



# Web Development - Vue.js



# What's Vue.js?

A progressive, incrementally-adoptable **JavaScript framework** for building UI on the web.

It is the **top growing** popular javascript framework accepted by developers.





# What's Vue.js?

vuejs / vue

Watch 5,623 Unstar 116,943 Fork 16,524

<> Code Issues 203 Pull requests 141 Projects 0 Insights

👉 A progressive, incrementally-adoptable JavaScript framework for building UI on the web. <http://vuejs.org>

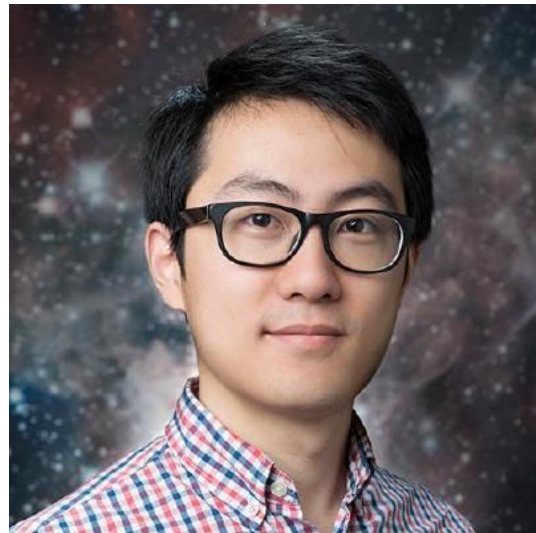
vue javascript frontend framework

2,621 commits 23 branches 228 releases 194 contributors MIT

framework accepted by developers.

# Meet the creator

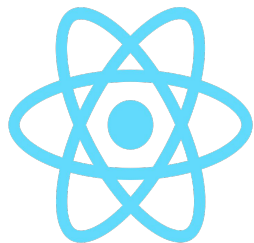
- Created by **Evan You**
- He started working on Vue.js after using AngularJS in a number of projects while working for Google
- "I figured, what if I could just extract the part that I really liked about Angular and build something really lightweight."
- Vue was originally released in **February 2014**.





# Just another Javascript framework?

It's one of the Big Three (React, Angular e Vue.js)





# Why choose Vue.js?

Best qualities of Vue.js:

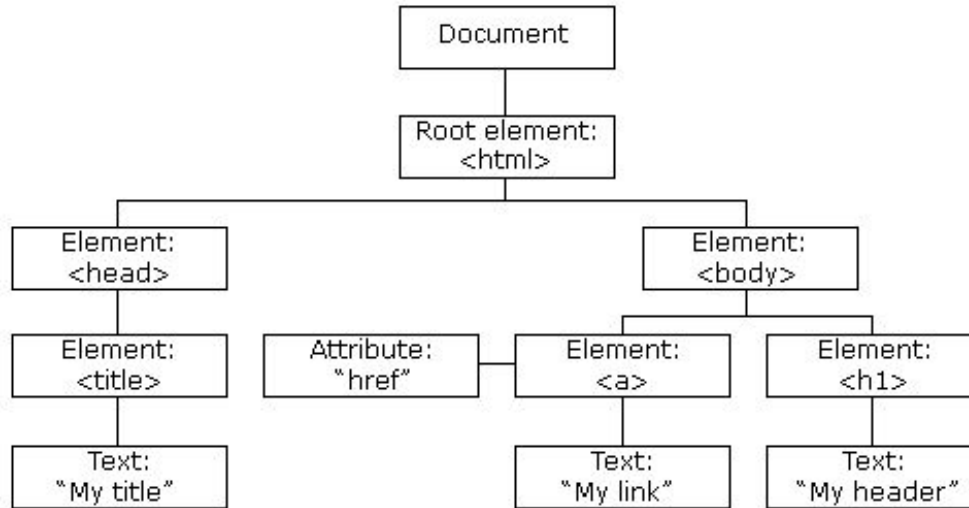
- Easy to learn
- Good Documentation
- Best performance of the top three frameworks

But current job market is less than that for React and Angular



# Important concept

## Document Object Model (DOM)





# Let's work!

Use the URL above if you want to follow:

- <https://codesandbox.io/s/vanilla>

Open the super useful Vue cheat sheet:

- <https://vuejs-tips.github.io/cheatsheet>





# The beginning:

Vue.js may be used as library and integrated with existing projects.

You can easily add it to a html file by using:

```
<script src="https://unpkg.com/vue"></script>
```

It can also be used for creating sophisticated Single-Page Applications when used in combination with modern tooling and supporting libraries. I'll show you how later!



# The Vue instance

## Template syntax:

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

## The Vue instance:

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

# Reactive programming:

$$a := b + c$$

Imperative programming

Reactive programming





# Let's get into directives (*v-stuff*)

## What are they for?

Reactively apply side effects to the DOM when the value of its expression change



# Let's get into directives (*v-stuff*)

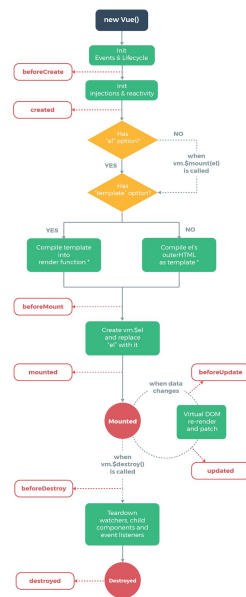
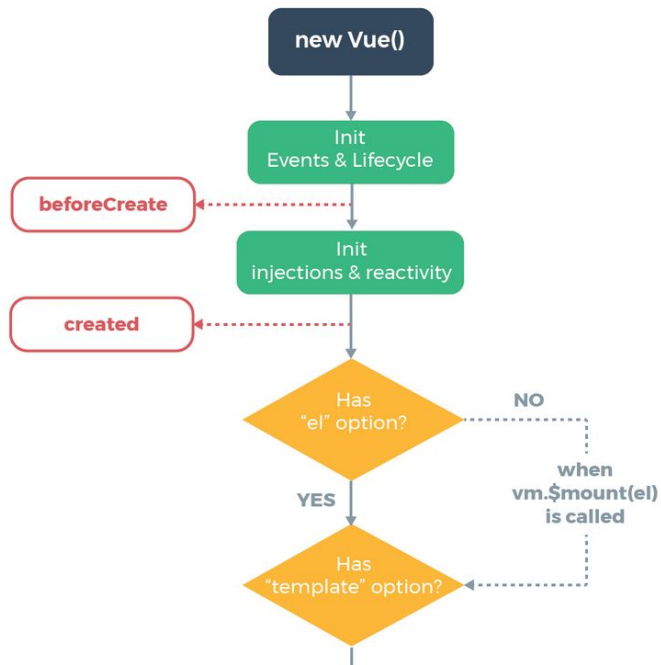
There's a bunch of them!

- 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

Let's try some!



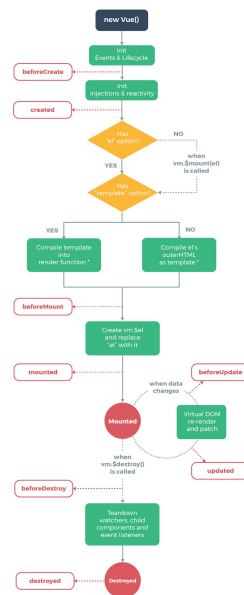
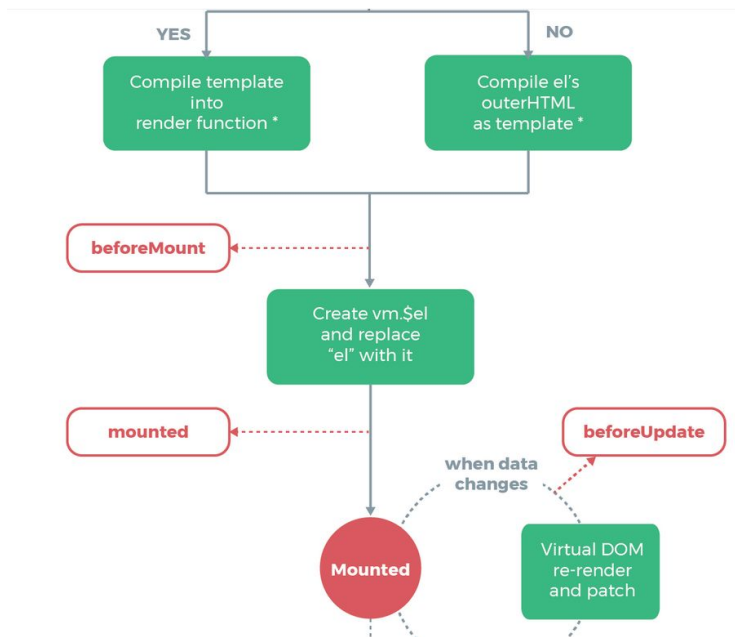
# Vue instance lifecycle:



\*template compilation is performed ahead of time if using a build step, e.g. single-file components



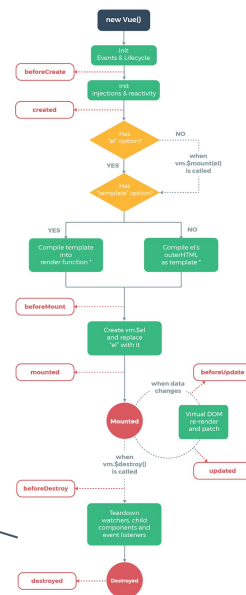
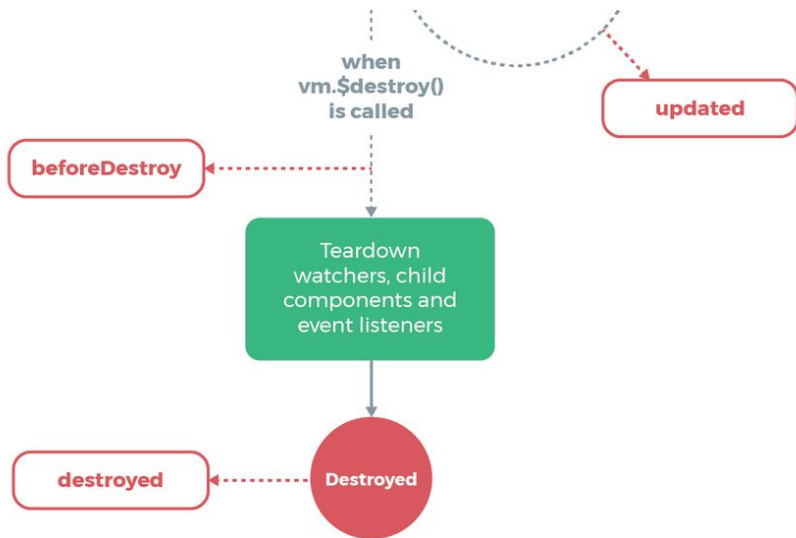
# Vue instance lifecycle:



