**HTML – CSS – Kurs**

Jochen Stärk

Anfang 2016

Inhalt

[Stunde 1 3](#_Toc442365103)

[HTML Grundlagen 3](#_Toc442365104)

[Übung: Bauen einer Hallo Welt Seite 3](#_Toc442365105)

[CSS = Cascade Style Sheet 4](#_Toc442365106)

[Stunde 2 6](#_Toc442365107)

[Exkurs - Pfadangabe 6](#_Toc442365108)

[Exkurs – Domain 6](#_Toc442365109)

[Exkurs – Hintergrund-Grafik: 6](#_Toc442365110)

[Übung: Video einbinden 6](#_Toc442365111)

[Exkurs: Video-Formate: 7](#_Toc442365112)

[Exkurs: Grafikformate und Programme: 7](#_Toc442365113)

[Übung: Drucken aus html heraus 7](#_Toc442365114)

[Exkurs: HTML-Farbencodes 8](#_Toc442365115)

[Übung: Mail schreiben 8](#_Toc442365116)

[Übung Links 8](#_Toc442365117)

[Exkurs: Impressum 8](#_Toc442365118)

[Exkurs: Php 9](#_Toc442365119)

[Stunde3 9](#_Toc442365120)

[Mime-types= multipurpose Internet mail extensions 10](#_Toc442365121)

[Formulare 10](#_Toc442365122)

[Input-Objekte 10](#_Toc442365123)

[Function in Javascript: 10](#_Toc442365124)

[Aktionen 11](#_Toc442365125)

[Validierung- Werte mit Javascript abfragen und auswerten 11](#_Toc442365126)

[Exkurs – Das „=“ Zeichen, Zuweisung oder Vergleich 11](#_Toc442365127)

[Werte/Style mit Javascript verändern 11](#_Toc442365128)

[Animationen 12](#_Toc442365129)

[Methodik – Homepage-Projekt 12](#_Toc442365130)

[Stunde4 14](#_Toc442365131)

[Spezialelemente: 14](#_Toc442365132)

[jQuery 14](#_Toc442365133)

[Accordion 14](#_Toc442365134)

[Document ready 15](#_Toc442365135)

[Datepicker 15](#_Toc442365136)

[Eventhandler 15](#_Toc442365137)

[Widget: Tab 15](#_Toc442365138)

[SEM 15](#_Toc442365139)

[Methodik- Auslieferung über Webserver 16](#_Toc442365140)

# Stunde 1

Quellen:

* Empfehlung: wiki.selfhtml.org (HTML/CSS/Javascript-Referenz) und
* "HTML &CSS: Erfolgreich Websites gestalten und programmieren" von Jon Duckett.
* Codecademy

Tools:

* Kdiff3 (Differenzenprogramm, einerseits für Dateien andererseits direkt auf Ordnerebene
* Texteditoren: z.B. Blue Griffin, Atom

## HTML Grundlagen

HTML vs. http

* HTML ist ein Format zur Darstellung von Daten.
* Protokoll geht es um Reihenfolge der Daten. Protokoll ist http.

**Tipp:**

www.Esatus.com => Seite für Chromium (Open Source Basis für Chrome)

Mit “View source” kann man Source-Code anschauen

Texteditor (Atom)

### Übung: Bauen einer Hallo Welt Seite

<!DOCTYPE html>

**<html>**

**<!--Kommentar> </-->**

**<body>**

<p style=”font-size:15em”> hallo welt <br>

</p>

**</body>**

**</html>**

|  |  |  |
| --- | --- | --- |
| **HTML-Tags** | **Was** | **Beispiel-Attribute** |
| h1 | Oberste Überschrift |  |
| p | Paragraph, Absatz | style=““ |
| br | Zeilenumbruch (braucht kein schließen, weil nicht streng xml basiert) |  |
| div | Blockelement, Formatdefinition für gesamten Bildschirmbreite |  |
| <span> | Fließtext, Formatdefintion nur soweit Text reicht |  |
| <a href=““> | Anker, für Link | href (Dateilink)  target: wo geöffnet werden soll (\_blanc) |
| <img src=““> | Bild einbinden | src=““,  alt=““ (alternate text, falls Bild nicht angezeigt werden kann)  style=““ |
| table | Tabelle |  |
| tr | table row |  |
| td | table data |  |
| ul | unordered list |  |
| ol | ordered list |  |
| li | list Item |  |
| em | Textauszeichnung: kursive Schrift |  |
| strong | Textauszeichnung: fette Schrift |  |

