JavaScript am AEG

Paul Pasler | AEG Reutlingen, Stefan Gaum, Kursstufe

Über mich

- Paul Pasler, 27, Degerschlacht

- 2007: Abi am AEG
- 2010: Ausbildung zum Flachdrucker
- 2013: Bachelor Medieninformatik an der Hochschule RheinMain
- Ab 2013: Master „Human Centered Computing“ an der Hochschule Reutlingen

- Seit 2012 Nebenjob bei //SEIBERT/MEDIA
- Entwicklung einer Pluginsuite für Confluence
-

- Hobbies: Fußball, Hunde, Käfer schrauben

Was euch erwartet

- Lektion 1: Einführung
- Lektion 2: JavaScript Grundlagen
- Lektion 3: HTML-Manipulation I
- Lektion 4: HTML-Manipulation II – jQuery
- Lektion 5: Wünsch Dir was!

- Alle Lektionen findet ihr im 

Wiki

Was ihr braucht

Euren Kopf und einen Browser



Tatsächlich ist es nicht mehr :)

Hochschule Reutlingen
Reutlingen University

INF
Fakultät
Informatik
Reutlingen

JavaScript am AEG

Lektion 1: Einführung

Paul Pasler | AEG Reutlingen, Stefan Gaum, Kursstufe

Was ist eigentlich JavaScript (JS)?

- Skriptsprache für dynamisches HTML in Webbrowsern
  - Auswerten von Benutzerinteraktion
  - Verändern, Nachladen, Generieren von Inhalt
- JavaScript wird auf ca. 90% aller Websites genutzt

- JavaScript hat nichts (mehr) mit der Programmiersprache Java zu tun!

http://de.wikipedia.org/wiki/JavaScript#/media/File:Unofficial_JavaScript_logo_2.svg

Verwendung

- JavaScript wird Client-Seitig im Browser ausgeführt
- Einfache Aufgaben werden mit JS realisiert
  - Formular Validierung
  - Email-Adressen Verschlüsselung
  - Dialog Fenster Anzeige
- Dynamisches Nachladen von Inhalten (AJAX)

Verwendung: Beispiele

- Bilderslidern
- Interaktiven Tabellen
- Diagramme
- Präsentationen
- Animationen

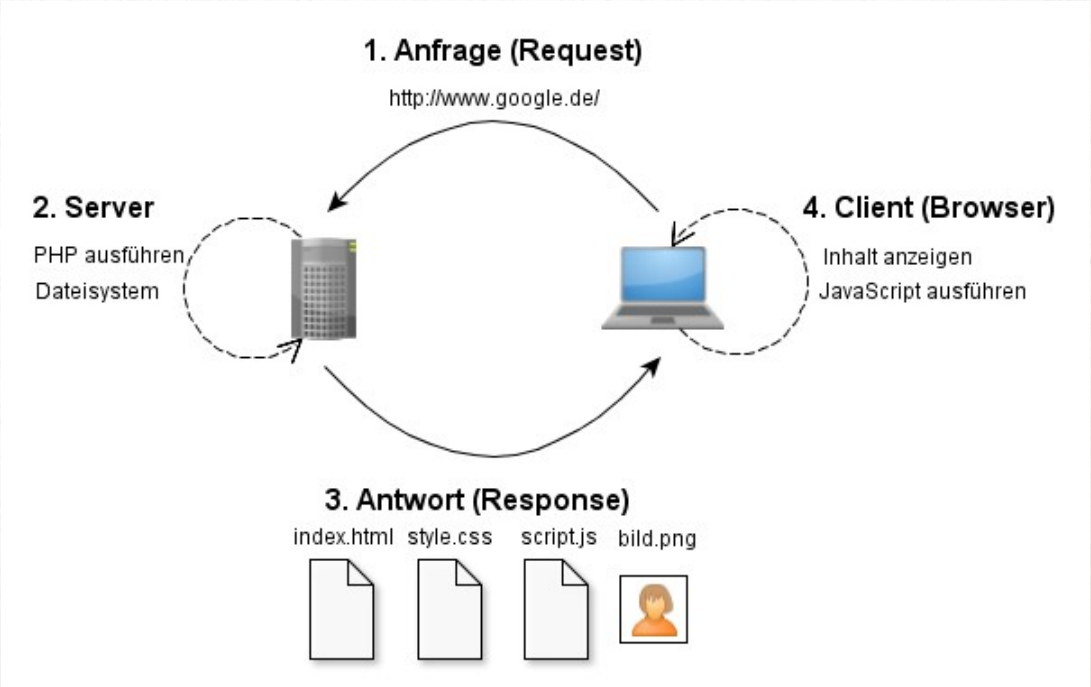
Lektion 5: Wünsch dir was

- Was wollt noch kennenlernen?
    - JavaScript Funktionen
    - REST-Ful Webservice
- Bibliotheken
    -

*API = Application Programming Interface (Programmierschnittstelle)

Funktionsweise

- JS wird Client-Seitig auf dem Browser ausgeführt
- Unterschied zu PHP!



