

Teil 7: Bootstrap Framework verwenden

Bis hierhin haben wir alles HTML und CSS von Grund auf selber programmiert. Unser Projekt, das *Web Portfolio*, hat bereits einige HTML-Seiten mit etwas Inhalt, eine noch unfertige Navigation und etwas Styling.

Ganz von Hand unser Projekt zu einem kompletten und modernen Webauftritt weiter zu entwickeln wäre **eine riesige Menge Arbeit**! Hier ist eine Liste mit den paar wichtigsten Aufgaben, die wir noch bewältigen müssten (keine Angst, es gibt eine Lösung!):

- Schriftart und Textgrösse für alle Überschriften und Abschnitte anpassen.
- Knöpfe, Textfelder, etc. schön gestalten (sehen standardmässig recht veraltet aus).
- Abstände definieren und Elemente auf der Seite positionieren (ist sehr schwierig).
- Einen einheitlichen Stil definieren, damit alle verschiedenen Browser in etwa das Gleiche anzeigen.
- Die Webseiten optimieren für verschiedene Bildschirmgrößen (Mobile, Tablet und Desktop)
- Und Einiges mehr...

Zum Glück sind wir nicht alleine mit diesen Aufgaben - dies betrifft mehr oder weniger alle Webdesigner. Da es kaum sinnvoll ist, wenn alle immer wieder die gleichen Aufgaben lösen, sind sogenannte *Frameworks* entstanden, die diese Arbeit viel einfacher machen.



Das bekannteste dieser *Frameworks* heisst **Bootstrap** und wurde von Twitter entwickelt. Das *Bootstrap-Framework* ist eine enorme erleichtert für uns als Webentwickler. Wir werden Schritt-für-Schritt entdecken, was wir mit *Bootstrap* tun können. Hier aber schon mal eine Kostprobe, welche Auswirkungen *Bootstrap* auf unser Projekt haben kann:

Beispiel ohne Bootstrap.



Beispiel mit Bootstrap.



Mit *Bootstrap* sehen die Elemente nicht nur besser aus, sondern sie werden auch in allen Browsern praktisch gleich dargestellt.

Bootstrap einsetzen

Dokumentation

Die offizielle Dokumentation von [Bootstrap](#) ist der beste Ort, um zu sehen, was man alles mit Bootstrap machen kann. Sie sollten sich diese Webseiten irgendwo speichern, denn Sie werden dort immer wieder etwas nachschlagen müssen.

Bootstrap-CSS einbinden

Der wichtigste Teil von Bootstrap besteht aus einem grossen CSS. Für etwas komplexere Elemente gibt es auch noch eine JavaScript-Datei. Wir werden aber für den Moment nur die CSS-Datei verwenden.

Nun wollen wir das Bootstrap-CSS in unsere HTML-Seiten einbinden:

1. Öffnen Sie die [Bootstrap-Webseite](#). Sie sehen dort, dass es mehrere Möglichkeiten gibt, Bootstrap herunterzuladen. Wir wählen die Variante über das *BootstrapCDN*. Damit müssen wir eigentlich gar nichts herunterladen, sondern nur im HTML angeben, wo die zusätzliche CSS-Datei liegt. Sie wird dann bei jedem Aufruf unserer Webseite direkt von den *CDN*-Servern heruntergeladen. *CDN*-Server sind Server, die über die ganze Welt verteilt sind, damit sie eine Datei sehr schnell ausliefern können.
2. Kopieren Sie also vom Abschnitt *BootstrapCDN* die erste Zeile mit dem `link`-Element.
3. Fügen Sie das `link`-Element in ihre HTML-Seiten ein. Wichtig ist, dass Sie es oberhalb von unserem `link`-Element für die `main.css`-Datei einfügen. Dadurch hat unser `main.css` eine höhere Wichtigkeit. Das bedeutet, dass wir CSS-Regeln aus dem Bootstrap-CSS mit unseren eigenen Regeln überschreiben können.
4. Kopieren Sie die drei Zeilen für JS, Popper.js und jQuery.
5. Fügen Sie diese JavaScript-Dateien zuunterst in den Body-Bereich ihrer HTML-Seite ein.

