





The JavaScript is a fast changing environment and because of that I've tried to stay away from choosing a front-end framework. It's a difficult choice to make, but recently I had to make it.

While building a product using only jQuery and Handlebars, I noticed that I was missing structure in my JavaScript code. To make sure that I build an understandable and maintainable codebase, I needed to pick a side for my frontend battles.

Please note the content of this talk is my opinion. You're welcome to disagree.



Techno VANILLAJS	Time Passe 1995 31	ngers* Stars*	Status ALWAYS ON TIME	
MOOTOOLS	2006 04	% Ø2K	CANCELLED	
JQUERY	2006 65	2 35K	STILL ON TIME	
	2003 04	2 426	UHO KNOWS?	
COFFEESCRIPT	2009 =	C 12K	SLOWING DOWN	
BACKBONE.JS	2010 .1	% 23K	ON TIME	is the real
BOOTSTRAP	2011 09	% 86K	ON TIME - POPULAR	question!
EMBER.JS	2011 =	IC 14K	NEW TRAIN COMING	
METEOR	2012 =	IC 28K	BOARDING	
REACT	2013 =	C 28K	CRAZY BOARDING	
*PASSENGERS BA	SED ON W3TE	CHS SURVEYS	*STARS ON GITHUB	

But how do I choose... JavaScript frameworks come and go at a fast pace.



This article describes exactly how difficult the JavaScript environment is to follow up.

How it feels to learn JavaScript in 2016

https://hackernoon.com/how-it-feels-to-learn-javascript-in-2016-d3a717dd577f#.5qtvzhvy7



But what's the cause of this high speed innovation in JavaScript frameworks and tools? The browser is an exceptional environment to develop for. With multiple browsers around, all with their own JavaScript implementation, we've always encountered limitations. And while almost all other environments only change slowly, for example: a new version of iOS or Android is only released once a year, the browser does constantly.



So let's take a look what our options are, because I can't cover all frameworks in one session. From this list we'll have a look at: Angular 2, Ember and React.

These are the onces that didn't make the cut:

- Sproutcore
- Knockout
- Dojo
- Vue
- Backbone.js
- Polymer
- Aurelia



- Angular 2 is more of a platform then a framework with all the tools in the ecosystem
- Version 2 has just been released and is a complete rewrite. Migrations from AngularJS are possible but won't be easy. The release path during the release candidates has been a very bumpy road, with multiple RC's breaking backwards compatibility. Not something I expect from an OSS project.
- Angular 2 uses a component based architecture. It leafs a lot of freedom to the develop on how to organise and structure code. But this also means that a well thought architecture is required for large codebases.
- Angular 2 allows you to develop in JavaScript, TypeScript or Dart. It looks like most people are using TypeScript
- Angular 2 is a project backed and drive by Google



Much cleaner and easier templating in Angular 2 then in AngularJS (Angular 1).



An example of a component for Angular 2 written in TypeScript



- Angular 2 is more of a platform then a framework with all the tools in the ecosystem
- Version 2 has just been released and is a complete rewrite. Migrations from AngularJS are possible but won't be easy. The release path during the release candidates has been a very bumpy road, with multiple RC's breaking backwards compatibility. Not something I expect from an OSS project.
- Angular 2 uses a component based architecture. It leafs a lot of freedom to the develop on how to organise and structure code. But this also means that a well thought architecture is required for large codebases.
- Angular 2 allows you to develop in JavaScript, TypeScript or Dart. It looks like most people are using TypeScript
- Angular 2 is a project backed and drive by Google



- Ember is a framework/platform that was forked from Sproutcore and is closely related to the Ruby on Rails community. That explains the heavy use of convention over configuration.
- It's a very opinionated. It's the Ember way or the highway. This means there is not a lot of room for a lot of flexibility.
- For templating, Ember uses Handlebars
- Ember has a really clear and clean release path
- It is a project backed by the community instead of a major tech company (like Angular 2 and React)



"It's the Ember way or the highway"

Yehuda Katz is one of the creators of Ember



- Ember is a framework/platform that was forked from Sproutcore and is closely related to the Ruby on Rails community. That explains the heavy use of convention over configuration.
- It's a very opinionated. It's the Ember way or the highway.
- For templating, Ember uses Handlebars
- Ember has a really clear and clean release path
- It is a project backed by the community instead of a major tech company (like Angular 2 and React)

	Č.
	EMBER
Manual Prime Manual Prim Manual Prime Manual Prime </th <th></th>	
Aug Sep Oct Nov Dec Jan Feb Mar Apr May 2015	

Ember uses three branches for it's releases: Canary, Beta and Release.

Features are developed on the Canary branch with an on/off switch. At the start of a development cycle, the features are choosen for that beta version and they are switched on in the beta branch. Several beta version are released, but when an issue is raised with a feature and it can't be solved by the end of the development cycle, it's pulled out of the beta.

After a development cycle, a new minor version is released. The duration of a development cycle is 6 weeks.

Recently Ember has added a long-term support version (LTS). The latest LTS release is 2.8.3, made on November 1st, 2016

More infor on: http://emberjs.com/builds/



- Ember is a framework/platform that was forked from Sproutcore and is closely related to the Ruby on Rails community. That explains the heavy use of convention over configuration.
- It's a very opinionated. It's the Ember way or the highway.
- For templating, Ember uses Handlebars
- Ember has a really clear and clean release path
- It is a project backed by the community instead of a major tech company (like Angular 2 and React)



- In contrast of Angular 2 and Ember, React offers only the View layer of MVC
- It's combined a lot with Flux or Backbone to create a full MVC architecture
- The main focus for React is DOM manipulation and that's where it's very good at. It uses the concept of Virtual DOM to do this very fast.
- Like Angular 2 it uses a components based architecture.
- The project is created by and backed by Facebook



With React templating happens inside of the components.



- In contrast of Angular 2 and Ember, React offers only the View layer of MVC
- It's combined a lot with Flux or Backbone to create a full MVC architecture
- The main focus for React is DOM manipulation and that's where it's very good at. It uses the concept of Virtual DOM to do this very fast.
- Like Angular 2 it uses a components based architecture.
- The project is created by and backed by Facebook



So, how do we choose? Different situations exist in which you'll have to choose.

- For you project: you might have to choose a framework to use in a new/existing project, from a more business point of view. Then the following conditions are important:
 - Does the framework fits your project
 - In case of a existing project, allows the framework for a gentle migration or not
 - Which frameworks are popular? Can we easily recruit new resources?
- For yourself: you might want to choose a framework for yourself to invest into. Then the following conditions are important:
 - Does the framework fits your style of programming?
 - Are there a lot of opportunities that require the knowledge of the frameworks



Although I currently work on an Angular 2 project (the decision was made before I arrived). My preference goes to Ember, mainly because of it's style. I like working with convention over configuration and I'm a fan of a nice structure. I was shocked by the release path of Angular 2 and seeing that with Ember it's well organised gives me confidence.



Pedrillo is Saas solution I'm building for music orchestra's. Currently it's not a Single Page Application, but that's something I want to change and I'll start working on transition to Ember very soon.



Feel free to ask me questions! Now, at the bar or via email! ;)