

Web-Technologien

- Dipl.-Inf., Dipl.-Ing. (FH) Michael Wilhelm
- Hochschule Harz
- FB Automatisierung und Informatik
- mwilhelm@hs-harz.de
- <http://www.miwilhelm.de>
- Raum 2.202
- Tel. 03943 / 659 338

Inhalt

1. Einleitung
2. HTML und CSS
3. HTML-Formulare
4. **Javascript**
5. PHP-OOP-Datenbanken
6. Ajax / fetch
7. JSP

Java-Script

- Javascript ist eine leicht erlernbare und anwendbare Objekt-Sprache, die dazu dient, dynamische Elemente in HTML-Dateien einzuführen.
- Javascript ist kein direkter Bestandteil von HTML
- Es ist eine eigene Programmiersprache
- Es ist **ähnlich** wie Java
- Unterschiede in der Deklaration, Events und in OOP
- Vom **W3C** wurde das **Document Object Model (DOM)** verabschiedet (Version 1,0 bis 2,0)
- DOM ist ein Interface, mit dem man auf die Elemente eines HTML-Dokuments zugreifen kann (bel. Sprache)

Java-Script

- Wurde 1995 von Netscape für dynamische Webseiten entwickelt.
- Richtiger Name lautet "ECMAScript"
- Es ist eine interpretierte Sprache mit Syntax im C-Stil
- Durch die Berühmtheit von Java wurde ECMA wurde aus Marketing-Gründen in JavaScript umbenannt.
- Java und JavaScript sind völlig verschieden und haben nicht wirklich etwas gemeinsam:
 - Variablendeklaration, for-Schleifen, OOP unterschiedlich
- Ursprünglich wurde JavaScript für dynamisches HTML entwickelt (Grundlage von HTML5)
- Heute kann JavaScript für Webseiten, Anwendungen, Backend (Server-Side) Anwendungen und als anwendungsintegrierte Schnittstelle für Experten (In App Console) verwendet werden.

Java-Script

Eigenschaften von Javascript:

- Es ist eine Skriptsprache.
- Objektorientiert
- Integriert in HTML-Dateien
- Extern auch in Java-Dateien gespeichert
- Wird benötigt für HTML-Formulare
- Wird beim Client ausgeführt
- Keine Variablendeklaration, trotzdem sinnvoll (Strict-Modus)
- Kein Compiler
- Debugger verfügbar (Eclipse, Plug In)
 - <https://addons.mozilla.org/de/firefox/addon/firebug/>
- Strict-Modus
 - "use strict"
- Viele Sprach-Erweiterungen in HTML5

Java-Script Tag (Abschnitt)

- Das Tag `<script>` definiert alle Textzeilen als Script

Beispiel:

- `<script type="text/javascript">`
- `"use strict"`
 - `// Aktion`
- `</script>`

Variablen

- Definition:
 - . let name="My Name is Nobody";
 - . let datum;
 - . let alter = 123;
 - . let t = document.title;
 - . let abc; // noch undefiniert
 - . let myArray = new Array[10];
- Variablen können neu definiert werden
- Variablen können unterschiedliche Werte erhalten
- W1="123"*1 // Konvertierung String nach Zahl
- sStr=123+"" // Konvertierung Zahl nach String

- let d = **parseFloat**("123.456");
- d = parseFloat("s123.456"); // Error und Rückgabewert ist NaN
- let k = **parseInt**("123"); // liefert 123
- n = parseInt("100px"); // liefert 100
- doublevalue = parseFloat("123.456"); // liefert 123.456
- doublevalue = parseFloat("123,456"); // liefert 123
- let sStr = k.toString();

Variablen

- Definition:
 - . let sMnr="1234";
 - . let iMnr=sMnr+0;
 - . alert(sMnr); Ausgabe einer Meldung
 - . alert(iMnr);

