



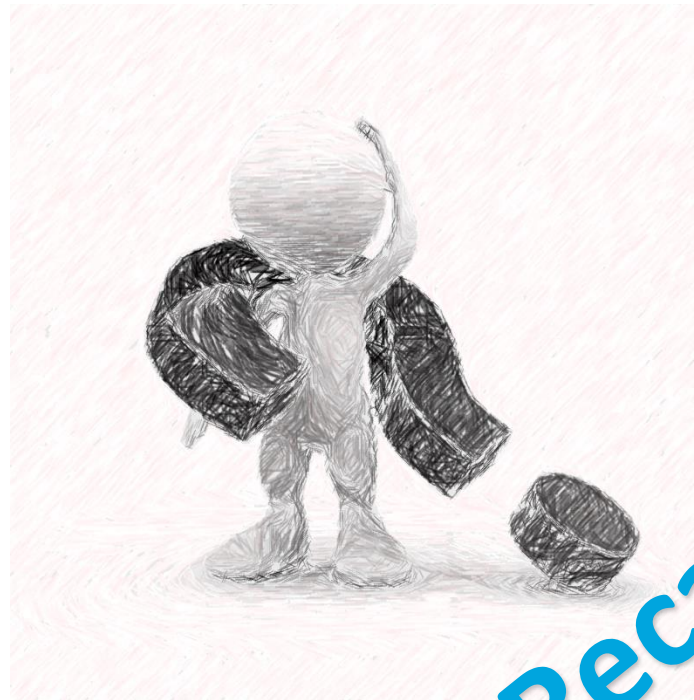
# Willkommen bei der Höheren Berufsbildung Uster

## JavaScript Raphael Ritter

Ein Angebot der Berufsfachschule Uster und der Höheren Fachschule Uster

# Agenda

- Prüfung Grundlagen
- jQuery
  - Einführung
  - DOM Traversierung
  - DOM Manipulation
- jQuery Validation



Recap

# Unterlagen der HSR

Einige Inhalte sind aus den Folien / Übungen des CAS  
Frontend Engineering der Hochschule für Technik Rapperswil.



# HSR

HOCHSCHULE FÜR TECHNIK  
RAPPERSWIL

FHO Fachhochschule Ostschweiz

[www.hsr.ch](http://www.hsr.ch)

<http://hsr.ch/CAS-Front-End-Engineering.12432.0.html>

# Prüfung Grundlagen

- Maximal 30 Minuten Zeit
- Prüfung kann jederzeit abgegeben werden, anschliessend bitte leise sein oder den Raum verlassen



