

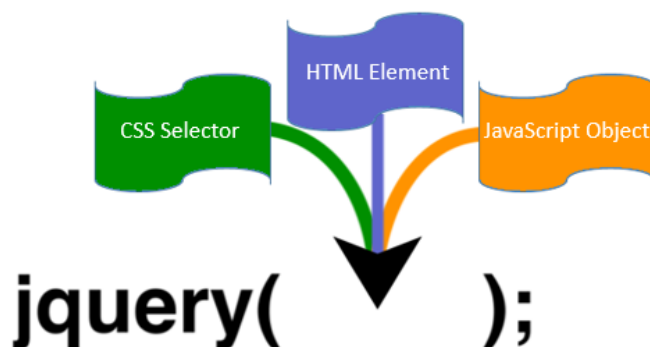
## jQuery – eine weit verbreitete JavaScript Bibliothek

jQuery ist eine JavaScript-Bibliothek, die vom amerikanischen Software-Entwickler John Resig entwickelt und 2006 unter der freien MIT-Lizenz veröffentlicht



wurde. Die Bibliothek stellt umfangreiche Optionen zur HTML- und CSS-Manipulation, zur Steuerung von Events sowie zur Beschleunigung von Ajax-Funktionalitäten bereit. Diese lassen sich unkompliziert nutzen, indem jQuery in den HTML-Quelltext des jeweiligen Webprojekts eingebunden und mithilfe von Funktionen gesteuert wird. jQuery ist in vielen Content-Management-Systemen und Webframeworks wie Joomla oder WordPress bereits integriert und besticht nicht nur durch den großen Funktionsumfang, sondern auch durch die große Community.

### Einsatzzweck von jQuery



jQuery vereinfacht die Programmierung mit der dynamischen Skriptsprache JavaScript erheblich. Die gesamte jQuery-Bibliothek besteht dabei aus einer einzigen JavaScript-Datei, die die grundlegenden DOM-, Ajax-, Ereignis- und Effekt-Funktionen enthält. Damit stellt die Bibliothek eine umfangreiche

Sammlung von Programmteilen dar, mit deren Hilfe die Elemente von Webprojekten bearbeitet werden können. So wählen Sie beispielsweise Objekte aus und verändern deren Aussehen (Farbe, Position etc.), was zwar prinzipiell auch mit JavaScript (durch herkömmlichen Zugriff auf das Document Object Model) möglich, aber deutlich aufwändiger zu realisieren ist. Ferner können Sie mit jQuery noch gezielter auf Aktionen Ihrer Nutzer reagieren – dank ereignisgesteuerter Programmierung der Seitenelemente. Die User lösen zuvor definierte Ereignisse per Mauszeiger oder durch Texteingabe aus und bekommen die jeweiligen Inhalte oder Animationen präsentiert. Auch grafische Effekte wie Texteinblendungen etc. sind schnell und mit nur einer einzigen Codezeile eingefügt.

Dass der Einsatz von jQuery stets für aktuelle Webprojekte von Interesse ist, liegt hauptsächlich an der einfachen Erweiterbarkeit der JavaScript-Bibliothek. Tausende Plug-ins, die die Programmierung weiter vereinfachen und jQuery noch mächtiger machen, stehen auf der offiziellen jQuery-Homepage – größtenteils kostenfrei – zum Download bereit. Ist die gewünschte Funktion weder im Standardumfang noch als Plug-in verfügbar, können sich erprobte Entwickler auch daran versuchen, eine eigene Erweiterung zu gestalten.

### Einsatz und Möglichkeiten von jQuery im Überblick

- Einfache Möglichkeiten, Elemente auf der Website (DOM) auszuwählen (um dann später damit Sachen zu machen wie Aussehen ändern, Animationen oder Aktionen).
- Was bisher in JavaScript über „*getElementById*“ und „*getElementsBy...*“ gemacht wurde, geht jetzt flott und problemlos über `$("#bereich1")` und `$(".farbe1")`. (Mehr zur jQuery-Syntax später)
- Das DOM manipulieren – sprich die ausgewählten Elemente entsprechend ändern (Farbe, Position etc.)
- Event-System: Je nachdem, was der Nutzer macht, kann man einfach mit jQuery darauf reagieren. Ein Mausklick und ein Bereich wird eingeblendet
- Animation und Effekte sind extrem einfach umsetzbar. Über einen Einzeiler können z.B. erweiterte Texte sanft eingeblendet werden usw.
- Es sind zahlreiche Plug-Ins verfügbar und so ist im Handumdrehen eine Bildergalerie eingebaut. Man muss also nicht immer wieder neu das Rad erfinden.

