

# Tradeoffs

---



**Cory House**

@housecor | reactjsconsulting.com



# Agenda



6 Key Tradeoffs



# Tradeoffs

Framework  
Concise  
Template-centric  
Separate template  
Standard  
Community backed

Library  
Explicit  
JavaScript-centric  
Single-file component  
Non-standard  
Corporate backed



# 1

## Tradeoff Framework vs Library

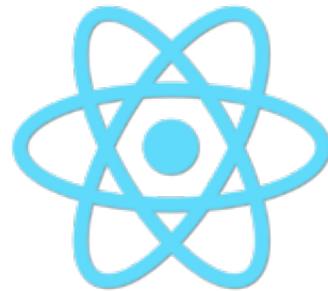


## Framework



- Clear opinions
- Less decision fatigue
- Less setup overhead
- More cross-team consistency

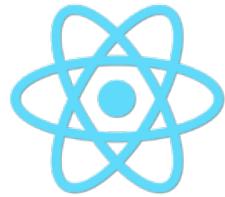
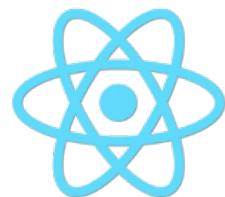
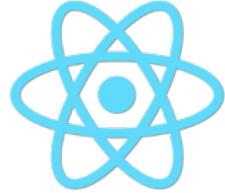
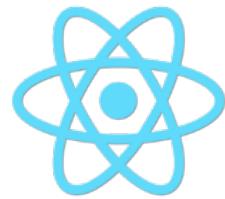
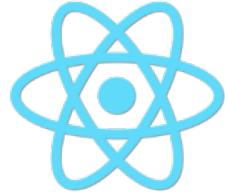
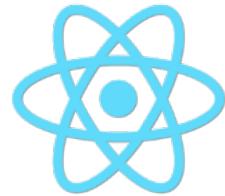
## Library



- Light-weight
- Use on existing apps
- Pick what you need
- Choose best tech
- Popular boilerplates exist



## My Existing App

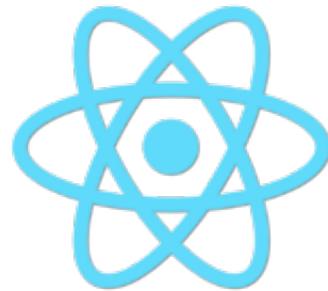


## Framework



- Clear opinions
- Less decision fatigue
- Less setup overhead
- More cross-team consistency

## Library



- Light-weight
- Use on existing apps
- Pick what you need
- Choose best tech
- Popular boilerplates exist



# Features

	<b>React</b>	<b>Angular</b>
<b>Components</b>	✓	✓
<b>Testing</b>		✓
<b>HTTP library</b>		✓
<b>Routing</b>		✓
<b>I18n</b>		✓
<b>Animation</b>		✓
<b>Form validation</b>		✓



# Features

	React	Angular
Components	✓	✓
Testing	Jest, Mocha	✓
HTTP library	Fetch, Axios	✓
Routing	React Router	✓
I18n	react-intl	✓
Animation	react-motion	✓
Form validation	react-forms	✓
CLI	create-react-app	angular-cli





This repository

Search

Pull requests Issues Marketplace Explore

[facebookincubator / create-react-app](#)[Watch](#) ▾

989

[Unstar](#)

34,391

[Fork](#)

5,684

[Code](#)[Issues 245](#)[Pull requests 116](#)[Projects 0](#)[Wiki](#)[Insights ▾](#)

Create React apps with no build configuration.

[react](#)[zero-configuration](#)[build-tools](#)[1,197 commits](#)[27 branches](#)[136 releases](#)[366 contributors](#)[BSD-3-Clause](#)

Branch: master ▾

[New pull request](#)[Create new file](#)[Upload files](#)[Find file](#)[Clone or download ▾](#)

davidleger95 committed with Timer Resolved issue #2971 (#2989) ...

Latest commit 206f9e8 4 days ago

[.github](#)

Issue template: fix env list ordering (#2697)

3 months ago

[packages](#)

Resolved issue #2971 (#2989)

4 days ago

[tasks](#)

Revert "run npm 5.4.0 in CI (#3026)" (#3107)

5 days ago

[template](#)

Add note for using CHOKIDAR\_USEPOLLING in virtual machines to enable ...

7 months ago

[.eslintignore](#)

Fix internal linting setup and add missing headers (#1741)

6 months ago

[.eslintrc](#)

[Internal] Warn for missing brackets (#2151)

4 months ago

[.gitignore](#)

NPM version check for tip (#1193)

7 months ago

filter projects

TAGS

AGE

198 Projects

Sort ▾

 create-react-app

Create React apps with no build configuration.

74,942 1d [dependency](#) [linter](#) [minimal](#) next.js

Next.js is a lightweight framework for static and server-rendered applications.