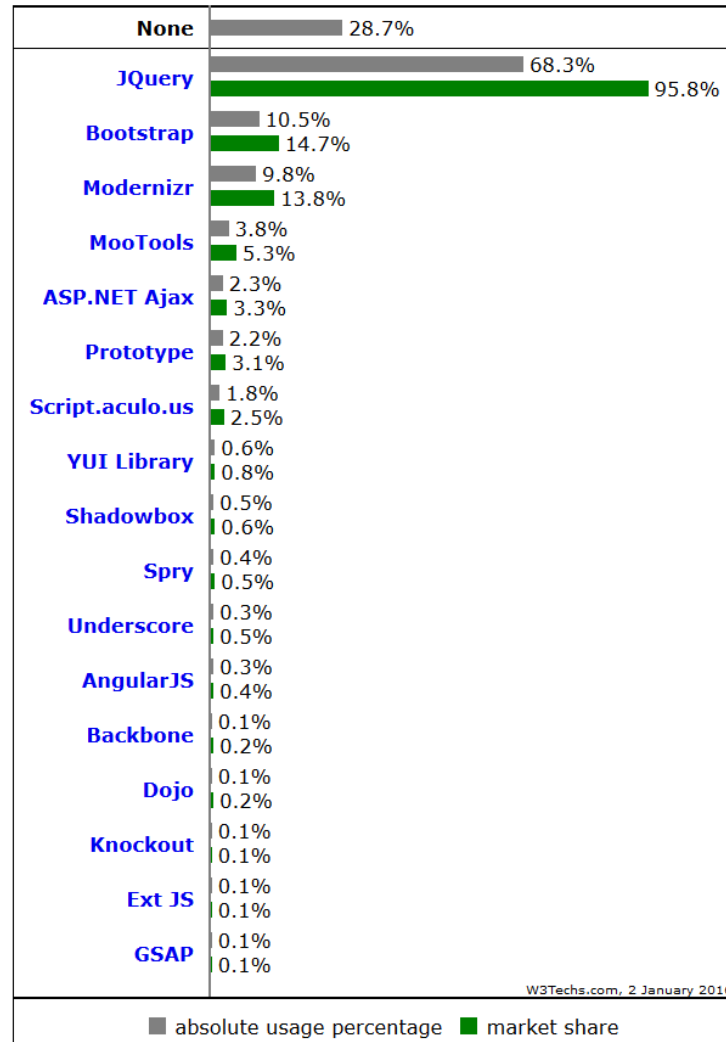
Bezeichnung des Moduls

# jQuery

# JavaScript Frameworks

- Zahlreiche JavaScript Bibliotheken die das Erstellen von Rich Internet Applikationen vereinfachen. Damit können „Desktop-Ähnliche“ Applikationen gebaut werden.
  - Vereinfachung von Ajax
  - Wiederverwendung von Code / Lösungen
- Beispiele
  - JQuery
    - <http://jquery.com>
  - Angular JS
    - <https://angularjs.org/>
  - Aurelia
    - <http://aurelia.io/>
  - Ember
    - <http://emberjs.com/>

# Marktanteile Libraries





# JavaScript

# jQuery

- Stellt verschiedene Komponenten zur Verfügung
- Sind sehr einfach zum einbinden und anwenden
- Web Seite: <http://jquery.com>
- Mittels jQueryUI hat es UI Komponenten für fast jeden Anwendungsfall
- jQuery 1.x für alle Webbrowser
- jQuery 2.x für moderne Webbrowser (Ab IE 9.0)



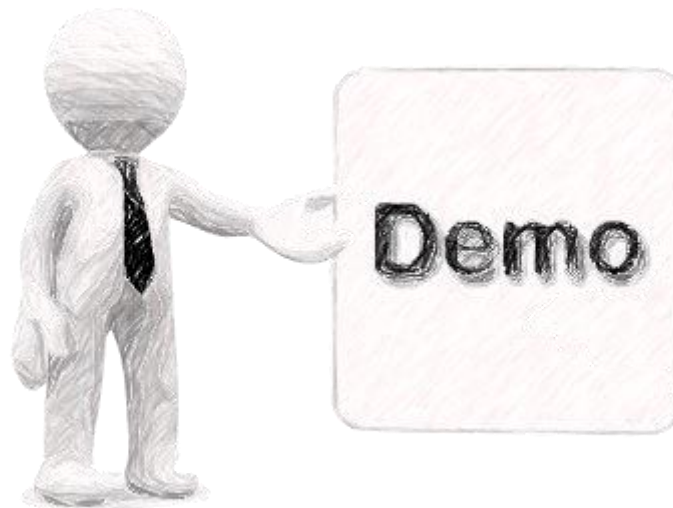
## jQuery

- Query ist darauf ausgerichtet HTML-Elemente zu selektieren und auf diese ausgewählten Elemente Operationen anzuwenden
- Die Entwickler von jQuery legen besonderes Augenmerk auf die Funktionalität unter verschiedenen Browsern
- Die Funktionalität kann mittels Plugins-Schnittstelle erweitert werden
- jQuery-Funktionen werden analog JavaScript in HTML-Dokumente eingebettet oder durch einen Link auf eine .js-Datei eingebunden

