



CODERS.BAY

EINFÜHRUNG

BIBLIOTHEKEN VS. FRAMEWORKS

MVC FRAMEWORKS

BOOTSTRAP

FRAMEWORKS VS. BIBLIOTHEKEN

QUELLE: <https://t3n.de/news/library-vs-framework-unterschiede-1022753/>

FRAMEWORKS VS. BIBLIOTHEKEN

Ziel beider: Funktionsumfang eines Programms erweitern

BIBLIOTHEK

- Sammlung von Klassen und Funktionen
- Einige arbeiten dabei im Hintergrund und bleiben nach außen hin verborgen
- Programmierschnittstelle, kurz API (kurz für application programming interface), ermöglicht dann den Zugriff auf die nötigen Funktionen

FRAMEWORK

- spezielle Form der Library
- kein fertiges Programm, sondern muss an vorgesehenen Stellen ergänzt werden
- es werden geeignete Klassen implementiert und registriert, welche dann von dem Framework genutzt und kontrolliert werden können

FRAMEWORKS VS. BIBLIOTHEKEN

ZUSAMMENGEFASST

„Das Framework stellt eine wiederverwendbare, gemeinsame Struktur für Anwendungen zur Verfügung. Entwickler binden das Framework in ihre eigenen Anwendungen ein und erweitern es so, dass es ihre bestimmten Anforderungen erfüllt.“

Bei einer Library ruft der Entwickler innerhalb seines Codes die Klassen und Funktionen der Bibliothek direkt auf. Hingegen ruft das Framework, eine bestimmte Form der Library, vorgesehene Funktionen, wenn nötig selbständig auf.

MVC FRAMEWORK

QUELLEN:

https://de.wikipedia.org/wiki/Model_View_Controller

<https://www.datenbanken-verstehen.de/lexikon/model-view-controller-pattern/>

ZUSAMMENGEFASST

Das **Model View Controller Pattern** liefert 2 **große Vorteile**.

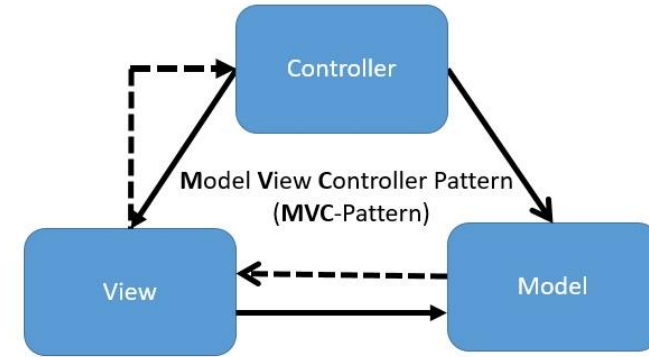
Darstellung, Datenmanagement und Programmlogik getrennt, sodass **Änderungen** innerhalb eines Teilbereichs **keine negativen Auswirkungen auf die jeweilig Anderen** haben. So z.B. neue Nutzeroberfläche an dieselben Model- und Controller-Klassen angeschlossen oder interne Operationen der Datenstruktur, zum Beispiel ein verbesserter Suchalgorithmus, unbemerkt eingebracht werden.

Darüber hinaus **erleichtert diese Aufteilung die Arbeit in Teams**. Sofern die Schnittstellen zwischen Model-, View- und Controller-Klassen bekannt sind, arbeiten die Entwickler aller drei Teilbereiche völlig autark.

BEISPIEL: MVC-PATTERN IN DER DATENBANKENTWICKLUNG

Die **Einkaufsabteilung eines Lebensmittelladens** muss auf Knopfdruck in Erfahrung bringen, wie viele der gelagerten Produkte abgelaufen sind und nachbestellt werden müssen. **Nahrungsmittel** besitzen jeweils (stark vereinfacht) einen **Namen** und ein **Mindesthaltbarkeitsdatum**. Dies stellt das **Model** dar. Der **View** beinhaltet zu beginn eine leere Liste und einen **Button**. Die **Betätigung des Buttons** veranlasst den **Controller**, **alle Artikel aus dem Model zu laden** und auf deren Verfallsdatum zu prüfen. Er **instruiert die Datenbank**, die betreffenden Artikel zu entfernen. Diese **Änderung wird dem View mitgeteilt**, welcher sie als auszusortieren in der Liste darstellt.