43,702 5h [dependency](#) [babel](#) [hmr](#) [linter](#) [tests](#) [webpack 3](#) gatsby Blazing fast static site generator for React41,127 59min [dependency](#) [css modules](#) [babel](#) [redux](#) [hmr](#) [linter](#) [router](#) [tests](#) [webpack 4](#) react-boilerplate

:fire: A highly scalable, offline-first foundation with the best developer experience and a focus on performance and best

24,298 

# 2

## Tradeoff Concise vs Explicit



# Two-way binding

Less coding

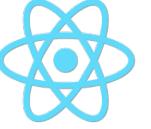
Automatic

```
let user = 'Cory';
```

```
<input  
  type="text"  
  value={user}>
```

"Magically"  
kept in-sync

# One-way binding



More control

More explicit

Easy to debug

```
state = { user: 'Cory' };
```

```
function handleChange(event) {  
  this.setState({  
    user: event.target.value  
  });  
}
```

Explicit change  
handler

```
<input  
  type="text"  
  value={this.state.user}  
  onChange={this.handleChange}>
```



## Two-way binding

Less coding  
Automatic

```
let user = 'Cory';
```

```
<input  
  type="text"  
  value={user}  
/>
```

Doable with React,  
but not popular or  
recommended.



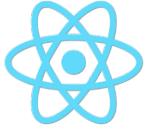
## One-way binding

More control  
More explicit  
Easy to debug

```
state = { user: 'Cory' };
```

```
function handleChange(event) {  
  this.setState({  
    user: event.target.value  
  });  
}
```

```
<input  
  type="text"  
  value={this.state.user}  
  onChange={this.handleChange}  
/>
```



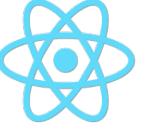
# Two-way binding

Less coding  
Automatic

```
let user = 'Cory';
```

```
<input  
  type="text"  
  value={user}  
/>
```

Don't worry,  
you can  
declare one  
per form. ☺

One-way binding 

More control  
More explicit  
Easy to debug

```
state = { user: 'Cory' };  
  
function handleChange(event) {  
  this.setState({  
    user: event.target.value  
  });  
}
```

```
<input  
  type="text"  
  value={this.state.user}  
  onChange={this.handleChange}  
/>
```



# 3

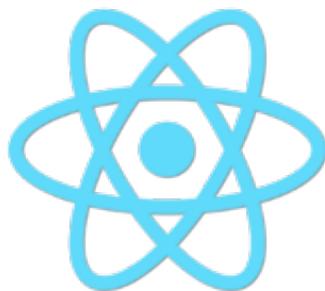
## Tradeoff Template-centric vs JavaScript-centric





**“JS” in HTML**

Learn their unique syntax



**“HTML” in JS**

Learn JS



## Conditional



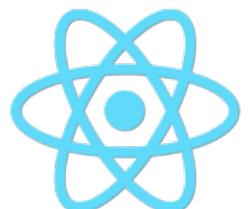
```
<h1 *ngIf="isAdmin">Hi Admin</h1>
```



```
<h1 v-if="isAdmin">Hi Admin</h1>
```



```
<h1>{{if isAdmin 'Hi Admin'}}</h1>
```



```
{isAdmin && <h1>Hi Admin</h1>}
```

Since plain JS, you get:  
1. Autocomplete support  
2. Error messages



# Loop



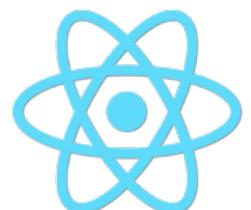
```
<div *ngFor="let user of users">{{user.name}}</div>
```



```
<div v-for="user in users">{{user.name}}</div>
```



```
{{#each users as |user|}}
<div>{{user.name}}</div>
{{/each}}
```



```
users.map(user => <div>{{user.name}}</div>)
```



# Event



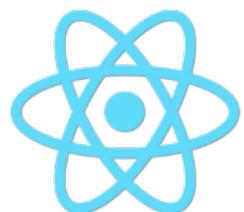
```
<button (click)="delete()">Delete</button>
```



```
<button v-on:click="delete">Delete</button>
```



```
<button onclick={{action 'delete'}}>Delete</button>
```



```
<button onClick={delete}>Delete</button>
```





**if**

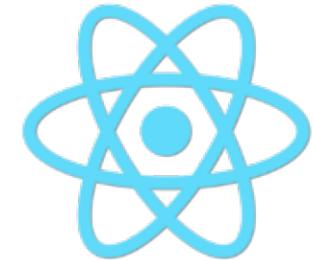
ngIf



v-if



#if



if or &&

**loop**

ngFor

v-for

#each

map

**click**

(click)

v-on:click

onclick

onClick



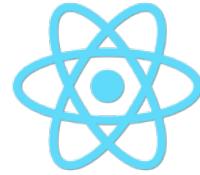


## Template-centric

Little JS knowledge required

Avoid confusion with JS binding

Rule of least power



## JavaScript-centric



## Angular Docs:

JavaScript expressions that have or promote side effects are prohibited, including:

- assignments ( `=` , `+=` , `-=` , ...)
- `new`
- chaining expressions with `;` or `,`
- increment and decrement operators ( `++` and `--` )

Other notable differences from JavaScript syntax include:

- no support for the bitwise operators `|` and `&`
- new [template expression operators](#), such as `|` , `?.` and `!.`



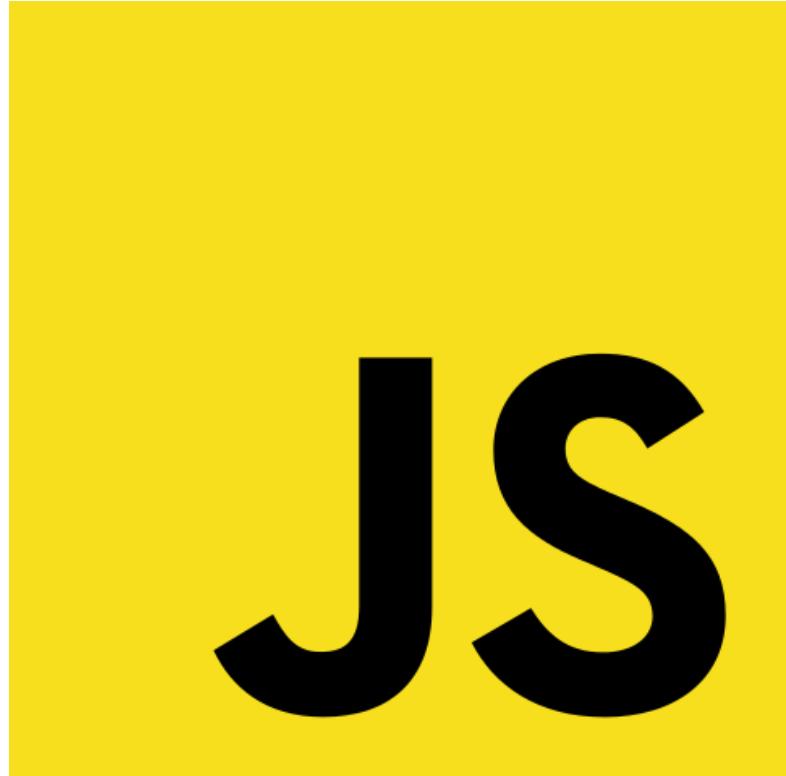
## Template-centric

- Little JS knowledge required
- Avoid confusion with JS binding
- Rule of least power

## JavaScript-centric

- Little framework-specific syntax
- Fewer concepts to learn. It's JS.
- Less code
- Easy to read
- Encourages improving JS skills





Modules  
Let and Const  
Enhanced Object literals  
Default Parameters  
Template Strings  
Classes  
Arrow functions  
Promises  
Destructuring  
Spread Operator



## Template-centric

- Little JS knowledge required
- Avoid confusion with JS binding
- Rule of least power

## JavaScript-centric

- Little framework-specific syntax
- Fewer concepts to learn. It's JS.
- Less code
- Easy to read
- Encourages improving JS skills
- Skills transfer

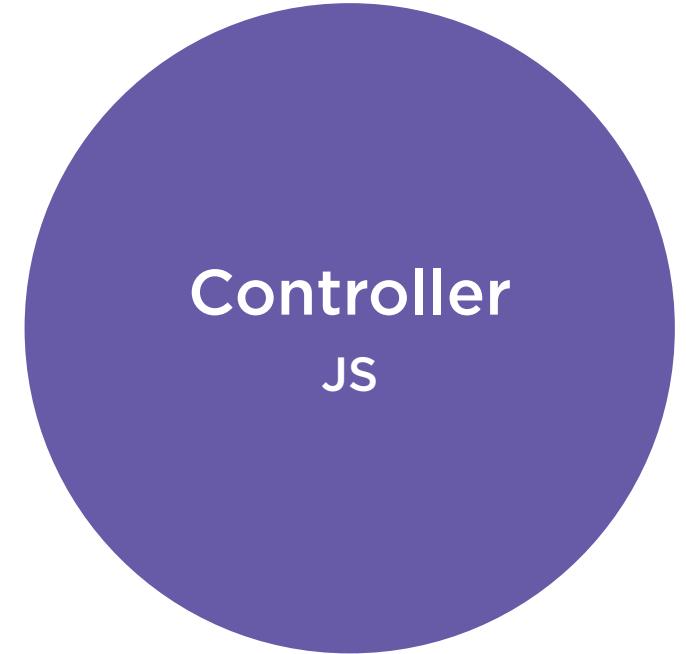
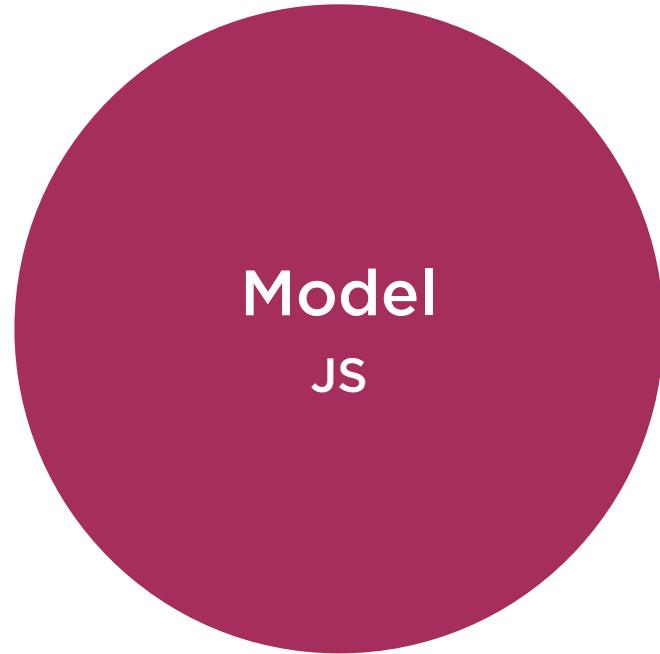