Funktionsweise: Vor- und Nachteile

Vorteile

- Weniger Last auf dem Server
- Einstellungen des Clients werden berücksichtigt (Bspw. Fensterbreite)
- Sofortige Rückmeldung bei Falscheingaben (Formularvalidierung)

Nachteile

- Mehr Last beim Client
- Je nach System, längere Ladezeiten (Smartphones)
- Veränderung der Seite während des Ladevorgangs
- Ausführung hängt vom System / Browser ab

Sicherheit

- Sensible Daten werden mittlerweile häufig über das Internet verschickt (Bspw. Banking)
- Eine Seite mit Schadcode könnte sehr gefährlich sein
- JS wird daher in einer sog. „Sandbox“ und somit isoliert ausgeführt
- JS hat keine Zugriff auf andere Tabs oder gar das Dateisystem

- Dennoch kann JS PopUps anzeigen, auf andere Seiten weiterleiten, Browserfenster schließen oder ungewollt Schadcode enthalten
- Der beste Schutz ist aufpassen und immer aktuelle Browser zu verwenden!

Geschichte

- 1995 entwickelte Brendan Elch den JS-Vorläufer „LiveScript“
- Der Name JavaScript, kommt von dem Plan JavaApplets und LiveScript zu Verbinden
- 1998 erste Standardisierung nach ECMA und ISO
- Seitdem ständige Weiterentwicklung

Beispiel: Hello World

- JavaScript Code kann direkt im <head>-Bereich einer HTML-Seite eingebettet werden

```
<html>
  <head>
    <script>
      ... dein Code hier ...
    </script>
  </head>
  <body>
    ... weitere HTML-Elemente ...
  </body>
</html>
```

- „Hello World“-Ausgabe im Browserfenster

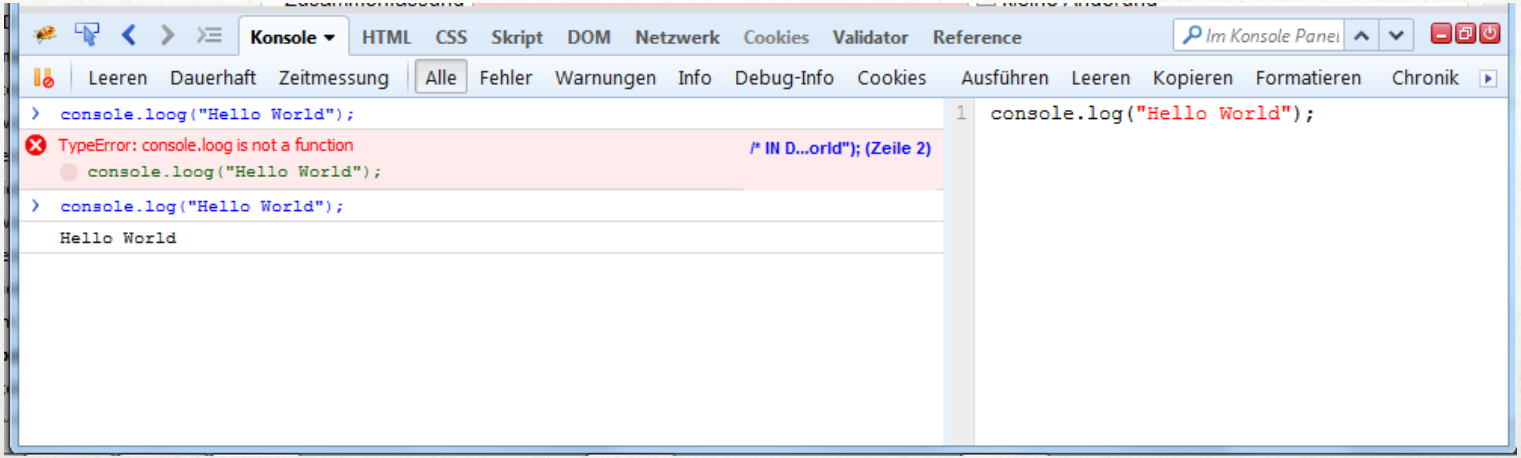
```
<script>
  alert("Hello World!");
</script>
```

