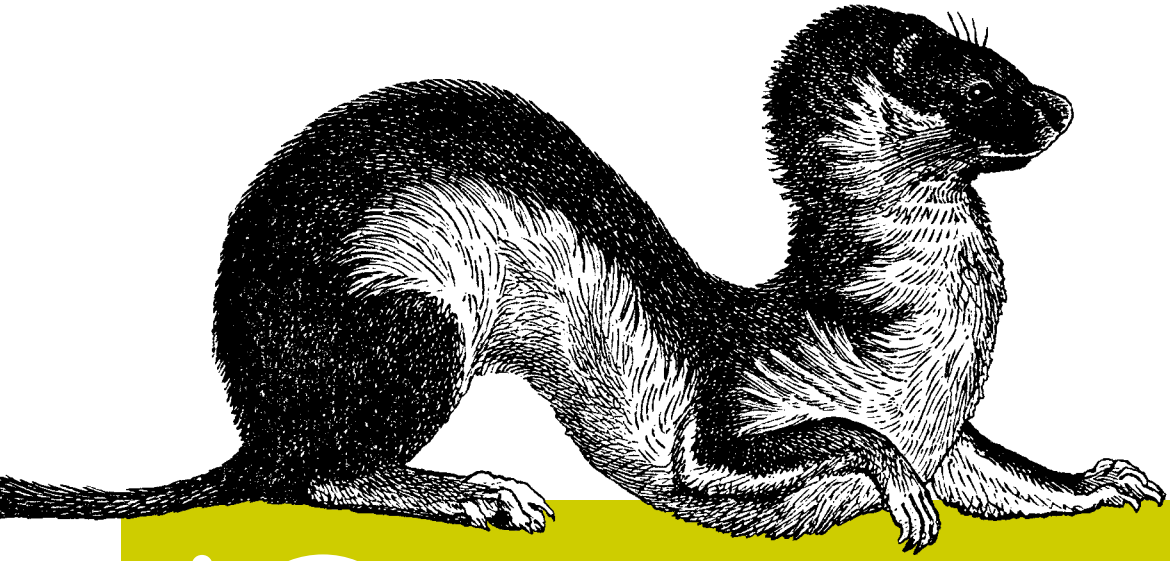
Lösungen für jQuery-Entwickler



jQuery Kochbuch

jQuery Community Experten

Deutsche Übersetzung von Thomas Demmig

O'REILLY®

Inhalt

KAPITEL 1

Grundlagen von jQuery

Cody Lindley

1.0 Einleitung

Da Sie zu einem Kochbuch über jQuery gegriffen haben, gehen die Autoren dieses Buches davon aus, dass Sie eine vage Idee davon haben, was jQuery genau ist und was es tut. Kochbücher sind im Allgemeinen für ein Publikum geschrieben, das schon gewisse Grundlagen im Bereich des betreffenden Themas besitzt, die ausgebaut werden sollen. Daher wird das Format Rezept – Lösung – Diskussion dazu genutzt, Ihnen schnell Lösungen für häufiger vorkommende Probleme anbieten zu können. Sind Sie aber ein jQuery-Anfänger, dann werfen Sie das Buch nicht gleich wütend gegen eine Wand, während Sie wilde Verwünschungen gegen uns aussprechen. Wir haben dieses Kapitel extra für Sie geschrieben.