# 4

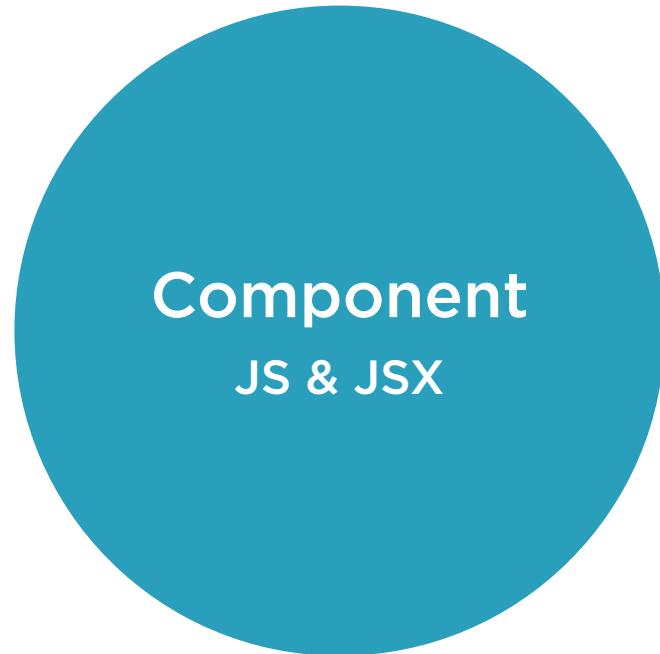
## Tradeoff Separate Template vs Single File



MVC



React





**Ben Alman**

@cowboy



+ Follow

Facebook: Rethink established best practices™



...

---

RETWEETS

21

FAVORITES

21



4:40 PM - 29 May 2013



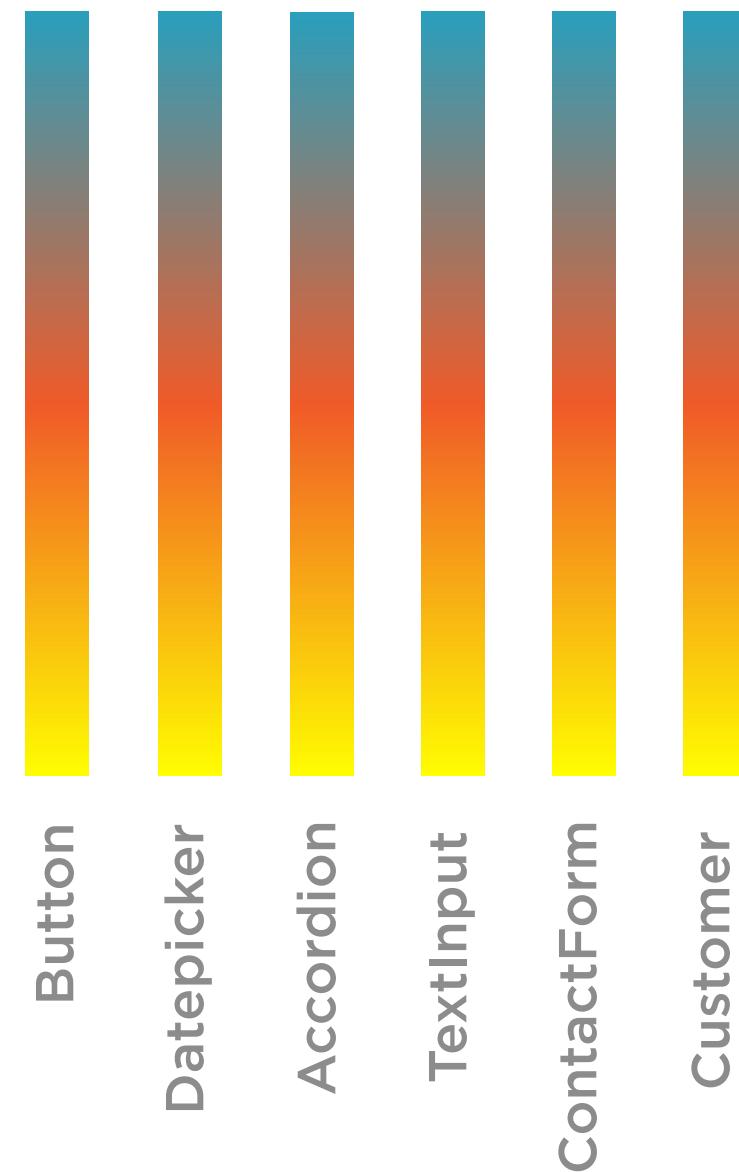
**What about separation of concerns?**



Separate technologies, but *intertwined* concerns.