So sollte Ihre HTML-Dateien nun etwa aussehen (evtl. mit einer neueren Versionsnummer und ein paar „integrity“-Zahlen von Bootstrap):

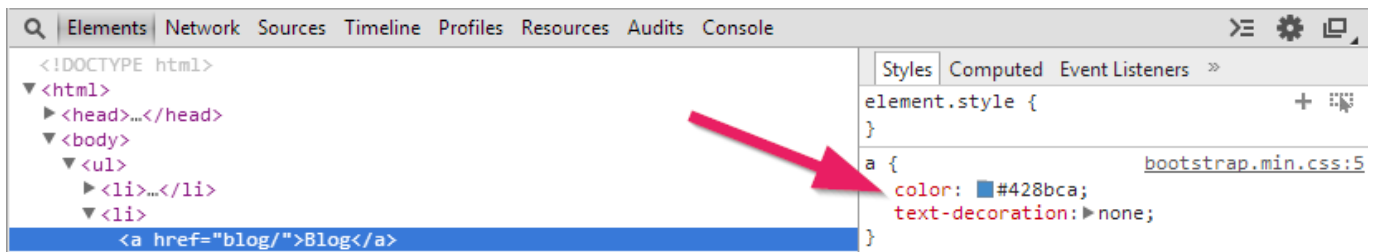
index.html

```
<html>
  <head>
    <meta charset="utf-8">
    <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.c
ss">
    <link rel="stylesheet" href="/main.css">
    <title>Web Portfolio von Marco</title>
  </head>
  <body>
```

```
    <!-- Hier steht irgend ein Inhalt -->
    <script
src="https://code.jquery.com/jquery-3.3.1.slim.min.js"></script>
    <script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.3/umd/popper.min.
js"></script>
    <script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/js/bootstrap.min.js"
></script>
  </body>
</html>
```

Wie funktioniert Bootstrap?

Manche *Styles* aus dem Bootstrap-CSS werden direkt auf HTML-Elemente angewandt. Schauen wir uns zum Beispiel einen Link in *Chrome DevTools* an, sehen wir, wie im Bootstrap-CSS die Link-Farbe geändert wird und mit `text-decoration` der Unterstrich entfernt wird:



Viele Bootstrap-Styles werden aber erst aktiviert, wenn man eine entsprechende **CSS-Klasse** hinzufügt.

Bootstrap Klassen verwenden

Als Beispiel schauen wir uns an, wie mit Bootstrap die Textausrichtung geändert werden kann. In der Bootstrap-Dokumentation unter [Ausrichtungsklassen](#) finden Sie eine kurze Anleitung zur Textausrichtung.

Wenn wir also zum Beispiel den h2-Titel zentrieren möchten, müssten wir die Klasse `text-center` hinzufügen:

```
<h2 class='\"text-center\"'>Willkommen!</h2>
```

Unser h1-Titel hat bereits ein `class`-Attribut. Wenn wir auch diesen Titel zentrieren möchten, können wir das `text-center` als zweite Klasse hinzufügen. **Mehrere CSS-Klassen werden immer durch Leerschläge getrennt:**

```
<h1 class='\"title text-center\"'>Web Portfolio von Marco</h1>
```

Was aber, wenn wir einen grösseren Teil unserer Webseite zentrieren möchten? Da wäre es recht mühsam, wenn wir bei jedem Element einzeln ein `text-center` hinzufügen müssten. Es ist deshalb möglich, dass man auf einem übergeordneten Element eine Klasse hinzufügt. Viele CSS-Regeln können so auf alle Unterelemente weitervererbt werden.

Wir könnten sogar auf dem `body`-Element eine `text-center`-Klasse hinzufügen. Damit wird gleich alles zentriert. Wenn man aber nur einen Teil zentrieren möchte, verwendet man oft sogenannte `div`-Elemente.

div-Elemente

Mit dem `<div>`-Element werden Elemente zu einem Block gruppiert. Für Bootstrap werden wir `div`-Elemente recht oft verwenden, um Styles auf einen ganzen Abschnitt der Seite anzuwenden.

