one flew over the cuckoo's nest

@picanteverde

javascript fullstack

picanteverde.wordpress.com

github.com/picanteverde

picanteverde@toptal.com

picanteverde.com

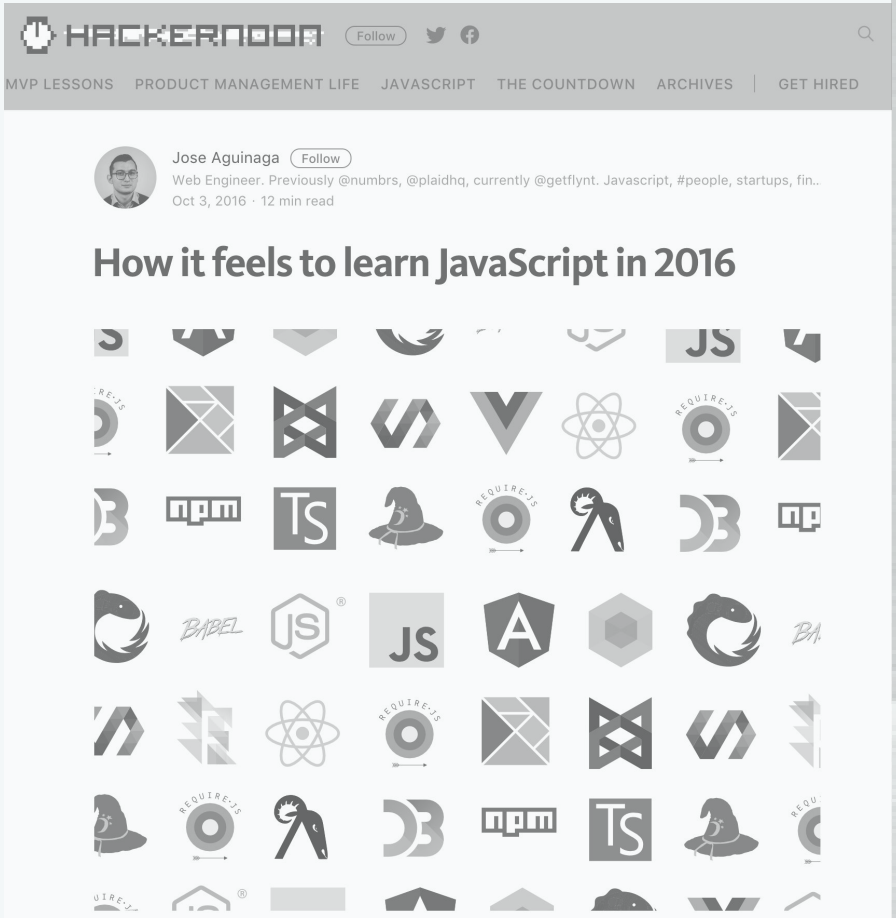
(a.k.a. alejandro hernández)

one flew over the cuckoo's nest

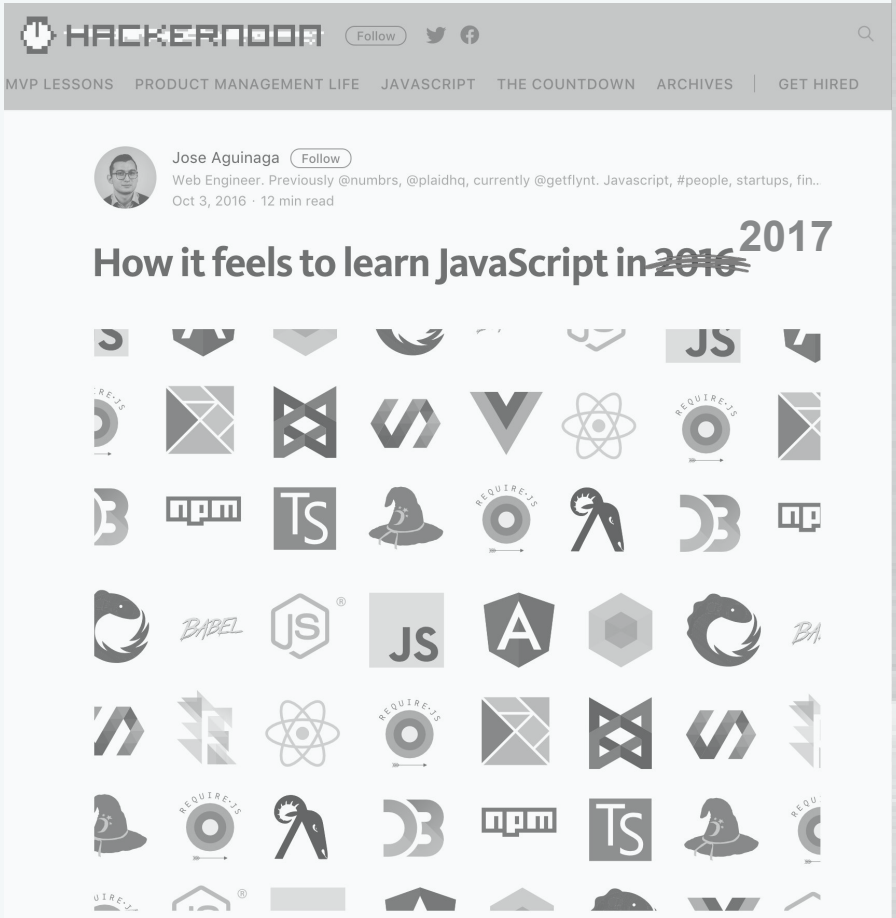

how crazy we are . . .
how crazy we'll be . . .

how crazy we are?

current state of javascript

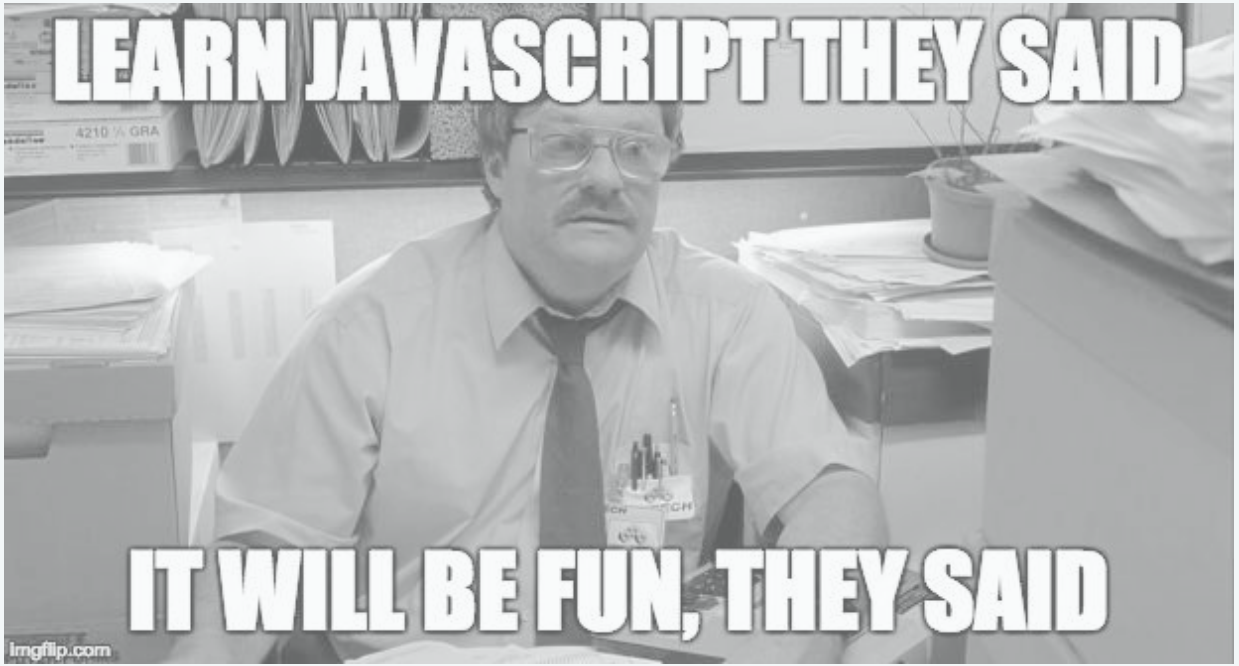


https://hackernoon.com/how-it-feels-to-learn-javascript-in-2016-d3a717dd577f#.1jvkycde9



https://hackernoon.com/how-it-feels-to-learn-javascript-in-2016-d3a717dd577f#.1jvkycde9

javascript fatigue



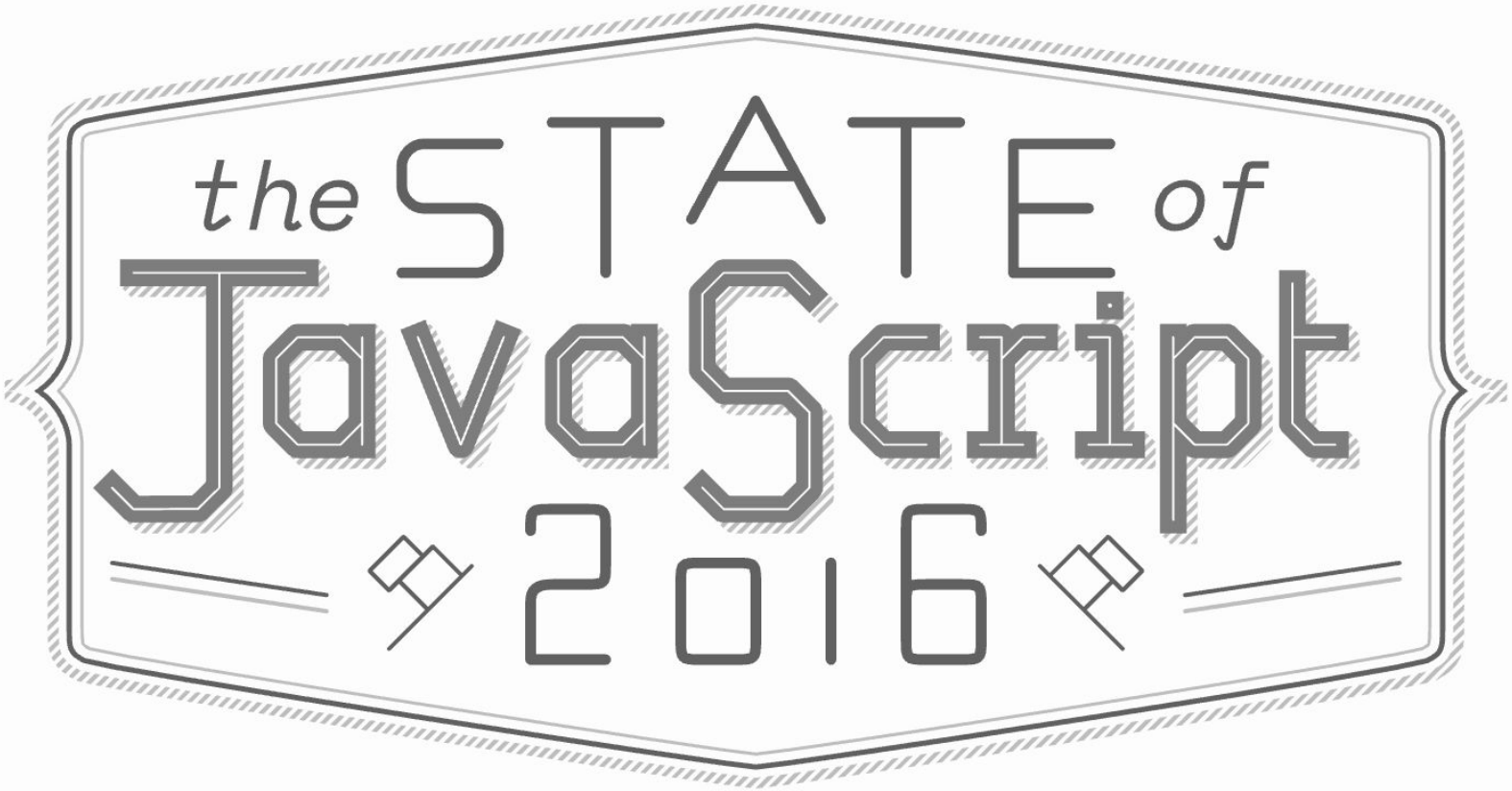
from Ken Russell's "The Devils"

The JavaScript phenomenon is a mass psychosis

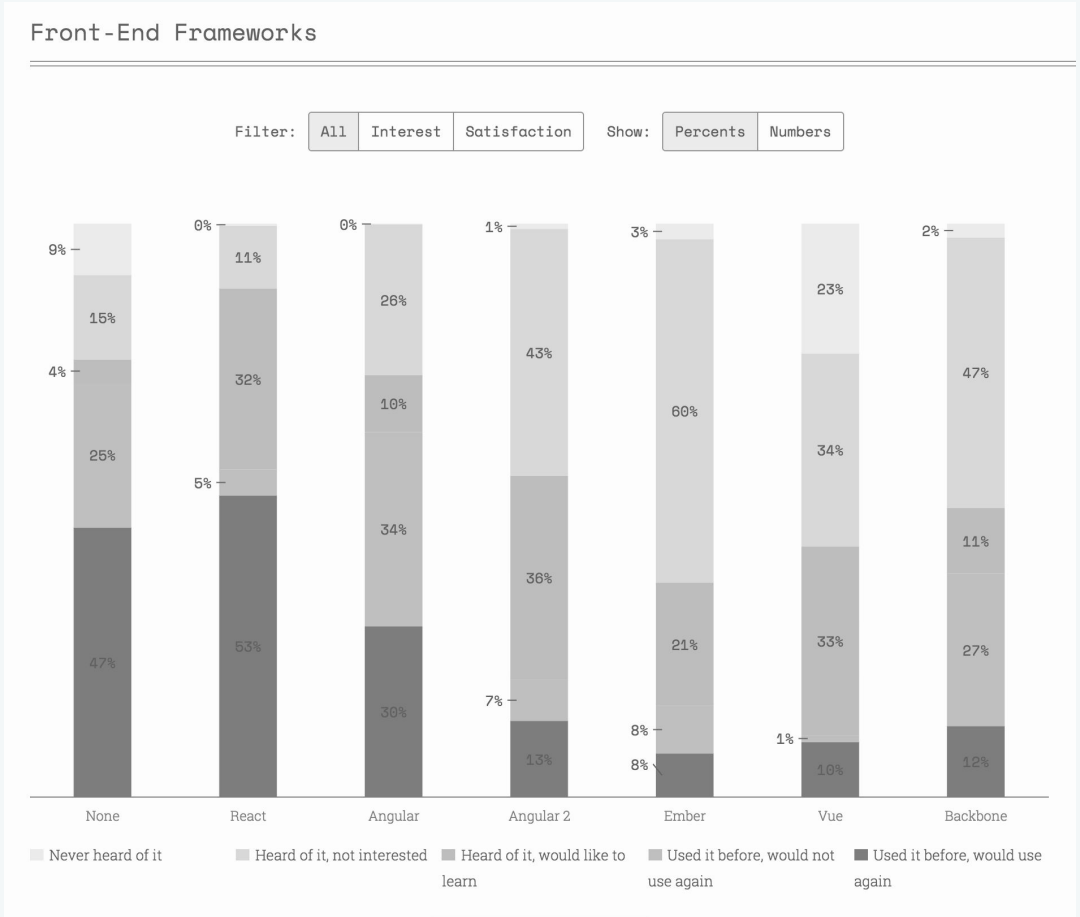
. . .

