



Comparing Hot JavaScript Frameworks

Matt Raible • <http://raibledesigns.com>

Photos by  McGinity Photo

 RAIBLE DESIGNS
ENTERPRISE OPEN SOURCE CONSULTING



Who is **Matt Raible**?

Montanan, Father, Husband, Skier,
Mountain Biker, Whitewater Rafter

Open Source Connoisseur

Web Developer and UI Architect

Blogger on raibledesigns.com



Bus Lover

What about You?

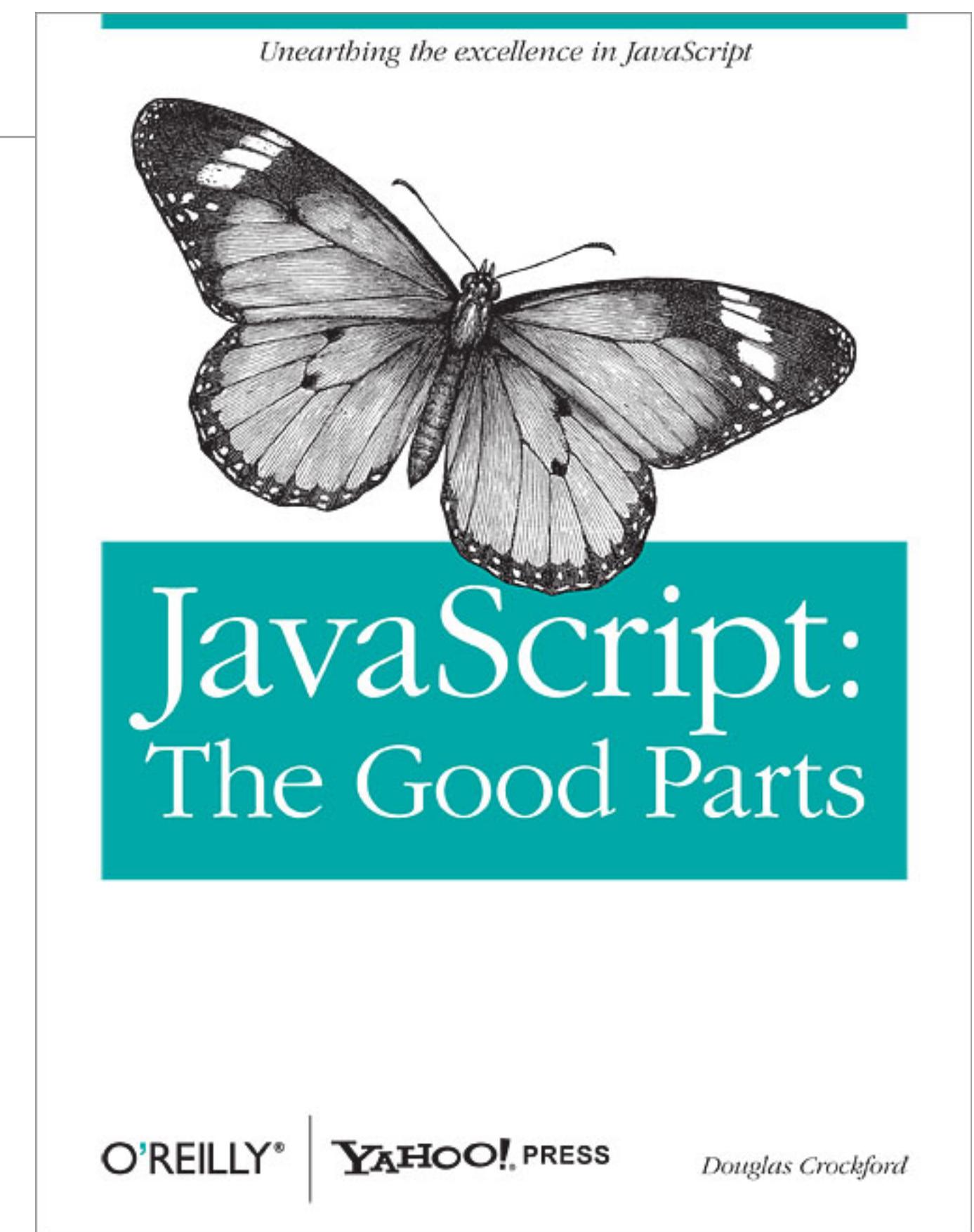
How long have you been programming in
JavaScript?

Do you remember IE6?

Are you a Java Developer?

Do you run and test your apps in Chrome?

What JavaScript Frameworks do you use?

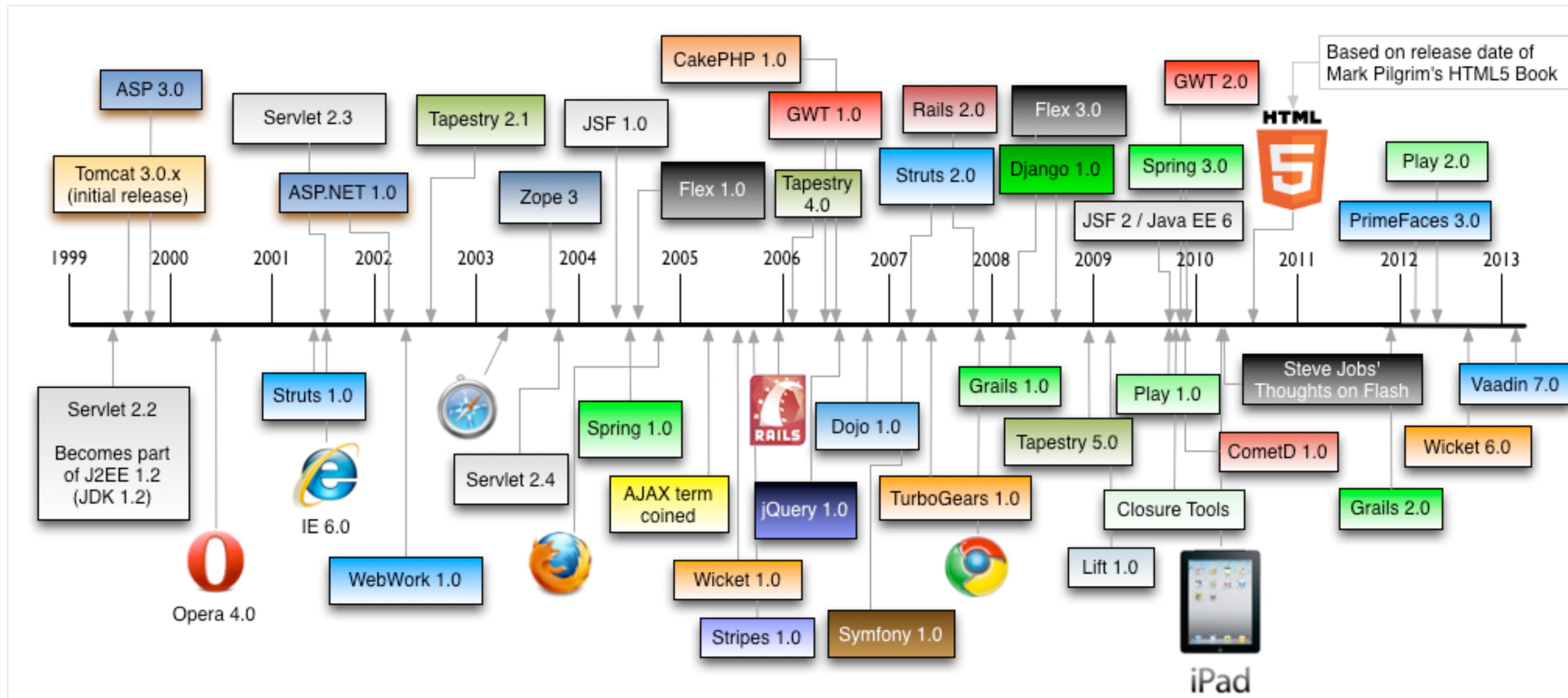


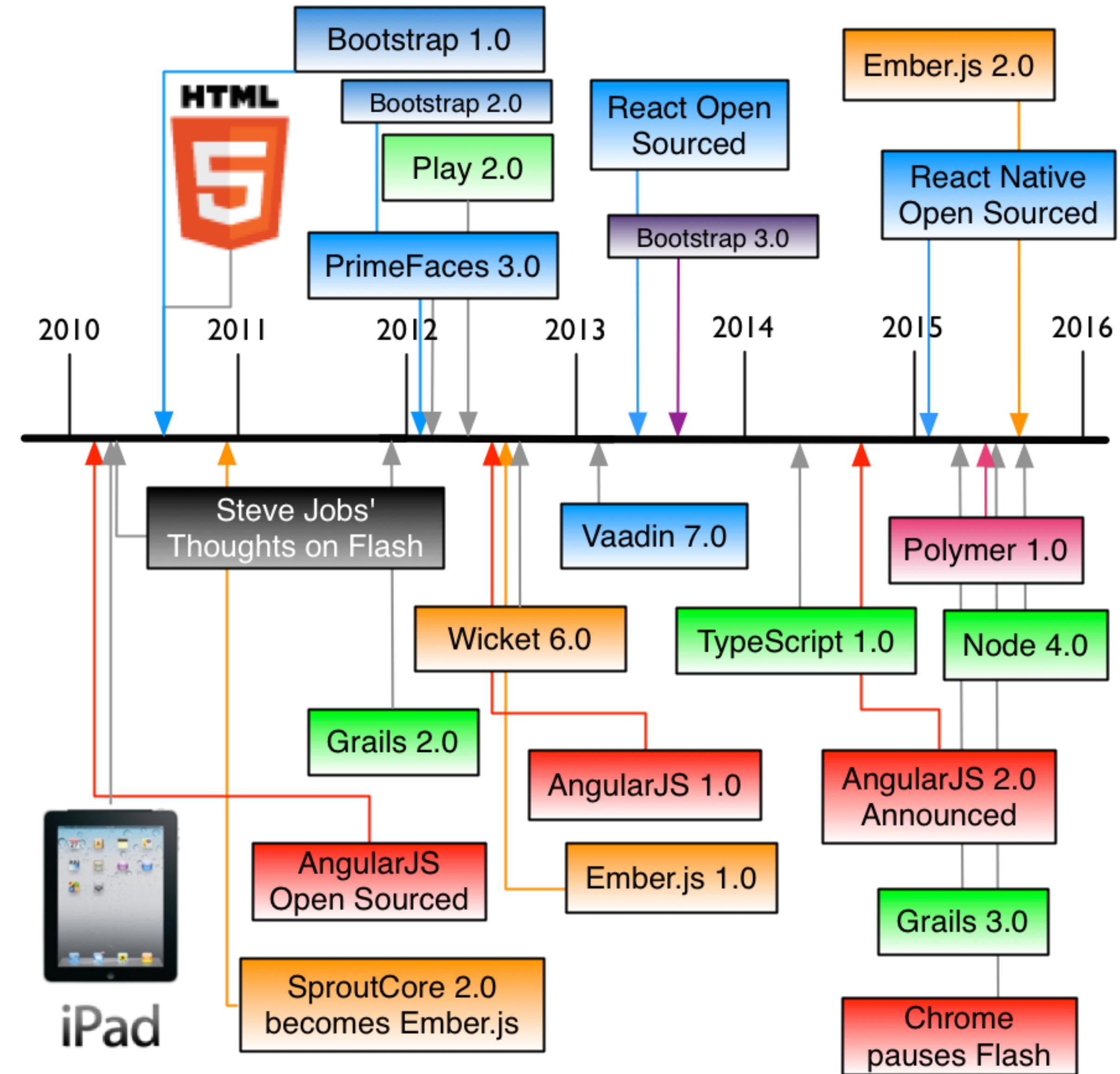
So where's the Spring MVC in all of this?

What's the best framework
in JavaScript Land?

Let's look at the last time I talked about
web frameworks... Feb 2014

Early 2013





Too many web frameworks?



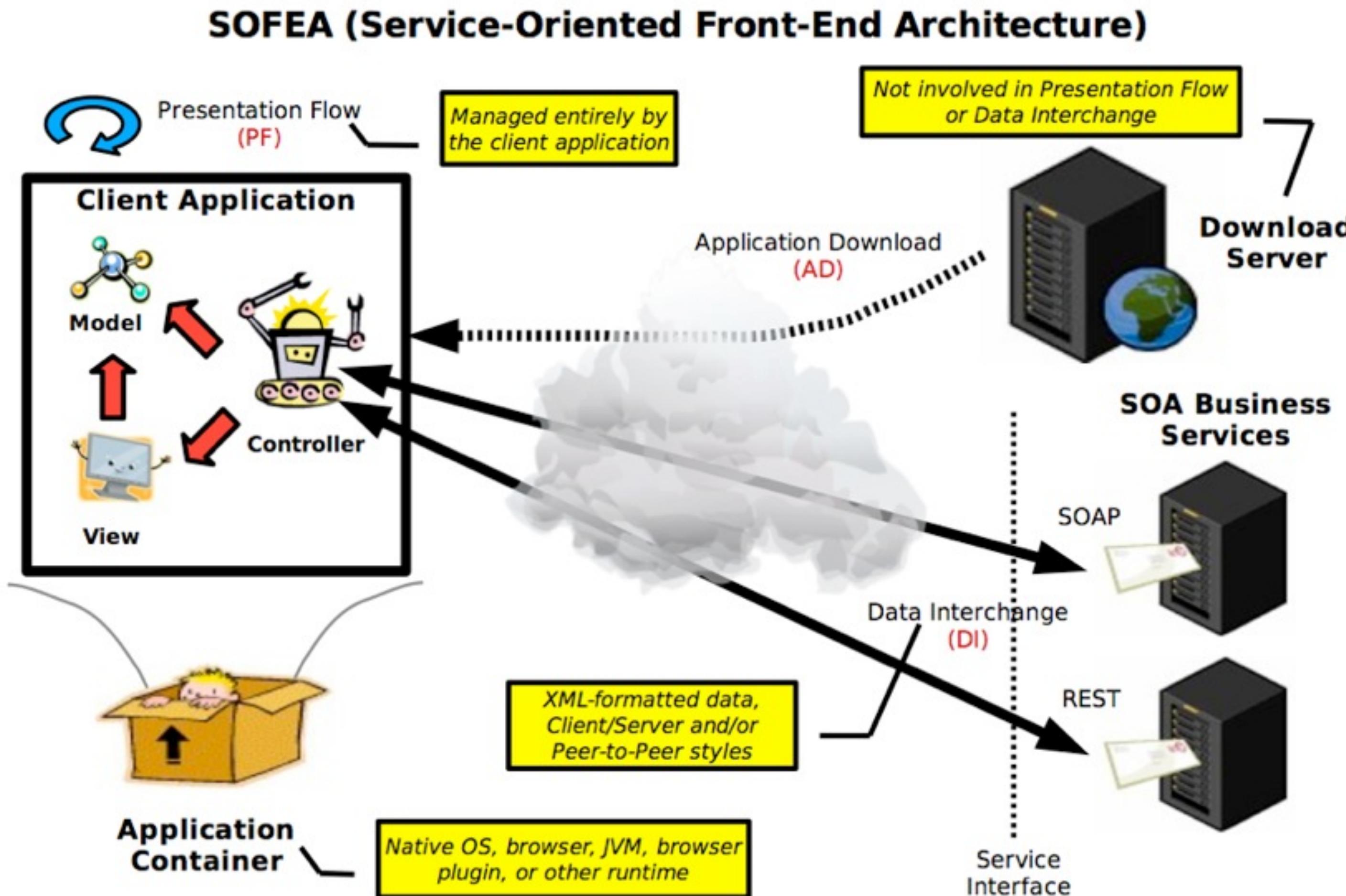
tapestry



Struts²



Remember SOFEA?



SOFEA became a Reality



BACKBONE.JS

JavaScript MVC



ANGULARJS
by Google

SPROUTCORE

HTML



Cappuccino



Sencha Ext JS



PhoneGap

SOFEA Performance Issues



 Engineering

TUESDAY, MAY 29, 2012

Improving performance

To connect you to information we've been reviewing

Our First No



Client-side MVC is not a silver bullet

February 26th, 2013

This is an edited repost of [a comment of mine](#) on Amy's blog post about [why we shut down Charm](#).

Charm, as it is, is using Backbone.js, Underscore.js and Zepto on the front-end, and Rails 2.3, Postgres, memcached, redis, resque, and for websockets Sinatra, and a few other things. The front-end is communicating with the back-end via a JSON API.

I've come to the realization that this much **client-side processing and decoupling is detrimental to both the speed of development, and application performance** (a ton of JavaScript has to be loaded and evaluated each time you fire up the app). It's better to let the server handle HTML rendering and minimize the use of JavaScript on the client.

traditional web frameworks are *still* relevant

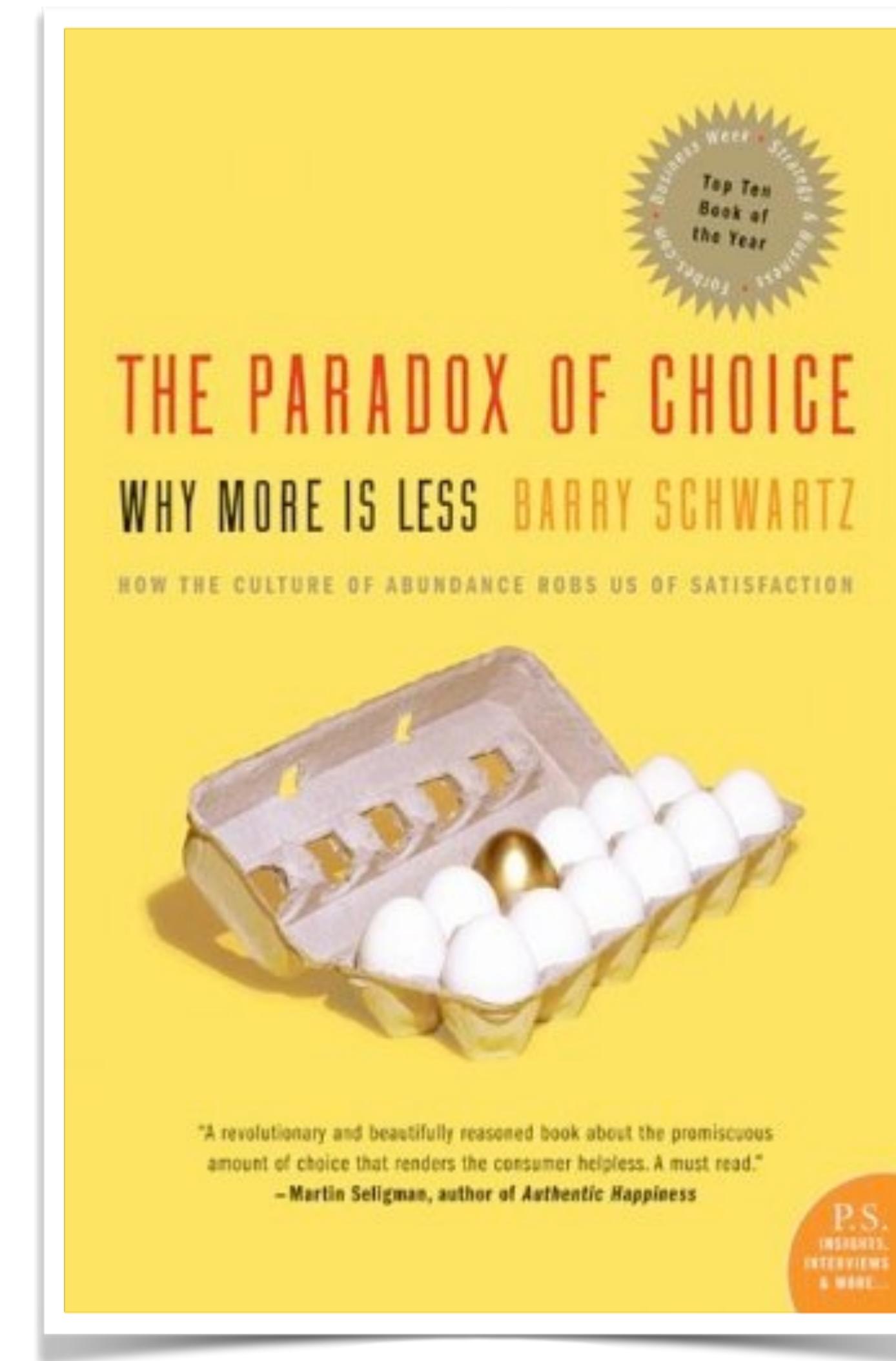
traditional web frameworks are *still* relevant

server-side rendering is still relevant

server-side rendering is still relevant

server-side rendering is still relevant

The Paradox of Choice



Good Decisions Involve

1. Figure out your goal or goals
2. Evaluate the importance of each goal
3. Array the options
4. Evaluate how likely each of the options is to meet your goals
5. Pick the winning option
6. Modify goals

Maximizer vs. Satisficer

- ▶ maximizer - tries to make **best possible** choice
- ▶ satisficer - tries to find **first suitable** choice



(Photo: [Tori Cat](#) at Flickr)

There's only three choices here!

The Real Problem



How to Constrain Choices

- ▶ **2006:** Choose based on the type of app you're developing
- ▶ **2010:** Choose based on 20 different criteria (the Matrix)
- ▶ **2013:** Narrowed it to 6
 - Community / Support
 - HTML5
 - REST
 - Mobile
 - Performance
 - Page Speed



2015: Why would you limit your choices?

Learn as much as you can, everyday.

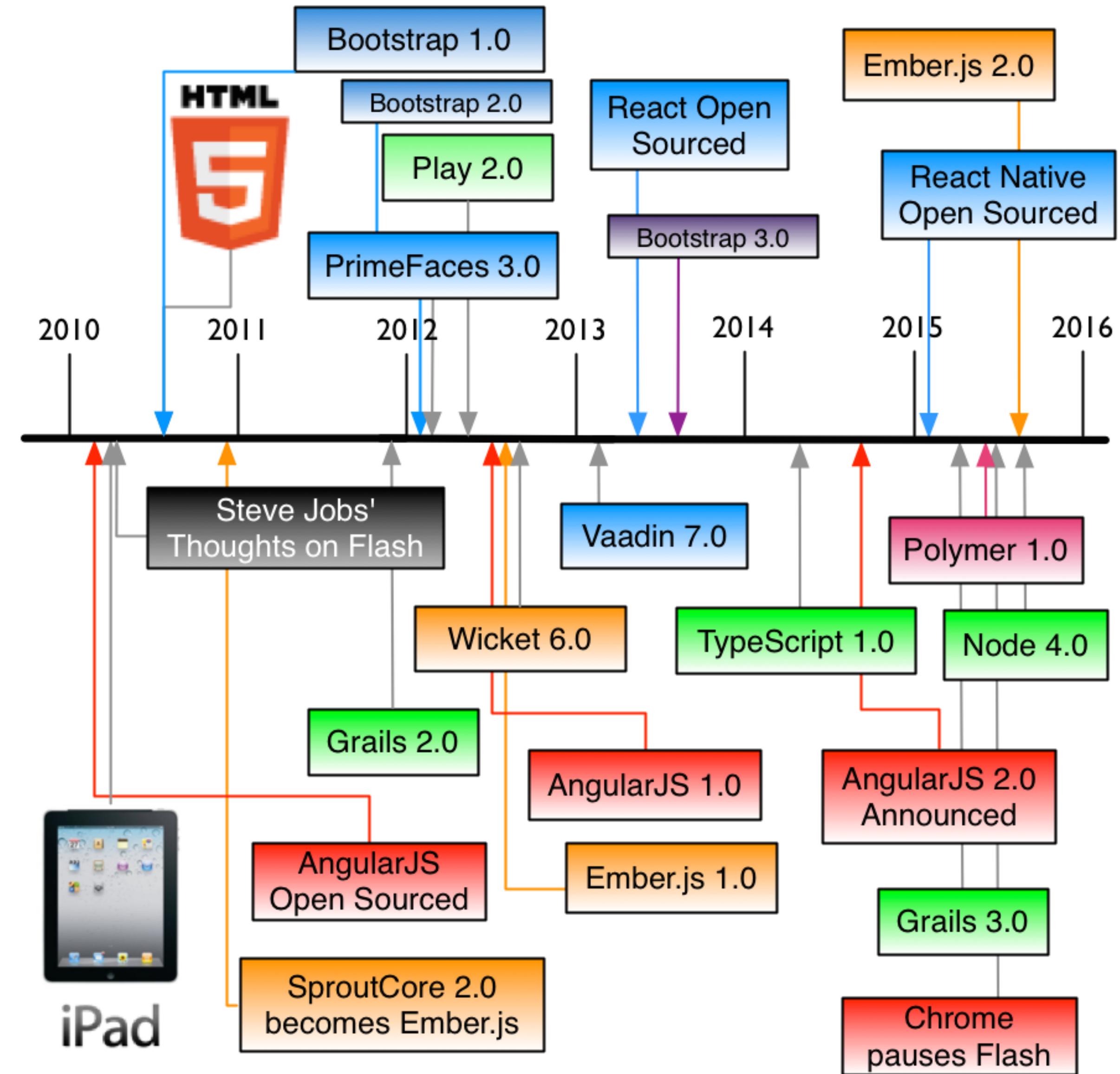
Why should you care?