MVC FRAMEWORK



MODELL (MODEL)

- enthält Daten, die von der Präsentation dargestellt werden
- ist von Präsentation und Steuerung unabhängig
- Für den Browser ist die HTML-Seite der Datenkern seines Models. Aus der Perspektive des Gesamtsystems ist sie nur eine Sicht auf das Gesamtmodel, welches auf dem Server lokalisiert ist.

PRÄSENTATION (VIEW)

- zuständig für die Darstellung der Daten des Modells und die Realisierung der Benutzerinteraktionen
- kennt das Modell, dessen Daten sie präsentiert, ist nicht für Verarbeitung dieser Daten zuständig
- von der Steuerung unabhängig
- Browser kümmert sich um allgemeinen Funktionen, wie Darstellung von Text, Formularelementen und eingebetteten Objekten

STEUERUNG (CONTROLLER)

- verwaltet View und Model
- wird von View über Benutzerinteraktionen informiert, wertet diese aus und nimmt Anpassungen an View sowie Änderungen an Daten im Model vor
- Browser akzeptiert Formulareingaben und sendet diese ab oder nimmt das Anklicken eines Links entgegen -> sendet HTTP-Request an Server
- Controller-Programmteil verarbeitet die Daten der HTTP-Requests und erstellt neue View

MVC FRAMEWORK

JAVASCRIPT

Webseite kann Programmcode enthalten, normalerweise JavaScript, z. B. für die browserseitige Validierung von Formulareingaben oder Steuerungslogiken zum Nachladen von Inhalten

Dieser Programmcode lässt sich wiederum nach dem MVC-Muster gliedern und so als Teil des Gesamtsystems betrachten

Zu beachten ist, dass der Einsatz von clientseitiger Logik, welche nach dem MVC-Muster strukturiert ist, von einem serverseitig verwendeten MVC zu differenzieren ist

JAVASCRIPT- BIBLIOTHEKEN UND AJAX-ANBINDUNG

Hier wird ein Teil der Programme der Model-View-Controller-Architektur clientseitig im Browser eingesetzt, während ein anderer Teil, insbesondere das Model, auf dem Server verbleibt. JavaScript-Bibliotheken stellen vielfältige Widgets zur Verfügung. Diese Anwendungen nehmen eine Zwischenstellung zwischen Webanwendungen und desktopartigen Widget-Bibliotheken (kleine Programme) ein.

SERVERSEITIGE WEBANWENDUNGEN

Der serverseitige Controller wertet in der Regel die eintreffenden Daten (*Request*) des Browsers aus. Meist tritt er dabei als Moderator zwischen Model und View auf. Serverseitig werden unter dem View diejenigen Programmteile verstanden, die den HTML-Code für die Antwort (*Response*) erzeugen. Häufig arbeitet der View mit HTML-Templates, deren Platzhalter mit den Daten des Models ersetzt werden.

DER CONTROLLER ALS MITTLER ZWISCHEN MODEL, VIEW UND WEBSERVER

Ein typischer Funktionsablauf (ohne HTTP-Redirect):

```
Browser  -> HTTP-Request ->  Webserver -> Controller  
                                                <=> (beliebig häufig) Model oder View  
Browser  <- HTTP-Response <- Webserver <- Controller
```

- einfachster Fall: ordnet Model und View zu
- kann aber auch vielfältige weitere Aufgaben übernehmen
- Praxis variiert in Abhängigkeit vom persönlichen Programmierstil, Webservern, Programmiersprachen, Frameworks, dem Einsatz von Unit-Tests und Projektanforderungen

EXKURS: UNIT TESTS

Ein Modultest (auch von englisch **unit test** als Unittest oder als Komponententest bezeichnet) wird in der Softwareentwicklung angewendet, um die funktionalen Einzelteile (**Units**) von Computerprogrammen zu testen, d. h., sie auf korrekte Funktionalität zu prüfen.

HTTP-REDIRECT

Nach dem Ändern des Models (create oder update) empfiehlt sich ein HTTP-Redirect. Mit dieser Technik wird das irrtümliche mehrfache Absenden durch einen Seitenreload verhindert. Außerdem wird dadurch das Schreiben des vorherigen Datensatzes vom Lesen des nachfolgenden Datensatzes getrennt, so dass sich die Controller sinnvoller organisieren lassen.

Bei erfolgreicher Validierung greift die erste Anfrage, die noch den vorherigen Datensatz behandelt, schreibend auf das Model zu. Danach erfolgt ein Redirect. Die zweite Anfrage greift lesend zu und präsentiert bereits den nächsten Datensatz zur Bearbeitung. Für den Benutzer des Browsers fühlt sich das wie ein einziger Aufruf an.

