls können Sie (vor allen Dingen unter Linux) npm auch über das Installationssystem Ihres Betriebssystems installieren⁴. Die Installation ist in der Regel vollkommen unproblematisch und selbsterklärend. Achten Sie nur darauf, dass Sie npm nicht aus Versehen bei der Installation abwählen. Ansonsten können Sie im Anschluss an die Installation von node.js jQuery dann mit npm installieren, denn jQuery ist als Paket im npm-Repository registriert. Sie können die neueste Version von jQuery mit dem folgenden Befehl installieren:

```
npm install jquery
```

Dadurch wird jQuery im Verzeichnis `node_modules` installiert. In `node_modules/jquery/dist/` finden Sie eine unkomprimierte Version, eine komprimierte Version und eine Map-Datei.

jQuery ist aber auch als Paket bei **Bower** (<https://bower.io/>) registriert. Bower ist ein Open-Source-Paketverwaltungstool für clientseitige Webentwicklung, das auf Node.js aufsetzt. Es ermöglicht ebenso das Installieren und Aktualisieren von Programmbibliotheken und Frameworks mithilfe eines in Node.js geschriebenen Kommandozeilentools. Um Bower selbst zu installieren, verwenden Sie npm:

```
npm install -g bower
```

⁴ Etwa mit `sudo apt install npm` unter Debian-Systemen.

```

Microsoft Windows [Version 10.0.16299.19]
(c) 2017 Microsoft Corporation. Alle Rechte vorbehalten.

C:\Windows\System32> npm install -g bower
npm http GET https://registry.npmjs.org/bower
npm http 200 https://registry.npmjs.org/bower
npm WARN deprecated bower@1.8.2: ...psst! Your project can stop working at any moment because its dependencies can change. Prevent this by migrating to Yarn: https://bower.io/blog/2017/how-to-migrate-away-from-bower/
npm http GET https://registry.npmjs.org/bower/-/bower-1.8.2.tgz
npm http 200 https://registry.npmjs.org/bower/-/bower-1.8.2.tgz
C:\Users\ralph\AppData\Roaming\npm\bower -> C:\Users\ralph\AppData\Roaming\npm\node_modules\bower\bin\bower
C:\Windows\System32>

```

Bild 1.5 Bower muss bei Bedarf erst einmal selbst installiert werden.

Sie können dann die neueste Version von jQuery selbst mit dem folgenden Befehl installieren:

```
bower install jquery
```

Dadurch wird jQuery in das Installationsverzeichnis von Bower installiert, wobei *bower_components* voreingestellt ist. Innerhalb von *bower_components/jquery/dist/* finden Sie eine unkomprimierte Version, eine komprimierte Version und eine Map-Datei. Das Paket enthält neben der Standardverteilung weitere Dateien. In den meisten Fällen können Sie diese Dateien ignorieren.



Praxistipp

Wie angedeutet kann es bei der Installation von jQuery über die Paketmanager zu Problemen kommen. In dem Fall laden Sie einfach die jQuery-Datei wie oben beschrieben direkt aus dem Browser. Das genügt in den meisten Fällen.



Hinweis

Der Einsatz von jQuery & Co ist auch über ein **CDN** möglich. Auf den jeweiligen Webseiten der Projekte finden Sie entsprechende Hinweise. CDN steht für **Content Distribution Network** oder auch **Content Delivery Network**. Es bezeichnet ein Netz lokaler und über das Internet verbundener Server, mit dem Inhalte möglichst effektiv ausgeliefert werden sollen. Dazu werden die Daten im Netz gecacht und Anfragen nach Last verteilt. Für jQuery gibt es gleich mehrere CDN-Anbieter, unter denen Sie auswählen können (Details sind auf der Homepage von jQuery zu finden). Der konkrete Einsatz ist einfach, denn im Grunde bedeutet das nur, dass die jeweiligen Ressourcen von fremden Servern eingebunden werden. Etwa wird eine Skriptdatei über `https://code.jquery.com/jquery-3.2.1.min.js` statt eines relativen URL oder eines Verweises auf Ihren eigenen Server eingebunden. Ihre Ressourcen werden entlastet und in der Regel ist die CDN-Hardware samt den Servern im Hintergrund vielfach schneller und leistungsfähiger als Ihre Hardware beziehungsweise Ihr Serversystem. Sie sollten sich aber auch darüber klar sein, dass die Betreiber

eines CDN dann jedes Laden einer Ihrer Webseiten unmittelbar mitbekommen und Sie von der Verfügbarkeit des CDN abhängen. Die Beispiele im Buch verwenden überwiegend das jQuery-Framework von einem CDN, aber das ist natürlich für die Funktionalität irrelevant.

1.5.2.2 UI – das Framework für eine grafische Oberfläche

Wenn Sie nun mit jQuery UI grafische Weboberflächen verwenden wollen, benötigen Sie auch dieses Framework. Denn jQuery UI ist nicht in der normalen jQuery JavaScript Library vorhanden. jQuery UI wird als eigenständiges Projekt im Rahmen des gesamten jQuery-Frameworks geführt und enthält neben JavaScript-Dateien auch andere Ressourcen wie CSS-Dateien und Grafiken. Die Homepage des Projekts finden Sie unter der Adresse <http://jqueryui.com>. Von da aus können Sie dieses Framework über den Download-Link laden. Wenn Sie den Download durchgeführt haben, erhalten Sie in der Regel eine komprimierte ZIP-Datei, die Sie – wie auch die jQuery-Bibliothek – auf Ihrem Server extrahiert zur Verfügung stellen und dann via der zentralen JavaScript-Datei von jQuery UI in Ihre Webseite einbinden. Die Details möchte ich jedoch bis zu den Kapiteln über jQuery UI zurückstellen.

1.5.2.3 Unterstützung für die mobile Welt

Nun gibt es auch noch ein weiteres Framework, das explizit auf dem Kern-Framework aufsetzt und das wir im Buch behandeln werden – jQuery Mobile (<http://jquerymobile.com/>). Auch hier existieren verschiedene Varianten zum Download, unter anderem eine ZIP-Datei, die alle notwendigen Ressourcen enthält. Aber auch zu diesen genauen Details möchte ich erst kommen, wenn wir beim passenden Kapitel angekommen sind.

1.5.2.4 Sizzle und QUnit

Auf der gemeinsamen Webseite aller offiziellen Projekte rund um jQuery finden Sie neben dem Kernframework und jQuery UI sowie jQuery Mobile derzeit zwei weitere Unterprojekte, die im letzten Kapitel kurz behandelt werden sollen.

- Das **Sizzle**-Projekt (<http://sizzlejs.com/>) stellt eine reine JavaScript-CSS-Selektorenengine bereit, um auf einfache Weise in eine Gastgeberbibliothek (wie etwa jQuery) eingegliedert zu werden.
- Unter **QUnit** (<http://qunitjs.com/>) finden Sie ein Unit-Testing-Framework speziell für JavaScript. So etwas setzt man in der sogenannten testgetriebenen Entwicklung ein, die landläufig unter **xUnit** bekannt ist und gerade bei vielen mächtigen Programmiersprachen wie Java oder C++ eingesetzt wird.

1.5.2.5 Bootstrap

Wie angedeutet, wollen wir über das eigentliche Habitat von jQuery noch ein Projekt in dem Buch vorstellen, das aber explizit auf dem jQuery-Kernframework aufsetzt – **Bootstrap** (<https://getbootstrap.com/>). Dieses Framework ist jQuery Mobile sehr ähnlich und arbeitet nach dem Motto „Mobile first“, unterstützt aber ein sogenanntes Responsive Design und ist

auch unmittelbar für Desktop-Anwendungen einzusetzen. Wir werden auch dazu einen kleinen Exkurs machen und dort alles Notwendige zur grundsätzlichen Verwendung betrachten.

■ 1.6 Die Browser

Was Sie für die Tests Ihrer Webseiten, die Sie mit jQuery & Co programmieren, zwingend brauchen werden, sind natürlich Webbrowser, die jQuery unterstützen. Und natürlich müssen Sie bei der Verwendung von jQuery auch einkalkulieren, dass die Besucher Ihrer Seiten einen gewissen Mindeststandard einhalten müssen. Wie bei den meisten Frameworks werden in jQuery einige minimale Voraussetzungen an den Browser des Besuchers gefordert⁵. Die Mindestanforderungen an die Browser können sich über die verschiedenen Versionen von jQuery verändern, aber allgemein unterstützt jQuery auch recht antike Browser. Derzeit können Sie unter <http://jquery.com/browser-support/> nachschlagen, welche Versionen von jQuery welche Browser unterstützen. Andere Browser können ebenso funktionieren, aber das wird nicht offiziell garantiert. In der Dokumentation finden Sie einige Browser offiziell aufgeführt, die grundsätzlich funktionieren, in denen es aber einige bekannte Probleme gibt.