Sonstiges:

* "Entities": Sonderzeichen wie &ndash; und &uuml;
* Exkurs über Umlaute: mit UTF8 im Header über Charset, statt mit &Uuml
* Zeichenkodierung mit UTF-8 vs "ASCII"
* Font-size Em: standard „m“-größe. Passt Größe an Browser an. Besser als px für Pixel.

**Tipp:** Verwenden von Chromium zum "Prüfen" (Rechtsklick auf Objekt, z.B. Text, öffnet die  
Entwicklertools, darin kann - im Arbeitsspeicher - editiert werden)

<html>

**<head>**

<meta charset="utf-8">

**</head>**

<body>

<p style=”font-size:15em”> hallo welt <br>

</p>

</body>

</html>

## CSS = Cascade Style Sheet

Wie wird etwas dargestellt?

Varianten für Style-Definitionen:

* **Inline-CSS in HTML**
* Syntax: <Tag style = “ccs-Attribut1:Wert1 ccs-Attribut2:Wert1 “
* Bespiel:
* <img style="float:right" src="ProgrammersScared.jpg" alt="Programmerscared">
* <p style="font-size:15em">hallo welt<br> </p>
* **CSS-Anwendung im Header**

<html>

<head>

<meta charset="utf-8">

**<style>**

**/\*CSS kommentar\*/**

**</style>**

</head>

<body>

</body>

</html>

* + über Style-Namen (quasi wie Inline-CCS) ???
    - Syntax:
      * Style: *Style-Name*=“ccs-attribut1:Wert1 ccs-attribut2:Wert1…”
      * Body: <tag style=”*Style-Name”*>
    - Beispiel
      * myFont =”font-size:15em”
      * <p style="myFont”>hallo welt<br> </p>
  + direkt am Tag
    - Syntax:
      * Style: Tag **{**ccs-attribut1:Wert1; ccs-Attribut2:Wert1…**}**
      * Body: nur Tag-Aufruf
    - Beispiel:
      * table {font-size:5em; border-collapse: collapse }
* über id-Attribut
  + - Syntax:
      * Style: #*ID-Name* **{**ccs-attribut1:Wert1**;** ccs-Attribut2:Wert1…}
      * Body: <tag id=”*Klassen-Name*”>
  + über class-attribut
    - Syntax:
      * Style: **.***Klassen-Name* {ccs-attribut1:Wert1 Wert2 Wert3; ccs-Attribut2:Wert1…}
      * Body: <tag class=”*Klassen-Name*”>
    - Beispiel:

.kleinesDiv {background-color:tomato;width:130px;float:left}

* **Über Dateien**
  + Stylesheet: alles was zuerst unter dem Tag <style> war.
  + Syntax: Style: <link href="mystyle.css" rel="stylesheet">

|  |  |  |  |
| --- | --- | --- | --- |
| **CCS-Attribute** | **Wo** | **Was** | **Beispiel-Wert** |
| float: | img | Wie fließt Text um Bild | right, left |
| text-align |  | Bündigkeit beim Absatz |  |
| font-size: |  | Schriftgröße | 15em, 150 px |
| color |  | Achtung: nicht text-color bei CSS |  |
| font-color | p, … | Schriftfarbe | tomato, green, …. |
| background-color | div (nix p) | Hintergrundfarbe |  |
| text-decoration | h1 |  | underline, none |
| border-collapse | table | Rahmen | collapse |
| padding | td |  |  |
| border | td | Rahmen | 5px dashed red |
| width | span | Breite | 33% |

- Erste Erwähnung des **Box-Modells**

(in den Entwicklertools: jedes Box-Element ist umgeben von Padding/Border/Margin)