I recently received this kind message on LinkedIn from the president of a Canadian cybercrime technology company:

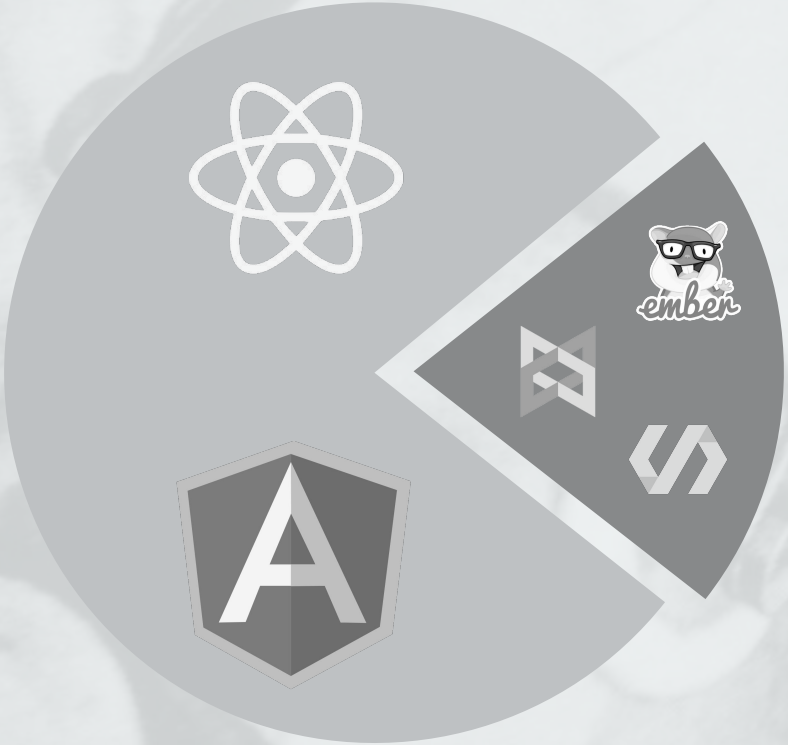
https://medium.com/@richardeng/the-javascript-phenomenon-is-a-mass-psychosis-57adebb09359

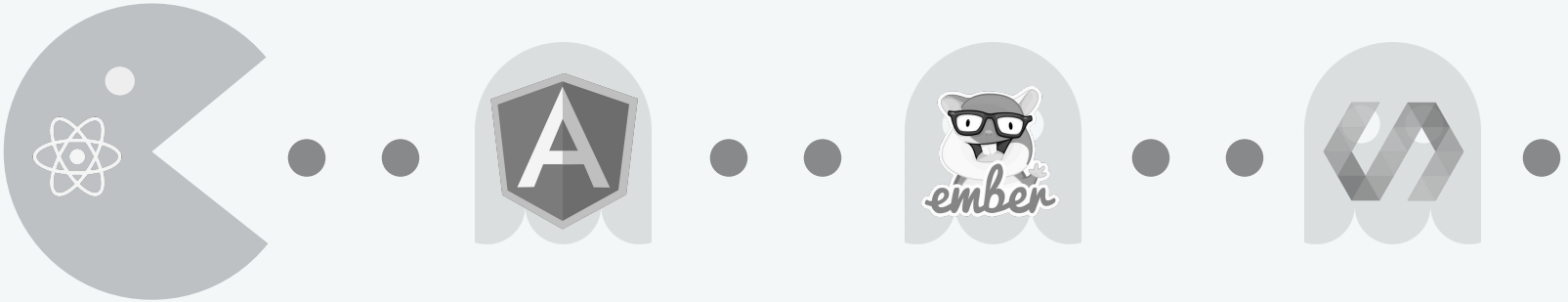


http://stateofjs.com/

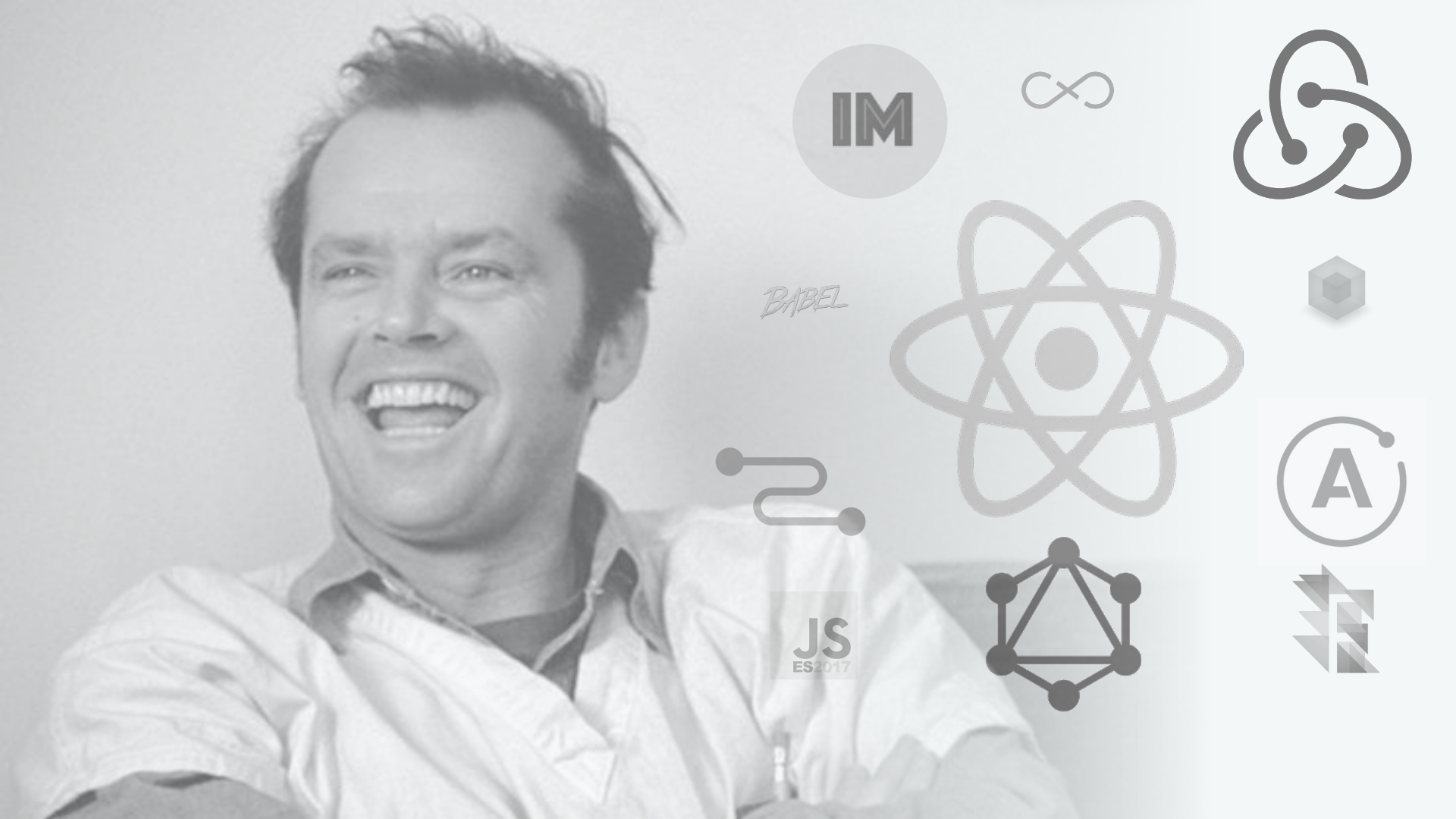

Front-End Frameworks
Filter: All Interest Satisfaction
Show: Percents Numbers
9%
15%
4%
25%
47%
0%
11%
32%
5%
53%
0%
26%
10%
34%
30%
1%
43%
36%
7%
13%
3%
60%
21%
8%
8%
23%
34%
33%
1%
10%
2%
47%
11%
27%
12%
None
React
Angular
Angular 2
Ember
Vue
Backbone
Never heard of it
Heard of it, not interested
Heard of it, would like to learn
Used it before, would not use again
Used it before, would use again


polymer.js


ember


ember


react.js


IM
BABEL
JS
ES2017
A

https://www.youtube.com/watch?v=ZCuYPiUlONs

~~this.setState({...})~~

vs.

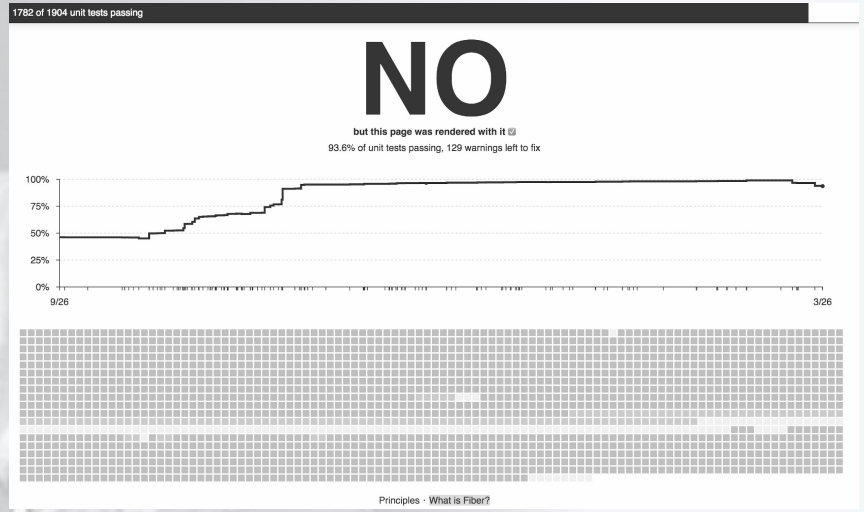
```
this.setState((state, props) => {
  return {...}
})
```

16:39 / 31:47

start update 1 componentWillUpdate

start update 2 componentWillUpdate
componentDidUpdate

27:50 / 31:47



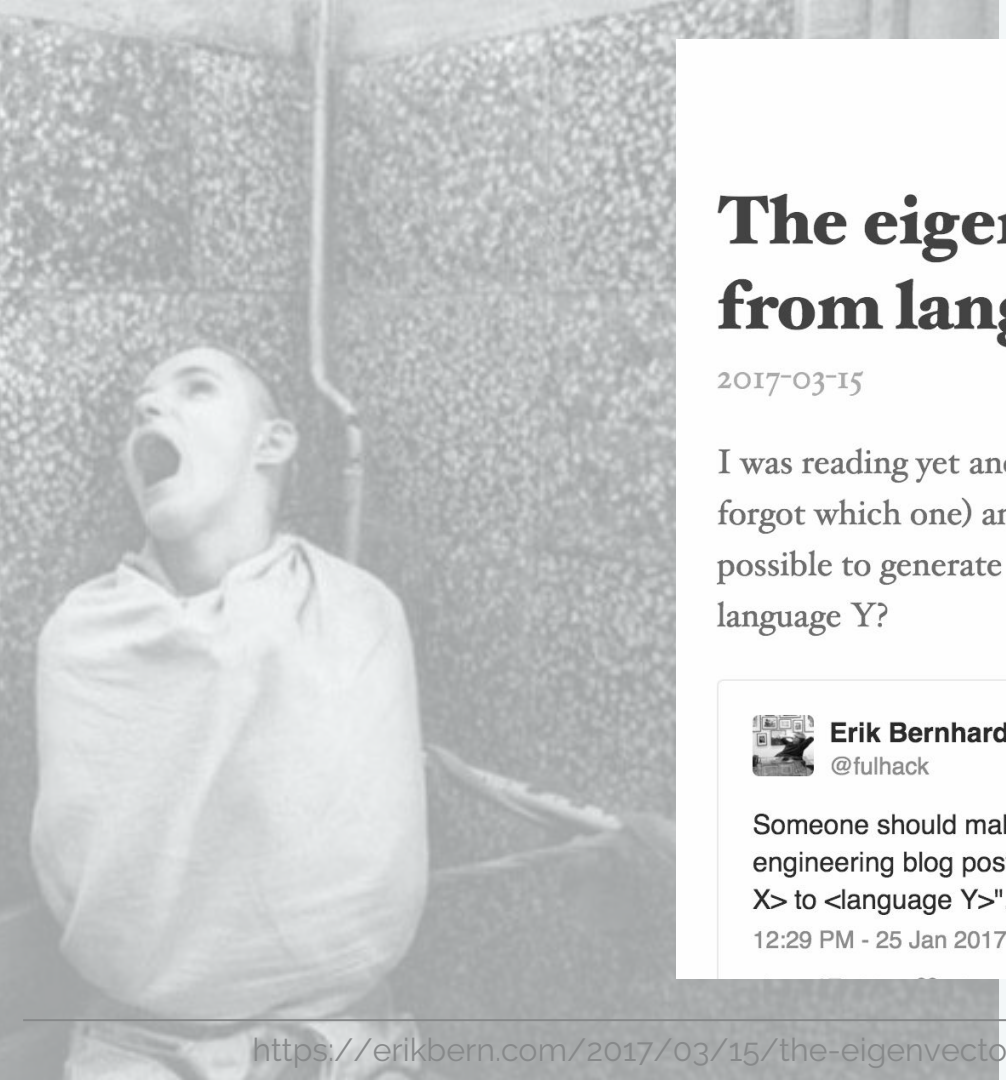
http://isfiberreadyyet.com/

Why we chose Vue.js over React

10 DECEMBER 2016

Qwintry team recently started active migration to Vue.js as a frontend framework in all our legacy and new projects:

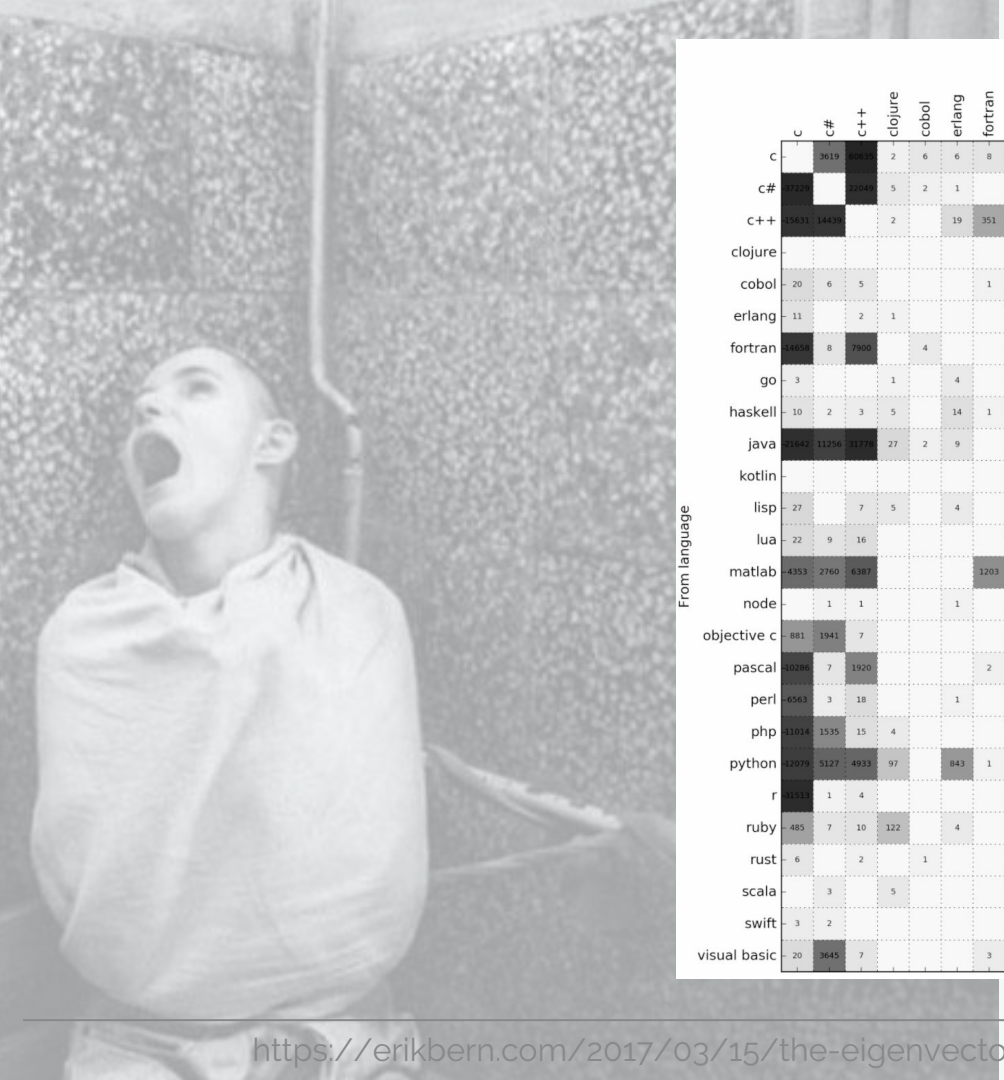
- in legacy Drupal system (qwintry.com)
- in our new, completely rewritten qwintry.com branch
- in Yii2-powered b2b system (logistics.qwintry.com)
- in all our smaller internal and external projects (mostly with PHP and Node.js backends)

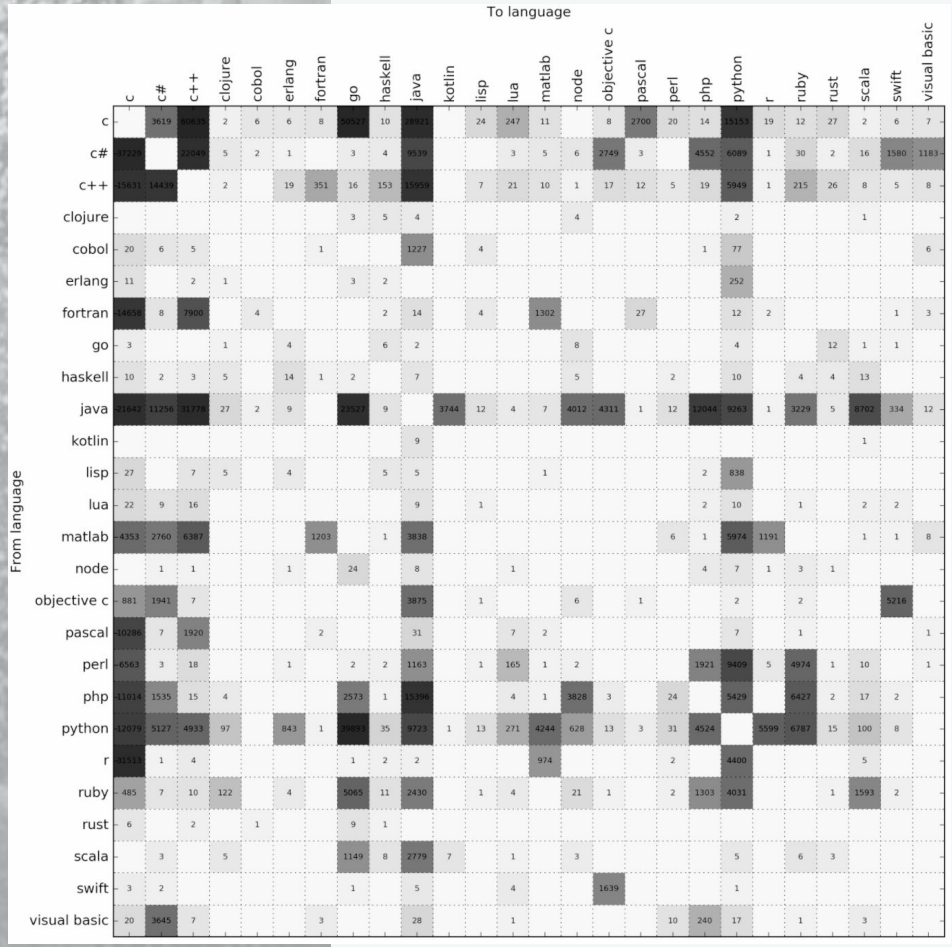


The eigenvector of "Why we moved from language X to language Y"

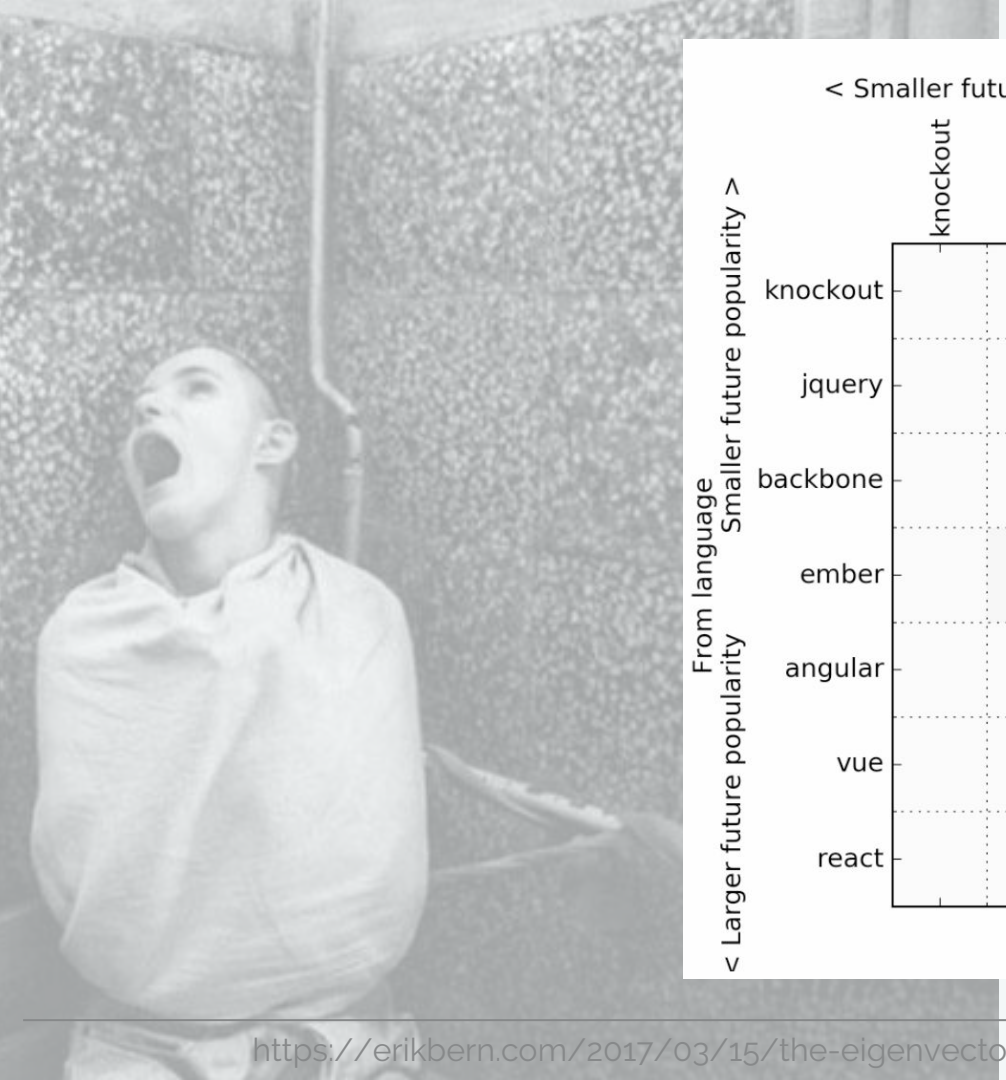
2017-03-15

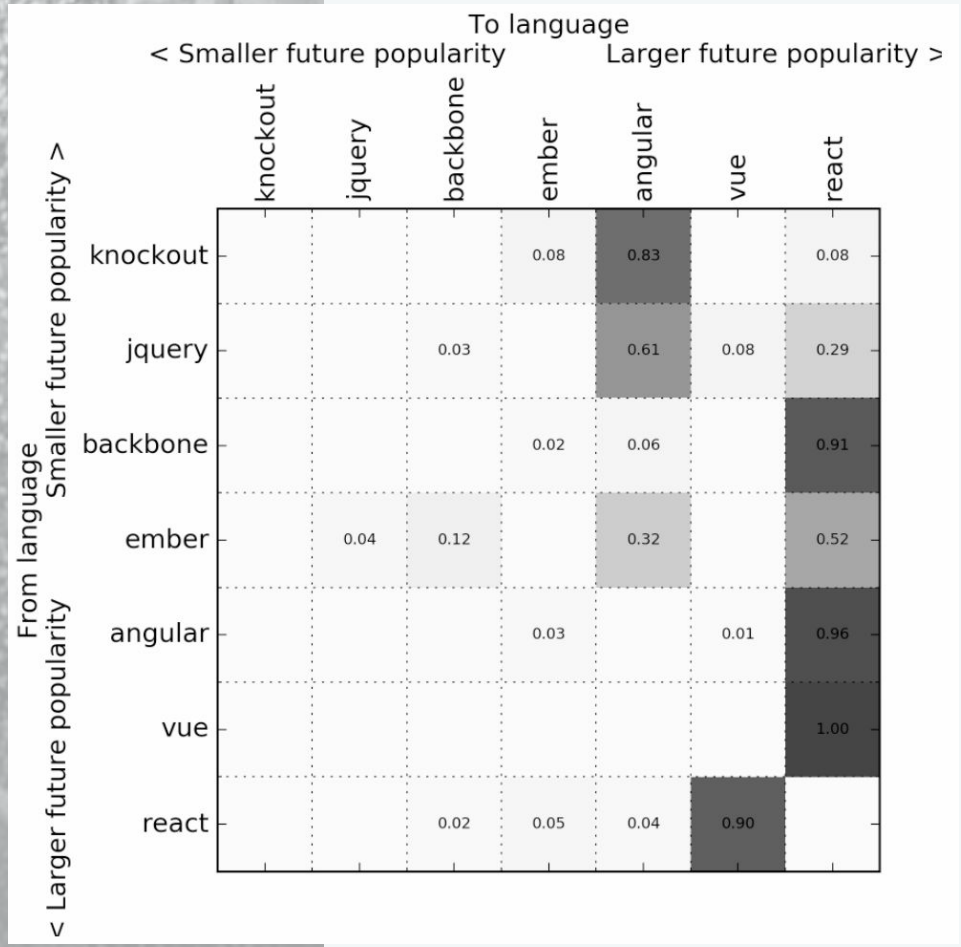
I was reading yet another blog post titled “Why our team moved from to " (I forgot which one) and I started wondering if you can generalize it a bit. Is it possible to generate a N * N contingency table of moving from language X to language Y?





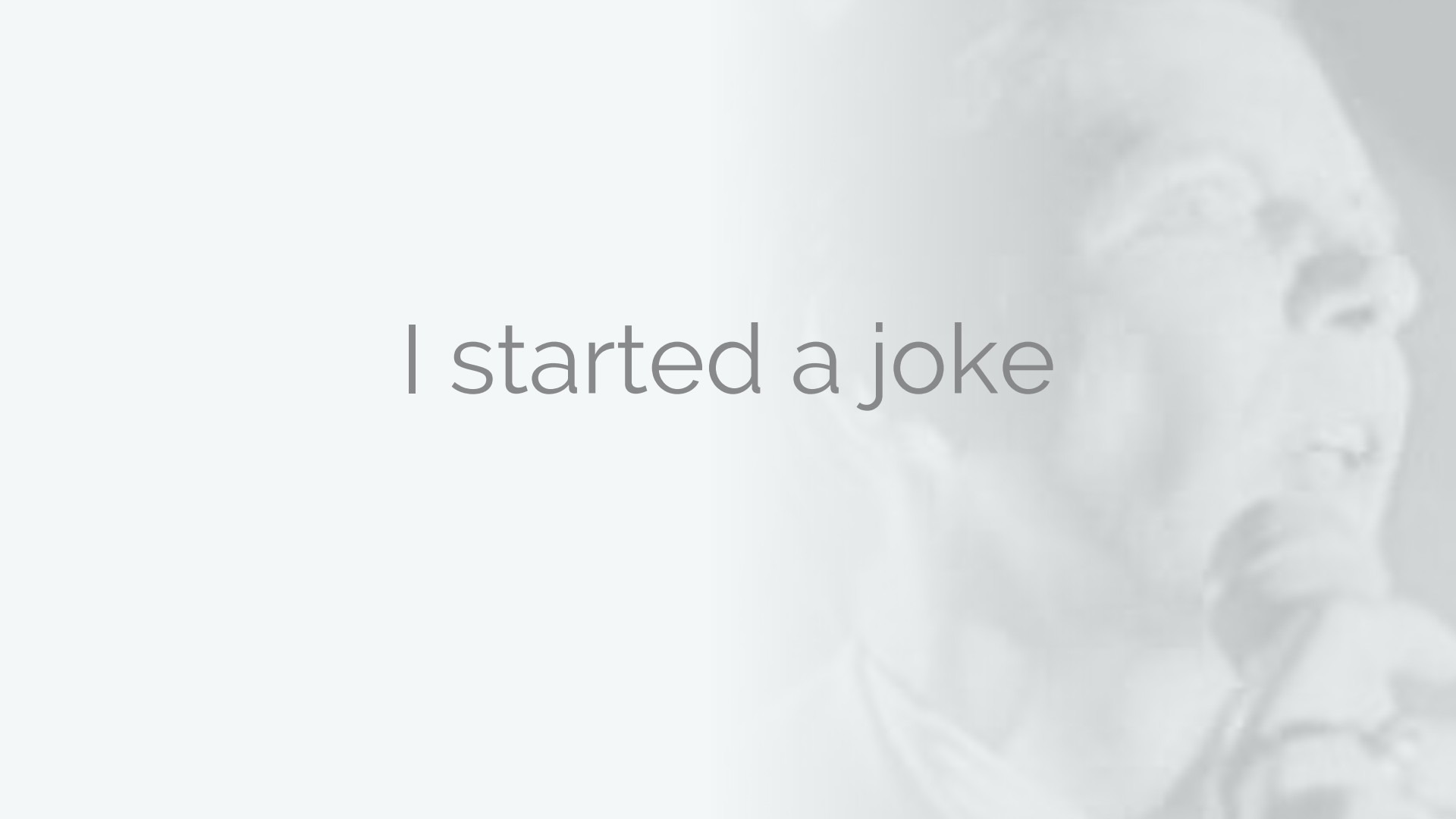
https://erikbern.com/2017/03/15/the-eigenvector-of-why-we-moved-from-language-x-to-language-y.html





https://erikbern.com/2017/03/15/the-eigenvector-of-why-we-moved-from-language-x-to-language-y.html