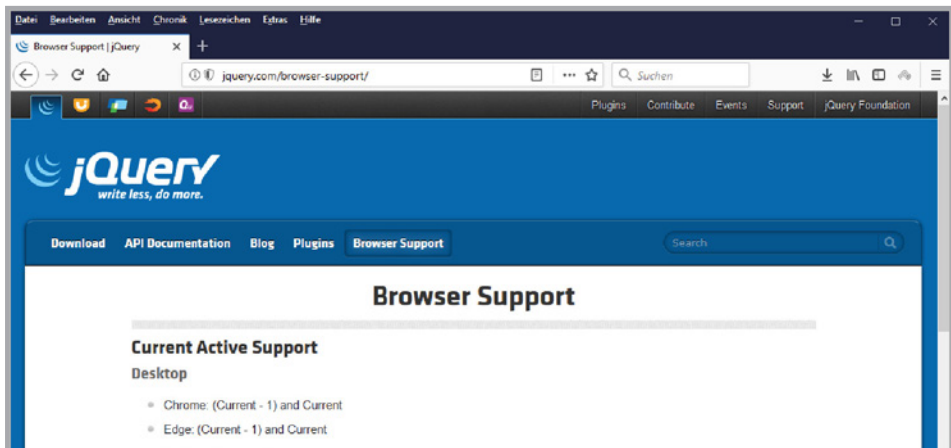


Bild 1.6 Die minimalen Anforderungen an Browser

Sie sollten nun als Webseitenersteller auf jeden Fall mehrere Browser zur Verfügung haben. Denn leider gilt auch beim Einsatz eines zuverlässigen und etablierten Frameworks wie jQuery, dass man Webapplikationen in allen relevanten Browsern testen muss.

⁵ Wir hatten schon darüber gesprochen, dass Internet Explorer vor der Version 9 in der zukünftigen Planung des Frameworks keine Rolle mehr spielen und nur mit einer Extravariante unterstützt werden.



Praxistipp

Selbstverständlich können Sie auf einem Rechner beziehungsweise einem Betriebssystem mehrere Versionen eines Browsers parallel installieren. Nur beim Internet Explorer gibt es da Probleme, denn dieser gräbt sich tief in das Windows-Betriebssystem ein und gestattet erst einmal nur die Installation einer Ausführung. Allerdings können neue Versionen des Internet Explorers in einen Modus umgeschaltet werden, dass sie ältere Varianten simulieren. Das geht für einen ersten Eindruck, wie sich eine Webseite unter einer alten Version des Browsers verhält, ganz gut. Aber es ist nur ein erster Eindruck, denn es wird nach meiner Erfahrung eine „idealisierte“ Variante des Alt-Browsers simuliert. Besser ist es definitiv, wenn Sie eine echte Engine der verschiedenen alten Internet Explorer zur Verfügung haben. Dazu hilft es, auf virtuelle Maschinen zu setzen oder aber man nimmt ein Tool wie den kostenlosen (aber leider mittlerweile etwas veralteten) **IETester** (<https://ietester.de.softonic.com/>). Dieser emuliert die Anzeige und die Verhaltensweise von Webseiten im Internet Explorer aller Versionen ab 5.5. Die Software greift dazu auf die echten Render- sowie die JavaScript-Engines der einzelnen alten Browser-Versionen zurück.

1.6.1 Verschiedene Betriebssysteme und virtuelle Maschinen zum Testen

Wie schon erwähnt, ist die Wahl eines Betriebssystems zur Arbeit mit jQuery weitgehend eine Frage persönlicher Vorlieben oder gegebener Zwänge. Das betrifft jedoch nicht die Wahl einer Testumgebung. Idealerweise haben Sie zum Testen Ihrer Applikationen mehrere Betriebssysteme zur Verfügung, denn die Besucher Ihrer Webangebote werden auch verschiedene Betriebssysteme verwenden.

Natürlich ist Windows das Referenzsystem schlechthin. Aber Linux und MacOS haben eine absolut relevante Verbreitung. Es ist teilweise interessant zu sehen, wie sich verschiedene Webapplikationen unter verschiedenen Betriebssystemen verhalten, obgleich die Unterschiede eigentlich nicht sonderlich groß sein sollten. Wenn es also geht, sollten Sie zum Testen Ihrer Applikationen verschiedene Betriebssysteme zur Verfügung haben.

Nun brauchen Sie nicht unbedingt mehrere Rechner oder parallel zu Ihrem Betriebssystem ein weiteres Betriebssystem installieren. Insbesondere für Linux gibt es ausgezeichnete Live-CDs beziehungsweise Live-DVDs, von denen Sie dieses Betriebssystem direkt starten können, ohne dass auf Ihrer Festplatte irgendeine Änderung durchgeführt wird. Für Leser mit ausreichend leistungsfähiger Hardware könnte es ebenso äußerst interessant sein, sich eine sogenannte Virtualisierungslösung (VM – Virtual Machine) wie den **VMWare Player** (<http://www.vmware.com/de>) oder **VirtualBox** (<http://www.virtualbox.org/>) von Oracle anzusehen. Diese gibt es – zumindest für den privaten Gebrauch – kostenlos und sie simulieren innerhalb eines bereits laufenden Betriebssystems ein anderes Betriebssystem. Sie können damit beispielsweise innerhalb von Windows ein Linux-System starten und umgekehrt oder Sie können zum Beispiel unter einer Windows-Installation in einem weiteren Windows auch

eine andere Version des Internet Explorer installieren. Nicht zuletzt ist es in Bezug auf Ajax interessant, dass das sogenannte Gastsystem (das System, welches in der VM läuft) als Server oder Client fungieren kann und Sie damit auf einem Rechner zwei vollkommen getrennte Systeme zur Verfügung haben, um eine Client-Server-Beziehung wie in der Realität auszuprobieren.

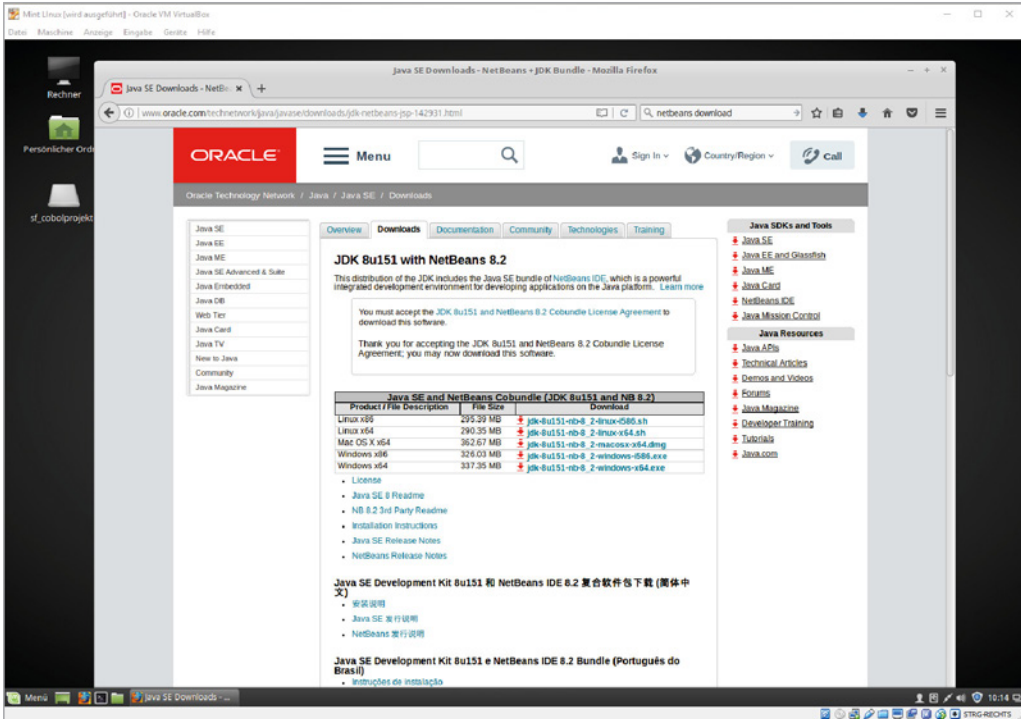


Bild 1.7 Ein Linux-System in einer VM

1.7 Die Entwicklungsumgebungen und nützliche Helferlein

Sie werden als Rückgrat einer RIA wie bei jeder Webseite so gut wie immer ein HTML-beziehungswise XHTML-Gerüst verwenden, zumindest was das endgültige Resultat beim Besucher einer Webseite angeht⁶.

