

Frontend Basics

CSS3 und Bootstrap 4

Agenda

- Wiederholung und Besprechung der Hausübung
- Warum CSS?
- Aufbau von CSS, Einbinden von CSS Dateien in HTML
- Was kann man mit CSS machen?
- Warum zahlt sich ein CSS Framework aus?
- Wie verwendet man Bootstrap?
- Was stellt uns Bootstrap bereit?

Fragen zur Hausübung?

Wo gab es Schwierigkeiten?

Was war besonders einfach?

Wiederholung

Warum CSS?

- Trennung von Inhalt und Design
- Anwendbarkeit auf verschiedene HTML Seiten
- Layout kann verschieden formatiert sein
<http://www.csszengarden.com/>
- Aussehen unabhängig von Inhalt

Aufbau von CSS

- CSS bezieht sich auf HTML Elemente
- Syntax: Liste von Formateigenschaften
- Selektoren
 - Tag Name
 - Class Attribut (.)
 - Id Attribut (#)
- Reihenfolge der Codes ist entscheidend

```
tagName{  
    Eigenschaft: Wert;  
}  
  
.className{  
    Eigenschaft: Wert;  
}  
  
#idName{  
    Eigenschaft: Wert;  
}  
  
selector{  
    Eigenschaft: Wert !important;  
}
```

CSS in HTML einbinden

```
<link href="styles/style.css" rel="stylesheet" type="text/css">
```

```
/* in the css file */  
@import relative/path/subStyle.css;
```

- Link Tag im Header des HTML Grundgerüsts
- Reihenfolge auch hier entscheidend
- Relativer Pfad zum Stylesheet
- Stylesheets in eigenem Unterordner
- Aufsplitten mittels import (Übersichtlichkeit, Responsive Exceptions, Spezial Elemente, Views)
- Browser Caching beachten

Was kann man mit CSS machen?

- Text formatieren (Schriftart, Farben, Größe, ...)
- Textfluss-Kontrolle für Elemente (Bilder, Blockelemente, ...)
- innere (padding) und äußere (margin) Abstände und Rand
- Größen definieren (height, width, min und max)
- Positionen festlegen (Typ und Koordinaten)
- Hintergrund von Elementen (Bild, Farbe, Position, Wiederholung)
- Transitions (Elemente bewegen)

... und vieles mehr ...

Formatieren von Texten (wichtige Eigenschaften)

```
p{
  font-family: "Arial";
  font-size: 16px; /* pt, em */
  font-weight: normal; /* 400 */
  font-style: italic;
  color: #28C65D; /*hex or rgb: rgb(40,198,93), or rgba(40,198,93, 0.8) */

  text-transform: uppercase;
  text-decoration: underline;
  letter-spacing: 0.5;

  . . .
}
```

Textfluss Kontrolle

```
img.left-align{
  display: inline-block; /* block, none */
  float: left; /* right, none (i.e. to overwrite property) */
}

div.clear-float{
  clear: both; /* left, right, none */
}
```

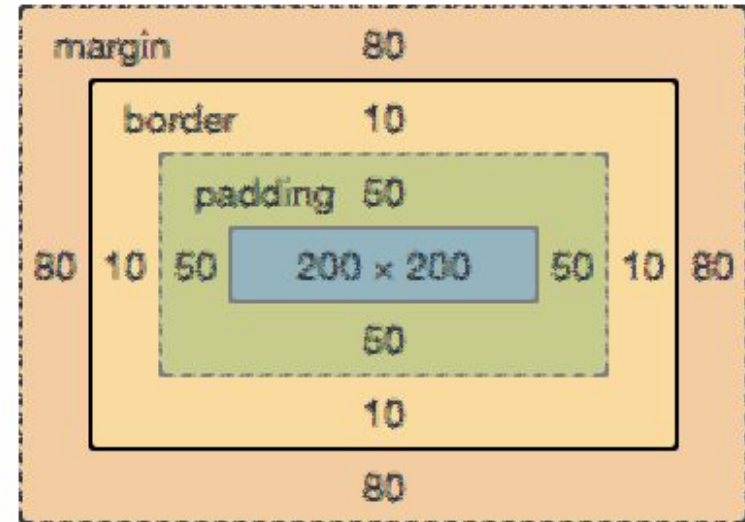
Abstände zu anderen Elementen

```
div.quadrat{
  width: 200px;
  height: 200px;

  padding: 50px;
  margin: 80px;

  border: 1px solid #ccc;
}

div#example{
  padding: 20px 10px 20px 10px;
  /* top, right, bottom, left */
  margin-bottom: 80px;
}
```