**Tipp**:- Jochen hatte die Browser-Erweiterung bei Chrome "Web Developer" kurz aufgerufen um  
HTML über einen externen Dienst (validator.w3c.org) prüfen zu können

# Stunde 2

Wiederholung:

**Tipp**:-die "Web developer" Firefox und Chromium extension von Chris Pederick

Tags:

Ul

Ol

<Img src=“img/karte.jpg“ style=‘max-width:75%‘>

### Exkurs - Pfadangabe

* Absolute Pfade: z.B. /a… beginnen mit /
* Relativer Pfad ohne /
* Pfad auf Server: Aufgaben eines Webservers (Verzeichnisse mit HTML-Dateien und Grafiken per HTTP weiterzugeben), Browser holt sich Dateien über http(s), Bildadresse kopieren
  + <a hfref=[<http://www.google.de/a/b/bild.jpg>.](http://.......)> hier </a> …… anker

### Exkurs – Domain

[www.example.de](http://www.example.de)

www = third level domain, server (www für Webserver, ftp für FTP-Server)

example= second level domain

de = top level domain

### Exkurs – Hintergrund-Grafik:

* Vordergrundgrafik: mit Maus verschiebbar
* Hintergrundgrafik: in CSS: Style=‘Background-image:url(img/Karte.jpg)‘

**Tipp**: CSSZengarden.com als anschaulichem Beispiel wie verschieden man dasselbe HTML mit CSS stylen kann

Style-Definition über:

* Id mit Definition mit hash #

<div id = „welt“> hallo </div>

Im Header: #welt{font-weight:bold, font-size: 15em}

* Class

Im Header .welt{font-weight: bold, font-size: 15e}

.fett{font-weight:bold}

.gross{fong-size:15em}

Aufruf: mit Leerzeichen hintereinander gehängt

<div class: ‚fett gross‘>

### Übung: Video einbinden

<video controls="controls" style='width:600px'>

<source src="videos/video.avi" type="video/mp4">

<source src="videos/video.ogg" type="video/ogg">

</video>

Der Beispielfilm war übrigens der Anfang des mit der Open-Source-Software Blender erzeugte  
Film "Big Buck Bunny" (s. bspw. [https://www.youtube.com/watch?v=YE7VzlLtp-4](https://3c.gmx.net/mail/client/dereferrer?redirectUrl=https%3A%2F%2Fwww.youtube.com%2Fwatch%3Fv%3DYE7VzlLtp-4))

### Exkurs: Video-Formate:

* Dateiformate (Container) wie AVI (und OGG), Ogg Dateiformat im Gegensatz zu AVI viel kleiner.

Eine **Containerdatei** oder kurz ein *Container* (von [englisch](https://de.wikipedia.org/wiki/Englische_Sprache) *container* ‚[Behälter](https://de.wikipedia.org/wiki/Container)‘) bezeichnet in der [Informationstechnik](https://de.wikipedia.org/wiki/Informationstechnik) eine [Datei](https://de.wikipedia.org/wiki/Datei), die ihrerseits wiederum unterschiedliche Dateien und [Dateitypen](https://de.wikipedia.org/wiki/Dateityp) enthalten kann. Als **Containerformat** hingegen wird die Art und Weise der Anordnung bezeichnet, wie verschiedenartige [Datenformate](https://de.wikipedia.org/wiki/Datenformat) innerhalb eines Containers gespeichert werden – also die innere [Datenstruktur](https://de.wikipedia.org/wiki/Datenstruktur) einer Containerdatei (Wikipedia: Containerdatei)

* Videocodecs wie mp4 (=H264) und "Ogg Theora".
* Es ist wichtig, zwischen dem kodierten Datenformat, also dem [Audioformat](https://de.wikipedia.org/wiki/Audioformat) und der verwendeten [Videokompression](https://de.wikipedia.org/wiki/Videokompression), und dem [Containerformat](https://de.wikipedia.org/wiki/Containerformat) zu unterscheiden. Beispielsweise erstellt der bekannte XviD-Codec [MPEG-4](https://de.wikipedia.org/wiki/MPEG-4)-Videospuren oder der MP3-Audiocodec [MP3](https://de.wikipedia.org/wiki/MP3)-Audiospuren. Diese beiden Klassen können in einem Dateicontainer (z. B. [AVI](https://de.wikipedia.org/wiki/Audio_Video_Interleave)) kombiniert und dann gespeichert werden (Wikipedia: Codec)
* Seite sollte mindestens unterstützen: ogg und mp4