I started a joke



This time you have definitely chosen the right libraries and build tools

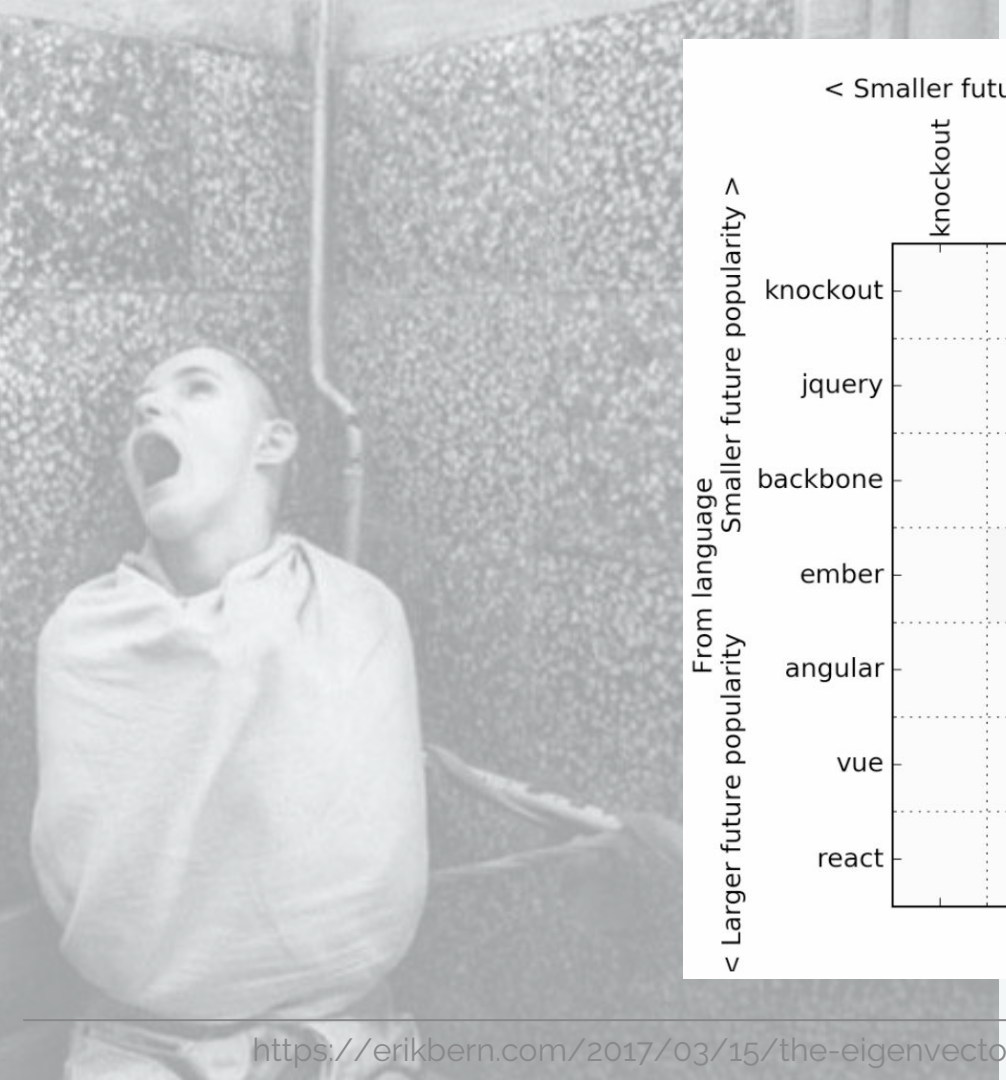
Real World

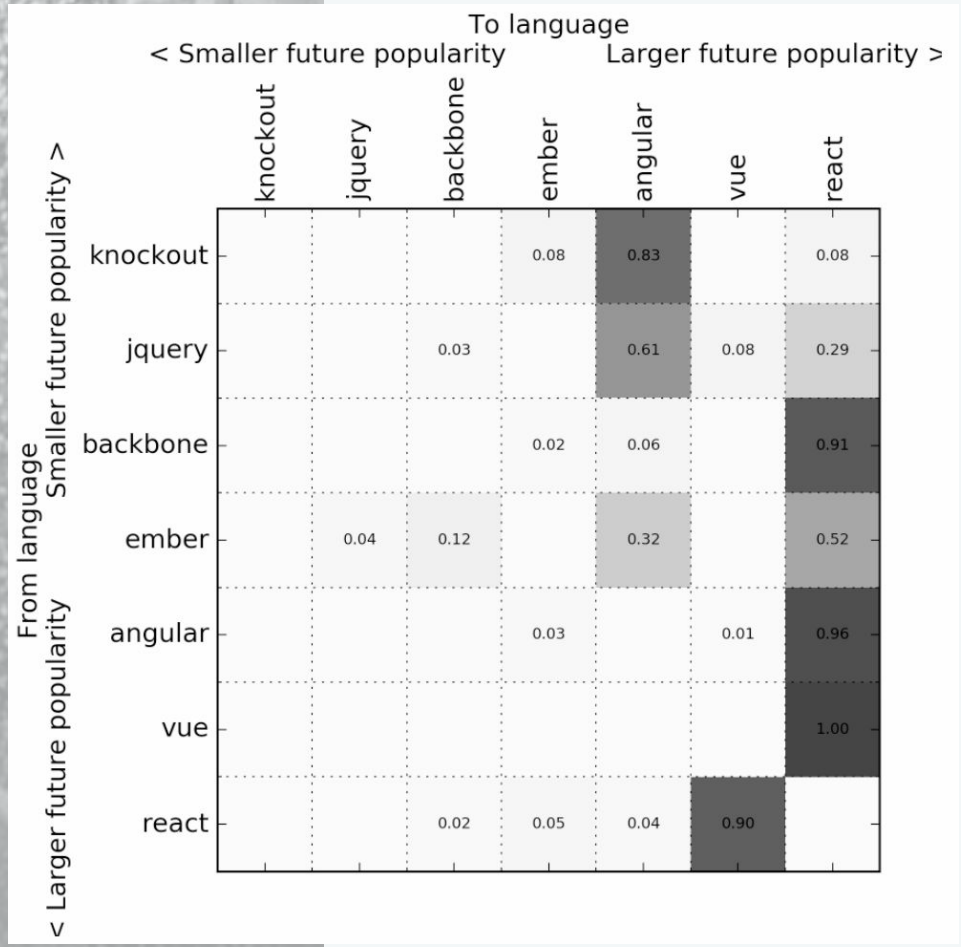
Rewriting Your Front End Every Six Weeks

O RLY?

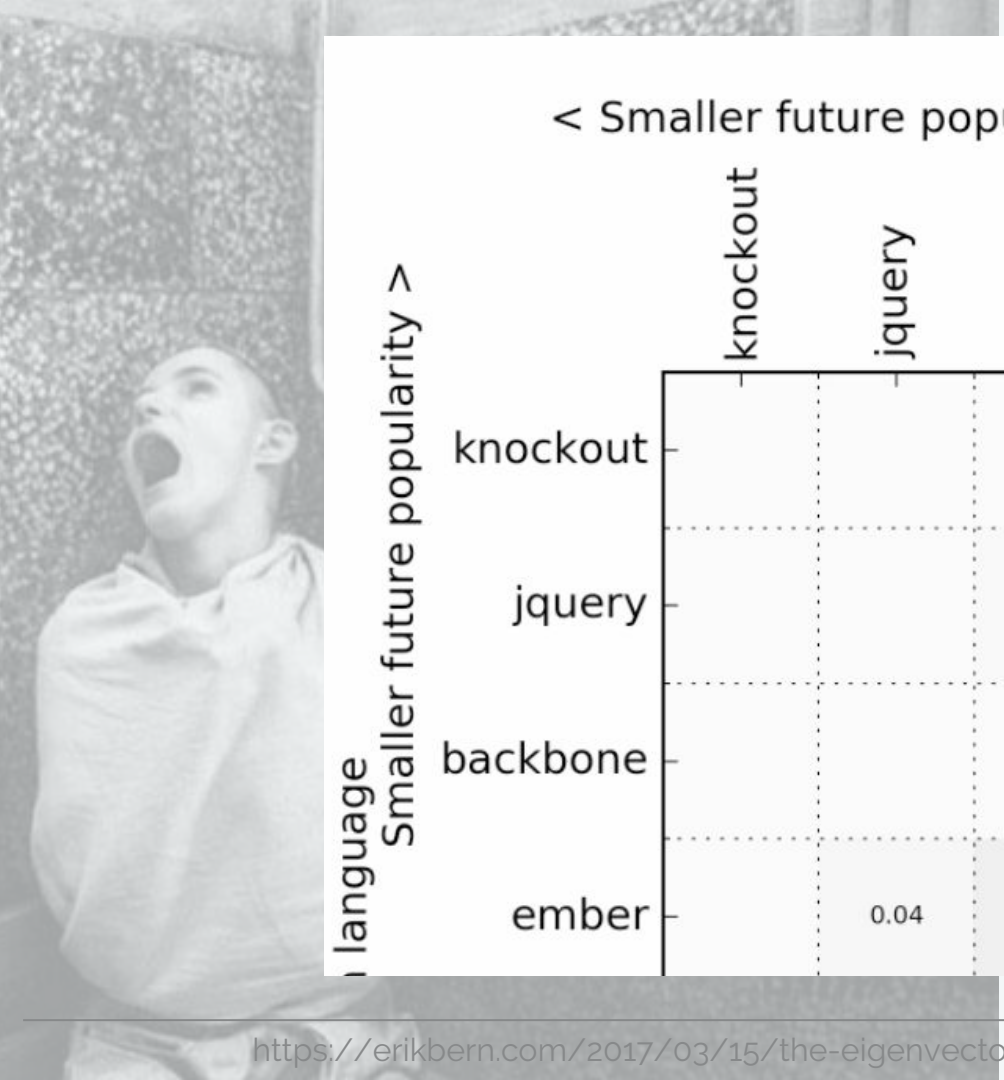
@ThePracticalDev

you have to rewrite your front end every six months





https://erikbern.com/2017/03/15/the-eigenvector-of-why-we-moved-from-language-x-to-language-y.html



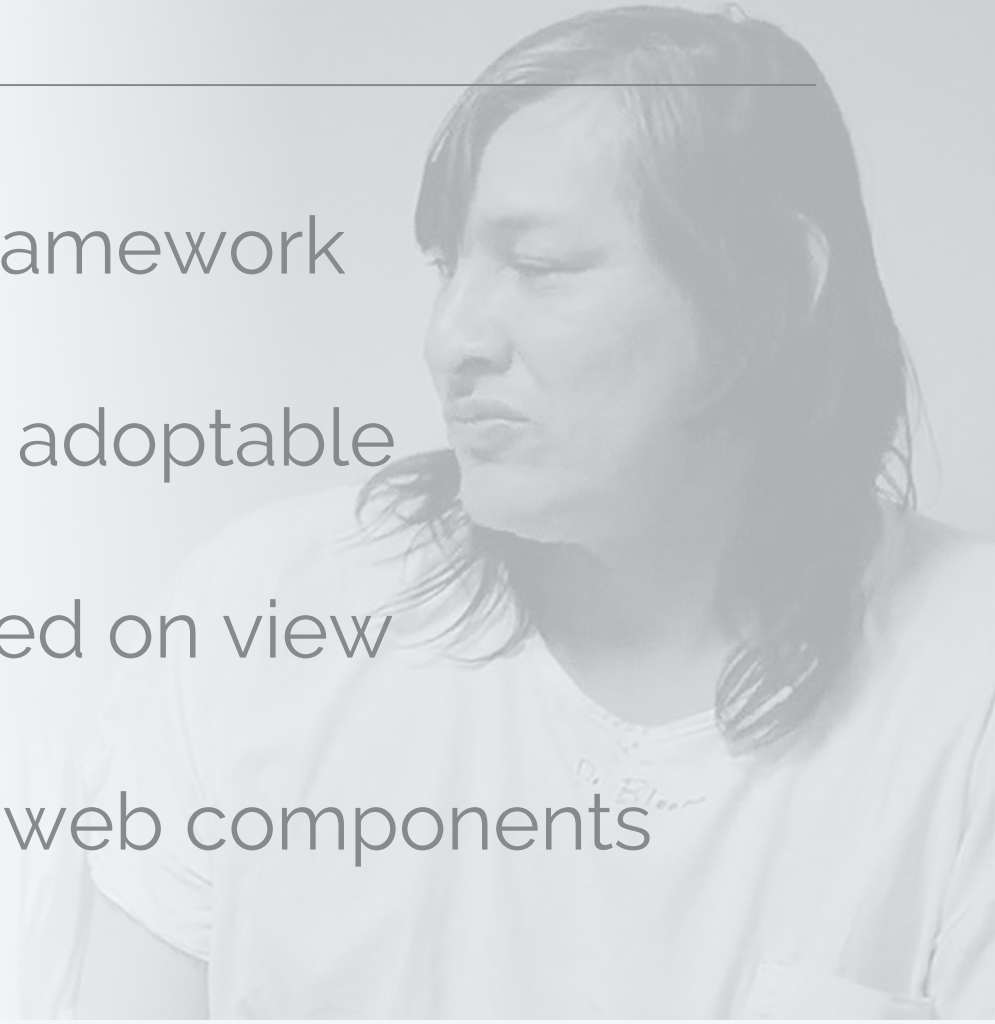
https://erikbern.com/2017/03/15/the-eigenvector-of-why-we-moved-from-language-x-to-language-y.html

- how crazy we are . . .
- how crazy we'll be . . .


