



Vue.js

Jakub Horváth  
HOR0282

# Vue.js

---

- Evan You
- Creative Technologist at Google
- Working full-time on Vue.js since 2016



# History




---

- Started in 2013
- First release in February 2014
- v1.0.0 in October 2015
- Latest release v2.6.6




# Comparison on GitHub

---




- **Vue.js**

 Watch	5,785	 Star	128,053	 Fork	18,241
---	-------	--	---------	--	--------

- **React**

 Watch	6,643	 Star	122,527	 Fork	22,212
---	-------	--	---------	--	--------

- **Angular**

 Watch	4,266	 Star	59,384	 Fork	28,933
---	-------	--	--------	--	--------

# Who use it?

---

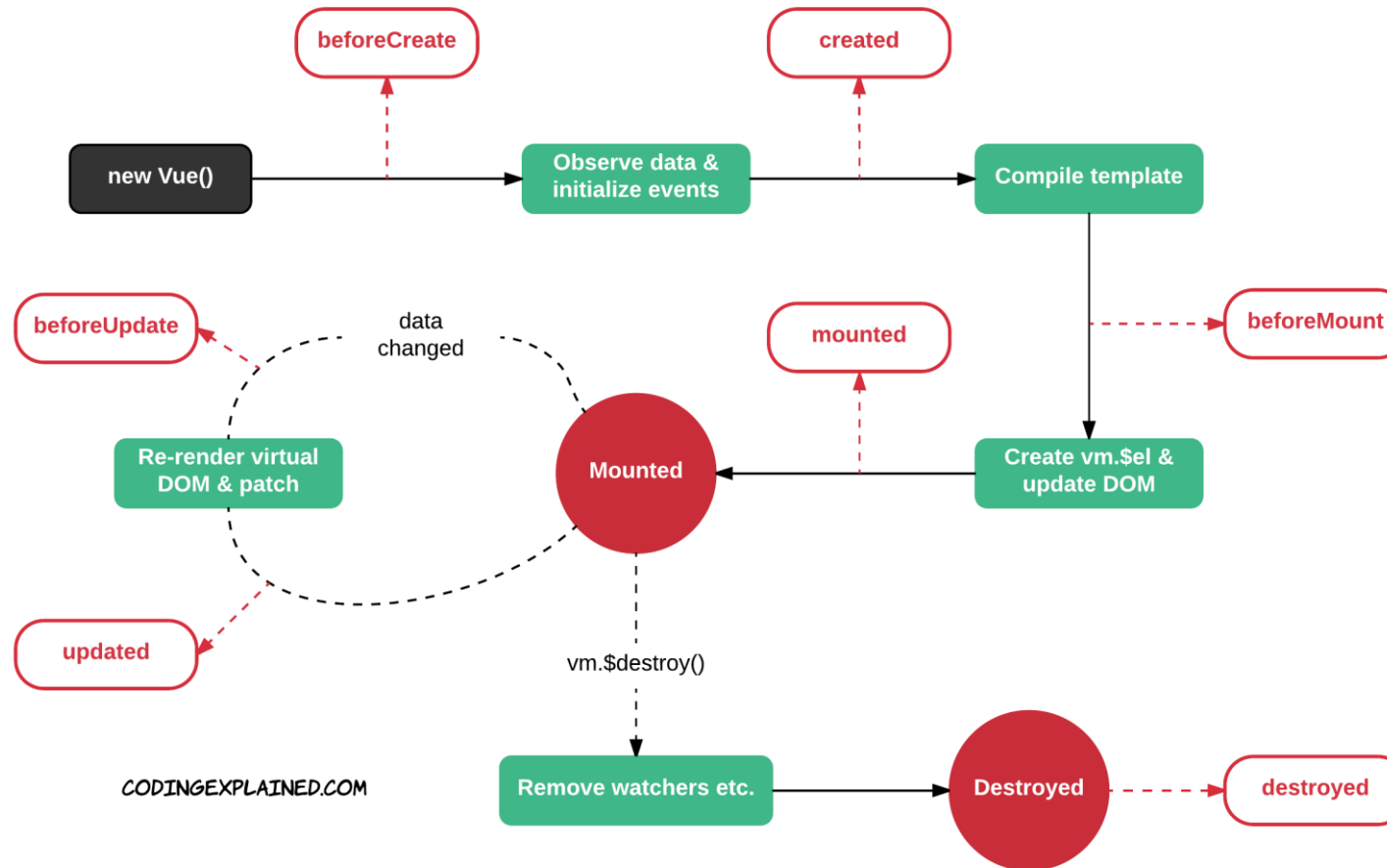
- Facebook
- Netflix
- Xiaomi
- WizzAir

# Information

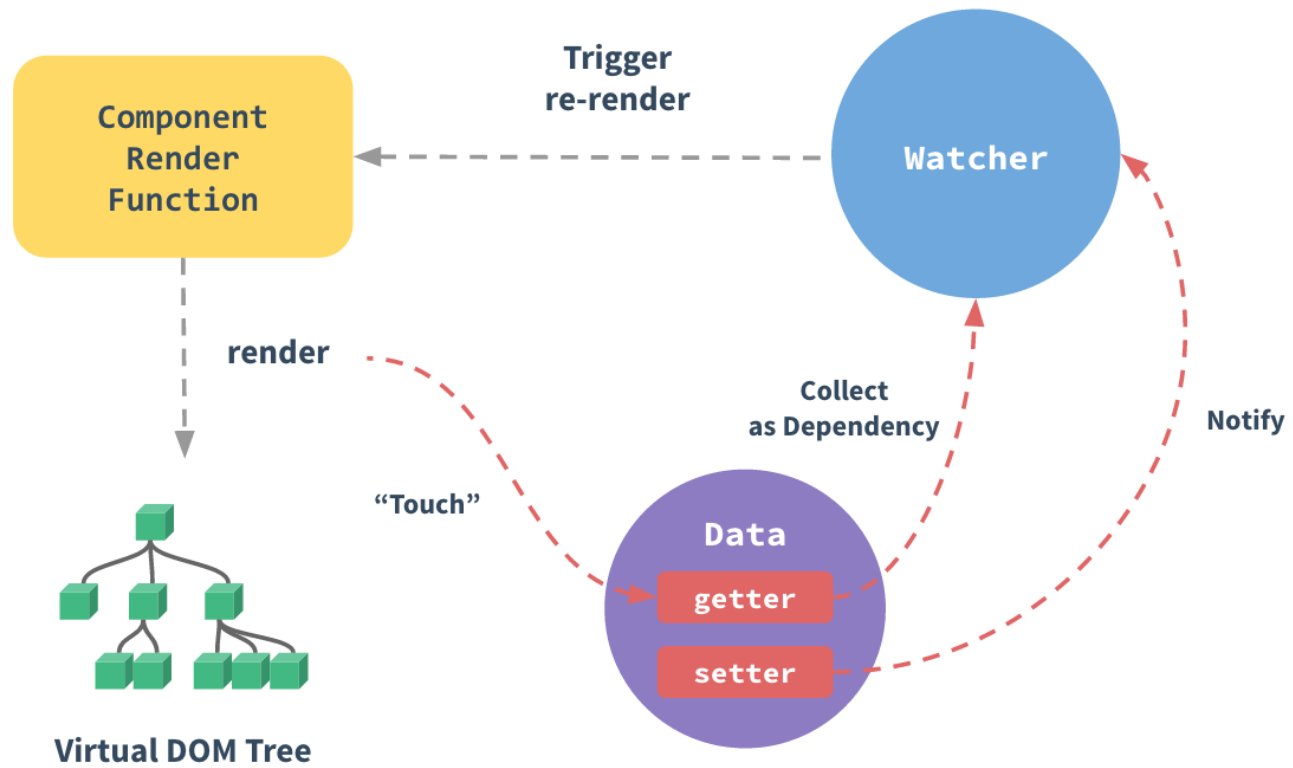
---

- Progressive framework
- Component oriented
- Mix the best of React and Angular

# Lifecycle



# Reactivity





# Declarative Rendering

---

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

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

# Directives

---

- **v-bind**

```
<!-- full syntax -->  
<a v-bind:href="url"></a>  
<!-- shorthand -->  
<a :href="url"></a>
```

- **v-on**

```
<!-- full syntax -->  
<a v-on:click="doMethod"></a>  
<!-- shorthand -->  
<a @click="doMethod"></a>
```

# Conditionals & Loops

---

- v-if

```
<div v-if="type === 'A'">A</div>  
<div v-else>B</div>
```

- v-for

```
<ul id="example">  
  <li v-for="i in items">{{ i }}</li>  
</ul>
```