- „Hello World“-Ausgabe in der JavaScript Konsole

```
<script>
  console.log("Hello World!");
</script>
```

Die JavaScript Konsole

- Moderne Browser (FireFox, Chrome) und InternetExplorer stellen mittlerweile eine Entwicklerkonsole zur Verfügung (Taste „F12“)
- Dort gibt es Anzeigen für HTML, CSS, Netzwerk usw.
- Wichtig für uns, die JavaScript Konsole



Aufgaben

- Öffnet einen Browser eures Vertrauens
- Untersuche deine Lieblingsseite mit der JavaScript Konsole
- Macht euch mit der Entwicklerkonsole und ihren Funktionen vertraut

- Erstellt in eurem Arbeitsverzeichnis einen Ordner „JavaScript/Lektion_1“
- und dort eine Datei „index.html“ mit dem Inhalt von Folie 13
- Öffnet die HTML Datei im Browser und führt das „Hello World“ Beispiel aus

Hochschule Reutlingen
Reutlingen University

INF
Fakultät
Informatik
Reutlingen

JavaScript am AEG

Lektion 2: Grundlagen JavaScript

Paul Pasler | AEG Reutlingen, Stefan Gaum, Kursstufe

Lektion 2: Grundlagen JavaScript

- JavaScript Code kann in der Entwicklerkonsole ausgeführt werden
- Nach jeder Folie könnt / sollt ihr gleich in der Konsole ausprobieren
- Wenn JS in der Konsole ausgeführt wird, wird nur der Wert der letzten Funktion ausgegeben.

```
5 + 5;
5 * 5;
// Konsolenausgabe 25
```

- Darum console.log() verwenden

```
console.log(5 + 5);
// Konsolenausgabe 10
console.log(5 * 5);
// Konsolenausgabe 25
```

Operatoren

- Für Berechnungen stehen bekannt Rechenoperatoren zur Verfügung „ +, -, *, / “
- Zudem gibt es den Modulo Operator „ % “ (Rest der ganzzahligen Division)

Aufgabe: Rechnen in der Konsole

5 + 5	5 * 5
5 – 5	25 / 5
26 / 5	25 % 5
26 % 5	

Datentypen

Es gibt in JS fünf Datentypen

- String (Text)
  „Hallo“, „1 2 - Polizei “
- Number (Zahl)
  17, 3.14
- Boolean (Wahrheitswert)
  true, false
- Array (Liste)
  [1, 2, 3], ['a', 'b', 'c']
- Object – kommt später

- Den Datentyp kann man mit *typeof* prüfen
  typeof(1) ergibt number

Aufgabe: Gib die Typen der gelernten Datentypen in der Konsole aus

13	false
2.71	[4, 5, 6]
'Hallo'	['x', 'y', 'z']
'3, 4 – Offizier'	[1, 'a', true]

Was fällt auf?

Datentypen: Besonderheiten

- Strings können mit “ oder ' umschlossen werden
  Wollt ihr Anführungszeichen in einem String verwenden geht das so

```
' Good "bye" ';
" Good 'bye' ";
```

- JS kann Strings addieren (konkatenieren, zusammenfügen)

```
'Hallo ' + ' AEG';
// ergibt 'Hallo AEG'
```

- JS kann aber auch Zahlen und Strings addieren

```
17 + 'Hallo ';
// ergibt '17Hallo'
```

Aufgabe: Addiere Strings und Zahlen

13 + ' AEG'	AEG + '13'
1 + 2 + 'AEG'	'AEG' + 2 + 1

Was fällt auf?

JavaScript am AEG

Lektion 3: HTML-Manipulation I

Paul Pasler | AEG Reutlingen, Stefan Gaum, Kursstufe

Lektion 3: HTML-Manipulation I

- Die wichtigste Eigenschaft von JavaScript ist die Manipulation einer HTML-Seite
- HTML-Elemente können hinzugefügt, verändert und gelöscht werden
- Aussehen und CSS-Eigenschaften können ebenfalls verändert werden
- Übungsdatei im Wiki


```
<body>
  <!-- Sichtbarer Dokumentinhalt im body -->
    <h1>JavaScript @ AEG: Lektion 2</h1>
    <a href="index.html" id="important" style="color: #f00;">
      Hello World
    </a>
    <p>
      <tt id="mono">
        &lt;a href=&quot;index.html&quot; id=&quot;important&quot; style=&quot;color: #f00;&quot;&gt;Hello
World&lt;/a&gt;
      </tt>
    </p>
</body>
```

Wiederholung: HTML-Element

```
<a href="index.html" id="important" style="color: #f00;">Hello World</a>
```

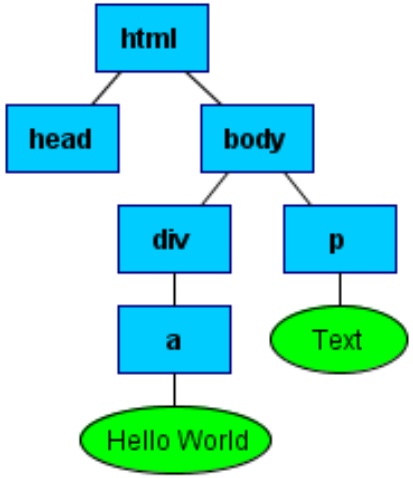
- **<a ...>...**
  HTML-Element (<a>-Tag)
- **href**
  Attribut mit dem Wert „index.html“
- **id**
  id-Attribut (Sollte eindeutig auf einer Seite sein)
- **style**
  Style-Attribut mit CSS-Anweisungen
- **„Hello World“**
  Inhalt des <a>-Tags (Können auch wieder HTML-Elemente sein <a> ...)

Weitere Infos unter: Das HTML DOM Element Objekt

Wiederholung: HTML-Dokument

- Das <body>-Tag ist der „Elternknoten“ des - und <a>-Tags, anders herum sind diese beiden Kindknoten vom <body>-Tag
- Das <body>-Tag ist aber nur der direkte Elternknoten des
- Das -Tag ist wiederum der direkte Elternknoten des <a>
- Das <p>-Tag liegt auf der selben Ebene, wie das -Tag

```
<html>
  <body>
    <div>
      <a href="index.html" id="important" style="color: #f00;">
        Hello World
      </a>
    </div>
    <p>
      Text
    </p>
  </body>
</html>
```



Das document-Objekt

- Spezielles JavaScript Objekt, das immer beim Laden einer HTML-Seite erzeugt wird
- Es ist der Wurzelknoten aller HTML-Elemente
- Von ihm aus lassen sich alle HTML-Elemente erreichen

- Oberhalb vom document-Objekt befindet sich das window-Objekt
- Das window-Objekt enthält Infos zum aktuellen Browserfenster
    - Breit und Höhe
    - Historie
- Weitere Infos unter: http://www.w3schools.com/

Kommentare in JavaScript

- Einzeiliger Kommentar

```
// Einzeiliger Kommentar
// Noch einer
```

- Mehrzeiliger Kommentar

```
/*
Mehr-
zeiliger
Kommentar
*/
```

Eigene Script-Datei einbinden

- Eine JavaScript-Datei hat die Endung *.js
- Eigene Skripte können im <head> eingebunden werden (Wie CSS-Dateien)

```
<head>
    <script type="text/javascript" src="pfad/zur/datei.js"></script>
</head>
```

Aufgabe:
- Erstelle eine JS Datei „script.js“ und füge einen console.log Befehl ein
- Füge die script.js zur HTML-Datei hinzu

Inhalt eines HTML-Element verändern

Problem: HTML finden und gezielt ansprechen

Lösung: Ansprechen über das id-Attribut

```
document.getElementById('id-name').innerHTML
```

Das bedeutet folgendes

- Suche im gesamten HTML-Inhalt (document)
- nach einem HTML-Element mit der id „important“ (getElementById(„important“))
- und hole die Inhalts-Eigenschaft (innerHTML).

