

# Exploring JavaScript Frameworks

The Point of



# Holly Schinsky



devgirlfl



devgirl.org




# Vue.js is...



The Progressive  
JavaScript Framework

**pro·gres·sive**

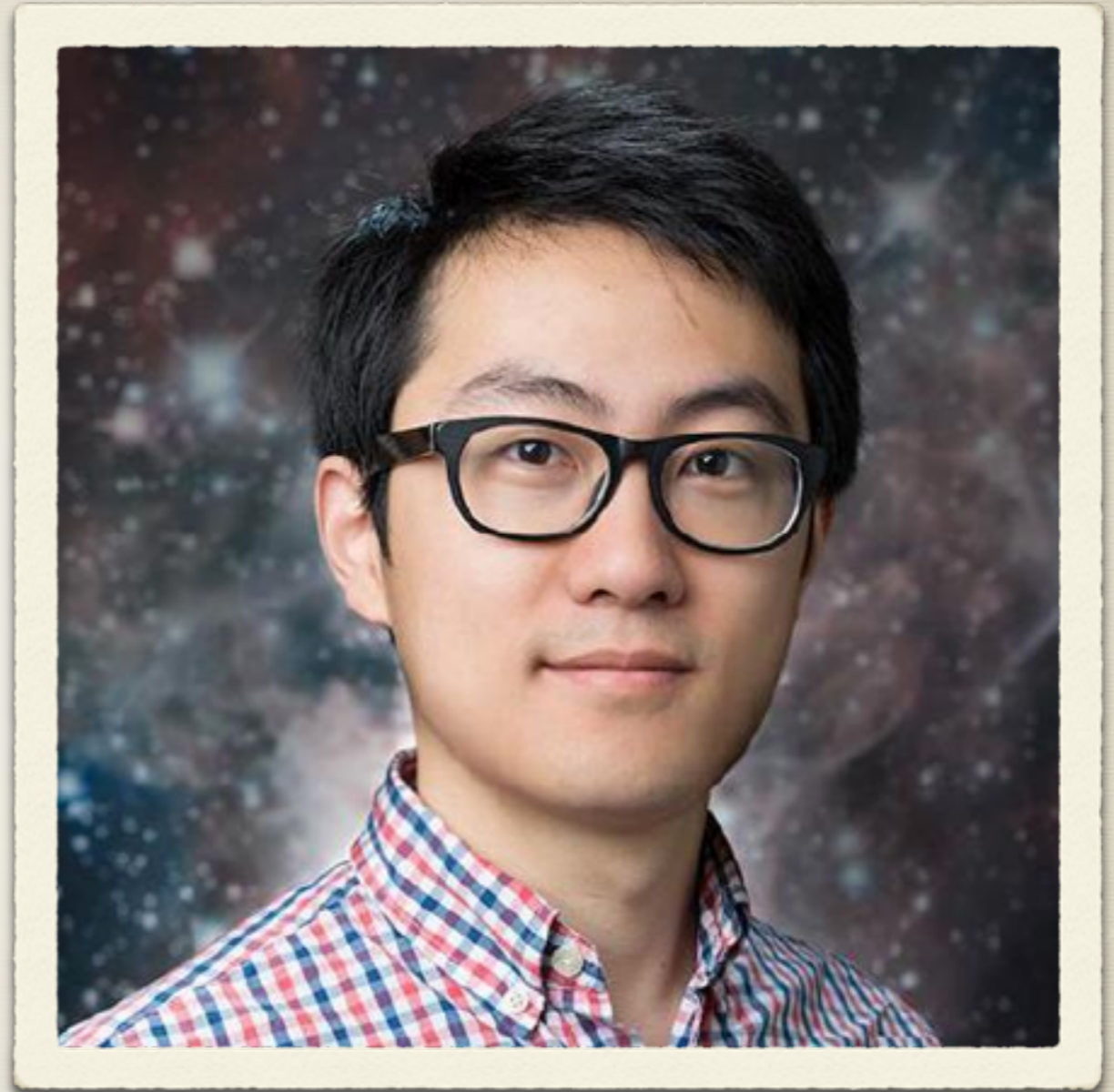
/prə'gresiv/ 

*adjective*

1. happening or developing gradually or in stages; proceeding step by step.  
"a progressive decline in popularity"  
*synonyms:* continuing, **continuous**, increasing, growing, developing, **ongoing**, accelerating, escalating; [More](#)

