



CODERS.BAY

JQUERY TEIL 2

VERSTECKEN UND EINBLENDEN EINES CONTAINERS

Die Funktion `$(function() { })` sorgt dafür, dass der danach folgender Code erst ausgeführt wird, wenn das Dokument geladen ist.

Durch `$('#beispiel')` wird das Element mit `id="beispiel"` ausgewählt. Danach folgt nach einem Punkt, was geschehen soll: Der Methodenaufruf `hide()` versteckt das Element. Nach einem weiteren Punkt steht ein weiterer Befehl, der angibt, was nach dem Verstecken geschehen soll. `fadeIn()` dient in jQuery dazu, ein Element langsam einzublenden. Die Angabe in Klammern bestimmt die Geschwindigkeit der Animation in Millisekunden.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <style>
      #beispiel {
        width: 400px;
        background-color: orange;
        color: black;
        font: 200% sans-serif;
        text-align: center;
        padding: 20px;
      }
    </style>
  </head>
  <body>
    <div id="beispiel">Guten Morgen</div>
    <script src="jquery.js"></script>
    <script>
      $(function() {
        $('#beispiel').hide().fadeIn(4000);
      });
    </script>

  </body>
</html>
```

Man kann in jQuery Elemente ganz komfortabel über die von CSS bekannten Selektoren auswählen

- Element nach Namen selektieren: `$('p')` wählt alle p-Elemente aus
- Elemente nach Klasse selektieren: `$('.meineKlasse')`
- Nachfahren eines Elementes selektieren: `$('nav a')`
- Elemente nach Attributen auswählen: `$('a[href$=doc]')` wählt alle a-Elemente aus, sofern sie ein href-Attribut haben, dessen Wert auf doc endet.
- auch möglich seit CSS3: `:nth-child()`, `:nth-child(even)`,.....
 - `:even` wählt gerade Kindelemente aus
 - `:odd` wählt ungerade aus
 - `:animated` wählt nur animierte Elemente aus
- Formularelemente
 - `:radio` selektiert Radiobuttons
 - `:submit` selektiert den Submit-Button

FORMATIERUNGEN ZUWEISEN UND ELEMENTINHALTE BEARBEITEN



Das HTML-Dokument beinhaltet zwei Überschriften und eine Liste. Nach der Einbindung von jQuery steht ein weiteres `script`-Element. Innerhalb von `$(function() { und });` wird die `h1`-Überschrift ausgewählt und mithilfe von `.css()` die Formatierung verändert. Übergeben werden die gewünschten Formatierungen in der Objektliteralsyntax. Das heißt, man schreibt innerhalb der runden Klammern von `css()` geschweifte Klammern, die auf- und zugehen müssen `{}`. Innerhalb dieser geschweiften Klammern wird den Eigenschaften ihr Wert zugewiesen, indem ein Doppelpunkt dazwischen angegeben wird. Mehrere Anweisungen werden durch ein Komma getrennt

Interessant ist das Setzen von Formatierungen mit jQuery dann, wenn diese Formatierung dynamisch als Reaktion auf eine Aktion des Nutzers erfolgen

In den Browserentwicklungstools kann man gut sehen, wie jQuery gewirkt hat. Dort erkennt man, dass jQuery die Formatierungen über `style`-Attribute ergänzt hat. Der Befehl `.css()` ist praktisch, um direkt CSS-Formatierungen durchzuführen, was im Fall dynamischer Zuweisungen oft gemacht wird.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Formatierung anpassen</title>
  </head>
  <body>
    <h1>Eine Überschrift</h1>
    <h2>Eine weitere Überschrift</h2>
    <ul>
      <li>Fenchel</li>
      <li>Minigruke</li>
      <li>Salat</li>
    </ul>
    <script src="jquery.js"></script>
    <script>
      $(function() {
        $('h1').css({
          'background-color' : '#aaa',
          'color' : 'white',
          'border' : '20px dashed black',
        });
        $('li:even').css('font-style', 'italic');
      });
    </script>

  </body>
</html>
```

Schöner ist es jedoch, die Formatierung im CSS zu belassen und statt dessen Klassen hinzuzufügen über `addClass()`

```
<style>
  .ueb {
    background-color: #aaa;
    color: white;
    border: 20px dashed black;
  }
  .bet {
    font-style: italic;
  }
</style>
```

```
$('#h1').addClass('ueb');
$('#li:even').removeClass('bet');
```

Parallel zu `addClass()` zum Ergänzen von Klassen gibt es `removeClass()`, das eine Klasse entfernt. Zusätzlich existiert `toggleClass()`, mit der eine Klasse hinzugefügt wird, wenn sie nicht vorhanden ist, oder entfernt, wenn sie bereits beim Element vorhanden ist.