BOOTSTRAP

QUELLEN:

- <https://www.ionos.at/digitalguide/websites/web-entwicklung/bootstrap-fuenf-alternativen-zum-twitter-framework/>
- <https://entwickler.de/online/javascript/bootstrap-4-2-579835887.html>
- <https://getbootstrap.com/>
- [https://de.wikipedia.org/wiki/sass_\(stylesheet-sprache\)](https://de.wikipedia.org/wiki/sass_(stylesheet-sprache))

UI-FRAMEWORKS

Aufgabenbereiche

der Oberfläche die passende Optik zu verleihen, dabei aber zu keinem Zeitpunkt die Usability aus den Augen zu verlieren.

plattformübergreifend funktionieren

gute SEO-Basis

viele Programmierer machen von **UI-Frameworks** (User Interface) Gebrauch

Wahl häufig auf Bootstrap, dessen Bestandteile für mobile Endgeräte und alle gängigen Browser optimiert sind und problemlos an die eigenen Vorstellungen angepasst werden können

große Verbreitungsgrad sorgt jedoch auch dafür, dass sich aus diesem Grund Webseiten stark ähneln

kleinerer Kritikpunkt die Komplexität des Frontend-Frameworks, das auf sehr viel CSS und JavaScript zurückgreift und zudem das **HTML-Mark-up** umfänglicher macht – was sich u. a. auf die Ladezeit des Webprojekts auswirken kann.

ZURB FOUNDATION

modular aufgebaute Foundation-Framework hat Ursprung in einem Style-Guide, den die Webdesign-Agentur ZURB zunächst ausschließlich für Projekte ihrer Kunden verwendete

Herzstück, ein zwölfspaltiges, **flexibles Grid-Layout**, ermöglicht die Gestaltung eines responsiven Layouts, das sich automatisch den verschiedenen Displaygrößen und Auflösungen der verschiedenen Endgeräte anpasst

basiert auf **Sass**, einer Stylesheet-Sprache, die das Erstellen und Bearbeiten der CSS-Dateien erheblich vereinfacht

wesentlich **weniger Templates**, und auch die Unterstützung durch andere Plattformen ist deutlich geringer

Individualität entwickelter Frontends geht

bietet verschiedene Kurse und **persönlichen Support für Ihr Frontend-Projekt** sowie eine spezielle Version des Frameworks zur Gestaltung von E-Mail-Newslettern an

PURE.CSS

Mitte 2013 veröffentlichte Yahoo mit Pure.CSS eine Basisstruktur für die Entwicklung von Web-Frontends

Pure basiert auf der Scalable and Modular Architecture for CSS, kurz SMACSS, die dafür sorgt, dass wiederkehrende Elemente wie Tabellen, Buttons oder Formulare vom Basis-Design (Schriftart, Layout etc.) getrennt sind und eigene Konventionen besitzen können

Das Frontend-Framework kann in einer responsiven und einer nichtresponsiven Variante heruntergeladen

Alle Module bauen auf dem Open-Source-Stylesheet Normalize.css auf, das die standardmäßigen Stile von HTML-Elementen durch optimierte, **browserübergreifende Stile** ersetzt

keinerlei JavaScript-Anwendungen

Startpunkt für Projekt und damit auch mit wesentlich größeren Freiheiten verbunden

SEMANTIC UI

2013 veröffentlichte der Programmierer Jack Lukic seine Framework-Lösung zur Frontend-Entwicklung unter dem Namen Semantic UI

Kernpunkt dieser Codesammlung, das Schreiben von HTML-Code durch nutzerfreundliche, intuitive Konventionen zu vereinfachen

über 3.000 semantische CSS-Klassen, die einfach anzuwenden sind und den Entwicklungsprozess optimieren sollen

über 20 verschiedene Designvorlagen in der Standardvariante

Umfang etwas komplexer als bei dem Twitter-Framework

mit einem hohen Maß an Einarbeitung verbunden

UIKIT

Open-Source-Lösung zur Frontend-Programmierung der in Hamburg beheimateten Firma YOOtheme, die eine große Expertise bei der Entwicklung von Webapplikationen sowie Themes für WordPress, Joomla und den hauseigenen Website-Builder YOOtheme Pro vorzuweisen hat

umfangreiche Sammlung aus HTML-, CSS- und JavaScript-Komponenten untersteht der freien MIT-Lizenz und kann daher bedenkenlos genutzt und verändert werden