### Exkurs: Grafikformate und Programme:

* Unterschied Vektor/Rastergrafiken (Pixelgrafik): Vektorgrafik speichert Koordinaten, Pixelgrafik speichert die einzelnen Pixel ab.
* Tool: yed [https://www.yworks.com/products/yed](https://3c.gmx.net/mail/client/dereferrer?redirectUrl=https%3A%2F%2Fwww.yworks.com%2Fproducts%2Fyed): so wie Viseo (leider “nur” Freeware)
* Incscape: Vektorgrafikprogramm; [http://www.inkscape.org](https://3c.gmx.net/mail/client/dereferrer?redirectUrl=http%3A%2F%2Fwww.inkscape.org))
* Gimp: Pixel Graphik
* Pdf macht vektor und pixel (container, kein Format)
* Postscript auch Vector .eps
* .svg scalable vector graphik (auch ein xml format: kann auch schreiben)
* Pixelgraphik: JPG (verlustbehaftet) und PNG (groß)

**Tipp:**

* Openclipart public domain, frei nutzbar, veränderbar; Openclipart.net als Quelle für praktisch urheberrechtsfreie SVGs
* Caniuse.com: zeigt, welche Dateiformate, ab welcher Browserversion
* Sodipodi
* Creativ-Cloud: von da progrogramme
* .gif nicht mehr empfehlenswert, weil nur 256 Farben. Man nutzt .png
* Fotos: jpg, sonst alles png für selbsterstelltes oder svg für Vektorgraphiken
* Über css kann man auch animieren
* Lorum ipsum text generator
* Computer science paper generator: html generierte Seite: [https://pdos.csail.mit.edu/archive/scigen/](https://3c.gmx.net/mail/client/dereferrer?redirectUrl=https%3A%2F%2Fpdos.csail.mit.edu%2Farchive%2Fscigen%2F)

### Übung: Drucken aus html heraus

<p>

<h1> Druckansicht </h1>

Klicken Sie <a href =“javascript:window.print()“ > hier </a> hier um zu drucken

</p>

Aufruf PseudoProtokoll: Javascript:

* Syntex: <a href =“**javascript**:window.print()“ >

Soll selbst nicht gedruckt werden: CSS-Media Query

* Syntax:
* style: **@media print**{.screenonly{display: none;background-color:#ff0000}}
* body: <div class = “screenonly”>, <p..>

### Exkurs: HTML-Farbencodes

* (RGB mit je zwei Hexadezimalziffern) wie #000000  
  (schwarz), #ffffff (weiß), #808080 (grau), #00FF00 (grün), #008000  
  (dunkelgrün)

### Übung: Mail schreiben

Schreib mir <a href = “**mailto**:jstaerk@usegroup.de> eine mail </a>

Achtung: spam!!!!!!

Aufruf Pseudoprotokoll: Mailto

* Syntax: <a href = “mailto:jstaerk@usegroup.de>

**Tipp**: Capture hilft: recapture verwendet javascript

Einbindung externer Javascript-Dateien mit <script src

* head: <script src='https://www.google.com/recaptcha/api.js'></script>
* body: <div class="g-recaptcha" data-sitekey="6LeA4ckSAAAAAOC-2oSAuPOWlMqvnrVfPrZYT-qY"></div>

### Übung Links

* Man kann Links auch um Bilder legen
  + <a href="http://vajkonny.de"><img style='max-width:75%' src="img/karte.jpg"></a
* a {color:lime} => Färbt alle links grün
* -**CSS-"pseudo"-Klassen** wie a:hover
  + a:hover{color:blue} => wenn man drüber fährt wird Link blau

**Tipp**: -Emulieren von Mobilgeräten und Tablets mit dem "Device-Mode" des  
ChromXXX-Inspektors, weil es nicht automatisch auch am iphone läuft