# Creator - Evan You

- \* Initial release in Feb 2014
- \* Previously worked at Google and Meteor
- \* Works on Vue full time
- \* Funded by the Patreon campaign



*“...what if I could just extract the part that I really liked about Angular and **build something really lightweight** without all the extra concepts involved?”*

# Why Vue?

- \* Approachable
- \* Scalable
- \* Productive
- \* ... makes developers happy :)



# It's Popular

angular / angular.js

Watch 4,455

Star 57,196

Fork 28,550

vuejs / vue

Watch 3,843

Star 68,933

Fork 9,982

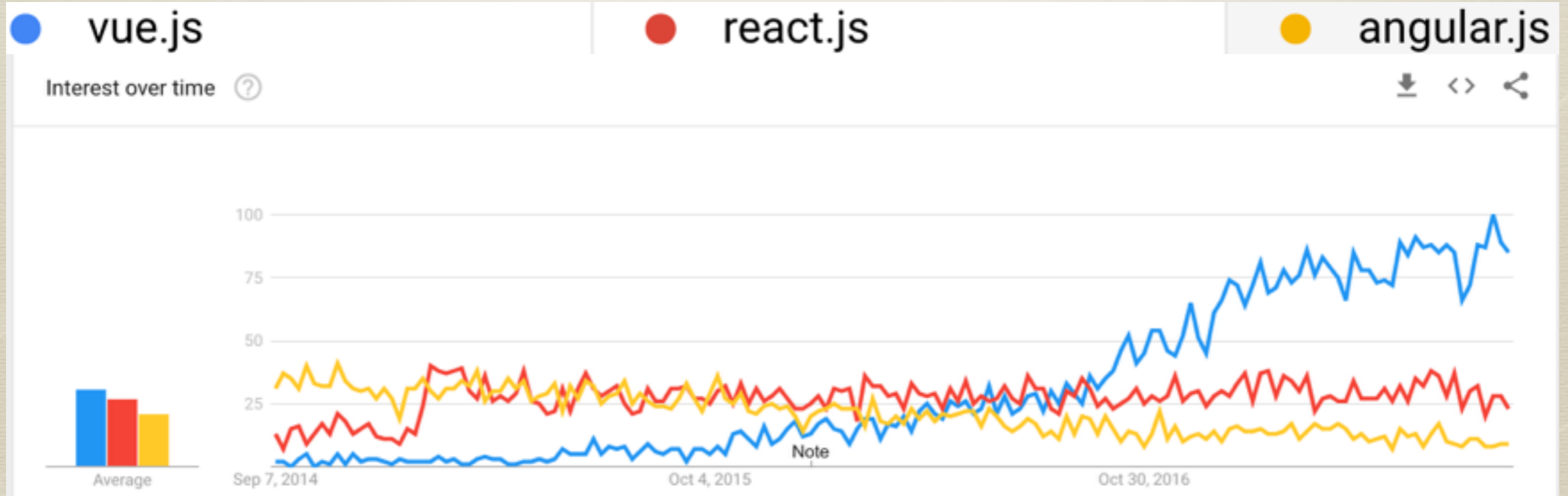
facebook / react

Watch 5,029

Star 77,251

Fork 14,695

# and growing...



# Vue Features

- \* Reactive Interfaces
- \* Declarative Rendering
- \* Data Binding
- \* Directives
- \* Template Logic
- \* Components
- \* Event Handling
- \* Computed Properties
- \* CSS Transitions and Animations
- \* Filters

