

Einführung in Bootstrap

Bootstrap ist das beliebteste HTML, CSS und JavaScript Framework zur Entwicklung von Responsive *mobile first* Webprojekten.

Zeig mir mehr

Bootstrap

Einführung

In der folgenden Arbeit wird das Frontend-Framework [Bootstrap](#) genauer vorgestellt. Zunächst wird das Framework kurz vorgestellt. Anschließend erfolgt ein Tutorial zum Setup des Frameworks, sowie eine Vorstellung der enthaltenen Funktionen mit Anwendungsbeispielen. Schließlich folgt ein Vergleich mit anderen CSS-Frameworks.

Das Dokument wurde als HTML-Webseite erstellt, um eine direkte Vorstellung der UI-Komponenten zu ermöglichen. So können diese direkt in den Text eingebunden und der Quelltext untersucht werden, ohne auf Screenshots und ausufernde Codeblöcke zurückgreifen zu müssen. Dieses Dokument liefert also eine beispielhafte Webseite, wie sie ausschließlich mittels Bootstrap-Komponenten erstellt werden kann.

Bootstrap ist ein Framework welches mittels [HTML \(HYPERTEXT MARKUP LANGUAGE\)](#), [CSS \(CASCADING STYLE SHEETS\)](#) und JavaScript zur Entwicklung von Weboberflächen eingesetzt wird. Dabei liegt der Fokus zunächst auf der Darstellung auf mobilen Endgeräten, also einer "mobile first"-Strategie, wobei responsives Webdesign korrekte Darstellung auf allen Geräten ermöglicht. Responsive Web Design bedeutet, dass sich die Darstellung einer Webseite auf die Auflösung, sowie die Eingabemethoden des (mobilen) Endgeräts anpasst. So kann stets eine korrekte Darstellung und eine angenehme Nutzungserfahrung gewährleistet werden, egal ob mittels Desktop-PC, Tablet oder Smartphone.

Bootstrap ist seit August 2011 Open Source, der Quellcode ist auf [GitHub](#) verfügbar. Das Framework entstand ursprünglich bei Twitter unter dem Namen *Twitter Blueprint*, wo Mark Otto die diversen im Unternehmen eingesetzten Tools zur Oberflächenentwicklung vereinheitlichen wollte. Die aktuelle Version [4.5.3](#) wurde am 13. Oktober 2020 veröffentlicht. Bootstrap unterstützt alle aktuellen großen Browser, sowohl auf Desktop- als auch Mobilplattformen.

Bootstrap bietet eine große Sammlung von HTML- und CSS-basierten Designvorlagen für häufig eingesetzte UI-Komponenten wie Formulare, Buttons, Navigationsleisten oder Bilderkarussells. Diese erlauben eine einfache Erstellung responsiver Webseiten. Vorteil ist, dass diese Elemente bereits standardisiert in Bootstrap enthalten sind und mittels weiterer Parameter konfiguriert werden können, ohne CSS-Code schreiben zu müssen.

Setup

Um mit Bootstrap loszulegen, werden nur einige Imports im Webdokument benötigt. Bootstrap besteht aus CSS-Stylesheets und JavaScript-Plugins, welche inkludiert werden müssen. Zur Integration existieren verschiedene Möglichkeiten:

Manueller Download der kompilierten und minimierten CSS-Bundles und JavaScript-Plugins

Download und eigene Kompilierung der [Sass \(Syntactically Awesome Stylesheets\)](#)-, JavaScript- und Dokumentationsdateien. Dafür wird ein Sass-Compiler, Node.js, Ruby, npm sowie ein Autoprefixer benötigt. Alle weiteren Build-Werkzeuge werden mit dem Sourcecode mitgeliefert.

Installation via Paketmanager (npm, yarn, NuGet, etc.). Benötigt zudem einen Sass-Compiler und Autoprefixer.

Auslieferung einer gecacheten Bootstrap-Version mittels jsDelivr

Die Integration der kompilierten Pakete mittels *jsDelivr* stellt dabei die einfachste Möglichkeit dar.