⁶ Selbst wenn Sie mit einem Toolkit wie dem Google Web Toolkit (GWT) zur Entwicklungszeit in Java arbeiten oder bei einer ASP.NET-Webseite etwa mit ASPX beziehungsweise einer Webform agieren oder die Webseite auf einem Apache-Server mit PHP generieren, wird letztendlich im Browser des Besuchers HTML ankommen.

1.7.1 Editoren – oft bereits mit gewisser Hilfestellung

Zur Erstellung des HTML-Quelltextes sowie sämtlicher anderen verwendeten Techniken wie CSS, JavaScript etc. genügt deshalb als Minimalausstattung bereits ein reiner Klartexteditor, wie er bei jedem Betriebssystem mitgeliefert wird.

In der Praxis verwenden Sie jedoch meist mächtigere Programmierwerkzeuge, die Sie bei der Erstellung und Analyse des Quelltextes unterstützen. Solche Programme kennen etwa einige Bestandteile einer Programmier- oder Beschreibungssprache (etwa HTML, CSS oder JavaScript) und unterstützen einfache und teilweise auch komplexere Standardvorgänge, etwa das Maskieren (die kodierte Darstellung) von Sonderzeichen, das Einfügen von Quellcodeschablonen oder eine bessere Übersichtlichkeit durch farbliche Kennzeichnung von bekannten Befehlen. Einige Editoren bieten einem Programmierer auch die Befehle einer verwendeten Sprache direkt an – etwa durch Menüs oder Symbolleisten, wo der Anwender diese auswählen kann (auch mit der Maus). **Notepad++** (<http://notepad-plus-plus.org/>) ist zum Beispiel ein hervorragender Editor mit so einer Unterstützung für Windows. Für Windows, aber auch Linux und macOS ist **Brackets** (<http://brackets.io/>) eine sehr gute Wahl.

Ein weiteres Feature, das einige Programme bieten, ist die Bereitstellung von verschiedenen Ansichten eines Dokuments. Das ist insbesondere bei reinen HTML-Editoren oft zu finden. Dort kann man in vielen Fällen zwischen der Vorschau einer Webseite (also wie sie in einem Browser aussehen wird), einem grafischen Erstellungsmodus und vor allem einer Ansicht des eigentlichen HTML-Codes hin- und herschalten.

1.7.2 Integrierte Entwicklungsumgebungen

In den letzten Jahren gab es enorme Fortschritte bei Entwicklungstools für die Webprogrammierung und es existieren mittlerweile einige richtige IDEs (Integrated Development Environment), die den Vergleich zu IDEs in mächtigen Sprachen wie Java oder C# nicht scheuen brauchen. Diese gestatten eine Programmierung und eine Ausführung von Webprojekten auf einer integrierten, gemeinsamen Oberfläche. Umgekehrt muss man aber auch festhalten, dass einige sehr gute Tools wie **Aptana** beziehungsweise das **Aptana Studio** (<http://aptana.com/>), was unmittelbar auf der kostenlosen, mächtigen und universellen Entwicklungsumgebung **Eclipse** (<http://www.eclipse.org>) basiert, teils nicht mehr konsequent weiterentwickelt werden und damit für moderne Anforderungen ein bisschen den Anschluss verlieren. Sie sollten bei der Wahl einer IDE oder eines Tools also immer darauf achten, ob es noch gepflegt wird und zukunftssicher erscheint. Ich möchte drei IDEs explizit empfehlen, wobei Sie sehr gerne Ihre persönlichen Favoriten verwenden können.

Eclipse

Wenn Sie auf Eclipse setzen, gibt es bereits in den meisten Konfigurationen Unterstützung für Web-Technologien, aber man kann die IDE auch mit ein paar kostenlosen und dennoch sehr mächtigen Plugins erweitern. An erster Stellen sei das **Web Tools Platform Project (WTP)** genannt (<https://www.eclipse.org/webtools/>). Dabei sollte man im Auge behalten, dass es verschiedene Varianten von Eclipse (Packages) gibt, die für verschiedene Ziele vor-konfiguriert sind. Wenn Sie Eclipse noch nicht besitzen, führt zur zukünftigen Erstellung

von Webprojekten einer der einfachsten Wege über die neueste Version der Eclipse-IDE für Java-EE-Entwickler (*Eclipse IDE for Java EE Developers*). Sie können die zum Zeitpunkt der Bucherstellung aktuellste Version von <http://download.eclipse.org> bzw. <https://www.eclipse.org/downloads/packages/eclipse-ide-java-ee-developers/oxygen1a> herunterladen. Darin ist das WTP bereits enthalten. Aber auch die *Eclipse IDE for JavaScript and Web Developers* (<http://www.eclipse.org/downloads/packages/eclipse-ide-javascript-and-web-developers/oxygen1a>) eignet sich wunderbar. Diese ist etwas schlanker und hat den „Vorteil“, dass sie nicht überfrachtet ist mit Features, die Sie bei der Entwicklung von Webseiten mit jQuery möglicherweise nicht brauchen und die eher verwirren. Aber auch die *Eclipse for PHP Developers* beinhaltet explizit die gewünschten Webfeatures.

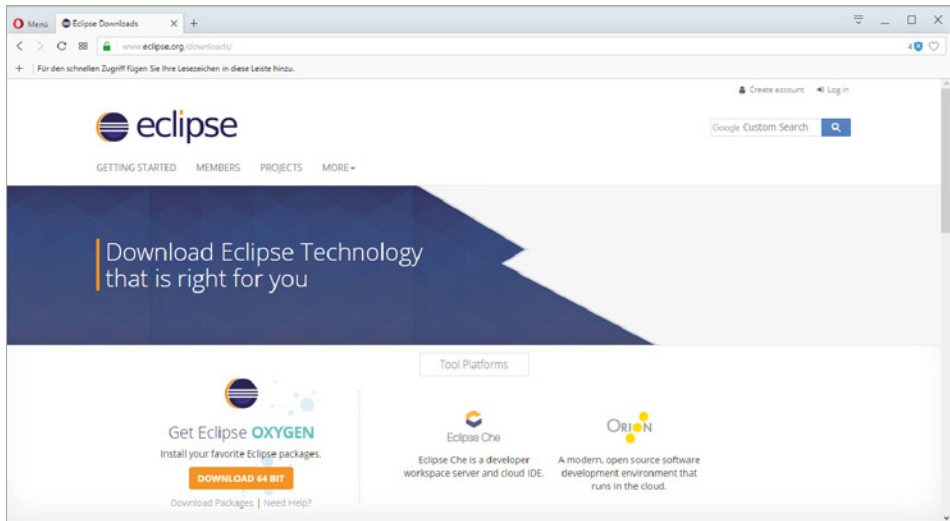


Bild 1.8 Hier gibt es Eclipse allgemein zum Download.

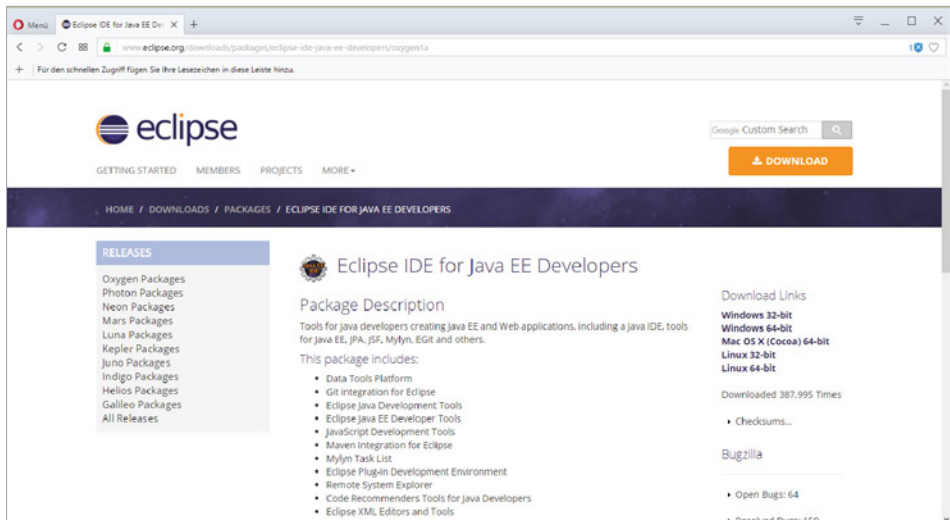


Bild 1.9 Die Eclipse-IDE für Java-EE-Entwickler enthält bereits Web-Tools

Es kann allerdings sein, dass Sie vor einer Installation von Eclipse eine Java-Umgebung benötigen, denn Eclipse ist in Java geschrieben. Dabei gibt es einmal die sogenannte **JRE** (Java Runtime Environment), deren wesentlichen Bestandteil die **JVM** (Java Virtual Machine – ein virtueller Prozessor) ist und die zum reinen Ausführen von Java-Programmen vorhanden sein muss. Allerdings sollten moderne Betriebssysteme automatisch eine JRE bereitstellen. Darüber hinaus gibt es das **JDK** (Java Development Kit), das für die Entwicklung von Java-Anwendungen notwendig ist und auch noch einmal eine JRE installiert, aber IDEs auf Java-Basis wie Eclipse und NetBeans in der Regel mitbringt. Bei Bedarf finden Sie beispielsweise das JDK auf den Java-Seiten (Java SE Download) von Oracle unter <http://www.oracle.com/technetwork/java/javase/downloads/index.html> zum Download.