See [vuejs.org](https://vuejs.org) for full details

# Vue Basics

- \* Apps are made up of nested and reusable components

```
<div id="app">  
  <app-nav></app-nav>  
  <app-view>  
    <app-sidebar></app-sidebar>  
    <app-content></app-content>  
  </app-view>  
</div>
```



# Simple Vue Example

```
<div id="demo">  
  <p>{{ message }}</p>  
  <input v-model="message">  
</div>
```

```
var demo = new Vue({  
  el: '#demo',  
  data: {  
    message: 'Vue is the best!'  
  }  
})
```

Vue is the best!!

Vue is the best!!

## index.html


# Quick Start

```
<html>
<head>
<title>My VueJs App</title>
</head>
<body>

  <!-- include Vue.js in our page -->
  <script src="https://unpkg.com/vue"></script>
  <!-- Container for Vue instance to manage -->
  <div id="app">
  </div>

  <script>
    // create a new Vue instance mounted to app
    var vm = new Vue({
      el: '#app'
    });
  </script>

</body>
</html>
```



# index.html

```
<html>
<body>
  //..
  <script src="https://unpkg.com/vue"></script>
  <!-- Container for Vue instance to manage -->
  <div id="app">
    <hello></hello>
  </div>
  <template id="hello-template">
    <div>
      <h1>Hello World!</h1>
    </div>
  </template>
  <script>
    // Register the hello component
    Vue.component('hello', {
      template: '#hello-template'
    });


    // create a new Vue instance and mount it to app div
    var vm = new Vue({
      el: '#app'
    });
  </script>
  //...
```

## Quick Start

# Component Registration

## Global

```
Vue.component('hello', {  
  template: '#hello-template',  
  // ...  
})  
new Vue({  
  el: '#app',  
  // ...  
})
```



Defined before root app instance

## Local

```
var Goodbye = {  
  template: '<div> Goodbye World!</div>'  
}  
  
Vue.component('hello', {  
  template: '#hello-template',  
  components: {  
    'Goodbye': Goodbye  
  }  
});
```

## Local by Module

```
import Hello from './components/Hello'  
  
export default {  
  name: 'app',  
  components: {  
    Hello  
  },  
  // ...  
}
```

## hello.vue

# Single file .vue components

```
<template>
  <div class="hello">
    <h1>{{ msg }}</h1>
  </div>
</template>
```

```
<script>
export default {
  name: 'hello',
  data () {
    return {
      msg: 'Welcome to Your Vue.js App'
    }
  }
}
</script>
```

```
<style scoped>
h1, h2 {
  font-weight: normal;
}
.hello {
  color: magenta;
}
</style>
```

# Templates

- \* Mustache-like syntax for basic text interpolation (like Angular)

```
<h1>Message: {{ msg }}</h1>
```

- \* Support JavaScript expressions

```
<h1>Msg: {{ msg.split('').reverse().join('') }}</h1>
```

- \* Use **v-bind** directive (or **:** prefix) within an HTML attribute

```
<div v-bind:id="dynamicId"></div>
```

```
<button :disabled="isButtonDisabled">Go</button>
```

# Vue Data

- When data items change, the view 'reacts' automatically to them
- Data properties are only reactive if they existed when the instance was created
- **v-model** directive allows two-way binding on form elements

```
<input type="text" v-model="message"/>
```

```
Vue.component('demo', {  
  template: '#demo-template',  
  data: function () {  
    return {  
      fname: 'Holly',  
      lname: 'Schinsky',  
      isDisabled: false,  
      items: ['Milk', 'Bread', 'Cereal', 'Peanut Butter'],  
      counter: 0  
    }  
  },  
})
```

