

## Frameworks

Die Auswahl an Frontend-Frameworks ist groß. Beispiele sind

- Angular,
- Ember oder
- React,
- Bootstrap

die grob zusammengefasst als „Komponenten“ beschrieben werden können.

- Polymer ist eine Alternative dazu und hat sich weniger den „Komponenten“ verschrieben als viel mehr den „Web Components“ bzw. den „Standard-APIs“. Damit hat man produktiv einsetzbare Komponenten zur Verfügung.

## Frameworks für responsives Design

Ein Framework ist ein Satz vordefinierter CSS-Regeln, die Sie für die schnelle Entwicklung verwenden können; normalerweise beinhalten sie typografische Elemente, Formulare und recht häufig auch Layout-Komponenten. Eines der besseren Frameworks, das in vielen bekannten Websites verwendet wird, ist *Blueprint.css*. Das bekannteste aktuelle Framework ist jedoch *Bootstrap* von Twitter. Es bietet viele vorformatierte Layout-, Typografie- und Formularoptionen, zahlreiche wiederverwendbare Komponenten und auch eine Erweiterbarkeit durch JavaScript.

### Vorteile von Frameworks:

Bei so gut wie allen Projekten muss man auch Aufgaben lösen, die an sich nichts Neues sind ... und bei denen die Wahrscheinlichkeit groß ist, dass sie bereits jemand anderes gelöst hat. Wenn man als Einzelkämpfer unterwegs ist, kann man sich nach und nach eine eigene Bibliothek mit Funktionalitäten aufbauen, die man bereits in Projekten gebraucht hat und die sich wiederverwenden lassen. Oder aber man kann zu einem Framework greifen und den Code von anderen weiterverwenden, der oft besser durchdacht und gründlicher getestet ist.

Frameworks bieten aber weitere Vorteile:

- Statt viele Zeilen Code selbst zu schreiben, kann man auf vordefinierte Funktionen/Bibliotheken zurückgreifen. Damit wird die Entwicklung einfacher und schneller. Es gibt üblicherweise Tools für die Formularüberprüfung, für den Umgang mit Sessions/Cookies, zum Filtern von Input, zur Authentifizierung, für Tests und vieles mehr.
- **Sicherheit** ist heute ein wichtiges Thema. Frameworks bieten oft bewährte Strategien, um die Sicherheit der Anwendung zu gewährleisten. So werden z.B. bei der Ausgabe von Inhalten üblicherweise HTML-spezifische Sonderzeichen automatisch maskiert oder es gibt automatisch ein zusätzliches verstecktes Feld in Formularen, um CSRF (Cross-Site-Request-Forgery)<sup>1</sup> zu verhindern.
- **Organisation:** Frameworks geben einen Weg vor, wie der Code zu organisieren und zu schreiben ist. Das reicht von einfachen Vorgaben für Speicherorte von Dateien über Konventionen der Benennung bis hin zur Aufteilung von Aufgaben. Damit muss man im Team diese Vereinbarungen nicht mehr treffen und könnte beispielsweise auch leichter während

des Projekts zusätzliche Entwickler hinzunehmen. Ebenso ist es einfacher, bei Problemen Unterstützung von der Community des Frameworks zu erhalten.

- **Weniger ist mehr:** Die Codemenge, die man selbst schreiben muss, reduziert sich beim Einsatz eines Frameworks; das bedeutet, es geht schneller, und außerdem minimiert sich die Fehlerquote.

Frameworks haben natürlich auch Nachteile:

- Jedes Framework braucht **Zeit zur Einarbeitung** und so lohnt sich der Einsatz eines Frameworks als Einzelkämpfer eher, wenn Sie mehrere Projekte damit realisieren. Dieser Nachteil gilt nicht für die Entwicklung im Team, wo man ohne den Einsatz eines Frameworks eine genaue Dokumentation mit Vorgaben für die Code-Organisation erstellen müsste.
- Vom Framework vorgesehene Aufgaben realisiert man im Allgemeinen wesentlich schneller. Wenn man aber etwas anderes umsetzen möchte, kann der **Aufwand** größer sein.

Frameworks stellen Bausteine für typische Layoutsituationen und User-Interface-Elemente bereit. Außerdem **umgehen sie bekannte CSS-Bugs**, insbesondere für ältere Browser.

Wenn man das System einmal verinnerlicht hat, muss man sich keine Gedanken mehr über CSS-Code machen; man baut einfach das HTML-Gerüst der Seite auf und fügt Klassen mit den entsprechenden Elementen hinzu, den Rest erledigt das Framework.

Die meisten Frameworks lassen sich in zwei Klassen einteilen:

1. Layout-Frameworks
2. UI-(User-Interface)Frameworks

UI-Frameworks (wie Bootstrap) gehen über das reine Layout weit hinaus. Neben dem Raster bieten sie Bibliotheken von vorgefertigten User-Interface-Komponenten, die nicht nur per CSS gestylt, sondern über JavaScript auch funktional ausgestattet sind. Diese Vielfalt hat aber ihren Preis – die Frameworks sind viel komplexer, greifen zum Teil auf weitere Bibliotheken und Tools wie jQuery oder CSS-Präprozessoren zurück und erfordern einige Einarbeitungszeit.

## **Einige responsive Frameworks**

Übersicht auf <http://usablica.github.io/front-end-frameworks/compare.html>

Bootstrap 3 und Foundation 5 werden hier gegenübergestellt: <http://designmodo.com/bootstrap-vs-foundation/>

- YAML (yet another multicolumn layout)  
äußerst flexibles Gerüst, das gut geplant und durchdacht ist; ist responsive und für mobilen Einsatz ausgelegt  
Layout-Framework für die Erstellung von Websites spezialisiert  
[www.yaml.de](http://www.yaml.de)
- Foundation  
ein sehr umfangreiches Applikations-Framework mit vielen Funktionalitäten

ebenfalls mit responsiven und „Mobile-First-Ansatz“

[www.foundation.zurb.com](http://www.foundation.zurb.com)

- Bootstrap

Bootstrap berücksichtigt beim Aufbau des Rasters bis zu vier Ansichten: sehr kleine Bildschirme (Smartphones) „col-xs“; kleine Bildschirme (Tablets) „col-sm“; mittlere Bildschirme (Desktops < 1.200 px) „col-md“; große Bildschirme „col-lg“.

[www.getbootstrap.com](http://www.getbootstrap.com)

- pureCSS

[www.purecss.io](http://www.purecss.io)

## **JavaScript-Bibliotheken und andere Helfer**

Viele der erwähnten Frameworks nutzen weitere Bibliotheken zur Unterstützung, wie z.B. jQuery.

jQuery hat sich zu DER JavaScript-Standardbibliothek entwickelt. Es stellt einfache Funktionen bereit, mit denen sich DOM-Elemente ansprechen und verändern lassen. Es verfügt über mehrere Zusatzpakete wie jQuery UI (User Interface Elemente) und jQuery Mobile (JavaScript-Funktionen für mobile Websites).

Man kann es über ein CDN (Content Delivery Network) auf der Website einbinden und muss es somit nicht herunterladen und dann auf der Website anbieten.

[www.jquery.com/download](http://www.jquery.com/download)

### Quellen:

Andrea Ertel in Responsive Webdesign, Anpassungsfähige Websites programmieren und gestalten, Bonn 2015, Galileo Press, S. 217-245

Florence Maurice in PHP 5.6 und MySQL 5.7, dpunkt, 2015, S. 447-448

Responsive Webdesign, Bonn 2015, Galileo Press, S.217-245

Martin Splitt, Web Components mit Polymer, Heidelberg, 2017, dpunkt Verlag