*Hü: welche Seiten möchte ich haben, Mindmap mit Freemind*

### Exkurs: Impressum

* Denic: alle Informationen über Webseiten
* Whois abfrage
* nachsehen ob das nur für kommerzielle Seiten gilt. § 5 TMG spricht von  
  "geschäftsmäßige[n], in der Regel gegen Entgelt angebotene[n Seiten]" .  
  Also streng formaljuristisch gesehen wahrscheinlich sogar ja,  
  geschäftsmäßig bedeutet in dem Zusammenhang aber wahrscheinlich eher  
  dauerhaft als kommerziell und wenn ich einen privaten Reiseblog mit  
  Werbung garniere gilt das wahrscheinlich auch als geschäftsmäßig.

### Exkurs: Php

* ab 2 html-dateien, php generiert html, läuft am Webserver
* Cms (Content Management System): typo 3, drupa, wordpress, jumla alle über php und opensource
* 25% laufen unter Wordpress

Zum Thema PHP-Programmieren: ich hatte wie gesagt letztes Jahr drei  
Webinare gehalten, Materialien anbei, hier die Links zu den Videos (kann  
mir mal jemand sagen ob die noch funktionieren?).  
  
Stunde 1: Testsystem aufsetzen (Windows), 52 minuten.  
Im Browser abspielen:  
[https://usegroup.webex.com/usegroup-en/ldr.php?RCID=928f6a182153c5e2f96a13b96fcd0178](https://3c.gmx.net/mail/client/dereferrer?redirectUrl=https%3A%2F%2Fusegroup.webex.com%2Fusegroup-en%2Fldr.php%3FRCID%3D928f6a182153c5e2f96a13b96fcd0178)  
  
Stunde 2: Datenbanken, 1 hour 2 minutes,  
[https://usegroup.webex.com/usegroup-en/ldr.php?RCID=0bd58504bb8140caac306f6f9467d8c4](https://3c.gmx.net/mail/client/dereferrer?redirectUrl=https%3A%2F%2Fusegroup.webex.com%2Fusegroup-en%2Fldr.php%3FRCID%3D0bd58504bb8140caac306f6f9467d8c4)  
  
Stunde 3: Debugging, 59 minutes,  
[https://usegroup.webex.com/usegroup-en/ldr.php?RCID=1eb0c7ceeca7bd1f06fe146ca4be7140](https://3c.gmx.net/mail/client/dereferrer?redirectUrl=https%3A%2F%2Fusegroup.webex.com%2Fusegroup-en%2Fldr.php%3FRCID%3D1eb0c7ceeca7bd1f06fe146ca4be7140)

|  |  |  |
| --- | --- | --- |
| **HTML-Tags** | **Was** | **Beispiel-Attribute** |
| video | bindet Video ein | controls, style |
| source | innerhalb von Video, welcher Dateityp | src, type |
| script | ruft unter <head> ein script auf |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **CCS-Attribute** | **Wo** | **Was** | **Beispiel-Wert** |
| max-width | Img | Maximal Breite | 75% |
| height | Img | Höhe | 200% |
| background-image: | Img | Hintergrundgraphik | url(img/Karte.jpg |
| background-repeat | Img | Hintergrundgraphik wiederholen | no-repeat |
| a:hover | A | Pseudoklasse zum umfärben, wenn Mauszeiger über Objekt | {color:blue} |

|  |  |  |
| --- | --- | --- |
| **Javascript-Funktion** | **Was** |  |
| window.print() | Druckt Fenster |  |

# Stunde3

Wiederholung:

Codec

Pixel vs.vectoren

**Tipp**: Zen-coding erspart alles selbst zu schreiben. Zen Coding heißt jetzt Emmet,  
[https://de.wikipedia.org/wiki/Emmet\_(Software](https://3c.gmx.net/mail/client/dereferrer?redirectUrl=https%3A%2F%2Fde.wikipedia.org%2Fwiki%2FEmmet_%28Software)) und ist toll um sich  
bspw. ein HTML-Grundgerüst hinzulegen (Emmet html>head+body)

## Mime-types= multipurpose Internet mail extensions

<script type=“text/javascript“ > für .js

Stellt Dateiformat klar. Hier: Javascript kommt als Text.