React
Vue

vue.js

- progressive framework
- incrementally adoptable
- core lib focused on view
- virtual dom & web components

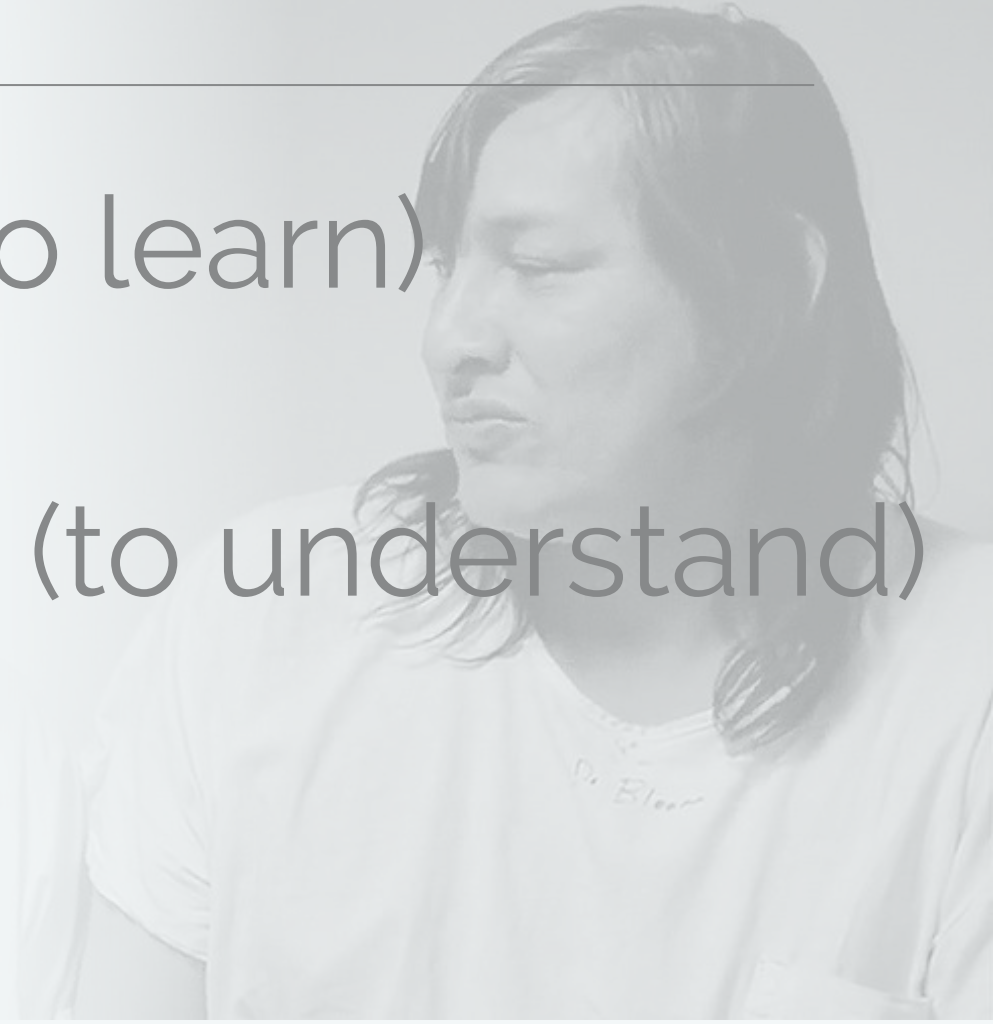


vue.js

- easy
- simple

vue.js

- easy (to learn)
- simple (to understand)



vue.js

- declarative rendering
- virtual dom
- web components

app

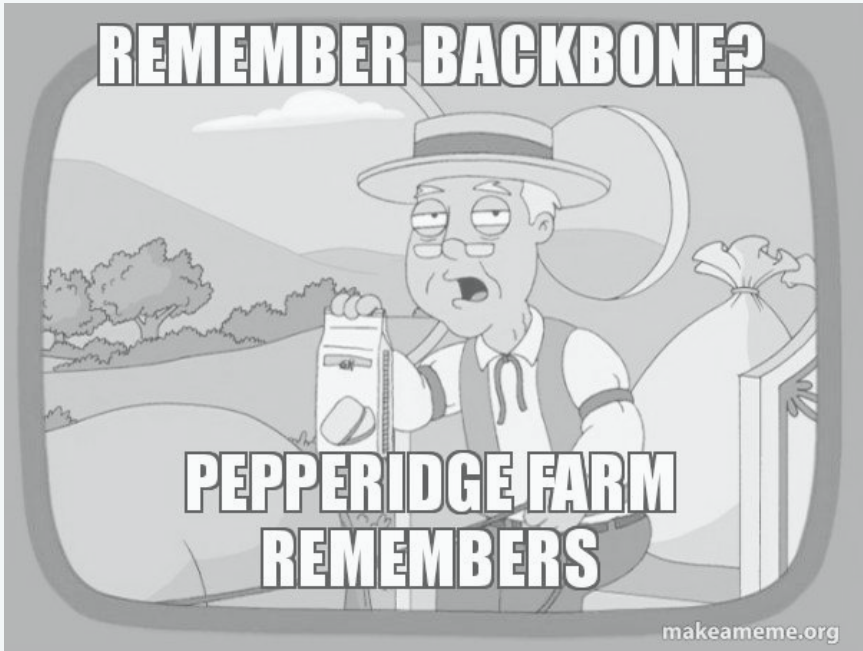
```
<div id="app">
 {{ message }}
</div>
```

```
var app = new Vue({
 el: '#app',
 data: {
   message: 'Hello Vue!'
 }
})
```

app

```
<div id="app">
 {{ message }}
</div>
```

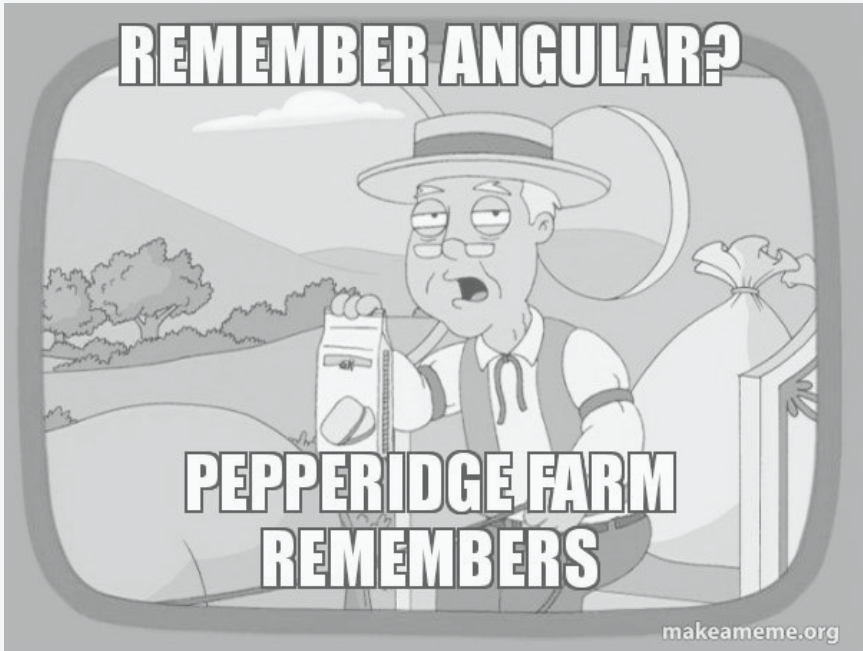
```
var app = new Vue({
 el: '#app',
 data: {
   message: 'Hello Vue!'
 }
})
```



directives

```
<span
   v-bind:title="tooltip">
   hover the mouse
</span>
```

```
var app = new Vue({
 el: '#app',
 data: {
   tooltip: 'tooltip text!'
 }
})
```

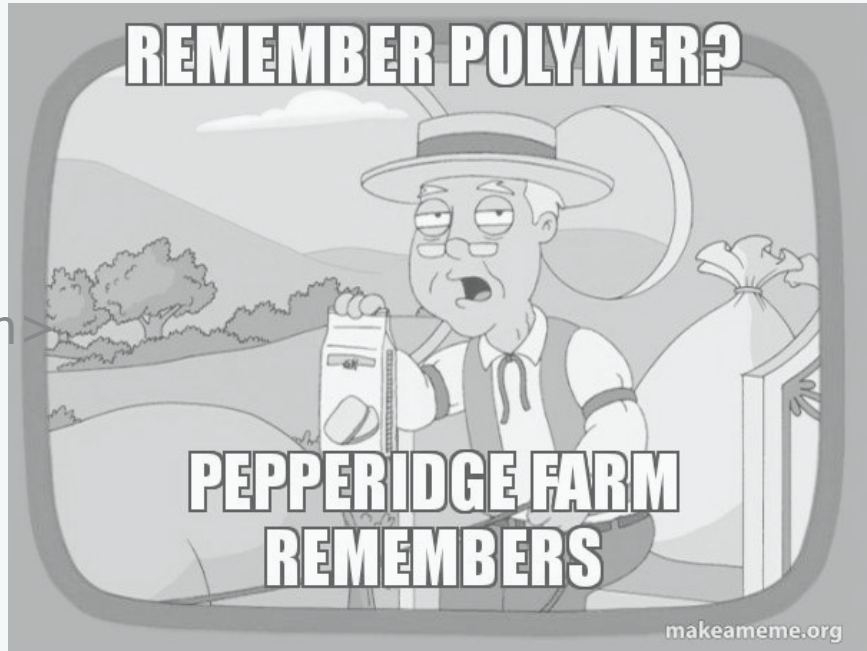


demo

components

```
Vue.component('todo-item', {
 template: '<li>This is a todo</li>'
})
```

```
<ol>
 <todo-item></todo-item>
</ol>
```



props

```
Vue.component('todo-item', {
 props: ['todo'],
 template: '<li>{{ todo.text }}</li>'
})

 <ol>
   <todo-item v-for="item in groceryList"
v-bind:todo="item"></todo-item>
 </ol>
```

props

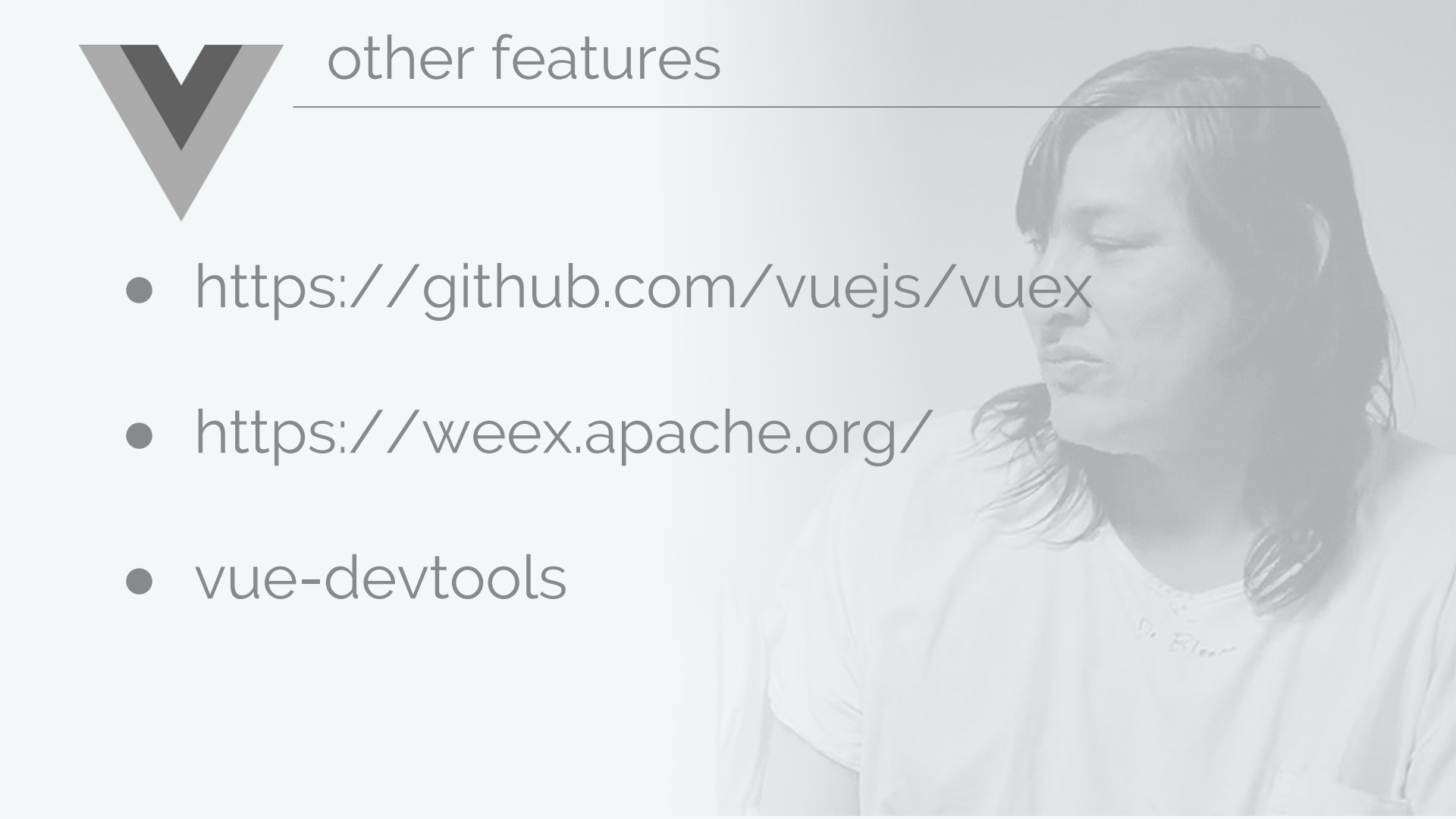
```
Vue.component('todo-item', {
 props: ['todo'],
 template: '<li>{{ todo.text }}</li>'
})

 <ol>
   <todo-item v-for="item in groceryList"
v-bind:todo="item"></todo-item>
 </ol>
```

demo

vue-cli

```
$ npm install -g vue-cli
$ vue init webpack my-project
? Project name my-project
? Project description A Vue.js project
? Author Alejandro Hernández <picanteverde@gmail.com>
? Vue build runtime
? Install vue-router? Yes
? Use ESLint to lint your code? Yes
? Pick an ESLint preset AirBNB
? Setup unit tests with Karma + Mocha? Yes
? Setup e2e tests with Nightwatch? Yes
$ cd my-project
$ npm install
$ npm run dev
```



other features

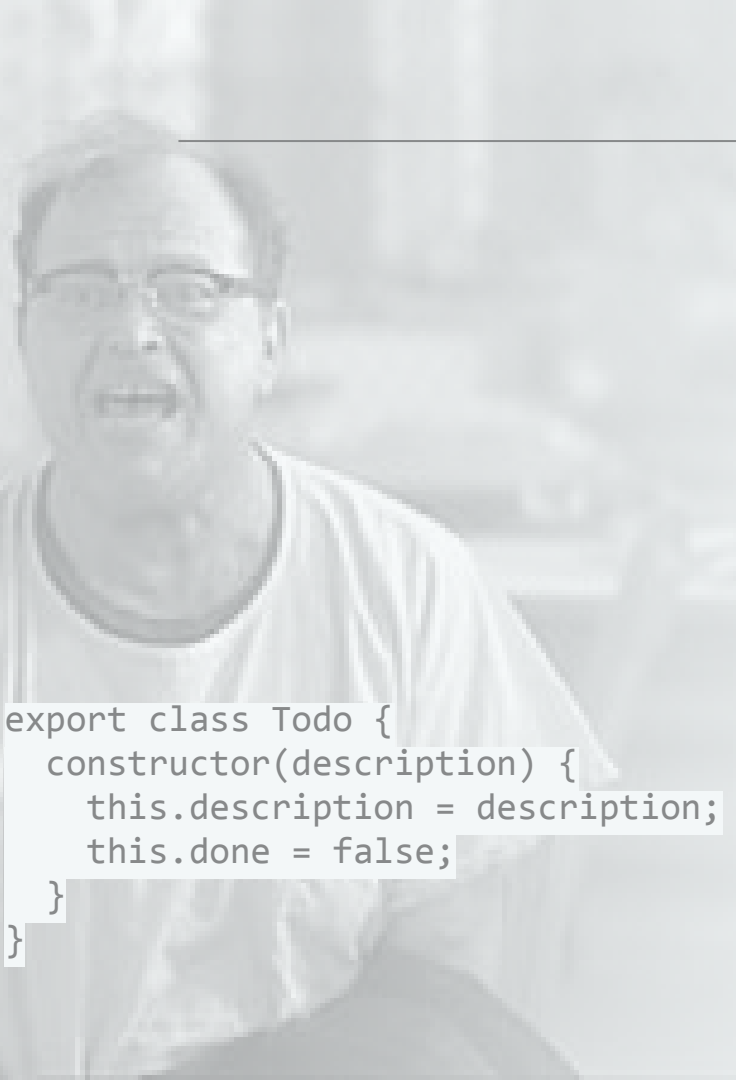
- https://github.com/vuejs/vuex
- https://weex.apache.org/
- vue-devtools