 - . let i; Variable wird „erzeugt“
 - . let i=123; Variable erhält einen Wert (Typ integer)
 - . i="abcd"; Variable erhält einen Wert (Typ string)
 - . i=123.456; Variable erhält einen Wert (Typ double)

Java-Script-Sprachelemente

- if-else-Anweisung à Java

- for-Schleife

```
for (let i=start; i<end ; i++) {  
    let item = array[i];  
}  
for (let i in array) {  
    let item = array[i];  
}  
for (let item of array) {  
    // let item = array[i]; nicht notwendig  
}
```

Java-Script-Sprachelemente

- switch

```
switch (bedingung) {  
    case 0: break;  
}
```

- while-Schleife à Java

```
while (bedingung) {  
}
```

- Do-while-Schleife à Java

```
do {  
} while (bedingung)
```

Java-Script-Sprachelemente

- Arrays

```
let feld = [];  
feld.push(42);  
feld.push(41);  
feld.push("FB AI"); // keine festen Datentypen  
  
let size = feld.length;           // 3  
let first = feld[0];             // FB AI  
feld[0] = 43;                   // neues 1. Element  
feld [9] = "FB W";  
size = feld.length;             // 10!
```

Java-Script-Sprachelemente

- Objekte, Ein Objekt hat beliebig viele Werte, Variablen

```
let customer = {  
};  
customer.firstname = "Thomas";  
customer.lastname = "Miller";  
customer.no = 42;
```

Attribute werden, wie in PHP, Python,
automatisch **zur Laufzeit** eingefügt!
Dies gilt nicht für Methoden

Java-Script-Sprachelemente

■ Funktionen

```
function calc( a ) {  
    "use strict";  
    // Aktion  
    let b = a+1;  
    return b;  
}
```

Eine Rückgabe wird nicht explizit deklariert.

Struktur eines Java-Scriptes

- Innerhalb des Head-Blockes, nach dem <head>-Abschnitt
- Das Tag <src> definiert die externe Datei (optional)
 - <script src="http://www.myserver.de/Script.js">

■ JavaScript-Objektreferenz

- windows.location: Zugriff auf die URL
- history (Seiten des Users)
- document Alle Elemente der Seiten
 - URL, HTML-Elemente, forms, images
- forms Formularabschnitte etc.
- navigator Browser-Information
- screen: (availHeight, availWidth, height, width, colorDepth, pixelDepth)

Beispiele für Eigenschaften

- document.title Titel
- document.fgcolor Farbe des Seitentextes
- document.bgcolor Farbe des Seitenhintergrund
- document.lastModified Datum der letzten Änderung
- form.action URL für das CGI-Script
- location.hostname URL-Adresse

Beispiele für Methoden

- document.write Schreibt Text in eine Datei
- form.submit Übermittelt das Formular
- window.alert (alert) Meldungsfenster
- window.open(URL,target) neue Adresse

Beispiele für Ereignisse (1)

Allgemeines zu Event-Handlern

- onAbort bei Abbruch
- onBlur beim Verlassen eines Elements (!focus)
- onChange Aufruf beim Verlassen eines Elementes
= onChangeFocus
- onClick beim Anklicken
- onDbClick bei doppeltem Anklicken
- onError im Fehlerfall
- onFocus beim Aktivieren
- onKeyDown bei gedrückter Taste
- onKeyPress bei gedrückt gehaltener Taste
- onKeyUp bei losgelassener Taste
- onLoad beim Laden einer Datei

Beispiele für Ereignisse (2)

- `onMouseDown` bei gedrückter Maustaste
- `onMouseMove` bei weiterbewegter Maus
- `onMouseUp` bei losgelassener Maustaste
- `onMouseout` (beim Verlassen des Elements mit der Maus)
- `onMouseover` (beim Überfahren des Elements mit der Maus)
- `onReset` beim Zurücksetzen des Formulars
- `onSelect` beim Selektieren von Text
- `onSubmit` beim Absenden des Formulars
- `onUnload` beim Verlassen der Datei
- `javascript:` bei Verweisen