# Props

—————→ `<hello v-bind:message="message"></hello>`  
parent component  
passes

```
Vue.component('hello', {  
  template: '#hello-template',  
  props: ['message'],  
  data: function () {  
    return {  
      fname: 'Holly',  
      lname: 'Schinsky',  
    }  
  }  
})
```

—————→ `<h1>{{ message }}</h1>`  
hello component  
uses



# Events

- \* Use `v-on` or `@` prefix (shorthand)

```
<button v-on:click="addMore(4, $event)">Add 4</button>
```

```
<button @click="addToCount">Add 1</button>
```

- \* Modifiers restrict handling to certain specific events

```
<a v-on:click.stop="doThis"></a>
```

```
<input v-on:keyup.enter="submit">
```

# Directives

```
<li v-for="item in items">
  {{ item }}
</li>
```

```
<input type="text" v-model="message"/>
```

```
<button v-on:click="counter+=1">Add 1</button>
```

```
<h1 v-if="isWinner">Winner winner chicken dinner!</h1>
<h1 v-else>Sorry about your luck</h1>
```

```
<textarea v-bind:disabled='isDisabled' v-bind:class='{ disabled:
isDisabled }'>{{ taVal }} </textarea>
```

```
v-text
v-html
v-show
v-if
v-else
v-else-if
v-for
v-on
v-bind
v-model
v-pre
v-cloak
v-once
```

# Filters

Use the pipe `|` symbol to add a filter function:

```
<!-- in text interpolation -->
{{ message | capitalize }}
```

```
<!-- in v-bind -->
<div v-bind:id="rawId | formatId"></div>
```

Define filter function in instance options:

```
new Vue({
  // ...
  filters: {
    capitalize: function (value) {
      if (!value) return ''
      return value.toString().toUpperCase();
    }
  }
})
```

# Computed Properties

```
Vue.component('demo', {  
  template: '#demo-template',  
  //..  
  computed: {  
    fullName: function () {  
      return this.fname + ' ' + this.lname;  
    }  
  }  
})
```

- Computed properties are cached based on their dependencies.
- Use for more complex logic to keep it out of your HTML or for things that don't require reactive properties

# Lifecycle Hooks

Run code at different stages of a Vue component lifecycle

```
new Vue({  
  data: {  
    a: 1  
  },  
  created: function () {  
    console.log('a is: ' + this.a)  
  }  
})  
// => "a is: 1"
```