aurelia.io

- modern architecture
- two-way data binding
- custom elements
- typescript

- mv-vm
- es2015 classes
- stay out of the way

aurelia.io



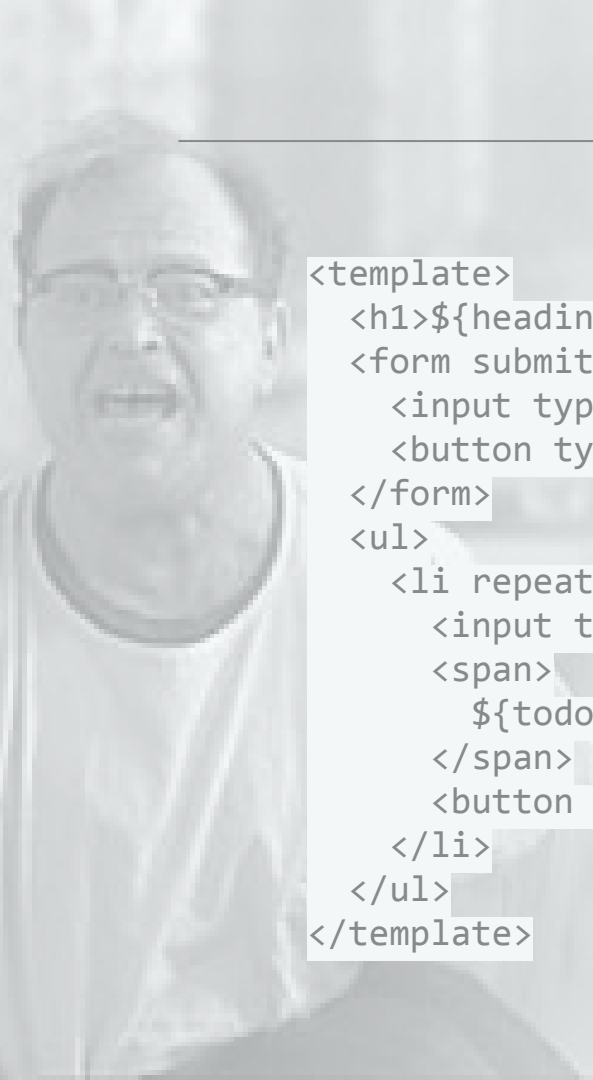
```
export class Todo {
  constructor(description) {
    this.description = description;
    this.done = false;
  }
}
```

```
import {Todo} from './todo';
export class App {
  constructor() {
    this.heading = "Todos";
    this.todos = [];
    this.todoDescription = '';
  }

  addTodo() {
    if (this.todoDescription) {
      this.todos.push(new Todo(this.todoDescription));
      this.todoDescription = '';
    }
  }

  removeTodo(todo) {
    let index = this.todos.indexOf(todo);
    if (index !== -1) {
      this.todos.splice(index, 1);
    }
  }
```

aurelia.io



```
<template>
  <h1>${heading}</h1>
  <form submit.trigger="addTodo()">
    <input type="text" value.bind="todoDescription">
    <button type="submit">Add Todo</button>
  </form>
  <ul>
    <li repeat.for="todo of todos">
      <input type="checkbox" checked.bind="todo.done">
      <span>
        ${todo.description}
      </span>
      <button click.trigger="removeTodo(todo)">Remove</button>
    </li>
  </ul>
</template>
```

demo

using basic aurelia project setup

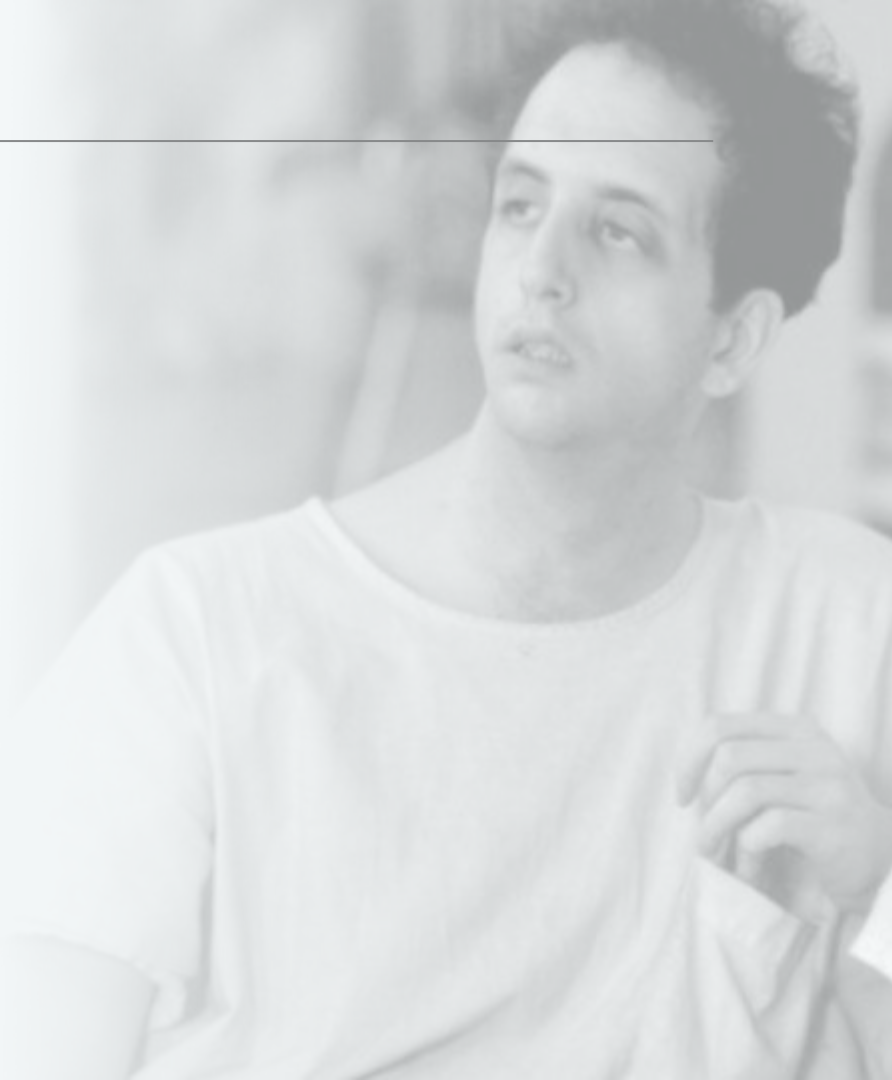
aurelia-cli

```
$ npm install aurelia-cli -g
$ au new
configs..
Installs..
$ cd proj-name
$ au run --watch
```

demo

mithril.js

- virtual dom
- fast (8k +perf)
- simple



mithril.js

Download size

Mithril (8kb)

Vue + Vue-Router + Vuex + fetch (40kb)

React + React-Router + Redux + fetch (64kb

Angular (135kb)

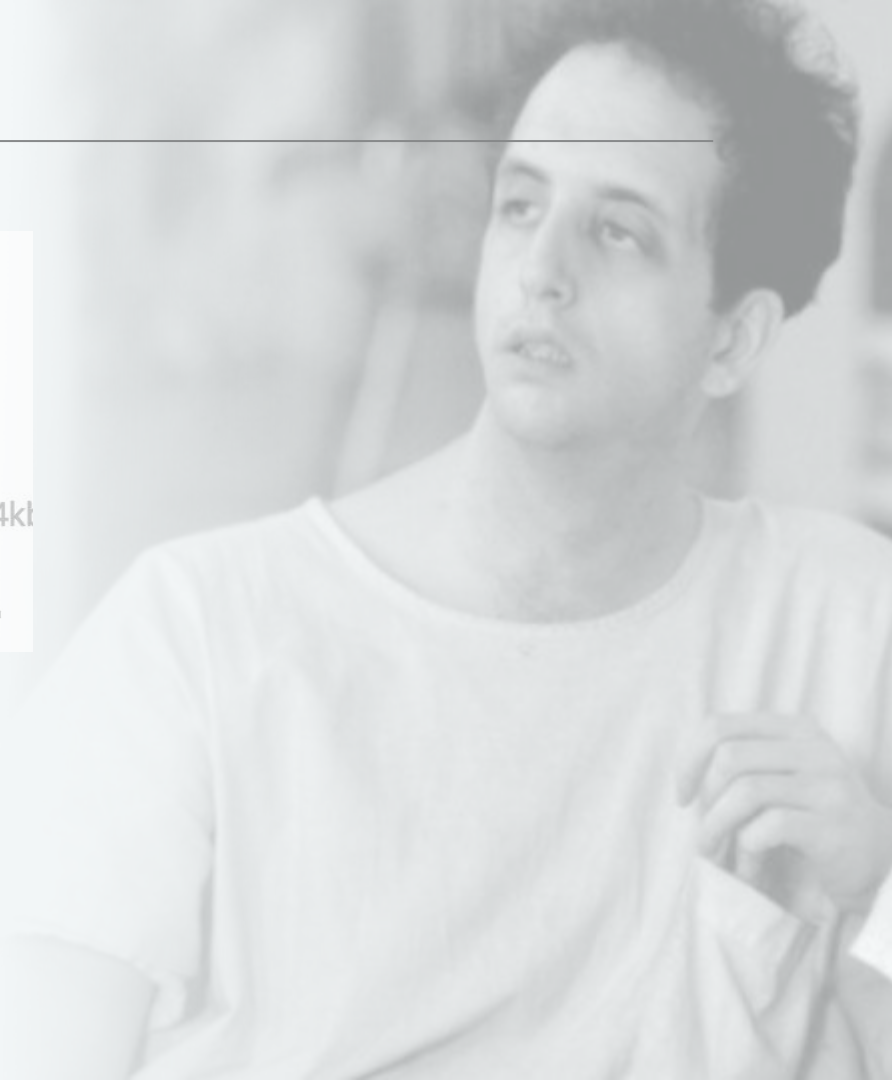
Performance

Mithril (6.4ms)

Vue (9.8ms)

React (12.1ms)

Angular (11.5ms)

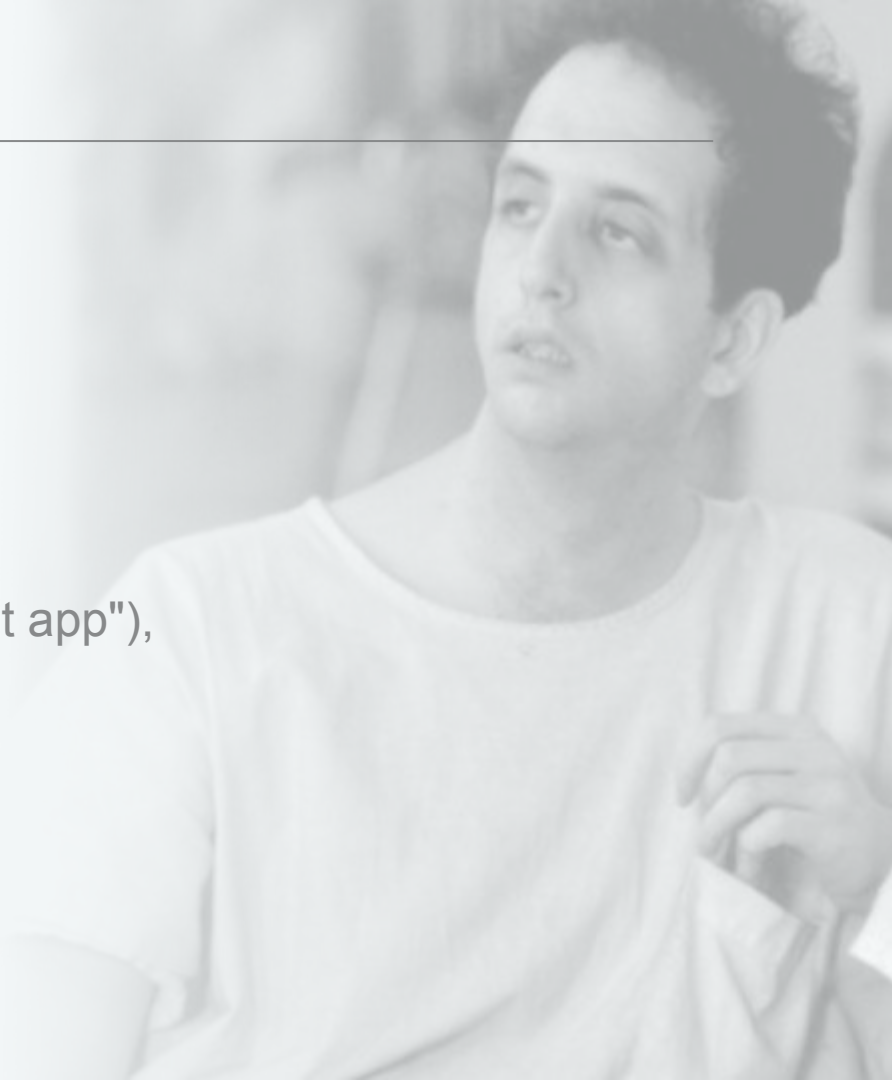


virtual dom

```
var root = document.body

m.render(root, "Hello world");

m.render(root, m("main", [
      m("h1", {class: "title"}, "My first app"),
      m("button", "A button"),
    ]));
```