CSS

Für CSS genügt es, den Stylesheet `<link>` im Beginn des `<head>`-Elements vor allen anderen CSS-Stylesheets zu laden:

```
<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.5.3/dist/css/bootstrap.min.css" integrity="sha384-TX8t27EcRE3e/ihU7zmQxVncDAy5uIKz4rEkgIXeMed4M0jlfIDPvg6uqKI2xXr2" crossorigin="anonymous">
```

JavaScript

Die JavaScript-Plugins werden nicht zwingend für das Bootstrap-Grundgerüst benötigt, jedoch basieren eine Vielzahl der integrierten Komponenten auf JavaScript, wie beispielsweise Dropdown-Felder, einklappbare Collapse-Bereiche, oder Toast- und Alert-Mitteilungen.

Bootstrap benötigt neben dem eigenen JavaScript-Plugins auch noch die "Tooltip & Popover Positioning Engine" Popper, sowie jQuery. Popper kann gebündelt mit dem Bootstrap-Paket, oder auch separat eingebunden werden. In jedem Fall muss das jQuery-Paket gesondert geladen werden.

Die zwei bzw. drei `<script>`-Anweisungen werden einfach am Ende des `<body>`-Tags eingefügt:

```
<!-- Option 1: jQuery and Bootstrap Bundle (includes Popper) -->
<script src="https://code.jquery.com/jquery-3.5.1.slim.min.js" integrity="sha384-DfXdz2htPH0lsSSs5nCTpuj/zy4C+OGpamoFVy38MVBnE+IbbVYUew+OrCXaRkfj" crossorigin="anonymous"></script>
<script src="https://cdn.jsdelivr.net/npm/bootstrap@4.5.3/dist/js/bootstrap.bundle.min.js" integrity="sha384-ho+j7jyWK8fNQe+A12Hb8AhRq26LrZ/JpcUGGOn+Y7RsweNrtn/tE3MoK7ZeZDyx" crossorigin="anonymous"></script>

<!-- Option 2: jQuery, Popper.js, and Bootstrap JS -->
<script src="https://code.jquery.com/jquery-3.5.1.slim.min.js" integrity="sha384-DfXdz2htPH0lsSSs5nCTpuj/zy4C+OGpamoFVy38MVBnE+IbbVYUew+OrCXaRkfj" crossorigin="anonymous"></script>
<script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.1/dist/umd/popper.min.js" integrity="sha384-9/reFTGAW83EW2RDu2S0VKAiZap3H66lZ81PoYlFhbGU+6BZp6G7niu735Sk7lN" crossorigin="anonymous"></script>
<script src="https://cdn.jsdelivr.net/npm/bootstrap@4.5.3/dist/js/bootstrap.min.js" integrity="sha384-w1Q4orYjBQndcko6MimVbzy0tgp4pwB41Z7lr30WKz0vr/aWKhXdBNmNb5D92v7s" crossorigin="anonymous"></script>
```

Eine vollständige, minimale Startvorlage folgt hinter diesem Button:

Quellcode anzeigen

Funktionen

Zunächst werden die Hauptfunktionen des Frameworks vorgestellt. Dabei wird im Kapitel Layout erst das zugrundeliegende Grid-System erklärt. Anschließend folgen Beispiele, wie die aus HTML bekannten Elemente erweitert wurden, um Typographie zu modifizieren und Bilder einzubinden. Schließlich wird eine Auswahl der im Framework mitgelieferten Komponenten gezeigt.

Layout

Container

`container` sind die Grundbausteine für Layouts in Bootstrap. In Containern können Inhalte eingesetzt, gepolstert (*padding*) und ausgerichtet werden. Dabei kann entweder die maximale Breite in Pixeln festgelegt werden, durch Verwendung von `.container-fluid` stets die maximale Bildschirmbreite verwendet werden, oder aber auch dynamische Skalierungen basierend auf *Breakpoints* verwendet werden.