Wenn Sie Eclipse bereits installiert haben, können Sie in neueren Versionen von Eclipse (ab der Version Kepler) das WTP über den Marktplatz installieren. Wählen Sie andernfalls in Eclipse das Menü **HELP** und wählen Sie **INSTALL NEW SOFTWARE** und dann die gewünschten Funktionen aus den Kategorien **WEB**, **XML**, **JAVA EE AND OSGI ENTERPRISE DEVELOPMENT** aus. Sie kommen auch über die Willkommenseite von Eclipse zum Marktplatz und können dort die Auswahl vornehmen.

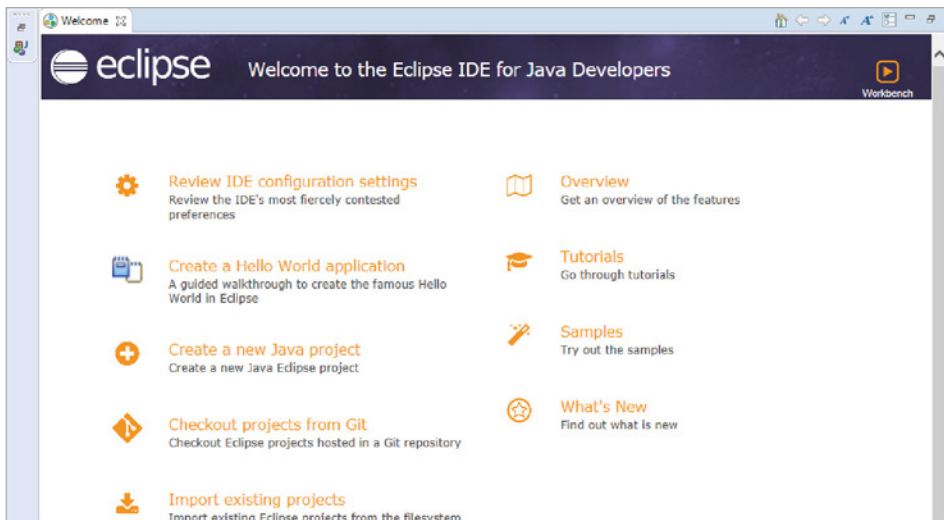


Bild 1.10 LAUNCH THE ECLIPSE MARKETPLACE führt aus dem Willkommensdialog zum Marktplatz.

Als eine Erweiterung für Eclipse möchte ich noch die Eclipse Web Developer Tools (<https://marketplace.eclipse.org/content/eclipse-web-developer-tools-0>) nennen. Mittlerweile ist Eclipse so weit mit modernen Browsern und Webseiten zur Bereitstellung von Software gediehen, dass man sogar das Plugin zur Installation per Drag & Drop aus der Webseite in den Workspace von Eclipse ziehen kann.

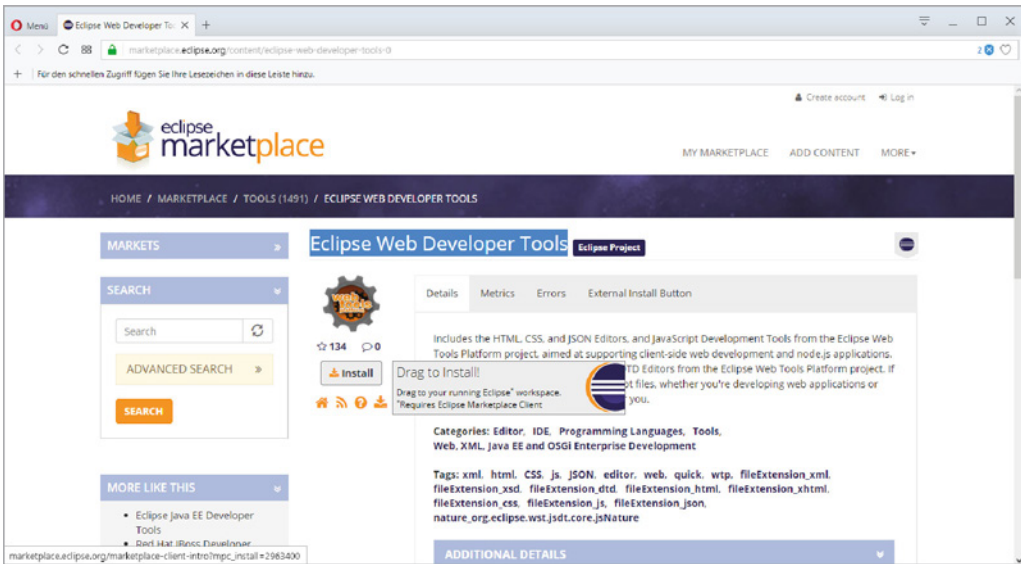


Bild 1.11 Man kann bei einigen Plugins aus der Webseite per Drag & Drop installieren.

Aber auch der traditionelle Weg über die Hilfe und Installation von neuer Software funktioniert, wie er oben beschrieben wurde.

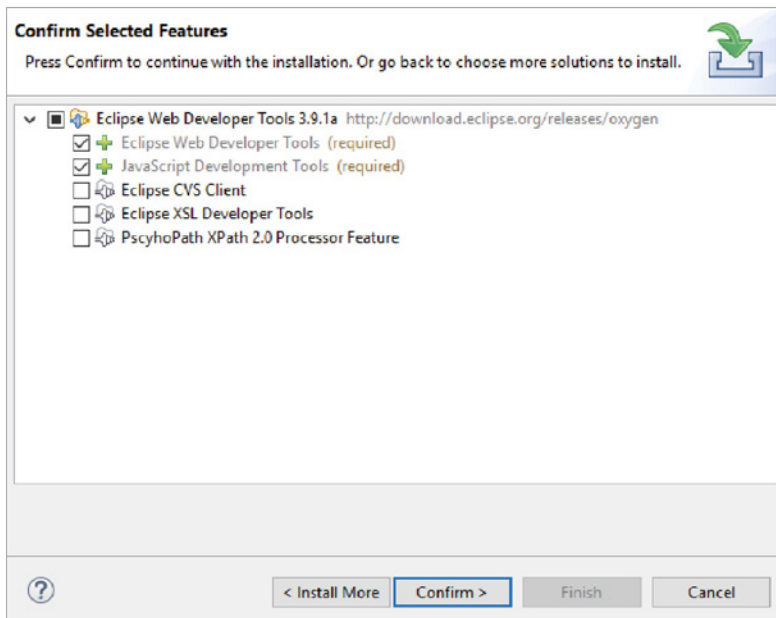


Bild 1.12 Eclipse bietet schon immer über das HELP-Menü Erweiterungsmöglichkeiten.

Es ist nun nach einer Erweiterung von Eclipse in der Regel notwendig, dass Sie das Programm neu starten, damit die Erweiterung bereitsteht.

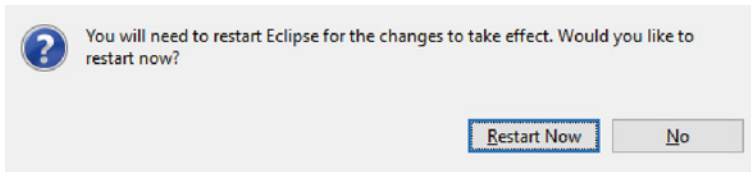


Bild 1.13 Ein Neustart von Eclipse ist notwendig

Alle diese modernen Web-Erweiterungen von Eclipse bieten einen Quellcodeeditor und unterstützen mit zahlreichen Features direkt JavaScript, HTML und CSS sowie den Zugriff auf das DOM-Objektmodell und auch Ajax selbst. Dabei versucht eine Code-Assist-Funktion, verschiedene Benutzereingaben in Form einer Autovervollständigung zu ergänzen und die Syntax wird durch Syntaxhighlighting (farbliche Kennung von Schlüsselbegriffen und Syntaxstrukturen) hervorgehoben. Besonders interessant ist aber die Möglichkeit, die Eigenschaften und Methoden von Objekten angezeigt zu bekommen und JavaScript zu debuggen.

