

Allgemeines

- Externe JavaScript-Bibliothek
- Vereinfacht die Nutzung von JavaScript für Webseiten
- Erlaubt:
 - Einfachen Zugriff auf HTML-Elemente
 - Einfache Ausführung von Aktionen auf diese Elemente
- Bibliothek muss entweder auf lokalem Server installiert werden oder extern verlinkt werden
- Download von: [jQuery.com](https://jquery.com)
- Einbindung im Kopfbereich
- Alternativ: Google Content Delivery Network
- Google CDN schneller, da ggf. schon in Cache, aber problematisch wegen DGSVO (vgl. Google Fonts)

```
\textless head\textgreater
  \textless script src="jquery-3.7.1.min.js"\textgreater
  \textless /script\textgreater
\textless /head\textgreater

// from CDN
\textless head\textgreater
  \textless script src="https://ajax.googleapis.com/ajax/
    libs/jquery/3.7.1/jquery.min.js"\textgreater
  \textless /script\textgreater
\textless /head\textgreater
```

Syntax

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

- Alle jQuery-Kommandos beginnen mit dem Dollar-Zeichen

```
$(this).hide(); // versteckt das aktuelle Element
```

```
$("h2").hide(); // versteckt alle \textless h2\textgreater
-Elemente
```

```
$(".myClass").hide(); // versteckt alle Elemente der
// der Klasse "myClass"
```

```
$("#myID").hide(); // versteckt das Element mit
// der ID "myID"
```

Selektoren

- Selektion aller Elemente mit dem Tag **tagname**:

```
$ ("tagName")
```

- Selektion des Elements mit der ID **id**:

```
$ ("#id ")
```

- Selektion aller Elemente der Klasse **class**:

```
$ (".class ")
```

- Selektion aller Elemente mit dem Tag **tagname**, deren Attribut **attribute** den Wert **value** aufweist

```
$ ("tagName[attribute='value'] ' ')
```

- Mehrere Alternative können durch Kommata getrennt kombiniert werden

```
$ ("#myId , . myClass , . my2ndClass")
```

Ereignisse/Aktionen

- Ereignisse lösen Aktionen aus
- z.B. Maus-Klick, Mauszeiger auf ein Element bewegen, Tastatureingaben, Scrollen, etc.
- Entsprechende Events werden nach dem Selektor mit dem Punktoperator abgetrennt
- Damit auch eine entsprechende Aktion erfolgt, muss diese als Funktionsaufruf innerhalb der Ereignisverarbeitung definiert werden
- Funktionen können auch direkt inline definiert werden

```
$( "h1" ).mouseenter ( );  
  
// ———  
  
$( "h1" ).mouseenter ( function ( ) {  
    alert ( "Beruehrt" );  
} )
```

- Mausereignisse:
 - click
 - dblclick
 - mouseenter
 - mouseleave
- Tastaturereignisse
 - keypress
 - keydown
 - keyup
- Formularereignisse
 - submit
 - change
 - focus
 - blur
- Seitenereignisse
 - load
 - resize
 - scroll
 - unload

Ready-Funktion

```
$(document).ready(function(){
    $("button").click(function(){
        $("#element2hide").hide();
    });
});
```

- Es ist sinnvoll, jQuery-Anweisungen innerhalb eines Ready- Events auszuführen:
 - Verhindert die Ausführung vor Aufbau der kompletten Webseite
 - Hier: Auszublendes Element könnte u.U. noch gar nicht angezeigt werden

Selektorenzusätze

Selektorzusatz Beispiel Beschreibung

:first \$("p:first") Erstes Absatzelement

:last \$("p:last") Letztes Absatzelement

:even \$("tr:even") Gerade Tabellenreihen

:odd \$("tr:odd") Ungerade Tabellenreihen

parent >child \$("div >p") <p>-Elemente, die direkte Kindelemente eines <div>-Containers sind

parent descendant \$("div p") <p>-Elemente, die Teil eines <div>-Containers sind

parent + next \$("div + p") <p>-Elemente, die unmittelbar vor oder hinter einem <div>-Container stehen

element siblings \$("div p") <p>-Elemente, die auf gleicher Stufe wie ein <div>-Container stehen

:eq(index) \$("li ol:eq(2)") 2. Element in einer nummerierten Liste

:gt(value) \$("li ul:gt(5)") Alle Elemente in einer unnummerierten Liste ab dem 7.(!) Element

:lt(value) \$("li ol:lt(4)") Die Elemente 1-4 (Indizes 0-3) einer nummerierten Liste

:not(select or) \$("input:not(empty)") Alle nicht leeren Input-Elemente

:focus \$("":focus") Element, was aktuell den Fokus hat

:contains(text) \$("":contains('TEST')) Alle Elemente, die das Wort "TEST" enthalten

:has(selector) \$("div:has(ul)") Alle <div>-Container, die ein unnummeriertes Listenelement enthalten