Eine zentrale CSS-Klasse in Bootstrap ist `container`. Der `container` passt den Inhalt automatisch an die Seitenbreite an und sorgt für einen sinnvollen Abstand zum Rand. Sie sollten auf Ihrer Bootstrap-Seite immer ein `div`-Element mit der `container`-Klasse haben. Lesen Sie die [Beschreibung von Container](#) auf der Bootstrap-Webseite.

Wir packen nun den Hauptinhalt unserer Startseite zwischen ein `<div class="container">` und `</div>`. (Um die Navigation werden wir uns weiter unten noch speziell kümmern.)

Der `body`-Teil Ihrer Hauptseite sollte nun wie folgt aussehen (achten Sie auf sauberes Einrücken der Elemente, damit der Code übersichtlich bleibt):

index.html

```
<body>
  <ul>
    <li><a href="/">Home</a></li>
    <li><a href="/blog/">Blog</a></li>
    <li><a href="/projekte/">Projekte</a></li>
    <li><a href="/kontakt/">Kontakt</a></li>
  </ul>

  <div class="container">
    <h1 class="title">Web Portfolio von Marco</h1>

    <h2>Willkommen!</h2>

    <p>Schön, dass Sie auf meiner Seite vorbeischaauen.</p>

    <p>Sie können sich ruhig etwas umsehen. Im Blog dokumentiere ich meine
    Erfahrungen beim Programmieren. Daneben können Sie auch meine Webprojekte
    anschauen. Viel Spass.</p>
```

```


<p>Marco :-)</p>
</div>
</body>
```

Im Browser beginnen die `div`-Elemente **immer auf einer neuen Zeile**. Weil sie die ganze verfügbare Breite in Anspruch nehmen nennt man solche Elemente auch [Block-Elemente](#).

span-Elemente

Im Gegensatz zu `<div>`-Elementen beginnt ein ``-Element nicht auf einer neuen Zeile. Ein `span`-Element ist nur so breit wie sein Inhalt. Man nennt solche Elemente auch [Inline-Elemente](#), weil sie sich innerhalb einer Zeile einbetten.

Hier ein Beispiel, wie man ein `span`-Element einsetzen könnte:

```
Ich möchte <span class="text-warning">diese Worte</span> besonders
hervorheben.
```

Navigation mit Bootstrap

Nun wollen wir endlich die Navigation schöner gestalten mit der Hilfe von Bootstrap. In der Bootstrap-Dokumentation wird eine [Navbar](#) beschrieben. Das Beispiel dort ist jedoch sehr komplex und beinhaltet Menüs, Textfelder und Knöpfe. Wir wählen eine etwas einfachere Variante.

Kopieren Sie den folgenden Code für die Navigation:

index.html

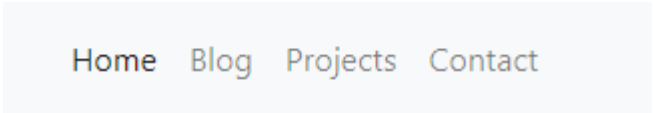
```
<nav class="navbar navbar-expand-lg navbar-light bg-light">
  <div class="container">
    <button class="navbar-toggler" type="button" data-toggle="collapse"
data-target="#navbarMenu">
      <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbarMenu">
      <ul class="navbar-nav">
        <li><a class="nav-link active" href="/">Home</a></li>
        <li><a class="nav-link" href="/blog/">Blog</a></li>
        <li><a class="nav-link" href="/projects/">Projekte</a></li>
        <li><a class="nav-link" href="/contact/">Kontakt</a></li>
      </ul>
    </div>
  </div>
```

```
</div>
</nav>
```