Visual Studio

Wie schon an anderer Stelle erwähnt, wird jQuery auch von großen Playern im Entwicklerumfeld in ihre Tools integriert beziehungsweise darin unterstützt, beispielsweise von Microsoft in neuen Versionen des **Visual Studio**. Gerade wenn Sie aus dem Bereich von ASP.NET kommen, ist das Visual Studio, das es auch in einer kostenlosen Community-Version gibt, für Sie sicher eine hervorragende Entwicklungsumgebung (<http://www.microsoft.com/visualstudio>). Bereits in der Grundinstallation stehen Editoren und Vorlagen zur Verfügung und die hervorragende Codeunterstützung von Visual Studio greift auch bei Web-Technologien. Mit sogenannten Extensions wie den Web Essentials kann man Visual Studio aber bei Bedarf auch erweitern.

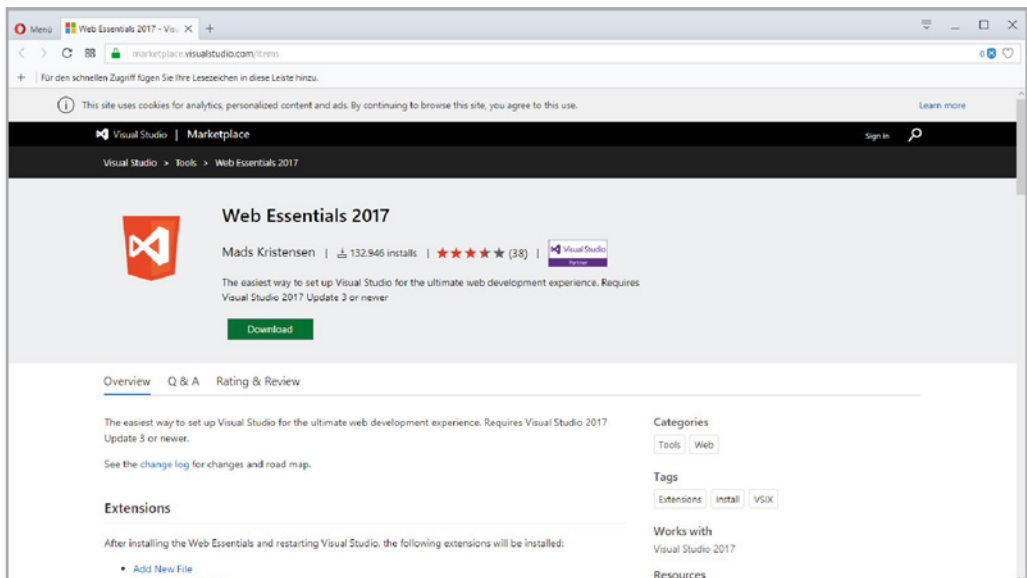


Bild 1.14 Erweiterungen für Visual Studio

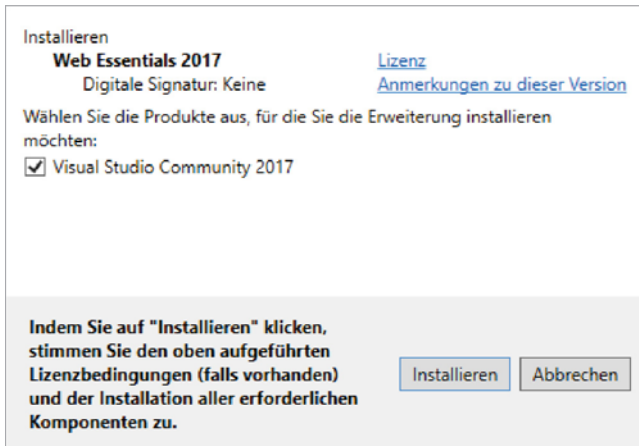


Bild 1.15 Installation der Erweiterung

Aber auch sonst werde ich mehr und mehr zum Fan von Visual Studio, denn die IDE wird mit diversen Extensions nahezu universell einsetzbar.

NetBeans

Als Alternative zu Eclipse und Visual Studio soll noch mit NetBeans (<http://netbeans.org/downloads/index.html>) die kostenlose Java-IDE von Sun respektive Oracle erwähnt werden. Wie Eclipse ist auch NetBeans selbst in Java geschrieben und wie bei Eclipse kann man mit NetBeans nicht nur Java-Programme schreiben (obwohl es auch bei NetBeans ganz klar der Schwerpunkt ist), sondern es werden mittlerweile auch diverse andere Sprachen unterstützt. Und eben auch Web-Technologien. Und noch eine Gemeinsamkeit zu Eclipse ist der vollständig modulare Aufbau, der durch Plugins erweitert werden kann. Die Installation von NetBeans ist wie bei Eclipse vollkommen unproblematisch. NetBeans als Bundle wird mit dem JDK zusammengefasst und Sie benötigen also in der Regel kein separates JDK (wobei das nicht schadet).

1.7.3 Kleine Helferlein

Neben vollständigen IDEs gibt es noch eine ganze Reihe kleiner Helferlein für Webseiten-ersteller und Webprogrammierer – egal, ob sie mit jQuery arbeiten oder nicht. Die nachfolgende Tabelle gibt ein paar Tipps aufgrund der Erfahrungen des Autors.

Tabelle 1.1 Nützliche Helferlein für die Entwicklung im Web

Tool	Beschreibung	Webseite
Dokumentations-tools	Aus Java und anderen mächtigen Sprachen kennt man Tools zur automatisierten Dokumentation von EDV-Projekten aus den Quelltexten selbst. Diese gibt es mittlerweile auch für JavaScript. Dabei kommen auch dort meist spezielle Kommentare im Quelltext zum Einsatz.	https://github.com/jsdoc3/jsdoc
JSFiddle	Das Tool ist eine Art Spielwiese beziehungsweise Testumgebung für JavaScript-Frameworks. Innerhalb einer RIA kann ein Entwickler mit kleinen Codeschnipseln experimentieren, Frameworks (natürlich auch jQuery) ausprobieren oder auch nur CSS-Regeln ändern. In einem Fensterbereich der RIA sieht man unmittelbar, wie sich ein Code auswirken würde.	http://jsfiddle.net/
JSLint	Bei diesem Tool handelt es sich um einen Qualitätstester, mit dem Sie JavaScript-Code überprüfen können. Man kann damit Fehler, aber vor allen Dingen unsaubere Codestrukturen finden.	http://www.jshint.com/
Kompressoren	Wie bereits besprochen, gibt es für jQuery verschiedene Varianten der Bibliothek zum Download, u. a. eine mit <i>min</i> gekennzeichnete minierte Version, die um Kommentare und überflüssige Leerzeichen sowie Zeilenumbrüche reduziert ist. Funktional ist diese identisch mit der unkomprimierten Version, die im Quellcode kommentiert und gut lesbar, aber dementsprechend größer ist. Solch eine Kompression können Sie auch für Ihre eigenen JavaScripts, HTML-Seiten sowie CSS-Dateien nutzen. Dazu gibt es einige Anbieter, die das direkt über Webformulare ermöglichen und von denen Sie hier zwei Vorschläge sehen.	http://htmlcompressor.com/compressor/

1.7.4 Node.js und Git

Kommen wir noch einmal auf **Node.js** beziehungsweise **NodeJS** zurück und erweitern die Ausführung um **Git**. Bei beiden Projekten handelt es sich im weiteren Sinne um Softwareverwaltungsmittel mit zum Teil überschneidenden Anwendungen, zum Teil aber auch ganz unterschiedlichen Schwerpunkten. Mit beiden werden Sie aber unter Umständen beim Umgang mit jQuery (was wir schon im Zusammenhang mit npm gesehen haben) und Webprogrammierung im Allgemeinen konfrontiert.

Git ist eine weitere Software zur verteilten Versions- und Softwareverwaltung, die aktuell noch häufiger als Node.js zu diesem Zweck eingesetzt wird.

Wenn Sie Git zur Installation von Software auf Ihrem Computer nutzen wollen, installieren Sie am einfachsten einen Git-Client, den es für verschiedene Betriebssysteme gibt (<http://git-scm.com/downloads>).

1.7.5 Integrierte Entwicklungstools in Browser und Browser-Add-ons

Alle modernen Browser besitzen mittlerweile integrierte Entwicklungstools, die Sie oft mit der Taste F12 oder einem Kontextmenü aufrufen können. Diese sind inzwischen so leistungsfähig, dass man in der Regel für die Web-Entwicklung keine Browsererweiterungen mehr benötigt, was in den vergangenen Jahren eigentlich unabdingbar war.

