



# Wer ist der komische Typ da vorne?



- Name: Tim Stoffel
- Alter: 35
- Firma: CTS Eventim Solutions GmbH
- Position: Software Development Expert Web (Frontend Entwickler)
- In der Firma seit: 2012
  
- Bei Fragen? Fragen!
  
- [tim.stoffel@eventim.de](mailto:tim.stoffel@eventim.de)

- Was ist jQuery? Und wo kommt es her?
- Workshop Vorbereitung
- jQuery
  - Einbindung von jQuery
  - Selektoren und Hilfsfunktionen in jQuery
  - Event Handling in jQuery
  - DOM Manipulation mit jQuery
  - jQuery(this)
  - Plugins und Erweiterungen für jQuery
- Frage? und Antwort!

# Was ist jQuery? Und wo kommt es her?



- jQuery ist eine freie JavaScript Bibliothek
  - Es vereinfacht den Umgang mit JS
  - Es stellt Hilfsfunktionen zur Verfügung
- Die erste Version von jQuery wurde 2006 bereitgestellt
- Zusammen mit anderen Bibliotheken, kämpfte es darum JavaScript einfacher zugänglich zu machen.
- Im Laufe der Jahre sind die meisten anderen Bibliotheken wieder verschwunden
- jQuery jedoch hat überlebt
- Vorteile:
  - Es vereinfacht die JavaScript Syntax (heute weniger als noch 2006)
  - Es kann einfach einem Projekt hinzugefügt werden (auch wenn dieses schon lange besteht)
  - jQuery ist kompatibel mit fast allen Browsern (bei JavaScript muss man pro Methode prüfen)
  - Es gibt zahlreiche Erweiterungen und Plugins, die einem das Leben einfacher machen können
    - jQuery UI (Enthält UI Bausteine, die out of the box funktionieren)
    - jQuery Mobile (Auf touch Eingaben spezialisiert)
    - QUnit (Unit testing mit jQuery)
- Nachteile:
  - Es werden immer ALLE möglichen Methoden und Helfer in die Seite eingebunden, nicht nur die die man benötigt. (86kb für die minifizierte jQuery Datei)
  - Mit JavaScript kann man mittlerweile fast alles genauso einfach wie mit jQuery erledigen.

- Git Clone des Workshop Templates:

```
Microsoft Windows [Version 10.0.17134.829]  
(c) 2018 Microsoft Corporation. Alle Rechte vorbehalten.  
  
C:\Users\tstoffel\Desktop>git clone https://github.com/tstoff/jquery_workshop.git jqueryWorkshop_
```

**[https://github.com/tstoff/jquery\\_workshop.git](https://github.com/tstoff/jquery_workshop.git)**

- Hier nochmal der Befehl erklärt:

```
Microsoft Windows [Version 10.0.17134.829]  
(c) 2018 Microsoft Corporation. Alle Rechte vorbehalten.  
  
C:\Users\tstoffel\Desktop>git clone https://github.com/tstoff/jquery_workshop.git jqueryWorkshop_
```

Git Befehl zum "herunterladen"      URL des Repositories      Der Ordner in den geklont werden soll

- Jetzt öffnet den Ordner mit dem geklonten Code in eurer IDE

- Öffnet nun den Ordner in den ihr das Repository geklont habt
- Öffnet die Index.html im Browser
- Für diesen Workshop **wichtig**:
  - Benutzt **NICHT** euren localhost wie ihr es gewohnt seid um die Webseite im Browser zu öffnen, sondern öffnet die Seite „manuell“.

# 1. Einbindung von jQuery



- Wie integriert man jQuery in die eigene Webseite?

```
<script src="obj/js/jquery-3.4.1.min.js" type="text/javascript"></script>
```

- oder

```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js" type="text/javascript"></script>
```

- Benötigt man ältere jQuery Versionen (z.b. für alten Browser support):
  - Bibliothek download von jQuery selbst:
    - <https://code.jquery.com/jquery/>
  - Von Goolge:
    - <https://developers.google.com/speed/libraries/#jquery>
  - Oder für zukünftige Projekte:
    - <https://www.npmjs.com/package/jquery>



# 1. Einbindung von jQuery

- jQuery nutzt im groben folgende Syntax:

```
$('#someId').css('background-color', '#AAAAAA');
```