# Conditional HTML Classes

---

```
<div :class="{  
  'active': isActive,  
  'text-danger': HasError  
}">  
</div>
```

```
//...  
data: {  
  isActive: true,  
  hasError: false  
}
```

# Method Event Handlers

---

```
<div id="example">  
  <button v-on:click="greet">Greet</button>  
</div>
```

```
var example2 = new Vue({  
  el: '#example',  
  data: {  
    name: 'Vue.js'  
  },  
  methods: {  
    greet: function (event) {  
      // `this` inside methods points to the Vue instance  
    }  
  }  
})
```

# Form Input Bindings

---

```
<input v-model="message" placeholder="edit me">
<p>Message is: {{ message }}</p>
```

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

# Components

---

```
<div id="app ">
  <ol>
    <todo-item
      v-for="item in groceryList"
      v-bind:todo="item"
      v-bind:key="item.id"
    ></todo-item>
  </ol>
</div>
```

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

```
var app = new Vue({
  el: '#app',
  data: {
    groceryList: [
      { id: 0, text: 'Vegetables' },
      { id: 1, text: 'Cheese' },
      { id: 2, text: 'Whatever' }
    ]
  }
})
```

# Global Registration

---

```
Vue.component('component-a', { /* ... */ })  
Vue.component('component-b', { /* ... */ })  
Vue.component('component-c', { /* ... */ })
```

```
new Vue({ el: '#app' })
```

```
<div id="app">  
  <component-a></component-a>  
  <component-b></component-b>  
  <component-c></component-c>  
</div>
```



# Local Registration

---

```
var ComponentA = { /* ... */ }  
var ComponentB = { /* ... */ }  
var ComponentC = { /* ... */ }
```

```
new Vue({  
  el: '#app',  
  components: {  
    'component-a': ComponentA,  
    'component-b': ComponentB  
  }  
})
```

# Properties

---

- Passing Static Props
- Passing Dynamic Props
- One-Way Data Flow
- Prop Validation

```
Vue.component('my-component', {  
  props: {  
    propA: Number,  
    // Multiple possible types  
    propB: [String, Number],  
    // Required string  
    propC: {  
      type: String,  
      required: true  
    },  
    // Number with a default value  
    propD: {  
      type: Number,  
      default: 100  
    }  
  }  
})
```

# Slot Element

---

```
<alert-box>
  Something bad happened.
</alert-box>
```

```
Vue.component('alert-box', {
  template: `
    <div class="demo">
      <slot>Default Error</slot>
    </div>
  `
})
```

# Vue Instances

---

```
var data = { a: 1 }  
var vm = new Vue({  
  el: '#example',  
  data: data  
})
```

```
vm.$data === data // => true
```

```
vm.$el === document.getElementById('example') // => true
```

# Instance Lifecycle hooks

---

- created
- mounted
- updated
- destroyed

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

# Properties

---

- Computed
- Watched
- Methods

# Event Modifiers

---

- **.stop**
- **.prevent**
- **.capture**
- **.self**
- **.once**
- **.passive**

```
<!-- the click event's propagation will be stopped -->  
<a v-on:click.stop="doThis"></a>
```

```
<!-- the submit event will no longer reload the page -->  
<form v-on:submit.prevent="onSubmit"></form>
```

# Key Modifiers

---

- `.enter`
- `.tab`
- `.delete`
- `.esc`
- `.up`
- `.down`
- `.left`
- `.right`

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



# Caveats

---

```
var vm = new Vue({
  data: {
    items: ['a', 'b', 'c']
  }
})
vm.items[1] = 'x' // is NOT reactive
vm.items.length = 2 // is NOT reactive
```

```
// Vue.set
Vue.set(vm.items, indexOfItem, newValue)
```

```
// Array.prototype.splice
vm.items.splice(indexOfItem, 1, newValue)
```

# Object Change Detection

---

```
var vm = new Vue({
  data: {
    a: 1
  }
})
// `vm.a` is now reactive

vm.b = 2
// `vm.b` is NOT reactive
```

# Thanks For Your Attention!

---

[HTTPS://VUEJS.ORG/V2/GUIDE/](https://vuejs.org/v2/guide/)

[HTTPS://ROBINCK.GITHUB.IO/VUE-PRESENTATION/](https://robinck.github.io/vue-presentation/)

[HTTPS://WWW.SLIDESHARE.NET/DAVIDLIEN1/VUEJS-INTRODUCTION](https://www.slideshare.net/DAVIDLIEN1/VUEJS-INTRODUCTION)