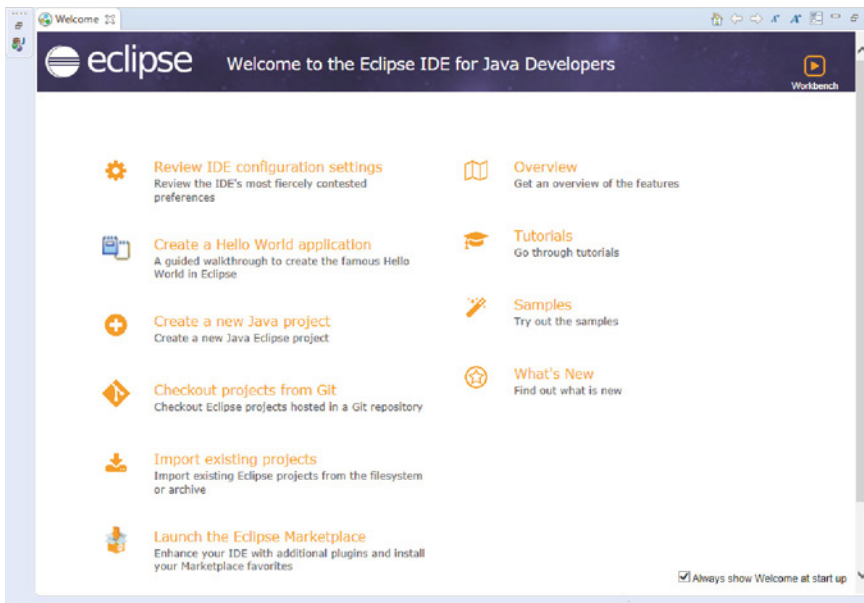


Bild 1.16 Die integrierten Entwicklungstools in einem Firefox

Diese integrierten Tools erlauben eine detaillierte Analyse einer Webseite – vom Aufbau des DOM über die Stylesheets und Skripte bis hin zum konkreten Netzwerkverkehr.

1.7.5.1 Add-ons

Sie können des Weiteren auch die meisten Browser um Tools zur Web-Entwicklung (und natürlich auch anderer Dinge) erweitern. Gerade für den Webbrowser Firefox gab es in der Vergangenheit zahlreiche Erweiterungen (sogenannte Add-ons), die viele weitere nützliche Features für die Webentwicklung bereitstellten. Etwa Firebug oder den COM Inspector. Diese Erweiterungen gibt es zwar zum Teil immer noch, aber durch die schnellen Versionszyklen und diversen Umstellungen des Browser-APIs ist die Anzahl verfügbarer Add-ons in neuen Versionen erheblich eingedampft worden.

1.7.6 Der Webserver zum realistischen Testen

Gerade bei Ajax kommt es ja im Wesentlichen darauf an, dass von einem Browser aus Daten von einem Webserver angefordert und ohne Neuladen der Webseite in diese eingebaut werden. Deshalb ist es für die praktische Ausführung und damit auch das Testen von solchen Applikationen unabdingbar, dass Sie entweder Zugang zu einem Webserver im Internet haben und dort Programme oder Skripte ausführen können. Das wird letztendlich auch für ein Ajax-Projekt in der Praxis notwendig sein. Allerdings ist es in der Praxis ebenso unüblich, schon bei der Entwicklung einer Webapplikation direkt auf einem Webserver im Internet zu arbeiten, vor allem, wenn Sie nur Dinge testen wollen. Aber auch ohne Ajax ist es für den echten Test einer Webapplikation unabdingbar, dass man sie unter realen Bedingungen auf einem Webserver ausprobiert. Sie sollten sich also eine Testumgebung mit einem Webserver auf einem lokalen Rechner oder in einem lokalen Netzwerk schaffen. Gerade Linux-Distributionen beinhalten fast immer bereits ein oder mehrere Webserver. Auch bringen verschiedene Entwicklungsumgebungen für Webapplikationen einen integrierten Webserver mit, etwa das oben schon genannte Aptana oder das Visual Studio. Damit sind Sie natürlich auf der sicheren Seite.

1.7.6.1 XAMPP

Aber auch wenn Sie keinen Webserver automatisch zur Verfügung haben oder sich die Sache nur möglichst einfach machen wollen, bietet sich ein Rundum-Sorglos-Paket wie **XAMPP** an, das Sie für verschiedene Betriebssysteme aus dem Internet laden können (unter <http://www.apachefriends.org/de/>).

Dieses Paket bezeichnet eine Sammlung von Programmen mit dem Webserver **Apache** im Zentrum, der durch das Datenbankmanagementsystem **MySQL** bzw. in neuen Versionen dessen Fork⁷ **MariaDB** (inklusive **phpMyAdmin** zur Administration des Datenbankmanagementsystems) und PHP-Unterstützung, den FTP-Server **FileZilla** sowie einige weitere Webtechnologien ergänzt wird. Sie brauchen dieses Paket mit einem einfachen Assistenten nur zu installieren und schon haben Sie einen voll funktionstüchtigen Webserver in einer Grundkonfiguration zur Verfügung.

Beachten Sie aber, dass diese Pakete von XAMPP in der Grundeinstellung ausschließlich für lokale Testzwecke konfiguriert sind. Um die Sache möglichst einfach zu halten, sind sämtliche Sicherheitseinstellungen niedrig eingestellt.

Sobald die Installation von XAMPP fertig ist, können Sie entweder Apache manuell starten oder aber auch so einrichten, dass Apache als Dienst beziehungsweise Prozess in Ihrem Betriebssystem integriert und sogar automatisch beim Start des Rechners aufgerufen werden kann. XAMPP stellt zur Verwaltung ein komfortables und sehr einfach anzuwendendes Kontrollprogramm zur Verfügung.

⁷ Eine vollkompatible Abwandlung, die sich nur lizenzrechtlich unterscheidet.

■ 1.8 Zusammenfassung

Sie haben in dem einleitenden Kapitel erfahren, um was es in dem Buch geht und an wen es sich wendet sowie wie das Buch aufgebaut ist. Dazu wurden vor allen Dingen alle wichtigen Voraussetzungen besprochen, damit Sie erfolgreich die Erstellung von RIAs auf Basis von jQuery angehen können.

2

Sprung ins kalte Wasser

In diesem Kapitel werden wir ohne weitere Vorbereitungen Kontakt zu jQuery schaffen und erste Beispiele mit jQuery erstellen. Wir springen also direkt ins kalte Wasser. Sie sollen bereits in dieser frühen Phase Ihres Einstiegs in dieses faszinierende Thema ein Gefühl für das bekommen, was man mit jQuery anstellen kann und was Ihnen dieses Framework bringt. Dabei wird bewusst in Kauf genommen, dass zu diesem Zeitpunkt bei den Quelltexten Fragen offen bleiben. Diese Fragen werden aber im Laufe der folgenden Kapitel geklärt. Die Erläuterungen zu den Listings werden auch in dieser Phase nur oberflächlich¹ sein, um nicht vom Stock zum Stöckchen zu geraten. Wir wollen möglichst schnell zur Praxis mit jQuery kommen und erst einmal spielen. Und das bedeutet Beispiele erstellen.

■ 2.1 Zugriff auf Elemente und Schutz des DOM

Wenn Sie sich bereits etwas mit der Programmierung im WWW auskennen, wissen Sie, dass man auf die Bestandteile einer Webseite per JavaScript oder einer anderen Skriptsprache im Browser über ein Objektmodell mit Namen **DOM** (Document Object Model) zugreifen kann. Es gibt für so einen Zugriff verschiedene Standardtechniken, die aber alle ihre spezifischen Schwächen haben. Insbesondere müssen Sie beim Zugriff auf ein einziges Element der Webseite (oder eine Gruppe) in der Regel ziemlich viele Zeichen eingeben. Das ist mühselig und fehleranfällig. Die meisten Frameworks stellen deshalb eine Notation zur Verfügung, über die so ein Zugriff mit einer verkürzten, vereinheitlichten Schreibweise erfolgen kann. Und zudem kompensieren die dahinterliegenden Mechanismen der Frameworks diverse Schwächen der Standardzugriffsverfahren, indem sie vor allen Dingen browserabhängige Besonderheiten kompensieren sowie diverse fehlende Funktionalitäten des reinen DOM-Konzepts ergänzen. Besonders wichtig – diese Kompensation ist in der Regel auf allen offiziell unterstützten Browsern getestet und funktioniert deshalb sehr zuverlässig.

¹ Aber keinesfalls unwichtig.