Each *component* is a separate concern.



# React: Nested Components





LIBRARY

Home  
**NavLink**

Browse

Paths

Mentors

Channels

Bookmarks

Notes

**SidebarNav****AuthorPhoto**Cory House  
Pluralsight Author

Cory is an independent consultant with over 15 years of experience in software development. He is a Microsoft MVP, ASP Insider, and a member of the Telerik developer experts program. As a software architect at Autotrader, Cory specializes in creating single page applications for the automotive industry using C#, .NET, Node, and JavaScript. Cory coaches teams around the world on clean coding practices, software architecture, and front-end development. He also speaks regularly at regional and international conferences like NDC, Fluent, and Codemash. Cory lives in Kansas City where he blogs at [bitnative.com](http://bitnative.com) and is active on Twitter as @housecor.

**AuthorSummary**

Courses authored by Cory House

**AuthorCourses****CourseSummary****Building a JavaScript Development Environment**

by Cory House Beginner 5h 19m 10 November 2016 ★★★★★ (163)

**Building Applications with React and Redux in ES6**

by Cory House Intermediate 6h 13m 20 May 2016

★★★★★ (558)

**Building Applications with React and Flux**

by Cory House Intermediate 5h 8m 12 August 2015 ★★★★★ (967)

**HTML5 Web Component Fundamentals**

by Cory House Beginner 5h 3m 09 January 2015 ★★★★★ (370)

**Becoming an Outlier: Reprogramming the Developer Mind**

by Cory House Intermediate 2h 33m 24 April 2014 ★★★★★ (724)

**Architecting Applications for the Real World in .NET**

by Cory House Intermediate 2h 52m 07 January 2014 ★★★★★ (1532)

**Clean Code: Writing Code for Humans**

by Cory House Intermediate 3h 10m 07 October 2013 ★★★★★ (1590)

**StarRating**



LIBRARY

Home  
NavLink

Browse

Paths

Mentors

Channels

Bookmarks

Notes

## SidebarNav

## AuthorPhoto

Cory House  
Pluralsight Author

Cory is an independent consultant with over 15 years of experience in software development. He is a Microsoft MVP, ASP Insider, and a member of the Telerik developer experts program. As a software architect at Autotrader, Cory specializes in creating single page applications for the automotive industry using C#, .NET, Node, and JavaScript. Cory coaches teams around the world on clean coding practices, software architecture, and front-end development. He also speaks regularly at regional and international conferences like NDC, Fluent, and Codemash. Cory lives in Kansas City where he blogs at [bitnative.com](http://bitnative.com) and is active on Twitter as @housecor.

## AuthorSummary

Courses authored by Cory House

## AuthorCourses

## Building a JavaScript Development Environment

by Cory House

Beginner

5h 19m

10 November 2016

★★★★★ (163)

## CourseSummary

## Building Applications with React and Redux in ES6

by Cory House

Intermediate

6h 13m

20 May 2016

★★★★★ (558)

## Building Applications with React and Flux

by Cory House

Intermediate

5h 8m

12 August 2015

★★★★★ (967)

## HTML5 Web Component Fundamentals

by Cory House

Beginner

5h 3m

09 January 2015

★★★★★ (370)

## Becoming an Outlier: Reprogramming the Developer Mind

by Cory House

Intermediate

2h 33m

24 April 2014

★★★★★ (724)

## Architecting Applications for the Real World in .NET

by Cory House

Intermediate

2h 52m

07 January 2014

★★★★★ (1532)

## Clean Code: Writing Code for Humans

by Cory House

Intermediate

3h 10m

07 October 2013

★★★★★ (1590)

## StarRating



Cory

housecor@gmail.com

...