## Genereller Aufbau der jQuery Syntax

Mit jQuery-Code können Sie auf die HTML-Elemente Ihres Webprojekts zugreifen.

Diese HTML-Elemente ...

- ... wählen Sie über einen Selektor aus, der sich syntaktisch an CSS-Selektoren orientiert.
- Darauf folgt in der Regel eine Aktion, die beschreibt, auf welche Art und Weise das ausgewählte Element verändert werden soll.
- Selektor und Aktion wird zusätzlich ein Dollarzeichen (\$) vorangestellt, um den Code als jQuery-Funktion auszuzeichnen. Dies geschieht, um Verwechslungen bei der Verwendung verschiedener Bibliotheken zu vermeiden. `$()` dient dabei als Abkürzung für die `jQuery()`-Funktion.

Die Basis-Syntax lautet also:

```
$(Selektor).Aktion()
```

## Das Document-Ready-Event in jQuery

Eine unverzichtbare Codezeile, die jeglichen jQuery-Auszeichnungen in einem HTML-Dokument voranstehen sollte, ist das Document-Ready-Event.

Dieser Code stellt sicher, dass alle darin enthaltenen jQuery-Kommandos erst ausgeführt werden, nachdem sämtliche HTML-Elemente geladen worden sind. Einerseits werden so Fehlermeldungen vermieden, wenn z. B. ein Element versteckt werden soll, das bisher noch gar nicht angezeigt wird. Andererseits ermöglicht das Document-Ready-Event auch, den Code in den *<head>*-Bereich einzufügen. Der syntaktische Aufbau der angesprochenen Codezeile entspricht dem folgenden Muster:

```
<script>
$(document).ready(function(){
    //weiterer jQuery-Code
});
</script>
```

## Einbinden der jQuery-Bibliothek in die eigene Webseite

Bevor wir jQuery nutzen können, müssen wir die jQuery-Bibliothek in unsere Website einbinden.

Es gibt zwei Arten um jQuery einzubinden. Es können alle benötigten Dateien von jQuery heruntergeladen und auf dem eigenen Webspace bereitgestellt werden (<https://jquery.com>) oder es wird ein Link in der eigenen HTML-Seite gesetzt und dann jQuery genutzt. (=Einbindung über ein CDN).

### Option 1:

**Syntax zur lokalen Einbindung (nach Download) (exemplarisch)**

```
<head>
<script src="pfad-zur-jquery/jquery.js"></script>
</head>
```

### Option 2:

**Einbindung über ein CDN (in diesem Beispiel über Google Hosted Libraries)**

<https://developers.google.com/speed/libraries?hl=de#jquery>

```
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.3/jquery.min.js"></script>
</head>
```

## Die wichtigsten jQuery-Selektoren

Die jQuery-Selektoren sind die wohl wichtigsten Bestandteile der JavaScript-Bibliothek. Mit ihnen passen Sie die verschiedenen Elemente Ihrer Website an. Dabei gibt es unterschiedliche Selektorenarten, die HTML-Einheiten nach verschiedenen Kriterien wie ID, Klasse, Typ etc. finden und auswählen.

### 1. Element-Selektor

Der häufig verwendete Element-Selektor ordnet HTML-Elementen die jeweiligen Aktionen anhand ihres Namens zu. So können Sie mit dem folgenden jQuery-Code z. B. definieren, dass alle `<p>`-Elemente – also alle Textblöcke – ausgeblendet werden, wenn der Website-Besucher auf einen Button klickt:

```
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("p").hide();
  });
});
</script>
```

### 2. id-Selektor

Ein weiterer wichtiger jQuery-Selektor ist der `#id`-Selektor. Mit ihm zeichnen Sie im HTML-Dokument ein einzelnes Element aus, das danach, beispielsweise mit JavaScript, ganz gezielt verändert werden kann, ähnlich wie es auch bei CSS-Manipulationen der Fall ist. Sollen nicht alle Textblöcke bei einem Buttonklick des Users ausgeblendet werden, sondern nur ein ganz bestimmter, fügen Sie diesem `<p>`-Element einen id-Wert hinzu (`<p id="testblock">`) und ergänzen den Element-Selektor im oberen Codebeispiel durch den `#id`-Selektor:

```
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("#testblock").hide();
  });
});
</script>
```

### 3. class-Selektor

Ein dritter einfacher Selektor ist der `.class`-Selektor, der zuvor definierte Klassen auswählt und ebenfalls nach der bekannten Methode eingesetzt wird.