Selektor

Methode

Parameter

- Selektoren:
  - Genauso wie im CSS (als string in „“)
    - # für ID's
    - . Für Klassen
    - Ohne Symbol für vordefinierte Tags wie z.B **body**
- Methoden:
  - Eine Auswahl von Hilfsmethoden folgt später (im Beispiel oben css Veränderung)
- Parameter:
  - Je nach Funktion nötig oder nicht (folgt auch später)



# 1. Einbindung von jQuery

- jQuery function als „Container“ für DOM ready:

```
1  $("document").ready(function(){  
2  
3  });|
```

- In kürzer

```
1  $(function(){  
2  
3  });
```

- Mit jQuery notation

```
1  jQuery(function(){  
2  
3  });|
```

- Und jetzt ihr!
  - **Aufgabe 1:**
    - Implement jQuery in the cloned website
    - Use the jQuery .js file from the **obj/js/** folder
    - **OR** use the google library:
      - <https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js>
    - Add a file with the name “**script.js**” in the obj folder
    - Write a console output when the DOM is ready
      - Add your code inside the script.js file

# 1. Einbindung von jQuery

tstoff / jquery\_workshop

Watch 0

Star 0

Fork 0

Code

Issues 0

Pull requests 0

Projects 0

Wiki

Security

Insights

Settings

jQuery workshop related files

Edit

Manage topics

4 commits

5 branches

0 releases

1 contributor

Branch: master

New pull request

Create new file

Upload files

Find File

Clone or download

Switch branches/tags

Latest commit b3bcd38 3 days ago

Find or create a branch...

Branches

Tags

✓ master

task2

task3

task4

task5

up, css and objs for the first task

3 days ago

y to project

3 days ago

up, css and objs for the first task

3 days ago

up, css and objs for the first task

3 days ago

Lösungen zur jeweils vorherigen Aufgabe

[https://github.com/tstoff/jquery\\_workshop](https://github.com/tstoff/jquery_workshop)

## 2. Selektoren in jQuery

- Beispiele für Selektoren:

```
// Element mit der ID
jQuery('#someId')

// Element mit der Klasse
jQuery('.someClass')

// Element mit einer ID UND einer bestimmten Klasse
jQuery('#someId.someClass')

// Element mit einer Klasse, die unterhalb eines Elements mit einer bestimmten ID liegt
jQuery('#someId .someClass')

// Element mit dem Attribut name UND dem gesuchten Namen
jQuery('[name=someName]')
```

- Wann benutzen wir IDs und Klassen zum selektieren?
  - **IDs** nutzen wir **NUR** wenn ein Element einmal auf der Seite vorkommen darf! Wie z.B. Forms.

```
// Element mit der ID  
jQuery('#someId')
```

- **Klassen** nutzen wir für alle anderen Fälle. Klassen sollten allerdings ein „js-“ prefix erhalten, z.B: „.js-someClass“

```
// Element mit der Klasse  
jQuery('.someClass')
```

## 2. Selektoren in jQuery

- Mehrfache Selektion:

```
// Unterschiedliche Elemente in einem Selektor  
jQuery('#someId, #anotherId, #aThirdId')
```

- Best Practice (Beispiel):
  - Selektoren Simpel halten!

```
// Nicht gut  
jQuery('.someClass input#someId').eineFunktion()  
jQuery('.someClass input#someId').zweiteFunktion()  
jQuery('.someClass input#someId').dritteFunktion()
```

```
// Besser  
jQuery('#someId').eineFunktion()  
jQuery('#someId').zweiteFunktion()  
jQuery('#someId').dritteFunktion()
```

## 2. Selektoren in jQuery

- Best Practice:

```
// Am Besten  
var element = jQuery('#someId');  
  
element.eineFunktion();  
element.zweiteFunktion();  
element.dritteFunktion();
```

- Chaining:

```
jQuery('#someId').css('background-color', '#AAA').fadeOut(2000);
```

- Erst css Styling ändern und dann über 2000 Milisekunden ausfaden
- Man spart sich einen Selektor



## 2. Hilfsfunktionen in JQuery

- Beispiele für Hilfsfunktionen:

- Die **.css** Hilfsfunktion:

```
jQuery('#someId').css('background-color', '#AAA').fadeOut(2000);
```

- Die **.addClass** Hilfsfunktion:

```
jQuery('.someClass').addClass('anotherclass');
```