Beispielliste folgt

## Formulare

<body>

**<form>**

<input type=“submit“ > </input>

**</form>**

</body>

Kommunikation nicht mit dem Browser sondern mit dem Webserver.

### Input-Objekte

* Input-Text-feld: Dein Name: <input name=”name” value=”vorgabewert” id=”namensfeld” type= „text“>
* Input-Bestätigungsfeld <input type=“submit“>
* Onsubmit = „alert(‚Hello Welt‘)“, onsubmit = Attribut unter form

### Function in Javascript:

Syntax:

* function myName() {}

Beispiel:

* function sayHello{alert ‚hallo‘ + document.getElmentByID(‚namensfeld‘).value}
* back to Input Objekte

**Auswahlknopf**

Du findest diesen Kurs:

* sehr gut <Input name=”Bewertung” type =”radio” value=“1”> <br>
* Gut <Input name=”Bewertung” type =”radio” value=“2”>

Browser weiß durch den gleichen Namen, name = „Bewertung“, dass es eine Auswahl ist. Die beiden Buttons können aber unterschiedliche ID haben.

**Checkbox**

<Input name=”Bewertung” type =”Checkbox” >

Tag: **Textarea** Eingabe mit Zeilen und Spalten

**Drop Down**

Tag: select

Tag: Option

**Mehrfachauswahl**

<select multible =“multible“>

### Aktionen

* onchange=“sayHello()“
* onkeyup=“sayHello()“
* Alert(“Hallo”)

### Validierung- Werte mit Javascript abfragen und auswerten

alert (‚hallo‘ + document.getElmentByID(‚namensfeld‘).value}:

(Man erkennt, ob Browser abgeschickt hat in der Adresszeile)

Ü: Formular wird nur geschickt, wenn Wert unter Name vernünftig

unter Form: Onsubmit “return sayHello()“

Und unter function sayHello

Function sayhello () {

If (document.getElementByID(‘namensfeld’).value.length == 0){

alert(document.getElementByID(„msg“).innerHTML);

Return false;}

Return true;

}

### Exkurs – Das „=“ Zeichen, Zuweisung oder Vergleich

= Zuweisung

== Vergleich, <,>,>=,<=, != Ergebnis: true (1), false (0)

### Werte/Style mit Javascript verändern

Document.getElementbyID.innerHTML: er schreib es in HTML raus.

Blockelment in HTML: <div id=”msg” style=“background-color :red”>text</div>

Ü: Formular: wenn jemand den eigenen Namen reinschreibt, kommt alert

**Tipp**: Javascript auch über „prüfen“-Entwicklertools und Console aufrufbar

**In Javascript noch mehr HTML machen (Beispiel 4)**

* document.getElemntbyID(‘msg’).innerHTML =“<Strong>HEllo </strong>“
* mit <table>
* document.getElemntbyID(‘msg’).style.display=”none” //entspricht style =”display:none”
* document.getElemntbyID(‘msg’).style.display=”#ff00ff”
* Aktion im normalen HTML <div onclick>
* Variablendefinition im Script Var x=10; x=x+10

**Kurios**: Web archive: WayBackMaschine, da hat einer das Interent runtergeladen und man kann sich viele Webseiten anschauen in alten Versionen

## Animationen

Css-Attribut: Position

Css angewendet auf msg: transition: all 1s ease-in-out;

Er nimmt sich 1 sec um das zu verschieben

Unter welchen Umständen machen Animationen Sinn => es folgt Webseite

Wann und wie motion design sinn macht:  
[http://motionographer.com/2015/07/06/user-interface-design-why-motion-matters/](https://3c.gmx.net/mail/client/dereferrer?redirectUrl=http%3A%2F%2Fmotionographer.com%2F2015%2F07%2F06%2Fuser-interface-design-why-motion-matters%2F)