Falls Sie ein wenig Auffrischung oder überhaupt eine Starthilfe benötigen, weil Sie über jQuery nicht viel oder gar nichts wissen, vermittelt Ihnen dieses erste Kapitel (die weiteren Kapitel gehen davon aus, dass Sie die Grundlagen beherrschen) einen Einstieg in jQuery. Haben Sie noch so gar kein Wissen über JavaScript und das DOM,dann sollten Sie sich vielleicht überlegen, ob es überhaupt sinnvoll ist, mit jQuery arbeiten zu wollen, *ohne* die Sprache JavaScript und ihr Zusammenwirken mit dem DOM zumindest in Grundzügen zu kennen. In diesem Fall empfehle ich ein Buch über DOM und JavaScript, bevor Sie sich wieder mit jQuery beschäftigen. Mein Tipp ist da *JavaScript - Das umfassende Referenzwerk (http://www.oreilly.de/catalog/jscript5ger)* von David Flanagan (O'Reilly) als Einführung. Aber lassen Sie sich nicht davon abhalten, jQuery noch vor dem DOM und JavaScript zu erlernen. Viele haben sich gerade auf diesem Weg Wissen über diese Technologien angeeignet. Es mag zwar nicht der ideale Weg sein, aber er kann funktionieren.

So, dann wollen wir uns mal eine formale Definition von jQuery und eine kurze Beschreibung ihrer Funktionalität anschauen:

> jQuery ist eine Open Source JavaScript-Bibliothek, die die Interaktionen zwischen einem HTML-Dokument oder genauer dem Document Object Model (DOM) und JavaScript vereinfacht.

Oder einfacher gesagt – und für die klassischen JavaScript-Hacker – jQuery macht die Arbeit mit Dynamic HTML (DHTML) total einfach. Insbesondere erleichtert jQuery das Durchlaufen und Bearbeiten des HTML-Dokuments, den Umgang mit Browser-Events, DOM-Animationen, Ajax-Interaktionen und JavaScript-Entwicklung für verschiedene Browser.

Mit der formalen Beschreibung von jQuery im Hinterkopf wollen wir uns nun anschauen, warum Sie jQuery eventuell nutzen sollten.

Warum jQuery?

Es mag ein bisschen seltsam erscheinen, die Vorzüge von jQuery in diesem Kochbuch aufzuzählen, denn vermutlich kennen Sie sie bereits.

Ich renne also vielleicht offene Türen ein, aber trotzdem sollten wir uns kurz anschauen, warum sich ein Entwickler für jQuery entscheiden könnte. Ich möchte damit auch Ihr Grundlagenwissen über jQuery verbessern, indem ich erst das »Warum« erkläre, um dann zum »Wie« zu kommen.

Ich will jQuery nicht mit der Konkurrenz vergleichen, um die Bedeutung von jQuery zu unterstreichen. Denn ich glaube nicht, dass es einen direkten Konkurrenten gibt. Zudem bin ich der Meinung, dass die einzige Bibliothek, die sowohl die Bedürfnisse der Designer als auch die der Programmierer erfüllt, jQuery ist. In diesem Kontext steht jQuery einfach für sich alleine.

Ich gehe fest davon aus, dass all die bekannten JavaScript-Bibliotheken und -Frameworks ihre Nische und ihren Wert haben. Es wäre verrückt, alle miteinander vergleichen zu wollen, aber es wird doch immer wieder gemacht. Ich war sogar selbst dran beteiligt. Aber nachdem ich mir viele Gedanken dazu gemacht habe, glaube ich ehrlich daran, dass alle JavaScript-Bibliotheken ihren Zweck haben. Es hängt nur mehr davon ab, wer sie nutzt und wie sie genutzt werden, als davon, was sie wirklich können. Zudem scheint es so, dass die kleinen Unterschiede zwischen JavaScript-Bibliotheken in Bezug auf die größeren Ziele der JavaScript-Entwicklung oft unwichtig sind. Ohne jetzt noch philosophischer zu werden, will ich einfach mal eine Liste von Eigenschaften präsentieren, die jQuery zum Vorteil gereichen:

- Es ist Open Source und das Projekt ist unter einer MIT- und einer GNU General Public License (GPL)-Lizenz lizensiert. Es ist frei, und das sogar mehrfach!
- Es ist klein (18 KB minifiziert) und gezipped (114 KB unkomprimiert).
- Es ist unglaublich beliebt, hat also viele Anwender und ausreichend Beitragende als Entwickler und Botschafter.
- Es normalisiert die Unterschiede zwischen Web-Browsern, so dass Sie es nicht tun müssen.
- Es hat einen kleinen Footprint mit einer einfachen, aber doch klugen Plugin-Architektur.
- Es gibt ein umfangreiches Repository mit Plugins (*http://plugins.jquery.com/*), das seit der Veröffentlichung von jQuery immer weiter gewachsen ist.

