Anisa

Anisa\_duraj@outlook.com

CSS-Framework

Vergleiche verschiedenes Framework

Inhalt

[Was ist ein CSS-Framework? 2](#_Toc211511206)

[Bootstrap 2](#_Toc211511207)

[Materialize 3](#_Toc211511208)

[Bulma 3](#_Toc211511209)

[jQuery 4](#_Toc211511210)

# Was ist ein CSS-Framework?

Ein CSS-Framework ist dafür da, um den Code zu vereinfachen. Denn was man dort findet, sind Code-Teile, die für die Gestaltung der Webseite sehr hilfreich sind. Man sieht, wie es aussehen sollte, und hat dort auch gleich den Code, den man in seine Webseite hinzufügen kann, und man muss den ganzen Code nicht noch selber schreiben. Bei den Frameworks findet man Schaltflächen, Menüs und Eingabefelder sehr oft, und das Ganze auch in verschiedenen Variationen. Es gibt auch welche, die nicht nur CSS-Code anbieten, sondern auch welche, die etwas grösser sind und zusätzlich noch JavaScript-Code enthalten.

Jedoch gibt es nicht nur ein Framework, sondern mehrere verschiedene, die auch nicht gleich sind. In diesen Punkten können sich Frameworks unterscheiden:

* Die Grundbreiten, die für die Viewports möglich sind, und auch manchmal die Breakpoints.
* Die verschiedenen Elemente wie px (Pixel), Prozent und em (aktuelle Schriftgrösse).
* Die Anzahl der Spalten und Reihen.

Das waren ein paar wenige Beispiele. Es gibt aber noch viele weitere Punkte, bei welchen sie sich unterscheiden.

# Bootstrap

**Vorteile von Bootstrap:**

* Man kann das Layout sehr gut und einfach ändern. Es ist ebenfalls gut geeignet für verschiedene Bildschirme (Handy, Tablet oder Computer).
* Man hat 6 Breakpoints, welche **xs**, **sm**, **md**, **lg**, **xl** und **xxl** sind.  
  Das Grid hat 12 Spalten und passt sich auch automatisch an die Breakpoints an.
* Es nutzt **px** und **Prozent**, was das Anpassen an die verschiedenen Bildschirmgrössen vereinfacht.
* Durch **Media Queries** ist es einfacher, die Webseite für verschiedene Bildschirmgrössen zu gestalten.
* Es ist sehr anfängerfreundlich.
* Grosse Community.

**Nachteile:**

* Die Standard-Designs sehen oft gleich aus.
* Wenn man individuelle Designs hat, kann man diese nur beschränkt verändern.

# Materialize

Vorteile

* 12-Spalten-Grid-System
* Basierend auf Material Design (Farben, Schatten, Animationen)
* Enthält JavaScript-Komponenten (z. B. Modals, Dropdowns)
* Gute Standard-Optik
* Es hat ein sehr Modernes Design
* Viele vorprogrammierte Komponenten
* Einfach zu nutzen

Nachteile

* Nicht so flexibel mit Design
* Nicht Grosse Community.

# Bulma

Vorteile:

* Komplett auf Flexbox aufgebaut
* Responsives 12-Spalten-Grid-System
* Keine Abhängigkeit von JavaScript (Kein Java vorhanden)
* Klare, moderne Optik
* Design sehr sauber und auch minimalistisch
* Einfach zu lernen die Klassenstruktur ist sehr klar

Nachteile:

* Es hat nicht so viele Fertige Komponenten
* Man muss den Java teil selber schreiben, denn gibt es nicht bei Bulma

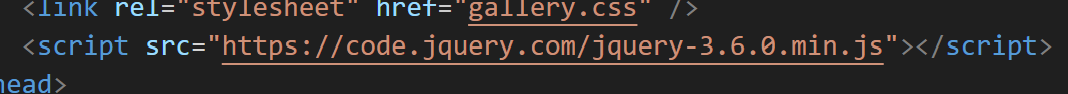
Was ist das Beste?

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Namen: | Java | Design | Schwierigkeit | Flexibilität | Layout | Summe |
| Bootstrap | 5 | 3 | 4 | 2 | 2 | 16/25 |
| Materialize | 5 | 4 | 4 | 3 | 3 | 19/25 |
| Bulma | 0 | 2 | 5 | 3 | 4 | 14/25 |

Wenn man die Liste beachtet, sieht man das Materialize sehr gut im vergleich zu den anderen ist und es sehr gut Vorteile hat.

# jQuery

Bevor ich jedoch mit dem jQuery beginnen konnte, musste ich mich ein wenig vor Bereiten das ganze habe ich so gemacht in dem ich bei codecademy die Übungen zu JavaScript gemacht habe das ganze hat eine weile gedauert da man die ganzen Grundlagen da durch gelernt hat. Man musste ebenfalls auch sehr viel lesen. Da die ganzen Funktionen so erklärt wurde aber auch musste man am Schluss das gelesene Anwenden aber nicht nur dort wo man es gelesen hat, sondern auch da bei weiteren aufgaben. Daran habe ich dann auch über eine Stunde gearbeitet. Bis ich dann zu einem Quiz kam bei welchem das gelernte überprüft wurde.

Nachdem ich damit fertig war, habe ich mich über jQuery informiert und darüber, was ich alles machen muss. Dabei bin ich darauf gestossen, dass man Folgendes in den Head eingibt: weil man das genutzt hat und auch damit gearbeitet hat. Nachdem ich das gemacht hatte, konnte ich schon anfangen, durch die Webseite zu stöbern und nachzuschauen, was ich für meine eigene Webseite gebrauchen könnte. Ausserdem habe ich herausgefunden, dass man, wenn man JavaScript-Codes verwendet, auch passende Klassen oder IDs braucht, da man diese nicht immer auf der Webseite finden kann und sie daher selbst schreiben muss. Dabei ist mir aufgefallen, dass es gar nicht so einfach ist, das Ganze zu finden, da es viele verschiedene Codeteile gibt – nicht nur ein paar, sondern sehr viele.

Im Grossen und Ganzen ist das Ganze ein sehr gutes Tool, jedoch muss man sich einarbeiten, damit man genau weiss, was wo ist. An sich ist es eine sehr schöne und gute Idee, das so zu machen, dass man einfach Code nehmen kann und dann nur noch die Klassen erstellen und einfügen muss.

|  |  |  |  |
| --- | --- | --- | --- |
| Testbereich | Gewünschtes Ergebnis | Zustand | Bemerkung |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |