```
$(document).ready(function() {
  var rheinwerk = $('.rheinwerk');
  rheinwerk.on('click',function(event) {
    event.preventDefault();
    $(this).addClass('verlag');
    if (rheinwerk.hasClass('idee')) {
      rheinwerk
        .find('#autor')
        .delay(2017)
        .fadeIn(2017);
    }
```

Ajax, Selekto
Data, UI, Effe
DOM-Manipula
Traversing, C
Widget Factor
s, Siz

Bongers · Vollendorf · Lange

jQuery 3

Das umfassende Handbuch

- Grundlagen, Einsatz, Praxisbeispiele
- Plugins nutzen und erstellen, jQuery UI, Ajax
- Animationen, Tabellen, Bildergalerien, Formulare

Alle Codebeispiele zum Download

Rheinwerk Computing

A HTML und CSS

B JavaScript und DOM

Stichwortverzeichnis

Vorwort

Der größte Nutzen von jQuery liegt in einer Programmierschnittstelle, die einfach zu verstehen und einfach zu benutzen ist. Darüber hinaus ist die Beschäftigung mit jQuery ein wunderbarer Einstieg in die Programmiersprache JavaScript überhaupt.

Sie wollen Webauftritte planen, gestalten, entwickeln und dem Benutzer eine aufregende, moderne Oberfläche bieten, die Informationen im Hintergrund lädt, ohne die gesamte Website neu zu laden? Sie wollen Ihrer Site eine dynamische, hierarchisch aufgebaute Navigation hinzufügen, die zuverlässig funktioniert und die einzelne Ebenen per Mausbefehl öffnet? Eine benutzerfreundliche Bildergalerie in Ihre Website integrieren, die auf Befehl eine vergrößerte Ansicht eines Vorschaubilds lädt und einblendet? Daten in eine bereits aufgerufene Webseite laden und als Tabelle anzeigen? Ein Shopsystem aufbauen und die Formulareingaben validieren sowie in Abhängigkeit von Benutzereingaben zusätzliche Formularfelder sperren oder anzeigen? Sie wollen sich für all das aber nicht in kryptische Programmierarbeiten vertiefen müssen?

Dann ist *jQuery* Ihr Freund. jQuery hilft Ihnen, fortschrittliche Benutzerschnittstellen zu schaffen und aufregende, animierte, interaktive Websites zu entwerfen. jQuery vereinfacht den Entwicklungsprozess, nimmt Ihnen lästige Routinearbeiten beim Programmieren ab und eröffnet Ihnen Möglichkeiten, von denen Sie gar nicht gewusst haben, dass ein Browser sie beherrscht.

jQuery steuert dabei keine neuen JavaScript-Funktionalitäten bei, sondern vereinfacht lediglich den Umgang mit allen und besonders den schwer zu nutzenden JavaScript-Methoden. jQuery wird so zum Leitfaden, der den Zugang zu JavaScript beinahe zum Kinderspiel macht.

jQuery – Das umfassende Handbuch (4. Auflage)

Mit der jQuery-Version 3.x tritt in der jQuery-Welt eine weitere Konsolidierung und Anpassung an die aktuelle Browserwelt ein. Letzte aus alten Versionen stammende Inkonsistenzen werden beseitigt, und das Framework wird, unter Beibehaltung des Funktionsumfangs und der API, durch das Abwerfen von Altlasten nochmals erheblich verschlankt. Für Anwendungen, die auf älteren jQuery-Versionen basieren, stehen, wie gehabt, Migrationstools zur Verfügung.

Allen Anfechtungen durch Hypes und Modeparadigmen in der Webprogrammierung zum Trotz steht jQuery unangefochten als das meistverbreitete JavaScript-Framework da. Seine API, die Formulierung und der Funktionsumfang seiner Methoden gelten als beispielhaft. Nicht grundlos bildet jQuery das Fundament vieler weiterer beliebter Anwendungen, wie Bootstrap oder Angular, wobei Ersteres auf jQuery direkt aufsetzt, Letzteres eine (abgespeckte) Version von jQuery fest integriert, aber auch hervorragend mit dem »ausgewachsenen« jQuery harmoniert.

Die clevere Grundauslegung von jQuery sorgt zudem nicht nur für Erweiterbarkeit über Plugins (die Oberflächen-Widgets von jQuery UI sind ein Beispiel hierfür), sondern sorgt außerdem dafür, dass jQuery nicht nur »herkömmlich« (synchron), sondern auch asynchron (mittels RequireJS) oder serverseitig (als NodeJS-Anwendung) genutzt werden kann.

