

Einheit 8

jQuery

jQuery ist eine Bibliothek für JavaScript, die viele Funktionen zum Auswählen, Auslesen und Verändern von HTML-Elementen und ihren Eigenschaften bereitstellt. jQuery wird seit 2006 entwickelt und wird von sehr vielen Websites eingesetzt: 2014 waren es mehr als drei Viertel der 10.000 meistbesuchten Websites.

Einbindung in HTML

Die aktuelle Version von jQuery kann von der Website des Projekts heruntergeladen werden und muss in das Verzeichnis, in dem die Website entwickelt wird kopiert werden.

```
<script type="text/javascript" src="scripts/libaries/
jquery.js"></script>
```

JavaScript-Bibliotheken müssen wie ganz normales JavaScript-Dateien in die HTML-Datei eingebunden. Wichtig ist dass die Bibliothek über den eigenen JavaScript-Dateien, in denen ihre Funktionen verwendet werden, steht.

Selektion

```
$(selector).action().action();
```

Ein Aufruf in jQuery beginnt immer mit einer Selektion. Darauf folgt nach einem Punkt ein Funktionsaufruf. Mehrere Funktionsaufrufe können aneinander gereiht werden.

```
$("header")
$(".tag")
$("main .box")
```

Eine Selektion besteht aus dem Dollarzeichen, dann wird in Klammern der Selektor geschrieben. Meistens soll ein HTML-Element ausgewählt werden. Dazu wird in Zollstrichen ein Selektor im Stile eines CSS-Selektors angegeben.

```
$(window)
$(document)
$(this)
```

In jQuery gibt es einige Selektoren, die besondere Objekte auswählen. Mit window wird das Browserfenster ausgewählt, so dass dessen Größe o.ä. ausgelesen werden kann. document wählt die gesamte Seite aus, so dass zum Beispiel ermittelt werden kann, wie weit gescrollt wurde. this wählt immer das aktuelle Objekt aus, auf das sich eine action bezieht. Das kann sehr hilfreich sein, wenn verschiedene Funktionen ineinander verschachtelt werden.

Größe

Funktionen in jQuery können (teilweise) sowohl zum Auslesen von Werten, als auch zum Schreiben verwendet werden.

```
var width = $("main #box").width();
var height = $("main #box").height();
```

In diesem Codebeispiel wird die Höhe und die Breite des selektierten Elements in Variablen geschrieben.

```
var width = 200;
$("main #box").width(width)
```

In diesem Codebeispiel wird die Größe des Elements auf 200 gesetzt.

jQuery ignoriert dabei das in CSS eingestellte Box-Modell: .width() und .height() beziehen sich immer auf die Größe des Inhalts. Zum Auslesen der Größe der gesamten Box (Inhalt, padding, border-width) können die Funktionen .outerWidth(), bzw. .outerHeight() verwendet werden. Diese Funktionen können aber keine Größe setzen.

```
var height = $("main #box").outerHeight();
```

Position

```
var offsetTop = $("main #box").offset().top;  
var offsetLeft = $("main #box").offset().left;
```

jQuery kann mithilfe der Funktion `.offset()` die Position eines Elements auslesen und setzen. Anders als bei den Größenfunktionen gibt es hier allerdings nicht für jede Dimension eine eigene Funktion, sondern die Funktion gibt mehrere Werte in Form eines JavaScript-Objekts aus. Der gewünschte Wert kann durch anhängen der Seite angegeben werden (`.top`, `.right`, `.bottom`, `.left`).

Achtung: Es handelt sich hier nicht um Funktionen, deshalb sind keine Klammern nötig.

```
$("main #box").offset({top: 10, left: 20});
```

Damit Werte gesetzt werden können, muss der Funktion ein JavaScript-Objekt mit den Werten übergeben werden. Ein JavaScript-Objekt wird in geschweiften Klammern angegeben, dann folgen jeweils Eigenschaft und Wert, getrennt durch einen Doppelpunkt. Mehrere Eigenschaften sind mit Kommas getrennt.

.ready()

```
$(document).ready( function() {  
    // jQuery-Code  
});
```

Viele jQuery-Funktionen beziehen sich auf HTML-Elemente. JavaScript wird vom Browser ausgeführt, so bald es geladen ist – unter Umständen ist in diesem Moment die Seite aber noch gar nicht vollständig aufgebaut (die Elemente sind also noch nicht da). Das kann zu Fehlern führen. Es ist deshalb sinnvoll, allen jQuery-Code in die oben angegebene Funktion zu schreiben – er wird dann erst ausgeführt, wenn das document „ready“ ist, also alle Elemente aufgebaut sind.

Klassen

```
$("#main #box").addClass("active");  
$("#main #box").removeClass("active");  
$("#main #box").toggleClass("active");
```

Mit den Funktionen `.addClass()`, `.removeClass()` und `.toggleClass()` können HTML-Elementen Klassen hinzugefügt, entfernt, bzw. getoggelt werden (das bedeutet wenn sie nicht vorhanden sind werden sie hinzugefügt, wenn sie vorhanden sind entfernt). Die Klassen werden in dem Moment verändert, in dem die Funktion aufgerufen wird.

Mouseevents

Man kann Mouseevents an Elemente hängen, das bedeutet dass wenn über diesem Element das entsprechende Mouseevent aus geführt wird, wird der in der Funktion angegeben Code ausgeführt.

```
$("#main .box").click( function()    {  
    $(this).toggleClass("active");  
});
```

Dieser Beispielcode bewirkt, dass bei einem Klick auf eine Element mit der Klasse box die Klasse active getoggelt wird. Innerhalb der Funktion wird der Selektor `this` verwendet, damit die Klasse nicht beim Klicken auf ein Element mit der Klasse box allen Elementen mit der Klasse box hinzugefügt oder entfernt wird, sondern nur bei dem, dass das Event ausgelöst hat, also das, auf welches geklickt wurde.

Mögliche Mouseevents sind `.click()`, `.dblclick()`, `.mouseenter()`, `.mouseleave()`, `.mousedown()` und `.mouseup()`.

Achtung: Der auszuführende Code wird nicht direkt in die Klammern geschrieben, sondern als Funktion übergeben, deshalb steht in den Klammern `function() { //code }`.

.scroll()

Das Event `.scroll()` wird ausgelöst, wenn der Nutzer im selektierten Element scrollt.

```
$(document).scroll( function()      {  
    var scrollTop = $(this).scrollTop();  
    $("main .box").offset({top: scrollTop});  
});
```

In diesem Fall wird document, also die gesamte Seite selektiert. Wenn auf der gesamte Seite gescrollt wird, wird mithilfe der Funktion .scrollTop() ausgelesen, wie weit im Dokument bereits gescrollt wurde. Dieser Wert wird den Elementen mit Klasse als Abstand von oben gegeben, das Element bleibt also immer auf gleicher Höhe im Fenster (es ist fixed).

```
$("main .box").scroll( function()    {  
    var scrollTop = $(this).scrollTop();  
    $(this).offset({left: scrollTop/2});  
});
```

.scroll() kann auch auf einzelne Elemente angewendet werden. Das macht nur sinn, wenn diese Elemente die CSS-Eigenschaft overflow: scroll besitzen.