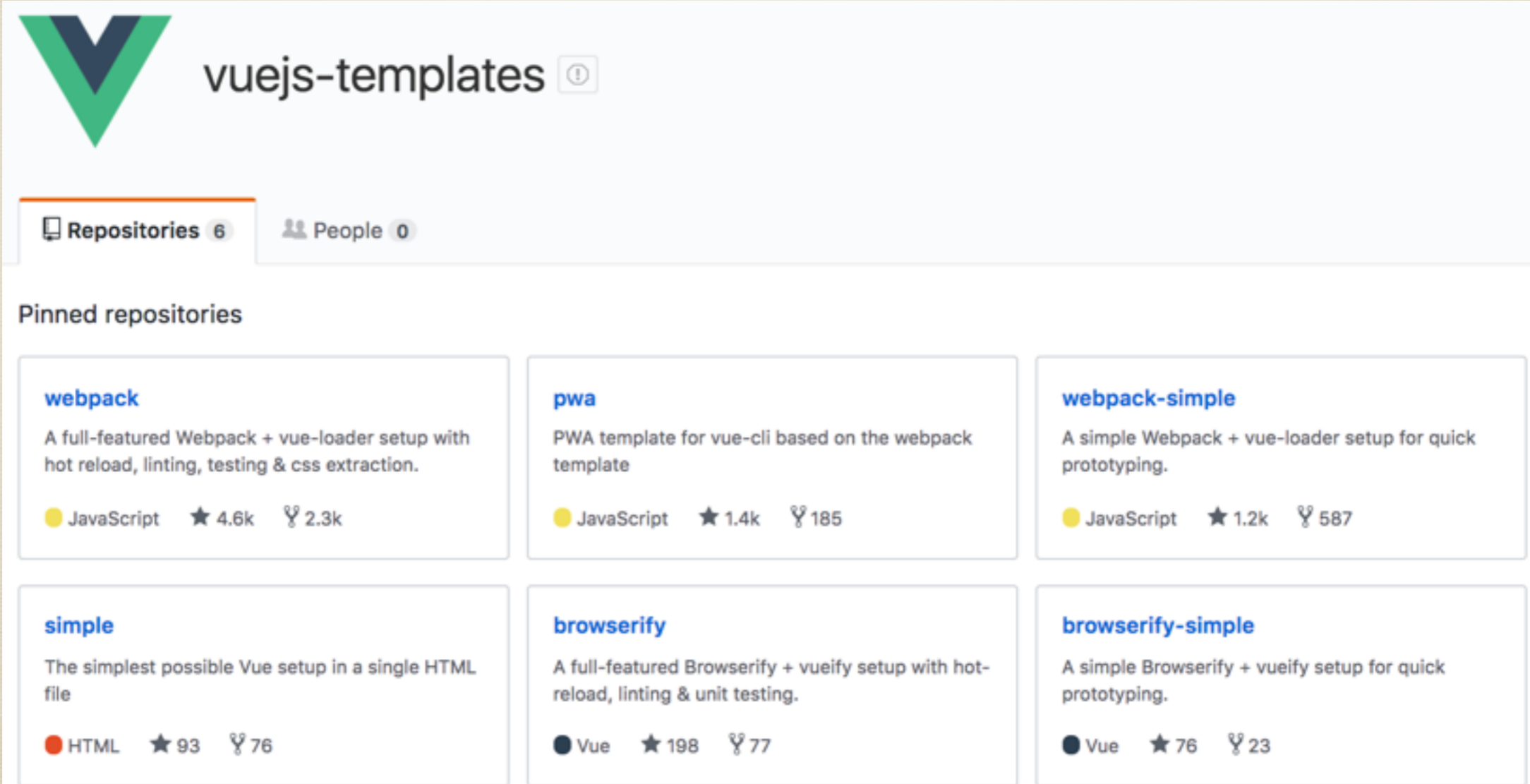
created(), mounted(), updated(), destroyed() etc

