

# Tradeoffs

---



**Cory House**

@housecor | reactjsconsulting.com



# Agenda



7 Key Tradeoffs



# Tradeoffs

**Framework**  
Concise  
Template-centric  
Separate template  
Standard  
No runtime  
Community backed

**Library**  
Explicit  
JavaScript-centric  
Single-file component  
Non-standard  
Feature-rich runtime  
Corporate backed



# 1 Tradeoff Framework vs Library

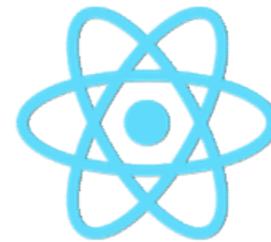


## Framework



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

## Library



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



# Features

	React	Angular
Components	✓	✓
Testing		✓
HTTP library		✓
Routing		✓
I18n		✓
Animation		✓
Form validation		✓



# Features

	<b>React</b>	<b>Angular</b>
<b>Components</b>	✓	✓
<b>Testing</b>	Jest, Mocha	✓
<b>HTTP library</b>	Fetch, Axios	✓
<b>Routing</b>	React Router	✓
<b>I18n</b>	react-intl	✓
<b>Animation</b>	react-motion	✓
<b>Form validation</b>	react-forms	✓
<b>CLI</b>	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	NDM version check for tip (#1102)		7 months ago

Find a React Starter Project...

with:

things you want

without:

things you don't want

Separate search terms with commas.

Examples: Try [redux](#) and [HMR](#), no [gulp](#) or [minimal react native](#) or [vanilla JavaScript](#).

143 matches

Most stars

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

[linter](#) [minimal](#)

34391

6

Create React apps with no build configuration.

[zeit/next.js](#)

[babel](#) [es6](#) [hmr](#) [linter](#) [live reload](#) [tests](#) [webpack](#) [webpack 2](#)

17057

76

Framework for server-rendered React apps

[kriasoft/react-starter-kit](#)

[babel](#) [es6](#) [hmr](#) [linter](#) [live reload](#) [tests](#) [webpack](#)

15398

104

React Starter Kit — isomorphic web app boilerplate (Node.js, Express, GraphQL, React.js, Babel, PostCSS, Webpack, Browsersync)

[react-boilerplate/react-boilerplate](#)

[babel](#) [es6](#) [hmr](#) [linter](#) [live reload](#) [react-router](#) [redux](#) [tests](#) [webpack](#) [webpack 2](#)

15291

80

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

[erikras/react-redux-universal-hot-example](#)

[babel](#) [es6](#) [hmr](#) [inline style](#) [linter](#) [live reload](#) [react-router](#) [redux](#) [tests](#) [universal](#) [webpack](#)

10364

98

A starter boilerplate for a universal webapp using express, react, redux, webpack, and react-transform

[davezuko/react-redux-starter-kit](#)

[babel](#) [es6](#) [hmr](#) [linter](#) [live reload](#) [react-router](#) [redux](#) [tests](#) [webpack](#) [webpack 2](#)

9640

69

Get started with React, Redux, and React-Router.

[coryhouse/react-slingshot](#)

[babel](#) [es6](#) [hmr](#) [linter](#) [live reload](#) [react-router](#) [redux](#) [tests](#) [webpack](#)

7290

57

React + Redux starter kit / boilerplate with Babel, hot reloading, testing, linting and a working example app, all built in

[andrewhfarmer.com/starter-project](http://andrewhfarmer.com/starter-project)

# 2

## Tradeoff Concise vs Explicit



## Two-way binding

Less coding

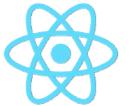
Automatic

```
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';
```

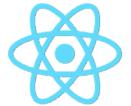
```
<input  
  type="text"  
  value={user}  
/>
```

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}  
/>
```



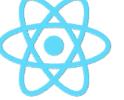
## Two-way binding

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

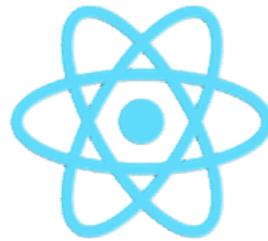
### Template-centric vs JavaScript-centric





**“JS” in HTML**

Learn their unique syntax



**“HTML” in JS**

Learn JS



## Conditional



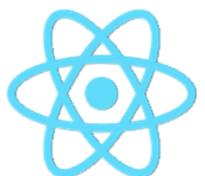
```
<h1 *ngIf="isAdmin">Hi Admin</h1>
```