Die aktuelle Bedeutung von jQuery

jQuery ist noch immer das am meisten eingesetzte Javascript-Framework.

Aktuell setzen fast drei Viertel aller Websites, bei denen mit JavaScript gearbeitet wird, jQuery ein. Betrachtet man die bei diesen Sites eingesetzten Frameworks, ist der Marktanteil noch höher: Hier geben 96% aller Websites jQuery den Vorzug (im Vergleich: AngularJS liegt bei nur 0,5%!). Andere, früher in diesem Segment bedeutende Frameworks wie MooTools, Dojo oder Ext JS sind marginalisiert oder werden, wie YUI (Yahoo), nicht mehr weiterentwickelt.

https://w3techs.com/technologies/overview/javascript_library/all

Durch seine Erweiterungsschnittstelle ist jQuery auch das Tool der Wahl als Basis für Eigenentwicklungen. Die meisten Programmierer von JavaScript-Plugins (und auch die wichtigsten Plugins) setzen auf jQuery auf. Weiterhin punktet jQuery mit universeller Kompatibilität über alle Browserplattformen, was durch den prüfenden Blick einer breiten Community unterstützender Entwickler gewährleistet bleibt.

Durch die mächtigen neuen nativen DOM-Funktionen, wie die Query-Selektoren, oder die in allen standardkonformen Browsern unterstützten CSS-Transitions scheint jQuery auf den ersten Blick in seinen Hauptdomänen an Boden zu verlieren. So gibt es Autoren, die beim Vergleich von jQuery und nativen JavaScript-Anweisungen zu dem Schluss kommen, dass man Ergebnisse mit ein

paar Zeilen mehr auch ohne jQuery realisieren kann:
http://youmightnotneedjquery.com

Die Überlegung ist durchaus legitim. Die einen Entwickler halten die Auswirkungen jedoch für unerheblich und argumentieren, dass die mächtigsten Features von jQuery es in jedem Fall rechtfertigen, das Framework zu laden. Es sei kein merklicher Performancegewinn durch native Scriptanweisungen zu erwarten. Andere Entwickler argumentieren, weniger sei mehr, der aktuelle Stand des JavaScript-Sprachschatzes sei so umfangreich, dass man auf natives (Vanilla-)JavaScript setzen kann und sich den Overhead von jQuery sparen kann. So bieten beispielsweise die modernen Browser, die auf den HTML5-Standard setzen, die so genannte Selector-API, die Abfragen am DOM-Baum ähnlich handhabt wie die mächtige Sizzle-Engine von jQuery. Diesen und andere Aspekte werden wir in einer Neuauflage diskutieren: Ab wann lohnt sich der Einsatz von jQuery? Und wir werden die Frage erörtern, ob man bei bestimmten Projekten, die keine populären jQuery-Plugins benötigen, auch mal auf jQuery verzichten kann. Sie können dieser Frage auch im Praxisteil nachgehen, und Sie werden lernen, abzuwägen, wann Sie jQuery einsetzen können und wann Sie auf natives JavaScript zurückgreifen können.

Was in der aktuellen Situation noch immer für jQuery spricht, ist die Verlässlichkeit im Umgang mit Kompatibilitätsproblemen und deren »Normalisierung«. Der Coding-Style, den jQuery ermöglicht, hat sich in vielen Fällen als Standardvorgehen etabliert und ist mit nativer Programmierung nicht nachvollziehbar.

Die Wirkung von jQuery oder jQuery-artigen Frameworks und ihrem Coding-Style strahlt auch auf Projekte aus, die scheinbar andere Anforderungen haben. Man denke an AngularJS, das eine eingebaute Lightversion (jQLite) von jQuery besitzt (eine Angular-Eigenentwicklung), beim zusätzlichen Laden von jQuery (dem Original) jedoch dieses einbindet. So bildet AngularJS mit jQuery ein hervorragendes Team.

Auch das bekannte CSS-Framework Twitter Bootstrap baut für seine Oberflächen-Widgets und JavaScript-Funktionen auf jQuery und stimmt deren Bedienung auf den jQuery-Coding-Style ab.

Durch seine Allgegenwart hat jQuery den sensationellen Touch seiner Anfangsjahre zwar verloren. Als solides und vertrautes Werkzeug für Entwickler leistet es derzeit und in der vorhersehbaren Zukunft weiter seine Dienste und spielt seine Stärken auch in Zusammenarbeit mit anderen Frameworks aus.