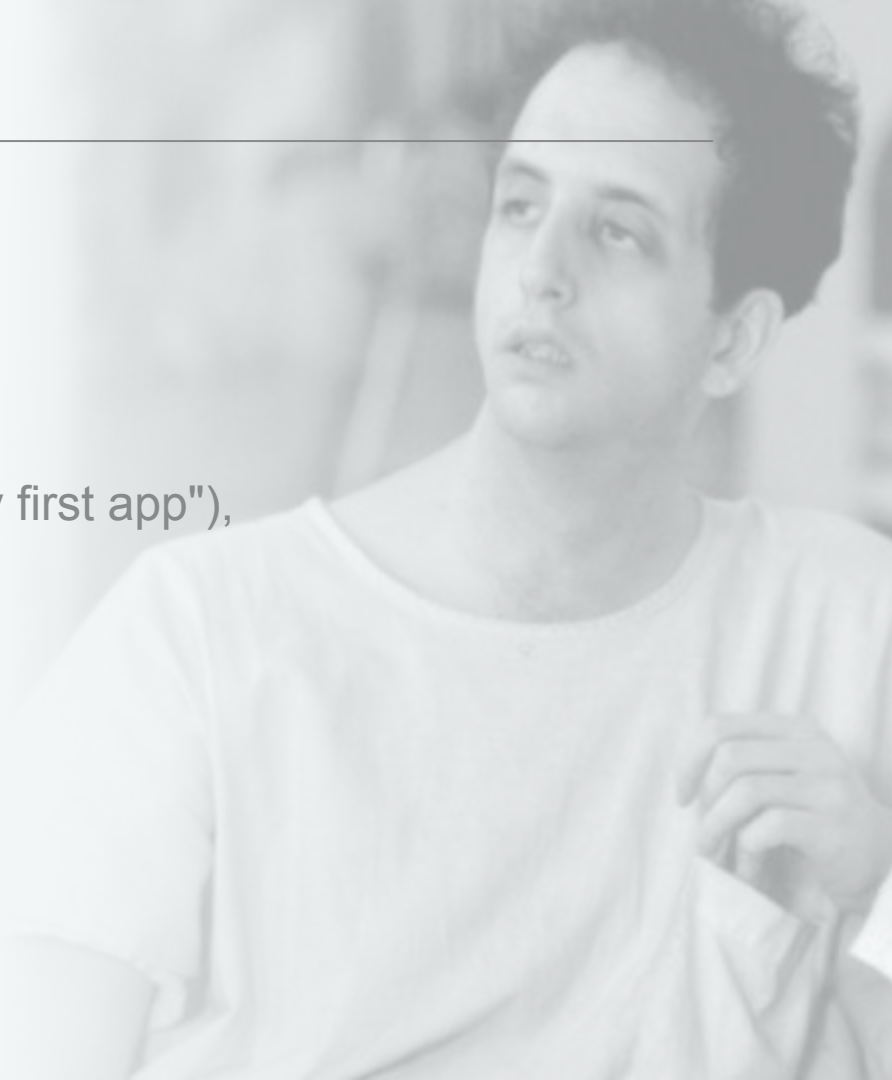


virtual dom

demo

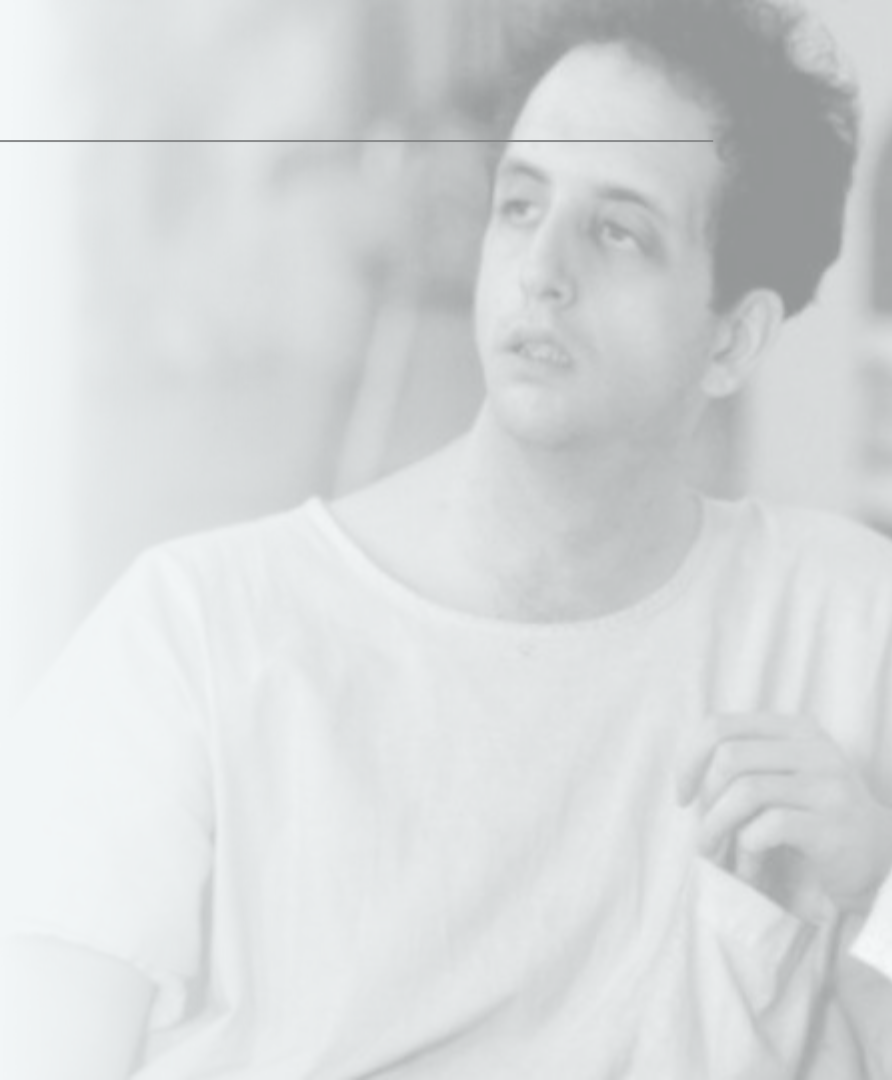
components

```
var Hello = {
    view: function() {
        return m("main", [
            m("h1", {class: "title"}, "My first app"),
            m("button", "A button"),
        ])
    }
}
m.mount(root, Hello)
```



router

```
m.route(root, "/splash", {
    "/splash": Splash,
    "/hello": Hello,
})
```

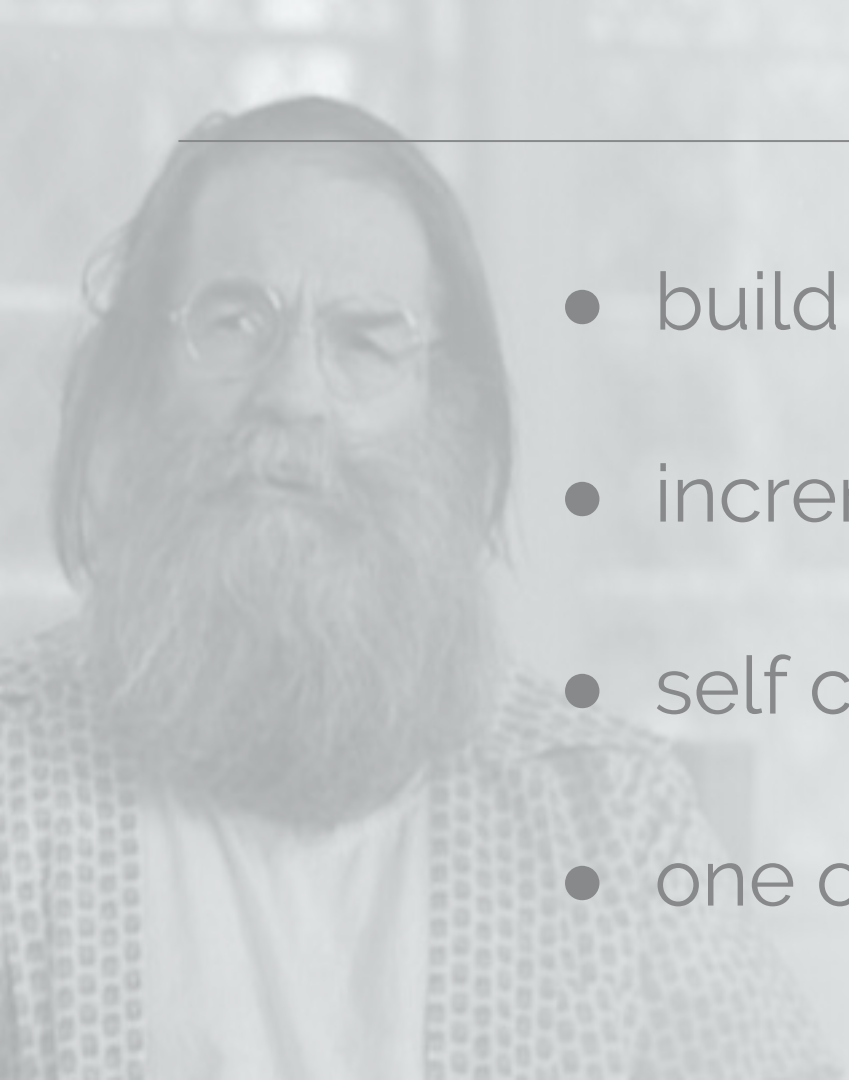


virtual dom

demo

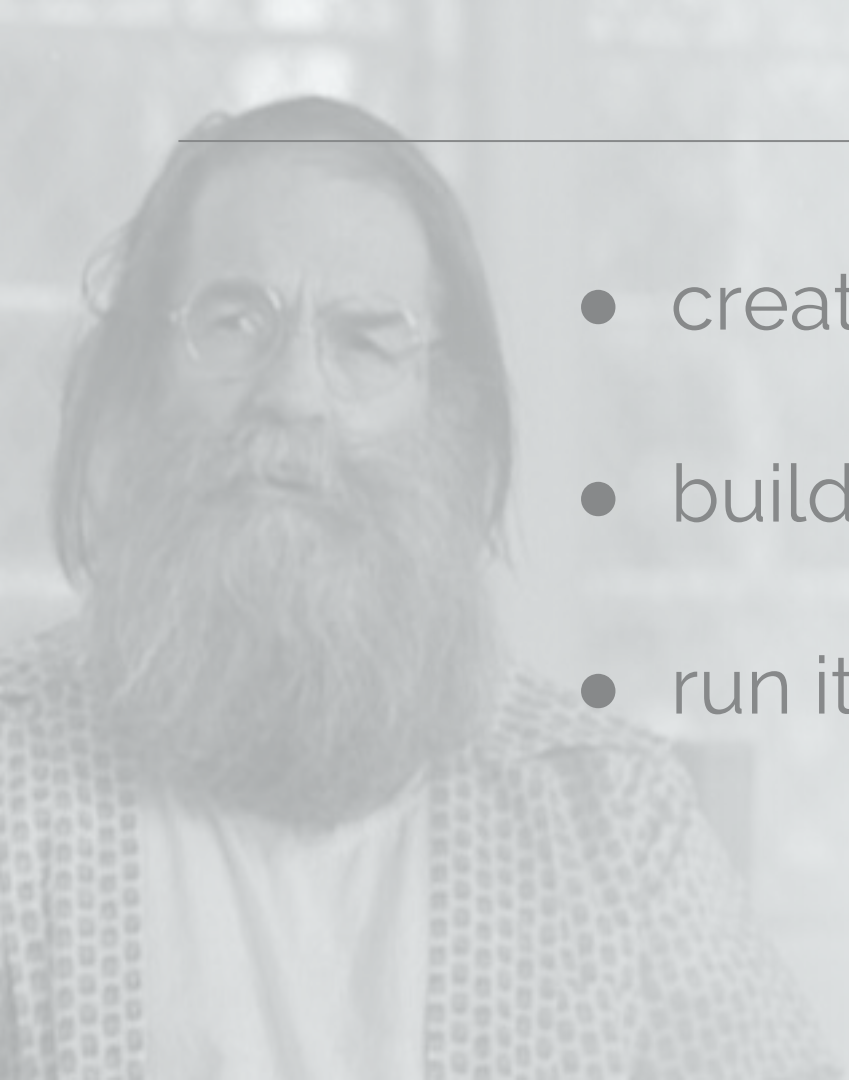
install

```
$ npm init -y
$ npm install mithril --save
$ npm install webpack --save
#  add       "start": "webpack src/index.js bin/app.js --watch"
$ mkdir src
$ touch src/index.js
$ echo "<script src="bin/app.js"></script>" >index.html
$ npm start
```



svelte

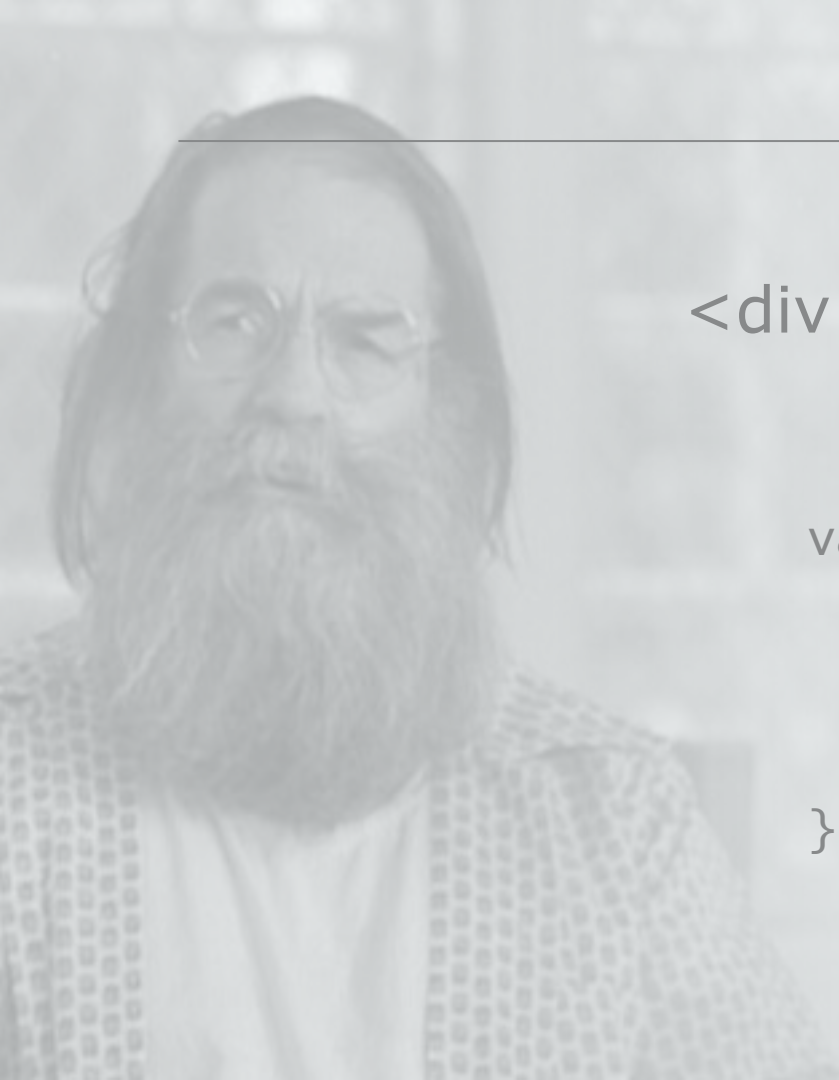
- build time framework
- incrementally adoptable
- self contained components
- one of the fastest



svelte

- create your component
- build it
- run it (no extra lib required)

components



```
<div>{{ message }}</div>
```

```
var app = new Msg({
  target: document.querySelector('app'),
  data: {
    message: 'Hello from Svelte!'
  }
});
```

install

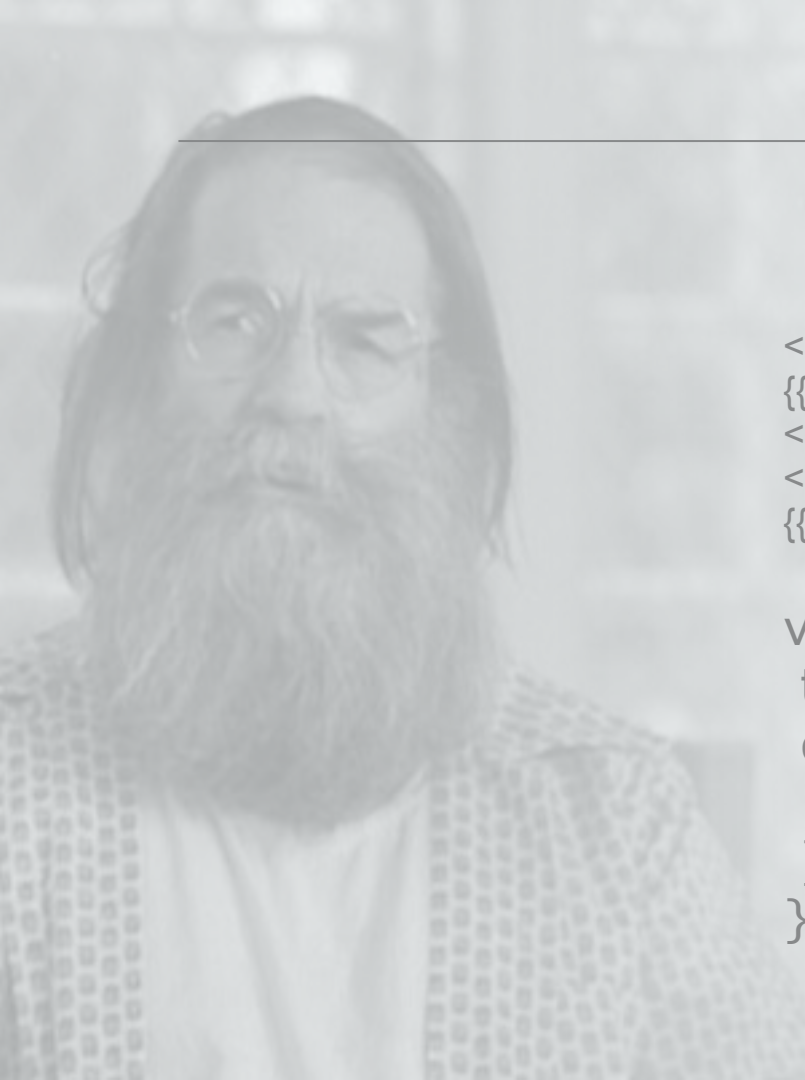
```
$ npm install -g svelte-cli

$ svelte compile --format iife Msg.html > Msg.js

$ http-server
```