[Media queries](#) sind Anfragen an das Endgerät, mit welchen beispielsweise die Bildschirmgröße, Orientierung oder Eingabemöglichkeiten (wie "hovering" mit der Maus) abgefragt werden können. Bootstrap verwendet media queries um die Bildschirmbreite abzufragen, die es dann in die Klassen von XS bis XL einteilt. Mithilfe dieser Breakpoints lässt sich die maximale Breite von Komponenten dynamisch festlegen.

Grid-System

Bootstrap untergliedert die Bildschirmfläche in ein Gitter mit bis zu 12 Spalten, welches automatisch anhand der Viewportgröße skaliert wird und auf [flexbox](#) basiert. Der Entwickler kann dabei zwischen 5 verschiedenen Spaltengrößen von "extra small" bis "extra large" wählen. Ohne zusätzliche Breitenangabe wird die Breite des Containers gleichmäßig auf die Anzahl an Spalten verteilt. Wird nur die

Breite einer Spalte festgelegt, so passen sich die verbleibenden Spalten automatisch an.

Durch das Responsive Design werden die Spalten bei einem Smartphone mit schmalen Bildschirm automatisch horizontal untereinander angezeigt.

Zeile mit gleich großen Spalten:

.col-md-6

.col-md-6

Zeile mit Spalten im Verhältnis 1:2:

.col-md-4

.col-md-8

Zeile mit Spalten im Verhältnis 1:3:

.col-md-3

.col-md-9

Quellcode anzeigen

Möchte man die Anzeige seiner Website in verschiedenen Geräten manuell steuern, kann dies über dass `class`-Attribut modifiziert werden. So kann beispielsweise spezifisch für einen mittelgroßen Tabletbildschirm eine andere Anzahl an Zeilen definiert werden:

`class="col-lg-4 col-md-6 col-xl-3"` zeigt am großen Bildschirm 4 Zeilen, bei mittleren 6 und bei sehr großen Bildschirmen 3 Zeilen. Die Anzahl an Spalten wird dann basierend auf der gesamten Zellenzahl angepasst.

Die Gitterspalten können auch ineinander verschachtelt werden. Der Parameter `offset` ermöglicht - wieder spezifisch zur Viewportgröße - einen Versatz der Spalten zu beschreiben. Mit zahlreichen weiteren Parametern lassen sich zudem auch die Ausrichtung der einzelnen Spalten im Dokument, deren Sortierung, und vieles weiteres modifizieren.

Inhalt

Typographie

Bootstrap erweitert die in HTML enthaltenen Textformatierungsfunktionen. Dieser Absatz wurde beispielsweise durch Verwendung der Klasse `class="lead"` hervorgehoben.

Mittels der `display`-Klasse kann eine größere, aber schmalere Überschrift definiert werden, dazu sei auf den [Titel](#) dieses Dokuments verwiesen. Fügt man innerhalb der `h*`-Headlines ein `<small>`-Element ein, kann man eine kleinere Unterüberschrift erstellen, wie [im nächsten Kapitel](#) zu sehen.

Absätze können zudem

links (`class="text-left"`)

mittig (`class="text-center"`),

rechts (`class="text-right"`),

oder mittels Blocksatz (benötigt etwas mehr inhaltslosen Text um die Wirkung zu sehen, am Besten einfach das Fenter verkleinern) (`class="text-justify"`)

ausgerichtet werden. Auch diese Formatierung kann für alle Viewportgrößen spezifisch definiert werden.

So ist dieser Text beispielsweise nur bei großen bis sehr großen Bildschirmen rechtsbündig.

Code

Mithilfe des `<code>`-Befehls, lassen sich Quellcode-Wörter hervorheben. Kombiniert man dies mit dem HTML-Befehl `<pre>`, welcher einen vorformatierten Textblock repräsentiert, so wird der Code statt inline als eigener Absatz ausgestellt.