#### 4. Weitere nützliche Selektoren

Selektor	Beschreibung
<code>\$("*")</code>	wählt alle Elemente aus
<code>\$(this)</code>	wählt das gegenwärtige Element aus
<code>\$("p:first")</code>	wählt das erste <code>&lt;p&gt;</code> -Element aus
<code>\$("ul li:first-child")</code>	wählt jeweils die erste Listenelemente <code>&lt;li&gt;</code> aller vorhandenen <code>&lt;ul&gt;</code> -Auflistungen
<code>\$("[href]")</code>	wählt alle Elemente mit einem href-Attribut aus
<code>\$("tr:even")</code>	wählt alle geraden Tabellenzeilen aus
<code>\$("tr:odd")</code>	wählt alle ungeraden Tabellenzeilen aus
<code>\$("p.intro")</code>	wählt alle <code>&lt;p&gt;</code> -Elemente mit der Klasse <code>class="intro"</code> aus

## Reaktion auf User-Events mit jQuery

Die Besucher Ihrer Website bzw. Nutzer Ihrer Webanwendung interagieren auf unterschiedliche Weise mit Ihrem Webprojekt – ob per Mausklick, Tastatureingabe, durch Ausfüllen eines Formulars oder das Vergrößern eines Fensters. Diese Ereignisse sind auch als DOM-Events bekannt und können in jQuery als Auslöser für Aktionen definiert werden.

Beispielsweise können Sie die Bewegung des Mauszeigers zum Auslöser für eine Aktion machen. Das erreicht man mit `mouseenter()` bzw. `mouseleave()`. Die erste Funktion reagiert darauf, dass der Nutzer mit dem Mauszeiger über das ausgewählte HTML-Element fährt, die zweite darauf, dass der Mauszeiger das Element wieder verlässt. Der folgende jQuery-Code bewirkt, dass der User über ein Dialogfenster benachrichtigt wird, wenn er einen Textblock mit dem Mauszeiger anvisiert:

```
<script>
$(document).ready(function(){
  $("p").mouseenter(function(){
    alert("Sie haben einen Textblock ausgewählt!");
  });
});
</script>
```

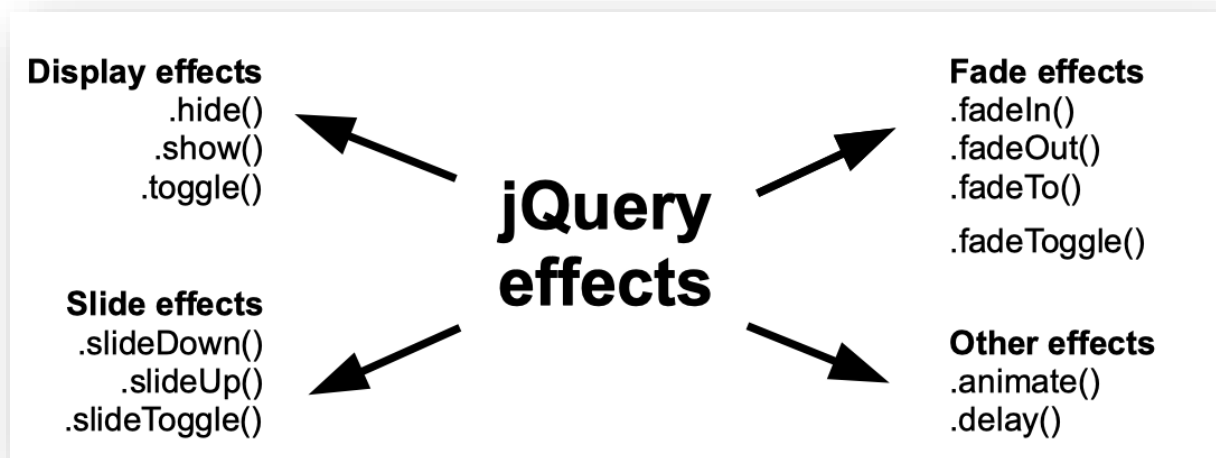
## Weitere Funktionen

Funktion	Beschreibung
<code>focus()</code>	wird ausgelöst, wenn ein Element in den Fokus genommen wird (per Mausklick oder Tab)
<code>blur()</code>	wird ausgelöst, wenn ein Element aus dem Fokus genommen wird
<code>keydown()</code>	wird ausgelöst, wenn eine Taste gedrückt und gehalten wird
<code>keyup()</code>	wird ausgelöst, wenn eine Taste losgelassen wird
<code>change()</code>	wird ausgelöst, wenn eine Tastatureingabe getätigt wurde oder eine Auswahl getroffen wurde
<code>scroll()</code>	wird ausgelöst, wenn in dem ausgewählten Element gescrollt wird
<code>select()</code>	wird ausgelöst, wenn Text in dem ausgewählten Formular-Eingabefeld markiert wird

## Effekt-Funktionen in jQuery

Ein Effekt wird normalerweise durch eine Benutzerinteraktion gestartet, die als jQuery-Ereignis bezeichnet wird. Beispielsweise können jQuery-Hover-Effekte ausgelöst werden, wenn der Cursor den Bereich eines angegebenen Elements betritt oder verlässt. Für andere Effekte ist möglicherweise ein **Klick oder ein anderes Ereignis erforderlich**.

