

Angular.JS?

React.JS?

Vanilla JS!

Emiliano Daddario, 2015

Creative Commons Attribution International  
4.0 license

# From "You have ruined HTML":

- «JSX sucks for many (...) reasons»
- Example: f\*\*\*ing copyright footer © josieusa
- ```
<div dangerouslySetInnerHTML={{__html:
'&copy; josieusa'}} />
```
- JSX devs make free software 'cos making a f\*\*\*ing copyright footer in JSX is too hard
- JSX is spelled like "Jay Has Sex". Can't blame Jay 'cos writing HTML ain't fun anymore. The aforementioned alternative, instead, is... :-)

# You have ruined CSS

- Years of "separation of concerns in CSS, HTML, JS", "separate CSS files" blah blah...
- ...React doesn't discourage inline styles!
- There's even React Inline to get rid of 'em!

# From "You have ruined HTML":

- «JSX sucks for many (...) reasons»
- Example: f\*\*\*ing copyright footer © josieusa
- ```
<div dangerouslySetInnerHTML={{__html: ' &copy; josieusa' }} />
```
- JSX devs make free software 'cos making a f\*\*\*ing copyright footer in JSX is too hard
- JSX is spelled like "Jay Has Sex". Can't blame Jay 'cos writing HTML ain't fun anymore. The aforementioned alternative, instead, well... :-)

From "You have ruined JavaScript":

«An Angular "service" is a singleton object created by a "service factory". These service factories are functions which, in turn, are created by a "service provider". "The service providers are constructor functions". When instantiated they must contain a property called \$get, which holds the service factory function.»

# WAT?

(photo credits: [1])



But you ain't ruined the new JS,  
HTML, CSS yet

- Babel JS: too new to be already ruined
- Babel is our only "framework" (joking, not really a framework)

# I was kidding

- I love Angular and especially React
- I believe all these projects owe something to the other projects
- They sure had INFLUENCE
- For example Angular's team worked on AtScript, now TypeScript, which is similar to Babel JS. Flux architecture in frameworks that use React, encouraged functional programming for sure. Guess what? EcmaScript 2015, and thus Babel JS, support functional programming far better than ES5
- Would Babel JS have existed the way it is, without Angular and React?
- In the future I bet features borrowed by React and Angular will become increasingly supported natively by browsers, and they will be increasingly influenced by new standard EcmaScript features



# But...

- Complete applications can be created with just Babel and a solid architecture
- Complete, that is, I made a todo app with:
  - jQuery replacement!
  - Unit tests!
  - Template engine!
  - Even a complete IDE!

# A complete IDE!

- Quark IDE!
- Quick and dirty: Quark code really sucks (but the todo app doesn't suck)
- Can be shared with an URL (including code)
- Surprise features...

# Enter MVI

- A clear architecture!
- Model, View, Intent
- True to the spirit of Smalltalk MVC
- 1-D flow: View  $\rightarrow$  Intent  $\rightarrow$  Model  $\rightarrow$  View
- No 2-way binding (anti-pattern? Angular 1 :-)
- Clearer deps using Reactive, not Interactive

# Demoed Babel and HTML5 features

- Comprehensions!
- Bind syntax!
- Arrow functions!
- Constants!
- HTML5: `querySelectorAll`!
- Not supported by Quark yet: ES2015 modules!
- Not supported by Quark yet: `Object.observe`

# What was the surprise in Quark?

- `PLAYBACK LIVE CODING` (idea of mine)
- I wasn't writing the code live, it was in playback!
- Hope you liked the joke!
- Also: "PRETEND YOU'RE WORKING" mode!

# Photo credits

- [1] <https://www.flickr.com/photos/bnazario/2442576879>