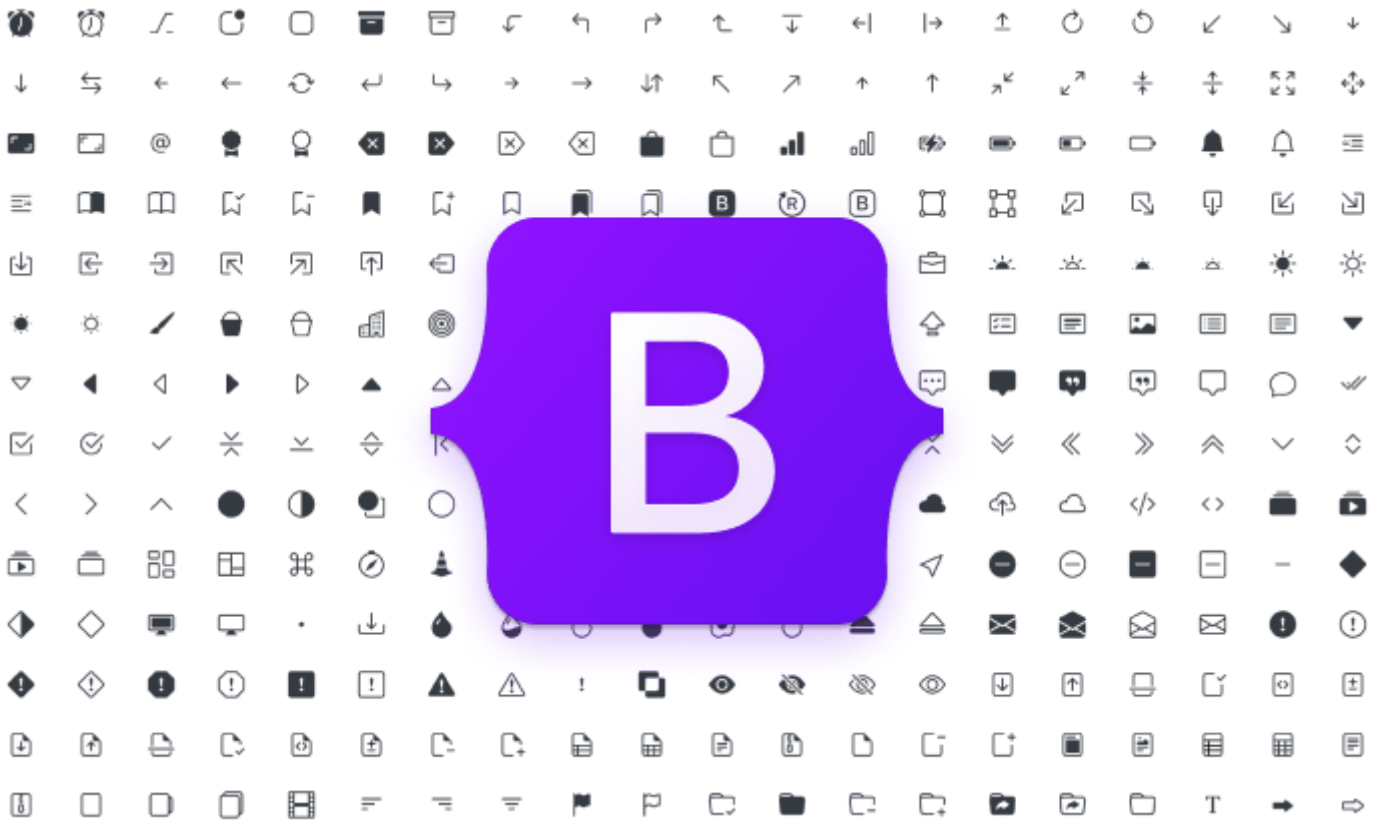
Zitate

Ein `blockquote`-Element erlaubt das Einbetten von Zitaten in den Text:

You can also give pretty look to your blockquotes — Just define the blockquotes using the standard `<blockquote>` element and bootstrap's style sheet will do the rest.
— by *TutorialRepublic*

Abbildungen

Mittels der Klasse `figure` lassen sich Bilder mit Text, üblicherweise der Bildunterschrift, gemeinsam ausrichten. Bilder werden dabei über `img` eingebunden. Durch Verwendung der Klasse `img-fluid` erhält man responsive Bilder, deren Breite sich an der Viewportgröße orientiert und das Bild automatisch korrekt skaliert. Eine Bildunterschrift wird mit `figcaption` und der Klasse `figure-caption`, optional mit Ausrichtung, angefügt.