# 5

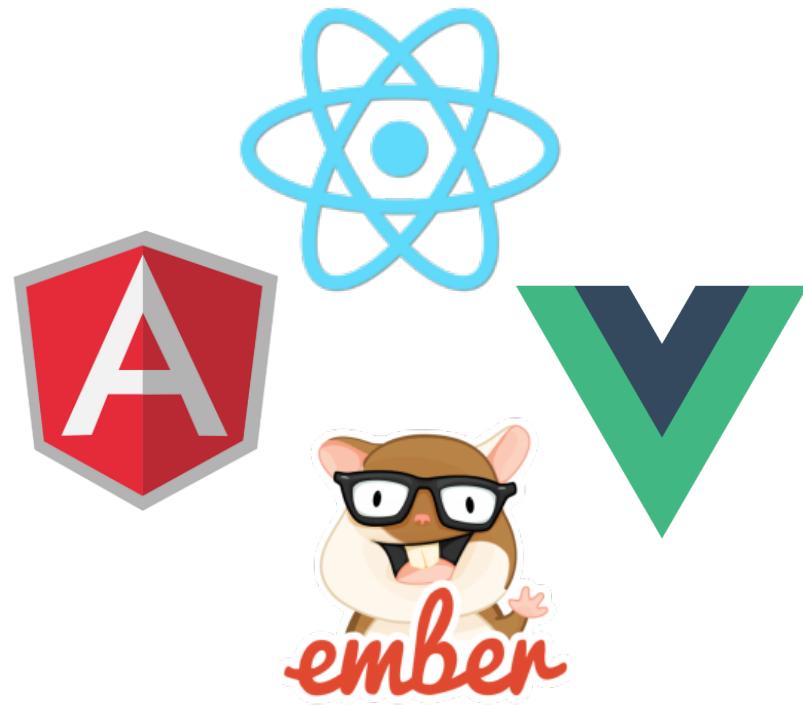
## Tradeoff Standard vs Non-standard



Standard



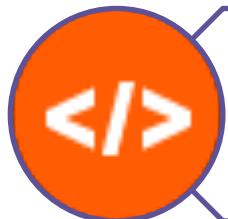
Non-standard





## Templates

Inert, reusable markup



## Custom Elements

Define our own elements



## Shadow DOM

Encapsulated styling





Home



Browse

Search...

Paths



Channels



Bookmarks



Q&amp;A



## Problem #4: No Bundling

What if you want to bundle some CSS, JS, and HTML together?

```
Consider Bootstrap.  
<script src="jquery.min.js"></script>  
<script src="bootstrap.min.js"></script>  
<link rel="stylesheet" href="bootstrap.min.css" />  
<script src="script.js"></script>  
<link rel="template" href="template.html" />
```

# HTML5 Web Component Fundamentals

by Cory House

Learn how to use the Shadow DOM, Custom Elements, Templates, and Imports to create reusable web components.

[Resume Course](#) [Bookmark](#) [Add to Channel](#) [Download Course](#)

[Table of contents](#)[Description](#)[Transcript](#)[Exercise files](#)[Discussion](#)[Learning Check](#)[Related Courses](#)

This course is part of: [HTML5 Path](#)

[Expand All](#)[Five Problems, One Solution](#)

23m 52s



Course author



Cory House

Cory is the principal consultant at [reactjsconsulting.com](http://reactjsconsulting.com), where he has helped dozens of companies transition to React. Cory has trained over 10,000 software developers at events and businesses...

Course info

Level Beginner

Rating (448)

My rating

Duration 5h 3m

Released 9 Jan 2015

Why not use the platform?



# Why Not Web Components?



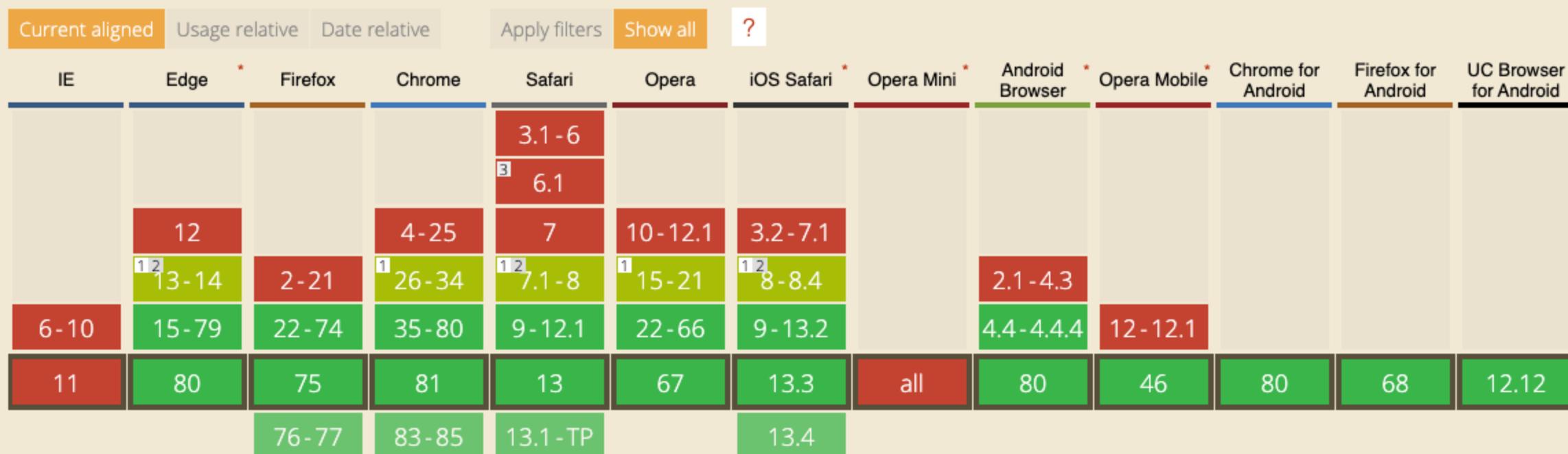
**Spotty browser support = polyfills req'd**



#

# HTML templates - LS

Method of declaring a portion of reusable markup that is parsed but not rendered until cloned.