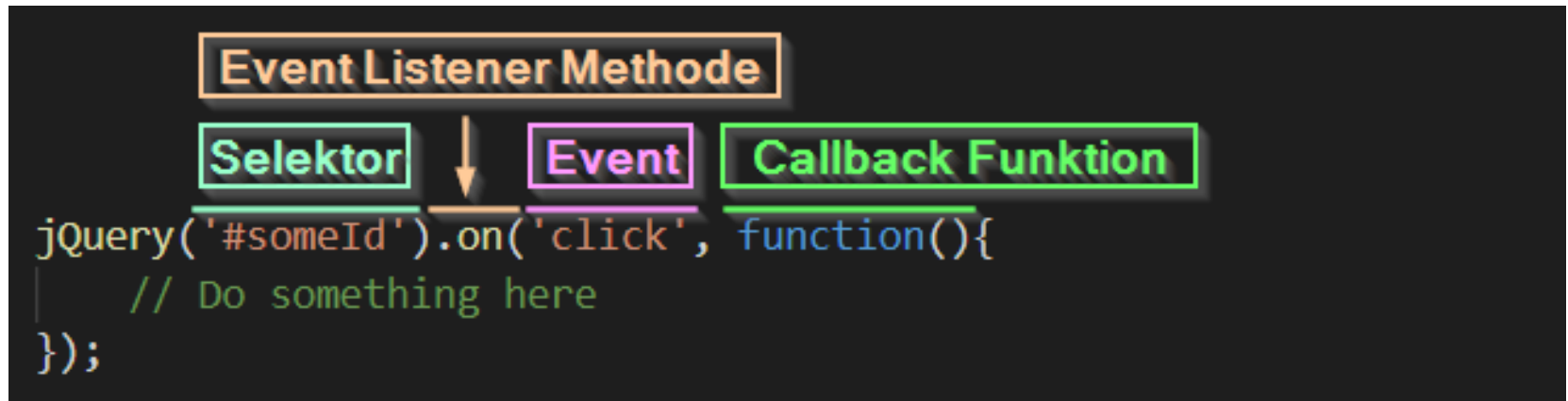
- Weitere Beispiele hier:

- <https://oscarotero.com/jquery/>

- Und jetzt ihr!
  - **Aufgabe 2:**
    - Select all artist images and change their opacity to 0.5 with jQuery
    - Hide the text box with jQuery (just for learning purposes)
    - Delete the console.log
  - **Hint:**
    - Use the **.css** jQuery helper function
    - Check the usage of this function in the jQuery Documentation
  - <https://api.jquery.com/>

# 3. Event Handling in jQuery

- Event Handler Syntax




- Event Listener Methode:
  - `on()` zum registrieren eines Event Handlers
  - `off()` zum entfernen eines Event Handlers
  - `bind()` älteres `on()`, in älterem legacy code findbar (ab Version 3.0 deprecated)
- Event
  - Das Browser Event auf das gehört werden soll
    - z.B. `click`, `mouseover`, `mouseenter`, `mouseleave`, `submit`


# 3. Event Handling in jQuery

- Beispiele für Eventhandler:


```
// Dies funktioniert NICHT
jQuery('#someId').on('click').css('background-color', '#AAA');
```



```
// Dies wiederum funktioniert
jQuery('#someId').on('click', function(){
    jQuery(this).css('background-color', '#AAA');
});
```



```
// In kürzer mit jQuery Hilfsfunktion
jQuery('#someId').click(function(){
    jQuery(this).css('background-color', '#AAA');
});
```



# 3. Event Handling in jQuery



- Nochmal ein kurzer Abstecher zu Hilfsfunktionen
  - Funktionieren mit und ohne Parameter auf verschiedene Weise

```
// Dies erzeugt einen Event Handler für das Klick event
jQuery('#someId').click(function(){
    // do something
});
```

```
// Dies simuliert einen Klick
jQuery( '#someId').click();
```

- Und jetzt ihr!
  - **Aufgabe 3:**
    - Change the opacity of every single image back to 1.0 if the mouse hovers over it. (use the `.fadeTo()` function if you like).
    - Change the opacity back to 0.5 if the mouse leaves the Image.
    - A click on every image displays the text box below the images.
      - The click should highlight that image with a yellow border.
      - The highlighting should only be shown at the last clicked image.
  - **Hint:**
    - Use the `.on` Event Handler Method to detect if the mouse is over an image
    - Check the usage of this function in the jQuery Documentation
      - <https://api.jquery.com/>