Größen definieren

```
spane.fixed-size{
    display: block;
    min-width: 200px;
    max-width: 350px;
    height: 20%; /*in relation to the browser window viewport */

    overflow: hidden;

    white-space: nowrap;

    . . .
}
```

Positionen festlegen

```
a#toTop{  
    position: fixed; /*absolute, relative*/  
    bottom: 20px;  
    right: 20px;  
  
    height: 35px;  
    width: 35px;  
  
    z-index: 2;  
}  
  
p{  
    text-align: center; /*left, right */  
}
```

Hintergrund von Elementen

```
ul{
  background: #00ff00 url("path/to/img.png") no-repeat center center;
}

div.sidebar-item{
  background-image: url("some/path/to/img.jpg");
  background-size: cover; /* contain, auto, and ... length, percentage */
  background-position: 20px /*xpos*/ 100px /*ypos*/;
}
```

Browser Spezifische Notation

```
-webkit-...    // For Webkit browser(Chrome, Safari...)  
-moz-...       // For Mozilla browser  
-o-...         // For Opera browser  
-ms-...        // For Microsoft browser  
none...        // For all browser(Newest version)
```

```
-webkit-transition: all 500ms ease;  
-moz-transition: all 500ms ease;  
-ms-transition: all 500ms ease;  
-o-transition: all 500ms ease;  
transition: all 500ms ease;
```

Transitions

```
div#some-item{  
    position: absolute;  
    left: 0px;  
}  
  
div#some-item.moving{  
    left: 300px;  
  
    transition: all 500ms ease;  
}  
  
// further Reading:  
// https://thoughtbot.com/blog/transitions-and-transforms
```


Links

- Tutorials

- https://developer.mozilla.org/de/docs/Web/Guide/CSS/Getting_started
- <https://www.w3schools.com/css/>

- Referenz

- <https://developer.mozilla.org/en-US/docs/Web/CSS/Reference>

- Specials

- <https://fonts.google.com/>
- <https://www.zeichenschatz.net/typografie/websichere-schriften.html>
- <https://thoughtbot.com/blog/transitions-and-transforms>
- <http://www.loremipsum.de/>
- <https://placeholder.com/>

Gruppenarbeit in 2er Teams

Bauen Sie das folgende HTML Dokument mit HTML und CSS nach:
`file:///var/www/FH-Technikum/FB_2019/Einheit2/public_html/index.html`

Beachten Sie

- die Abstände zum Browserfenster und der Elemente zueinander
- die verschiedenen Schriftarten (mit und ohne Serifen)
- die Innenabstände zu Rahmenlinien
- die unterschiedlichen Farben

Zeit dazu: 30 Minuten

Ein Seitentitel

Ein Untertitel

Eine Überschrift

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue dui dolore te feugait nulla facilisi. Lorem ipsum dolor sit amet, consetetur adipiscing elit, sed diam nonumy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.

Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue dui dolore te feugait nulla facilisi.

450 x 600

Powered by HTML.COM

Bootstrap 4

Warum zahlt sich ein CSS Framework aus?

- vordefiniertes Grid Layout
- viele vordefinierte Stile, die in allen Browsern gleich aussehen
- Geschwindigkeits-Gewinn durch vordefinierte Klassen
- responsive out-of-the-box

Wie verwendet man Bootstrap?

<https://getbootstrap.com/docs/4.1/getting-started/introduction/>

```
<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css" integrity="sha384-MCw98/SFnGE8fJT3GXwEOngsV7Zt27NXFoaoApmYm81iuXoPkF0JwJ8ERdknLPM0" crossorigin="anonymous">
```

Binden Sie dieses Skript im Header ein.
Danach können Sie loslegen.

1 2 3 4 5 6 7 8 9 10 11 12

BOOTSTRAP 4 GRID SYSTEM

Bootstraps GRID Layout

```
.col-*[1-12] // lg, md, sm, sx
```

```
<div class="row">  
  <div class="col-lg-4 col-sm-12"> ... </div>  
  <div class="col-lg-4 col-sm-12"> ... </div>  
  <div class="col-lg-4 col-sm-12"> ... </div>  
</div>
```


Was bietet Bootstrap?