## jQuery

- Die jQuery-Funktionalität ist als JavaScript-Code implementiert
- Die Bibliothek steht in komprimierten Form als jquery-  
<version>.min.js auf <http://jquery.com/> zum Download bereit
- Einbinden als lokale Quelle
  - Download und Bereitstellung der JavaScript-Bibliothek im eigenen Webspacespace  
`<script type="text/javascript" src="jquery-1.9.0.min.js"></script>`
- Zentrale Webserver von CDN (Content Delivery Network)
  - Beispiel: Google CDN  
`<script  
src="http://ajax.googleapis.com/ajax/libs/jquery/1.10.2/jquery.min.js">`
  - Beispiel: Microsoft CDN  
`<script src="http://ajax.aspnetcdn.com/ajax/jQuery/jquery-1.10.2.min.js">`

# JavaScript jQuery



## jQuery

- Das wichtigste Konzept in jQuery ist die Verwendung von Elementselektoren
- Anstelle von getElementByXX() wird mittels \$(...) ein (oder mehrere) Element(e) ausgewählt
- Auf diesen werden dann alle Aktionen ausgeführt
- Beispiele
  - \$("#id") -> Element mit id selektieren
  - \$(".klasse") -> Alle Elemente mit dieser Klasse
  - \$("div, p, a, span, li") -> Alle Elemente dieser Tags selektieren

# JavaScript

# jQuery

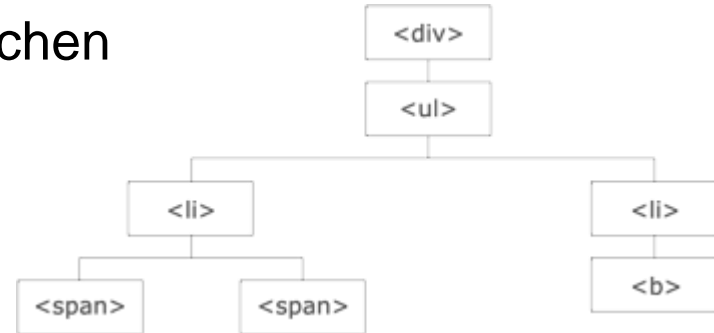
Syntax	Description
<code>\$("*")</code>	Selects all elements
<code>\$(this)</code>	Selects the current HTML element
<code>\$("p.intro")</code>	Selects all <code>&lt;p&gt;</code> elements with class="intro"
<code>\$("p:first")</code>	Selects the first <code>&lt;p&gt;</code> element
<code>\$("ul li:first")</code>	Selects the first <code>&lt;li&gt;</code> element of the first <code>&lt;ul&gt;</code>
<code>\$("ul li:first-child")</code>	Selects the first <code>&lt;li&gt;</code> element of every <code>&lt;ul&gt;</code>
<code>\$("[href]")</code>	Selects all elements with an href attribute
<code>\$("a[target='_blank']")</code>	Selects all <code>&lt;a&gt;</code> elements with a target attribute value equal to <code>"_blank"</code>
<code>\$("a[target!='_blank']")</code>	Selects all <code>&lt;a&gt;</code> elements with a target attribute value NOT equal to <code>"_blank"</code>
<code>\$(":button")</code>	Selects all <code>&lt;button&gt;</code> elements and <code>&lt;input&gt;</code> elements of type="button"
<code>\$("tr:even")</code>	Selects all even <code>&lt;tr&gt;</code> elements
<code>\$("tr:odd")</code>	Selects all odd <code>&lt;tr&gt;</code> elements

## jQuery

- jQuery Code wird normalerweise in der `$(document).ready` Funktion gestartet
- Diese wird aufgerufen sobald das ganze HTML Dokument inklusive JavaScript Dateien geladen sind
- ```
$(document).ready(function(){  
    // jQuery-Code  
    $("#test").css("color", "#FFFFFF");  
});
```

# jQuery – DOM Traversieren

- `<div>` ist Vater (**parent**) von `<ul>` und Vorfahr (**ancestor**) von sämtlichen Elementen innerhalb von `<div>`
- `<ul>` ist Vater (**parent**) von beiden `<li>` und Kind (**child**) von `<div>`
- Linke `<li>` ist Vater (**parent**) von `<span>` und Kind (**child**) von `<ul>` und Nachkomme (**descendant**) von `<div>`
- `<span>` ist Kind (**child**) vom linken `<li>` und Nachkomme (**descendant**) von `<ul>` und `<div>`
- Die beiden `<li>` sind Geschwister (**siblings**)
- Rechte `<li>` ist Vater (**parent**) von `<b>` und Kind (**child**) von `<ul>` und Nachkomme (**descendant**) von `<div>`
- `<b>` ist Kind (**child**) vom rechten `<li>` und Nachkomme (**descendant**) von `<ul>` und `<div>`