## 4. jQuery(this)

- Ihr könnt in jQuery ein Element, dass ihr gerade selektiert habt für ein weiteres Query nutzen, wenn ihr euch in dessen Scope befindet.
- Dies ist sehr nützlich wenn ihr das Element zur Laufzeit nicht kennt.
- Ein Beispiel:

```
<ul>
  <li class="someClass">Text1</li>
  <li class="someClass">Text2</li>
  <li class="someClass">Text3</li>
  <li class="someClass">Text4</li>
</ul>
```

```
jQuery('.someClass').each(function(){
  jQuery(this).css('color', '#000');
});
```



## 5. DOM Manipulation in jQuery

- Auslesen von HTML mit der `.html()` Hilfsfunktion

```
// Liest das HTML aus dem Inhalt des Elements aus  
jQuery('#someId').html();
```

```
<!-- Der HTML Schnipsel -->  
<div id="someId">  
|   <p>Ich bin ein Testtext</p>  
</div>
```

```
<!-- Das gibt die .html() Funktion zurück -->  
<p>Ich bin ein Testtext</p>
```

# 5. DOM Manipulation in jQuery

- Schreiben von HTML mit der `.html()` Hilfsfunktion

```
// Verändert das html in dem div Container mit der id 'someId'  
jQuery('#someId').html('<i>Der Text verändert sich</i>');
```

```
<!-- Der HTML Schnipsel vorher -->  
<div id="someId">  
| <p>Ich bin ein Testtext</p>  
</div>
```

```
<!-- Der HTML Schnipsel nachher -->  
<div id="someId">  
| <i>Der Text verändert sich</i>  
</div>
```

- Andere hilfreiche Funktionen um den DOM zu manipulieren
  - **.append()** fügt HTML in ein Element als Kind am Ende hinzu
  - **.prepend()** fügt HTML in ein Element als Kind am Anfang hinzu
  - **.after()** fügt HTML als Sibling hinter dem selektierten Element hinzu.
  - **.before()** fügt HTML als Sibling vor dem selektierten Element hinzu.
  - **.clone()** erstellt eine Kopie des selektierten Elements, welches mit den oberen Methoden an eine neue Stelle im DOM platziert werden kann.

- Und jetzt ihr!
  - **Aufgabe 4:**
    - Use the JavaScript Object from Github (task4 Branch Readme) to Mock a backend and display the text and the headline corresponding to the chosen artist picture.
  - **Hint:** Mock the Backend by saving the JavaScript object in a const.
  - **Hint2:** Do not use jQuery only ;-)
  - **Hint3:** Data attributes are a great way to give html tags specific content besides classes or id's.

# 6. Plugins und Erweiterungen für jQuery



- jQuery lebt von seinen vielfältigen, kostenlosen Plugins
- Einige werden professionell veröffentlicht
  - jQuery UI - <https://jqueryui.com/>
  - jQuery Mobile - <https://jquerymobile.com/>
- Doch wofür braucht man Plugins?
  - Erweitern der jQuery Bibliothek
  - Einfaches nutzen eines bestimmten Features
    - z.B. Lightboxen, Slider, Akkordeons, Formular Validierung ...
- Und wie bindet man externe Plugins in die eigene Seite ein?

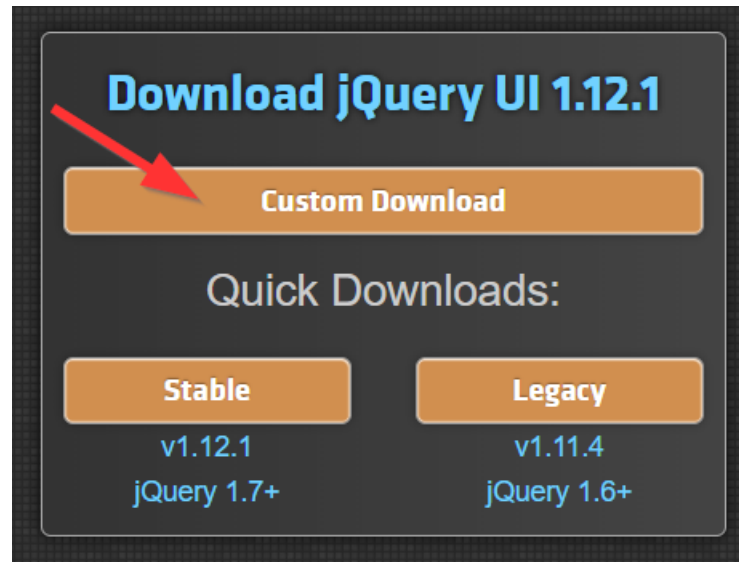
# 6. Plugins und Erweiterungen für jQuery