Zuweisung mit = hinter dem Ausdruck

Aufgabe:
- Lese den Wert von Element mit der id „important“ aus
- Ändere den Wert des Elements

Einschub: HTML fertig geladen

- Wann JavaScript ausgeführt wird, ist leider NICHT festgelegt
- Es kann vorkommen, dass JS ausgeführt wird,
  obwohl noch nicht das gesamte HTML gerendert (und somit vorhanden) ist
- Man kann dem Browser aber mitteilen, dass man so lange warten möchte

```
document.addEventListener( "DOMContentLoaded", function(){
    console.log("HTML ready!!!");
    // Wird erst ausgeführt, wenn das HTML fertig ist
});

console.log("HTML ready???");
// Nicht sicher wann das ausgeführt wird
```

Aufgabe:
- Baue diese Funktion in dein Script ein und teste

HTML-Attribut ändern

- Ändern von Eigenschaften eines HTML-Elementen
- Attribute sind Schlüssel / Wert Paare innerhalb des öffnenden HTMLElement-Tags

  Beispiele: `src, target, class oder style`
- Funktioniert wie bei der innerHTML-Eigenschaft

```
document.getElementById("important").href = "neuer Link";
document.getElementById("important").target = "neues Target";
```

Aufgabe:
- Ändere das Link-Ziel des „important“ Elements auf „index.html“
- Füge dem Link die Eigenschaft inzu, sodass sich der Link in einem neuen Tab öffnet

Variablen in JavaScript

- Variablen verweisen auf Werte und Objekte
- Mit dem Schlüsselwort „var“ und einem Variablennamen wird sie deklariert
  mit einem = und einem Wert initialisiert

```
var x;      // Deklaration
x = 5;      // Initialisierung
var y = 3; // Beides
var a, b;  // Deklaration zweier Variablen
```

- Auch HTML-Element lassen sich so speichern

```
var element = document.getElementById("important");
```

Aussehen eines HTML-Elements verändern (CSS)

- Wichtige und sehr nützlich Funktion
- Einfache Syntax

```
var element = document.getElementById("important");

// Setzt die Schriftgroesse auf 25px
// document.getElementById("important").style.fontSize = "25px";
element.style.fontSize = "25px";

// Versteckt das Element (Das Element ist aber noch da!)
element.style.display = "none";
```

Aufgabe:
- Schreibe den Link-Text in Fett
- Versteck den Link und zeige ihn dann wieder an

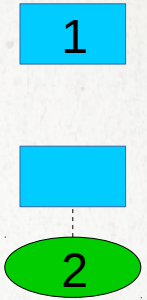
HTML-Element hinzufügen

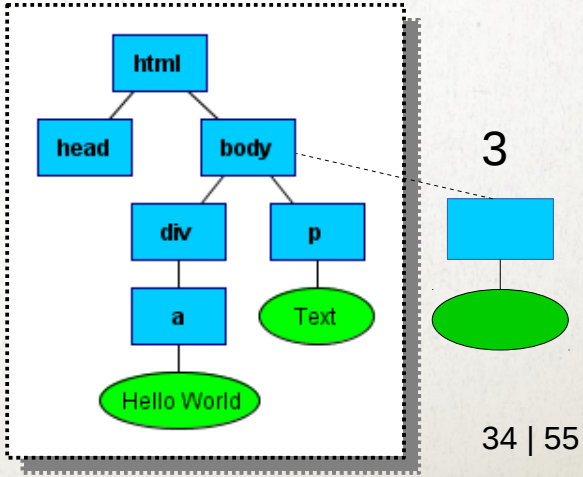
- Um ein HTML-Element hinzuzufügen muss folgendes getan werden:

1) Das neue Element muss erstellt werden

2) Das Element bekommt einen Inhalt

3) Das Element braucht einen Eltern-Knoten und wird als Kind-Knoten eingefügt





HTML-Element hinzufügen

```
// Speichere das HTML-Element in der Variable "newElement"
var newElement = document.createElement("h2");

// Fülle den Inhalt von newElement mit "Hallo Neue Welt"
newElement.innerHTML = "Hallo Neue Welt";

// Weise newElement die id newElementId zu
newElement.id = "newElementId";

// Füge das neue Element unterhalb vom <body>-Tag als letztes Element ein
document.body.appendChild(newElement);
```

Aufgabe:

- Füge einen neuen Link, mit der id „wichtig“, der auf das AEG Wiki zeigt (unterhalb des <body>-Tags)

HTML-Element löschen

- Im Unterschied zum Ausblenden mit CSS (`display: none`),
  kann ein HTML-Element auch gelöscht werden
- Dafür wird das Element von seinem Elternknoten gelöst und ist somit verschwunden
  - Im einfachsten Fall ist der <body> das Eltern-Element
  - Andernfalls muss das Eltern-Element erst geholt werden
    (Gilt auch für das Einfügen)