Editfeld ansprechen

Die Form aus einem Array holen:

```
sStr = document.forms[0].theName1.value
alert(sStr)
sStr = document.forms[1].theName2.value
alert(sStr)
```

Oder die Form als Parameter übergeben:

```
function calc( form ) {
    sStr = form.theName1.value
}
function calc( f ) { //oder
    sStr = f.theName1.value
}
```

Editfeld ansprechen

Die Form mit einem Namen versehen:

```
<form name="form1">
```

```
  Eingabe der Zahl: <input type="text"
    name="zahl" size="5" value="234.45">
</form>
```

Die Form mit einem Namen ansprechen:

```
function checkform() {
  sStr = document.form1.zahl.value;
  ...
}
```

CheckBox ansprechen

```
<form>
<input type="checkbox" name=CAD1 > AutoCAD<br />
<input type="checkbox" name=CAD2 checked="checked" > TurboCAD <br />
<input type="checkbox" name=CAD3 checked="checked" > Eagle<br />
<input type="button" value= "Ok" onclick="check(this.form)">
</form>
```

```
function check( form ) {
  if (form.CAD1.checked ) {
    alert ("CAD1 gesetzt");
  }
  else {
    alert ("CAD1 nicht gesetzt");
  }
}
```

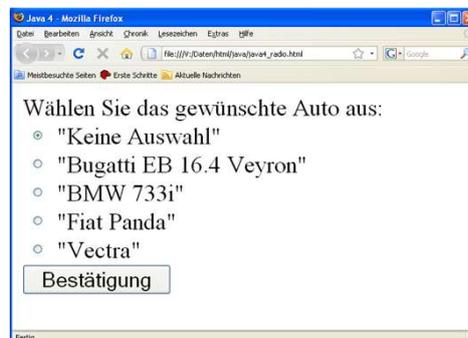
Radiobutton ansprechen

```
<form>
Wählen Sie die Option aus: <br />
  <input type="radio"      name="Radio1"
                        value="VW" > "Bugatti" <br />
  <input type="radio"      name="Radio1"
                        value="BMW" checked> "BMW 733i" <br />
  <input type="radio"      name="Radio1"
                        value="FIAT" > "Fiat Panda" <br />

  <input type="button" value="Ok" onclick="check(this.form)">
</form>
```

Radiobutton ansprechen

```
<script language="javascript1.1">
function check(f) {
  for (i=0; i<f.Radio1.length; i++) {
    if ( f.Radio1[i].checked) {
      alert("Gewählter Index: "+(i) );
      alert(f.Radio1[i].value);
    }
  } // transfer
}
</script>
```



ComboBox ansprechen

```
<form>
  Wählen Sie das gewünschte Auto aus: <br />
  <select name="liste">
    <option> "Bugatti EB 16.4 Veyron"</option>
    <option > "BMW 733i"</option>
    <option selected > "Fiat Panda"</option>
  </select>
<p>
  <input type="button" value=" Bestätigung " onclick="check(this.form)">
</p>

</form>
```

Java:

```
index = document.forms[0].liste.selectedIndex; // numerisch
sStr = index+""
alert(sStr)
```

ComboBox ansprechen

```
<form>
  <select name="wochentag" size="4" multiple="multiple" >
    <option value="Montag" > Montag </option>
    <option value="Dienstag" selected > Dienstag </option>
    <option value="Mittwoch" > Mittwoch </option>
    <option value="Donnerstag" selected> Donnerstag </option>
    <option value="Freitag"> Freitag </option>
  </select>

  <input type="button" value="senden" onclick="check(this.form);" >
  <input type="reset" value="Löschen"></p>

</form>
```

ComboBox ansprechen

```
<script language="javascript">
<!-- start script here -->
function check( form ) {
  alert(form.wochentag.length); // bestimmt die Länge
  for (i=0; i<form.wochentag.length; i++) {
    if ( form.wochentag.options[i].selected) {
      alert("Gewählter Index: "+(i)+"\n" + form.wochentag.options[i].value );
    }
  }
  return;
}
// End of script -->
</script>
```