Alle CSS-Snippets existieren sowohl in einer **Less**- als auch in einer **Sass-Variante** = Anpassung der Style-Vorlagen für geübte Programmierer erfrischend einfach ist

keinerlei Probleme bei der Darstellung gängige Internetbrowser

Die Kernkomponenten sind in folgende sechs Kategorien untergliedert

ausführliche Dokumentation

MATERIALIZE

basiert auf den Prinzipien des Material Designs, das Google 2015 eingeführt hat und mittlerweile in einem Großteil seiner Apps verwendet

Designkonzept baut auf kartenähnlichen Flächen auf („Flat Design“), aber dennoch **mit vielen Animationen und Schatten in Szene gesetzt** werden

12-spaltiges Grid-System, diverse CSS- und JavaScript-Komponenten, über 700 offizielle Material-Design-Symbole in einem Iconfont sowie **Roboto**, die Standardschriftart des Google-Designkonzepts

Sass geschriebene SCSS-Quelldateien

Die Dateien des Materialize-Projekts können Sie entweder auf [GitHub](#) oder auf der [Homepage materializecss.com](http://Homepage.materializecss.com) herunterladen. Dort finden Sie auch die zwei verfügbaren Templates „Starter“ und „Parallax“, sowie den Mark-up-Code für den Bezug der Projektdateien über das **Content Delivery Network cdnjs**.

ZUSAMMENFASSUNG: SO FINDEN SIE DIE PASSENDE BOOTSTRAP-ALTERNATIVE

vorgestellten Alternativlösungen haben meist viel mit Twitter-Framework gemeinsam:

zentrale Komponente ist in allen Fällen ein **Grid-System**, das sich durch Flexibilität auszeichnet und einen großen Anteil an der erfolgreichen **Realisierung geräteübergreifender Weboberflächen** hat. Die vordefinierten Mindest- bzw. Maximalmaße für die verschiedenen Displaygrößen unterscheiden sich zwar von Framework zu Framework, allerdings haben Sie hierbei als Entwickler in der Regel große Freiheiten bei der Anpassung. Das gilt nicht nur für Gitternetze, sondern auch für die übrigen CSS-Elemente, die in einigen Fällen außerdem sehr intuitiv eingebunden werden können, was bei Bootstrap nicht immer der Fall ist.

Unterschiede existieren bei den Klassensystemen der Bootstrap-Alternativen, wobei insbesondere Semantic UI heraussticht. Anstelle typischer CSS-Klassen hat das UI-Framework seinem Namen folgend über 3.000 eigene semantische Klassen integriert, die das Code-Programmieren noch intuitiver machen sollen.

Vereinfachung beim Codieren gewähren die meisten Sammlungen darüber hinaus durch die Unterstützung der Stylesheet-Sprachen Less und Sass, mit deren Hilfe Sie die Stylesheets Ihres Webinterfaces unkompliziert anpassen, bevor Sie diese anschließend in das gewohnte CSS-Format umwandeln, damit sie der Internetbrowser des Besuchers lesen und richtig interpretieren kann.

Lediglich Pure.CSS verzichtet auf beide **CSS-Präprozessoren**, bietet mit der Scalable and Modular Architecture for CSS (SMACSS) aber einen interessanten Ansatz für die Nutzung von CSS.

ZUSAMMENFASSUNG: SO FINDEN SIE DIE PASSENDE BOOTSTRAP-ALTERNATIVE

nicht möglich, eine einzelne zur Königslösung für die Frontend-Programmierung zu deklarieren

Entscheidung in erster Linie von Ansprüchen des Webprojekts abhängig und stelle dir folgenden Fragen:

Sind die Komponenten ansprechend **strukturiert**?

Bietet das **Grid-System** die gewünschten Optionen?

Wie wichtig ist die **Unterstützung älterer Browserversionen**?

Entspricht der **Umfang** an CSS- und JavaScript-Modulen den Anforderungen?

Ist die **Größe der Codes** (komprimiert und nicht komprimiert) verhältnismäßig?

Sind große **Freiheiten bei der Anpassung der Codes** notwendig und wenn ja, welche Möglichkeiten offeriert das UI-Framework?

Legst du Wert auf eine große **Community** (Weiterentwicklung, Erweiterungen, Templates, Foren, Tutorials, etc.)?

