

jQuery: Einführung

- Vorzüge von jQuery erkennen und Arbeitsweise verstehen
- Einfache Manipulationen in Web-Seiten vornehmen können
 - Elementselektion
 - Eigenschaften- und CSS-Manipulation
 - Elementmanipulation und Elemente zur Seite hinzufügen
 - Einfache Animationen
 - Ereignisbehandlung
- Grundlegende Funktionen von jQuery verstehen und anwenden können
- jQuery in ein Web-Projekt integrieren können
- Eclipse anpassen können

Literaturhinweis

jQuery, Das umfassende Handbuch, Bongers, Vollendorf, Lange,
Rheinwerk Computing, ISBN 978-3-8362-5664-3

jQuery ist...

eine Erweiterung der Programmiersprache JavaScript um

- Elemente oder Elementgruppen im Dokumentenbaum leicht ansprechen zu können
- gefundene Elemente zu manipulieren, neue Elemente hinzufügen, Attribute hinzufügen, CSS manipulieren
- Daten im JSON-Format über Ajax in bestehendes Dokument einzufügen, ohne Dokument neu zu laden
- komplexe Animationen einfach einzufügen
- Event-Handling einfach einzufügen
- komfortable Bedienelemente einzufügen (Datepicker, Dropdown-Menüs, Tabs, Akkordeon-Elemente)
- Erweiterungen hinzuzufügen (jQuery UI, jQuery Mobil)
- Überbrückung von Browser-Unterschieden und JavaScript-Versionen vorzunehmen

denn reines JavaScript ist in seiner Bedienung zu umständlich und fehleranfällig

JavaScript kann bei jQuery weiterhin verwendet werden

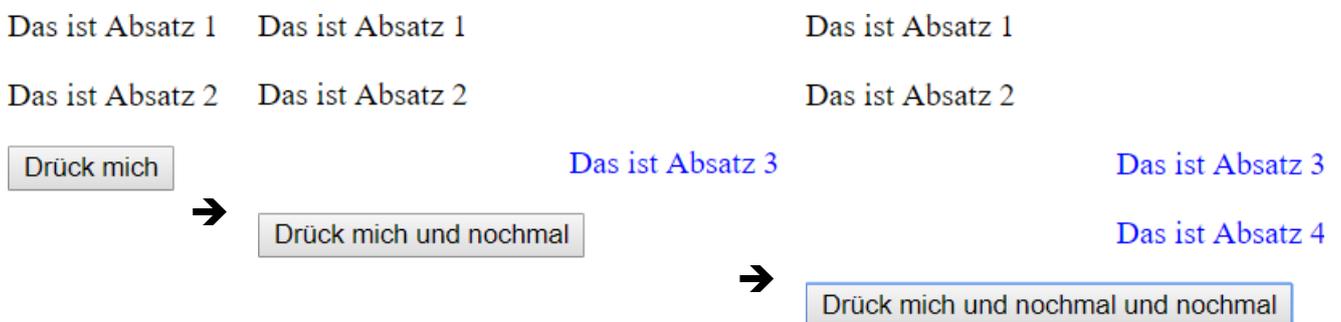
Beispiel

```

<!DOCTYPE html>
<html>
<head>
  ...
  <script type="text/javascript" src="lib/jquery-3.3.1.min.js">
  </script>
  <script type="text/javascript">
    $(document).ready(function() {
      $("#buttonId").click(function() {
        var p = $("<p>")
          .text("Das ist Absatz " + ($("#p").length + 1))
          .attr("align", "right")
          .css("color", "blue");
        $("#div").append(p);
        $("p")
          .slideUp()
          .slideDown();
        $("#buttonId").val($("#buttonId").val() + " und nochmal");
      });
    });
  </script>
</head>
<body>
  <div>
    <p>Das ist Absatz 1</p>
    <p>Das ist Absatz 2</p>
  </div>
  <input id="buttonId" type="button" value="Drück mich">
</body>
</html>

```

text() liefert alle Textinhalte auch der Kindelemente oder setzt diese
 val() liest und setzt value-Eigenschaft
 attr(), css() liest und setzt HTML-, CSS-Eigenschaft



WICHTIG: Viele Methoden geben Objekt auf welche sie aufgerufen werden wieder zurück. Dadurch Verkettbarkeit (*Chaining*) möglich
 var definiert lokale Variable, ohne var wird Variable global