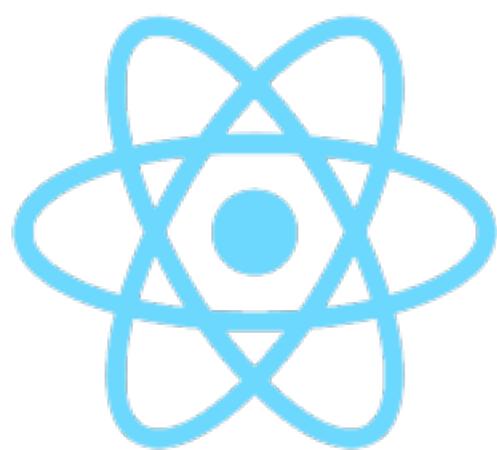
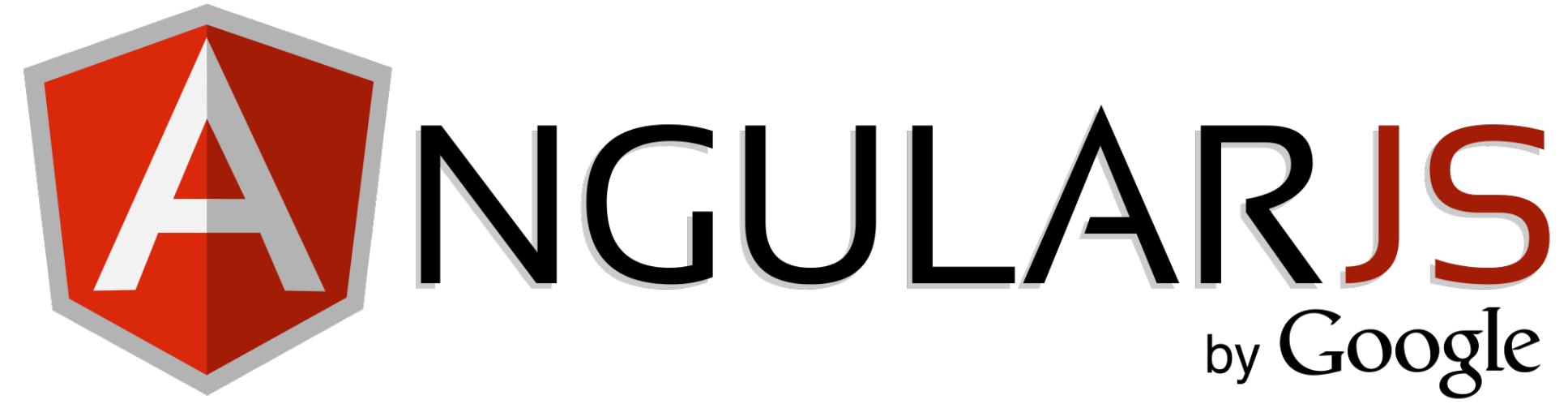
We Tested How Googlebot Crawls Javascript And Here's What We Learned

Think Google can't handle JavaScript? Think again. Contributor Adam Audette shares the results of a series of tests conducted by his colleagues at Merkle | RKG to examine how different JavaScript functions would be crawled and indexed by Google.

Adam Audette on May 8, 2015 at 12:32 pm

<http://searchengineland.com/tested-googlebot-crawls-javascript-heres-learned-220157>





React

“If I have seen further than others,
it is by standing upon the
shoulders of giants.”

Sir Isaac Newton



Yevgeniy Brikman

Programmer, entrepreneur, writer, speaker, traveler

twitter.com/brikis98

www.hello-startup.net

[Node vs. Play Framework](#)

The Framework Scorecard

	Angular	Ember	React
1. Learn			
2. Develop			
3. Test			
4. Secure			
5. Build			
6. Deploy			
7. Debug			
8. Scale			
9. Maintain			
10. Share			

For each feature, we'll discuss...

- 1** Much worse than most frameworks
- 5** About the same as most frameworks
- 10** Much better than most frameworks

The History of AngularJS

Started by Miško Hevery in 2009

GWT = 3 developers, 6 months

AngularJS = 1 developer, 3 weeks

Learn more:

 <https://www.youtube.com/watch?v=X0VsStcCCM8>



GetAngular Matt

web.archive.org/web/20091002201638/http://www.getangular.com/

Beta Home Get Started Demo Documentation Blog Pricing Support My Account

Get Angular

Introducing `<angular>`. Simple apps, Made simple.

Mock up your application in your favorite HTML editor; Bring it to life with `<angular>` in minutes.

[Sign up](#) [View demos](#)

 **Managed Database**
`<angular>` manages data for you

 **Embeddable**
In third-party sites such ad blogs and wikis

 **HTML & CSS**
Full control of the look and feel

 **Declarative**
No need to learn a new language

 **Security**
Authentication and permissions

 **Rich Widgets**
Date-picker, Barcodes, Charts, etc.

Watch how easy it is!



Design web application in 3 easy steps.

Introducing the `<angular>` way to build web applications:

1. Create HTML/CSS templates with your favourite HTML editor
2. Copy & Paste a single line of JavaScript into your HTML template
3. Annotate the HTML/CSS template with `<angular>` conventions

The History of Ember.js

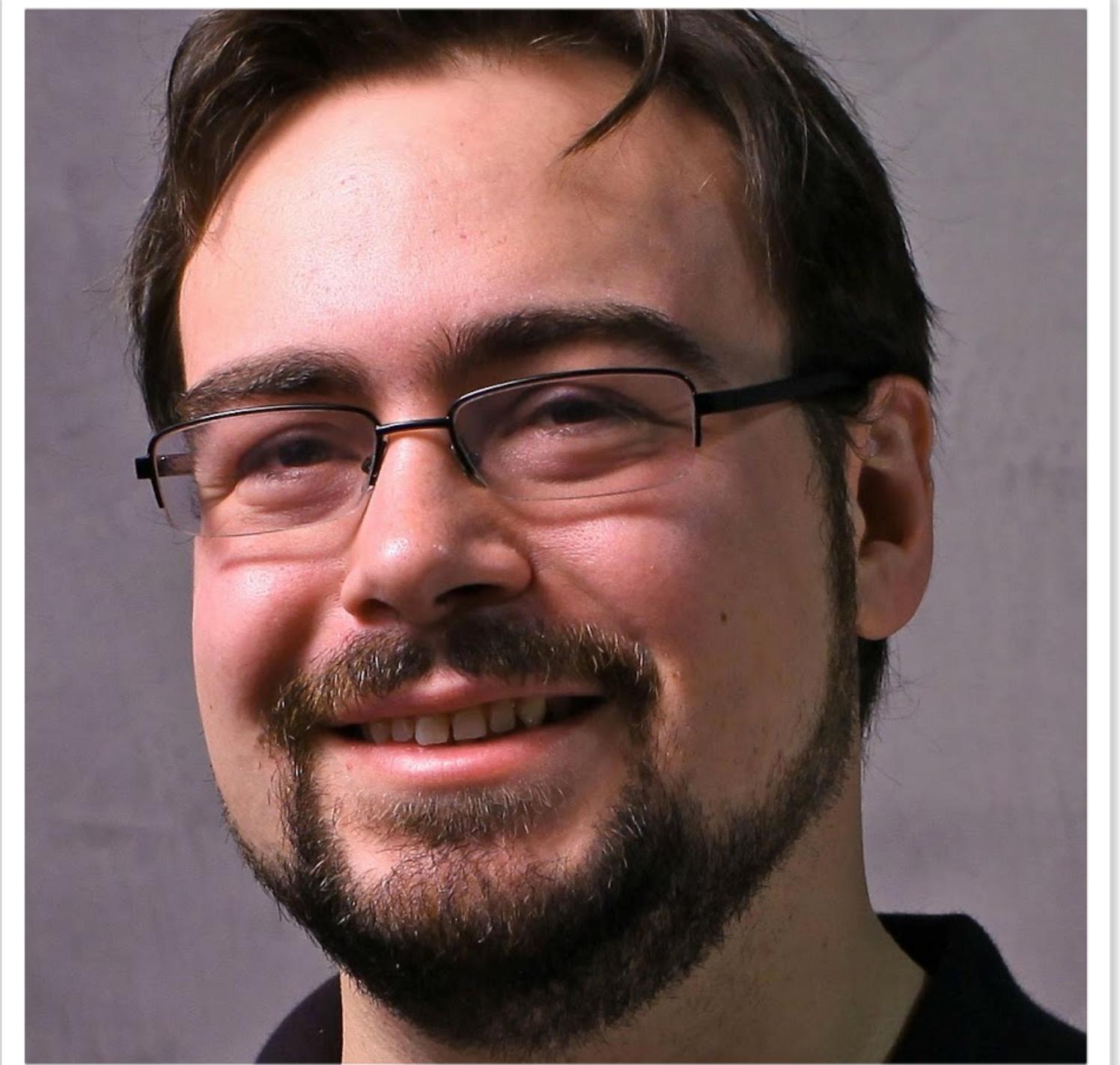
SproutCore was initially created in 2007 by Sproutit

In June 2010, the creator of SproutCore, Charles Jolley, left Apple to start Strobe

Strobe acquired by Facebook in November 2011

In December 2011, the SproutCore 2.0 framework was renamed to Ember.js

Founded by Yehuda Katz



The History of React

React was created by Jordan Walke

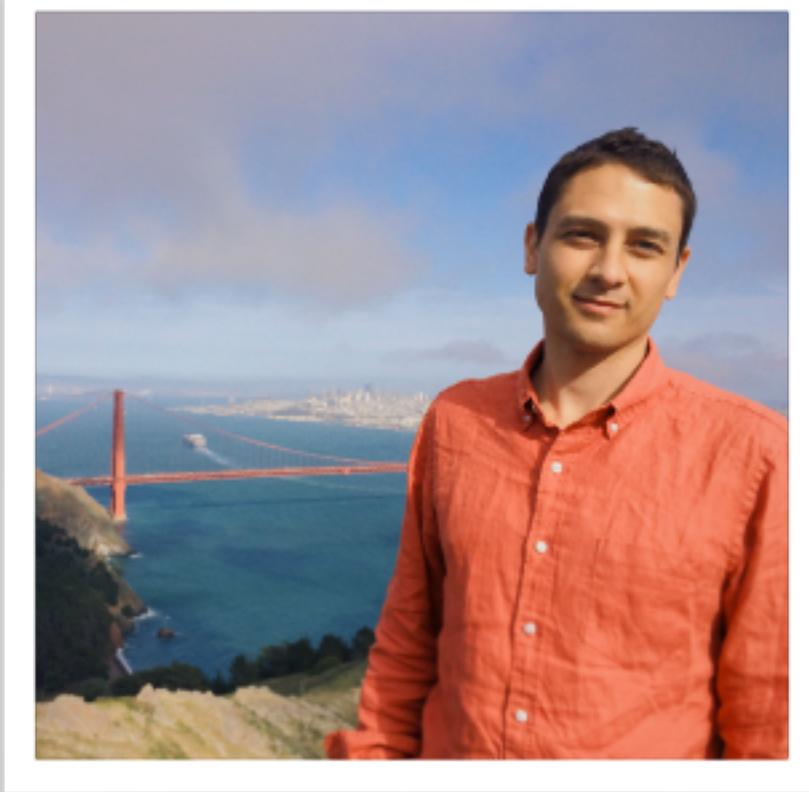
Inspired by XHP, an HTML components framework for PHP

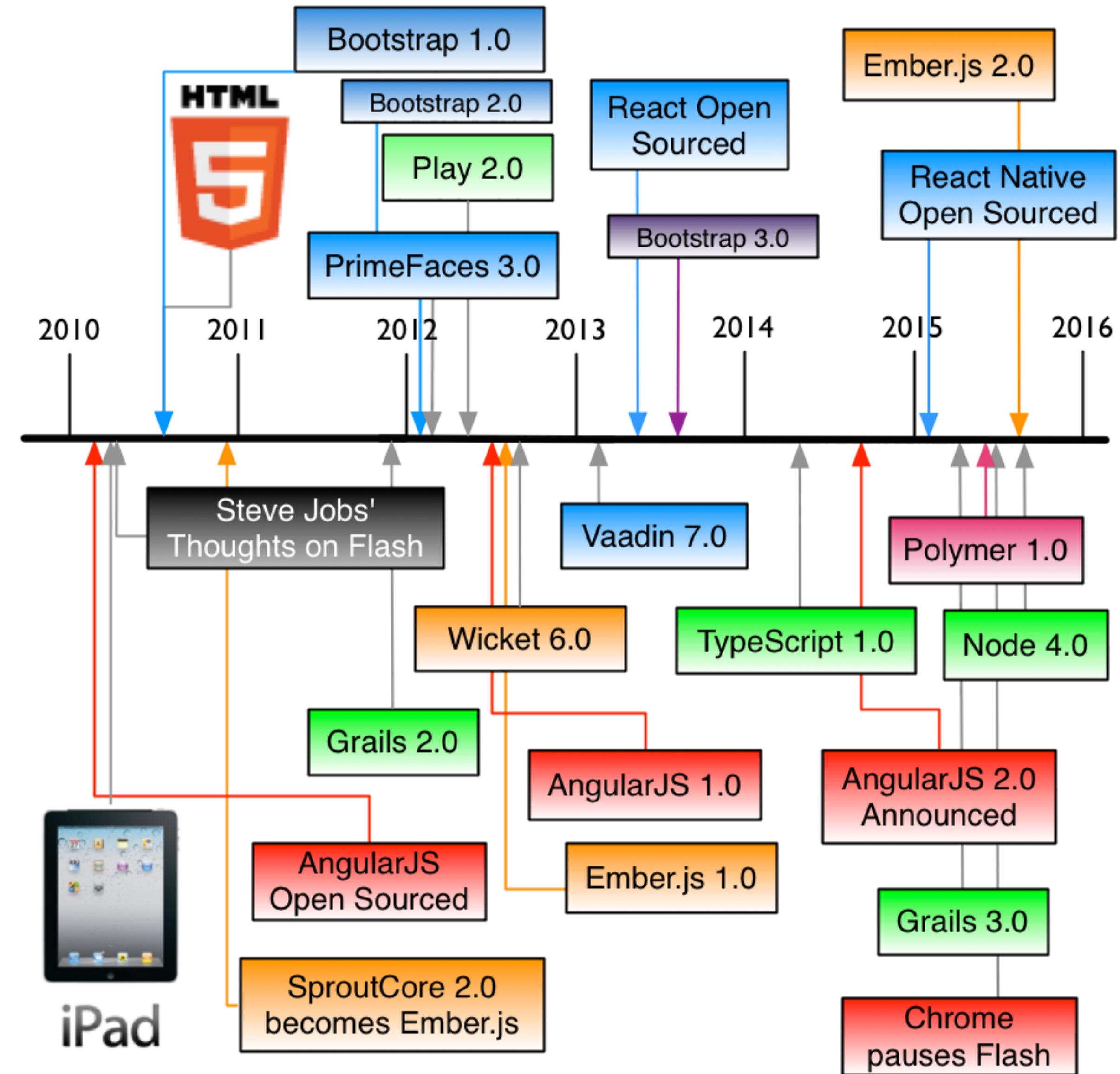
Open sourced in May 2013

Within one year, had large sites

Khan Academy, New York Times, Airbnb

+ Facebook and Instagram





	Angular	Ember	React
1. Learn			
2. Develop			
3. Test			
4. Secure			
5. Build			

Angular: Hello World

```
1. <!doctype html>
2. <html ng-app>
3.   <head>
4.     <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.4.5/angular.min.js">
</script>
5.   </head>
6.   <body>
7.     <div>
8.       <label>Name:</label>
9.       <input type="text" ng-model="yourName" placeholder="Enter a name here">
10.      <hr>
11.      <h1>Hello {{yourName}}!</h1>
12.    </div>
13.  </body>
14. </html>
```

AngularJS — Superheroic JavaScript Framework

Matt

https://angularjs.org

ANGULARJS Home Learn Develop Discuss Search

AngularJS by Google

HTML enhanced for web apps!

[View on GitHub](#) [Download \(1.4.5 / 1.2.28\)](#) [Design Docs & Notes](#)

Follow +AngularJS on [g+](#) Follow @angularjs 79.2K followers Tweet 4,808

Learn Angular in your browser for free!

Shaping up with Angular.js Matt

campus.codeschool.com/courses/shaping-up-with-angular-js/intro

SHAPING UP WITH ANGULAR.JS



Learn to build an application using Angular.js

[START COURSE](#)

SPONSORED BY  PRODUCED BY 

WHAT YOU'LL LEARN IN THIS COURSE

- Level 1 | Flatlander's Gem Store
- Level 2 | Built-in Directives
- Level 3 | Forms
- Level 4 | Custom Directives
- Level 5 | Services

COURSE BREAKDOWN

- 12 VIDEOS
- 27 CHALLENGES
- 6 BADGES

BADGES YOU'LL EARN

- 1 
- 2 
- 3 
- 4 
- 5 
- SHAPING UP WITH ANGULAR.JS 

AngularJS: Tutorial: Tutorial x Matt

https://docs.angularjs.org/tutorial

ANGULARJS Home Learn ▾ Develop ▾ Discuss ▾

v1.4.6-build.4222 (snapshot) / Tutorial

Tutorial

- 0 - Bootstrapping
- 1 - Static Template
- 2 - Angular Templates
- 3 - Filtering Repeaters
- 4 - Two-way Data Binding
- 5 - XHRs & Dependency Injection
- 6 - Templating Links & Images
- 7 - Routing & Multiple Views
- 8 - More Templating
- 9 - Filters
- 10 - Event Handlers
- 11 - REST and Custom Services
- 12 - Applying Animations

The End

PhoneCat Tutorial App

Click or press / to search

[Improve this Doc](#)

A great way to get introduced to AngularJS is to work through this tutorial, which walks you through the construction of an AngularJS web app. The app you will build is a catalog that displays a list of Android devices, lets you filter the list to see only devices that interest you, and then view details for any device.

The screenshot shows a web application interface. At the top left is a sidebar with a search bar containing 'Google' and a sort dropdown set to 'Alphabetical'. Below this is a list of devices: LG Axis, Nexus S, and two smaller thumbnail images. To the right of the list is a detailed card for the Nexus S. The card has a large image of the phone at the top. Below it is a summary section: 'Nexus S' followed by a paragraph of text about the device's features. Underneath is a grid of four smaller images of the Nexus S from different angles. At the bottom of the card are four tables with technical specifications: 'Availability and Networks', 'Battery', 'Storage and Memory', and 'Connectivity'. The 'Availability' table lists network providers like M1, O2, Orange, Singtel, StarHub, T-Mobile, and Vodafone. The 'Battery' table includes 'Type' (Lithium Ion (Li-Ion)), 'Talk Time' (6 hours), and 'Standby time (max)' (428 hours). The 'Storage and Memory' table lists 'RAM' (512MB) and 'Internal Storage' (16GB). The 'Connectivity' table lists 'Network Support' (Quad-band GSM: 850, 900, 1800, 1900 Tr-band HSPA: 900, 2100, 1700 HSPA type: HSUPA (7.2Mbps), HSUPA (5.76Mbps)), 'WiFi' (802.11 b/g/n), 'Bluetooth' (Bluetooth 2.1), and 'Infrared' (X).

Follow the tutorial to see how Angular makes browsers smarter — without the use of native extensions or plug-ins:

Matt

<https://builtwith.angularjs.org>

ANGULARJS Home Learn Develop Discuss Search

210 NEAT THINGS
BUILT WITH

ANGULARJS
by Google

Submit your AngularJS app

Search

Tags

Unsorted

VICE
RIDECHANNEL
GOOGLEDEVELOPERS

App Engine Office Hours

YouTube Developers Live

Google Maps Developers Office Hours

THRASHERMAGAZINE SURFER

YouTube on PS3

The YouTube application for Sony's PlayStation 3.

Production Entertainment Animations Local Storage Video API Google Closure No jQuery

angular/angular-seed

GitHub, Inc. [US] https://github.com/angular/angular-seed

This repository Search Pull requests Issues Gist

angular / angular-seed Watch 708 Star 9,451 Fork 5,715

Seed project for angular apps. <http://angularjs.org/>

179 commits 3 branches 0 releases 41 contributors

Branch: master angular-seed / +

chore(protractor): update to use protractor v2.1.0
petebacondarwin authored on Jun 15 latest commit a50f4c8a7a

File	Commit Message	Time Ago
app	chore(bower.son): update angular dep to 1.3.x, html5-boilerplate	3 months ago
e2e-tests	chore(protractor): update to use protractor v2.1.0	3 months ago
.bowerrc	chore/bower: move bower_components under the app folder	a year ago
.gitignore	chore(dependencies): remove inline dependencies (use npm & bower inst...	2 years ago
.jshintrc	refact(*): further architectural improvements	a year ago
.travis.yml	refact(*): further architectural improvements	a year ago
LICENSE	docs(LICENSE): update copyright year to 2014	2 years ago
README.md	docs(README): add alternative git clone method	10 months ago
bower.json	chore(bower.json): Update angular dependency to ~1.4.0	3 months ago
karma.conf.js	refact(*): further architectural improvements	a year ago
package.json	chore(protractor): update to use protractor v2.1.0	3 months ago

Code Issues 37 Pull requests 22 Wiki Pulse Graphs

HTTPS clone URL <https://github.com/>

You can clone with [HTTPS](#), [SSH](#), or [Subversion](#).

Clone in Desktop Download ZIP

A screenshot of a web browser displaying the Angular.io homepage. The page has a blue header with the Angular logo and navigation links for FEATURES, DOCS, ABOUT, and CONTRIBUTE. On the right, there's an 'INSTALL' button with a cloud icon. The main content area features a large red Angular logo with a white 'A' inside a pentagon shape. Below it, a text overlay reads: 'Angular is a development platform for building mobile and desktop web applications'. At the bottom, there are two buttons: 'LEARN IN 5 MINS' and 'DOWNLOAD'. The browser interface includes standard elements like tabs, a back/forward bar, and a toolbar with various icons.

Angular is a development platform for building mobile and desktop web applications

LEARN IN 5 MINS DOWNLOAD

Learning Angular: API Docs

The screenshot shows a web browser displaying the AngularJS API Reference page at <https://docs.angularjs.org/api>. The page is for version v1.4.6-build.4222 (snapshot). The main content area is titled "AngularJS API Docs". It features a sidebar on the left with a search bar and a list of categories: "ng", "function", and "directive". The "ng" category is expanded, showing sub-categories like "bind", "bootstrap", "copy", etc. The "function" category is also expanded, listing methods like "bind", "bootstrap", "copy", etc. The "directive" category lists "a", "form", "input", "input[checkbox]", and "input[date]". The main content area includes a welcome message, a note about Angular Prefixes (\$ and \$\$), and sections for "Angular Modules", "ng (core module)", and a table of core components.

Welcome to the AngularJS API docs page. These pages contain the AngularJS reference materials for version **1.4.5 permanent-internship**.

The documentation is organized into **modules** which contain various components of an AngularJS application. These components are **directives**, **services**, **filters**, **providers**, **templates**, global APIs, and testing mocks.

Angular Prefixes \$ and \$\$: To prevent accidental name collisions with your code, Angular prefixes names of public objects with `$` and names of private objects with `$$`. Please do not use the `$` or `$$` prefix in your code.

Angular Modules

ng (core module)

This module is provided by default and contains the core components of AngularJS.

Directives	This is the core collection of directives you would use in your template code to build an AngularJS application. Some examples include: <code>ngClick</code> , <code>ngInclude</code> , <code>ngRepeat</code> , etc...
Services / Factories	This is the core collection of services which are used within the DI of your application. Some examples include: <code>\$compile</code> , <code>\$http</code> , <code>\$location</code> , etc...
Filters	The core filters available in the ng module are used to transform template data before it is rendered within directives and expressions. Some examples include: <code>filter</code> , <code>date</code> , <code>currency</code> , <code>lowercase</code> , <code>uppercase</code> , etc...

Learning Angular: Videos

The screenshot shows a web browser window with the URL <https://www.youtube.com/user/angularjs>. The page displays the AngularJS YouTube channel interface. At the top, there's a large video thumbnail for a video titled "An Angular2 Todo App: First look at App Develop...". Below the thumbnail, the channel name "AngularJS" is displayed, along with a "Subscribe" button and a subscriber count of 76,649. The main content area features a list of videos, with the first video being the one shown in the thumbnail. To the right of the video list, there's a sidebar titled "Related channels" containing links to other YouTube channels: John Lindquist, Dan Wahlin, Accio Code, ng-conf, and Tony Alicea.