Eine Bildunterschrift unter dem Bild, rechtsbündig.

Quellcode anzeigen

Komponenten

Im folgenden Abschnitt werden einige der im Framework enthaltenen Komponenten, welche in dieser Webseite verwendet wurden, kurz vorgestellt. Dies ist nur eine kleine Auswahl aller Komponenten, Bootstrap bietet über 20 Stück mit zahlreichen Varianten.

List Groups

List Groups führen die Darstellung von Listen in eine zeitgemäßere Optik über. So werden die Listenelemente in Boxen dargestellt, welche zudem deaktiviert, eingefärbt oder in Buttons verwandelt werden können. Dadurch wird die Funktionalität einfacher Aufzählungen deutlich erweitert. Dazu muss nur das HTML-Element für eine unnummerierte Liste `ul` um die Klasse `list-group`, und die Listenelemente `li` um die Klasse `list-group-item` erweitert werden. Auch eine horizontale Darstellung ist möglich. Das folgende Beispiel kombiniert einige der Möglichkeiten.

Springe zum Titel des Dokuments	Normales Listenelement	Eine Gefahrenmeldung
---	------------------------	----------------------

Forms

Während Formulare auch mittels HTML und CSS manuell erstellt werden können, vereinfacht das Framework dies deutlich. Ein Formular ist so nur wenige Zeilen Code entfernt, wie das folgende Login-Formular zeigt:

E-Mail-Adresse

E-Mail-Adresse eingeben

We'll never share your email with anyone else.

Passwort

Passwort

☐ Eingeloggt bleiben

Anmelden

Quellcode anzeigen

Jumbotron

Jumbotrons werden eingesetzt, um dem Nutzer sofort deutlich hervorstechende, wichtige Aussagen mitzuteilen. Diese werden besonders in der Einleitung von Webseiten benutzt, um das Interesse beim Besucher zu wecken und wichtige Marketingphrasen darzustellen. Auch in diesem Dokument ist es als "Eye-Catcher" am [Beginn der Website](#) eingesetzt.

Navbar

Die Navigationsleiste wird am oberen Bildschirmrand dargestellt. Sie dient zur schnellen Navigation zwischen den einzelnen Bereichen einer Webseite. In diesem Fall wurden nur auf die Kapitel dieses Dokuments verwiesen, üblicherweise werden dort aber alle Unterseiten einer Webseite verlinkt.

Navbar liefert einige Unterkomponenten mit, u.a. folgende:

- `.navbar-brand` um die Marke oder Firma hervorzuheben
- `.navbar-nav` zur Navigation innerhalb der Webseite, mit Unterstützung für Dropdown-Felder
- `.form-inline` für eingebettete Formulare, wie Suchfelder
- `.nav-item` für grundlegende Textbausteine im Header.

Je nach Hintergrundfarbe der Webseite, kann `.navbar-light` oder `.navbar-dark` als helles bzw. dunkles Farbschema für die Navigationsleiste genutzt werden.

Collapse

Mit der Collapse-Komponente lassen sich einklappbare Elemente definieren. Sie werden in diesem Dokument bei allen "Quellcode anzeigen" Buttons verwendet. Dabei werden die zu klappenden `<div>`-Elemente der Klasse `class="collapse"` zugewiesen. Zum Klappen muss keine JavaScript-Methode erzeugt werden, sondern man gibt einem Link oder Button die Funktion `collapse` als Parameter für `data-toggle` mit. Im `data-target` wird nun noch die ID des ein- und auszuklappenden Zielobjekts angegeben, und schon ist man fertig.

Es können auch mehrere Objekte gleichzeitig geklappt werden, indem man als Target den JQuery-Selector angibt.