Editorzeile ansprechen

```
<form>
  <p>
    Eingabe der Zahl:
    <input type="text" name="zahl" size="5" value="234.45">
  <p>
    <input type="button" value="OK" onclick="check(this.form)">
  </p>
</form>
```

Java:

```
sStr= document.forms[0].zahl.value;
betrag1= sStr*1;
alert(betrag1)
```

Mehrzeiliger Editor ansprechen

```
<form>
  <p>
    Eingabe des Texte <br />
    <textarea name="theTextArea" rows="10" cols="50"> Input </textarea>
  <p>
    <input type="button" value="OK"          onclick="check(this.form)">
  </p>
</form>
```

Java:

```
sStr= document.forms[0].theTextArea.value;
sStr= f.theMemo.value;
alert(sStr)
```

Java-Script-Sprachelemente

- If-else-Anweisung
- Switch-Anweisung
- for-Schleife
- while-Schleife
- Do-while-Schleife
- Prozeduren / Funktionen

Adressen

- <http://de.selfhtml.org/javascript/sprache>
- <http://de.selfhtml.org/javascript/sprache/eventhandler.htm#onchange>

Java-Script-Vergleich

Korrekt	Falsch
<code>var i;</code>	<code>int i;</code>
<code>var i;</code>	<code>var int i;</code>
<code>for (i=0; i<10; i++)</code>	<code>for (int i=0; i<10; i++)</code>
<code>if (sStr == "") summe+=i;</code>	<code>if (sStr = "") summe+=i;</code>
<code>switch (k) { case 0: s=0; break; case 1: s=0; break; }</code>	<code>switch (sK) { case "ABC": s=0; break; case "DEF": s=0; break; }</code>

jQuery

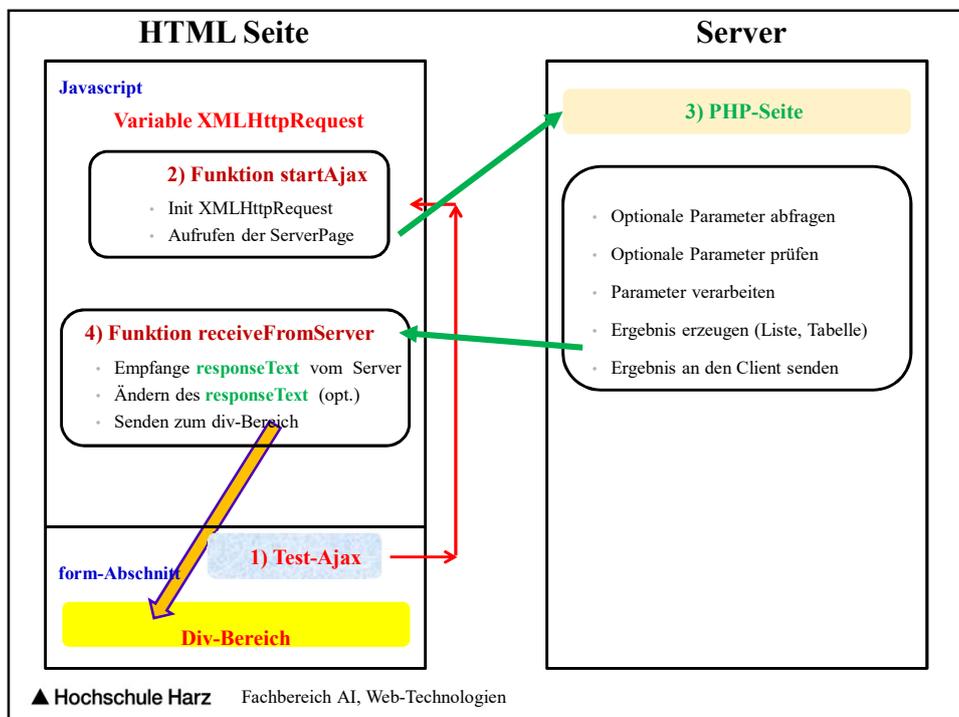
- jQuery ist ein Framework, welches die GUI-Programmierung stark vereinfacht:
 - TabRegister
 - ProgressBar
 - Accordion (aufklappbare Schalter)