An Angular2 Todo App: First look at App Development in Angular2

231,967 views 6 months ago

An Angular2 Todo App: First look at App Development in Angular2
Speaker: David East (Angular Core Contributor), @_davideast

With ng-conf right around the corner it's time to catch a glimpse of Angular2. Angular Core Contributor David East will give an overview of what's new in Angular and live code a Todo app with Angular2 and Firebase. ...

[Read more](#)

Intro to Angular2

AngularJS

Home Videos Playlists Channels Discussion About

Subscribe 76,649

Related channels

- John Lindquist
- Dan Wahlin
- Accio Code
- ng-conf
- Tony Alicea

Learning Angular: Videos

A screenshot of a web browser window showing the Egghead.io website. The title bar reads "AngularJS Lessons - Screen". The address bar shows the URL <https://egghead.io/technologies/angularjs>. The page content is titled "AngularJS Lessons" and features a large red Angular logo (a white letter A inside a red hexagon). Below the logo, the text "AngularJS Lessons" is displayed. A descriptive paragraph explains that AngularJS is an open-source JavaScript framework maintained by Google, designed to assist with single-page applications by adding MVC capability. A call-to-action button at the bottom encourages users to "Unlock all AngularJS PRO Lessons" by clicking.

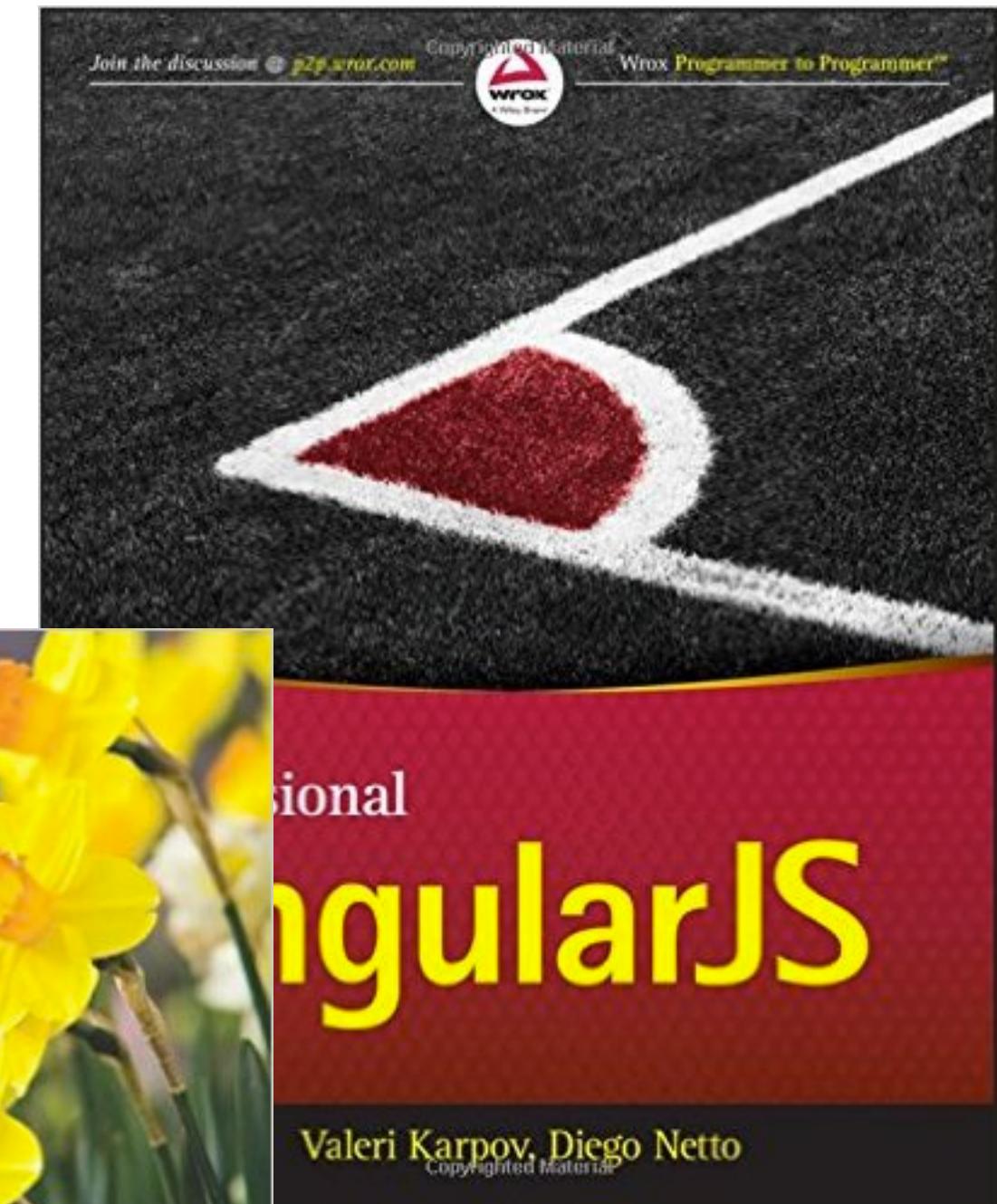
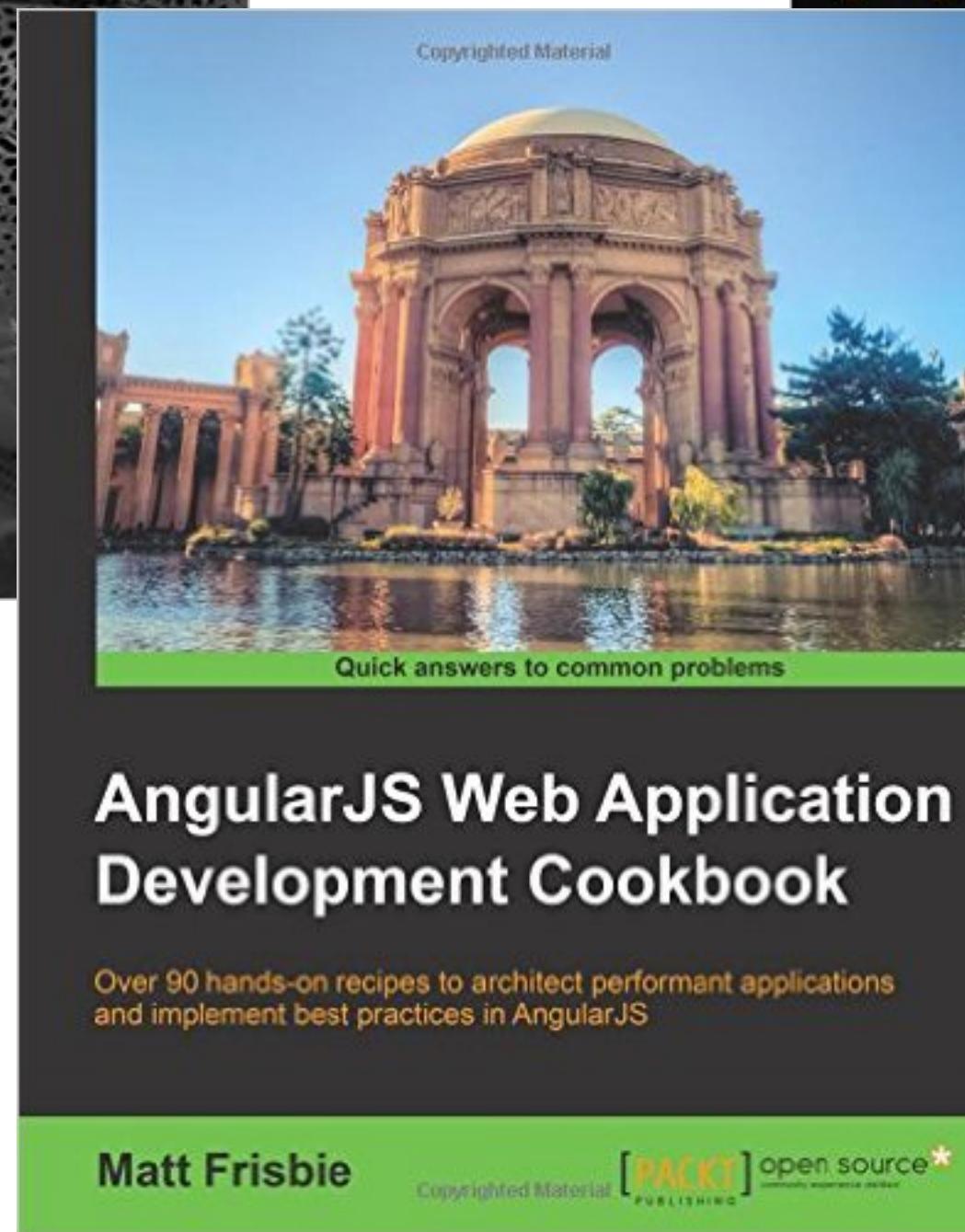
AngularJS Lessons

AngularJS is an open-source JavaScript framework, maintained by Google, that assists with running single-page applications. Its goal is to augment web-based applications with model–view–controller (MVC) capability, in an effort to make both development and testing easier.

Just starting out? [Check out this guide!](#)

Unlock all AngularJS PRO Lessons
click for instant access!

Learning Angular: Books



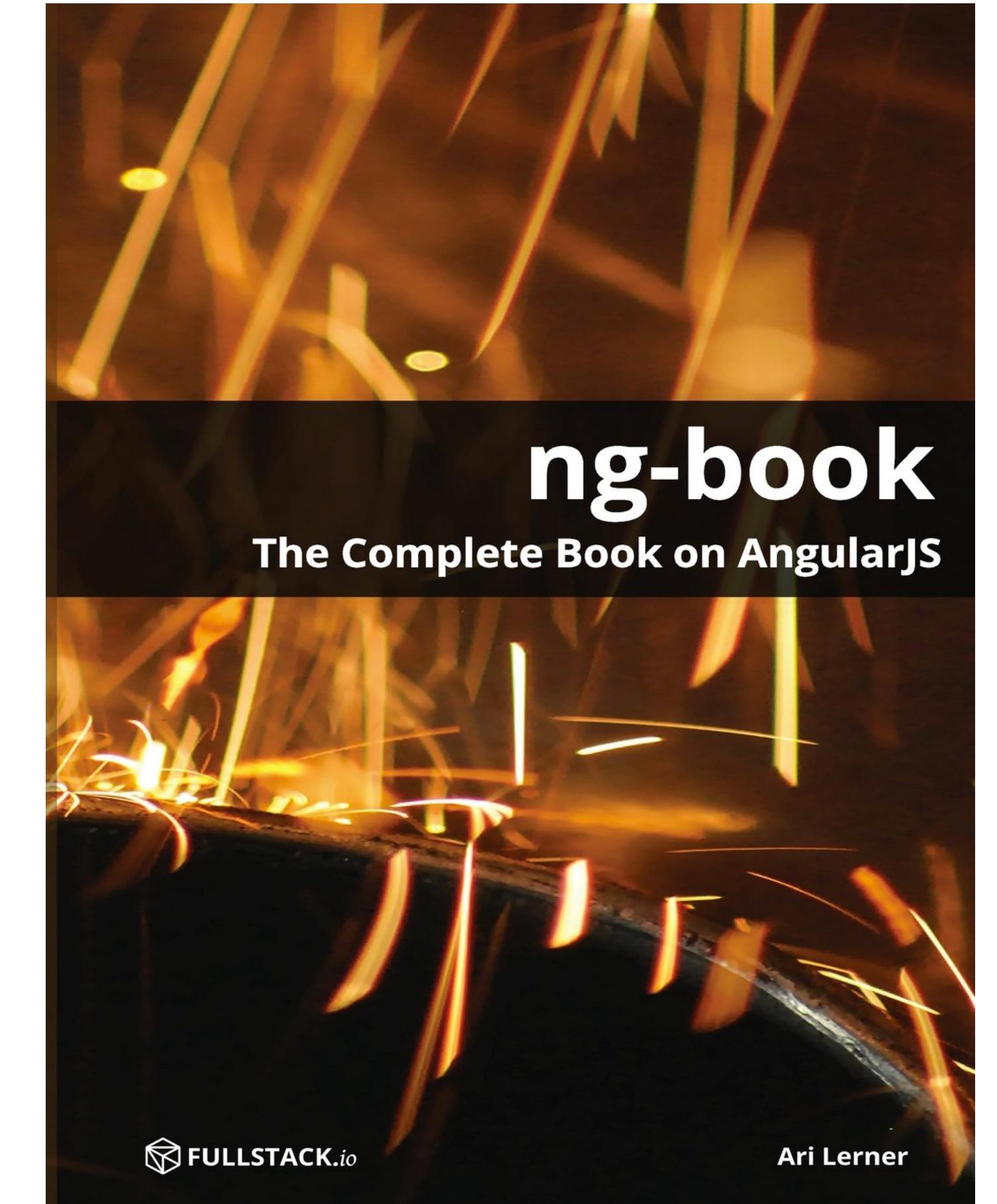
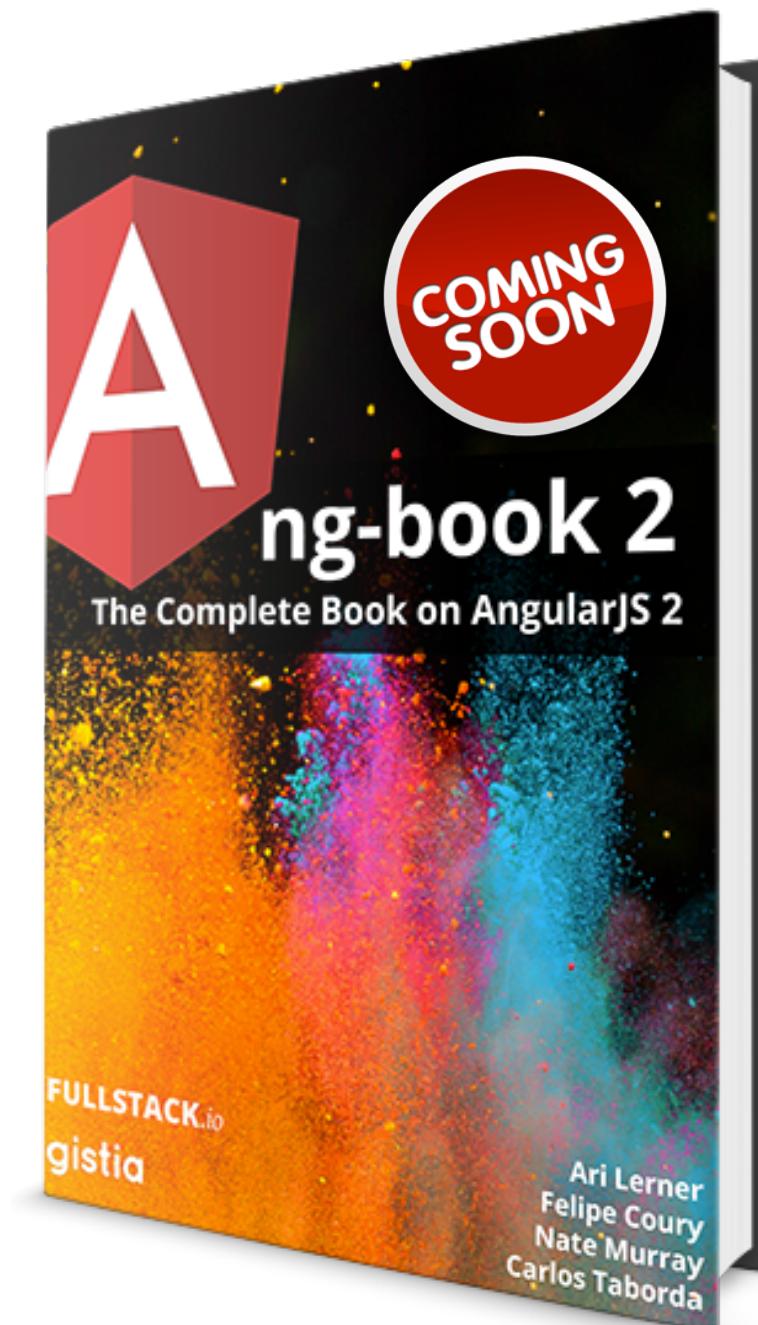
Learning Angular: ng-book

www.ng-book.com

Book and source: \$39

Book, source and videos: \$79

Team License: \$299



Learning Angular: Stack Overflow



Questions Tags Users Badges Unanswered Ask Question

Tagged Questions

info newest 10 featured frequent votes active unanswered

AngularJS is an open-source JavaScript framework. Its goal is to augment browser-based applications with Model–View–Whatever(MV*) capability and reduce the amount of JavaScript needed to make web applications functional. These types of apps are also frequently known as Single-Page Applications.

[learn more...](#) [improve tag info](#) [top users](#) [synonyms \(2\)](#)

1683

votes

22
answers

406k
views

Service vs provider vs factory

What are the differences between AngularJS module's Service, Provider and Factory?

[angularjs](#) [dependency-injection](#) [angularjs-service](#) [angularjs-factory](#) [angularjs-provider](#)

asked Mar 27 '13 at 17:59



Lior

11.9k ● 8 ● 22 ● 35

906

votes

6
answers

204k
views

Databinding in AngularJS

How does databinding work in the AngularJS framework? I haven't found technical details on their site. It's more or less clear how it works when data is propagated from view to model. But how does ...

[javascript](#) [angularjs](#)

asked Mar 13 '12 at 10:16



Pashec

5,910 ● 3 ● 14 ● 21

118,944
questions tagged

[angularjs](#) [about »](#)

Featured on Meta

Warlords of Documentation:
A Proposed Expansion of
Stack Overflow

Hot Meta Posts

- 15 [Classification \(or equivalent\) option for favorited questions](#)
- 20 [How to safely determine if an external link points to a legitimate website?](#)
- 10 [Reach out to SQL Server team?](#)

Favorite Tags [edit](#)

[Add a favorite tag](#)

	Angular	Ember	React
1. Learn	10		
2. Develop			
3. Test			
4. Secure			
5. Build			

	Angular	Ember	React
1. Learn	10		
2. Develop			
3. Test			
4. Secure			
5. Build			

Ember: Hello World

```
npm install -g ember-cli
```

```
npm install -g phantomjs2
```

```
ember new my-new-app
```

```
cd my-new-app
```

```
ember server
```

<http://guides.emberjs.com/v2.0.0/getting-started/>

Ember: Hello World

```
<!doctype html>
<html>
<head>
  <script src="https://code.jquery.com/jquery-2.1.4.min.js"></script>
  <script src="https://cdnjs.cloudflare.com/ajax/libs/ember.js/2.0.2/
ember.min.js"></script>
  <script src="https://cdnjs.cloudflare.com/ajax/libs/ember.js/2.0.2/
ember-template-compiler.js"></script>
  <script>
    var App = Ember.Application.create();
  </script>
</head>
<body>
  <script type="text/x-handlebars" data-template-name='index'>
    <p>Hello Ember!</p>
  </script>
</body>
</html>
```

<http://gilesbowkett.blogspot.com/2013/04/a-hello-world-in-emberjs.html>

Ember.js - A framework for c × Matt

emberjs.com

ember

ABOUT GUIDES API COMMUNITY BLOG BUILDS FORK US!

A framework for creating ambitious web applications

new
JOB BOARD!

npm install -g ember-cli
ember new my-app

More downloads

Follow @emberjs Follow on g+

MORE PRODUCTIVE OUT OF THE BOX.



Write dramatically less code with Ember's Handlebars integrated templates that update automatically when the underlying data changes.



Don't waste time making trivial choices. Ember.js incorporates common idioms so you can focus on what makes your app special, not reinventing the wheel.



Ember.js is built for productivity. Designed with developer ergonomics in mind, its friendly APIs help you get your job done—fast.

```
<div>
  <label>Name:</label>
  {{input type="text" value=name placeholder="Enter your name"}}
</div>
<div class="text">
  <h3>My name is {{name}} and I want to learn Ember!</h3>
</div>
```

Ember.js Tutorial - Code Sch x Matt

https://www.codeschool.com/courses/warming-up-with-ember-js

The screenshot shows the Code School website with the URL https://www.codeschool.com/courses/warming-up-with-ember-js. The page features a large background illustration of a forest with a path leading to a treehouse. A circular icon on the left contains the text "Warming Up With ember.js". A yellow button labeled "JAVASCRIPT" is positioned above the main title "WARMING UP WITH EMBER.JS". Below the title is a blue button labeled "START COURSE FOR FREE". The top navigation bar includes links for Courses, Screencasts, Pricing, CREATE FREE ACCOUNT, and Sign in. The bottom navigation bar shows the path "Paths > JavaScript > Warming Up With Ember.js" and buttons for "WATCH VIDEOS" and "DISCUSS COURSE". The "COURSE DESCRIPTION" section explains the goal of learning Ember.js from scratch. The "COURSE OVERVIEW" section highlights the first level, "Warming Up", which includes 3 Videos and 7 Challenges. The "ABOUT THE PROFESSOR" section features a profile for Adam Fortuna, detailing his background in Rails and his current interests in iOS development.

Code School
a Pluralsight company

Courses Screencasts Pricing CREATE FREE ACCOUNT Sign in

JAVASCRIPT

WARMING UP WITH EMBER.JS

START COURSE FOR FREE

Paths > JavaScript > Warming Up With Ember.js

WATCH VIDEOS DISCUSS COURSE

COURSE DESCRIPTION

Learn to demystify the Ember.js JavaScript framework by breaking it down. See how to build an application from scratch using Ember.js and Twitter Bootstrap.

COURSE OVERVIEW

LEVEL 1 | FREE LEVEL

Warming Up 3 Videos | 7 Challenges

Learn how to create an Ember application, set up your first Handlebars template, and meet the Ember Router.

ABOUT THE PROFESSOR

Adam Fortuna

Adam came to the beta in Orlando, FL for Code School's Rails Best Practices course and they haven't been able to keep him away since. As a Java/Flex convert, Adam made a home in the Rails, Ruby and JavaScript world and has lately started exploring the wonderful land of iOS.

Learning Ember: API Docs

The screenshot shows a web browser window displaying the Ember.js API documentation. The URL in the address bar is `emberjs.com/api/`. The page has a red header with the word "ember" in white. The main content area is titled "EMBER NAMESPACE". On the left, there is a sidebar with a "PROJECTS" section containing "Ember" (selected), "Ember Data", and "TAG: V2.0.1". Below that is a "MODULES" section listing various modules like "ember", "ember-application", "ember-debug", etc. Another "MODULES" section lists "Ember", "Ember.AutoLocation", "Ember.FEATURES", "Ember.Instrumentation", and "Ember.Location". The main content area starts with a "PUBLIC" section, followed by "DEFINED IN: packages/ember-metal/lib/core.js:8" and "MODULE: ember-metal". It explains that this namespace contains all Ember methods and functions, and future versions may overwrite it. It also notes that the shorthand `Em` can be used instead of `Ember`. The "METHODS" section lists methods such as \$, A, K, addListener, addObserver, assert, bind, cacheFor, copy, debug, isEqual, isNone, isPresent, merge, on, removeListener, removeObserver, runInDebug, sendEvent, and set. At the bottom of the methods list, there are checkboxes for "Show: Inherited", "Protected", "Private", and "Deprecated".

Learning Ember: Guides

The screenshot shows a web browser window displaying the Ember.js Guides and Tutorials website. The URL in the address bar is `guides.emberjs.com/v2.0.0/`. The page has a red header with the Ember logo and navigation links for ABOUT, GUIDES (which is highlighted), API, COMMUNITY, BLOG, and BUILDS. There is also a search bar and a 'FORK US!' button. On the left, there is a sidebar with a dropdown menu set to 'v2.0.0' and a list of topics: GETTING STARTED, THE OBJECT MODEL, TEMPLATES, ROUTING, COMPONENTS, CONTROLLERS, MODELS, SERVICES, TESTING, EMBER INSPECTOR, ADDONS AND DEPENDENCIES, CONFIGURING EMBER.JS, and CONTRIBUTING TO EMBER.JS. The main content area is titled 'GUIDES AND TUTORIALS' and contains an introduction, instructions for contributing, and a note about the next step.

v2.0.0

GETTING STARTED

THE OBJECT MODEL

TEMPLATES

ROUTING

COMPONENTS

CONTROLLERS

MODELS

SERVICES

TESTING

EMBER INSPECTOR

ADDONS AND DEPENDENCIES

CONFIGURING EMBER.JS

CONTRIBUTING TO EMBER.JS

GUIDES AND TUTORIALS

Welcome to the Ember.js guides! This documentation will take you from total beginner to Ember expert. It is designed to start from the basics, and slowly increase to more sophisticated concepts until you know everything there is to know about building awesome web applications.