[Notes](#)[Known issues \(1\)](#)[Resources \(8\)](#)[Feedback](#)

<sup>1</sup> Does not support `Document.importNode` on templates, nested templates or elements that contain templates.

<sup>2</sup> Does not support `Node.cloneNode` on templates, nested templates or elements that contain templates.

<sup>3</sup> Does not support `Element.replaceWith` on templates, nested templates or elements that contain templates.

# # Custom Elements (V1) - LS

Usage

% of all users



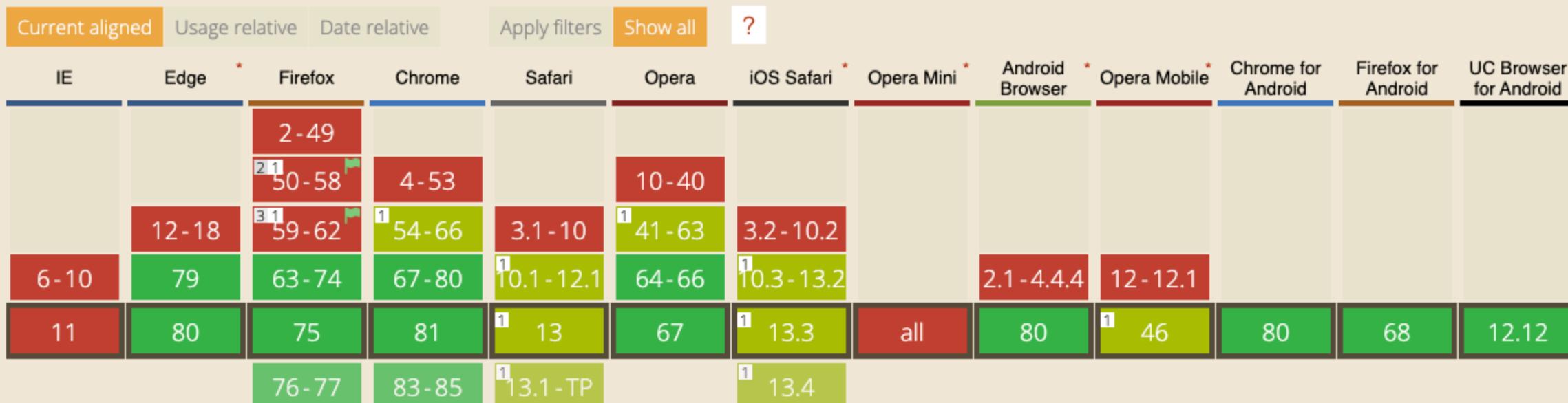
Global

72.86% + 17.69% = 90.55%

U.S.A.

44.11% + 48.09% = 92.2%

Method of defining new HTML tags.



Notes

Known issues (0)

Resources (11)

Feedback

<sup>1</sup> Supports "Autonomous custom elements" but not "Customized built-in elements".

<sup>2</sup> Enabled through the `dom.webcomponents.enabled` preference in `about:config`.

<sup>3</sup> Enabled through the `dom.webcomponents.customelements.enabled` preference in `about:config`.