HTML-Element löschen

- <body> als Eltern-Element

```
var element = document.getElementById("important");

document.body.removeChild(element);
```

- Beliebiges Eltern-Element (kann auch der <body> sein)

```
var element = document.getElementById("important");
Var parent  = element.parentElement

parent.removeChild(element);
```

Aufgabe:
- Lösche den gerade erstellten Link
- Lösche das tt-Element mit der id „mono“

Aufgaben

1) Ändere den Text des Links auf „Hallo Welt“
2) Ändere das Linkziel des Hello World Link auf „http://www.spiegel.de“
3) Füge dem Link die CSS-Eigenschaft fett hinzu
4) Füge dem body ein neues <p>-Element mit der id „neu“ hinzu
5) Lösche das neu erstellte <p>- Element wieder

6) Ersetze das <tt>-Tag durch ein -Tag mit gleichem Inhalt
7) Ermittle die Breite und Höhe deines Browserfensters und gib sie in der Konsole aus.
   Tipp: Das window-Element kann helfen.

JavaScript am AEG

Lektion 4: HTML-Manipulation II – jQuery

Paul Pasler | AEG Reutlingen, Stefan Gaum, Kursstufe

Lektion 4: HTML-Manipulation II - jQuery

- Informatiker sind von Haus aus faul!
- Ziele
    - Einfache Lösungen für aufwändige Probleme
    - Redundanzen (Wiederholungen) vermeiden
    - Code-Zeilen sparen
- Lösung: Funktionalität in Bibliotheken (Erweiterungen) kapseln

jQuery

- jQuery ist eine Erweiterung für JavaScript
  - Vereinfacht Aufrufe
  - Stellt neue Funktionalität zur Verfügung
- jquery.com

„jQuery is used by 63.8% of all the websites,
that is a JavaScript library market share of 95.1%“ *

* Quelle: http://w3techs.com/technologies/overview/javascript_library/all

jQuery einbinden

- jQuery ist nicht standardmäßig im Browser verfügbar und muss eingebunden werden
- Genau wie eigene Skripte, können auch Bibliotheken im <head> definiert werden
  - Entweder auf das Dateisystem heruntergeladen
  - Direkte Einbindung als Link

```
<head>
    <script type="text/javascript" src="pfad/zur/datei.js"></script>
    <script type="text/javascript" src="http://url/datei.js"></script>
</head>
```

Aufgabe:
- Lade dir die aktuelle jQuery Datei (Version 1.11.3) auf jquery.com herunter
- Binde jQuery in deine HTML-Seite ein

jQuery Features

- jQuery kapselt (überdeckt) JS-Funktionalität und bringt eine eigene Syntax mit
- jQuery-Funktionen beginnen mit „jQuery“ oder einem „$“
- Einfacher Zugriff auf HTML-Elemente mit CSS Selektoren
  - „ # “: id
  - „ . “: class
  - „tag“: <tag>

```
// jQuery Schreibweise für:
// document.getElementById("important");
   jQuery("#important");
   $("#important");
// deutlich kürzer :)
```

Aufgabe:
- Greife auf das HTML-Element mit der Id „important“ zu und gibt den Rückgabewert in der Konsole aus (JS und jQuery)

HTML fertig geladen - jQuery

- jQuery bringt eine eigene Funktion zur Abfrage, ob das HTML fertig geladen ist

```
// Wird erst ausgeführt, wenn das HTML fertig ist

document.addEventListener( "DOMContentLoaded", function(){
    // JS Code here
});

$('document').ready(function(){
    // JS Code here
});
```

Aufgabe:

- Füge die Funktion in deinem Script hinzu und teste welche der beiden schneller ist

Inhalt eines HTML-Element verändern

```
// Gibt "Hello World" in der Konsole aus
// console.log(document.getElementById("id").innerHTML)
   console.log($("#id").html());

// Ändert "Hello World" in "Goodbye World"
// document.getElementById("id").innerHTML = "neuer Text"
   $("#id").html("neuer Text");
```

Aufgabe:

- Lies den Wert vom Element mit der ID „important“ aus
- Ändere den Wert auf Goodbye World