Erklärung

1. Das äusserste nav-Element mit navbar als Klasse bewirkt, dass ein Balken oben am Rand dargestellt wird. Ein nav ist eigentlich das genau Gleiche wie ein div, ausser dass man an Suchmaschinen einen kleinen Hinweis gibt, dass da eine Navigation kommt.
2. Das nächste div mit der Klasse container bewirkt, dass die Abstände vom Rand gleich sind wie bei unserem Hauptinhalt. Beachten Sie jeweils das schliessende Tag `</div>`, welches dazugehört.
3. Das button-Element mit Klasse navbar - toggler ist der Knopf in der mobilen Ansicht, um das Menu auszuklappen. Verkleinern Sie mal das Browser-Fenster in der Breite. Dann sollte der Knopf erscheinen.
4. Im Gegensatz zur vorherigen Variante, haben wir bei jedem Link eine Klasse nav - link hinzugefügt. Diese bewirkt, dass der Link in der Navigation den entsprechenden Stil erhält.
5. Beachte: Das **aktive** Navigationselement ist jeweils mit der zusätzlichen Klasse active markiert. Dieses erscheint dann hervorgehoben.

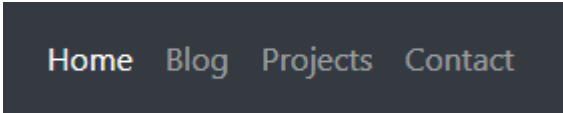
So ungefähr sollte es jetzt aussehen:



Home Blog Projects Contact

Dunkle Variante

Es gibt weitere Varianten der Bootstrap-Navigation:



Home Blog Projects Contact

Um diese zu aktivieren, ändern Sie die Klasse navbar-light in navbar-dark und bg-light in bg-dark um.

Optimieren für Mobile

Mit Bootstrap und der Navigation sieht die Webseite in einem Desktop-Browser schon recht ansprechend aus. Versuchen Sie mal, die Webseite auf einem Mobiltelefon aufzurufen (dazu müssen Sie die Seite [veröffentlichen](#)).

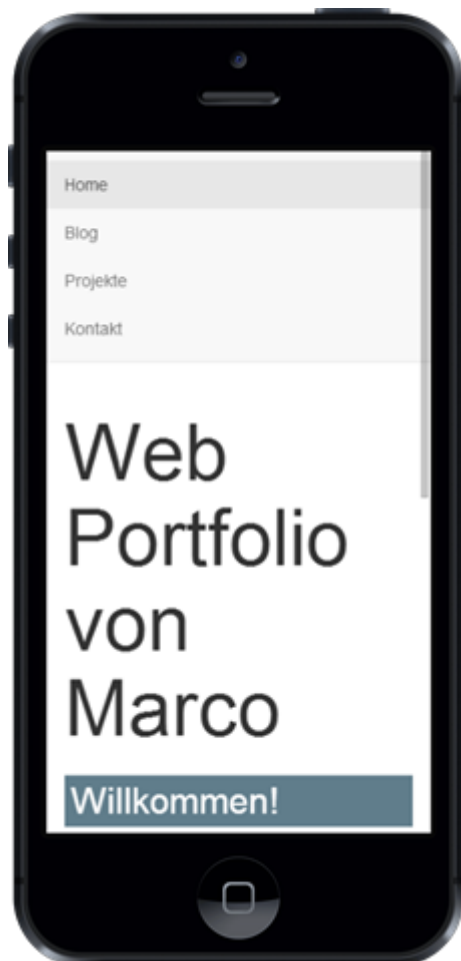
Sie werden feststellen, dass die Seite entweder abgeschnitten oder die Navigation viel zu klein dargestellt wird:



Mit einer meta-Information im head-Bereich können wir den mobilen Browsern mitteilen, dass sie die Seite vernünftig anzeigen sollen:

```
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.c
ss">
  <link rel="stylesheet" href="main.css">
  <title>Web Portfolio von Marco</title>
</head>
```

Nun sollte es etwa so aussehen:



Bootstrap wurde so entworfen, dass es auf mobilen Bildschirmen gut funktioniert. Im Screenshot oben sehen Sie, wie die Navigation automatisch untereinander angezeigt wird, sobald die Bildschirmbreite zu klein wird. Dies können Sie auch ausprobieren, indem Sie einfach das Browserfenster verkleinern.