**Tipp**: man kann auch direkt in Konsole ausprobieren, zB. Alert, .style.left=““

## Methodik – Homepage-Projekt

1. Information archtiecture ia
2. Wireframes/Mockups – Paper Prototypes
3. Und dann gleichzeitig:

* TexterIn
* GrafikerIn
* ProgrammiererIn‘
* DBA

Mockup, Wire Frame

Tipp: Balsamic

Keine echten Bilder, keine echten Texte, sonst gleitet die Diskussion in Details ab

**Tipp**: Pencil Open source für Mockup

Wireframes testen mit paper prototypes:  
[http://www.wmfra.de/vortraege/fail-fast-fail-often-mit-papierprototypen-tobias-gunther.html](https://3c.gmx.net/mail/client/dereferrer?redirectUrl=http%3A%2F%2Fwww.wmfra.de%2Fvortraege%2Ffail-fast-fail-often-mit-papierprototypen-tobias-gunther.html)

Zusammenfassung Tags:

|  |  |  |
| --- | --- | --- |
| **HTML-Tags** | **Was** | **Beispiel-Attribute** |
| form | Formular | onsubmit = „alert(‚Hello Welt‘)“  onsubmit = „return sayHello()“ |
| input | Input-Formular | type =  “text“,  “submit“,  “radio”- Auswahlbutton.(“name” gleich)  “checkbox”, value nutzen für Verarbeitung  name=”Vorname”  value=”Vorgabewert”  id=”EindeutigerSchlüssel”  onchange =”myFunction()”  onkeyup=”myFunction()” |
| textarea | Input-Textfeld mit mehreren Zeilen |  |
| select | Dropdownliste | onchange =“sayHello()“  multible = „multible“ für Mehrfachauswahl |
| option | Unter select, für die einzelnen Einträge der Dropdownliste |  |
| div | Blockelement | onclick=“sayHello()“ |

|  |  |  |  |
| --- | --- | --- | --- |
| **CCS-Attribute** | **Wo** | **Was** | **Beispiel-Wert** |
| display |  | Darstellung | none: Inhaltsbereich wird nicht angezeigt  hidden: Inhaltsbereicht bleibt da, aber leer |
| position | z.B. bei <div> | Position des Objektes | absolut, vom Fenster aus links oben, reagiert nicht auf Fenstergröße  relative  inline (keine weiteren Argument) |
| left |  | Ortsangabe von Links |  |
| width |  | Weite |  |
| top |  | Ortsangabe von oben |  |
| transition |  | Übergang | all 1s ease-in-out: Browser nimmt sich 1 sec um das zu verschieben |
| opacity |  | Durchsichtigkeit |  |

|  |  |  |
| --- | --- | --- |
| **Javascript** | **Was** | **Beispiel** |
| function myName(){} | Funktionsdefinition | Code-Zeilen mit Strichpunkt beenden |
| if(Entscheidung){} |  |  |
| else{} |  |  |
| return | Rückgabewert | true, false |
| Var x=10; | Variablen-Definition |  |
| // | einzeiliger Kommentar |  |
| /\* \*/ | mehrzeiliger Kommentar |  |
| alert() | Schreibt einen Text in ein Fenster | alert =“Hallo“ |
| console.log() | Schreibt was in die Konsole raus | console.log = „Hallo“  console.log(x) |
| + | Textverknüpfung |  |
| document.getElmentByID |  | .value  .innerHTML =“<Strong>HEllo </strong>“L (schreibt in HTML raus)  .style.display=”none” //entspricht style =”display:none”  .style.backgroundColor  .style.left  .style.top  .style.opacity |

# Stunde4

## Spezialelemente:

Warte-Animation heißt Throbber

Menüs heißen nach Essen

Brotkrumennavigation (Bread-crump-navigation) zeigt an, auf welchem Pfad ich auch die Seite gekommen bin.

* Hamburger Menu
* Kebab Menu
* Döner Menu
* Bento Menu
* Meatball Menu
* Double Hamburger Menu

## jQuery

Javaskript-bibliothek, Klicki-Bunti Front End

Interessant: Widgets

### Accordion

z.B. Aufklappen über **accordion**

html Grundgerüst auch ohne Java-Script

jQuery macht dann danach Speziell

<div id=“myaccordion“>

</div>

<script>

$(„#myaccordion“).accordion():

</script>

Wie bei ccs, würde auch über Klasse statt id funktionieren.