Blick in die Online-Manuals empfehlenswert für anschauliche Code-Beispiele der Module

selbst ausprobieren, mit Online-Tool Codeply: testen der Komponenten der verschiedenen UI-Frameworks direkt im Browser, ohne eine einzige Datei heruntergeladen zu haben

WAS IST BOOTSTRAP?

2010 von Twitter („Twitter Bootstrap“) entwickelt

Ziel: einheitliche Bibliothek für Gestaltung von Weboberflächen

sollte gemeinsame Basis schaffen, mit der alle Mitarbeiter arbeiten konnten, um schnell und einfach Websites zu gestalten.

Bootstrap-Framework besteht im einfachsten Fall aus einer CSS-Datei (*bootstrap.min.css*) und einer JavaScript-Datei (*bootstrap.min.js*)

Viele Designelemente können mit reinen CSS-Regeln umgesetzt werden

andere Komponenten, z. B. Dialogboxen, wird JavaScript benötigt

basiert auf Stylesheetsprache Sass

Normalfall Sass-Dateien nicht anpassen, da man mit der fertigen *bootstrap.min.css*-Datei arbeitet

ist ein Frontend-Framework, mit dem man Websites gestalten kann

werden HTML- und CSS-Vorlagen bereitgestellt, um unterschiedlichste Websiteelemente darzustellen

z.B.: Formulare, Buttons, Tabellen, Navigation und Grid-System für Layouts

durch JavaScript-Module möglich, Interaktionen (z. B. eine Bilder-Slideshow, Tabs und Dialogboxen) in Website einzubinden

alle Voraussetzungen, um responsive Webdesigns zu gestalten, die dann auch auf Smartphones oder Tablets optimal dargestellt werden

DAS MACHT BOOTSTRAP SO INTERESSANT

- Abgedeckt sind unter anderem die folgenden Elemente:
 - Buttons
 - Navigationselemente
 - Thumbnail-Übersicht
 - Dropdown-Menüs
 - Warnhinweise
 - Verlaufsanzeigen
 - Responsive Video-Einbettung

PROGRAMMIERKENNTNISSE SIND EINE WICHTIGE VORAUSSETZUNG

Auch wenn Bootstrap vorgefertigte Inhalte präsentiert, ohne HTML- und CSS-Know-how nicht sehr weit kommen
Einarbeitungszeit ist auch bei versierten Entwicklern sinnvoll – und zahlt sich in der Regel aus
diverse erstellte **Templates** anderer User, teils kostenfrei teils kostenpflichtig zum Download angeboten werden
Wer regelmäßig mit HTML, CSS und JavaScript zu tun hat, wird die Vorteile des Frameworks schnell zu schätzen wissen.

Vorteile

- enorme Zeitersparnis
- sehr guten Möglichkeiten bezüglich Responsive Webdesign
- regelmäßige Updates und Hilfestellungen durch große Community
- erstklassige Dokumentation zur Benutzung
- Open-Source-Projekt
- eines der beliebtesten Frameworks für Webdesign

EXKURS: SASS

Sass (***S**yntactically **A**wesome **S**tylesheets*) ist eine Stylesheet-Sprache, die als CSS-Präprozessor mit Variablen, Schleifen und vielen anderen Funktionen, die Cascading Style Sheets (CSS) nicht mitbringen, die Erstellung von CSS vereinfacht und die Pflege großer Stylesheets erleichtert.

BOOTSTRAP

Bootstrap besteht aus einzelnen Komponenten, die in eine Website eingefügt werden können. Dabei handelt es sich um HTML-Codes, die mit CSS-Klassen und ggf. JavaScript-Code ausgestattet sind.

Bootstrap 4 ist modular aufgebaut und wird in folgende vier Bereiche gegliedert:

- Layout

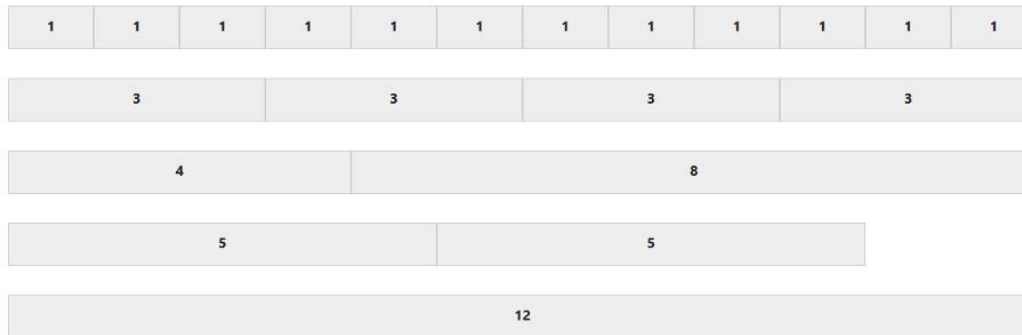
- Inhalt

- Komponenten

- Werkzeuge

LAYOUT

Standardraster in Bootstrap



-Herzstück ist das Raster

-Standardmäßig 12-spaltiges Rasterlayout mit einer Breite von 1140 Pixel (*.container*).

