Doing Web Apps

jQuery 1

Autor: Rüdiger Marwein
Letzte Änderung: 2014-10-24
Version: 1.1

Dieses Dokument darf – mit Nennung des Autoren - frei vervielfältigt, verändert und weitergegeben werden.

Der Inhalt ist sorgfältig recherchiert, mit dem Dokument ist jedoch keinerlei Garantie auf Fehlerfreiheit gewährleistet.

Dieser Inhalt ist unter einem Creative Commons Namensnennung Lizenzvertrag lizenziert. Um die Lizenz anzusehen, gehen Sie bitte zu http://creativecommons.org/licenses/by/2.0/de/ oder schicken Sie einen Brief an Creative Commons, 559 Nathan Abbott Way, Stanford, California 94305, USA.

Inhaltsverzeichnis

1.Was ist jQuery?.....3
2.Arbeiten im DOM mit jQuery.....3
3.HTML Angaben mit jQuery verändern.....3
4.CSS mit jQuery verändern.....4
5.Programmcode ausführen, sobald die Seite fertig geladen ist.....4
6.Verkettung.....5

1.Was ist jQuery?

jQuery ist eine Bibliothek für JavaScript, die man als Entwickler zuschalten kann. Sie vereinfacht insbesondere das Arbeiten mit dem DOM. Des weiteren werden browser-spezifische Besonderheiten vereinheitlicht.

Die Bibliothek kann heruntergeladen werden von

http://jquery.com/download/

Dreh und Angelpunkt ist das jQuery Objekt, das als „jQuery“ oder „$“ zur Verfügung steht. Jedes DOM Element kann mit einem jQuery-Objekt ummantelt werden und kann dann sofort mit einheitlichen Methoden manipuliert werden. Browser-Unterschiede sind somit kein Problem.

Sie können die jQuery Bibliothek herunterladen, ablegen und lokal einbinden oder bspw den Google CDN für eine dezentrale Einbindung nutzen:

```
<script type="text/javascript"
  src="//ajax.googleapis.com/ajax/libs/jquery/2.1.1/jquery.min.js">
</script>
```

Die jQuery API Dokumentation ist die beste Quelle für eine vollständige Liste aller verfügbaren Funktionen.

http://api.jquery.com

2.Arbeiten im DOM mit jQuery

Einen Knoten anhand seiner ID laden (Ergebnis ist ein DOM Knoten)

```
var elem = jQuery('#x');
```

Alle Knoten derselben Klasse laden (Ergebnis ist ein Array aus DOM Knoten)

```
var elem = jQuery('.x');
```

Alle Knoten eines bestimmten Tags laden (Ergebnis ist ein Array aus DOM Knoten)

```
var elem = jQuery('p');
```

Knoten mit CSS Selektoren laden (Ergebnis ist ein Array aus DOM Knoten)

```
var elem = jQuery('p.x');
var elem = jQuery('#id p span.x');
```

Ein DOM Objekt mit dem jQuery Objekt ummanteln

```
var elem = jQuery(document.body);
var elem = jQuery("<div />"); // neues Element ohne Position.
```

DOM Elemente innerhalb eines anderen DOM-Elements laden (Ergebnis ist ein Array aus DOM Knoten)

```
jQuery('#x').find('p');
```

DOM Operationen auf alles außer IDs sind aufwändig. Verwendet man ein und den selben Selektor mehrmals, speichert man das Ergebnis in einer Variablen zwischen.

```
var paragraphs = jQuery('p.news');
paragraphs.css('color', 'green');
paragraphs.css('margin-bottom', '12px');
```

Hat man einen DOM Knoten mit jQuery selektiert, kann man die jQuery-Funktionen anwenden um Manipulationen durchzuführen oder Werte auszulesen.

3.HTML Angaben mit jQuery verändern

```
var $knoten = jQuery('#knoten');
```

HTML eines Knoten auslesen

```
var inhalt = $knoten.html();
```

Einem Knoten HTML hinzufügen bzw ersetzen

```
$knoten.html('<strong>Inhalt ausgetauscht</strong>');
```

Eine Klasse hinzufügen

```
$knoten.addClass('highlight');
```

Eine Klasse entfernen

```
$knoten.removeClass('highlight');
```

Ein Attribut auslesen

```
var titleAttribute = $knoten.attr('title');
```

Ein Attribut setzen / ändern

```
$knoten.attr('title', 'Anderer Titel');
```

Mehrere Attribute auf einmal ändern

```
$knoten.attr( { title: 'Anderer Titel', lang: 'de' } );
```

4.CSS mit jQuery verändern

```
var $knoten = $('#knoten');
```

Eine CSS-Eigenschaft auslesen

```
$knoten.css('background-color');
```

Eine CSS-Eigenschaft setzen

```
$knoten.css('background-color', '#dde');
```

Mehrere CSS-Eigenschaften setzen

```
$knoten.css( { background-color: '#dde', display: 'block' } );
```

5.Programmcode ausführen, sobald die Seite fertig geladen ist.

DOM Manipulationen können erst sicher durchgeführt werden, wenn die Seite vollständig geladen ist; nach dem „domready“-Event.

Die nachfolgenden 2 Varianten, um dies zu tun, sind äquivalent.

Es wird eine Funktion an den ready-Event geheftet, die ausgeführt wird, sobald die Seite fertig geladen ist. Diese Funktion erhält das jQuery-Objekt / die jQuery-Funktion als Parameter mit dem Name „$“ (Dollar). Somit kann innerhalb dieser Funktion die Funktion „$“ verwendet werden in Stellvertretung für die Funktion „jQuery“.

```
jQuery(document).ready((function($) {
  // auszuführender Code
  $("p").addClass("text");
});

// ist identisch zu ...

jQuery(function($) {
  // auszuführender Code
  $("p").addClass("text");
});
```

6.Verkettung

Selektiert man mit jQuery ein Objekt , lassen sich sofort die Eigenschaften des Objekts verändern.

```
jQuery("#die-kennung").attr("title","Neuer Titel");
```

Jede Funktion, die an einem Objekt ausgeführt werden, liefern entweder das veränderte Objekt zurück oder das neu selektierte Objekt. Es können sofort weitere Funktionen ausgeführt werden.

```
jQuery("#die-kennung")          // Objekt #die-kennung
  .attr("title","Neuer Titel") // Objekt #die-kennung
  .css("color",“blue“)          // Objekt #die-kennung
  .find("p")                    // Objekt(-Liste) <p>
    .css("margin","20px");      // Objekt(-Liste) <p>
```

Ist das Ergebnis einer Selektion eine Liste von Elementen und man führt eine Aktion wie css() darauf aus, so wird die Funktion css() auf alle Elemente in der Liste angewendet.

Ist die Ergebnismenge einer Selektion leer und es werden darauf Funktionen angewendet, passiert einfach nichts.

Dieses Feature findet exzessive Verwendung im Vortrag des 9-jährigen Dmitri Gaskin zu jQuery: http://www.youtube.com/watch?v=8mwKq7_JlS8