These guides are written in Markdown and are [available on GitHub](#). If there is something missing, or you find a typo or mistake, please help us by filing an issue or submitting a pull request. Thanks!

We're excited for all of the great apps you're going to build with Ember.js. To get started, select a topic from the left. They are presented in the order that we think will be most useful to you as you're learning Ember.js, but you can also jump to whatever seems most interesting.

Good luck!

We're done with Guides and Tutorials. Next up: Getting Started - Installing Ember →

Learning Ember: Videos

No lessons on Egghead.io

www.emberscreencasts.com

emberwatch.com

Talks, Screencasts, Podcasts,

Tutorials, Books, Cookbook



Learning Ember: Books



Learning Ember: Stack Overflow

 stackoverflow

Questions Tags Users Badges Unanswered Ask Question

Tagged Questions info newest frequent votes active unanswered

Ember.js is an MVC application framework written in JavaScript. Consider making your question easier to answer by using the pre-configured templates from this tag's wiki (click learn more).

[learn more...](#) [improve tag info](#) [top users](#) [synonyms \(2\)](#)

291 votes **14** answers **53k** views

Separate REST JSON API server and client?

I'm about to create a bunch of web apps from scratch. (See <http://50pop.com/code> for overview.) I'd like for them to be able to be accessed from many different clients: front-end websites, smartphone ...

[ruby-on-rails-3](#) [rest](#) [backbone.js](#) [sinatra](#) [ember.js](#) asked Jun 7 '12 at 23:27  sivers 1,560 ● 3 ● 6 ● 7

190 votes **1** answer **49k** views

What are the key differences between Meteor, Ember.js and Backbone.js? [closed]

Learning Ember.js / Backbone.js has been on my to-do list for a while. Now that Meteor is out, I am just wondering if anyone with experience of Meteor, Ember.js and Backbone.js can summarize the key ...

[javascript](#) [backbone.js](#) [knockout.js](#) [ember.js](#) [meteor](#) asked Apr 11 '12 at 2:26  Kay Zhu 12k ● 3 ● 38 ● 60

16,530 questions tagged [ember.js](#) [about »](#)

Featured on Meta

 Warlords of Documentation: A Proposed Expansion of Stack Overflow

Hot Meta Posts

20 [Classification \(or equivalent\) option for favorited questions](#)
15 [Let's not put badge tracking everywhere](#)
75 [What do I do when I'm being insulted in a post?](#)

Favorite Tags [edit](#)
[Add a favorite tag](#)

	Angular	Ember	React
1. Learn	10	6	
2. Develop			
3. Test			
4. Secure			
5. Build			

React: Hello World

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8" />
    <title>Hello React!</title>
    <script src="build/react.js"></script>
    <script src="build/JSXTransformer.js"></script>
  </head>
  <body>
    <div id="example"></div>
    <script type="text/jsx">
      React.render(
        <h1>Hello, world!</h1>,
        document.getElementById('example')
      );
    </script>
  </body>
</html>
```

React: Hello World

Live JSX Editor

Compiled JS

```
var HelloMessage = React.createClass({
  render: function() {
    return <div>Hello {this.props.name}</div>;
  }
});

React.render(<HelloMessage name="John" />, mountNode);
```

React: Hello {name} without JSX

```
Live JSX Editor   Compiled JS

var HelloMessage = React.createClass({displayName: "HelloMessage",
  render: function() {
    return React.createElement("div", null, "Hello ", this.props.name);
  }
});

React.render(React.createElement(HelloMessage, {name: "John"}),
mountNode);
```

A JavaScript library for building user interfaces

React

A JAVASCRIPT LIBRARY FOR BUILDING USER INTERFACES

[Get Started](#) [Download React v0.13.3](#)

JUST THE UI

Lots of people use React as the V in MVC. Since React makes no assumptions about the rest of your technology stack, it's easy to try it out on a small feature in an existing project.

VIRTUAL DOM

React abstracts away the DOM from you, giving a simpler programming model and better performance. React can also render on the server using Node, and it can power native apps using [React Native](#).

DATA FLOW

React implements one-way reactive data flow which reduces boilerplate and is easier to reason about than traditional data binding.

A Simple Component

React components implement a `render()` method that takes input data and returns what to display. This example uses an XML-like syntax called JSX. Input data that is passed into the component can be accessed by `render()` via `this.props`.

JSX is optional and not required to use React. Try clicking on "Compiled JS" to see the raw JavaScript code produced by the JSX compiler.

Learning React: API Docs

A screenshot of a web browser displaying the React 'Getting Started' documentation. The browser window has a title bar showing 'Getting Started | React' and a tab bar with 'facebook.github.io/react/docs/getting-started.html'. The main content area shows the React logo and navigation links for 'Docs', 'Support', 'Download', and 'Blog'. On the right, there are links for 'GitHub' and 'React Native'. The main article is titled 'Getting Started' with a 'Edit on GitHub' link. It features sections for 'JSFiddle' (with examples) and 'Starter Kit' (with a download button). A sidebar on the left contains sections for 'QUICK START' (with 'Getting Started', 'Tutorial', and 'Thinking in React' links), 'COMMUNITY RESOURCES' (with 'Conferences', 'Videos', 'Complementary Tools', and 'Examples' links), and 'GUIDES' (with 'Why React?', 'Displaying Data', 'JSX in Depth', and 'JSX Spread Attributes' links).

Getting Started

[Edit on GitHub](#)

JSFiddle

The easiest way to start hacking on React is using the following JSFiddle Hello World examples:

- [React JSFiddle](#)
- [React JSFiddle without JSX](#)

Starter Kit

Download the starter kit to get started.

[Download Starter Kit 0.13.3](#)

In the root directory of the starter kit, create a `helloworld.html` with the following contents.

Learning React: Videos

A screenshot of a web browser displaying the React documentation website at facebook.github.io/react/docs/videos.html. The browser window has a title bar "Videos | React" and a toolbar with various icons. The main content area shows the "Videos" section of the React documentation. On the left, there's a sidebar with "QUICK START" links (Getting Started, Tutorial, Thinking in React), "COMMUNITY RESOURCES" links (Conferences, Videos, Complementary Tools, Examples), and "GUIDES" links (Why React?, Displaying Data, JSX in Depth, JSX Spread Attributes, JSX Gotchas, Interactivity and Dynamic UIs, Multiple Components, Reusable Components, Transferring Props, Forms). The main content area features a large video thumbnail for "Pete Hunt: React: Rethinking best practices – JSConf EU 2013". The video thumbnail shows Pete Hunt speaking at a podium with a large "JS" logo. A play button is overlaid on the video. To the right of the video, there's a summary text: "The framework cannot know how to separate your concerns for you. This tool is a React component. A highly cohesive building block for UIs loosely coupled with other components." Below the video, there's a quote: "'At Facebook and Instagram, we're trying to push the limits of what's possible on the web with React. My talk will start with a brief introduction to the framework, and then dive into three controversial topics: Throwing out the notion of templates and building views with JavaScript, "re-rendering" your entire application when your data changes, and a lightweight implementation of the DOM and events.' -- [Pete Hunt](#)".

Learning React: Videos

A screenshot of a web browser window showing the Egghead.io website. The title bar reads "React Lessons - Screencast" and the address bar shows the URL "https://egghead.io/technologies/react". The page content is titled "React Lessons" and features a large blue atomic symbol logo. Below the logo, the text reads: "Lots of people use React as the V in MVC. Since React makes no assumptions about the rest of your technology stack, it's easy to try it out on a small feature in an existing project." At the bottom, there is a red button with the text "Unlock all React PRO Lessons" and "click for instant access!"

React Lessons

Lots of people use React as the V in MVC. Since React makes no assumptions about the rest of your technology stack, it's easy to try it out on a small feature in an existing project.

Unlock all React PRO Lessons
click for instant access!

Learning React: Videos



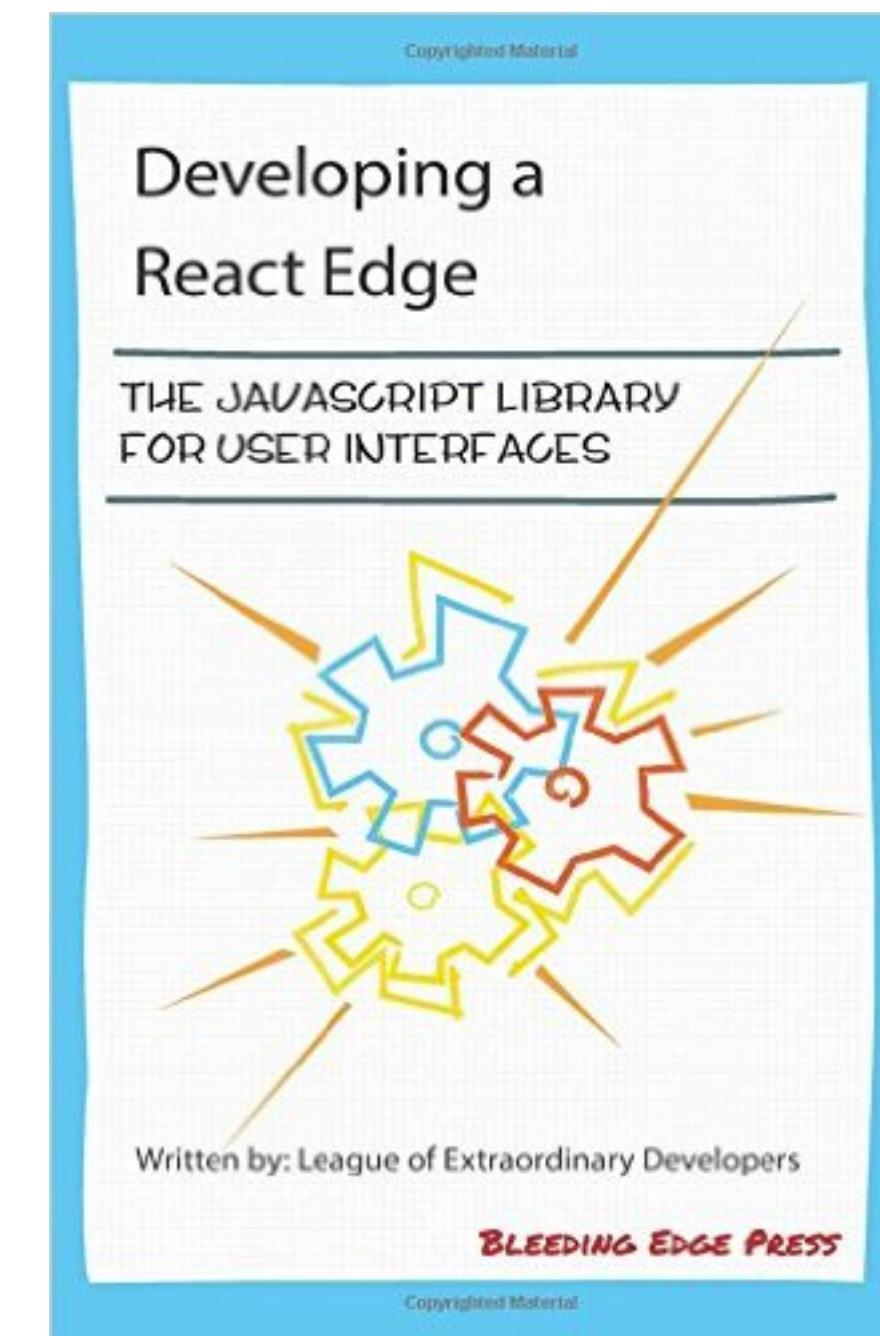
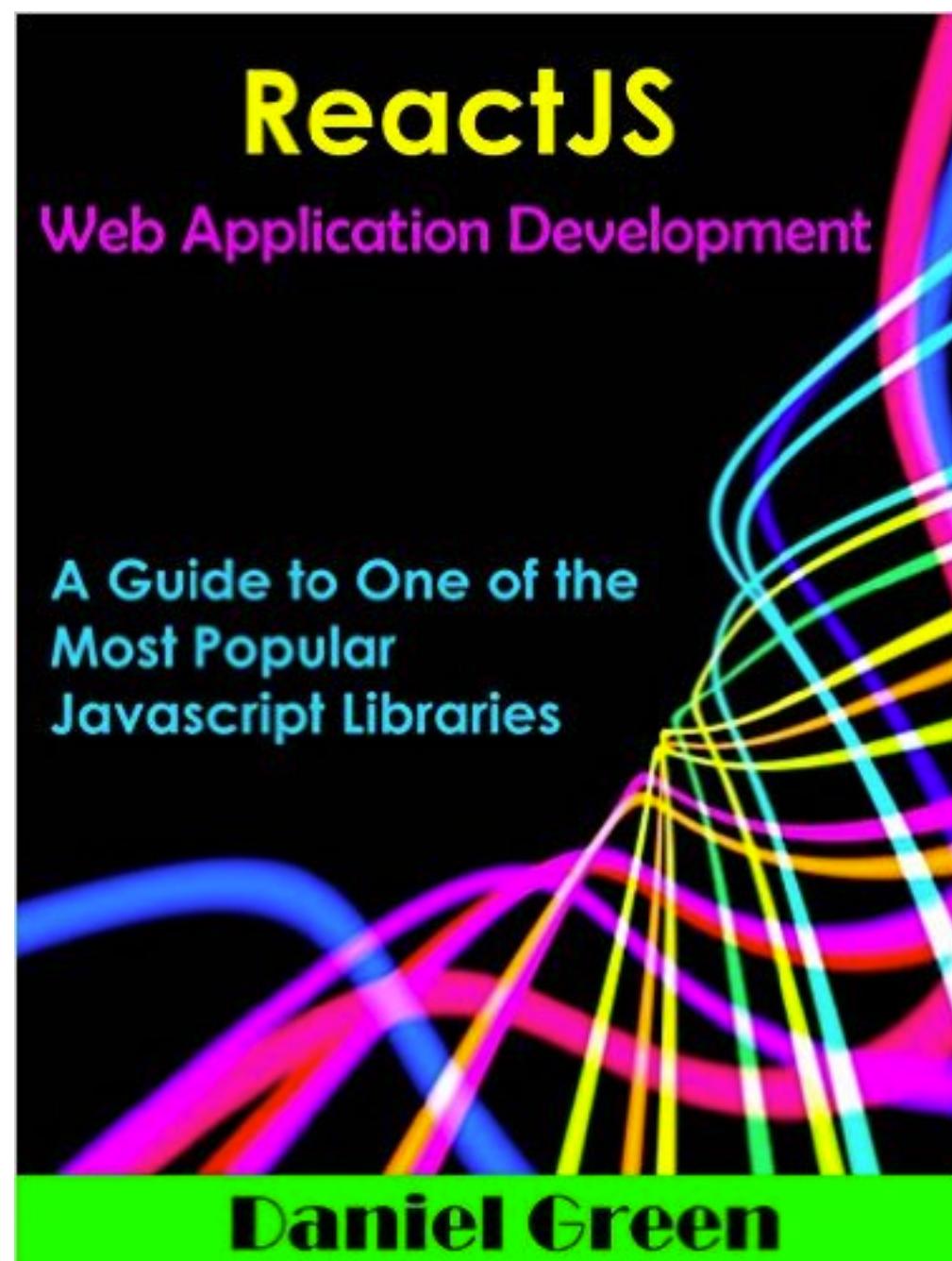
React is a "javascript library for building user interfaces" from the fine folks at Facebook & Instagram. It strictly focuses on the "V" in your MVC applications, and touts a blazing fast virtual DOM diff implementation.

This series will explore the basic fundamentals of React to get you started.

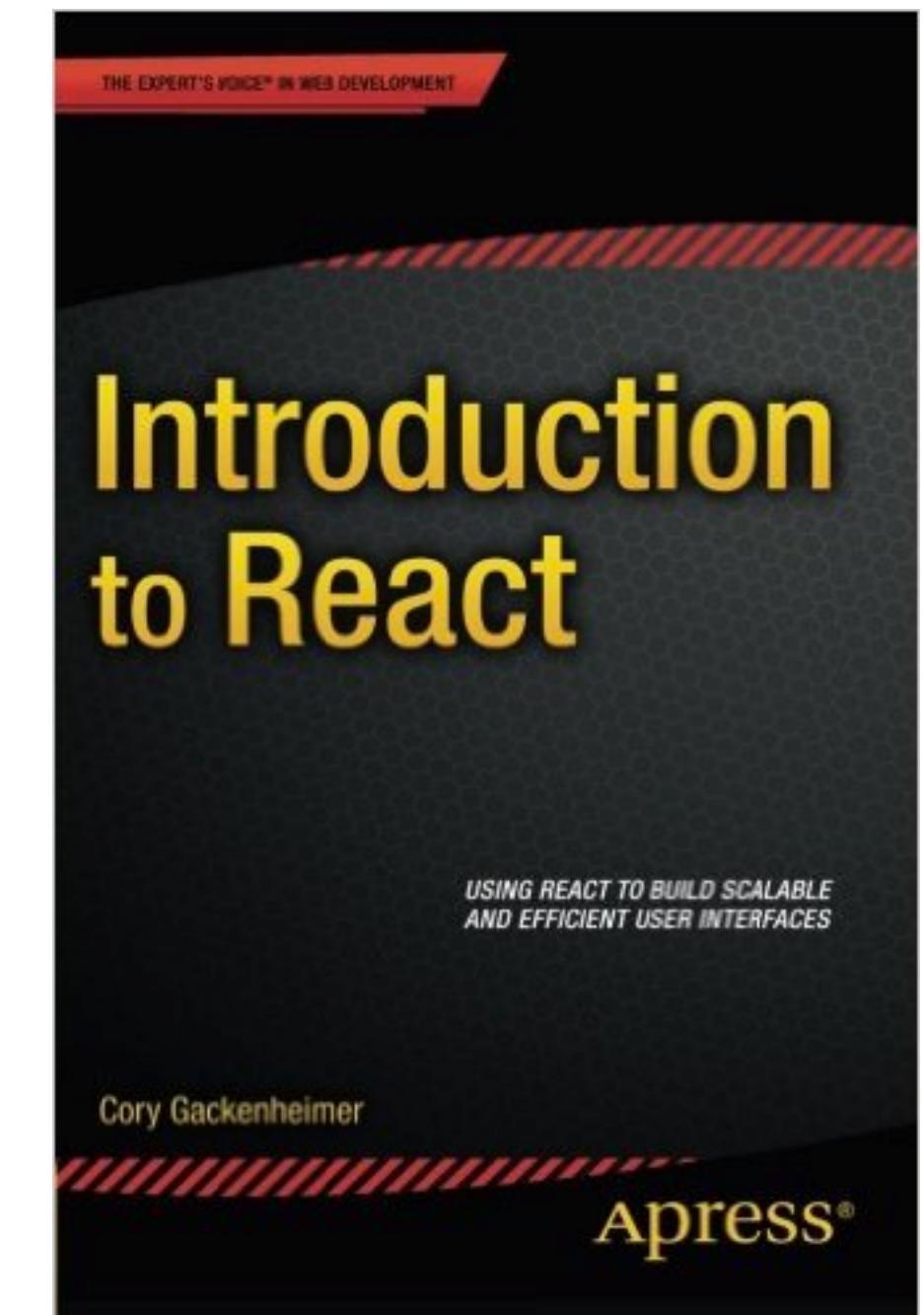
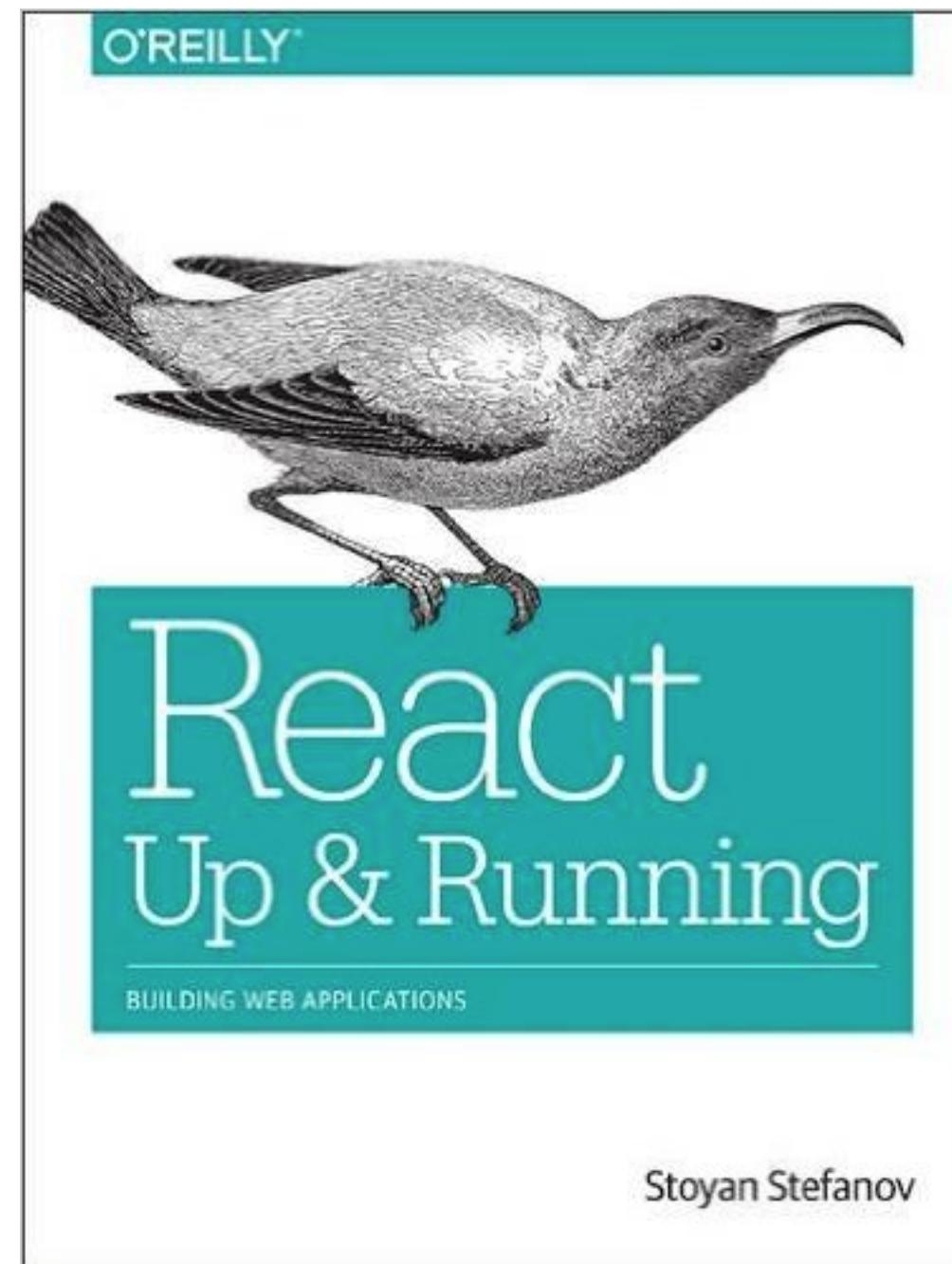
When you've finished with this series, be sure to check out [React Flux Architecture](#) for real-world React application development.

<https://egghead.io/series/react-fundamentals>

Learning React: Books



Learning React: Books (Coming Soon)



Learning React: Stack Overflow

 stackoverflow

Questions Tags Users Badges Unanswered Ask Question

Tagged Questions info newest 2 featured frequent votes active unanswered

React is a JavaScript library for building user interfaces. It uses a declarative paradigm that makes it easier to reason about your application and aims to be both efficient and flexible.

learn more... improve tag info top users synonyms

115 votes 3 answers 29k views

Why is React's concept of Virtual DOM said to be more performant than dirty model checking?

I saw a React dev talks at <http://www.youtube.com/watch?v=x7cQ3mrcKaY> and speaker mentioned that dirty-check of the model can be slow. But isn't calculating the diff between virtual DOMs actually even ...

javascript reactjs

asked Jan 14 '14 at 8:57 by  Danill 1,285 ● 2 ● 8 ● 21

81 votes 6 answers 16k views

Where should ajax request be made in Flux app?

I'm creating a react.js application with flux architecture and I am trying figure out where and when a request for data from the server should be made. Is there a any example for this. (Not TODO ...)

javascript reactjs reactjs-flux

asked Oct 29 '14 at 14:04 by  Eniz Gülek 450 ● 1 ● 5 ● 7

5,053 questions tagged reactjs about »

Featured on Meta

Warlords of Documentation: A Proposed Expansion of Stack Overflow

Hot Meta Posts

24 Reach out to SQL Server team?

38 Is this flagging behavior expected?

25 Can we increase the visibility of deleted posts?

Favorite Tags [edit](#)

Add a favorite tag

	Angular	Ember	React
1. Learn	10	6	8
2. Develop			
3. Test			
4. Secure			
5. Build			

Interesting: Stack Overflow Top Questions

AngularJS is an open-source JavaScript framework. Its goal is to augment browser-based applications with Model–View–Whatever(MV*) capability and reduce the amount of JavaScript needed to make web applications functional. These types of apps are also frequently known as Single-Page Applications

[learn more...](#) [improve](#) [tag info](#) [top users](#) [synonyms](#)

1683 votes

Service vs provider

What are the differences between a service and a provider?

angularjs dependency injection

22 answers

406k views

906 votes

Databinding in Angular

How does databinding work? I am trying to understand it more or less clear how it works.

javascript angularjs

6 answers

204k views

Ember.js is an MVC application framework written in JavaScript. Consider making your question easier to answer by using the pre-configured templates from the Ember.js documentation.

ember.js mvc

learn more... [improve](#) [tag info](#) [top users](#) [synonyms](#)

291 votes

Separate REST JSON API

I'm about to create a build system that will generate them to be able to be a REST API.

ruby-on-rails-3 rest

14 answers

53k views

React is a JavaScript library for building user interfaces. It uses a declarative paradigm that makes it easier to reason about your application and aims to be both efficient and flexible.

react

learn more... [improve](#) [tag info](#) [top users](#) [synonyms](#)

115 votes

Why is React's concept of Virtual DOM said to be more performant than dirty model checking?

I saw a React dev talks at <http://www.youtube.com/watch?v=x7cQ3mrcKaY> and speaker mentioned that dirty-check of the model can be slow. But isn't calculating the diff between virtual DOMs actually even ...

asked Jan 14 '14 at 8:57

 Daniil
1,285 ● 2 ● 8 ● 21

3 answers

29k views

javascript reactjs

190 votes

What are the key differences between Backbone.js and Ember.js?

Learning Ember.js / Backbone.js. I am new to front-end development and I am wondering if anyone with experience can help me understand what are the key differences between Backbone.js and Ember.js.

javascript backbone.js

1 answer

49k views

81 votes

Where should ajax request be made in Flux app?

I'm creating a react.js application with flux architecture and I am trying figure out where and when a request for data from the server should be made. Is there a any example for this. (Not TODO ...)

asked Oct 29 '14 at 14:04

 Eniz Gülek
450 ● 1 ● 5 ● 7

6 answers

16k views

javascript reactjs reactjs-flux

	Angular	Ember	React
1. Learn	10	6	8
2. Develop			
3. Test			
4. Secure			
5. Build			

Routing

RESTful routing

Extracts query and path params

ngRoute by default

De-facto: **ui-router**

Nested Routes



RESTful routing

Extracts query and path params

Uses conventions

Nested routes

Wildcards



Helping you **select** an MV* framework

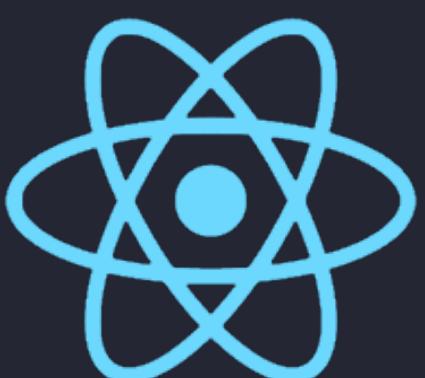
Framework selection guide

No routing, only V

Director used in TodoMVC example

React Router 2x more popular

Inspired by Ember



Routing with ngRoute



```
angular.module('todomvc', ['ngRoute'])
  .config(function ($routeProvider) {
    'use strict';

    var routeConfig = {
      controller: 'TodoCtrl',
      templateUrl: 'todomvc-index.html',
      resolve: {
        store: function (todoStorage) {
          // Get the correct module (API or localStorage)
        }
      }
    };

    $routeProvider
      .when('/', routeConfig)
      .when('/:status', routeConfig)
      .otherwise({
        redirectTo: '/'
      });
  });
}
```



Routing with Ember



```
(function () {
  'use strict';

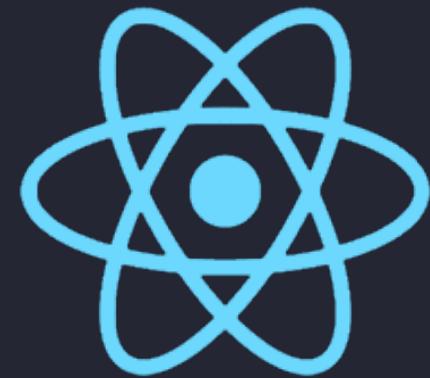
  Todos.Router.map(function () {
    this.resource('todos', { path: '/' }, function () {
      this.route('active');
      this.route('completed');
    });
  });

  Todos.TodosRoute = Ember.Route.extend({
    model: function () {
      return this.store.find('todo');
    }
  });

  Todos.TodosIndexRoute = Todos.TodosRoute.extend({
    templateName: 'todo-list',
    controllerName: 'todos-list'
  });

  // Todos.TodosActiveRoute & Todos.TodosCompletedRoute
})();
```

Routing with Director



```
var TodoApp = React.createClass({
  getInitialState: function () {
    return {
      nowShowing: app.ALL_TODOS,
      editing: null
    };
  },
  componentDidMount: function () {
    var setState = this.setState;
    var router = Router({
      '/': setState.bind(this, {nowShowing: app.ALL_TODOS}),
      '/active': setState.bind(this, {nowShowing: app.ACTIVE_TODOS}),
      '/completed': setState.bind(this, {nowShowing: app.COMPLETED_TODOS})
    });
    router.init('/');
  },
  // lots of event handlers and render()
});
```

Templates

HTML5 Templates

Angular-specific elements and attributes

Directives, EL, filters and forms



Handlebars

HTML-like DSL

Defined in <script> tags

Helpers

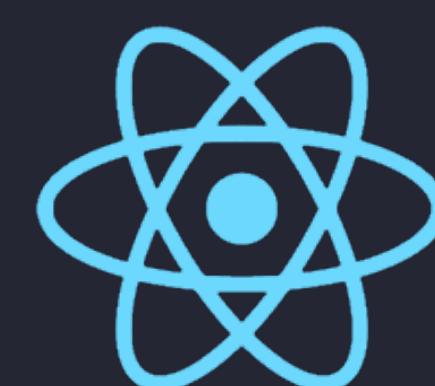


JSX or JS

JSX: Similar to E4X

Namespaced Components

JavaScript Expressions



Templates



```
<body ng-app="todomvc">
  <ng-view />

<script type="text/ng-template" id="todomvc-index.html">
  <section id="todoapp">
    <header id="header">
      <h1>todos</h1>
      <form id="todo-form" ng-submit="addTodo()">
        <input id="new-todo" placeholder="What needs to be done?"
               ng-model="newTodo" ng-disabled="saving" autofocus>
      </form>
    </header>
    <section id="main" ng-show="todos.length" ng-cloak>
      <input id="toggle-all" type="checkbox" ng-model="allChecked"
             ng-click="markAll(allChecked)">
      <label for="toggle-all">Mark all as complete</label>
      <ul id="todo-list">
        <li ng-repeat="todo in todos | filter:statusFilter track by $index"
            ng-class="{completed: todo.completed, editing: todo == editedTodo}">
```

Templates



Helping you **select** an MV* framework

Helping you **select** an MV* framework



```
<script type="text/x-handlebars" data-template-name="todo-list">
  {{#if length}}
    <section id="main">
      {{#if canToggle}}
        {{input type="checkbox" id="toggle-all" checked=allTodos.allAreDone}}
      {{/if}}
      <ul id="todo-list">
        {{#each}}
          <li {{bind-attr class="isCompleted:completed isEditing:editing"}}>
            {{#if isEditing}}
              {{todo-input type="text" class="edit" value=bufferedTitle
                focus-out="doneEditing" insert-newline="doneEditing"
                escape-press="cancelEditing"}}
            {{else}}
              {{input type="checkbox" class="toggle" checked=isCompleted}}
              <label {{action "editTodo" on="doubleClick"}}>{{title}}</label>
              <button {{action "removeTodo"}} class="destroy"></button>
            {{/if}}
          </li>
        {{/each}}
      </ul>
    {{/if}}
  </section>
</script>
```

Templates



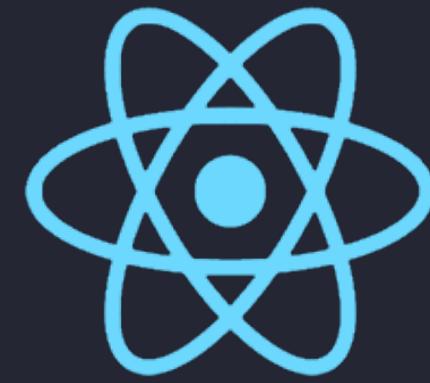
Helping you **select** an MV* framework

Helping you **select** an MV* framework



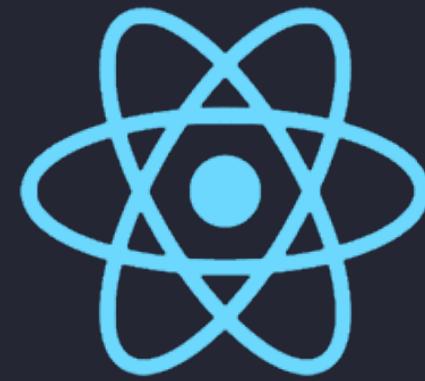
```
<script type="text/x-handlebars" data-template-name="todo-list">
  {{#if length}}
    <section id="main">
      {{#if canToggle}}
        {{input type="checkbox" id="toggle-all" checked=allTodos.allAreDone}}
      {{/if}}
      <ul id="todo-list">
        {{#each}}
          <li {{bind-attr class="isCompleted:completed isEditing:editing"}>
            {{#if isEditing}}
              {{todo-input type="text" class="edit" value=bufferedTitle
                           focus-out="doneEditing" insert-newline="doneEditing"
                           escape-press="cancelEditing"}}
            {{else}}
              {{input type="checkbox" class="toggle" checked=isCompleted}}
              <label {{action "editTodo" on="doubleClick"}}>{{title}}</label>
              <button {{action "removeTodo"}} class="destroy"></button>
            {{/if}}
          </li>
        {{/each}}
  </script>
```

Templates



```
if (todos.length) {
  main = (
    <section className="main">
      <input
        className="toggle-all"
        type="checkbox"
        onChange={this.toggleAll}
        checked={activeTodoCount === 0}>
      />
      <ul className="todo-list">
        {todoItems}
      </ul>
    </section>
  );
}
```

Templates



```
if (todos.length) {
  main = (...);
}

return (
  <div>
    <header className="header">
      <h1>todos</h1>
      <input
        ref="newField"
        className="new-todo"
        placeholder="What needs to be done?"
        onKeyDown={this.handleNewTodoKeyDown}
        autoFocus={true}
      />
    </header>
    {main}
    {footer}
  </div>
);
```

i18n

i18n/l10n for date,
number and
currency filters

angular-translate



No built-in i18n

i18n-js

ember-i18n



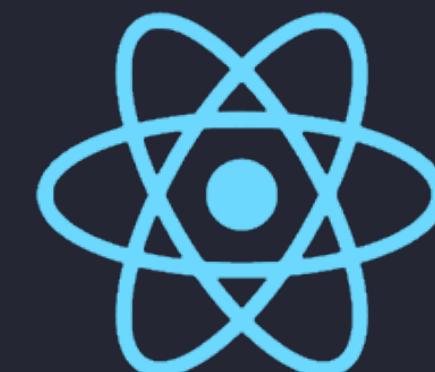
No built-in i18n

react-intl

Format.js



Components,
MixIns, Relative
Times



i18n

i18n/l10n for date,
number and
currency filters

angular-translate

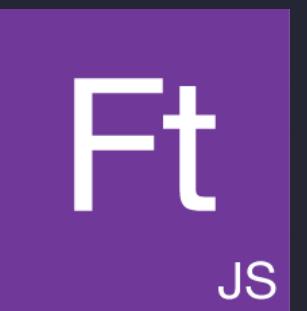


No built-in i18n

i18n-js

ember-i18n

ember-intl



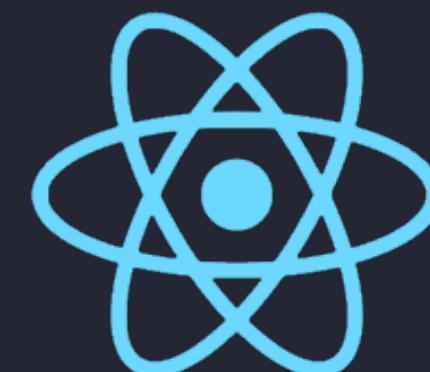
No built-in i18n

react-intl

Format.js



Components,
MixIns, Relative
Times



Form Binding and Validation

2-way data-binding
with **ngModel**

Validation services
provided

CSS classes for
invalid, required,
touched, dirty, etc.



1 or 2-way binding

DS.Errors

No UI Validation

ember-validations

Ember EasyForm

Ember Forms

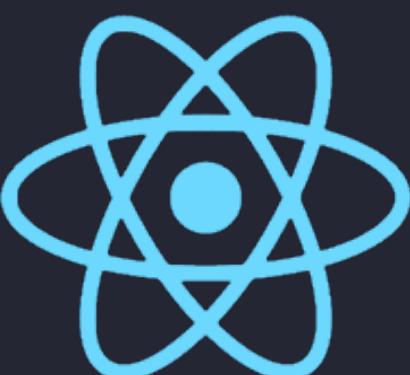


Form Components
and Events

propTypes in dev

react-validation-mixin

formsy-react



LOC in examples/angularjs



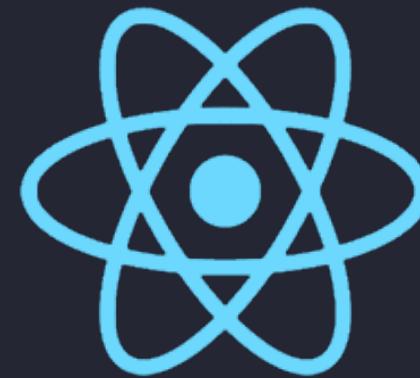
Language	files	blank	comment	code
Javascript	11	2715	15432	12005
CSS	2	75	5	439
HTML	1	1	0	74
JSON	1	0	0	16
SUM:	15	2791	15437	12534

LOC in examples/emberjs



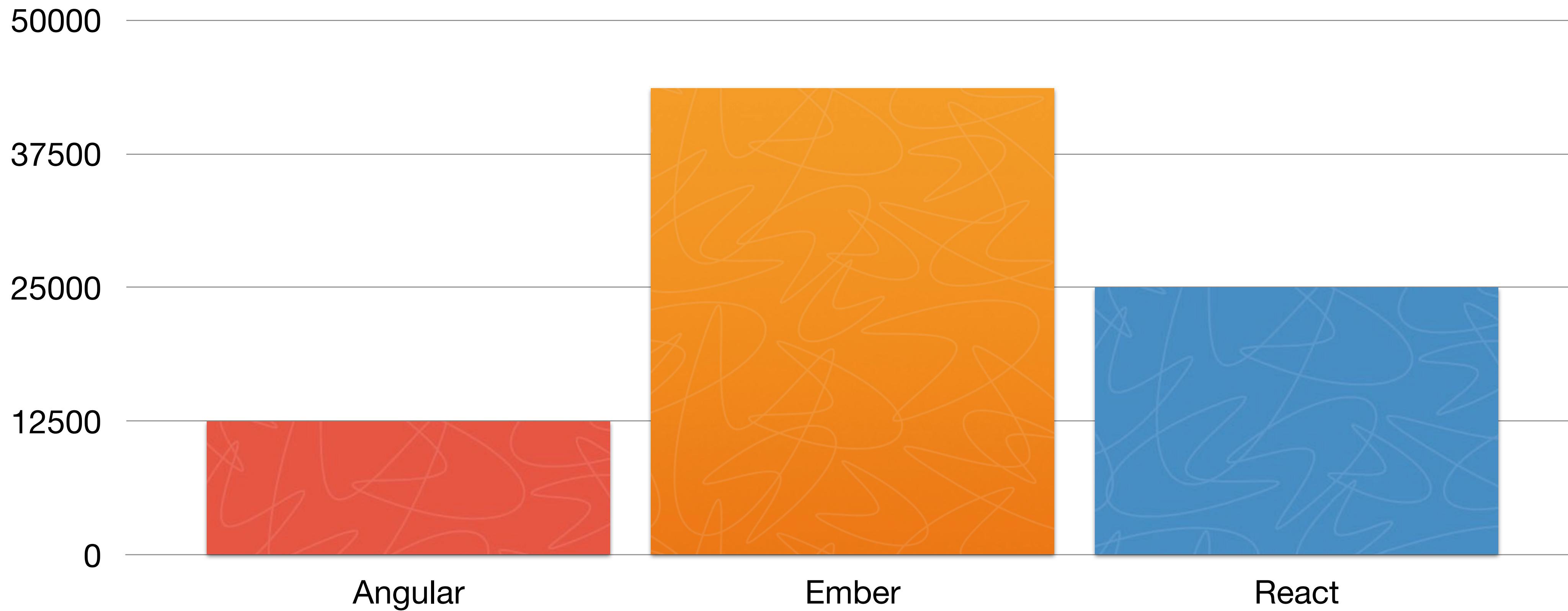
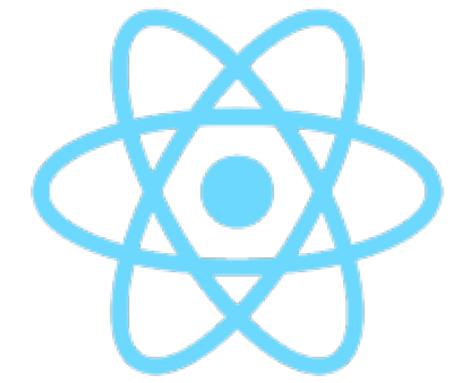
Language	files	blank	comment	code
Javascript	14	14574	26513	43031
CSS	2	75	5	439
HTML	1	0	0	88
JSON	1	0	0	11
SUM:	18	14649	26518	43569

LOC in examples/react

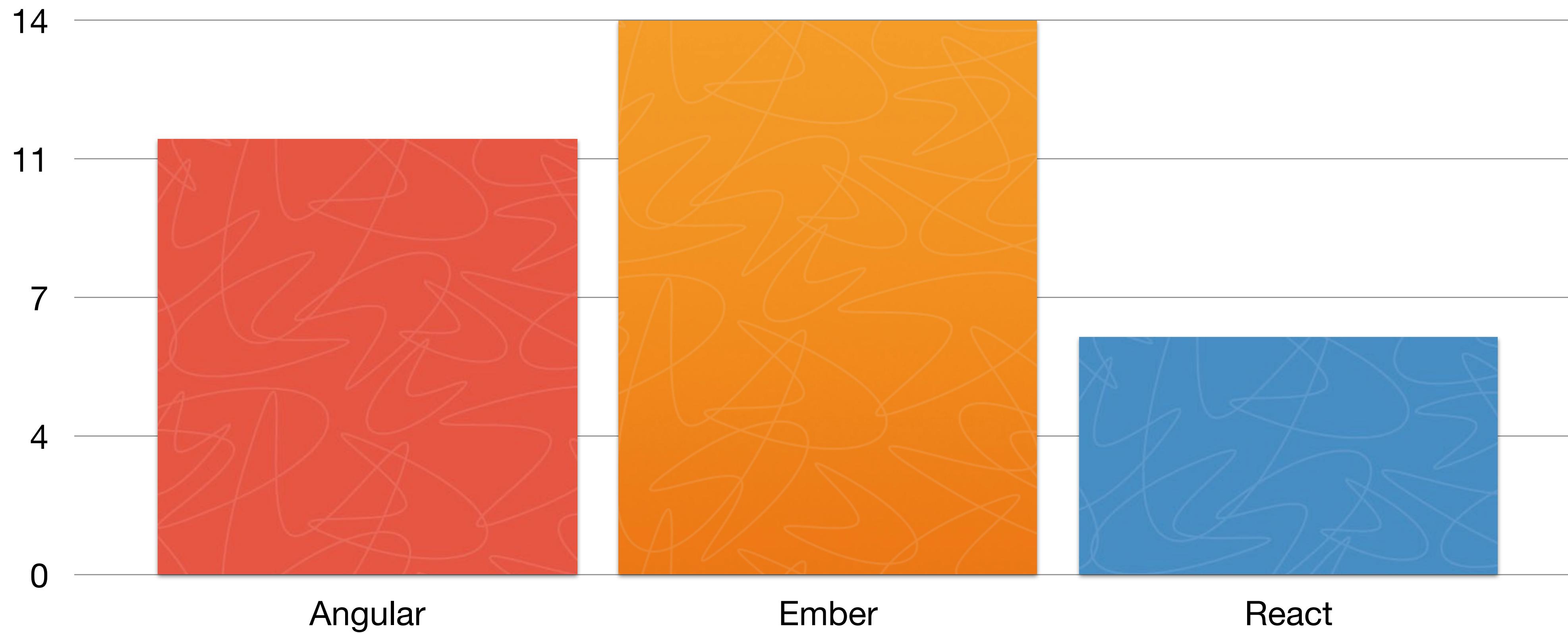
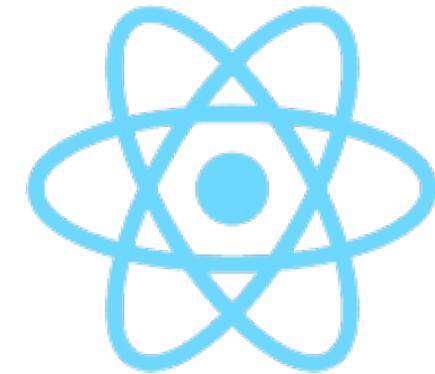


Language	files	blank	comment	code
Javascript	6	4783	9359	24530
CSS	2	75	5	439
HTML	1	2	3	26
JSON	1	0	0	9
SUM:	10	4860	9367	25004

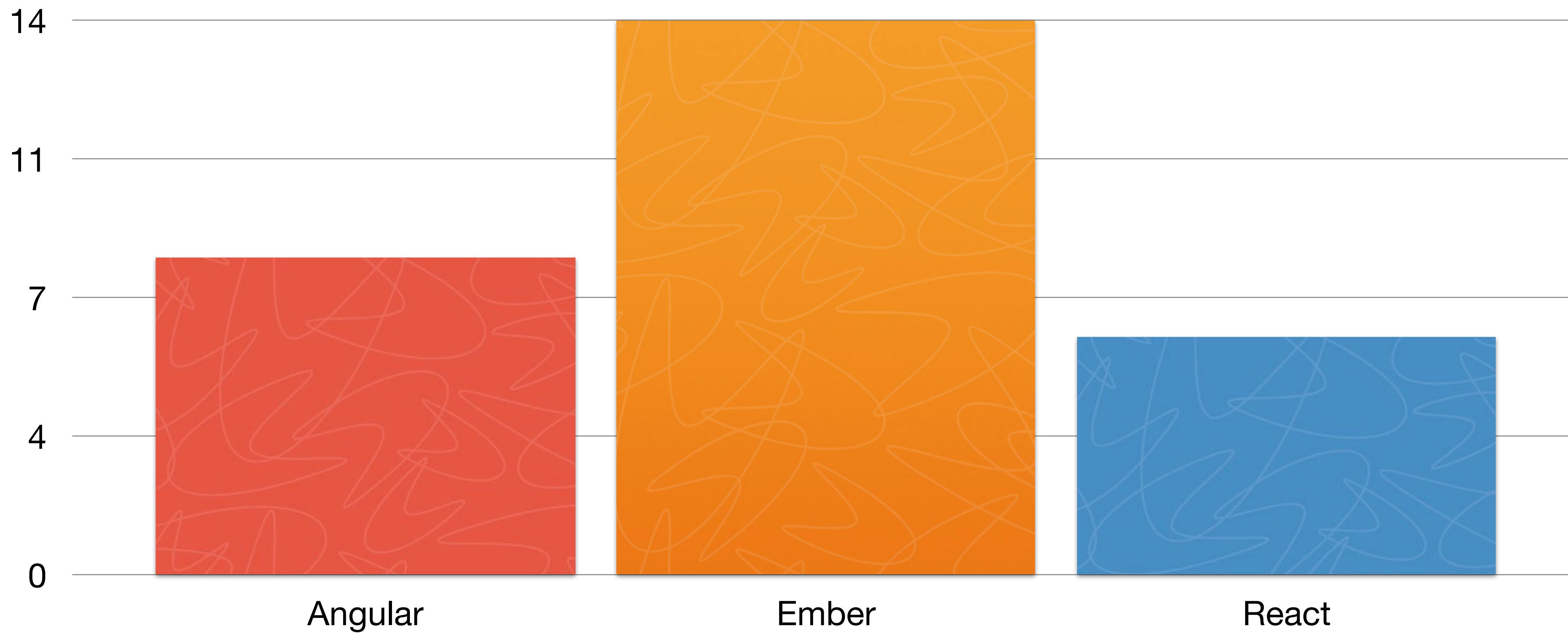
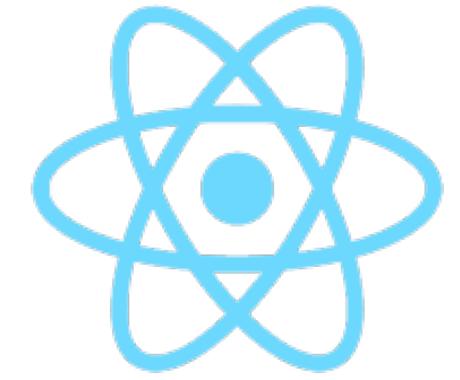
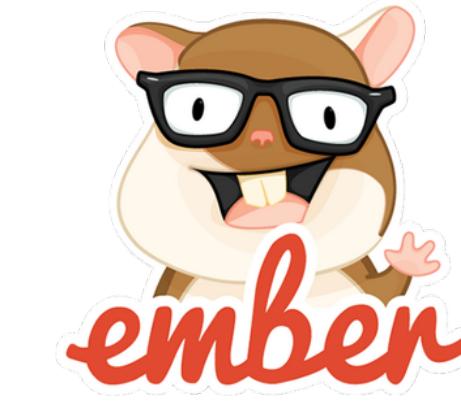
LOC in TodoMVC



Files in TodoMVC



Files in TodoMVC



Starter Kits

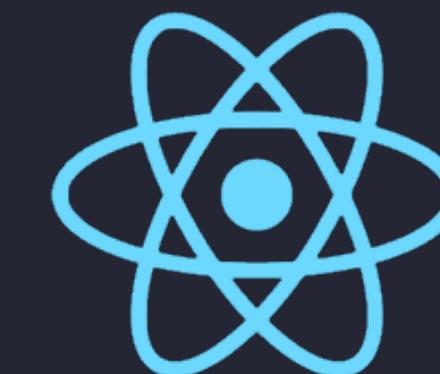
150K



77K

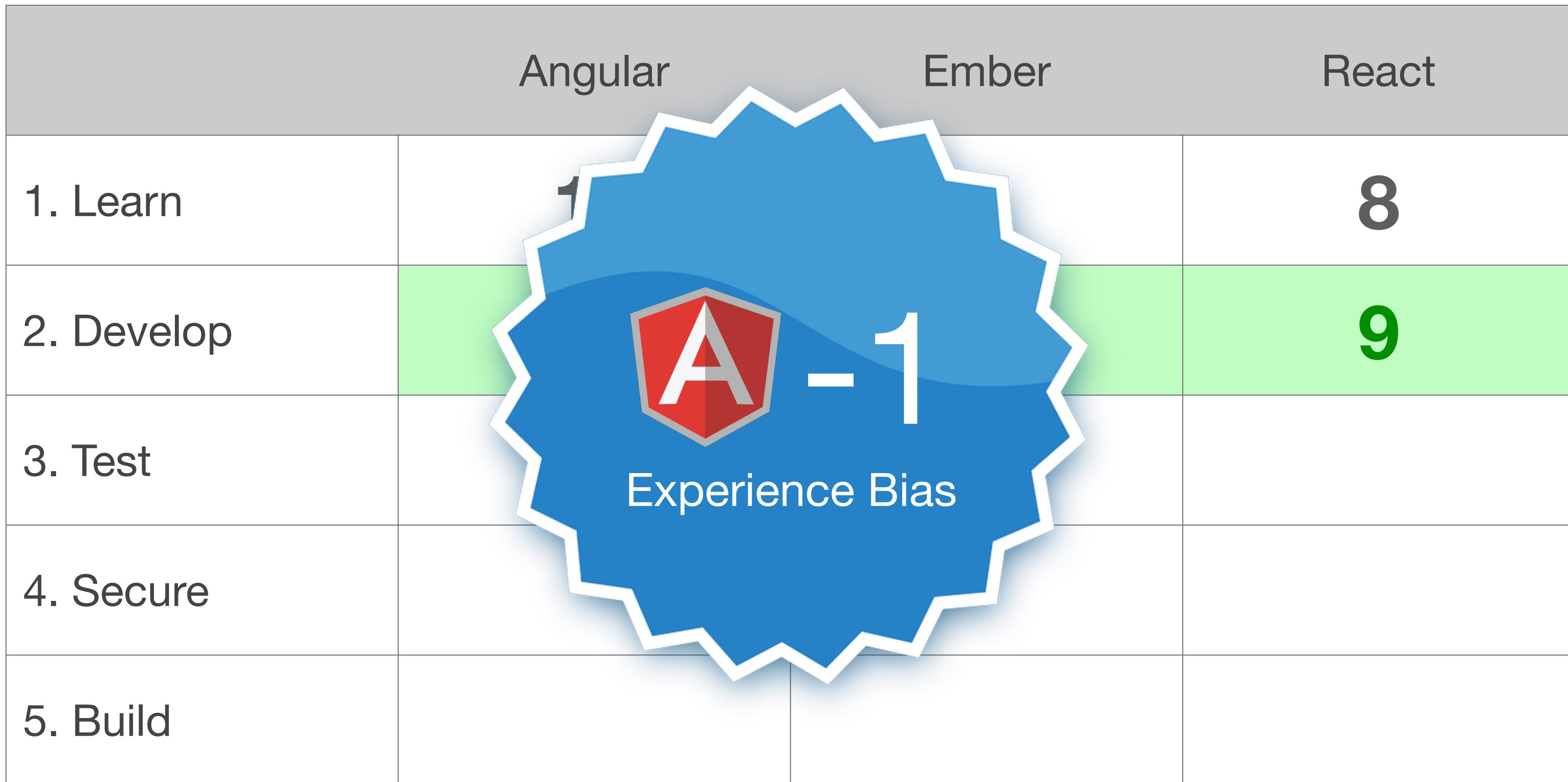


225K



	Angular	Ember	React
1. Learn	10	6	8
2. Develop	9	6	5
3. Test			
4. Secure			
5. Build			

	Angular	Ember	React
1. Learn	10	6	8
2. Develop	10	8	9
3. Test			
4. Secure			
5. Build			



	Angular	Ember	React
1. Learn	10	6	8
2. Develop	9	8	9
3. Test			
4. Secure			
5. Build			

Testing

Karma

Jasmine

angular-mocks

Protractor

QUnit

Testem

ember test

Acceptance Tests

ember generate
acceptance-test
<name>



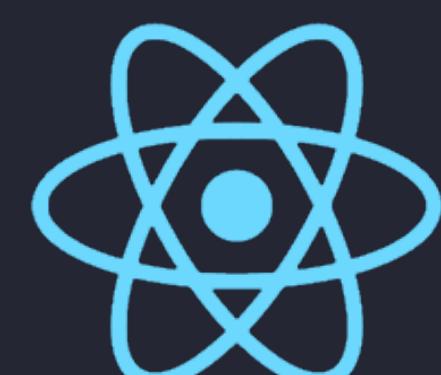
R.addons.TestTools

Jest

Shallow Rendering

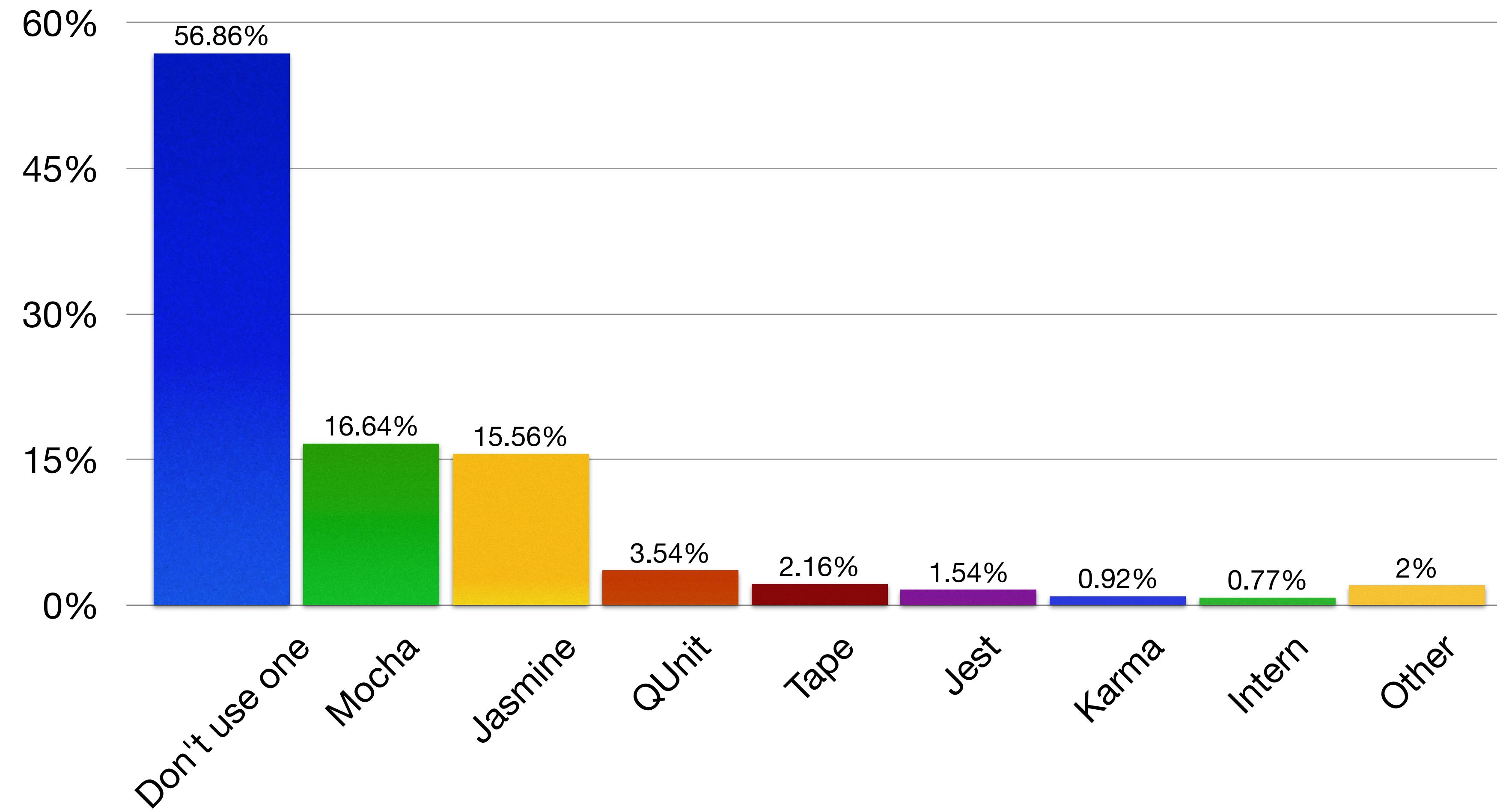
Mocha + Chai +
JSDOM

react-testing



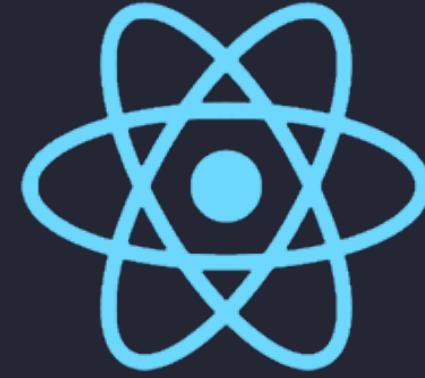
	Angular	Ember	React
1. Learn	10	6	8
2. Develop	9	8	9
3. Test	8	9	8
4. Secure			
5. Build			

What tool do you use to test JavaScript?



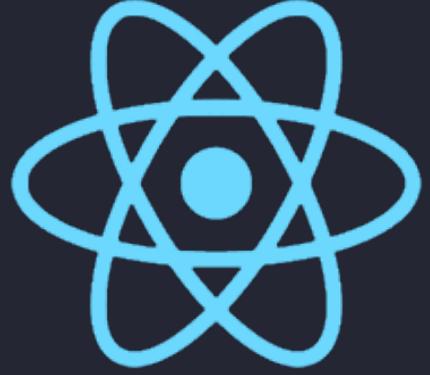
	Angular	Ember	React
1. Learn	10	6	8
2. Develop	9	8	9
3. Test	8	9	8
4. Secure			
5. Build			

Security



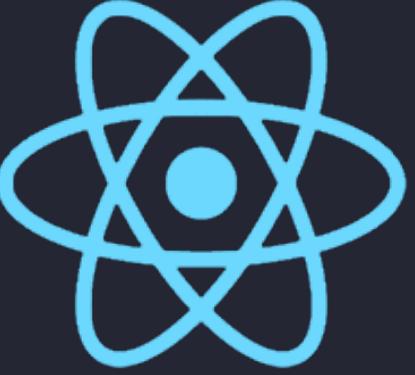
CSRF	✓
XSS	✓
CSP	⚠
Auth	✓
Advisories	✗

Security



	Angular	Ember	React
CSRF	✓	✗	
XSS	✓	✓	
CSP	⚠	✓	
Auth	✓	✓	
Advisories	✗	✓	

Security



	Angular	Ember	React
CSRF	✓	✗	✗
XSS	✓	✓	⚠
CSP	⚠	✓	⚠
Auth	✓	✓	✓
Advisories	✗	✓	✗



Node Security Project

SECURITY is a core value
of the NODE community

Node Advisories

2015		
uglify-js incorrectly handles non-boolean comparisons during minification Mon Aug 24 2015 12:29:10 GMT-0700 (PDT)	uglify-js Vulnerable: <code><= 2.4.23</code> Patched: <code>>= 2.4.24</code>	
Geddy Directory Traversal Mon Jul 27 2015 15:33:48 GMT-0800 (PST)	geddy Vulnerable: <code><13.0.8</code> Patched: <code>>=13.0.8</code>	
CSRF Vulnerability in jquery-ujs Wed Jun 24 2015 09:00:00 GMT+0000 (UTC)	jquery-ujs Vulnerable: <code><= 1.0.3</code> Patched: <code>>= 1.0.4</code>	
jsonwebtoken Verification Bypass Tue Mar 31 2015 19:00:00 GMT-0700 (PDT)	jsonwebtoken Vulnerable: <code><4.2.2</code> Patched: <code>>=4.2.2</code>	
marked regular expression denial of service Thur Jan 22 2015 09:33:48 GMT-0800 (PST)	marked Vulnerable: <code><=0.3.3</code> Patched: <code>>=0.3.4</code>	
marked VBScript Content Injection Thur Jan 22 2015 09:33:48 GMT-0800 (PST)	marked Vulnerable: <code><=0.3.2</code> Patched: <code>>=0.3.3</code>	

	Angular	Ember	React
1. Learn	10	6	8
2. Develop	9	8	9
3. Test	8	9	8
4. Secure	7	8	4
5. Build			