```
<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



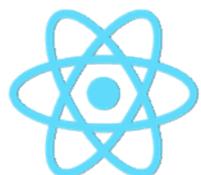
```
<div *ngFor="let user of users">{{user.name}}</div>
```



```
<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



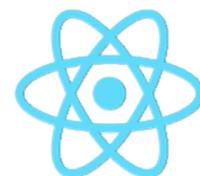
```
<button (click)="delete()">Delete</button>
```



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



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



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





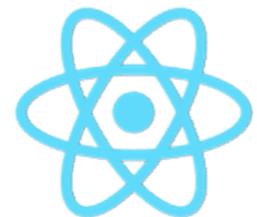
**if**      `ngIf`



`v-if`



`#if`



`if` or `&&`

**loop**      `ngFor`

`v-for`

`#each`

`map`

**click**      `(click)`

`v-on:click`

`onclick`

`onClick`



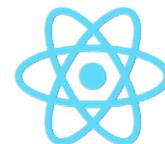


## Template-centric

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



# JS

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

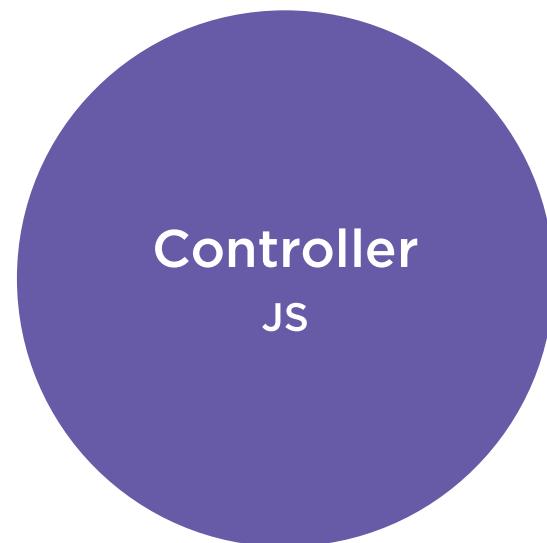
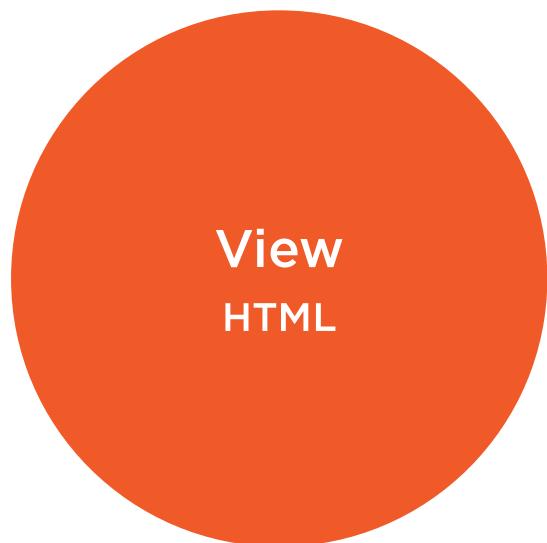
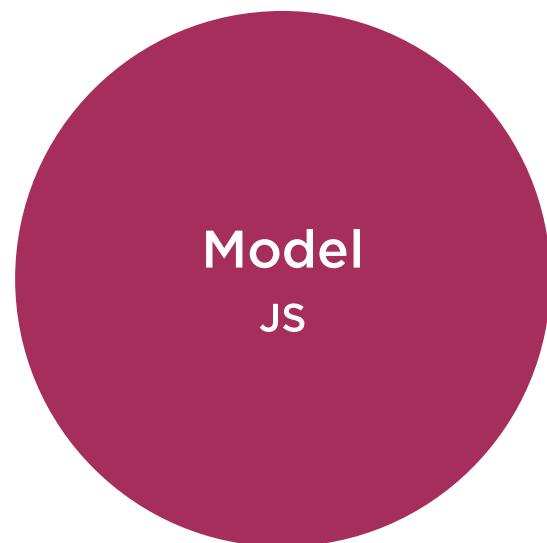


# 4

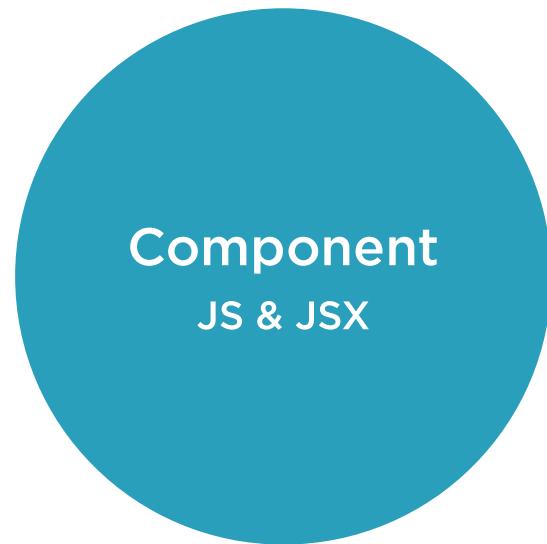
## Tradeoff Separate Template vs Single File



MVC



React





**Ben Alman**  
@cowboy



Follow

Facebook: Rethink established best practices™



RETWEETS

21

FAVORITES

21



4:40 PM - 29 May 2013



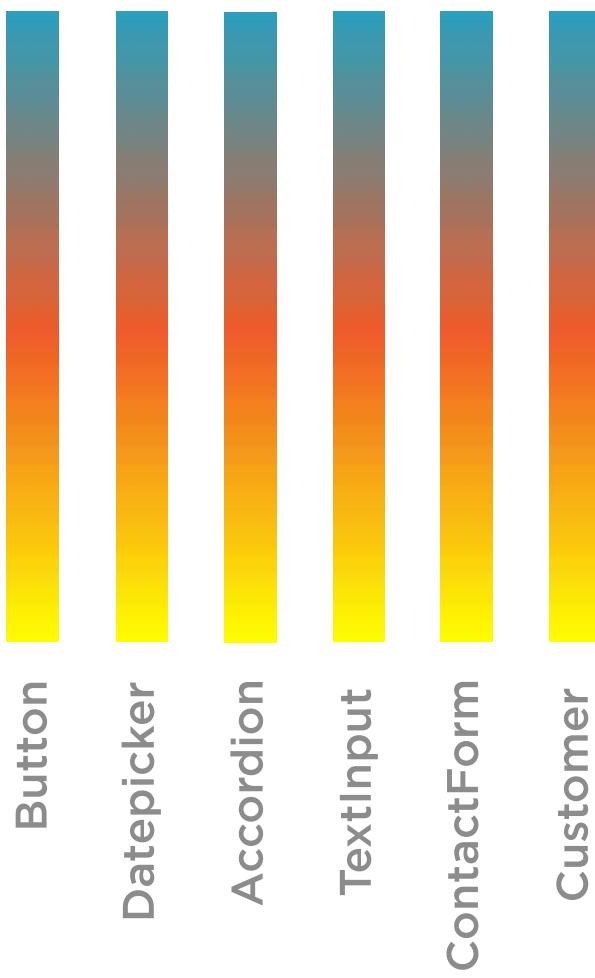
**What about separation of concerns?**



Separate technologies, but *intertwined* concerns.



Each *component* is a separate concern.



Source: speakerdeck.com/didoo/let-there-be-peace-on-css



# React: Nested Components



## SidebarNav

What do you want to learn?

## 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 and is active on Twitter as @housecor.

## AuthorSummary

Cory

housecor@gmail.com



...

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

## SidebarNav

What do you want to learn?

## 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 and is active on Twitter as @housecor.

## AuthorSummary

Cory

housecor@gmail.com



...

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

# 5

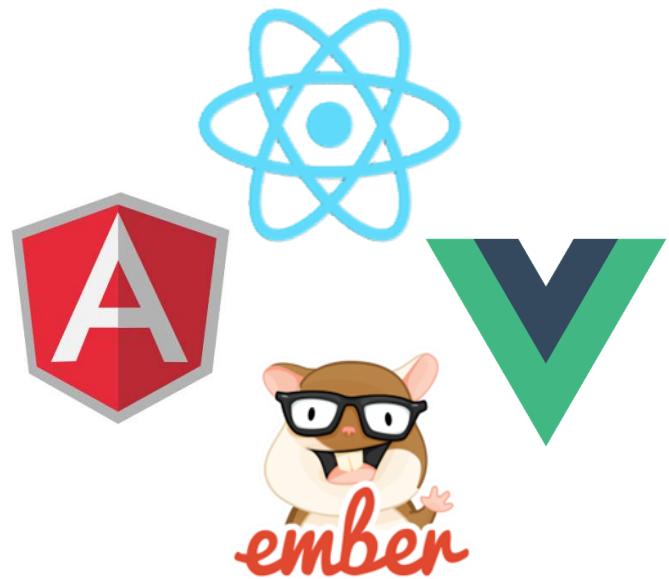
## Tradeoff Standard vs Non-standard



Standard



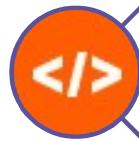
Non-standard





## Templates

Inert, reusable markup



## Custom Elements

Define our own elements



## Shadow DOM

Encapsulated styling



## Imports

Bundle HTML, JS, & CSS