<https://vuejs.org/v2/guide/instance.html#Instance-Lifecycle-Hooks>

# Vue CLI

```
$ npm install -g vue-cli
```

```
$ vue init <template-name> <project-name>
```



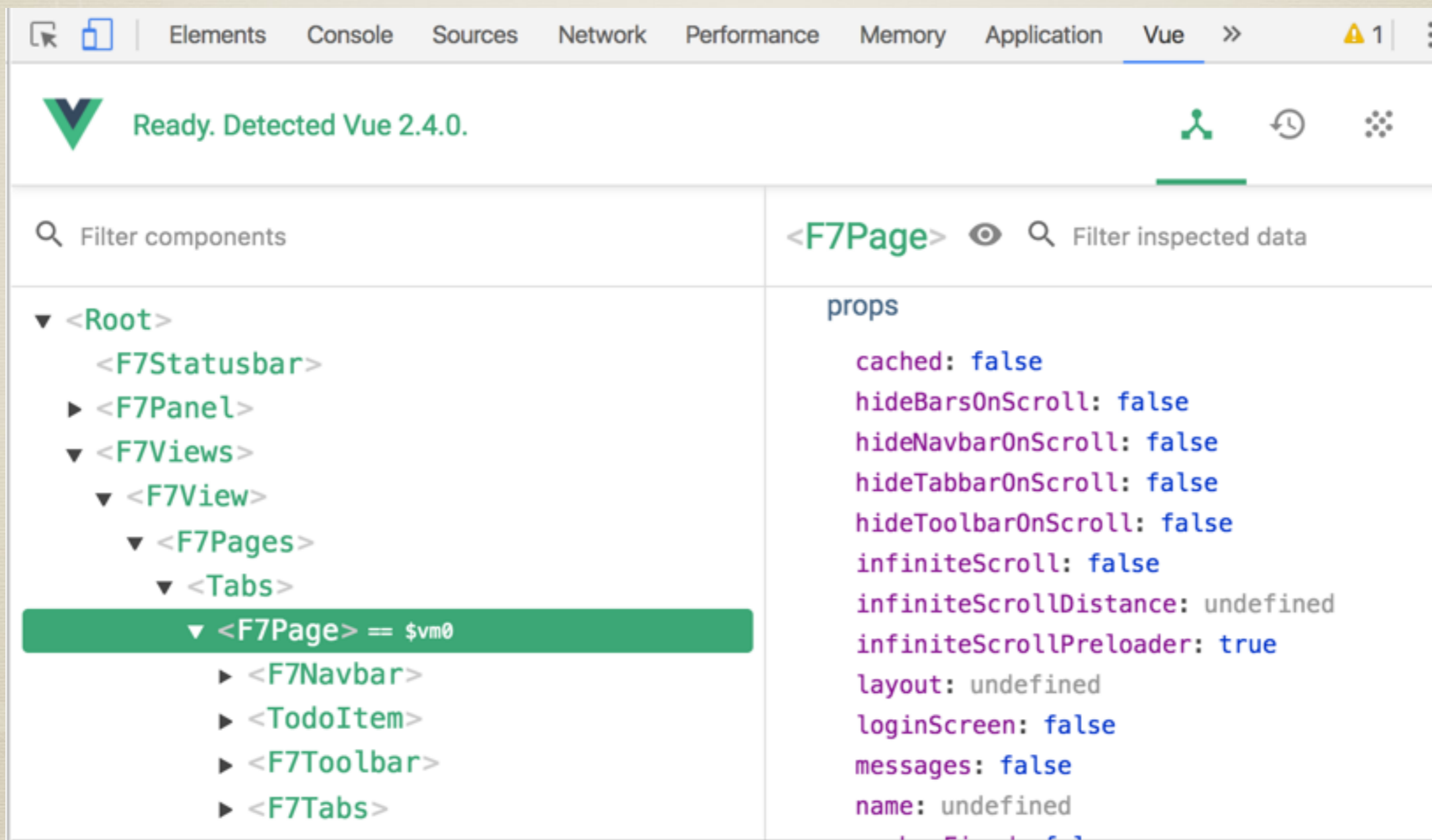
The screenshot shows the GitHub profile for 'vuejs-templates'. It features a green and blue 'V' logo and the repository name 'vuejs-templates'. Below the header, there are tabs for 'Repositories' (6) and 'People' (0). The 'Pinned repositories' section displays six repository cards:

- webpack**: A full-featured Webpack + vue-loader setup with hot reload, linting, testing & css extraction. (JavaScript, 4.6k stars, 2.3k forks)
- pwa**: PWA template for vue-cli based on the webpack template. (JavaScript, 1.4k stars, 185 forks)
- webpack-simple**: A simple Webpack + vue-loader setup for quick prototyping. (JavaScript, 1.2k stars, 587 forks)
- simple**: The simplest possible Vue setup in a single HTML file. (HTML, 93 stars, 76 forks)
- browserify**: A full-featured Browserify + vueify setup with hot-reload, linting & unit testing. (Vue, 198 stars, 77 forks)
- browserify-simple**: A simple Browserify + vueify setup for quick prototyping. (Vue, 76 stars, 23 forks)

```
$ vue init webpack my-vue-app
```