Mehrere Bausteine können wie ein Akkordion nacheinander aufgeklappt werden, indem man in jedem der zu klappenden Bereiche das gemeinsame Elternattribut `data-parent` setzt.

Card

Karten sind Container, welche aufgrund der hohen Flexibilität sehr häufig eingesetzt werden. Das Designelement der Karte ist besonders von Google's Material Design bekannt.

In dieser Webseite wurden sie beispielsweise für die [Kontaktkarte](#) verwendet.

Werkzeuge

Neben den Komponenten liefert das Framework zudem einige Werkzeuge zum Individualisieren des Webseitendesigns mit.

So können beispielsweise Container mit Rändern oder Schattierungen umgeben werden, Vordefinierte Farbklassen (`primary`, `secondary`, `success`, `danger`, etc.) führen die Aufmerksamkeit des Besuchers.

Zudem Klassen zur Skalierung von Objekten, wie eine Festlegung auf halbe Breite `class="w-50"`, zu deren Positionierung (z.B. `class="fixed-top"`) oder Ausrichtung (z.B. `class="align-bottom"`).

Vergleich zu anderen Frameworks Was Bootstrap kann - und was nicht

Bei Bootstrap handelt es sich um ein reines Frontend-Framework zur schnellen Erstellung von Benutzeroberflächen mittels vorgefertigter Komponenten. Mithilfe eines intuitiven Gittersystems können responsive Webseiten erstellt werden.

Ein großer Vorteil bei der Verwendung von Bootstrap liegt in der Zeitersparnis im Vergleich zur händischen Erstellung von CSS-Stylesheets. Außerdem bietet das Framework Cross-Platform-Unterstützung, wodurch man sich keine Sorgen machen muss, ob die entwickelte Webseite auf dem Androidmobiltelefon genauso gut aussieht, wie in einer Safari-Instanz auf einem Mac-Rechner. Auch liefert es eine konsistente UI, welche *"out of the box"*, ohne Modifikation, gut aussieht. Durch die *mobile first*-Philosophie wird eine angenehme Benutzererfahrung für mobile Nutzer sichergestellt, welche mit bereits über 50% des weltweit erzeugten Traffics einen immer größer werdenden Teil aller Internetnutzer darstellen.

Die Verwendung des Frameworks zieht aber natürlich auch Nachteile mit sich.

Da das Framework sehr populär ist, sollte eine starke Personalisierung der verwendeten Komponenten erfolgen, um sich von dem *"Einheitsbrei"* anderer Webseiten abzuheben. So müssen möglicherweise sehr viele Styles überschrieben werden.

Die Bindung an jQuery zur Verwendung von JavaScript wird von einigen Webentwicklern als Negativpunkt empfunden.

Das volle Framework kann zu schwer (im Sinne der Dateigröße) und langsam sein, weil meist nur ein kleiner Prozentteil der Komponenten verwendet wird. Daher ist ein "cherry-picking" der genutzten Komponenten und die eigene Kompilierung einer modifizierten Bootstrapvariante zu empfehlen.

Bootstrap ist *kein* Full-Stack-Framework, bietet also keinerlei Werkzeuge zur Lösung von Backendproblemstellungen. Es sollte also nur mit anderen Frontend-Frameworks verglichen werden.

Django

Wie bereits erwähnt lässt sich Bootstrap nicht wirklich mit Django vergleichen, da diese deutlich unterschiedliche Einsatzzwecke bedienen.

Während das in Python geschriebene Django ein Full-Stack-Framework ist, beschäftigt sich Bootstrap mit seiner Kombination aus CSS und JavaScript ausschließlich mit dem Frontend einer Webseite, also der dem Nutzer präsentierten Oberfläche. Bootstrap kann hingegen *mit* Django verwendet werden.