Achtung beim Drucken: sollte ausschauen, wie ohne jQuery. Tuts aber nicht.

Script kann innerhalb von Body stehen

#### Exkurs - Parameterübergabe in Funktionen über Arrays

* Arrays

Var hallo=[„hallo“, „welt“];  
Console.log(hallo[1]);

* Assoziative Arrays

Var assoziativ={name:”Karin”, alter:29};

Console.log(assoziativ[‘name’]);

* Verschacheltungen

Var assoziativ2={name:”Karin”, alter:[5,6,7]};

Console.log(assoziativ2[‘alter’][2]);

Diese Notation heißt Json - Javascript object noation

Xml validierbar, json nicht mehr validierbar

### Document ready

Javascript wird schon ausgeführt, bevor alles von der Seite geladen ist

Mit Documentready wird function innerhalb erst ausgeführt, wenn Seite geladen.

$( document ).ready(function() {}

### Datepicker

zeigt Kalender zur Auswahl an

Konfigurieren (yearRange, weekheader, firstday (Welchen Tag er zuerst auswählt), parsedate……)

**Tipp**: Moment.js, Date-Range-picker für 2 Tage angeben und dazwischen auswerten.

### Eventhandler

onSelect – Javascript Befehl

function mySelect (string, object){

alert(string)

}

Aufruf innerhalb datepicker

onSelect:mySelect (dem Attribut onSelect wird die Funktion mySelect übergeben (ohne Klammern und Parameter)

Syntax: key:value

Inlinedefintion auch möglich.

onSelect: function(dateText, inst){}

Man kann auch die jQuery-Datei direkt einbinden, nicht über http://

Vorteil: Server bekommt es nicht mit, läuft ohne Internet

Nachteil: schneller, wenn mehrere Seiten, weil dann schon bei einer anderen Seite geladen

### Widget: Tab

Im zugeklappten wird nicht gesucht.

**Tipp**: Google custom search: google Suche mit Google Datenbestand durchsucht nur diese Seite aber alle Tabs, laschen etc. Wird im Quellcode festgelegt, z.B. ecb Seite

mnoGoSearch

## SEM

* SEM Search engine marketing
  + Seo optimization: seite gut gestalten, wie viele links zeigen auf deine Webseite. Es gab algorithmus Page-Rancing (Herr Page hats erfunden), Linkjuice - Links auf gute Webseite. Linkfarmen, mitlerweile gibt es für Seiten keinen Pagerank mehr
    - Qualitätsindex: Google crawled alle Wochen und aktualisiert Qualitätindex, Schnelligkeit zählt auch dazu.
    - **Tipp**: Es gibt Google-Ranking Seite: seitwert, googlewebmaster-Tools
  + Sea advertisement: man zahlt Google

#### Tipp: # als Anker

in langem Text automatisch an eine Stelle Springen:

<h1 id=#“eins“> Hier kommt etwas wichtiges

</h1>

In Adresszeile #eins eingeben

Oder über <a href = „#eins“> springt bei draufklicken dorthin

Liste an den Anfang stellen

<ul>

<li><a href="#eins">Erste Überschrift</a>

</li>

</ul>

Ü Webseiten-Hülle aufbauen

Title

4 Divs:

1 header

1 navis link width 20%, min-height

1 Inhalt rechts width 80%, Attribut Float

1 footer

## Methodik- Auslieferung über Webserver

ssh (secure shell) => verschlüsselte Datenverbindung mit einem Server, verbindung zur Konsole, man kann auch Dateien übertragen, scp

Mit Programm filezilla einfach rüberkopieren auf Linux (Sind meistens Webserver) Webserver meist Unix.

**Rechtevergabe:**

Ugo (hugo) für User group others

chmod

Rwx read write execute

Bsp: -rw-r—r—

(r=4, w=2, x=1)

Datei mit Namen **Index.html** wird automatisch zuerst gesucht, standardmäßig.

Domäne und Privider aussuchen. Vom Provider Zugriffsagen