HTML-Attribut ändern

```
// Ändert das Ziel des Links auf "index.html"
// (Das Attribute wird verändert)
// document.getElementById("id").href = "index.html";
   $("#id").attr("href", "index.html");

// Linkziel wird in neuem Fenster geöffnet
// (Das Attribut wird neu hinzugefügt)
// document.getElementById("id").target = "_blank";
   $("#id").attr("target", "_blank");

   $("#id").attr("foo", "bar");
// id ist übrigens auch nur ein Attribut!
```

Aufgabe: „important“ verändern

- Neues Link-Ziel: http://jquery.com
- Soll sich in neuem Fenster öffnen
- Denk dir ein eigenes Attribute aus, fülle es und lese es wieder aus

Aussehen eines HTML-Elements verändern (CSS)

```
// Speichere das HTML-Element in der Variablen "element"
var element = $("#id");

// Setzt die Schriftgroesse auf 25px
// element.style.fontSize = "25px";
   element.css("font-size", "25px");

// Versteckt das Element (Das Element ist aber noch da!)
// element.style.display = "none";
   element.hide();

// Zeigt das Element wieder an
// element.style.display = "block";
   element.show();

// Sogar mehrere CSS-Anweisungen in einem
// Text wird gelb und bekommt einen schwarzen Rahmen
element.css({color: "yellow", border: "1px solid #000"});
```

Aufgabe: „important“ verändern

- Textfarbe: Grün
- Fett geschrieben
- Verstecken und wieder anzeigen

HTML-Element hinzufügen

- Jetzt wird es richtig schön!

```
// var newElement = document.createElement("h2");
// newElement.innerHTML = "Hallo Neue Welt";
// document.body.appendChild(newElement);

// 2 Zeilen gespart
$("body").append("<h2>Hallo Neue Welt</h2>");

// Fügt das Element ganz am Anfang ein
$("body").prepend("<h2>Hallo Neue Welt von Oben</h2>");
```

Aufgabe:
- Füge eine <p> mit dem Text „unten“
- und eine <h2> mit dem Text „oben“ im <body> ein

HTML-Element löschen

```
// var element = document.getElementById("important");
// document.body.removeChild(element);

// Element holen und löschen (Elternelement ist egal)
$("#id").remove();
```

Aufgabe:

- Lösche das HTML-Element mit der id „important“

HTML-Element finden I

- Zugriff via Id, Klasse und <tag>-Typ

```
<body>
  <a href="index.html" id="link">Link</a>
  <h2 class="heading">Überschrift</h2>
  <div>
    <p>Text</p>
  </div>
</body>
```

```
console.log($("#link").html());
// Link

console.log($(".heading").html());
// Überschrift

console.log($("div").html());
// <p>Text</p>
```

Aufgabe:

- Greife auf das <h1> Element zu
- Erweitere deine HTML-Datei um die Elemente Test
- Greife auf das Element mit der Klasse „test“ zu

HTML-Element finden II

- Zugriff via Klasse und <tag>-Typ muss nicht eindeutig sein
- Potentiell gibt es mehrere Ergebnisse als Liste
- Bei einem Funktionsaufruf, wie .html(),
  wird das erste Element genommen

```
<body>
  <div>
    <p>Text 1</p>
  </div>
  <div class="second">
    <p>Text 2</p>
  </div>
</body>
```

```
console.log($("div").html());
// Text 1

console.log($("div"));
// [div, div.second] eine Liste von Elementen

console.log($("div.second").html());
// Text 2
```

Aufgabe:
- Erweitere deine HTML-Datei um die Elemente
  Test 2
- Greife auf das Element zu

Was fällt auf?

Was kann jQuery noch?

- Wie gesehen ist jQuery sehr mächtig
  - Suche von Elementen mit bestimmtem Html-Inhalt
  - Elemente langsam Ein- / Ausblenden
  - Daten im Hintergrund vom Server abrufen
  - Auf click und hover reagieren
  - ….
- Alle Funktionen findet ihr in der API* unter
  - http://api.jquery.com/

*API = Application Programming Interface (Programmierschnittstelle)

JavaScript am AEG

Lektion 5: Wünsch Dir was!

Paul Pasler | AEG Reutlingen, Stefan Gaum, Kursstufe

Lektion 5: Wünsch dir was

- Was wollt noch kennenlernen?
- Technik
  - JavaScript Funktionen (Methoden)
  - REST-Ful Webservice (AJAX, JSON)
- Bibilotheken
  - Bilderslider
  - Interaktiven Tabellen
  - Diagramme
  - Präsentationen
  - Animationen

Danke, noch Fragen?