- Die API ist vollständig dokumentiert – sogar mit Code-Beispielen, was in der Welt der JavaScript-Bibliotheken ein echter Luxus ist. Nun, eigentlich war es jahrelang schon ein Luxus, dass es überhaupt eine Dokumentation gibt.
- Es ist freundlich, denn es stellt Möglichkeiten bereit, Konflikte mit anderen JavaScript-Bibliotheken zu vermeiden.
- Die Unterstützung durch die Community ist wirklich hilfreich. Es gibt eine Reihe von Mailing-Listen, IRC-Channels und unglaublich viele Tutorials, Artikel und Blog-Posts.
- Es wird offen entwickelt. Jeder kann zu Bugfixes, Verbesserungen und der Entwicklung an sich beitragen.
- Die Entwicklung geht stetig weiter, das Entwicklungs-Team hat keine Scheu, neue Releases zu veröffentlichen.
- Die Nutzung durch große Firmen sorgt für Langlebigkeit und Stabilität (zum Beispiel Microsoft, Dell, Bank of America, Digg, CBS, Netflix).
- Es berücksichtigt Spezifikationen des W3C, bevor es die Browser tun. Zum Beispiel unterstützt jQuery einen Großteil der CSS3-Selektoren.
- Es ist für die Entwicklung auf modernen Browsern getestet und optimiert (Chrome 1, Chrome Nightly, IE 6, IE 7, IE 8, Opera 9.6, Safari 3.2, WebKit Nightly, Firefox 2, Firefox 3, Firefox Nightly).
- Sowohl Designer als auch Programmierer können sehr viel aus jQuery herausholen.
- Seine Eleganz, die Methodologien und die Philosophie beim Ändern der Art, wie in JavaScript geschrieben wird, werden langsam selbst ein Standard. Überlegen Sie nur einmal, wie viele andere Lösungen die Selektions- und Auswahl-Muster übernommen haben.
- Der nicht erklärliche Nebeneffekt des guten Gefühls beim Programmieren ist ansteckend und lässt sich auch nicht vermeiden. Selbst Kritiker finden Teile von jQuery sehr gut.
- Die Dokumentation gibt es in unterschiedlichsten Facetten (zum Beispiel durch den API Browser, Dashboard Apps und Cheat Sheets), einschließlich eines Offline-API-Browsers (AIR Application).
- Es ist absichtlich so gestaltet, dass ordentliche JavaScript-Codingpraktiken gefördert werden.
- Es ist im Kern eine JavaScript-Bibliothek geblieben (statt eines Frameworks), stellt aber gleichzeitig ein Partner-Projekt für UI-Widgets und die Anwendungs-Entwicklung bereit (jQuery UI).
- Die Lernkurve ist nicht allzu steil, weil jQuery auf Konzepten aufbaut, die die meisten Entwickler und Designer schon kennen (zum Beispiel CSS und HTML).

Meiner Meinung nach ist es keine einzelne Eigenschaft, sondern die Kombination aller dieser Punkte, die jQuery von den anderen Lösungen abhebt, Das gesamte jQuery-Paket ist als JavaScript-Tool einfach unerreicht.