#

# Shadow DOM (V1) - WD

Usage

% of all users



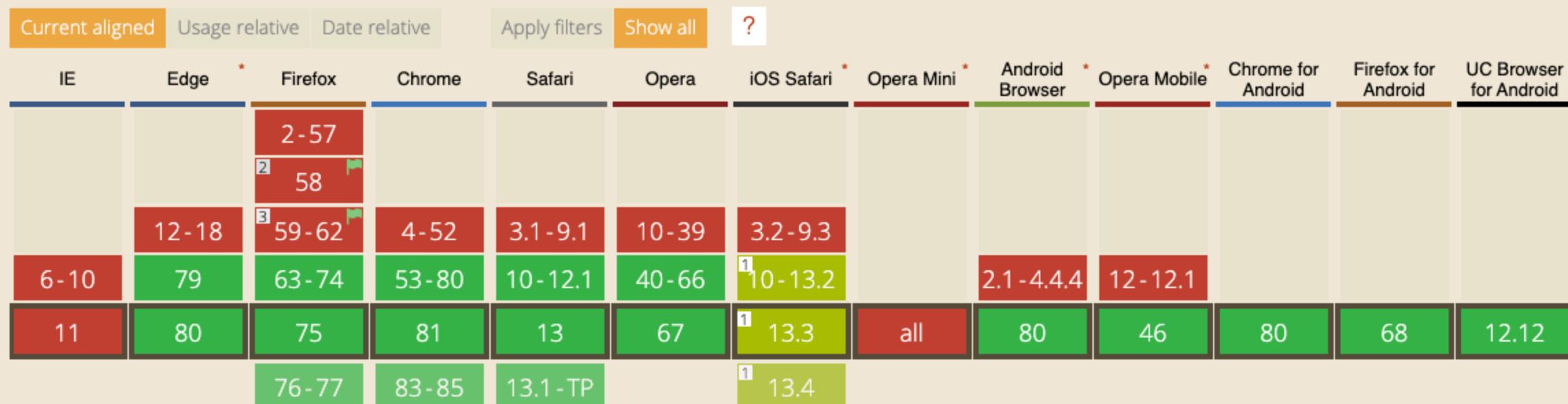
Global

77.49% + 13.39% = 90.88%

U.S.A.

47.25% + 45.14% = 92.39%

Method of establishing and maintaining functional boundaries between DOM trees and how these trees interact with each other within a document, thus enabling better functional encapsulation within the DOM & CSS.



Notes

Known issues (0)

Resources (8)

Feedback

<sup>1</sup> Certain CSS selectors do not work (`:host > .local-child`) and styling slotted content (`::slotted`) is buggy.

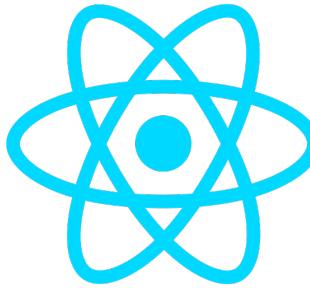
<sup>2</sup> Enabled through the `dom.localDOMComponents.enabled` preference in `about:config`.

# Why Not Web Components?



**Spotty browser support = polyfills req'd**  
**Doesn't enable anything new**





## Web Components

Templates

Custom Elements

Shadow DOM

Imports

## React

JSX, JS

Declare React components

CSS modules, CSS in JS, “inline”

One component per file



# Why Not Web Components?



**Spotty browser support = polyfills req'd**

**Doesn't enable anything new**

**JS libraries keep innovating**

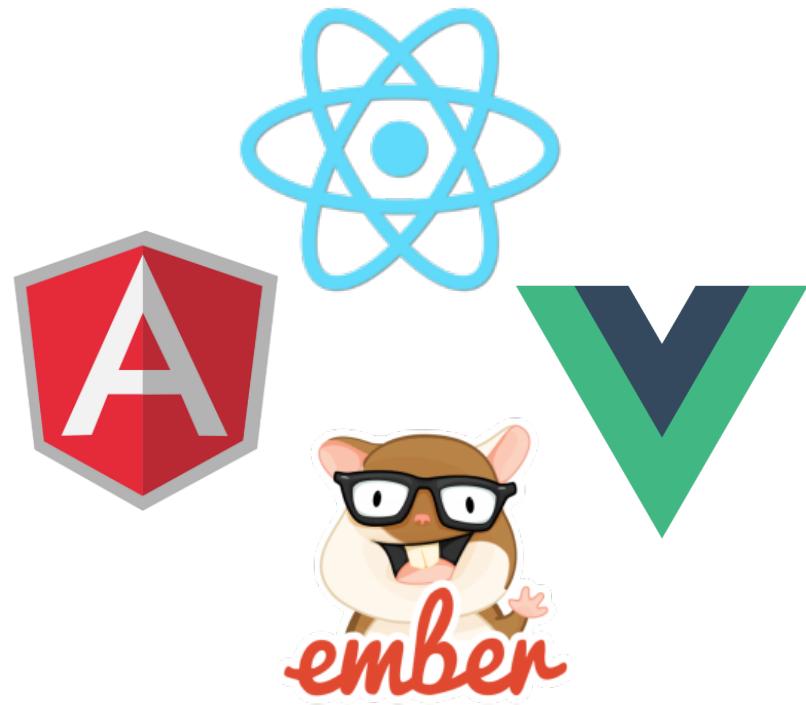
**Only run in the browser**



Standard



Non-standard

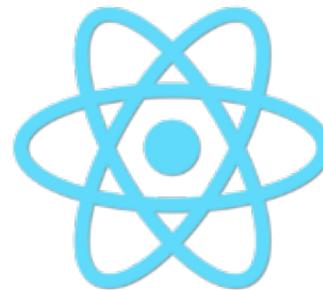


# 6

## Tradeoff Community vs Corporate Backing



# Corporate Backed



**Driven by FB's needs**

**Full-time development staff**  
**Over 1,000 contributors**  
**FB: World's 5<sup>th</sup> most valuable company**  
**50,000 components in prod**



# Summary



**Framework**

**Concise**

**Template-centric**

**Separate template**

**Standard**

**Community backed**

**Library**

**Explicit**

**JavaScript-centric**

**Single-file component**

**Non-standard**

**Corporate backed**

**Next up: Why not React?**