- Unterstützung
 - beim Layouten (Aufteilen der Elemente)
 - bei Tabellen
 - bei Formularen
 - bei Positionierung
 - bei responsiven Eigenschaften
- Vordefinierte Elemente für
 - User Interaktionen: Hinweise, etc.
 - Navigationsleisten die responsive sind
 - Acordion: Collapse
 - Popover
 - Slider
 - ...

Bootstrap

```
.col-*-[1-12] // lg, md, sm, sx
```

```
<div class="row">  
  <div class="col-lg-4 col-sm-12"> ... </div>  
  <div class="col-lg-4 col-sm-12"> ... </div>  
  <div class="col-lg-4 col-sm-12"> ... </div>  
</div>
```

Ein Beispiel gemeinsam betrachten

<https://getbootstrap.com/docs/4.0/examples/blog/>

Responsive Mode im Browser vorzeigen!

Links

- Bootstrap Alternative TailWind
 - <https://tailwindcss.com/docs/what-is-tailwind/>
- Bootstrap
 - <https://getbootstrap.com/>
 - <https://getbootstrap.com/docs/4.1/getting-started/introduction/>
 - <https://getbootstrap.com/docs/4.1/examples/>

Hausübung

1. Bauen Sie das Checkout Formular aus Einheit 1 in Bootstrap um. Beachten Sie dabei Labels für die Felder und Rahmen für die einzelnen Abschnitte im Formular.

Verwenden Sie zumindest folgende Felder: Anrede, Vor- und Nachname, Alter/Geburtsdatum, Lieferadresse mit den diversen Feldern, Zahlungsweise, Zustimmung zu einer fiktiven Datenschutzerklärung. Validieren Sie ihren HTML Code.

2. Bauen Sie die Bootstrap Navigation ein. Alternativ zum Formular, soll es einen Link zu Ihrem Lebenslauf geben. Setzen Sie Ihren Lebenslauf in Bootstrap um. Validieren Sie ihren HTML Code.

Abgabe

- im Moodle unter Uebung 2
- NACHNAME_FE_Uebung2.zip (kein *.rar)
 - mit allen Files des Projektes
 - bitte keine versteckten Files die sonst in Ihrem Filesystem vorkommen *.ts, *.ps, usw.
- Fragen bitte im Diskussionsforum (Keine Scheu! Wir sind hier um zu Lernen!)
- versuchen Sie es wirklich alleine

Wenn noch Zeit bleibt...

Javascript Erfahrung abfragen

Frontend Basics

JavaScript und jQuery

Agenda

- Wiederholung und Besprechung der Hausübung
- Wiederholung zu Javascript
 - Warum Javascript?
 - Aufbau von Javascript, Einbinden von Javascript Dateien in HTML
 - Arbeiten mit Javascript
- JavaScript und Objektorientiertes Programmieren
- Warum verwendet man jQuery?
- Wie verwendet man jQuery?
- Was stellt uns jQuery bereit?

Fragen zur Hausübung?

Wo gab es Schwierigkeiten?

Was war besonders einfach?

Freiwillige vor

Sie haben im Grunde alle die Übung vollständig bewältigt und die Ergebnisse sind allgemein sehr gut. Wir können das Thema HTML also als erlernt betrachten und uns weiter auf Javascript konzentrieren.

- Gewöhnen sie sich an, auch bei einem kleinen Start, eine Ordnerstruktur anzulegen im Projekt.
- den .idea Ordner sollen Sie eben nicht mit ins zip packen :)
- arbeiten Sie sauber beim Einrücken der Elemente. Immer den gleichen Abstand z.b.
- räumen Sie Ihren Code auf - Abgaben mit allerhand Files die nichts mit der Aufgabe zu tun haben bringen weniger Punkte bei der Bewertung, wie sauber Sie gearbeitet haben.
- Benützen Sie die automatische code formatier Funktion, bitte! :) So mancher HTML Fehler wäre zu vermeiden gewesen.
- Versuchen Sie sich anzugewöhnen die Einstiegsdatei ins projet stets index.html zu benennen
- Achtung: H1 wird je HTML Dokument nur einmal verwendet - Sie bekommen sonst ein Thema bei SEO (Search Engine Optimisation)
- Das Label Element umschließt nur einen Text und ggf. noch ein einzelnes Formularfeld, Sie können darin nicht mehrere Inputs zusammenfassen. Labels haben auch kein Attribut "required".
- Schießen Sie nicht übers Ziel hinaus nur um etwas zu beweisen. Erledigen Sie die Aufgabe so, dass der Aufwand der Angabe im Verhältnis bleibt.
- Wenn Ihre Anwendung nicht out of the box läuft, dann geben Sie ein deploy Script oder einen online Link an, wo die Anwendung auch wirklich läuft.
- IDs sind bitte immer Eindeutig zu vergeben
- Seien Sie achtsam, wenn Sie Code im Entwicklungsprozess ändern - räumen Sie alte Attribute auf, die evtl. nicht passen (z.b. option hat keinen type)