## LIBRARY

Home

Browse

Paths

Mentors

Channels

Bookmarks

Notes

# 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

Live mentoring

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

This course is part of: HTML5 Path

[Expand all](#)

Five Problems, One Solution

23m 52s

Templates

27m 50s

Custom Elements

43m 34s

Shadow DOM Fundamentals

34m 19s

Shadow DOM Insertion Points &amp; Events

53m 34s

Shadow DOM Styling

51m 34s

## Course author



Cory House

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.

## Course info

Level Beginner

Rating (371)

My rating

Duration 5h 3m

Released 9 Jan 2015

## Share course



Why not use the platform?



# Why Not Web Components?



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



Can I use

template

?

[Settings](#)

2 results found

# 

## HTML templates - LS

Global

78.66%

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

[Notes](#)[Known issues \(0\)](#)[Resources \(7\)](#)[Feedback](#)

No notes



## Can I use

## html imports

[Settings](#)

1 result found

# [HTML Imports](#) - WD

Global

56.46%

Method of including and reusing HTML documents in other HTML documents.

[Notes](#)[Known issues \(0\)](#)[Resources \(10\)](#)[Feedback](#)

MS Edge status: Under Consideration

Firefox status: not-planned



# Can I use custom elements ?

[Settings](#)

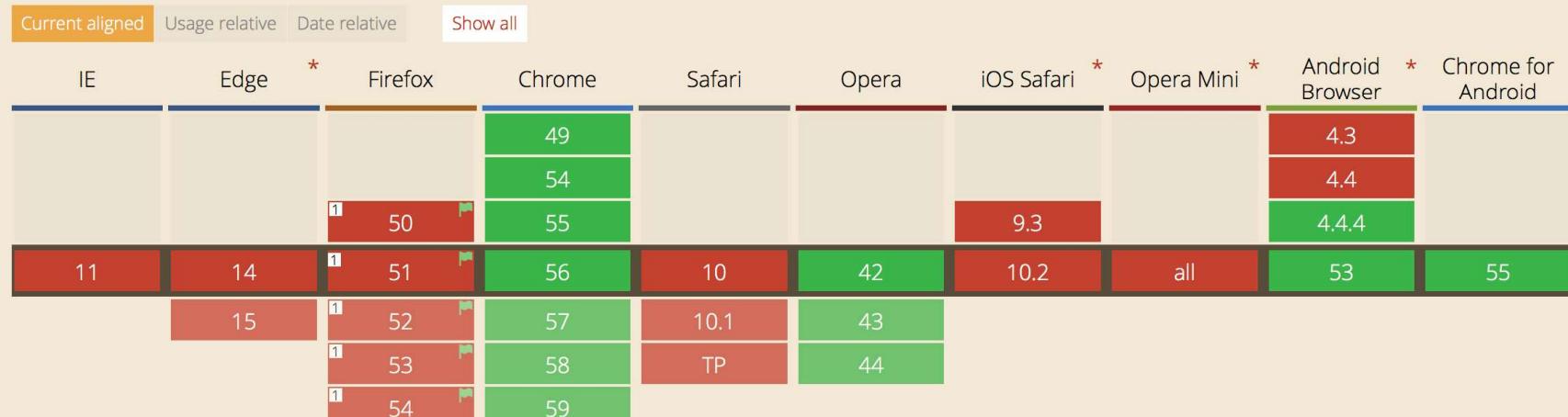
2 results found

## # Custom Elements v0 WD

Global

57.28%

Method of defining new HTML tags.

[Notes](#)[Known issues \(0\)](#)[Resources \(14\)](#)[Feedback](#)

MS Edge status: Under Consideration

Firefox status: in-development

WebKit status: Done



## Can I use

## shadow dom



Settings

2 results found

## # Shadow DOM v0 - WD

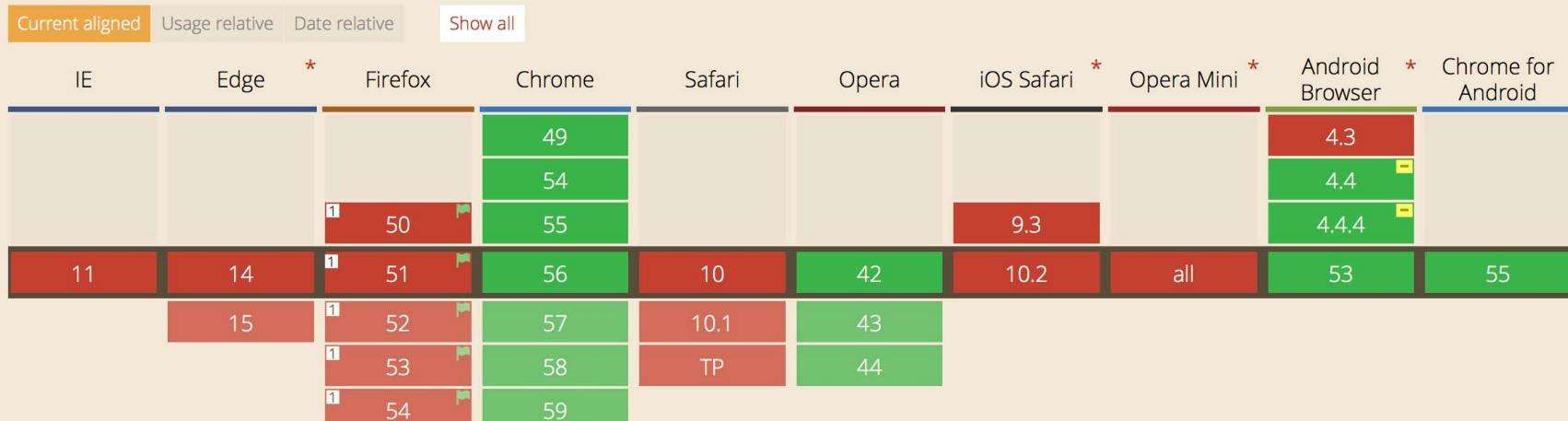
Global

58.71%

unprefixed:

52.92%

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.



Notes

Known issues (0)

Resources (10)

Feedback

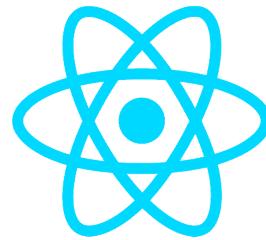


# Why Not Web Components?



**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**

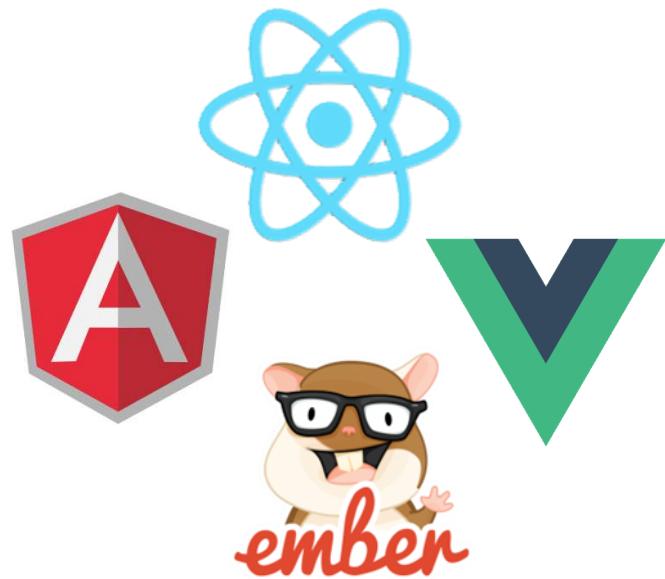
**Only run in the browser**



Standard



Non-standard



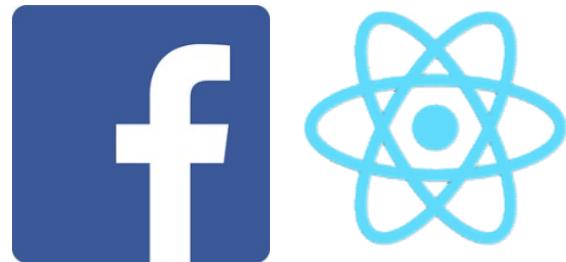
# 6

## 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  
30,000 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?