demo

directives

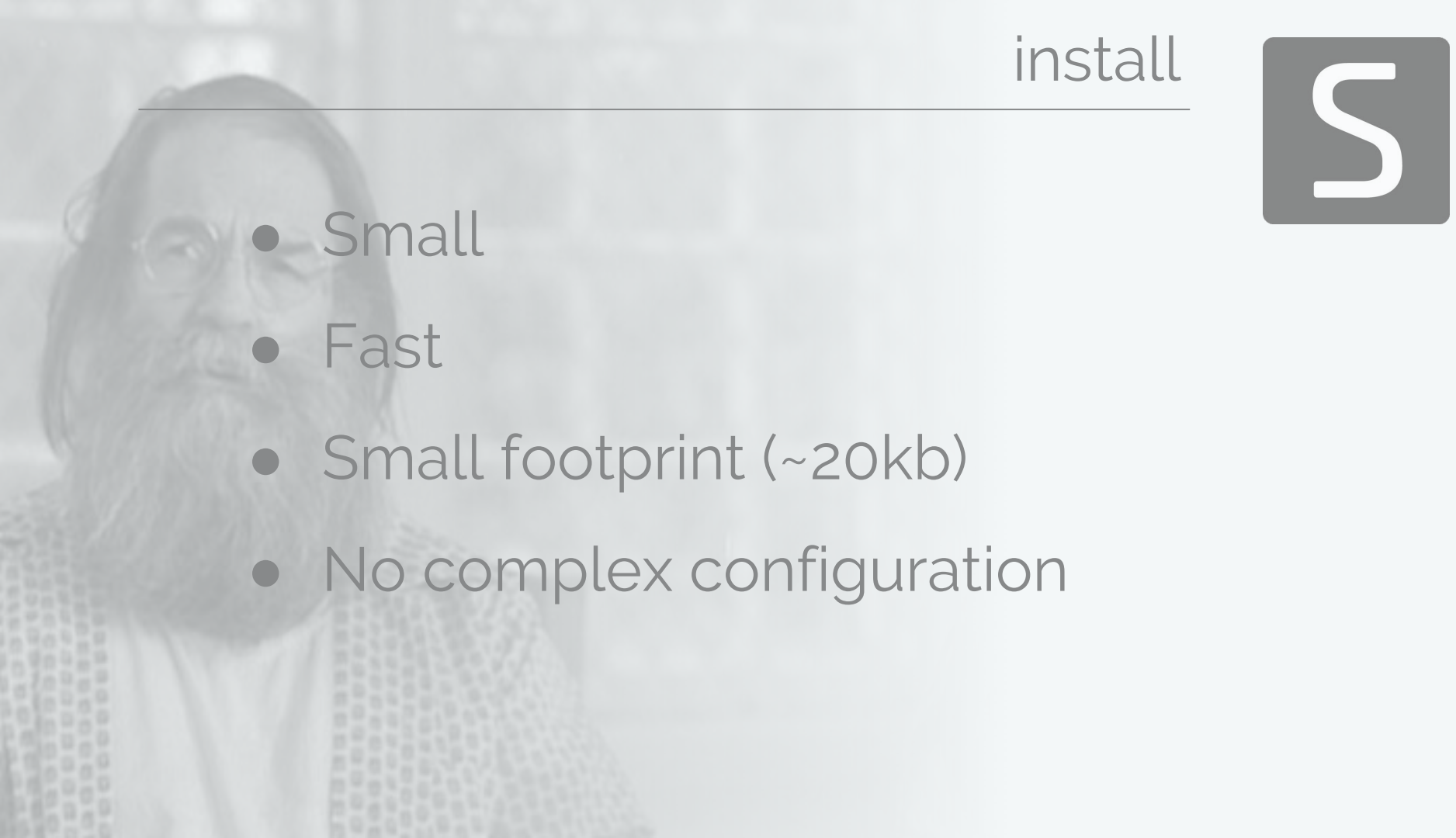


```
<input bind:value='count' />
{{#if count}}
<button on:click='set({ count: parseInt(count, 10) + 1 })'>+1</button>
<p>Count: {{count}}</p>
{{/if}}
```

```
var app = new Count({
 target: document.querySelector('app'),
 data: {
   count: 0
 }
})
```

demo

install



- Small
- Fast
- Small footprint (~20kb)
- No complex configuration

	angular v1.6.1-keyed	angular v2.4.3-keyed	bobril v5.0.4	domvm v2.0.1-keyed	elm v0.18.0	ember v2.10.0-beta.3	ember v2.6.1	kivi v1.0.0-rc2	knockout v3.4.1	mithril v0.2.5	mithril v1.0.0-alpha	nx v1.0.0-beta.1.1.0-keyed	plastiq v1.33.0	preact v7.1.0	ractive v0.8.9-keyed	react-lite v0.15.30	react v15.4.2-keyed	react v15.4.2-mobX-v3.0.1	react v15.4.2-redux-v3.6.0	vidom v0.7.1	vue v2.1.10-keyed	vanillajs-keyed
create rows Duration for creating 1000 rows after the page loaded.	254.72 ± 5.59 (1.96)	178.12 ± 3.25 (1.37)	142.90 ± 3.32 (1.10)	138.61 ± 1.78 (1.07)	164.66 ± 8.15 (1.27)	309.09 ± 11.50 (2.38)	348.78 ± 9.77 (2.69)	133.50 ± 2.30 (1.03)	338.68 ± 5.81 (2.61)	224.43 ± 2.15 (1.73)	153.54 ± 2.22 (1.18)	202.57 ± 4.92 (1.56)	176.55 ± 11.79 (1.36)	171.79 ± 5.04 (1.32)	320.99 ± 2.39 (2.47)	166.41 ± 7.13 (1.28)	184.29 ± 1.53 (1.42)	235.86 ± 8.55 (1.82)	208.07 ± 6.56 (1.60)	145.61 ± 3.95 (1.12)	151.97 ± 2.52 (1.17)	129.80 ± 2.95 (1.00)
replace all rows Duration for updating all 1000 rows of the table (with 5 warmup iterations).	253.66 ± 9.23 (1.80)	188.54 ± 6.65 (1.34)	157.97 ± 4.48 (1.12)	150.15 ± 1.22 (1.07)	168.52 ± 3.18 (1.20)	283.64 ± 4.39 (2.01)	303.32 ± 8.82 (2.15)	140.96 ± 0.90 (1.00)	359.00 ± 16.78 (2.55)	238.86 ± 1.25 (1.69)	163.23 ± 2.86 (1.16)	1006.57 ± 8.82 (7.14)	175.93 ± 6.98 (1.25)	190.51 ± 4.06 (1.35)	319.95 ± 5.50 (2.27)	219.74 ± 2.04 (1.56)	198.68 ± 6.05 (1.41)	224.46 ± 5.95 (1.59)	213.75 ± 1.68 (1.52)	154.56 ± 1.57 (1.10)	158.72 ± 2.70 (1.13)	140.94 ± 1.42 (1.00)
partial update Time to update the text of every 10th row (with 5 warmup iterations).	10.70 ± 0.30 (1.00)	9.86 ± 0.30 (1.00)	11.22 ± 0.39 (1.00)	15.07 ± 0.63 (1.00)	22.05 ± 1.42 (1.38)	14.91 ± 0.38 (1.00)	39.04 ± 1.22 (2.44)	9.33 ± 0.14 (1.00)	9.64 ± 0.22 (1.00)	51.39 ± 2.04 (3.21)	18.92 ± 0.48 (1.18)	11.04 ± 0.28 (1.00)	15.96 ± 1.74 (1.00)	12.72 ± 0.41 (1.00)	15.56 ± 0.60 (1.00)	27.62 ± 0.28 (1.73)	15.44 ± 0.75 (1.00)	14.45 ± 0.77 (1.00)	17.10 ± 1.07 (1.07)	11.29 ± 0.50 (1.00)	15.62 ± 0.25 (1.00)	9.62 ± 0.69 (1.00)
select row Duration to highlight a row in response to a click on the row. (with 5 warmup iterations).	6.12 ± 3.09 (1.00)	3.89 ± 1.81 (1.00)	4.72 ± 2.46 (1.00)	7.94 ± 0.76 (1.00)	15.07 ± 1.16 (1.00)	5.94 ± 0.65 (1.00)	9.39 ± 0.83 (1.00)	3.60 ± 1.51 (1.00)	9.53 ± 0.65 (1.00)	36.51 ± 1.42 (2.28)	11.39 ± 0.44 (1.00)	6.38 ± 0.11 (1.00)	5.06 ± 2.24 (1.00)	5.14 ± 2.16 (1.00)	7.03 ± 0.36 (1.00)	20.09 ± 0.54 (1.26)	6.86 ± 1.63 (1.00)	4.51 ± 0.70 (1.00)	7.58 ± 1.88 (1.00)	4.94 ± 1.69 (1.00)	8.37 ± 0.62 (1.00)	2.19 ± 0.83 (1.00)
swap rows Time to swap 2 rows on a 1K table. (with 5 warmup iterations).	12.15 ± 0.27 (1.00)	11.34 ± 0.35 (1.00)	10.32 ± 0.37 (1.00)	15.00 ± 0.68 (1.00)	25.80 ± 4.00 (1.61)	15.26 ± 0.68 (1.00)	37.63 ± 1.43 (2.35)	10.24 ± 0.56 (1.00)	12.99 ± 0.57 (1.00)	52.97 ± 0.99 (3.31)	18.86 ± 0.58 (1.18)	138.84 ± 0.85 (6.68)	11.04 ± 0.28 (1.00)	11.89 ± 0.21 (1.00)	18.28 ± 1.04 (1.14)	27.60 ± 0.37 (1.73)	13.42 ± 0.53 (1.00)	17.29 ± 0.38 (1.08)	14.00 ± 0.34 (1.00)	13.35 ± 0.60 (1.00)	16.81 ± 0.66 (1.05)	9.67 ± 0.23 (1.00)
remove row Duration to remove a row. (with 5 warmup iterations).	51.91 ± 2.06 (1.12)	48.68 ± 0.63 (1.05)	50.66 ± 1.83 (1.09)	54.28 ± 1.19 (1.17)	75.53 ± 1.21 (1.62)	53.36 ± 1.30 (1.15)	81.72 ± 1.18 (1.76)	49.76 ± 1.38 (1.07)	60.19 ± 6.90 (1.29)	88.99 ± 1.09 (1.91)	58.02 ± 1.95 (1.25)	150.21 ± 0.93 (3.23)	54.62 ± 4.05 (1.17)	49.30 ± 1.51 (1.06)	62.90 ± 1.17 (1.35)	67.50 ± 0.43 (1.45)	50.04 ± 1.01 (1.08)	61.03 ± 2.41 (1.31)	55.54 ± 1.88 (1.19)	52.00 ± 2.16 (1.12)	55.36 ± 2.16 (1.19)	46.51 ± 0.91 (1.00)
create many rows Duration to create 10,000 rows	2208.73 ± 35.23 (1.75)	1832.15 ± 11.42 (1.45)	1425.61 ± 24.13 (1.13)	1413.02 ± 14.02 (1.12)	1599.92 ± 26.46 (1.27)	2496.88 ± 24.94 (1.98)	2918.16 ± 94.58 (2.31)	1327.86 ± 16.38 (1.05)	3369.83 ± 104.62 (2.67)	2775.28 ± 40.40 (2.20)	1641.46 ± 19.80 (1.30)	2147.02 ± 13.45 (1.70)	1745.30 ± 24.88 (1.38)	1854.13 ± 47.47 (1.47)	2960.74 ± 47.00 (2.35)	2257.77 ± 46.29 (1.79)	1824.01 ± 11.51 (1.45)	2131.22 ± 34.17 (1.69)	1886.40 ± 23.79 (1.50)	1422.61 ± 16.49 (1.13)	1551.08 ± 17.83 (1.23)	1260.60 ± 13.52 (1.00)
append rows to large table Duration for adding 1000 rows on a table of 10,000 rows.	375.53 ± 7.18 (1.53)	288.12 ± 3.39 (1.17)	263.95 ± 3.66 (1.07)	348.49 ± 3.89 (1.42)	457.14 ± 16.91 (1.86)	457.75 ± 5.18 (1.86)	653.26 ± 14.54 (2.66)	245.79 ± 3.18 (1.00)	3782.61 ± 88.29 (15.39)	1312.40 ± 20.68 (5.34)	403.27 ± 4.74 (1.64)	344.85 ± 12.76 (1.40)	319.21 ± 6.55 (1.30)	316.13 ± 2.38 (1.29)	472.70 ± 11.09 (1.92)	1890.49 ± 19.53 (7.69)	327.64 ± 8.73 (1.33)	399.67 ± 6.93 (1.63)	352.52 ± 1.47 (1.43)	320.57 ± 5.83 (1.30)	369.41 ± 6.79 (1.50)	267.36 ± 1.39 (1.09)
clear rows Duration to clear the table filled with 10.000 rows.	772.27 ± 6.19 (3.71)	342.39 ± 8.86 (1.65)	216.50 ± 3.86 (1.04)	228.69 ± 3.03 (1.10)	221.22 ± 4.12 (1.06)	251.29 ± 2.22 (1.21)	549.75 ± 15.63 (2.64)	207.91 ± 4.74 (1.00)	585.13 ± 21.97 (2.81)	296.66 ± 2.76 (1.43)	237.72 ± 3.90 (1.14)	269.36 ± 6.72 (1.30)	244.32 ± 5.32 (1.18)	341.62 ± 5.36 (1.64)	663.90 ± 10.51 (3.19)	363.14 ± 2.36 (1.75)	410.94 ± 7.98 (1.98)	470.46 ± 8.38 (2.26)	415.37 ± 3.63 (2.00)	217.30 ± 4.58 (1.05)	248.82 ± 3.33 (1.20)	209.50 ± 3.24 (1.01)
slowdown geometric mean	**1.50**	**1.21**	**1.06**	**1.10**	**1.34**	**1.43**	**2.14**	**1.02**	**2.15**	**2.36**	**1.22**	**2.15**	**1.17**	**1.22**	**1.71**	**1.86**	**1.26**	**1.43**	**1.33**	**1.09**	**1.15**	**1.01**

recap

- now
- cool
- not cool
- simple
- fast

THE END

see you at svelte conf@2018