# Vue DevTools

<https://github.com/vuejs/vue-devtools>



The screenshot displays the Vue DevTools interface. At the top, a navigation bar includes tabs for Elements, Console, Sources, Network, Performance, Memory, Application, and Vue. The 'Vue' tab is active, showing a status bar with a green checkmark and the text 'Ready. Detected Vue 2.4.0.'. Below this, there are search filters: 'Filter components' on the left and '<F7Page>' with an eye icon and 'Filter inspected data' on the right. The main area is split into two panels. The left panel shows a component tree starting with '<Root>', which contains '<F7Statusbar>', '<F7Panel>', and '<F7Views>'. Under '<F7Views>', there is '<F7View>', which contains '<F7Pages>', which in turn contains '<Tabs>'. The '<F7Page>' component is highlighted with a green bar and labeled as '\$vm0'. The right panel shows the props for the selected '<F7Page>' component, listing various properties such as 'cached: false', 'hideBarsOnScroll: false', 'hideNavbarOnScroll: false', 'hideTabbarOnScroll: false', 'hideToolbarOnScroll: false', 'infiniteScroll: false', 'infiniteScrollDistance: undefined', 'infiniteScrollPreloader: true', 'layout: undefined', 'loginScreen: false', 'messages: false', and 'name: undefined'.

# Framework Similarities

- **Vue and Angular**
  - Similar templates
  - Use of directives syntax (ie: `{{ }}` and `v-if` to `ng-if` etc)
  - Some automatic data binding
- **Vue and React**
  - Uses a Virtual DOM
  - View layer focused
  - Reactive components



# Framework Differences

## - **Vue vs Angular**

- Not an opinionated stack, core focus is on the View layer
- Data binding in a one way data flow from parent to child
- Much smaller file size (20KB min+gzip)
- No dirty checking
- Works without additional tooling

## - **Vue vs React**

- Gives you a visually scannable template with encapsulated logic
- Templates vs JSX (less CPU)
- Faster ramp up vs less intuitive (JSX)
- Less boilerplate code needed in Vue
- Vue works out of the box (without a build system)

# Angular vs Vue

## Angular 1

```
var helloApp =
angular.module("helloApp", []);
  helloApp.controller("HelloCtrl",
function($scope) {
  $scope.name = "Calvin Hobbes";
});
```

## Angular 2

```
import { Component } from '@angular/core';

@Component ({
  selector: 'my-app',
  template: `<h1>Hello {{name}}</h1>`,
})
export class AppComponent { name = 'Angular'; }
```

## Vue

```
<div id="app">
  {{ message }}
</div>
var app = new Vue({
  el: '#app',
  data: {
    message: 'Hello Vue!'
  }
})
```

# React vs Vue

## React

```
var HelloMessage = React.createClass({
  render: function () {
    return <h1>Hello {this.props.message}!</h1>;
  }
});
```

```
ReactDOM.render(<HelloMessage message="World" />,
  document.getElementById('root'));
```

## Vue

```
<div id="app">
  {{ message }}
</div>
var app = new Vue({
  el: '#app',
  data: {
    message: 'Hello Vue!'
  }
});
```

# Vue and Mobile

- \* Small file size of Vue.js lends itself well to mobile
- \* Several UI libraries available with bindings
  - \* Framework7 - <https://framework7.io/vue/>
  - \* Onsen UI - <https://onsen.io/vue/>
- \* Get started quickly using PhoneGap starter project templates - <https://github.com/phonegap/phonegap-app-stockpile>

# Popular Add-ons

- \* **vuex** - state management - <https://vuex.vuejs.org>
- \* **vue-router** - routing between views in your SPA - <https://router.vuejs.org/en/>
- \* **axios** - HTTP Request library - <https://github.com/axios/axios>

# Resources

<https://github.com/vuejs/awesome-vue>

<https://css-tricks.com/intro-to-vue-1-rendering-directives-events/>

<https://alligator.io/vuejs/component-lifecycle/>

<https://madewithvuejs.com/>

<https://vuejs.org/2016/02/06/common-gotchas/>

<https://github.com/chrisvfritz/vue-2.0-simple-routing-example>