## jQuery – DOM Traversieren

- DOM aufwärts traversieren

| Method         | Description                                               |
|----------------|-----------------------------------------------------------|
| parent()       | Returns the direct parent element of the selected element |
| parents()      | Returns all ancestor elements of the selected element     |
| parentsUntil() | Returns all ancestor elements between two given arguments |

- DOM abwärts traversieren

| Medhod     | Description                                         |
|------------|-----------------------------------------------------|
| children() | Returns all direct children of the selected element |
| find()     | Returns descendant elements of the selected element |

## jQuery – DOM Traversieren

- DOM seitwärts traversieren

| Method      | Description                                                       |
|-------------|-------------------------------------------------------------------|
| siblings()  | Returns all sibling elements of the selected element              |
| next()      | Returns the next sibling element of the selected element          |
| nextAll()   | Returns all next sibling elements of the selected element         |
| nextUntil() | Returns all next sibling elements between two given arguments     |
| prev()      | Returns the previous sibling element of the selected element      |
| prevAll()   | Returns all previous sibling elements of the selected element     |
| prevUntil() | Returns all previous sibling elements between two given arguments |

# jQuery – DOM Traversieren

- DOM Filter

| Method   | Description                                                                                     |
|----------|-------------------------------------------------------------------------------------------------|
| first()  | Returns the first element of the selected elements                                              |
| last()   | Returns the last element of the selected elements                                               |
| eq()     | Returns an element with a specific index number of the selected elements                        |
| filter() | Reduce the set of matched elements to those that match the selector or pass the function's test |
| not()    | Remove elements from the set of matched elements                                                |

# JavaScript

# Übungen

- Übung 1.1
- Übung 1.2
- Übung 1.3



## jQuery – DOM Manipulation

- Get/Set Content
  - `text()` - Sets or returns the text content of selected elements
  - `html()` - Sets or returns the content of selected elements (including HTML markup)
  - `val()` - Sets or returns the value of form fields
- Set Attributes
  - `attr()` - set/change attribute values

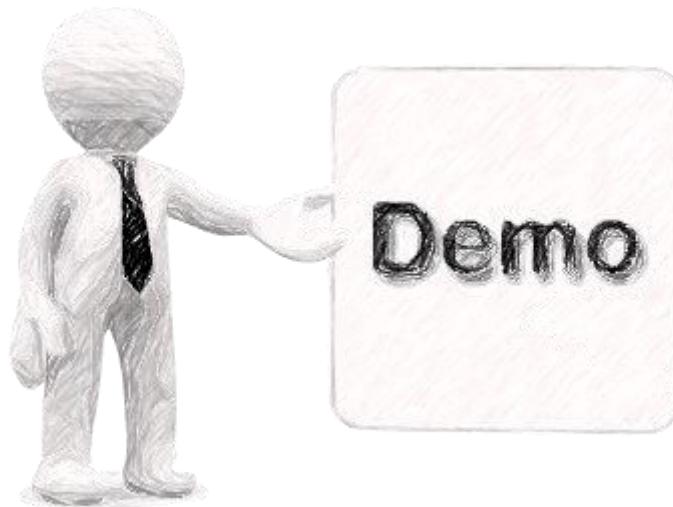
## jQuery – DOM Manipulation

- Add New HTML Content
  - `append()` - Inserts content at the end of the selected elements
  - `prepend()` - Inserts content at the beginning of the selected elements
  - `after()` - Inserts content after the selected elements
  - `before()` - Inserts content before the selected elements
- Remove Elements/Content
  - `remove()` - Removes the selected element (and its child elements)
  - `empty()` - Removes the child elements from the selected element

