



JavaScript und JavaScript Frameworks

Erste Schritte mit jQuery



Übersicht und Zielstellung

- In diesem Abschnitt erfahren Sie:
 - wie Webseiten für jQuery vorbereitet werden,
 - wie die allgemeine Syntax von jQuery ist,
 - ein paar einfache Beispiele
 - was Chaining ist,
 - wie Elemente selektiert werden,
 - wie ein Button mit einer Funktion versehen wird
 - wie HTML-Elemente ein- und ausgeblendet werden



Start in das Thema

Einbindung von jQuery

CDN-Integration: Die einfachste Möglichkeit, jQuery in eine Anwendung einzubinden, ist die Verwendung des Content Delivery Networks (CDN). Fügen Sie dann den folgenden Code in den Kopfbereich Ihrer HTML-Datei ein::

```
<script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
```

Herunterladen von jQuery: Sie können auch eine lokale Kopie von jQuery herunterladen und in Ihre Anwendung einbinden. Fügen Sie dann den folgenden Code in den Kopfbereich Ihrer HTML-Datei ein:

```
<script src="pfad/jquery-3.6.0.min.js"></script>
```

Einbindung von jQuery

Wenn die Bibliothek von jQuery aus einem CDN geladen werden soll, ist es auch ratsam den Fall der Fälle abzusichern, falls das Repository nicht erreicht und somit nicht geladen werden kann:

```
<script type="text/javascript" src=" https://ajax.googleapis.com/ajax/libs/  
jquery/3.6.4/jquery.min.js ">  
</script>
```

Fallback, falls jQuery nicht vom CDN geladen werden kann:

```
<script>window.jQuery || document.write('<script src="pfad/jquery-3.6.4.min.js"></script>')  
</script>
```

window.jQuery fragt ab, ob das jQuery-Objekt geladen werden konnte

Empfohlene Reihenfolge:

- zu erst jQuery einbinden,
- danach erst die Skripte

Die allgemeine Syntax von jQuery

Das jQuery-Objekt verfügt über eine Sammlung von Methoden, die für übliche JavaScript-Aktivitäten vorbereitet sind.

Um ein jQuery-Objekt zu erzeugen, nutzt man eine Funktion:

`$ ()`

In der Klammer werden entsprechende Parameter angegeben, die zur Weiterverarbeitung an die Funktion übergeben werden. Der Parameter ist in erster Linie ein Identifikator eines Elements im Dokument bzw. des Dokumentes.

`$ (argument)` // Bildung einer primären Collection

Ergänzt durch einen `.` werden Methoden aufgerufen, die eine Funktion aufrufen können:

`$ (argument) . methode () ;` // Bildung einer sekundären Collection

... es können auch weitere Methoden und Filter verkettet werden:

`$ (argument) . methode () . filter () . methode ()`

Die allgemeine Syntax von jQuery

Im jQuery unterscheidet man die genutzten Methoden

Transparente jQuery-Methoden:

- Eine transparente jQuery-Methode ist eine Methode, die im Hintergrund ausgeführt wird, ohne dass der Benutzer es bemerkt oder explizit darauf zugreift.
- Diese Methoden sind oft nützlich für die ordnungsgemäße Funktionsweise von jQuery, z.B. um bestimmte Funktionen auszuführen, sobald das Dokument vollständig geladen ist.
- Beispiele für transparente jQuery-Methoden sind `$.ready()` und `$.each()`.
- Diese Methoden führen keine Änderungen an den ausgewählten DOM-Elementen durch und geben auch kein Element oder Wert zurück.

Destruktive jQuery-Methoden:

- Eine destruktive jQuery-Methode ist eine Methode, die Änderungen am ausgewählten DOM-Element vornimmt und es dadurch verändern oder entfernen kann.
- Beispiele für destruktive jQuery-Methoden sind `remove()` und `empty()`.
- Diese Methoden können Änderungen an den ausgewählten DOM-Elementen durchführen und geben kein Element oder Wert zurück.

Terminierende jQuery-Methoden:

- Eine terminierende jQuery-Methode ist eine Methode, die das jQuery-Objekt zurückgibt und die Verkettung von Methoden beendet.
- Diese Methoden sind oft am Ende von Verkettungen von jQuery-Methoden nützlich, um das Ergebnis der Verkettung zurückzugeben.
- Beispiele für terminierende jQuery-Methoden sind `val()` und `html()`.
- Diese Methoden können Änderungen an den ausgewählten DOM-Elementen durchführen und geben ein Element oder Wert zurück.

Die allgemeine Syntax von jQuery

So kann man zum Beispiel ganz einfach dafür sorgen, dass alle Scripte erst verarbeitet werden, wenn das Dokument komplett geladen ist.

```
$(document).ready(function() {  
    // alle Definitionen  
    . . .  
});
```

Der Parameter **document** wird an das jQuery-Objekt übergeben.