Vorbereitungsarbeiten

Term Plugin in Eclipse installieren

Ermöglicht Code-Vervollständigung und integriert Dokumentation

1. Befehl Help. Install new Software. Add <http://oss.opensagres.fr/tern.repository/1.2.1/>
2. Im *Kontextmenü* des *Dynamischen Web-Projektes* unter Configure. Convert to Tern Project... unter Modules jQuery auswählen

jQuery herunterladen

- *Komprimierte Version* oder *Entwickler-Version* mit gesamten Quelltext
- Version 3 unterstützt Chrome ab Version 60, Edge ab Version 39, Firefox ab Version 35, Internet Explorer ab Version 9, Safari ab Version 10 und Opera ab Version 45
- Für ältere Browser-Versionen gibt es Migrationstools
- Lokale Einbindung (siehe vorne) oder über Google Online Repository

```
<script type="text/javascript"
  src="http://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
```

Zur Fehlersuche im Browser Funktionstaste [F12]

Entwicklertools mit Console, DOM-Explorer, Debugger, usw.

Funktionsweise von jQuery

```
<style type="text/css" media="screen">
  .green { color:#009933; }
  .blue { color:#0000FF; }
  .red { color: #CC0000; }
</style>
```

Erster Absatz

Zweiter Absatz

Dritter Absatz

```
<script type="text/javascript">
  $(document).ready(function() {
    $("#box p:contains('Zweiter Absatz')")
      .addClass("green");
    $(".start").addClass("red");
    $("#box p").click(function() {
      $(this).removeClass()
        .addClass("blue");
    });
  });
</script>
```

```
$("#box")    id="box"
$(".start")  class="start"
```

```
<div id="box">
  <p class="start">Erster Absatz</p>
  <p>Zweiter Absatz</p>
  <p>Dritter Absatz</p>
</div>
```

\$() Factory-Funktion

bildet aus Selektor (z.B. document oder #box p:contains()) *jQuery-Objekte* und stattet diese mit besonderen Funktionen aus (z.B. ready(), addClass())

Selektoren

legen fest, welche Elemente des DOMs zu jQuery-Objekten werden

\$() liefert alle jQuery-Objekte die Selektor identifiziert. Nachfolgende Methoden werden auf alle Objekte ausgeführt

ready() wird gestartet, wenn Dokument fertig geladen ist

Anonyme Funktion function() { ... } wird als Argument übergeben (*Callback-Funktion*)

Anstelle von \$(document).ready(function() { ... }); kann auch \$(function() { ... }); verwendet werden

Beispiel für HTML-Inhalte hinzufügen und Chaining

Inhaltsverzeichnis

- Erste Überschrift
- Zweite Überschrift
- Dritte Überschrift

Erste Überschrift

Text zur ersten Überschrift

Zweite Überschrift

Text zur zweiten Überschrift

Dritte Überschrift

Text zur dritten Überschrift

Inhaltsverzeichnis

- Erste Überschrift
- Zweite Überschrift
- Dritte Überschrift

Erste Überschrift

Text zur ersten Überschrift

Zweite Überschrift

Text zur zweiten Überschrift

Dritte Überschrift

Text zur dritten Überschrift

Inhaltsverzeichnis

- Erste Überschrift
- Zweite Überschrift
- Dritte Überschrift

Erste Überschrift

Text zur ersten Überschrift

Zweite Überschrift

Text zur zweiten Überschrift

Dritte Überschrift

Text zur dritten Überschrift

```
h1 { cursor: pointer; }
.green { color:#009933; background-color:#E2FFEC; cursor:default;}
.boxColor-0 { background-color: #CCCCCC; }
.boxColor-1 { background-color: #EEEEEE; }
.boxColor-2 { background-color: #999999; }
.content { background-color: #FFFF99; }
```

```
$(function() {
  var obj = $("#box h1");
  var contents = $("<div>");
  contents.attr("class","content");
  contents.html("<h1>Inhaltsverzeichnis</h1>");
  contents.insertBefore(obj.first());
  var content = "<ul>";
  obj.each(function(i) {
    content += "<li>" + $(this).text() + "</li>";
  });
  content += "</ul>";
  contents.append(content);
  obj.click(function() {
    obj.removeClass()
      .parent()
      .removeClass();
    $(this).addClass("green")
      .parent()
      .addClass("boxColor-" + obj.index($(this)));
  });
});
```

html() liefert oder setzt HTML-Inhalt
 each() durchläuft alle Objekte einer
 Liste, i gibt Stelle an
 insertBefore() und first()
 append()-Varianten

```
<div id="box">
  <h1>Erste Überschrift</h1><p>Text zur ersten Überschrift</p>
  ...
</div>
```