In der folgenden Abbildung sehen Sie häufig verwendete Funktionen für jQuery-Effekte.



Dies sind die am häufigsten verwendeten jQuery-Effekte:

- **Show und hide:**  
lässt Sie ein bestimmtes Element **ein-** oder **ausblenden**, wenn eine definierte Aktion erkannt wird.
- **Fade**  
zeigt und verbirgt auch ein Element, tut dies jedoch durch **Erhöhen** oder **Verringern der Deckkraft**. Es kann auch so eingestellt werden, dass es ein bestimmter Prozentsatz der Deckkraft ausgeblendet wird, anstatt das Element vollständig auszublenden.
- **Slide**  
zeigt und blendet auch Elemente aus, fügt aber einen **vertikalen Gleiteffekt** hinzu.
- **Animate**  
wird verwendet, um **benutzerdefinierte Animationen** zu erstellen und sie auf Ihre HTML-DOM-Elemente anzuwenden.

## Ein erstes Programm mit jQuery

Der vorliegende HTML-Code (inkl. jQuery / JavaScript) bewirkt, dass auf dem Bildschirm zwei Absätze (inkl. Überschrift) zu sehen sind. Sobald „**Erster wichtiger Absatz**“ vom User geklickt wird, öffnet sich ein Popup mit der Hinweismeldung: „**Der erste Absatz wurde geklickt**“.

Code: (Datei: „jquery1.html“)

```
1 <!DOCTYPE html>
2 <html lang="de">
3 <head>
4   <meta charset="UTF-8">
5   <title>jQuery Beispiel: IDs ansprechen über jQuery</title>
6   <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.3/jquery.min.js"></script>
7 <script>
8
9
10 $(document).ready(function(){
11   $("#absatz1").click(function(){
12     alert("Der erste Absatz wurde geklickt");
13   });
14 });
15
16
17 </script>
18 </head>
19 <body>
20 <h1>jQuery Beispiel: Klassen ansprechen über jQuery</h1>
21 <p id="absatz1">Erster <b>wichtiger</b> Absatz</p>
22 <p id="absatz2">Zweiter Absatz</p>
23 </body>
24 </html>
```

Output:

### jQuery Beispiel: Klassen ansprechen über jQuery

Erster **wichtiger** Absatz  
Zweiter Absatz

Auf js.do wird Folgendes angezeigt:

Der erste Absatz wurde geklickt

Ok

### Arbeitsauftrag 1:



Programmieren Sie dieses Beispiel möglichst eigenständig auf Ihrem PC nach um sich mit der jQuery-Syntax vertraut zu machen.

### Arbeitsauftrag 2:



Erweitern Sie den Code so, dass der erste Absatz rot markiert wird, sobald das Pop-Up Fenster vom User geschlossen wird. Tipp: Nutzen Sie die CSS-Funktion.

`$(selector).css(property,value)`

Property = „color“, „font-weight“ etc. – Value = „red“, „bold“ etc.





**Folgende Fragen sollten Sie nun beantworten können. Testen Sie Ihr Grundlagenwissen zu jQuery indem Sie diese beantworten.**

1. jQuery verwendet CSS-Selektoren um Elemente auswählen zu können. Richtig oder Falsch?

---

2. Welches Zeichen verwendet jQuery als Abkürzung für jQuery?

---

3. Sehen Sie sich den folgenden Selektor an: \$("div"). Was wählt dieser innerhalb einer HTML-Seite aus?

---

4. Ist jQuery eine Bibliothek für Client-Scripting oder Server-Scripting?

---

5. Welche jQuery-Funktion wird verwendet, um ausgewählte Elemente auszublenden?

---

6. In welcher Skriptsprache ist jQuery geschrieben?

---

7. Welche jQuery-Funktion wird verwendet, um zu verhindern, dass Code ausgeführt wird, bevor das Dokument vollständig geladen ist?

---

8. Was wählt dieser Selektor aus? \$("p#intro")

---

9. Welche Art von Funktionen enthält jQuery?

---