Die aufgerufene Methode **ready** wird ausgeführt. Sie überprüft ob das Dokument vollständig in den Browser geladen wurde.

Die anschließende anonyme Funktion wird dann erst ausgeführt, wenn die geprüfte Bedingung erfüllt wurde.

Erstes Beispiel auf die Schnelle

Nach erfolgreichem Laden eines Dokumentes soll der Text im Absatz mit der ID **demo** folgenden Text enthalten:
„Dieser Text ist per jQuery in dem Absatz mit der ID demo eingefügt worden.“

1. Schritt: Legen Sie ein neues Beispiel Dokument an. jQueryDemo01.html
2. Schritt: Binden Sie jQuery in das Dokument ein.
3. Schritt: Erstellen Sie im <body> folgenden Inhalt.

```
<p id="demo"></p>
```

4. Schritt: Ergänzen Sie nun im <head> den Code für das jQuery.

```
// Warten, bis das Dokument geladen ist
$(document).ready(function() {
// Selektieren Sie das Element mit der ID "demo" und ändern Sie seinen Textinhalt
$("#demo").text("Dieser Text ist per jQuery . . .!");
});
```

Kleine Übung

Erstellen Sie ein kleines Beispiel nach folgenden Vorgaben:

1. Aufgabe: Legen Sie ein neues Beispiel Dokument an. [jQueryUebung01.html](#)
2. Aufgabe: Legen Sie eine Überschrift an!
Erstellen Sie eine ungeordnete Liste mit vier Listen-Elementen.
3. Aufgabe: Weisen Sie anschließend folgenden Text in die Elemente!
die Überschrift: Die vier Jahreszeiten
die Listen-Elemente: Frühling / Sommer / Herbst / Winter

Ergänzen Sie optional: Definitions-Liste der Familie: Vater: Gerald

Elemente selektieren

Mit jQuery gibt es mehrere Möglichkeiten, um Elemente auf einer Webseite auszuwählen. Hier sind einige der häufigsten Methoden:

Elemente nach ihrem Typ auswählen:

Verwenden Sie den Typ des Elements, um es auszuwählen. Zum Beispiel:

```
$('p')    // wählt alle <p> Elemente auf der Seite aus  
$('a')    // wählt alle <a> Elemente auf der Seite aus
```

Elemente nach ihrer ID auswählen:

Verwenden Sie die ID des Elements, um es auszuwählen. Zum Beispiel:

```
$('#meine-id') // wählt das Element mit der ID "meine-id" aus
```

Elemente nach ihrer Klasse auswählen:

Verwenden Sie die Klasse des Elements, um es auszuwählen. Zum Beispiel:

```
$('.meine-klasse') // wählt alle Elemente mit der Klasse "meine-klasse" aus
```

Elemente selektieren

Elemente nach einem Attribut auswählen:

Verwenden Sie das Attribut des Elements, um es auszuwählen. Zum Beispiel:

```
$('input[type="text"]') // wählt alle <input> Elemente mit dem Attribut "type=text" aus  
$('img[alt="mein-alt-text"]') // wählt alle <img> Elemente mit dem Attribut "alt=mein-alt-text" aus
```

Kombinieren Sie Selektoren:

Sie können Selektoren kombinieren, um spezifischere Selektoren zu erstellen. Zum Beispiel:

```
$('div.meine-klasse') // wählt alle <div> Elemente mit der Klasse "meine-klasse" aus  
$('ul li') // wählt alle <li> Elemente innerhalb eines <ul> Elements aus
```

Verwenden Sie das Eltern-Kind-Beziehungen:

Sie können die Beziehungen zwischen Elementen verwenden, um bestimmte Elemente auszuwählen. Zum Beispiel:

```
$('ul > li') // wählt alle <li> Elemente aus, die direkt innerhalb eines <ul> Elements liegen  
$('div:first-child') // wählt das erste <div> Element aus
```


Elemente selektieren

Verwenden Sie Pseudo-Selektoren:

Sie können Pseudo-Selektoren verwenden, um bestimmte Zustände oder Eigenschaften von Elementen auszuwählen. Zum Beispiel:

```
$('input:focus') // wählt das <input> Element aus, das gerade den Fokus hat  
$('a:hover')     // wählt alle <a> Elemente aus, über die gerade der Mauszeiger schwebt
```

Verwenden Sie das Elementeigenschaften:

Sie können die Eigenschaften von Elementen verwenden, um sie auszuwählen. Zum Beispiel:

```
$('input[type="checkbox"]:checked') // wählt alle angehakten Checkboxes aus  
$('input[type="text"][name="vorname"]') // wählt alle Textfelder mit dem Namen "vorname" aus
```

Verwenden Sie das Index-basiertes Selektoren:

Sie können die Indexposition von Elementen verwenden, um sie auszuwählen. Zum Beispiel:

```
$('ul li:first') // wählt das erste <li> Element innerhalb eines <ul> Elements aus  
$('ul li:last')  // wählt das letzte <li> Element innerhalb eines <ul> Elements aus  
$('ul li:eq(2)') // wählt das dritte <li> Element innerhalb eines <ul> Elements aus (beginnt mit Index 0)
```

Elemente selektieren

Verwenden Sie das Filtern:

Sie können Filter verwenden, um Elemente anhand ihrer Eigenschaften oder Beziehungen auszuwählen. Zum Beispiel:

```
$('li').filter('.meine-klasse') // wählt alle <li> Elemente mit der Klasse "meine-klasse" aus  
$('li').has('a') // wählt alle <li> Elemente aus, die mindestens ein <a> Element enthalten
```

Verwenden Sie das Traversieren:

Sie können die Beziehungen zwischen Elementen verwenden, um Elemente auszuwählen, die sich innerhalb oder außerhalb eines bestimmten Bereichs befinden. Zum Beispiel:

```
$('ul').find('li') // wählt alle <li> Elemente aus, die innerhalb eines <ul> Elements liegen  
$('li').parent() // wählt das Elternelement aller <li> Elemente aus  
$('li').next() // wählt das nächste Element nach jedem <li> Element aus
```

Elemente selektieren

Verwenden Sie das Children-Selektor:

Sie können den Children-Selektor verwenden, um Elemente auszuwählen, die direkt unter einem bestimmten Elternelement liegen. Zum Beispiel:

```
$('ul#meine-liste > li') // wählt alle <li> Elemente aus, die direkt unter einem <ul> Element mit der ID "meine-liste" liegen
```

Verwenden Sie das Attribute-Selektor mit Regulären Ausdrücken:

Sie können das Attribut-Selektor verwenden und Reguläre Ausdrücke verwenden, um Elemente mit Attributen auszuwählen, die einem bestimmten Muster entsprechen. Zum Beispiel:

```
$('img[src$=".jpg"]') // wählt alle <img> Elemente aus, deren "src" Attribut mit ".jpg" endet  
$('a[href^="https://"]') // wählt alle <a> Elemente aus, deren "href" Attribut mit "https://" beginnt
```

Verwenden Sie das Contains-Selektor:

Sie können das Contains-Selektor verwenden, um Elemente auszuwählen, die einen bestimmten Text enthalten. Zum Beispiel:

```
$('p:contains("Lorem ipsum")') // wählt alle <p> Elemente aus, die den Text "Lorem ipsum" enthalten
```

Typische Beispiele für das Selektieren von Elementen

Nach erfolgreichem Laden eines Dokumentes soll ein passender Text in dem jeweiligen Listen-Element erscheinen

1. Schritt: Legen Sie ein neues Beispiel Dokument an. jQueryDemo02.html
2. Schritt: Binden Sie jQuery in das Dokument ein.
3. Schritt: Erstellen Sie im <body> eine ungeordnete Liste mit folgenden Elementen.

```
<li id="demo"></li>
<li class="klasse"></li>
<li title="titel"></li>
```

4. Schritt: Ergänzen Sie nun im <head> den Code für das jQuery.

```
$(document).ready(function() {
    $('#demo').text("Hier wurde die ID selektiert . . .!");
    $('.klasse').text("Hier wurde eine Klasse selektiert . . .!");
    $('li[title="titel"]').text("Hier wurde das Attribut selektiert . . .!");
});
```

5. Schritt: Erstellen Sie weitere Möglichkeiten für Selektoren. Legen Sie dazu die Datei jQueryUebung02.html an.

Einen Button mit Funktionen versehen

Ein typisches Beispiel bei der Arbeit mit jQuery ist der Einsatz von Buttons oder Links, die auf der Basis von Mausklicks individuelle Aktivitäten auslösen können. Hier einfach das Grundbeispiel

Der HTML-Code für einen Button:

```
<button id="meinButton">Klick mich!</button>
```

Der jQuery-Code, um dem Button eine Funktion zuzuweisen:

```
$(document).ready(function() {  
    $('#meinButton').click(function() {  
        alert('Du hast den Button geklickt!');  
    });  
});
```

Dieser Code bindet eine Funktion an den Klick-Event des Buttons mit der ID "meinButton". Wenn der Benutzer auf den Button klickt, wird eine Benachrichtigung angezeigt, die besagt, dass der Button geklickt wurde. Natürlich kann die Funktion angepasst werden, um etwas anderes zu tun, wie beispielsweise eine AJAX-Anfrage an den Server zu senden oder eine Animation auf der Seite auszuführen.

