FRONT END FRAMEWORKS

As you work in front end JS, you find some common tasks

- (re)generating HTML based on variables
- Making service calls
- Reporting errors to the user
- Repeating portions of output
- Dealing with browser/version differences

COMMON LIBRARIES

Libraries package some of this common code in a useful interface

* jQuery (DOM searching/manpulation, service calls, browser diffs)
* Moustache, HandlebarsJS (templating)
* A bajillion others

Libraries don't tend to care HOW you use the options they expose

FRAMEWORKS

A "Framework" doesn't just provide an interface to common code

Make it easy and convenient to follow a particular organization of code, user flow, and data flow

A framework is "opinionated"

- Helpful to do things the way the framework wants
- Problems to do something they don't want

MODERN LIBRARIES/FRAMEWORKS

People may often argue over whether a package is a library or a framework

What matters is the impact, not the title

- Angular (Angular vs AngularJS)
- React
- Vue
- Backbone
- Ember
- Meteor

WHICH LIBRARY/FRAMEWORK IS THE BEST?

It Depends

- Strengths, Weaknesses - which apply to this task?
- No single best

"What is the best language?"

- If there was one clear best there wouldn't be so many

WHY REACT?

- "View" only - unopinionated about the rest
  - Which makes it more flexible
- React is Declarative
  - Makes it less complex to read/change
- React maps closely to HTML output
- React components map closely to JS functions
  - Means same best practices
  - Means non-React skills increase React skills
  - Means React skills are useful even without React
- Also very popular right now (most? ¯_(ツ)_/¯)
  - Means helps with jobs

HOW DOES REACT WORK

Imagine a TODO list backed by services

- state is maintained in some JS variables
- When the state changes, you rewrite the HTML
- Ideally, you only change the HTML if there is a REAL change

React makes this easy

JSX

- `JSX` is a syntax that LOOKS like HTML, but transpiles to JS
- React is made of "components" that act like functions that output an HTML element
- Any logic that isn't presentation layer should NOT be in the JSX
  - still separating concerns, even if it feels like we aren't