- Im Normalfall kommen jQuery Plugins mindestens mit einer `.js` Datei
  - In vielen Fällen bereits minifiziert als `.min.js`
- Bei vielen Plugins werden oft auch andere Dateien benötigt
  - `.css` oder `.sass` bzw. `.less` Dateien zum stylen oder als styling templates.
  - Andere `.js` Dateien für die Funktionserweiterung des externen Plugins oder als dependency ohne die das Plugin an sich nicht funktionieren würde.
- **WICHTIG!** Das Skript für das Plugin muss **NACH** jQuery geladen werden. Ansonsten funktioniert es nicht.

# 6. Plugins und Erweiterungen für jQuery

- Plugin Einbindung am Beispiel jQuery UI:
  - jQuery UI bietet sehr detaillierte Möglichkeiten der Einbindung



- Jede Komponente kann einzeln ausgewählt werden
- Man erstellt sich quasi sein persönliches Bundle



# 6. Plugins und Erweiterungen für jQuery

- Nach dem Download erhält man eine Menge Dateien

external		<b>Hier steckt jQuery drin</b>	
images	Dateiordner		
AUTHORS	Textdokument	7 KB	Nein
index	Chrome HTML Document	6 KB	Nein
jquery-ui	Kaskadierendes Stylesheet...	10 KB	Nein
jquery-ui			
jquery-ui.min	Kaskadierendes Stylesheet...	9 KB	Nein
jquery-ui.min		<b>Hier ist das minifizierte jQuery UI Skript</b>	
jquery-ui.structure	Kaskadierendes Stylesheet...	7 KB	Nein
jquery-ui.structure.min	Kaskadierendes Stylesheet...	6 KB	Nein
jquery-ui.theme	Kaskadierendes Stylesheet...	4 KB	Nein
jquery-ui.theme.min	Kaskadierendes Stylesheet...	3 KB	Nein
LICENSE	Textdokument	2 KB	Nein
package.json	JSON-Datei	1 KB	Nein

# 6. Plugins und Erweiterungen für jQuery

- Nach dem integrieren der Skripte muss man das neue Plugin initialisieren

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>datepicker demo</title>
  <link rel="stylesheet" href="//code.jquery.com/ui/1.12.1/themes/s
  <script src="//code.jquery.com/jquery-1.12.4.js"></script>
  <script src="//code.jquery.com/ui/1.12.1/jquery-ui.js"></script>
</head>
<body>

<div id="datepicker"></div>

<script>
$( "#datepicker" ).datepicker();
</script>

</body>
</html>
```

**Stylesheets und Skripte**

**Datepicker initialisierung**

## 6. Plugins und Erweiterungen für jQuery

- Das Ergebnis sieht dann so aus:

June 2019						
Su	Mo	Tu	We	Th	Fr	Sa
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30						

# 6. Plugins und Erweiterungen für jQuery



- Und jetzt ihr!
  - **Aufgabe 5:**
    - Create a new Project folder „jqueryWorkshopPlugin“
      - Clone the same Repo from GitHub in that folder
    - Get the new Codebase from Github with the following command:

```
C:\Users\tstoffel\Desktop\jqueryWorkshopPlugin>git checkout taskPlugin
```

- Use the Slick Plugin to build an Image Slider for the Band Images
      - <https://kenwheeler.github.io/slick/>
    - Let the Slider show 2 Images and slide 1 Image at a time.
  - **Hint:**
    - Read the documentation carefully ;-)

**Das wars für heute!!!**

Ihr seid jetzt jQuery Experten!



- Noch etwas unklar in Bezug auf **jQuery**?
- Oder habt ihr Fragen zu mir oder **Eventim**?
  - **JETZT** ist die Zeit für Fragen!

# Feedback? Immer her damit!

- Gebt mir gerne Feedback:
- <https://www.surveymonkey.de/r/P6JC5VK>