Die Philosophie hinter jQuery

Die Philosophie von jQuery ist: »Schreibe weniger, schaffe mehr.« Diese Philosophie kann dabei in drei Konzepte aufgebrochen werden:

- Elemente finden (über CSS-Selektoren), mit denen dann etwas getan wird (über jQuery-Methoden)
- Mehrere jQuery-Methoden für ein Set aus Elementen verketten
- Den jQuery-Wrapper und die implizite Iteration nutzen

Wenn Sie Ihren eigenen jQuery-Code schreiben oder die in diesem Buch gefundenen Rezepte aufpeppen wollen, ist es unabdingbar, diese drei Konzepte im Detail verstanden zu haben. Lassen Sie uns diese unter die Lupe nehmen.

Elemente finden, mit denen dann etwas getan wird

Genauer gesagt – eine Menge von Elementen im DOM finden, mit der dann etwas getan wird. Lassen Sie uns zum Beispiel ein Szenario anschauen, bei dem Sie ein `<div>` vor dem Anwender verbergen, neue Inhalte dort hineinladen, ein Attribut des `<div>` ändern und schließlich das verborgene `<div>` wieder sichtbar machen wollen.

Diese Kombination an Aufgaben, umgewandelt in jQuery-Code, würde in etwa so aussehen:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html>
<head>
<script type="text/JavaScript"
src="http://ajax.googleapis.com/ajax/libs/jquery/1.3.2/jquery.min.js"></script>
</head>
<body>
<div>alter Inhalt</div>
<script>

//alle divs auf der Seite verbergen
jQuery('div').hide();

//Text in allen divs aktualisieren
jQuery('div').text('neuer Inhalt');

//Klassen-Attribut mit dem Wert updatedContent bei allen divs hinzufügen
jQuery('div').addClass("updatedContent");

//alle divs auf der Seite anzeigen
jQuery('div').show();

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

Lassen Sie uns diese vier jQuery-Anweisungen durchgehen:

- Die -Elemente auf der Seite verbergen, so dass sie der Anwender nicht mehr sieht.
- Den Text in den verborgenen durch neuen Text (neuer Inhalt) ersetzen.
- Das -Element durch ein neues Attribut (class) mit einem Wert (updatedContent) ergänzen.
- Das -Elements auf der Seite anzeigen, so dass es der Benutzer wieder zu sehen bekommt.

Wenn Ihnen der jQuery-Code jetzt noch etwas mystisch vorkommt, ist das schon in Ordnung. Wir werden uns im ersten Rezept dieses Kapitels die Grundlagen anschauen. Was Sie aber aus diesem Code-Beispiel mitnehmen sollten, ist das jQuery-Konzept »Elemente finden, mit denen etwas getan wird«. In diesem Beispiel haben wir alle -Elemente in der HTML-Seite mit Hilfe der jQuery-Funktion (jQuery()) gefunden und dann jQuery-Methoden genutzt, um mit dieser Menge etwas zu tun (zum Beispiel hide(), text(), addClass(), show()).

Verketten

jQuery ist so aufgebaut, dass Sie dessen Methoden verketten können. Warum sollten Sie nicht zum Beispiel ein Element finden und dann nacheinander Operationen für dieses anwenden können? Unser vorheriges Codebeispiel, das Konzept »Elemente finden, mit denen etwas getan wird« könnte durch das Verketten zu einer einzigen JavaScript-Anweisung kombiniert werden.

Dieser Code:

```
//alle divs auf der Seite verbergen
jQuery('div').hide();

//Text in allen divs aktualisieren
jQuery('div').text('neuer Inhalt');

//Klassen-Attribut mit dem Wert updatedContent bei allen divs hinzufügen
jQuery('div').addClass("updatedContent");

//alle divs auf der Seite anzeigen
jQuery('div').show();
```

kann durch Verketten auch so aussehen:

```
jQuery('div').hide().text('neuer Inhalt').addClass("updatedContent").show();
```

oder mit Einrückung und Zeilenumbruch:

```
jQuery('div')
    .hide()
    .text('neuer Inhalt')
    .addClass("updatedContent")
    .show();
```

Index

Symbole

A

B

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

Z