JavaScript (ECMAScript 6)

European Computer Manufacturers Association

JavaScript ist nicht zu verwechseln mit Java

Wofür braucht man Javascript

- Interaktionen auf der GUI
 - HTML für den Aufbau
 - CSS für die Gestaltung
- Datenaustausch mit dem Server

Wie verwenden wir Javascript

- gekapselt in einer *.js Datei
- möglichst wiederverwendbar
- wird in der HTML Datei geladen

```
<script src="js/some.js0bject.js"></script>
```

Arbeiten mit Javascript

- Variablen Allocation
- Datentypen
- Stringoperationen
- Operatoren
- Schleifen
- Funktionen
- DOM Selektoren
- Objekte?
- Anonyme Funktionen?
- Event handler?
- Debugging?

Objektorientiertes Programmieren mit JS

- Prototypenbasierte Programmierung als Unterart der OOP
- theoretisch Klassenlos - praktisch aber schon, spätestens seit ECMA6
- Statt Instanz wird ein Klon (eines Objektes) erzeugt
- Verknüpfung zum Prototypen über “prototype” Notation

Varianten von Objekten in JS (1)

```
//simple data object
let someObject = {

    name:      "John",
    birthYear: 1989

}

//possible access
alert("Name:" + someObject.name);

let someOtherObject = {

    propertyName: "John",
    //anonyme function / in closure
    methodGetName: function(){
        return this.propertyName;
    }
}

//possible Call
someOtherObject.methodGetName();
```

Varianten von Objekten in JS (2)

```
class myObject{
  constructor(name){

    this.propertyName = name;
    this.init();
  }

  init(){

    //do some initial work
  }

  outputGreeting(greeting){

    alert(greeting + " "+this.name);
  }
}
```

//possible call

```
let greetings = new myObject("John");
greetings.outputGreeting("Hello");
```

Anonyme Funktionen

- für **einfache** und stark kohäsive Teile im Code
- in Frameworks überaus verbreitet

```
let sum = function(a, b) {  
  return a + b;  
}
```

```
sum(8,12)
```

```
(function(a, b) {  
  return a + b;  
})(12,8);
```

Event Handling

- `addEventListener`
- wird einem Element zugewiesen
- erwartet eine Event Art und eine Aktion
- Übergibt das aktuelle Event

```
window.addEventListener( "click", someCallback );  
window.addEventListener( "click", function(event){ ... } );
```

<https://developer.mozilla.org/en-US/docs/Web/Events>

Debugging

- `console.log("Some Output", myObject);`
- Developer Console in Chrome
- preserve Log
- Breakpoints

Weiterführende Links

- [https://developer.mozilla.org/de/docs/Web/JavaScript/Introduction to Object-Oriented JavaScript](https://developer.mozilla.org/de/docs/Web/JavaScript/Introduction_to_Object-Oriented_JavaScript)
- <https://developer.mozilla.org/de/docs/Web/JavaScript/Reference>
- <https://developer.mozilla.org/en-US/docs/Web/Events>
- <https://developer.mozilla.org/en-US/docs/Web/API/EventTarget/addEventListener>
- Literatur: Harmes und Diaz: JavaScript - Objektorientierung und Entwurfsmuster, 2008

Fragen?

jQuery

Alternativen

- vue.js
- react.js
- angular 9

Warum jQuery?

- erleichtert das Arbeiten
- unterstützt das Entkoppeln von HTML
- Browsercompatibilität
- Community bringt Plugins, Components, etc.
- vereinfachtes Event Binding
- vereinfachtes Ajax Handling (Server-Kommunikation)

Wie verwendet man jQuery?

- Einbinden in HTML
- `<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>`
- Klassischer Start, Charakteristische \$-Notation

```
// A $( document ).ready() block.  
  
$( document ).ready(function() {  
    console.log( "ready!" );  
});
```

Was stellt uns jQuery bereit

- `$()`
Macht aus DOM-Elementen ein jQuery Objekte mit vielen Möglichkeiten
- Selektor-Dialekt zum schnellen Arbeiten im DOM
- Event-Binding
- Ajax Handling

Selektoren

```
let $someDOMElement = $('#elementId');  
  
let $similarElements = $('.emementsClass');  
  
let $allTagsOfType = $('h2');  
  
//combinations and nested elements  
  
let $tagsOfSomeClass = $('img.left-align');  
  
let $listItems1 = $('ul.someListClass li');  
let $listItems2 = $('ul.someListClass').find('li');  
  
//attributes  
let firstName = $( "input[name='first-name']" ).val();
```

jQuery Object

```
> $
< f (a,b){return new r.fn.init(a,b)}
> jQuery
< f (a,b){return new r.fn.init(a,b)}
> let $link = $('.nav-link.active');
< undefined
> console.log($link);
```

VM637:1

```
r.fn.init(4) [a.nav-link.active, a.nav-link.active, a.nav-link.active,
a#contacts-tab-5ca370df694555ca370df12b0d22.nav-link.active, prevObject: r.fn.init(1)]
  ▶ 0: a.nav-link.active
  ▶ 1: a.nav-link.active
  ▼ 2: a.nav-link.active
    accessKey: ""
    assignedSlot: null
    ▶ attributeStyleMap: StylePropertyMap {size: 0}
    ▶ attributes: NamedNodeMap {0: class, 1: href, 2: data-toggle, class: class, href: href, data-toggle: data-toggle,
autocapitalize: ""
    baseURI: "https://moodle.technikum-wien.at/my/"
    charset: ""
    childElementCount: 0
    ▶ childNodes: NodeList [text]
    ▶ children: HTMLCollection []
    ▶ classList: DOMTokenList(2) ["nav-link", "active", value: "nav-link active"]
    className: "nav-link active"
    clientHeight: 35
    clientLeft: 0
    clientTop: 0
    clientWidth: 86
    contentEditable: "inherit"
    coords: ""
    ▶ dataset: DOMStringMap {toggle: "tab"}
    .. ...
```

Event Binding Basics

```
$('#someButton').on('click', function(e){  
    //do something  
});  
  
$('input[type=text]').on('keyup', function(e){  
    console.log(e.key);  
});
```

Event Binding in Objekten

```
let myObject = function($inputField){
    this.$textField = $inputField;
};

myObject.prototype.initGUIEvents = function(){

    let self = this;

    this.$textField.on("change", function(){
        self.alertChange( $(this).val() );
    })

};

myObject.prototype.alertChange = function(fieldValue){
    alert(fieldValue);
};
```


Weiterführende Links

- <https://jquery.com/>
- <https://learn.jquery.com/>
- <https://developers.google.com/speed/libraries/#jquery>
- <https://vuejs.org/>
- <https://angular.io/>
- <https://reactjs.org/>

Gruppenarbeit in 2er Teams

Führen Sie die bisher kennen gelernten Technologien zusammen:

Erstellen Sie mit HTML und Bootstrap eine GUI in der es benutzbar angeordnete Textfelder für Vor-, Nachname und Alter gibt. Außerdem soll es eine Auswahlmöglichkeit für das Geschlecht geben und einen Bereich zur Ausgabe.

Durch Klicken auf einen Button sollen geprüft werden ob die Einträge leer sind. In dem Fall soll eine Fehlermeldung im Ausgabebereich angezeigt werden, die dem Benutzer sagt was fehlt. Andernfalls sollen alle eingegebenen Werte in einem Sinnvollen deutschen oder englischen Satz im Ausgabefeld verbunden und ausgegeben werden.

Überlegen Sie zuerst wie das Interface aussehen könnte (Paper Prototype). Konzipieren Sie die Funktionalität. Welche Elemente, Events und Aktionen haben Sie zu behandeln.

Verwenden Sie einen Objektorientierten Ansatz.

Moodle Test am nächsten Termin

- Multiple Choice
- Einheit 1 - 3
- Besonders die Warum und Wie Folien
- Hintergründe zu den Technologien
- Javascript Sourcecode Basics