-Dabei wird Rasterbreite entsprechend der Displaygröße via CSS-Breakpoints angepasst

-Alternative *.container-fluid*, die den Container immer über die komplette Displaygröße darstellt

DAS RASTER

Ein Raster besteht aus Zeilen (.row) und Spalten (.col), wobei die Spalten mithilfe von CSS-Klassen passend für das jeweilige Endgerät angeordnet werden können.

Normalerweise sind alle Rasterspalten in Bootstrap gleich groß. Durch Weglassen der Spaltenbreite (XX-Angabe) lassen sich jedoch Layouts mit unterschiedlich breiten Spalten darstellen

```
<div class="container">
  <div class="row">
    <div class="col-md-2">Sidebar</div>
    <div class="col-md-10">Inhalt</div>
  </div>
</div>
```

```
<div class="container">
  <div class="row">
    <div class="col">Sidebar 1 (.col)</div>
    <div class="col-7">Inhalt (.col-3)</div>
    <div class="col">Sidebar 2 (.col)</div>
  </div>
</div>
```

Displaygröße	Sehr klein	Klein	Mittel	Groß	Sehr groß
	Smartphone (Hochformat)	Smartphone (Hochformat)	Smartphone (Hochformat)	Smartphone (Hochformat)	Smartphone (Hochformat)
CSS-Präfix	.col-XX	.col-sm-XX	.col-md-XX	.col-lg-XX	.col-xl-XX

RESPONSIVE RASTERKLASSEN IN BOOTSTRAP

- noch weitere Regeln, um Raster anzupassen
 - mithilfe der Klasse `.w-100` einen Spaltenumbruch zu erzwingen
 - Vermischung von Klassen möglich
 - z. B. eine Sidebar, die auf dem Desktop-PC nur 20 Prozent der Breite einnimmt, auf dem Smartphone die Hälfte des Displays abdecken

HORIZONTALLE UND VERTIKALE ANORDNUNG VON ZEILEN UND SPALTEN

Neuerung in Bootstrap 4 ist die Möglichkeit der horizontalen und vertikalen Anordnung von Zeilen und Spalten innerhalb des Rasters

folgende Aufgaben beim Layoutentwurf kein Problem:

- Spaltenzwischenräume entfernen (`.no-gutters`)
- Spaltenreihenfolge anpassen (`.order-XX`, `.order-first`, `.order-last`)
- Versatz zwischen Spalten einfügen (z. B. `.offset-md-4`)
- Verschachtelungen von Rastern (neue `.row` einfügen)

```
<div class="container">
  <div class="row">
    <div class="col">
      <h1>Vertikale Ausrichtung (Zeile)</h1>
    </div>
  </div>
  <div class="row align-items-center test">
    <div class="col">Sidebar 1 (.col)</div>
    <div class="col-7">Inhalt (.col-3)</div>
    <div class="col">Sidebar 2 (.col)</div>
  </div>
  <div class="row">
    <div class="col">
      <h1>Vertikale Ausrichtung (Spalte)</h1>
    </div>
  </div>
  <div class="row test">
    <div class="col align-self-start">.col (oben)</div>
    <div class="col align-self-center">.col
(mittig)</div>
    <div class="col align-self-end">.col (unten)</div>
  </div>
```

```
<div class="row">
  <div class="col">
    <h1>Horizontale Ausrichtung (Zeile)</h1>
  </div>
</div>
<div class="row justify-content-start">
  <div class="col-2">.col-2 (links)</div>
</div>
<div class="row justify-content-center">
  <div class="col-2">.col-2 (mittig)</div>
</div>
<div class="row justify-content-end">
  <div class="col-2">.col-2 (rechts)</div>
</div>
<div class="row justify-content-around">
  <div class="col-2">.col-2 (ringsherum)</div>
  <div class="col-2">.col-2 (ringsherum)</div>
</div>
<div class="row justify-content-between">
  <div class="col-2">.col-2 (dazwischen)</div>
  <div class="col-2">.col-2 (dazwischen)</div>
</div>
</div>
```