mustache-security

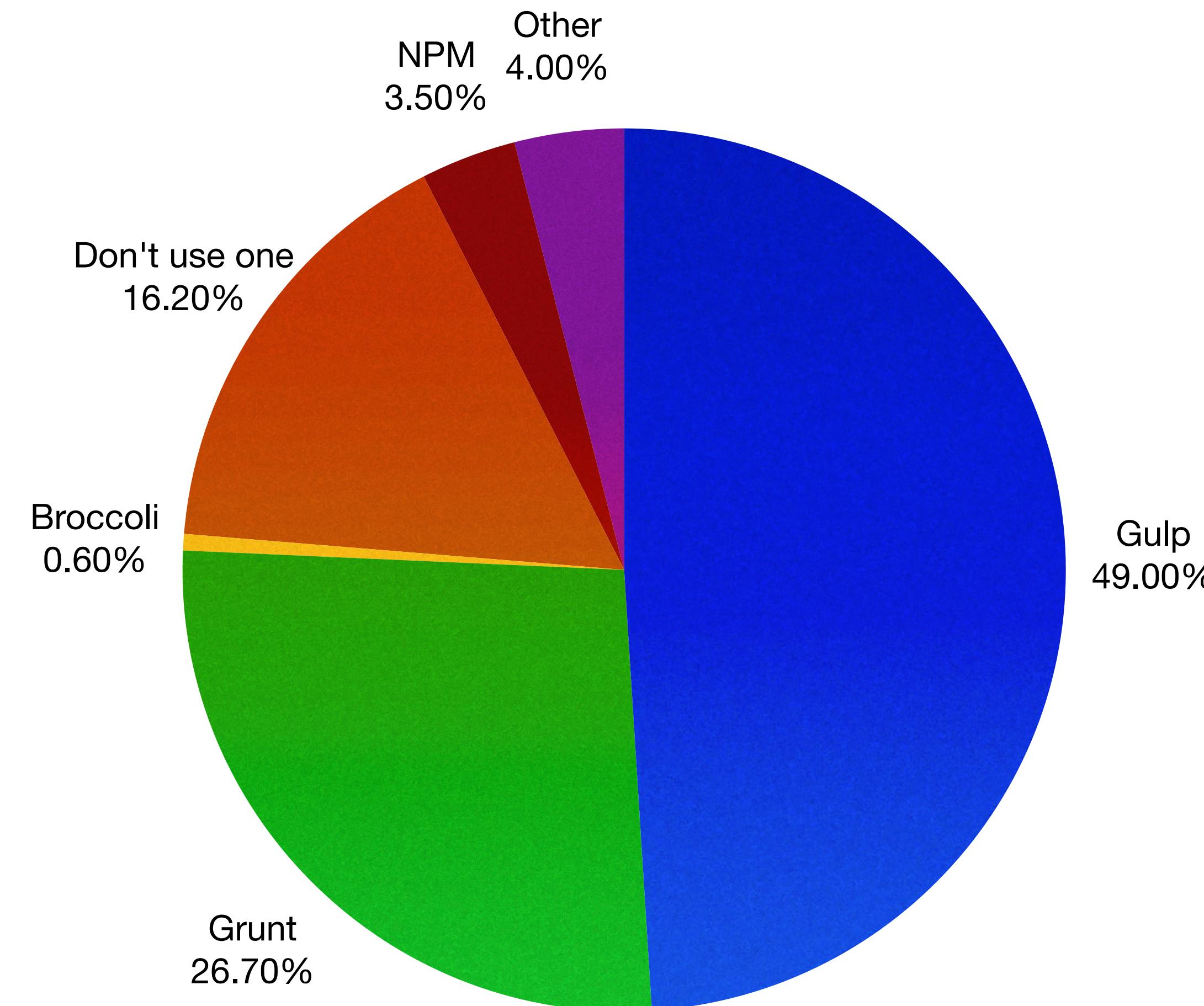
A wiki dedicated to JavaScript MVC security pitfalls

<https://code.google.com/p/mustache-security/>

Framework	{SEC-A}	{SEC-B}	{SEC-C}	{SEC-D}	{SEC-E}	{SEC-F}
VueJS	Fail	Fail	Fail	Fail	Fail	Fail
AngularJS 1.0.8	Fail	Fail	Fail	Fail	PASS	Fail
AngularJS 1.2.0	Fail	PASS	Fail	Fail	PASS	PASS
AngularJS 1.4.0	Fail	PASS	Fail	PASS	PASS	PASS
CanJS	Fail	Fail	PASS	Fail	Fail	Fail
Underscore.js	Fail	Fail	PASS	Fail	Fail	Fail
KnockoutJS	Fail	Fail	Fail	Fail	Fail	Fail
Ember.js	Fail	PASS	PASS	Fail	PASS	TBD
Polymer	TBD	TBD	TBD	TBD	TBD	TBD
Ractive.js 0.4.0	Fail	Fail	Fail	Fail	Fail	Fail
Ractive.js 0.7.2	Fail	Fail	PASS	Fail	Fail	Fail
jQuery	TBD	TBD	TBD	TBD	PASS	TBD
JsRender	Fail	Fail	Fail	Fail	Fail	Fail
Kendo UI	Fail	Fail	Fail	Fail	Fail	Fail

	Angular	Ember	React
1. Learn	10	6	8
2. Develop	9	8	9
3. Test	8	9	8
4. Secure	7	8	4
5. Build			

What task runner do you prefer using?



Building

Grunt / Gulp

Browserify

Webpack

Bower and NPM



Ember CLI

Broccoli-powered
asset pipeline

Addon System

Bower and NPM

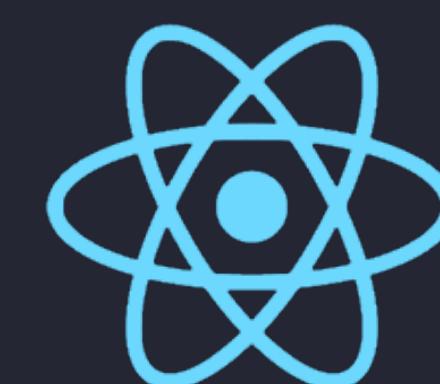


React Starter Kit

react-tools

Browserify

Webpack



Building

Grunt / Gulp

Browserify

Webpack

Bower and NPM



Ember CLI

Broccoli-powered
asset pipeline

Addon System

Bower and NPM



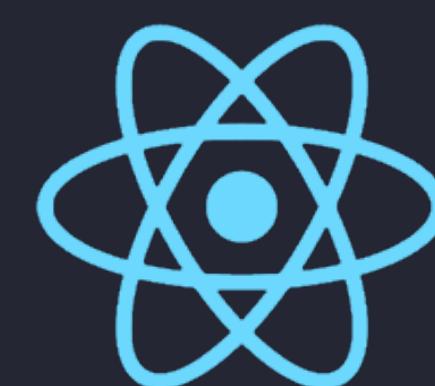
React Starter Kit

~~react-tools~~

Babel

Browserify

Webpack



	Angular	Ember	React
1. Learn	10	6	8
2. Develop	9	8	9
3. Test	8	9	8
4. Secure	7	8	4
5. Build	9	10	9

	Angular	Ember	React
6. Deploy			
7. Debug			
8. Scale			
9. Maintain			
10. Share			

	Angular	Ember	React
6. Deploy	10	10	10
7. Debug			
8. Scale			
9. Maintain			
10. Share			

	Angular	Ember	React
6. Deploy	10	10	10
7. Debug			
8. Scale			
9. Maintain			
10. Share			

Debugging

Debug from the
DOM

\$log service

Chrome and
debugger

Angular Batarang



LOG_TRANSITIONS

LOG_*

Handlebar Helpers

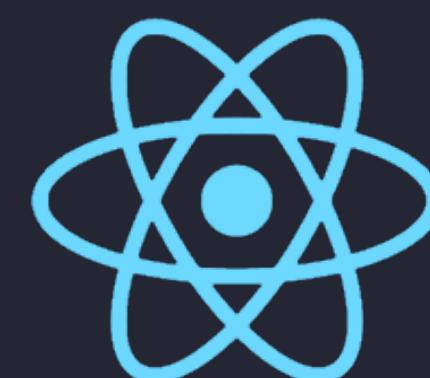
Ember Inspector



React Dev Tools

Pretty Diff

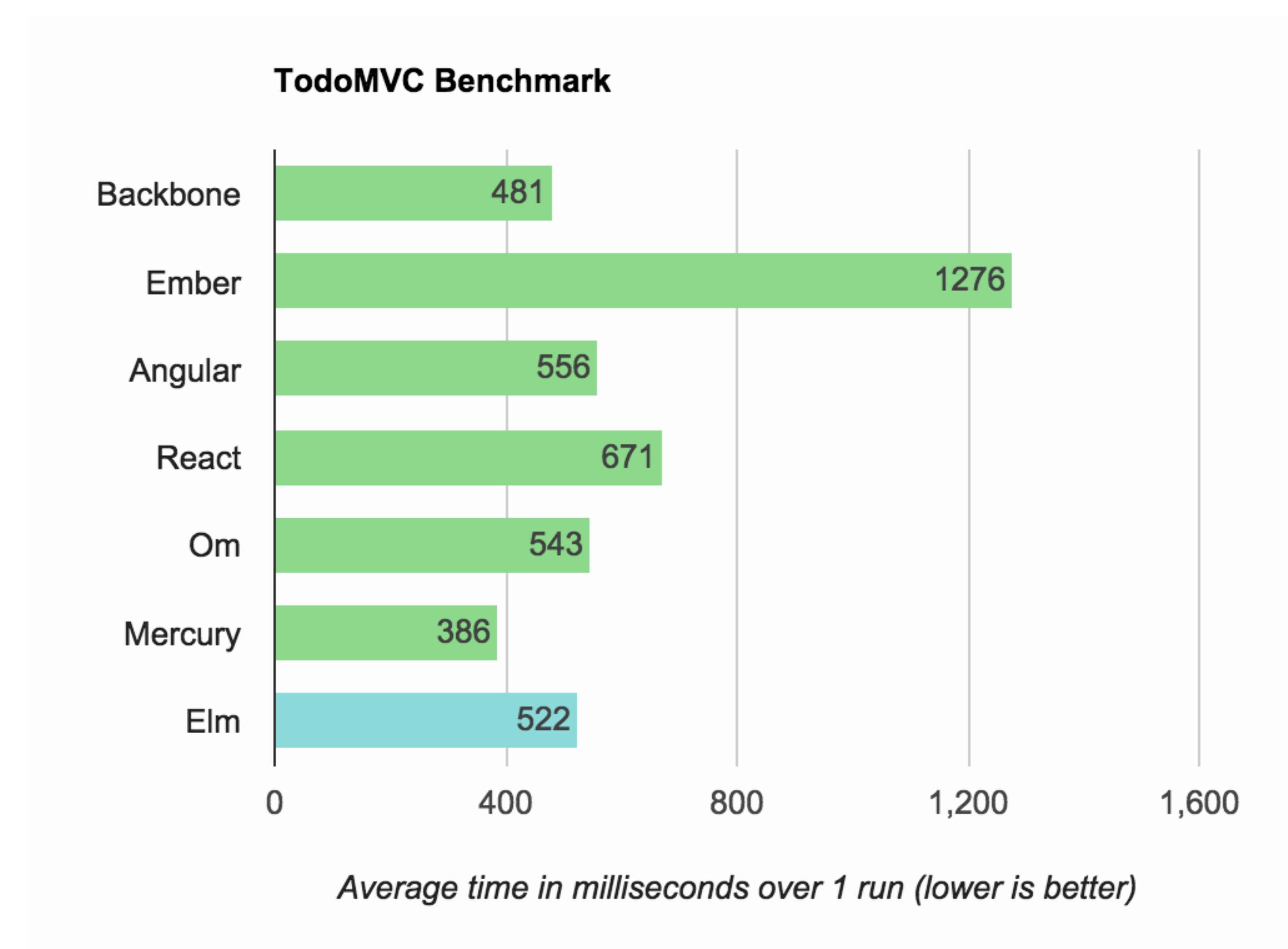
react-debug



	Angular	Ember	React
6. Deploy	10	10	10
7. Debug	7	10	7
8. Scale			
9. Maintain			
10. Share			

	Angular	Ember	React
6. Deploy	10	10	10
7. Debug	7	10	7
8. Scale			
9. Maintain			
10. Share			

TodoMVC Benchmark



	Angular	Ember	React
6. Deploy	10	10	10
7. Debug	7	10	7
8. Scale	9	7	9
9. Maintain			
10. Share			

apple.com/music

The screenshot shows the Apple Music website interface. At the top, there's a navigation bar with icons for volume, play/pause, and other controls, followed by the Apple logo and user account information. Below the navigation is a search bar labeled "Search". The main header features a large image of Taylor Swift wearing sunglasses, with the text "1989" and "Taylor Swift" next to it. A subtitle below the image reads "1989 is as brash and brilliant as the lights of Times Square." To the left of the main image is a small thumbnail for "T.S. 1989". On the right, there's a snippet of another album cover. Below the main header, there's a dropdown menu labeled "All Genres".

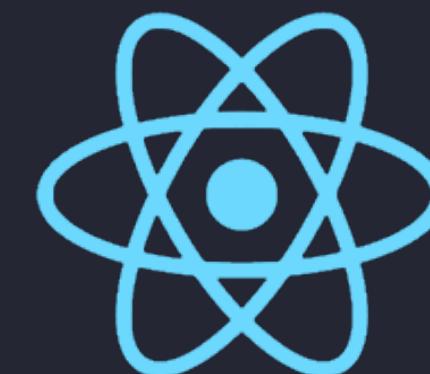
Hot Tracks >

Image	Artist / Song	Rating	Artist / Song	Rating	
	Freedom Pharrell Williams	E		Cool for the Summer Demi Lovato	E
	Cut the Cord Shinedown	E		Nothing But Trouble (From the Soundtrack ... Lil Wayne & Charlie Puth	E
	Finna Get Loose (feat. Pharrell Williams) Puff Daddy & The Family	E			

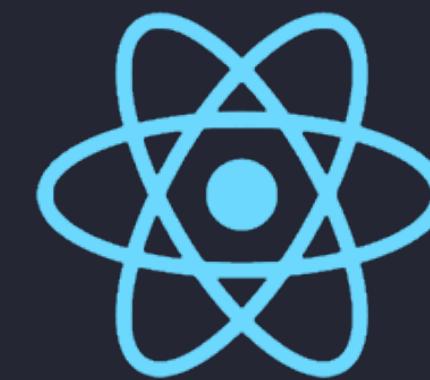
Playlists >

- Apple Editors Playlists**
The music we're obsessing over, plus handcrafted playlists.
- Activities Playlists**
The soundtrack to your next move, from working out to chilling out.
- Curators Playlists**
What Rolling Stone, Pitchfork, and more are listening to.

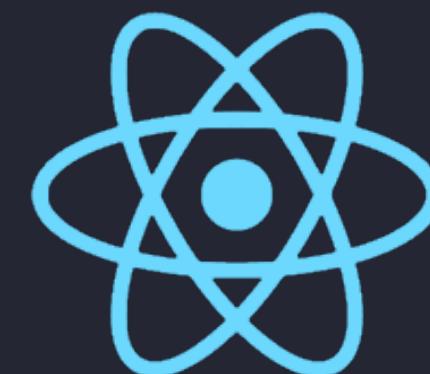
What about Isomorphic JavaScript?



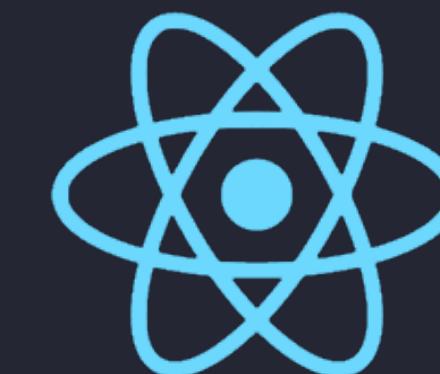
What about ~~Isomorphic~~ JavaScript?



What about Universal JavaScript?



Server-Side Rendering Support



	Angular	Ember	React
6. Deploy	10	10	10
7. Debug	7	10	7
8. Scale	9	7	9
9. Maintain			
10. Share			

	Angular	Ember	React
6. Deploy	10	10	10
7. Debug	7	10	7
8. Scale	9-1	7	9
9. Maintain			
10. Share			

	Angular	Ember	React
6. Deploy	10	10	10
7. Debug	7	10	7
8. Scale	8	7+1	9
9. Maintain			
10. Share			

	Angular	Ember	React
6. Deploy	10	10	10
7. Debug	7	10	7
8. Scale	8	8	9+2
9. Maintain			
10. Share			

	Angular	Ember	React
6. Deploy	10	10	10
7. Debug	7	10	7
8. Scale	8	8	10
9. Maintain			
10. Share			

	Angular	Ember	React
6. Deploy	10	10	10
7. Debug	7	10	7
8. Scale	8	8	10
9. Maintain			
10. Share			



