

jQuery => (Angular || React)

Moving from jQuery to Angular & React

Code at: <https://github.com/jeffreypriebe/scjs-jQNgRe-Places>

Jeffrey Priebe
jeffreypriebe@gmail.com
@jeffreypriebe

Comparing

	jQuery	Angular & React
Approach	Imperative	Declarative
Code	Commands to the Browser/DOM <pre><code>\$('element').doThisThing().then \$('element').click(\$('container > el').remove())</code></pre> About the DOM, not model/state	Changes of Model/State <pre><code>ng-click=del(this) \$scope.del => datastore.delete(modelItem)</code></pre> All model/state, DOM is abstracted away
Result	Large apps easily become a bunch of spaghetti = Harder to Maintain ...	Easier to Reason about: Small units of related code = Easier to Maintain ...

Getting Started: It's not Free

- jQuery

```
<script src=jQueryCDN>  
$( '.myElement' ).text( 'new' );
```

- Angular

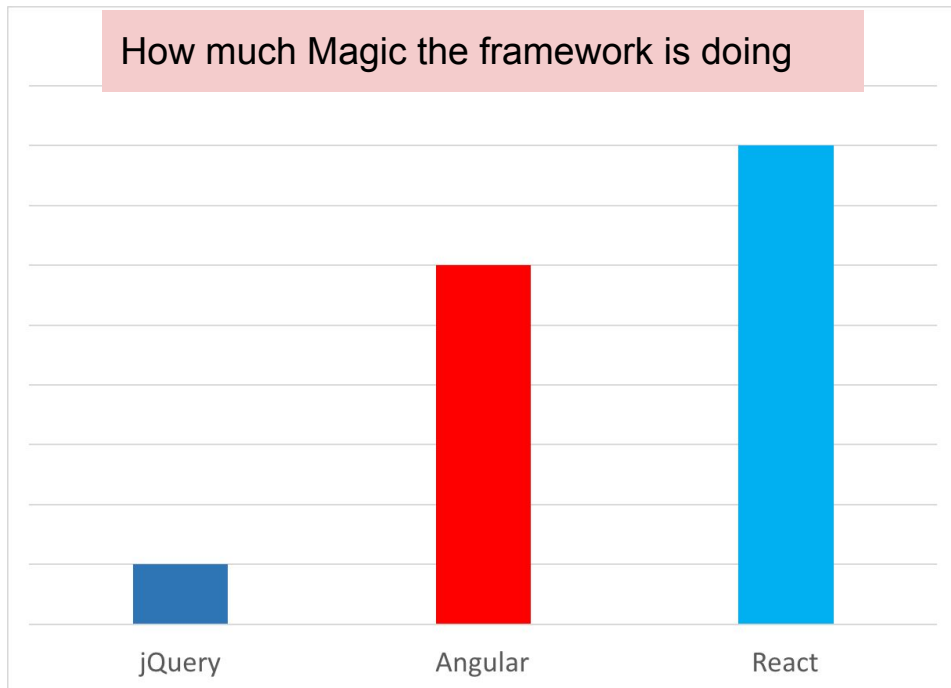
```
<script src=NgCDN>  
@ng-app  
angular.module...
```

Cursing

- React

```
modules + webpack...  
ReactDOM.Render  
JSX!?  
Component.render  
GOTO jQuery ?
```

Much Cursing



More to learn at start = more value

Demo



TO THE CODE!

Code Repo: <https://github.com/jeffreypriebe/scjs-jQNgRe-Places>

Disclaimers:

- Not about isomorphism, mobile builds: just “things running in web browsers.”
- Versions: jQuery 1, Angular 1 (not 2!), React 15.
- ES6/7: Cuz I can - not required (just like how you can bungee jump without a cord, if you wanted to).
- JSX
- No templating, no routing, no flux/redux: simplified. Zooming in past those things.
- h/t: <http://todomvc.com/>

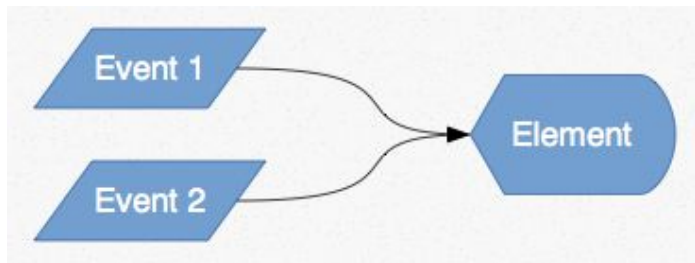
Code Comparison

<code>.click()</code> “bindEvents”	Attribute: <code>ng-submit</code>	Attribute: <code>onClick</code>
<code>.append(.map(\$(`<li />`))...)</code>	<code><li ng-repeat="place in places"></code>	<code>places .map(<Place> aka =)</code>
Repeated <code>.on('keyup blur', .removeClass('edit'))</code>	<code><li ng-class="{place == editing}"></code>	<code><li className="this. state.editing"></code>

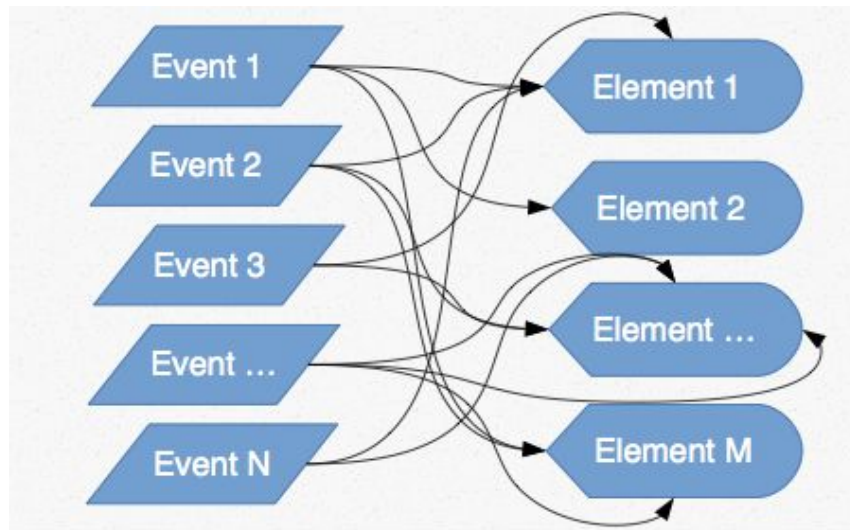
Bonus Content Follows
(not shown in presentation)

Why?

jQuery Starts as



And Incrementally grows into



Comparing

	jQuery	Angular & React
Approach	Imperative	Declarative
Code	Commands to the Browser/DOM <pre><code>\$('element').doThisThing().then() \$('element').click(\$('container > el').remove())</code></pre> All DOM, no model/state	Changes of Model/State <pre><code>ng-click=del(modelItem) \$scope.del => datastore.delete(modelItem)</code></pre> All model/state, DOM is abstracted away
Result	Large apps easily become a bunch of spaghetti = Harder to Maintain	Easier to Reason about: Small units of related code = Easier to Maintain
	JavaScript is separate but tightly coupled with HTML	Model logic, so uncoupled from the HTML Ng: Model logic is uncoupled React: JavaScript next to relevant HTML