Einen Button mit Funktionen versehen

Der HTML-Code für einen Button:

```
<p>Dieser Text wird ausgeblendet, wenn der Button geklickt wird.</p>  
<button id="ausblendenButton">Text ausblenden</button>
```

Der jQuery-Code, um dem Button eine Funktion zuzuweisen:

```
$(document).ready(function() {  
    $('#ausblendenButton').click(function() {  
        $('p').hide();  
    });  
});
```

Dieser Code bindet eine Funktion an den Klick-Event des Buttons, die den Inhalt des p-Tags ausblendet, wenn der Button geklickt wird.

Einen Button mit Funktionen versehen

Der HTML-Code für einen Button:

```
<p style="display: none;">Dieser Text wird eingeblendet, wenn der Button geklickt wird.</p>  
<button id="einblendenButton">Text einblenden</button>
```

Der jQuery-Code, um dem Button eine Funktion zuzuweisen:

```
$(document).ready(function() {  
    $('#einblendenButton').click(function() {  
        $('p').show();  
    });  
});
```

Einen Button mit Funktionen versehen

Der HTML-Code für einen Button:

```
<div style="background-color: yellow;">Dieser Hintergrund wird geändert, wenn der Button  
geklickt wird.</div>  
<button id="farbenButton">Farbe ändern</button>
```

Der jQuery-Code, um dem Button eine Funktion zuzuweisen:

```
$(document).ready(function() {  
    $('#farbenButton').click(function() {  
        $('div').css('background-color', 'green');  
    });  
});
```


Einen Button mit Funktionen versehen

Der HTML-Code für einen Button:

```
<div style="background-color: yellow;">Dieser Hintergrund wird geändert, wenn der Button  
geklickt wird.</div>  
<button id="farbenButton">Farbe ändern</button>
```

Der jQuery-Code, um dem Button eine Funktion zuzuweisen:

```
$(document).ready(function() {  
    $('#farbenButton').click(function() {  
        $('div').css('background-color', 'green');  
    });  
});
```

Einen Button mit Funktionen versehen

Der HTML-Code für einen Button:

```
<p class="klasse1">Diese Klasse wird geändert, wenn der Button geklickt wird.</p>  
<button id="klasseButton">Klasse ändern</button>
```

Der jQuery-Code, um dem Button eine Funktion zuzuweisen:

```
$(document).ready(function() {  
    $('#klasseButton').click(function() {  
        $('p').addClass('klasse2');  
    });  
});
```

Einen Button mit Funktionen versehen

Der HTML-Code für einen Button:

```
<p class="klasse1 klasse2">Diese Klasse wird geändert, wenn der Button geklickt wird.</p>
<button id="klasseButton">Klasse ändern</button>
```

Der jQuery-Code, um dem Button eine Funktion zuzuweisen:

```
$(document).ready(function() {
    $('#klasseButton').click(function() {
        $('p').removeClass('klasse2');
    });
});
```

Aufgabe: Legen Sie ein neues Beispiel Dokument an. jQueryDemo03.html
In diesem Dokument sollen Beispiele für Button und Funktionen angelegt werden.

HTML-Elemente ein- und ausblenden

Das Ein- und Ausblenden von HTML-Elementen ist ein weiteres typisches Beispiel für Interaktion.

Hierbei sollte man unterscheiden, ob man das Ein- und Ausblenden ruckartig oder stufenweise machen möchten:

Für das Umschalten sichtbar/unsichtbar: `.show()` oder `.hide()`

```
<button id="ein-btn">Einschalten</button>  
<div id="element">Hier ist das Element.</div>
```

```
$(document).ready(function() {  
    $("#ein-btn").click(function() {  
        $("#element").show();  
    });  
});
```

```
<button id="aus-btn">Ausschalten</button>  
<div id="element">Hier ist das Element.</div>
```

```
$(document).ready(function() {  
    $("#aus-btn").click(function() {  
        $("#element").hide();  
    });  
});
```


HTML-Elemente ein- und ausblenden

Für das langsame Ein- und Ausblenden: `.fadeIn()` oder `.fadeOut()`

```
<button id="einblenden-btn">Einblenden</button>
<div id="element">Hier ist das Element.</div>
```

```
$(document).ready(function() {
    $("#einblenden-btn").click(function() {
        $("#element").fadeIn();
    });
});
```

```
<button id="ausblenden-btn">Ausblenden</button>
<div id="element">Hier ist das Element.</div>
```

```
$(document).ready(function() {
    $("#ausblenden-btn").click(function() {
        $("#element").fadeOut();
    });
});
```

Hinweis: Die Methoden `fadeIn()` und `fadeOut()` erlauben durch Angabe von Parametern noch die Beeinflussung des Vorganges des Ein- und Ausblendens.

Aufgabe: Legen Sie ein neues Beispiel Dokument an. `jQueryDemo04.html`
In diesem Dokument sollen Beispiele für Ein- und Ausblenden angelegt werden.

VIELEN DANK!

