

# Workshop Javascript frameworks

---

## frameworks

JavaScript is de enige programmeertaal die wordt begrepen door je browser, en speelt daarom een cruciale rol in het tot leven brengen van het web. Hierdoor is JavaScript sterk in opkomst: er wordt in vergelijking tot andere talen steeds meer JavaScript geschreven. Er zijn een aantal redenen waarom er frameworks zijn en waarom ze nuttig zijn bij het ontwikkelen.

### **Nut van JavaScript frameworks**

#### **Voordelen**

JavaScript = populair. Volgens de statistieken van Github zijn JavaScript-repositories het meeste populair en actief op dit moment. JavaScript is tot op heden de enige client-scripting taal die je in alle browsers kunt gebruiken.

Naast dat dergelijke frameworks veel werk uit handen nemen, vinden veel developers het ook leuk om ze zelf te maken. Is de doorsnee developer misschien ook wat bovengemiddeld eigenwijs? Misschien denken ze het zelf beter / makkelijker / sneller te kunnen? Een goede reden kan zijn dat geen van alle frameworks precies doet wat jij wil en ze soms te ingewikkeld zijn waarvoor jij ze wil gebruiken en dat je daarom ervoor kiest om een framework zelf te maken.

Vergeleken met veelgebruikte programmeertalen (zoals Java en C++), is JavaScript een relatief eenvoudige taal. Of ja eenvoudig; het is vooral zeer laagdrempelig. Je kunt snel van start. Ook als minder ervaren programmeur.

Naast dat je niet veel andere keuze hebt als het gaat om client-side scripting in een browser, maakt deze eenvoud het ook nog eens aantrekkelijk om te beginnen aan de bouw van je eigen JavaScript framework.

#### **Nadelen**

Maken je applicatie soms onnodig ingewikkeld.

De beschikbare frameworks doen niet wat je wilt of zijn té complex.

### **Verschillen tussen framework en libraries**

Verschil is miniem. Je zou kunnen zeggen dat een framework een verzameling libraries is. Het is robuust. Veel verschillende bronnen op internet spreken elkaar tegen dus wat een framework lijkt te zijn, noemen zich een library.

#### **Een framework is een skelet wat een heleboel**

A framework is a skeleton where the application defines the "meat" of the operation by filling out the skeleton. The skeleton still has code to link up the parts but the most important work is done by the application.

Examples of libraries: Network protocols, compression, image manipulation, string utilities, regular expression evaluation, math. Operations are self-contained.

Examples of frameworks: Web application system, Plug-in manager, GUI system. The framework defines the concept but the application defines the fundamental functionality that end-users care about.

## **AngularJS 2**

Angular 2 is de opvolger van AngularJS 1.x en biedt tal van nieuwe mogelijkheden, betere performance, maar ook een compleet nieuwe syntaxis. De afgelopen jaren heeft het JavaScript-framework AngularJS de webdevelopment-wereld veroverd en is het uitgegroeid tot een van de populairste frameworks om webapplicaties te schrijven. Grote bedrijven als banken, verzekeraars, overheden en andere organisaties gebruiken inmiddels AngularJS in hun websites. In Angular2 wordt echter gebroken met het verleden en projecten zullen niet zomaar te migreren zijn. Er zijn nieuwe concepten (gebaseerd op ECMAScript 2015, de nieuwe versie van JavaScript), nieuwe annotaties en nieuwe manieren om componenten te ontwikkelen. Grote bedrijven zoals Domino's pizza, Ryanair, Paypal checkout en Google maken er gebruik van. Het is een open source framework gesupport door Google. Het beschrijft zichzelf als extensie op HTML om complexe web app te kunnen bouwen.

Angular is een MVC type framework. Het biedt two-way data binding tussen models en views. Dit betekent dat automatische update aan beide kanten mogelijk is.

### **Voordelen**

### **Nadelen**

feiten, hoe oud, wie gemaakt.