BILDER UND BILDUNTERSCHRIFTEN

- Problem, Bilder nur auf großen Displays optimal angezeigt
- Auf kleineren Bildschirmen zu groß, zu klein oder verzerrt dargestellt
- Abhilfe schafft Klasse *.img-fluid*, denn dann wird Bild responsive mit maximaler Breite von 100 Prozent
- Bildunterschriften sowie abgerundete Ecken sind einfach umsetzbar

BILDER UND BILDUNTERSCHRIFTEN

```
<div class="container">
  <div class="row">
    <div class="col">
      <h1>Bild (nicht responsive)</h1>
    </div>
  </div>
  <div class="row">
    <div class="col-6">
      
    </div>
  </div>
  <div class="row">
    <div class="col">
      <h1>Bild (responsive)</h1>
    </div>
  </div>
</div>
```

```
<div class="row">
  <div class="col-6">
    
  </div>
</div>
<div class="row">
  <div class="col">
    <h1>Bild mit Ecken</h1>
  </div>
</div>
<div class="row">
  <div class="col-6">
    
  </div>
</div>
</div>
```

TABELLEN

- große Datenmengen übersichtlich darstellen
- Tabelle wird mit Klasse `.table` initiiert
- kann Tabelle mittels der CSS-Klassen aus Tabelle nebenstehend formatieren
- einzelnen Tabellentypen lassen sich kombinieren.

Beschreibung	CSS-Klasse	Wo einfügen
Gestreifte Tabelle	<code>.table-striped</code>	<code><table></code>
Tabelle mit Rand	<code>.table-bordered</code>	<code><table></code>
Tabelle mit Zeilenmarkierung beim Überfahren mit der Maus	<code>.table-hover</code>	<code><table></code>
Kompakte Tabelle	<code>.table-sm</code>	<code><table></code>
Farbige Zellen oder Spalten oder komplette Tabellen	<code>.table-active</code> <code>.table-primary</code> <code>.table-secondary</code> <code>.table-success</code> <code>.table-danger</code> <code>.table-warning</code> <code>.table-info</code> <code>.table-light</code> <code>.table-dark</code>	<code><tr></code> , <code><td></code> oder <code><table></code>

KOMPONENTEN

- Herzstück von Bootstrap
- helfen bei Strukturierung der Website

BUTTONS EINBINDEN

- Schaltflächen bzw. Buttons werden eingesetzt, um Aufmerksamkeit des Besuchers zu lenken
 - um auf neue Seite zu verweisen
 - „Call to Action“ auszulösen, wie das Ausfüllen eines Formulars
- heben sich durch Form und Farbgebung deutlich ab
- durch Vergabe der *.btn*-Klasse eine simple Darstellung von Inhalten wie Links als Buttons

```
<p><a href="#" class="btn btn-primary" >.btn .btn-primary</a></p>
```

```
<p><a href="#" class="btn btn-secondary">.btn .btn-secondary</a></p>
```

```
<p><a href="#" class="btn btn-success">.btn .btn-success</a></p>
```

```
<p><a href="#" class="btn btn-danger">.btn .btn-danger</a></p>
```

```
<p><a href="#" class="btn btn-warning">.btn .btn-warning</a></p>
```

```
<p><a href="#" class="btn btn-info">.btn .btn-info</a></p>
```

```
<p><a href="#" class="btn btn-light">.btn .btn-light</a></p>
```

```
<p><a href="#" class="btn btn-dark">.btn .btn-dark</a></p>
```

```
<p><a href="#" class="btn btn-link">.btn .btn-link</a></p>
```

```
<p><a href="#" class="btn btn-primary btn-block" >.btn-block</a></p>
```

```
<p><a href="#" class="btn btn-primary float-right">.float-right</a></p>
```

```
<p><a href="#" class="btn btn-primary btn-lg">.btn-lg</a></p>
```

```
<p><a href="#" class="btn btn-primary btn-sm">.btn-sm</a></p>
```

```
<p><a href="#" class="btn btn-outline-primary">.btn-outline-primary</a></p>
```

```
<p class="text-center"><a href="#" class="btn btn-primary">.text-center (p-Element)</a></p>
```