```
var foo = ["10", "10", "10"];
foo.map(parseInt);
// Returns [ 10, NaN, 2 ]
```

```
[] + [] // ""
[] + {} // {}
{} + [] // 0
{} + {} // NaN
```

```
var a = {};
a[][] = 2;
alert(a[""]); // alerts 2
```

```
alert(Array(16).join("wat" - 1) + " Batman!");
```

<https://www.destroyallsoftware.com/talks/wat>

	Angular	Ember	React
6. Deploy	10	10	10
7. Debug	7	10	7
8. Scale	9	7	10
9. Maintain	3	5	4
10. Share			

JS Error Monitoring

Track:js

New Relic Browser

Raygun

Bugsnag

JS Monitor

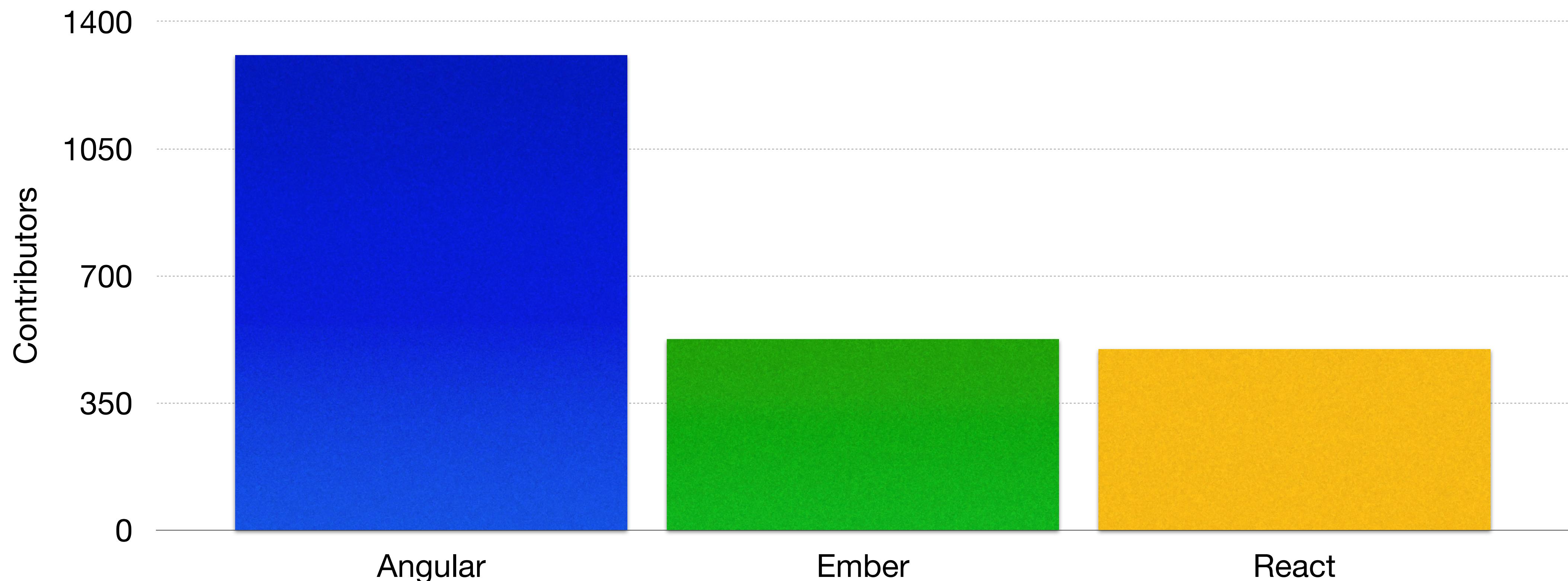
Qbaka



	Angular	Ember	React
6. Deploy	10	10	10
7. Debug	7	10	7
8. Scale	9	7	10
9. Maintain	3	5	4
10. Share			

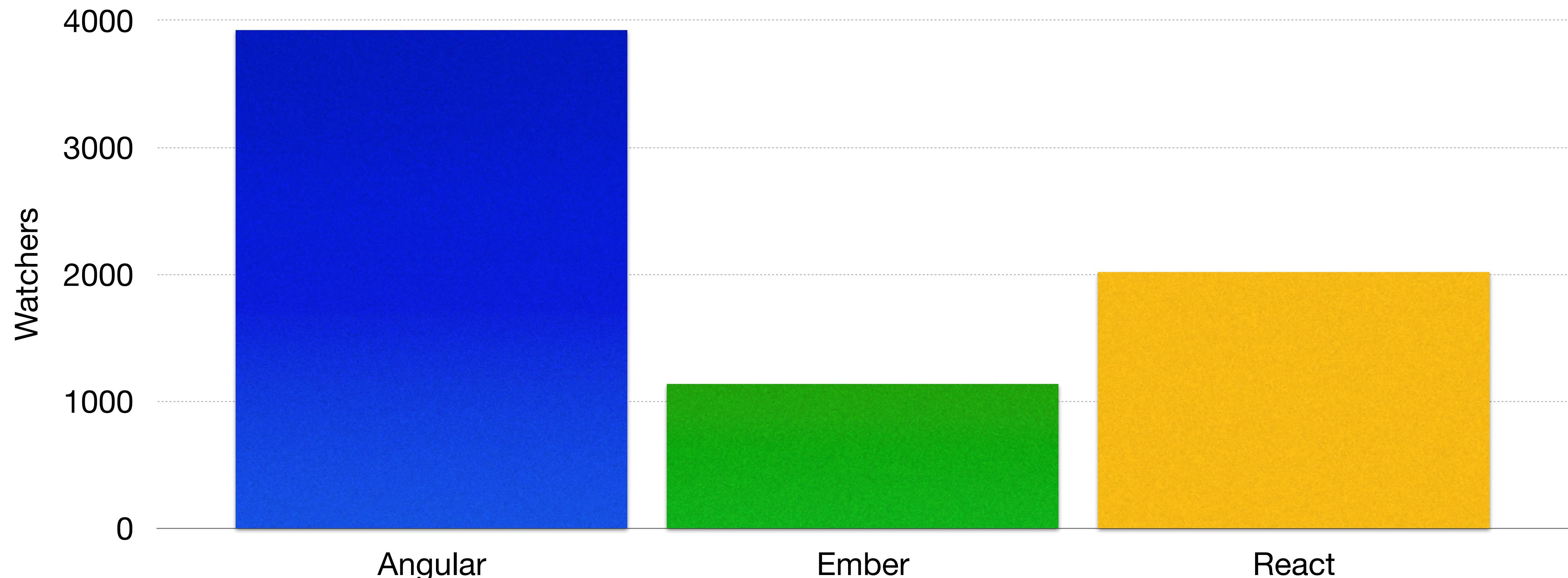
GitHub: Contributors

September 9, 2015



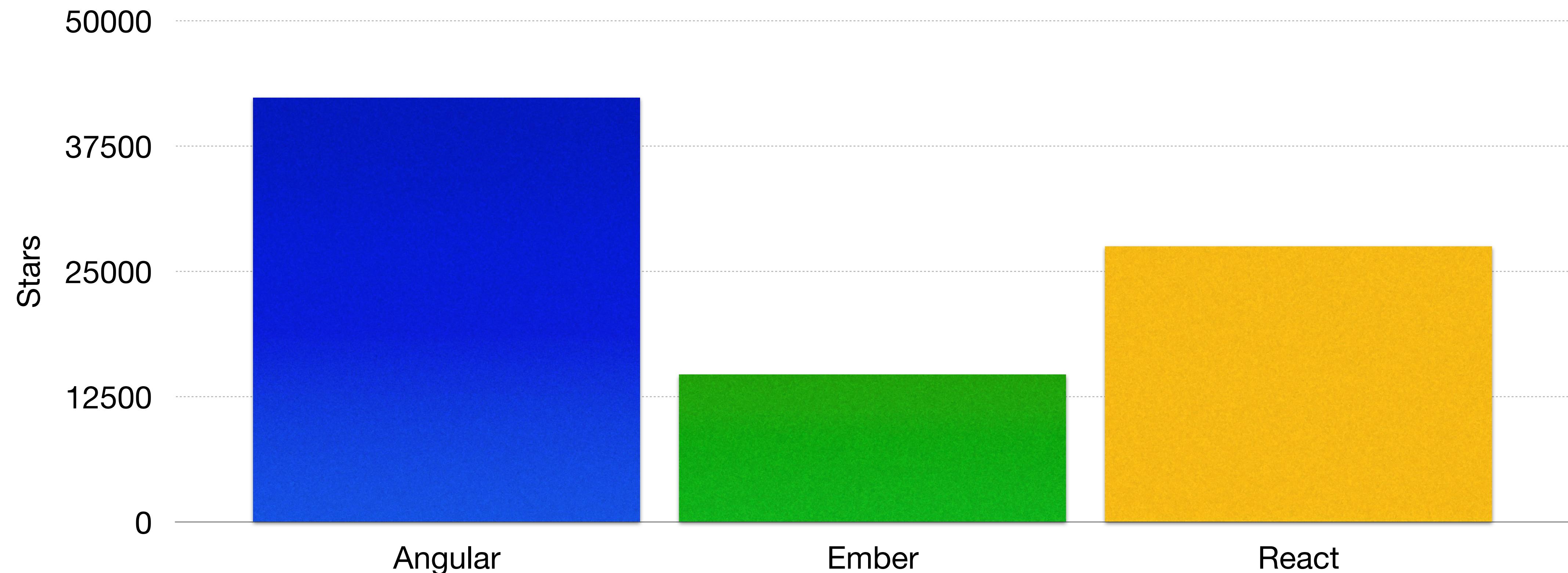
GitHub: Watchers

September 9, 2015



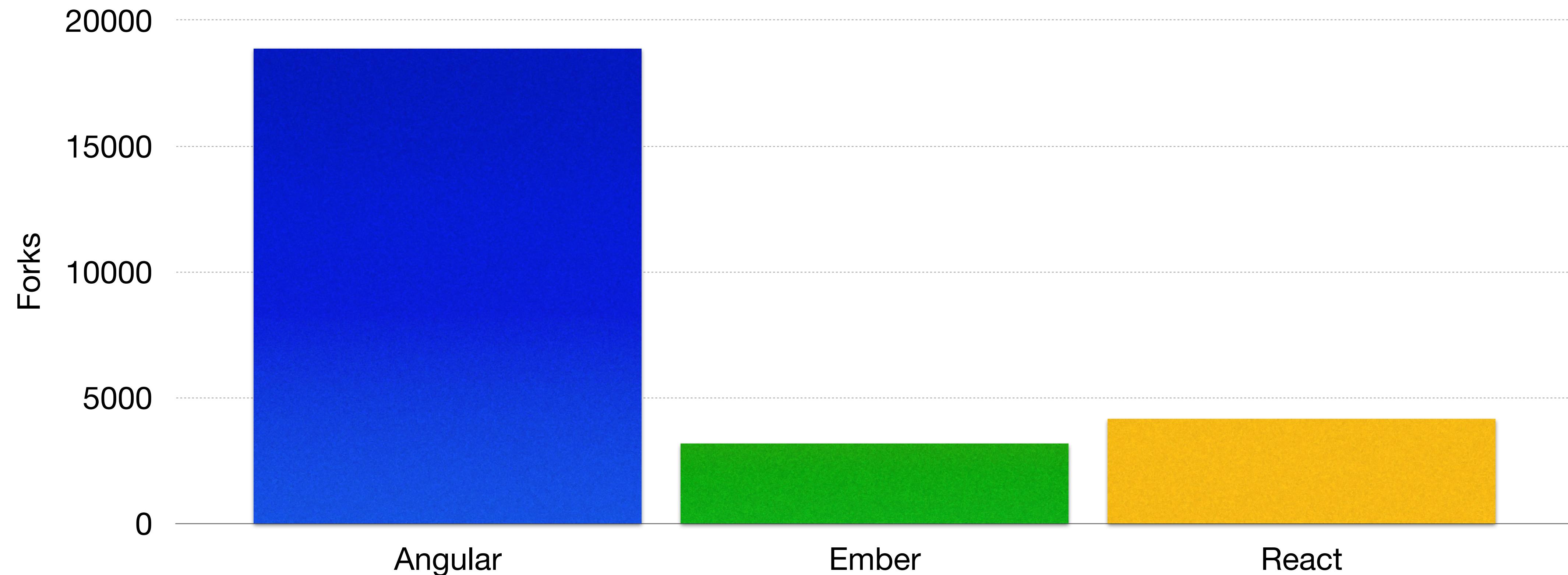
GitHub: Stars

September 9, 2015



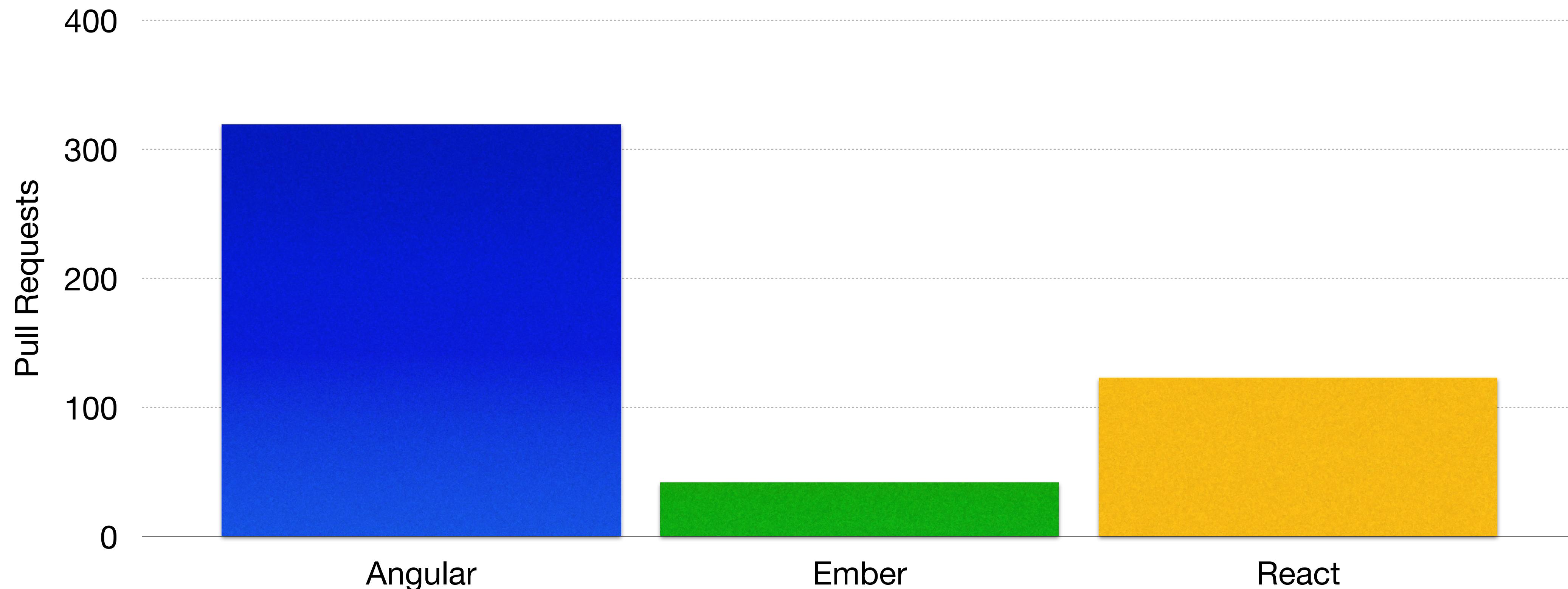
GitHub: Forks

September 9, 2015



GitHub: Pull Requests

September 9, 2015



Stack Overflow

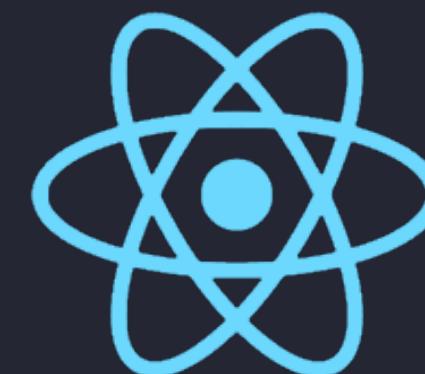
119K



16.5K



5K

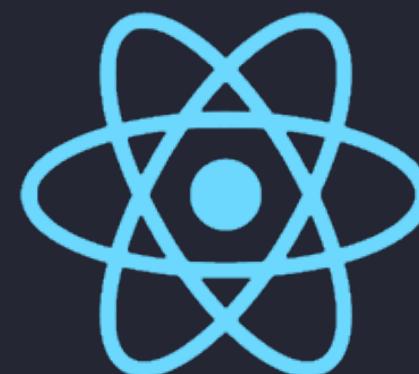


Google Group: Members

18K



2K



Addons, according to npmjs.org

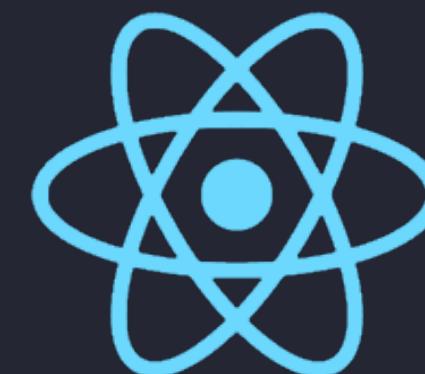
4085



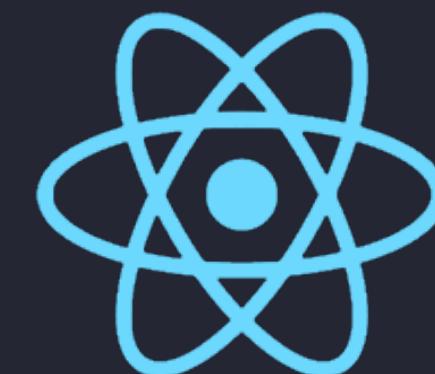
2249



5299



Commercial Support



Who's Using in Production

madewithangular

Google Trends
and Analytics

Amazon

Forbes

MSNBC



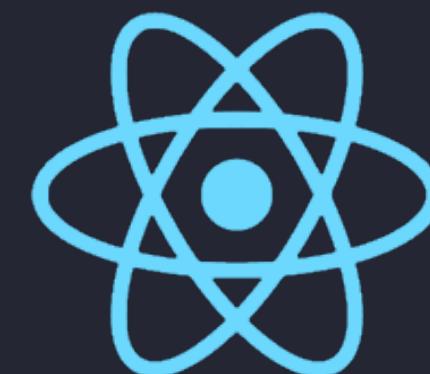
builtwithember.io

Apple Music

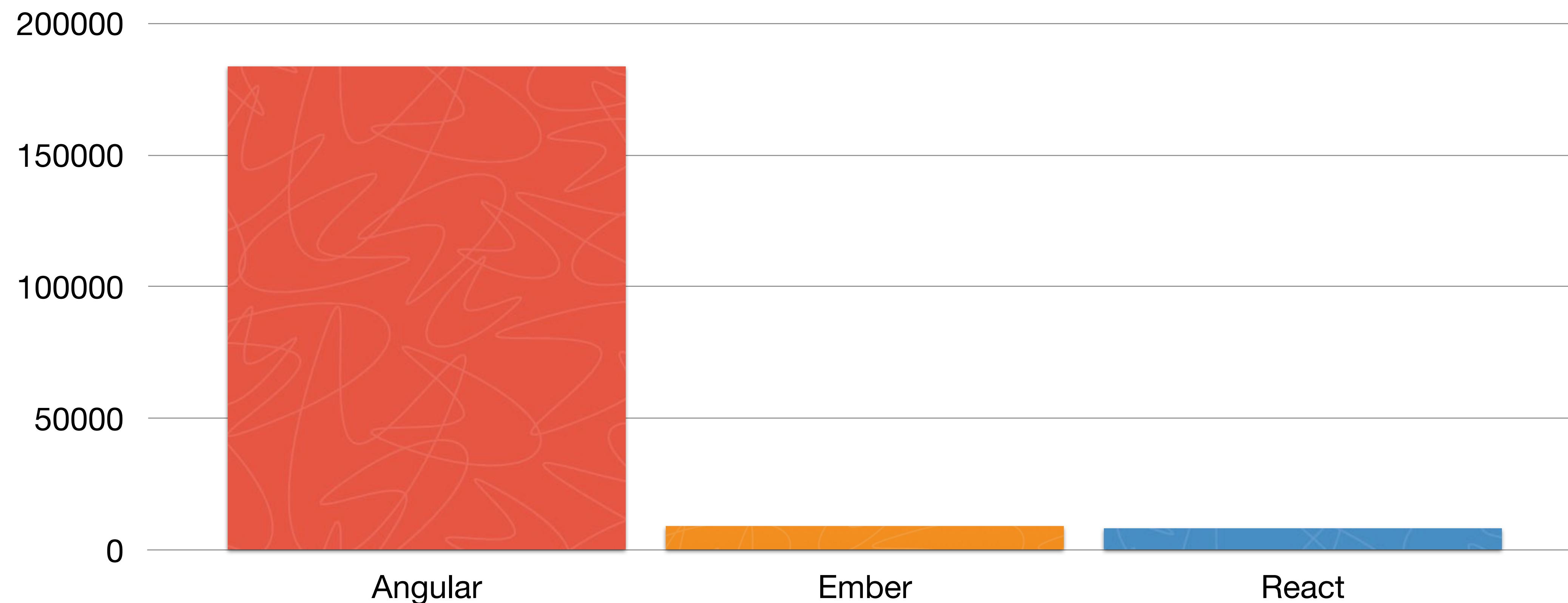
emberjs.com/ember-
users

Yahoo

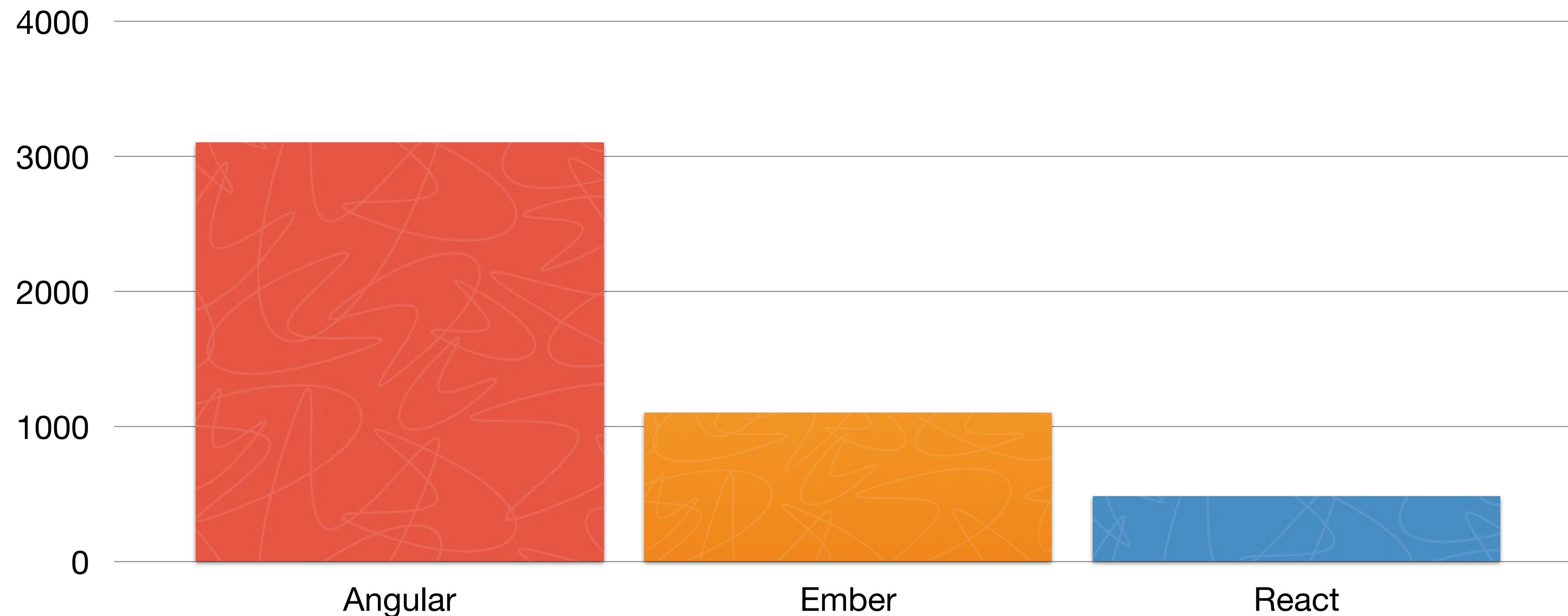
Square



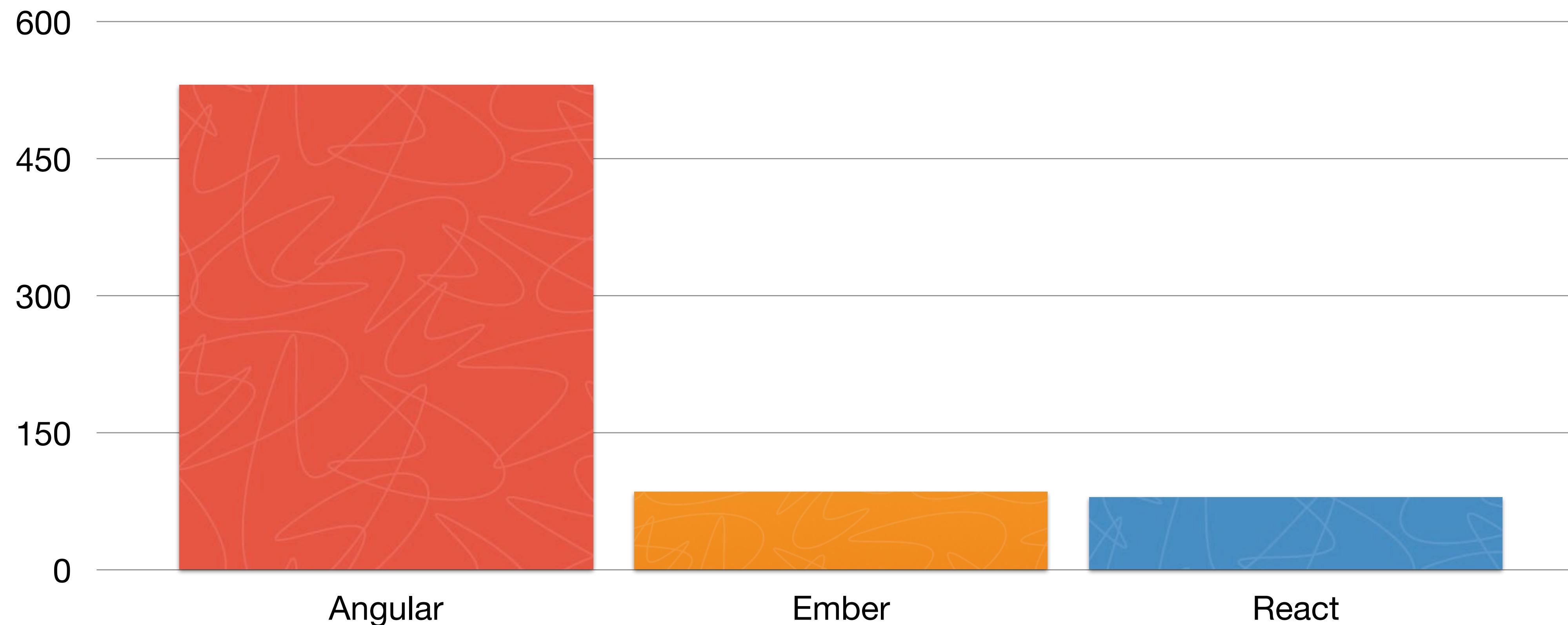
Candidates: LinkedIn



Jobs: LinkedIn



Jobs: Career Builder



Indeed Trends

angularjs, ember.js, react.js Job Trends



	Angular	Ember	React
6. Deploy	10	10	10
7. Debug	7	10	7
8. Scale	9	7	10
9. Maintain	3	5	4
10. Share	10	10	10

The Framework Scorecard

	Angular	Ember	React
1. Learn	10	6	8
2. Develop	9	8	9
3. Test	8	9	8
4. Secure	7	8	4
5. Build	9	10	9
6. Deploy	10	10	10
7. Debug	7	10	7
8. Scale	9	7	10
9. Maintain	3	5	4
10. Share	10	10	10

The Framework Scorecard

	Angular	Ember	React
1. Learn	10	6	8
2. Develop	9	8	9
3. Test	8	9	8
4. Secure	7	8	4
5. Build	9	10	9
6. Deploy	10	10	10
7. Debug	7	10	7
8. Scale	9	7	10
9. Maintain	3	5	4
10. Share	10	10	10
Total	82	83	79

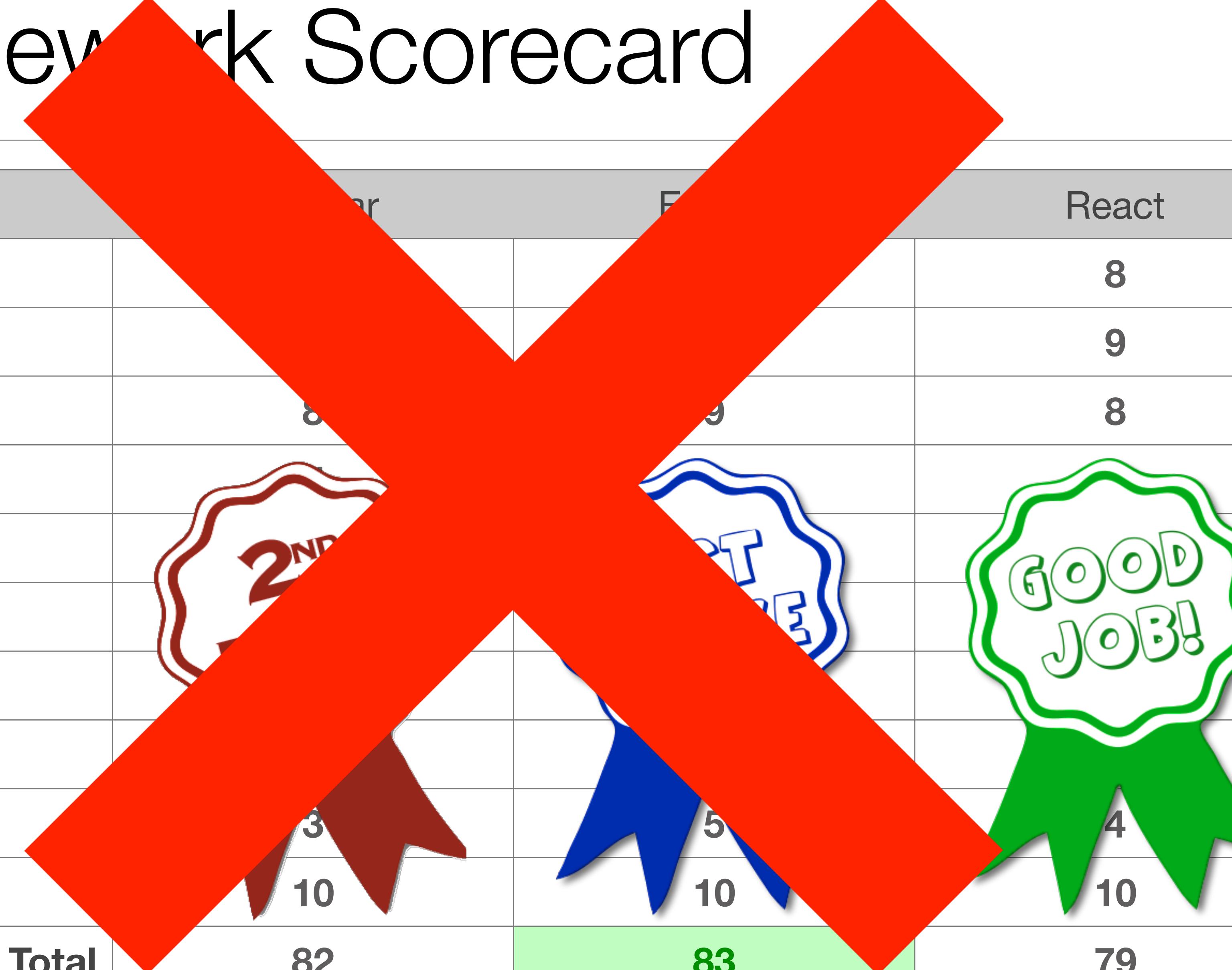
The Framework Scorecard

	Angular	Ember	React
1. Learn	10	6	8
2. Develop	9	8	9
3. Test	8	9	8
4. Secure			
5. Build			
6. Deploy			
7. Debug			
8. Scale			
9. Maintain			
10. Share			
Total	82	83	79



The Framework Scorecard

	Angular	React
1. Learn	8	8
2. Develop	9	9
3. Test	8	8
4. Secure	8	9
5. Build	9	9
6. Deploy	8	8
7. Debug	9	9
8. Scale	8	8
9. Maintain	9	9
10. Share	10	10
Total	82	83
		79

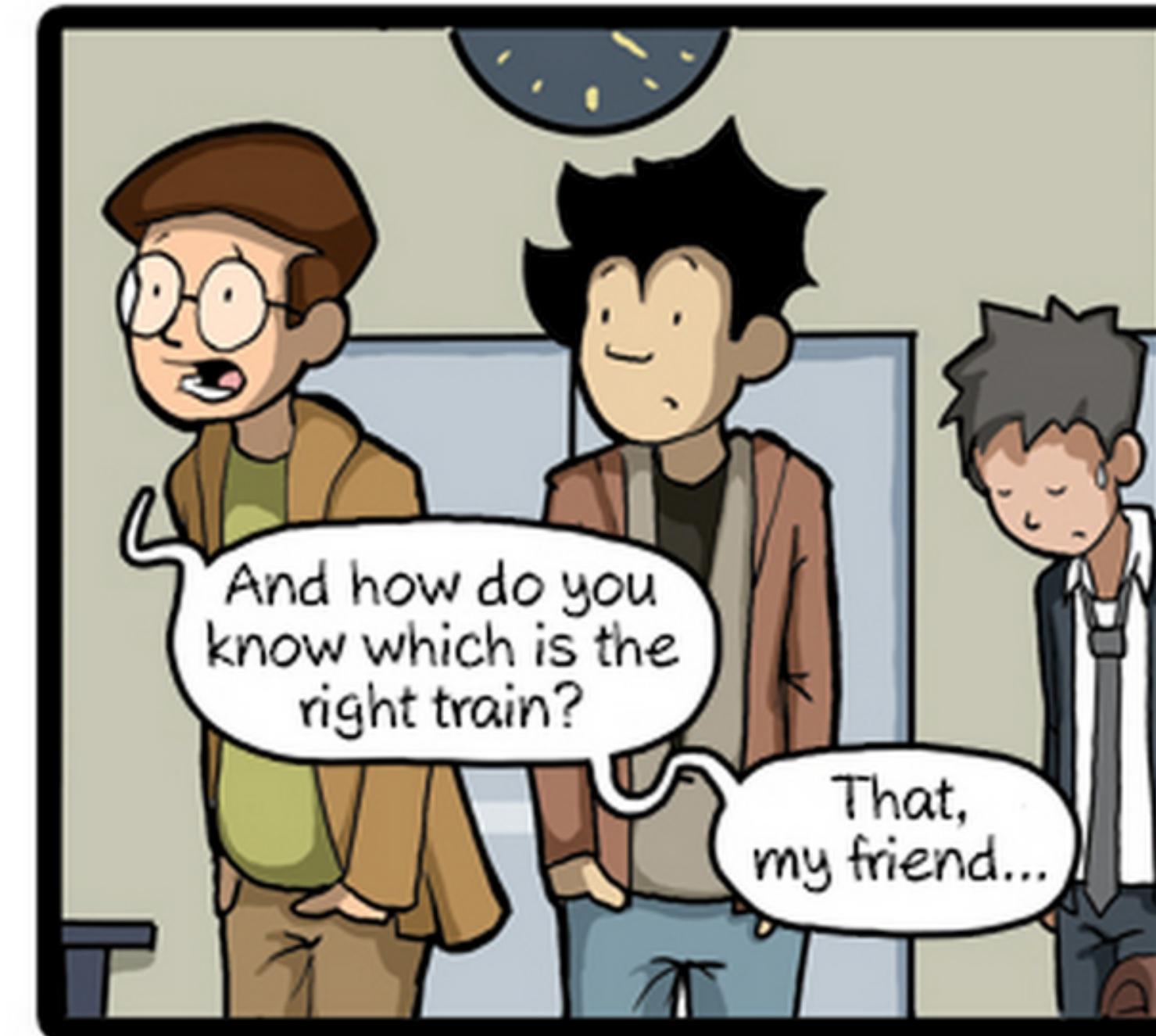
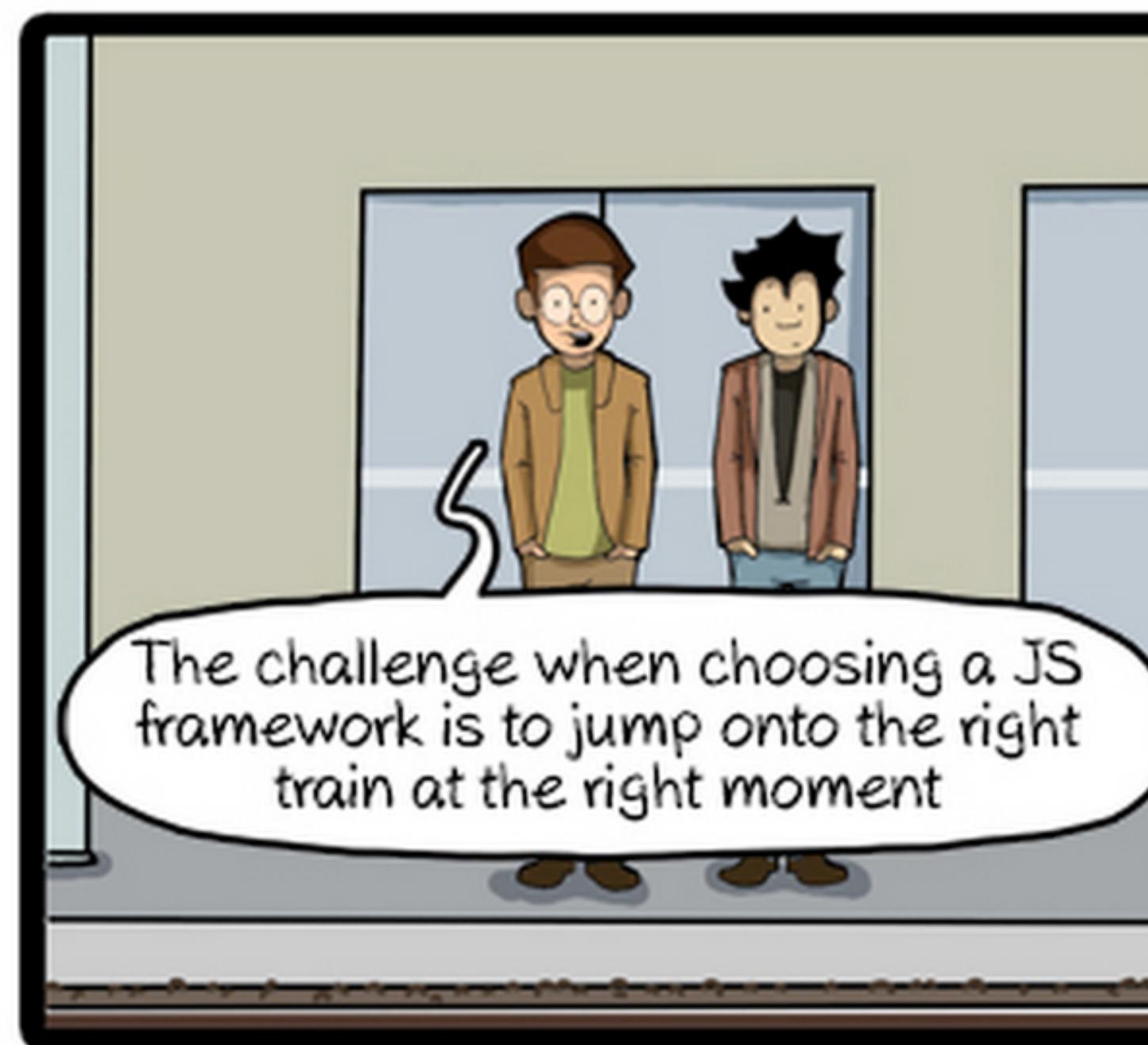


These are just my opinions!

And you know what they say about opinions!

Discuss: when should you use x?

Discuss: when should you *not* use x?



JAVASCRIPT FRAMEWORKS AND LIBRARIES

Techno	Time	Passengers*	Stars*	Status
VANILLAJS	1995	31%	-	ALWAYS ON TIME
MOOTTOOLS	2006	04%	02K	CANCELLED
JQUERY	2006	65%	35K	STILL ON TIME
PROTOTYPE	2005	04%	03K	CANCELLED
ANGULAR 1	2009	.2%	42K	WHO KNOWS?
COFFEESCRIPT	2009	=NC	12K	SLOWING DOWN
BACKBONE.JS	2010	.1%	23K	ON TIME
BOOTSTRAP	2011	09%	86K	ON TIME - POPULAR
EMBER.JS	2011	=NC	14K	NEW TRAIN COMING
METEOR	2012	=NC	28K	BOARDING
REACT	2013	=NC	28K	CRAZY BOARDING
ANGULAR 2	MEH	=NC	05K	MAYBE LATE, MAYBE NOT

*PASSENGERS BASED ON W3TECHS SURVEYS

*STARS ON GITHUB

...is the real
question!



Action!

Don't be afraid to try new things

Learn JavaScript

Try *one* of these frameworks

Form your own opinions

Or just wait a few months...



Questions?

Contact Information

-  <http://raibledesigns.com>
-  [@mraible](https://twitter.com/mraible)

Presentations

-  <http://slideshare.net/mraible>

Code

-  <http://github.com/mraible>