## jQuery – DOM Manipulation

- jQuery Manipulating CSS
  - `addClass()` - Adds one or more classes to the selected elements
  - `removeClass()` - Removes one or more classes from the selected elements
  - `toggleClass()` - Toggles between adding/removing classes from the selected elements
  - `css()` - Sets or returns the style attribute

# jQuery – DOM Manipulation





# JavaScript

# jQuery

- Einfache Handhabung von Komplexen Aufgaben mittels JavaScript
- Sehr gute Kapselung von AJAX
- Viele vorgefertigte Objekte, insbesondere auch UI-Elemente
- Sehr aktive Community mit vielen Lösungen für Standardfälle
- Ist auch in einer Mobile-Version verfügbar



# Übungen

- Übung 1.4 Eventmanagement



# JavaScript

# Web Forms

```
<form>
  First name:<br>
  <input type="text" name="firstname">
  <br>
  Last name:<br>
  <input type="text" name="lastname">
</form>
```

```
<form>
  <input type="radio" name="sex" value="male" checked>Male
  <br>
  <input type="radio" name="sex" value="female">Female
</form>
```

# Web Forms

```
<form action="action_page.php">
  <fieldset>
    <legend>Personal information:</legend>
    First name:<br>
    <input type="text" name="firstname" value="Mickey">
    <br>
    Last name:<br>
    <input type="text" name="lastname" value="Mouse">
    <br><br>
    <input type="submit" value="Submit">
  </fieldset>
</form>
```

- Weitere Informationen zu Web Forms können hier gefunden werden
  - [http://www.w3schools.com/html/html\\_forms.asp](http://www.w3schools.com/html/html_forms.asp)
  - <http://www.html5rocks.com/de/tutorials/forms/html5forms/>

## jQuery - Forms

- Um Forms zu validieren gibt es ein eigenes jQuery Plugin welches unter folgendem Link gefunden werden kann
  - <http://jqueryvalidation.org/documentation/>
- Das Plugin ermöglicht das einfache Validieren von Formularen
- Erklärungsvideo
  - <http://jqueryvalidation.org/>

## jQuery - Forms

- Die API des jQuery Validation Plugins hat folgende Möglichkeiten
  - `Validate()`  
Das selektierte Formular wird validiert
  - `valid()`  
Überprüft ob das Formular als gesamtes gültig ist oder nicht
  - `rules()`  
Kann so neue regeln hinzufügen oder wieder entfernen

## jQuery – Forms Validation

- required – Makes the element required.
- remote – Requests a resource to check the element for validity.
- minlength – Makes the element require a given minimum length.
- maxlength – Makes the element require a given maximum length.
- rangelength – Makes the element require a given value range.
- min – Makes the element require a given minimum.
- max – Makes the element require a given maximum.
- range – Makes the element require a given value range.
- email – Makes the element require a valid email
- url – Makes the element require a valid url
- date – Makes the element require a date.
- dateISO – Makes the element require an ISO date.
- number – Makes the element require a decimal number.
- digits – Makes the element require digits only.
- creditcard – Makes the element require a credit card number.
- equalTo – Requires the element to be the same as another one

## jQuery – Forms Validation

- Es müssen Rules definiert werden um festzulegen wie die entsprechenden Felder validiert werden sollen.
- Die Validationsregeln können jedoch auch im HTML festgelegt werden

```
$(".selector").validate({
  rules: {
    // simple rule, converted to {required:true}
    name: "required",
    // compound rule
    email: {
      required: true,
      email: true
    }
  }
});
```

```
$(".selector").validate({
  rules: {
    contact: {
      required: true,
      email: {
        depends: function(element) {
          return $("#contactform_email").is(":checked");
        }
      }
    }
  }
});
```



## jQuery – Forms Validation

- Es können auch verschiedene Validationsregeln für ein Feld gelten
- Die anzuzeigenden Validationstexte können frei gewählt

```
$(".selector").validate({  
  rules: {  
    name: "required",  
    email: {  
      required: true,  
      email: true  
    }  
  },  
  messages: {  
    name: "Please specify your name",  
    email: {  
      required: "We need your email address to contact you",  
      email: "Your email address must be in the format of name@  
    }  
  }  
});
```