BREADCRUMB

- Anspielung auf die Geschichte von Hänsel und Gretel
- Besucher verständlich darzustellen, wo genau er sich aktuell auf der Website befindet
- einfache HTML-Liste mit Links, die mit Trennzeichen dargestellt werden
- Mit richtigen Auszeichnung auch von Screenreadern gelesen werden
 - Barrierefreiheit
 - Suchmaschinenoptimierung

```
<div class="container">
  <div class="row">
    <div class="col">
      <nav aria-label="breadcrumb">
        <ol class="breadcrumb">
          <li class="breadcrumb-item">
            <a href="#">Startseite</a>
          </li>
          <li class="breadcrumb-item">
            <a href="#">Ebene 1</a>
          </li>
          <li class="breadcrumb-item active"
            aria-current="page">
            Aktuelle Seite
          </li>
        </ol>
      </nav>
    </div>
  </div>
</div>
```

NAVBAR

- Navigation: einfache Lösung -> Navbars umzusetzen
- Navbars standardmäßig responsive und werden über ganze verfügbare Breite angezeigt (fluid)
- können farblich angepasst werden
- Navbars im Normalfall nicht mit ausgedruckt (mit Klasse *.d-print* umgegangen werden)
- Bestandteile:
 - Firmenname als Link, Überschrift oder Bild (*.navbar-brand*)
 - Die eigentliche Navigation (*.navbar-nav*)
 - Hamburgermenü für Smartphoneansicht (*.navbar-toggler*)
 - Formulare (*.form-inline*)
 - Text (*.navbar-text*)
- normalerweise auf Website ganz oben angezeigt und verschwinden beim Scrollen
- durch Setzen von *.fixed-top* oder *.fixed-bottom* kann Navbar beim Scrollen oben oder unten am Bildschirmrand fixiert werden

Beispiele: <https://getbootstrap.com/docs/4.0/components/navbar/>

HILFSKLASSEN

- verschiedene Hilfsklassen stehen zur Verfügung
- mit *.border*-Klassen können Ränder um ein Element mit oder ohne Farbe angezeigt werden

HILFSKLASSEN: BORDER

```
<div class="container">
  <div class="row">
    <div class="col">
      <h1>Ränder</h1>
      <span class="border"></span>
      <span class="border-top"></span>
      <span class="border-right"></span>
      <span class="border-bottom"></span>
      <span class="border-top border-left"></span>
    </div>
  </div>
  <div class="row">
    <div class="col">
      <h1>Farbige Ränder</h1>
      <span class="border border-primary"></span>
      <span class="border-top border-secondary"></span>
      <span class="border-right border-success"></span>
      <span class="border-bottom border-danger"></span>
      <span class="border-top border-left border-warning"></span>
    </div>
  </div>
</div>
```


HILFSKLASSEN: FARBEN EINSTELLEN

- Klassen, um Farbe von Elementen anzupassen
- Egal ob bei Buttons, Navigationsleisten, Texte oder Umrandungen

```
<div class="container">
  <div class="row justify-content-between">
    <div class="col-5">
      <h1>Textfarbe</h1>
      <p class="text-primary">.text-primary</p>
      <p class="text-secondary">.text-secondary</p>
      <p class="text-success">.text-success</p>
      <p class="text-danger">.text-danger</p>
      <p class="text-warning">.text-warning</p>
      <p class="text-info">.text-info</p>
      <p class="text-light bg-dark">.text-light</p>
      <p class="text-dark">.text-dark</p>
      <p class="text-muted">.text-muted</p>
      <p class="text-white bg-dark">.text-white</p>
    </div>
    <div class="col-5">
      <h1>Hintergrundfarbe</h1>
      <div class="p-2 mb-2 bg-primary text-white">.bg-primary</div>
      <div class="p-2 mb-2 bg-secondary text-white">.bg-secondary</div>
      <div class="p-2 mb-2 bg-success text-white">.bg-success</div>
      <div class="p-2 mb-2 bg-danger text-white">.bg-danger</div>
      <div class="p-2 mb-2 bg-warning text-dark">.bg-warning</div>
      <div class="p-2 mb-2 bg-info text-white">.bg-info</div>
      <div class="p-2 mb-2 bg-light text-dark">.bg-light</div>
      <div class="p-2 mb-2 bg-dark text-white">.bg-dark</div>
      <div class="p-2 mb-2 bg-white text-dark">.bg-white</div>
    </div>
  </div>
</div>
```



CODERS.BAY

DANKE