Django verwendet die *Model-View-Controller*-Architektur, unterteilt alle Aufgaben einer Webseite also in drei Aufgabengebiete. Bootstrap kann in diesem Modell für den *View*-Teil komplementär eingesetzt werden. Dazu werden einfach die Bootstrap-Komponenten in die HTML-Dateien integriert. Zuvor muss dafür eine Bootstrapinstallation sichergestellt werden und die nötigen CSS und JavaScript-Dateien geladen werden, wie im Setup-Kapitel beschrieben.

Das Paket [django-bootstrap4](#) bietet zudem die Möglichkeit, mittels Python-Programmcode Bootstrap-spezifische Komponenten zu erzeugen und somit visuell ansprechende Templates zu erzeugen. Möchte man Bootstrap nur zur ansprechenderen Darstellung von Django-Formularen verwenden, bietet sich [django-crispy-forms](#) an.

Semantic UI bzw. Fomantic UI

Auch Semantic UI liefert eine breite Auswahl vordefinierter Komponenten in HTML, CSS und JavaScript. Das Framework basiert auf Prinzipien der natürlichen Sprache und soll deshalb besonders einfach von Menschen lesbar und verständlich sein.

Durch die einfachen, auf natürlicher Sprache basierenden Begrifflichkeiten, ist es besonders für Einsteiger sehr leicht zu verwenden. Objektklassen entstehen so aus einer Kombination von Adjektiven und Nomen, wie z.B. `<div class="ui vertical animated button">`. Durch die selbsterklärenden Namen ist der Designprozess auch schneller als mit den dutzenden zunächst kryptischen Klassen von Bootstrap möglich.

Ein weiterer Vorteil von Semantic UI sind die zahlreichen verfügbaren Themes, und die weitreichenden Individualisierungsmöglichkeiten zur Erstellung eigener Themes durch Anpassung von über 3.000 Variablen.

Nachteil von Semantic UI ist das weniger responsive Design, wodurch die Unterstützung für unterschiedliche Bildschirmgrößen weniger ausgereift ist, als bei Bootstrap. Bei Semantic UI sind die Breakpoints für verschiedene Viewportgrößen kein elementarer Bestandteil des Designs wie in Bootstrap.

Zuletzt sei erwähnt, dass das offizielle Semantic UI-Projekt seit Oktober 2018 "verwaist" ist und seitdem keine Updates mehr erhielt. Deshalb wurde von der Community ein Fork namens **Fomantic UI** entwickelt, indem das Framework weiterentwickelt wird. Dieser soll nach Ansicht der Entwickler in der Zukunft wieder mit dem ursprünglichen Entwicklungszweig zusammengeführt werden. Fomantic UI erweitert das ursprüngliche Framework beispielsweise um zahlreiche neue Komponenten wie einen Kalender, Platzhalter, Slider und Emojis.

Zusammenfassend ist Bootstrap das populärere Framework und es existieren deshalb mehr Ressourcen online, möchte man ein Design hingegen noch schneller erzeugen und vorallem individualisieren, sollte man auch einen Blick auf Semantic bzw. Fomantic UI werfen.

Bulma

Bulma basiert wie Bootstrap 4 auf Flexbox als flexibles Gridsystem zur Erstellung responsiver



django - The web framework for perfectionists with deadlines.



Semantic UI is a UI component framework based around useful principles from natural language.



Fomantic UI is a community fork of Semantic UI

Webseiten. Im Gegensatz zu Bootstrap verzichtet Bulma jedoch auf JavaScript, das Framework ist also ein einziges CSS-File.

Wie Bulma selbst [im Vergleich auf ihrer Webseite](#) schreibt, zeichnet sich das Framework besonders durch das einfach zu verwendende Grid-System und den verständlichen Syntax aus. Durch den Verzicht auf JavaScript ist es zudem eine sehr leichtgewichtige Lösung.

Bootstrap bietet im Vergleich dazu hingegen durch die Inklusion von jQuery Plugins mehr Interaktionsmöglichkeiten, und wie bereits erwähnt durch die hohe Popularität eine größere Community.