Ein Webdesign, welches auf die Bildschirmgröße reagiert, nennt man [Responsive Webdesign](#).

Hinweis: Bootstrap klappt die Navigation auch automatisch zusammen in der mobilen Ansicht. Dafür sind die JavaScript-Dateien zuständig, welche wir jeweils vor das schliessende `</body>`-Tag kopiert haben.

Bootstrap und Navigation auf allen Seiten

Bis jetzt haben wir Bootstrap und die Navigation nur auf unserer Startseite eingefügt. Das Ziel ist natürlich, dass diese auf allen Seiten eingebunden sind.

Mit kleinen Anpassung können wir den Code aus der Startseite in alle anderen Seiten (Blog, Blogeinträge, Kontakt, etc.) kopieren:

1. Viewport und Bootstrap-CSS einfügen

Kopieren Sie aus dem head-Bereich der Startseite die Zeile mit dem viewport und die Zeile mit dem Bootstrap CSS und fügen Sie diese in alle anderen Seiten ein:

```
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.c
ss">
```

2. JavaScript einfügen

Kopieren Sie aus vom Schluss vom body-Bereich der Startseite die Links auf die JavaScript-Dateien und fügen Sie diese in alle anderen Seiten ein:

```
<body>
  <!-- Hier steht irgend ein Inhalt -->
  <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js"></script>
  <script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.3/umd/popper.min.
js"></script>
  <script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/js/bootstrap.min.js"
></script>
</body>
```

3. Navigation einfügen

Kopieren Sie den gesamten Navigationsbereich aus der Startseite und fügen Sie diese in alle anderen Seiten ein. Achten Sie darauf, dass Sie jeweils beim aktuellen Link die active-Klasse hinzufügen:

```
<nav class="navbar navbar-expand-lg navbar-light bg-light">
  <div class="container">
    <button class="navbar-toggler" type="button" data-toggle="collapse"
data-target="#navbarMenu">
      <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbarMenu">
      <ul class="navbar-nav">
        <li><a class="nav-link" href="/">Home</a></li>
        <li><a class="nav-link active" href="/blog/">Blog</a></li>
        <li><a class="nav-link" href="/projects/">Projekte</a></li>
        <li><a class="nav-link" href="/contact/">Kontakt</a></li>
      </ul>
    </div>
  </div>
</nav>
```

4. Container-div einfügen

Wenn Sie die Unterseiten im Browser öffnen, werden Sie merken, dass die Seitenränder noch nicht korrekt sind. Um das zu korrigieren, müssen wir den gesamten Inhalt (alles unterhalb der Navigation) mit einem div umgeben. Dieses div muss die Klasse container haben. Korrigieren Sie das in allen Unterseiten:

```
[... Navigation ...]  
<div class="container">  
  [... Seiteninhalt ...]  
</div>
```

5. Testen

Nun ist es wichtig, dass Sie alle Seiten aufrufen und die **Links sorgfältig testen!**

Schauen Sie als nächstes Ihren Code an und stellen Sie sicher, dass das **HTML schön aufgeräumt** ist. Immer, wenn ein HTML-Tag geöffnet wird, dann sollte die nächste Zeile eingerückt werden. Das schliessende Tag wird dann wieder nach links gerückt.

Wie weiter?

→ Erfahren Sie die nächsten Schritte.

Diese Seite ist abgeleitet von code.makery von [Marco Jakob](#), verwendet unter [CC BY](#).
Sie ist lizenziert unter [CC BY-NC-SA](#) von Daniel Fahrni daniel.fahrni@bzz.ch.

From:

<https://wiki.bzz.ch/> - **BZZ - Modulwiki**

Permanent link:

<https://wiki.bzz.ch/modul/m293/tutorials/html-css/part7/start>

Last update: **2024/03/28 14:07**