## jQuery – Forms Validation

- Die anzuzeigenden Validationstexte können frei gewählt werden

```
$(".selector").validate({  
  rules: {  
    name: {  
      required: true,  
      minlength: 2  
    }  
  },  
  messages: {  
    name: {  
      required: "We need your email address to contact you",  
      minlength: jQuery.validator.format("At least {0} characters")  
    }  
  }  
});
```

## jQuery – Forms Validation

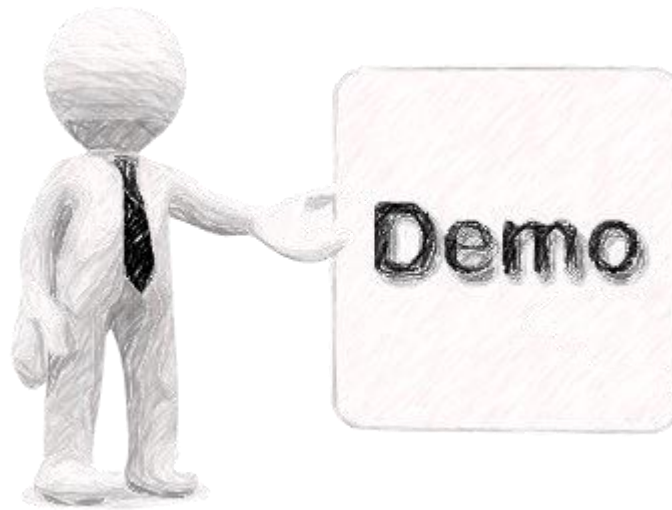
- Es können auch alle Fehler an einem Ort angezeigt werden

```
$(document).ready(function() {  
    $("#form1").validate({  
        errorLabelContainer: $("#form1 div.error")  
    });  
  
    var container = $('#div.container');  
    // validate the form when it is submitted  
    var validator = $("#form2").validate({  
        errorContainer: container,  
        errorLabelContainer: $("ol", container),  
        wrapper: 'li'  
    });  
  
    $(".cancel").click(function() {  
        validator.resetForm();  
    });  
});
```

## jQuery – Forms Validation

<http://jqueryvalidation.org/files/demo/>

<http://jqueryvalidation.org/files/demo/errorcontainer-demo.html>

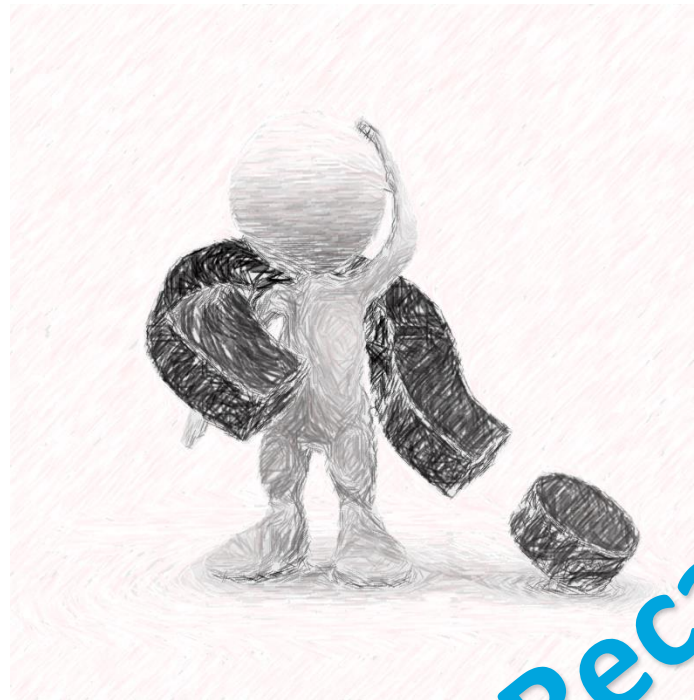


# JavaScript

# Übungen

- <http://www.sitepoint.com/basic-jquery-form-validation-tutorial/>





Recap