Eine weiterer praktischer jQuery-Befehl ist `html()`. Diesen kann man nutzen, um HTML-Code in ein Element zu schreiben.

mit `alert()` wird eine Meldung ausgegeben, es ist ein klassischer JavaScript-Befehl für Meldungsfenster. Im Beispiel dient es dazu, das Skript anzuhalten, damit man den ursprünglichen Text des Absatzes sehen kann. Danach wird mit `$('.weg')` das Element mit dem Attribut `class="weg"` ausgewählt. Über `html()` wird ein neuer Text in dieses Element geschrieben

Übriges kann man `html()` nicht nur nutzen, um HTML-Code in Elemente zu schreiben. Man kann es auch zum Auslesen von HTML-Code einsetzen. Dazu braucht man nur den Parameter weglassen.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>HTML-Code schreiben</title>
    <style>
      body {
        font-size: 120%;
        font-family: sans-serif;
      }
      p {
        background-color: #ddd;
        padding: 20px;
      }
    </style>
  </head>
  <body>
    <p class="weg">Ich bin dann mal weg...</p>>
    <script src="jquery.js"></script>
    <script>
      $(function() {
        alert('Ein Klick bitte!');
        $('.weg').html('Und jetzt komme ich');
      });
    </script>
  </body>
</html>
```

Das Interessante an JavaScript und damit auch an jQuery ist, dass man bereits im Client auf die Aktionen des Benutzers reagieren kann - man kann etwas Bestimmtes tun, wenn ein Benutzer mit dem Mauszeiger über einen Bereich der Webseite fährt, einen Bereich verlässt, etwas anklickt, eine Option bei einem Auswahlfeld wählt oder Ähnliches. All das könnte man hingegen mit PHP nicht, denn PHP kann immer erst tätig werden, nachdem der Link geklickt ist oder nachdem das Formular abgesendet wurde.

Die Aktionen des Benutzers lösen in JavaScript sogenannte Ereignisse oder *Events* aus. Die Reaktion auf diese Ereignisse nennt man Eventhandling-

BEISPIEL

Wenn Inhalte wechselnd ein- und ausgeblendet werden sollen, kann auf `toggle()` zurückgegriffen werden.

Bei einem Klick auf den ersten Absatz soll der zweite entweder ein- oder ausgeblendet werden.

Zuerst soll das Element mit `id="inhalt"` versteckt werden.

Außerdem soll etwas mit dem ersten Absatz geschehen. Über `css()` wird die Eigenschaft `cursor` auf `pointer` gesetzt. Zusätzlich soll der Klick abgefangen werden.

Genauso wie das Klickereignis kann man auch mit anderen Events arbeiten: Mit `mouseover` kann abgefangen werden, dass der Besucher einer Webseite die Mause über einen Bereich bewegt, mit `mouseout` kann darauf reagiert werden, wenn jemand eine Bereich mit der Maus wider verlässt. `change` erlaubt es bestimmten Code ausführen zu lassen, wenn jemand bei einer Auswahlliste eine andere Option wählt.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Ein-Ausblenden</title>
    <style>
      body {
        font-size: 120%;
        font-family: sans-serif;
      }
      p {
        background-color: #ddd;
        padding: 20px;
        width: 300px;
      }
    </style>
  </head>
  <body>
    <p class="einaus">Bitte klicken</p>
    <p class="inhalt">
      Dieser Inhalt wird wechselnd ein-ausgeblendet
    </p>
    <script src="jquery.js"></script>
    <script>
      $('#inhalt').hide();
      $('#einaus').css('cursor', 'pointer')
        .on('click', function() {
          $('#inhalt').toggle();
        })
    </script>
  </body>
</html>
```




CODERS.BAY

QUELLE

**Quelle: Maurice, Florence. PHP 7 und MySQL: Ihr
praktischer Einstieg in die Programmierung
dynamischer Websites
(German Edition) dpunkt.verlag. Kindle-Version.**