Das folgende Beispiel zeigt weiterhin eine andere wichtige Funktionalität von jQuery – den Schutz des DOM. Was es damit auf sich hat, wird natürlich noch viel genauer erläutert. Nur soweit vorab – beim Laden (Parsen) der Webseite verarbeiten verschiedene Browser die Webseite unterschiedlich und es kann beim Zugriff auf die Elemente der Webseite zu einer Vielzahl von Problemen kommen. Das gilt vor allen Dingen dann, wenn man in einem Skript **zu früh** auf Elemente einer Webseite zugreifen will – also bevor der Browser den DOM korrekt aufgebaut hat. Hier bietet jQuery ein zuverlässiges Verfahren, um diesem Problem Herr zu werden. Und was Ihnen das Beispiel quasi nebenbei noch zeigt ist, wie Sie per jQuery standardisiert auf Inhalte von Elementen mit Text zugreifen und auf Ereignisse reagieren können. Doch genug der Vorbemerkung – hier ist unser erstes Listing (*kap2_1.html*):

Listing 2.1 Das erste jQuery-Beispiel

```
<!DOCTYPE html>
<html lang="de" xmlns="http://www.w3.org/1999/xhtml">
<head>
  <meta charset="utf-8" />
  <title>Schutz des DOM</title>
  <link href="lib/css/kap2_1.css" rel="stylesheet" type="text/css" />
  <script src="https://code.jquery.com/jquery-3.2.1.min.js"
    integrity="sha256-hwg4gsxgFZh0sEEamdOYGBf13FyQuiTwlAQgxVSNgt4="
    crossorigin="anonymous"></script>
  <script type="text/javascript">
    $(document).ready(function () {
      $("#a").click(function () {
        $("#ausgabe").html("Muss das sein?");
      });
      $("#b").click(function () {
        $("#ausgabe").html("Ein nettes Spiel.");
      });
      $("#c").click(function () {
        $("#ausgabe").html("Ein seltsames Spiel. " +
          "Der einzig gewinnbringende Zug ist " + "nicht zu spielen!");
      });
    });
  </script>
</head>
<body>
  <h1>Willkommen bei WOPR</h1><h3>Wie wäre es mit einem kleinen Spiel?</h3>
  <button id="a">Tic Tac Toe</button><button id="b">Schach</button>
  <button id="c">Weltweiter Thermonuklearer Krieg</button>
  <div id="ausgabe"></div>
</body>
</html>
```

Erstellen Sie die HTML-Datei einfach in einem eigenen Arbeitsverzeichnis und speichern Sie sie unter dem genannten Namen.



Hinweis

Beachten Sie, dass jQuery in dem Listing (als auch folgenden) von einem **CDN** geladen wird, aber das ist für die Funktionalität irrelevant. Sie können die JavaScript-Datei von jQuery auch von Ihrem Webserver oder lokal referenzieren. Aber gerade für die ersten Beispiele ist die Referenz auf ein CDN einfacher, da Sie das Framework nicht selbst bereitstellen müssen. Die Attribute `integrity` `crossorigin` werden dabei für die Überprüfung der Unterressourcenintegrität (**SRI** – Subresource Integrity) verwendet. Dadurch können Browser sicherstellen, dass auf Servern von Drittanbietern gehostete Ressourcen nicht manipuliert wurden.

Die Verwendung von SRI wird allgemein empfohlen, wenn Bibliotheken aus einer Drittanbieterquelle geladen werden. Sollte es Probleme geben, wenn man beispielsweise eine Webseite aus einer IDE wie Visual Studio und deren internen Webserver oder Ihren lokalen Webserver lädt, verzichten Sie zur Entwicklungszeit einfach auf die beiden Attribute und notieren für die Referenz auf die jQuery-Bibliothek einfach `<script src="https://code.jquery.com/jquery-3.2.1.min.js"></script>`. Das funktioniert einwandfrei. Wenn Sie Webseiten veröffentlichen und ein CDN nutzen, sollten Sie aber auf jeden Fall die Attribute verwenden.

In der Praxis fasst man alle eigenen Ressourcen eines Projekts meist innerhalb eines eigenen Verzeichnisses zusammen. Für ein Webprojekt ist es das Sinnvollste, diese Verzeichnisse im freigegebenen Ordner Ihres Webserver anzulegen. Im Fall von Apache/XAMPP wäre das in der Regel das Verzeichnis `htdocs`. Das hat den Vorteil, dass Sie – wenn der Webserver läuft – zum Test unmittelbar über HTTP und einen richtigen Webaufruf gehen können und nicht nur die Datei über das FILE-Protokoll in den Browser laden müssen (also das klassische Öffnen als Datei oder das einfache Reinziehen der Datei in den Browser). Letzteres ist ja nicht praxisorientiert, da später die Seiten natürlich auch vom Besucher über einen Webserver angefordert werden.

Wenn Sie mit einer IDE wie Eclipse oder dem Visual Studio arbeiten, kann man meist direkt aus der IDE eine Webseite über einen integrierten Webserver ausführen und in einen Browser Ihrer Wahl (der natürlich installiert sein muss) laden. In Visual Studio können Sie die Ausführung über die Funktionstaste `STRG + F5` aufrufen.

Im Head der Webseite sehen Sie einen Verweis auf eine CSS-Datei `kap2_1.css`, die wir der Vollständigkeit hier kurz angeben wollen (sie spielt aber im Grunde keine Rolle):

Listing 2.2 Die referenzierte CSS-Datei

```
body {
  background:black; color:white; font-size:20px;
}
#ausgabe {
  background:white; color:red; font-size:20px; padding:10px; margin:10px;
  border-width:1pt; border-style:solid; width:350px; min-height:75px;
}
```



Praxistipp

Wenn Sie in Visual Studio in einer Projektmappe mehrere Projekte zusammenfassen, müssen Sie immer das richtige Startprojekt festlegen, wenn Sie aus der IDE eine Webseite über den integrierten Webserver von Visual Studio in einen Browser laden wollen. Sonst kann es zu Fehlern kommen. Das Startprojekt können Sie im Projektmappen-Explorer mit einem Klick mit der rechten Maustaste auf die passende Projektmappe und das Kontextmenü festlegen.

HTTP Error 404.0 - Not Found

Die gesuchte Ressource wurde entfernt oder umbenannt, oder sie steht vorübergehend nicht zur Verfügung.

Wahrscheinlichste Ursachen:

- Das angegebene Verzeichnis oder die angegebene Datei ist auf diesem Webserver nicht vorhanden.
- Die URL enthält einen Schreibfehler.
- Der Zugriff auf die Datei wird von einem benutzerdefinierten Filter oder Modul wie URLScan eingeschränkt.

Mögliche Vorgehensweise:

- Erstellen Sie den Inhalt auf dem Webserver.
- Überprüfen Sie die Browser-URL.
- Überprüfen Sie das Ablaufverfolgungsprotokoll für Anforderungsfehler, und überprüfen Sie, welches Modul "SetStatus" aufruft. Klicken Sie [hier](#), um weitere Informationen zu erhalten.

Detailed Error Information:

Module	IIS Web Core	Requested URL	http://localhost:53262/kap2_1.html
Notification	MapRequestHandler	Physical Path	F:\xampp\htdocs\jquery\Kap2\Kap2\kap2_4\kap2_1.html
Handler	StaticFile	Logon Method	Anonym
Error Code	0x80070002	Logon User	Anonym
		Verzeichnis für Ablaufverfolgung	F:\Users\ralph\Documents\IISExpress\TraceLogFiles\KAP2_4

More Information:

Dieser Fehler weist darauf hin, dass die Datei oder das Verzeichnis auf dem Server nicht vorhanden sind. Erstellen Sie die Datei oder das Verzeichnis, und führen Sie die Anforderung erneut aus.

[View more information »](#)

Bild 2.1 Bei einer Meldung der Art findet der Webserver die HTML-Datei nicht oder darf darauf nicht zugreifen.



Praxistipp

Die CSS-Datei befindet sich in einem Unterverzeichnis *lib* des Projektverzeichnisses, in dem die Webseite gespeichert wurde. Dieses Verzeichnis enthält noch ein weiteres Unterverzeichnis *css*, worin konkret die Datei gespeichert ist. Wenn wir mit externen JavaScript-Dateien arbeiten, werden diese dann in einem Unterverzeichnis *js* des Unterverzeichnisses *lib* des Projektverzeichnisses gespeichert. Diese Strukturierung hat sich in der Praxis so oder ähnlich auf breiter Front durchgesetzt. Das bedeutet, dass sich auch die jQuery-Bibliothek genau da befinden wird, wenn Sie statt eines CDN eine heruntergeladene Version verwenden, die Sie dann selbst bereitstellen. Aber selbstverständlich können Sie auch eine ganz andere Pfadstruktur wählen. Nur sollten Sie grundsätzlich strukturieren und das konsequent durchziehen.