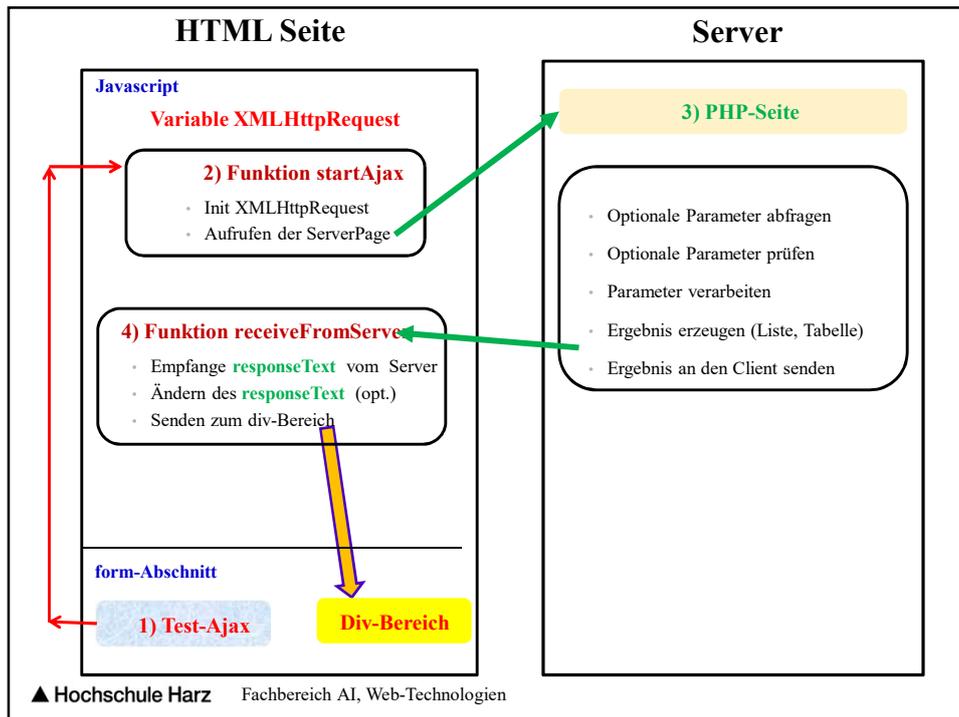
- [Jakob Westhoff](#)
- Plug-in-Entwicklung mit [jQuery](#)
- ISBN: 978-3-86802-052-6

- <http://jqueryui.com/>

Ajax

- **Asynchronous JavaScript and XML:**
 - Eine Seite wird nicht mehr vollständig neu geladen
 - Es werden nur Teile nachgeladen
- **Christian Wenz**
- Ajax
- ISBN: 978-3-86802-045-8
- **Steven Douglas Olson**
- Ajax on Java
- ISBN 978-0-596-10187-9
- EUR29.00





jQuery

- jQuery ist ein Framework, welches die GUI-Programmierung stark vereinfacht:
 - TabRegister
 - ProgressBar
 - Accordion (aufklappbare Schalter)
- Jakob Westhoff
- Plug-in-Entwicklung mit jQuery
- ISBN: 978-3-86802-052-6
- <http://jqueryui.com/>

Ajax

- **Asynchronous JavaScript and XML:**
 - Eine Seite wird nicht mehr vollständig neu geladen
 - Es werden nur Teile nachgeladen

- [Christian Wenz](#)
- Ajax
- ISBN: 978-3-86802-045-8

- [Steven Douglas Olson](#)
- Ajax on Java
- ISBN 978-0-596-10187-9
- EUR29.00