Während Bulma von 2016 noch sehr jung ist, und auch versucht an der "bleeding edge" der Browsertechnologie zu bleiben, trägt das 5 Jahre ältere Bootstrap möglicherweise mehr Altlasten mit sich herum, bietet dafür aber beispielsweise auch besseren Support für den Internet Explorer, was je nach zu erreichender Zielgruppe nicht zu vernachlässigen ist.

Materialize

Android-Benutzern kommt die Designsprache von Materialize CSS sicher bekannt vor, basiert sie doch, wie der Name bereits vermuten lässt, auf [Material Design](#) von Google. Wie Bootstrap greift Materialize dabei auch auf CSS und JavaScript zurück.

Das Framework ist besonders für Fans des flachen Google Designs interessant, denn Webseiten im Google-Look lassen sich damit leicht entwerfen. Möchte man besonders Nutzer von (Android-)Mobiltelefonen mit seiner Webseite erreichen, kann dieses einheitliche und wiedererkennbare Design von Vorteil sein.

Nachteile von Materialize CSS sind die fehlende Kompatibilität mit älteren Browsern (besonders Internet Explorer), sowie die vergleichsweise kleine Entwicklercommunity, und ist deshalb nur fortgeschrittenen Webentwicklern zu empfehlen.

Weitere mögliche CSS-Framework-Alternativen, aufgelistet nach Anzahl der Sterne in GitHub sind Folgende:

tailwindcss	31.7k
Foundation	28.9k
Pure.css	21.2k

Fazit

Bootstrap eignet sich sehr gut für den Einstieg in die Web-Frontend-Entwicklung, sowie für die schnelle Erstellung einfacher Webseiten. Die zahlreichen vorgefertigten Bausteine ermöglichen einen vereinfachten Entwicklungsprozess, wodurch besonders prototypische Webseiten sehr schnell umgesetzt werden können. Durch die große Popularität des Frameworks existiert zudem eine große Entwicklercommunity die beim Einstieg und weit darüber hinaus behilflich sein kann.

Wird hingegen eine sehr einzigartige Webseite benötigt, muss ein großer Aufwand in die Modifikation der einzelnen Komponenten investiert werden, um sich von anderen mittels Bootstrap entwickelten Webseiten abzusetzen. Auch der etwas kompliziertere Syntax mit den unzähligen Klassen kann für Anfänger überwältigend sein, Bulma und Fomantic UI sind im Vergleich leichter lesbar.

Wie immer gilt es also bei der Auswahl des Frameworks, das richtige Werkzeug für das gegebene Projekt zu wählen. Bootstrap ist dabei sicherlich oft eine gute Wahl.

Quellen

[Codebeispiele: Bootstrap Tutorial \(TutorialRepublic\)](#)

[Codebeispiele: Introduction Bootstrap v.4.5 \(Offizielle Dokumentation\)](#)

[What Are the Pros and Cons of Using Bootstrap in Web Development? \(BootstrapCreative\)](#)

[8 CSS-Frameworks im Vergleich \(t3n\)](#)



Bulma is a modern CSS framework based on Flexbox.



Materialize, a CSS Framework based on Material Design

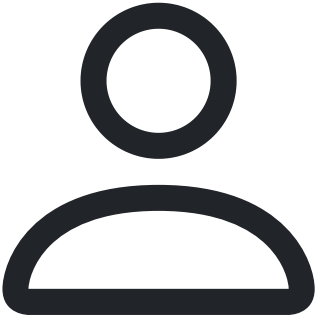
[Semantic UI or Bootstrap? Discover the pros and cons of each framework \(Developerz\)](#)

[Bulma: Das Flexbox-Framework im Vergleich zu Foundation, Bootstrap und Co. \(t3n\)](#)

[Battle of the Best Front-End Frameworks: Bootstrap vs Materialize \(Coder Academy Australia\)](#)

Über

Diese Website wurde als interaktive Ausarbeitung zur Vorlesung "Agile Webentwicklung mit Python" entwickelt.



Simon Müller
Hochschule Augsburg
Student im Master Informatik hier
an der Hochschule in Augsburg.

Kontaktieren