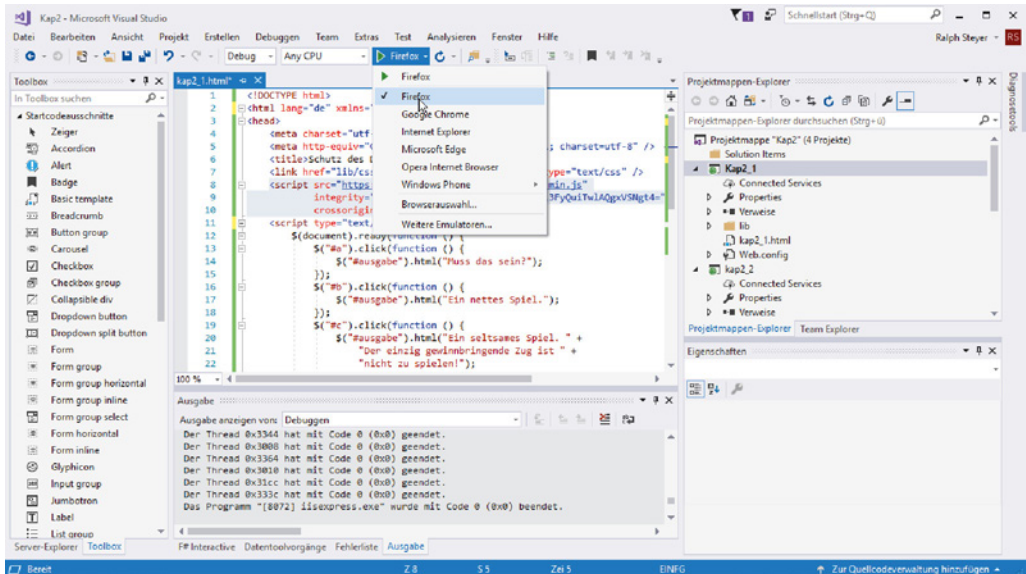


Bild 2.2 Die Projektstruktur – hier im Projektmappen-Explorer von Visual Studio gut zu sehen

In den Zeilen hinter der Referenz auf die externe CSS-Datei sehen Sie die Referenz auf eine externe JavaScript-Datei – die jQuery-Bibliothek, die in dem konkreten Fall wie gesagt von einem CDN geladen wird. In den folgenden Zeilen steht ein gewöhnlicher JavaScript-Container. In diesem wird mit `$(document)` die Webseite angesprochen. Die Funktion `$()` steht in jQuery für eine Kurzschreibweise, um ein Element der Webseite zu referenzieren. Sie ist **der** (!) zentrale Dreh- und Angelpunkt des gesamten Frameworks und Sie finden diese verkürzten Zugriffsschreibweisen auch in den folgenden Zeilen immer wieder. Nur wird dort als Parameter eine sogenannte ID eines Elements verwendet.

**Hinweis**

Beachten Sie, dass ein Element (im Sinne von jQuery) als Parameter von `$()` nicht in Hochkommata eingeschlossen wird, eine ID (oder ein anderer Selektor) hingegen schon.

Widmen wir uns kurz der Methode `ready()`. Diese stellt sicher, dass die enthaltenen Aufrufe erst dann ausgeführt werden, wenn die Webseite vollständig geladen und der DOM korrekt aufgebaut wurde. Wie schon angedeutet und ohne richtig in die Tiefe zu gehen – das ist bereits ein Feature, dessen Wert man hoch einschätzen muss.

**Hinweis**

Für die Leser mit entsprechenden Vorkenntnissen ein kleiner Vorgriff – die Methode `ready()` ist eine Alternative für den Eventhandler `onload`. Dieser Eventhandler galt jedoch lange Zeit als unzuverlässig, weil er fehlerhaft in verschiedenen älteren Browser implementiert war.

Im Inneren der `ready()`-Methode werden drei Ereignisbehandlungsroutinen notiert, die jeweils die Reaktion bei einem Klick auf die angegebenen Elemente spezifizieren. In unserem Beispiel sind das drei Schaltflächen, die jeweils mit einer eindeutigen ID gekennzeichnet sind.

**Hinweis**

Die Methode `click()` kapselt naheliegender Weise den Funktionsaufruf des Eventhandlers `onclick`.

Die Zuordnung zur richtigen Funktion erfolgt über die ID und das Auslösen der Funktion innerhalb der Methode `click()`. Beachten Sie, dass wir hier eine sogenannte **anonyme Funktion** (also ohne Bezeichner) verwenden.

Interessant wird es, wenn ein Anwender nun auf eine Schaltfläche klickt. Dabei wird in einem Bereich der Webseite eine spezifische Textausgabe angezeigt. Dazu verwenden wir wieder `$()` und eine ID für die Selektion des Bereichs (ein `div`-Block) und die Methode `html()` für den Zugriff auf den Inhalt.

**Hinweis**

Die Methode `html()` ist in jQuery die Alternative zur Verwendung von `innerHTML`. Das Interessante dabei ist, dass `innerHTML` schon in der Praxis seit vielen Jahren verwendet, aber erst mit HTML5 offiziell standardisiert wird.



Bild 2.3 Die Webseite mit den drei Buttons – der Anwender hat die dritte Schaltfläche angeklickt.



Hinweis

Wir werden in allen folgenden Beispielen darauf verzichten, den gesamten Head der Webseite abzdrukken. Die Notation würde nur Platz im Buch verschwenden, da sie immer (fast) gleich ist. Nur bei interessanten Änderungen wird der Code abgedruckt.

2.2 Veränderung der Webseite mit DHTML

Grundsätzlich kann man mit Style Sheets die Optik einer Webseite viel besser und effektiver gestalten als mit reinem HTML. Insbesondere kann man damit das Layout von der Struktur der Seite abtrennen. Diese Aussagen sollten – so richtig sie auch sind – für Sie kalter Kaffee sein. Wenn Sie nun die Style Sheets einer Seite dynamisch mit JavaScript verändern, reden wir von DHTML. Aber auch Animationseffekte wie das Ein- und Ausblenden von Teilen einer Webseite über andere JavaScript-Techniken gehören dazu. Lassen Sie uns in diesem und dem folgenden Beispiel ansehen, wie Sie animierte Webseitenänderungen mit jQuery schnell, einfach und bequem und dennoch zuverlässig in den unterschiedlichen Browsern bewerkstelligen können. In diesem Beispiel wechseln wir im Wesentlichen dynamisch die CSS-Klasse eines Elements.

Zuerst betrachten wir eine kleine CSS-Datei, die in der folgenden Webseite eingebunden werden soll und im *lib/css*-Verzeichnis gespeichert sein sollte (*kap2_2.css*):

Listing 2.3 Die neue CSS-Datei

```
body {  
    background: black; color: white; font-size: 20px;  
}  
div {
```

```

background: white; color: red; font-size: 20px; padding: 10px; margin: 10px;
border-width: 1pt; border-style: solid; width: 350px; min-height: 75px;
}
.mKlasse {
background: red; color: yellow; font-size: 24px;
}

```

In der CSS-Datei passiert nicht sonderlich viel. Es werden ein paar Eigenschaften für die gesamte Webseite und alle Elemente vom Typ *div* festgelegt. Von hauptsächlichem Interesse ist die Klasse *.mKlasse*. Diese soll beim Laden der folgenden Webseite noch nicht verwendet werden, sondern erst dynamisch bei einer Anwenderaktion zugewiesen werden. Und dazu kommen JavaScript und jQuery zum Einsatz (*kap2_2.html*):

Listing 2.4 Verändern der verwendeten CSS-Klasse

```

...
<script type="text/javascript">
$(document).ready(function () {
$("#a").click(function () {
$("#c").addClass("mKlasse");
});
$("#b").click(function () {
$("#c").removeClass("mKlasse");
});
});
</script>
</head>
<body>
<h1>Veränderung von Style Sheets mit jQuery</h1>
<button id="a">CSS-Klasse hinzufügen</button>
<button id="b">CSS-Klasse entfernen</button><hr />
<div id="c">Niemand ist weiter von der Wahrheit entfernt als derjenige,
der alle Antworten weiß.</div><hr />
<div id="div1">
Vom Wahrsagen lässt sich wohl leben, aber nicht vom Wahrheit sagen.
</div>
</body>
</html>

```

In dem Beispiel sehen Sie unterhalb einer Überschrift zwei Schaltflächen und zwei Texte jeweils innerhalb eines DIV-Bereichs, der mit einer Trennlinie abgegrenzt wird. Das ist pures HTML. Im Head finden Sie wieder die Verknüpfung mit der CSS-Datei (nicht abgedruckt).