:empty \$("p:empty") Alle <p>-Elemente, die leer sind

:hidden \$("p:hidden") Alle <p>-Elemente, die nicht sichtbar sind

:visible \$("h1:visible") Alle <h1>-Elemente, die sichtbar sind

:input \$("":input") Alle Input-Elemente

:button \$("":button") Alle Button-Elemente

:image \$("":image") Alle Bild-Elemente

:enabled \$("":enabled") Alle aktiven Input-Elemente

:disabled \$("":disabled") Alle inaktiven Input-Elemente

:selected \$("":selected") Alle ausgewählten Input-Elemente

:checked \$("":checked") Alle angeklickten Check-Boxen

Beispiel

```
<!DOCTYPE html>
<html>
<head>
  <script src="https://ajax.googleapis.com/ajax/libs/jquery\
    /3.7.1/jquery.min.js"></script>
  <script>
    $(document).ready(function() {
      $("#hide").click(function() {
        $("img").hide();
      });
      $("#show").click(function() {
        $("img").show();
      });
    });
  </script>
</head>
<body>
  
  <button id="hide">Hide</button>
  <button id="show">Show</button>
</body>
</html>
```

Blenden (Fading)

- Einblenden ausgewählter Elemente mit der Geschwindigkeit speed ("slow", "fast", Zeit in ms)
 - Anschließend Aufruf der Callback—Funktion
 - Alle Parameter optional

```
$(selector).fadeIn(speed, callback);
```
- Ausblenden ausgewählter Elemente:

```
$(selector).fadeOut(speed, callback);
```
- Ein- oder Ausblenden je nach Zustand:

```
$(selector).fadeToggle(speed, callback);
```
- Ein-/ Ausblenden bis zu einem festgelegten Transparenzgrad

```
$(selector).fadeTo(speed, opacity, callback);
```

Klappen (Sliding)

- Ausklappen ausgewählter Elemente mit der Geschwindigkeit speed ("slow", "fast", Zeit in ms)
 - Anschließend Aufruf der Callback—Funktion
 - Alle Parameter optional

`$(selector).slideDown(speed, callback);`

- Einklappen ausgewählter Elemente:
`$(selector).slideUp(speed, callback);`
- Ein- oder Ausklappen je nach Zustand:
`$(selector).slideToggle(speed, callback);`

Animationen

- Einfache Animationen
`$(selector).animate(parameter,speed, callback);`
- Positionsparameter für Abstand von Seitenbegrenzung (left, right, top, bottom)
- `left:'250px'` // Verschieben zur Pos. 250px von linkem Seitenr.
 - Mehrere Parameter werden durch Kommata voneinander getrennt
 - Standardmäßig haben alle HTML-Elemente eine statische Position, die nicht verschoben werden kann.
 - Um die Position zu verändern, müssen Sie zuerst die CSS-Eigenschaft position des Elements auf relativ, fest oder absolut einstellen!
 - Alle CSS-Styles als Parameter möglich!
- Beispiele: opacity, width, height, fontSize, etc.
- Relative Angaben durch vorangestelltes += möglich, z.B.: `'+=50px'`

Beispiel

```
<!DOCTYPE html>
<html>
  <head>
    <script src="https://ajax.googleapis.com/ajax/libs/
jquery/3.7.1/jquery.min.js"></script>
    <script>
      $(document).ready(function(){
        $("button").click(function(){
          $("div").animate({
            left: '+=25px',
            opacity: '0.5',
            height: '+=15px',
            fontSize: '+=4px',
            width: '100px'
          });
        });
      });
    </script>
  </head>
  <body>
    <button>Run Animation</button>
    <div style="background:#8080F0;
      height:50px; width:50px; position:absolute;">TEST
    </div>
  </body>
</html>
```

Inhalte und Attribute auslesen

- Zugriff auf Elementinhalte
\$(selector).text();
- Elementinhalte inklusive HTML-Tags:
\$(selector).html();
- Werte von Formularfeldern:
\$(selector).val();
- Werte von Attributen:
\$(selector).attr(attrName);

Inhalte und Attribute setzen

- Zugriff auf Elementinhalte
`$(selector).text(value);`
- Elementinhalte inklusive HTML-Tags:
`$(selector).html(value);`
- Werte von Formularfeldern:
`$(selector).val(value);`
- Werte von Attributen:
`$(selector).attr(attrName, value);`

Elemente löschen oder leeren

- Elemente entfernen
`$(selector).remove();`
- Elementinhalte entfernen:
`$(selector).empty();`

Elemente einfügen

- Inhalte am Ende des selektierten Elements einhängen
`$(selector).append(newContent)`
- Inhalte am Anfang des selektierten Elements einhängen:
`$(selector).prepend(newContent);`
- Inhalte nach dem selektierten Elements einfügen:
`$(selector).after(newContent);`
- Inhalte vor dem selektierten Elements einfügen :
`$(selector).before(newContent);`
- Mehrere Elemente können durch Kommata voneinander getrennt gleichzeitig eingefügt werden