\*template compilation is performed ahead of time if using a build step, e.g. single-file components



# Vue instance lifecycle:



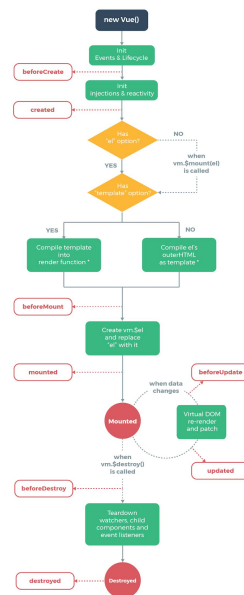
\*template compilation is performed ahead of time if using a build step, e.g. single-file components





# Vue instance lifecycle:

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>My first Vue app</title>
5   <script src="https://unpkg.com/vue"></script>
6 </head>
7
8 <body>
9   <div id="app">
10    <p v-if="condition">
11      {{message}}
12    </p>
13    <p v-else>
14      {{falsemessage}}
15    </p>
16    <button v-on:click="change">Oi, clica-me</button>
17  </div>
18
19  <script>
20    var vm = new Vue({
21      el: '#app',
22      data: {
23        message: 'Hello Vue!',
24        falsemessage: 'Bye Vue!',
25        condition: true
26      },
27      beforeCreate: function() {
28        alert("before creation")
29      },
30      methods: {
31        change(){
32          this.condition = !this.condition
33        }
34      }
35    })
36  </script>
37 </body>
38
39 </html>
```



\*template compilation is performed ahead of time if using a build step, e.g. single-file components

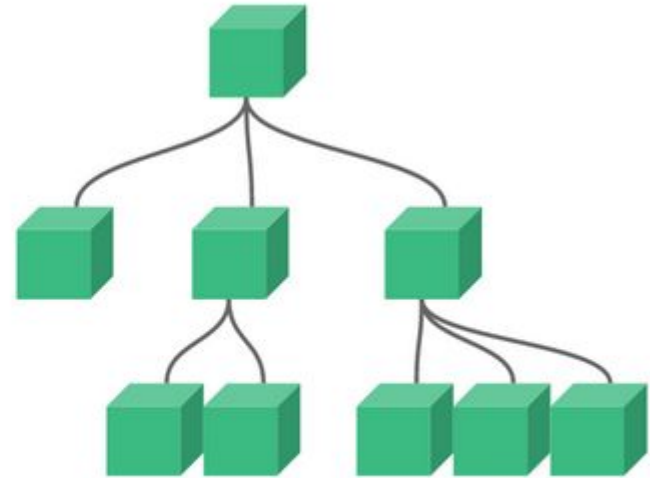


# Let's breath for a second

What did we learn so far?

- What Vue is for
- Reactivity
- How to include vue in your web page
- Directives (v-stuff)
- Methods
- Life Cycle

# Vue components:





# Very useful libraries:

- *Vuex*: Centralized store for all the components in an application
- *Vue Router*: Routing manager
- *Vue CLI*: Vue project manager
- *Vuetify or Vue Material*: Ui toolkits
- *Vue Loader*



# Vue CLI:

## Installation:

```
npm install -g @vue/cli
```

## Creating a project:

```
vue create name_of_project
```

or

```
vue ui
```



# Vue CLI:

### Create a new project

Details

Presets

Features

Configuration

Project folder

test1

/...nts/HackerSchool/Workshop NodeJS/test1

Package manager

npm

Additional options

Overwrite target folder if it exists

Git repository

Initialize git repository (recommended)

Cancel

Next

/mnt/c/Users/Renato Henriques/Documents/HackerSchool/Workshop NodeJS

Ready on http://localhost:8000

21/10/2018, 10:29:32



# Vue CLI:

### Create a new project

Details

**Presets**

Features

Configuration

**i** A preset is an association of plugins and configurations. After you've selected features, you can optionally save it as a preset so that you can reuse it for future projects, without having to reconfigure everything again.

Select a preset

☐ test  
vue-router, vuex

☐ Default preset  
babel, eslint

☒ **Manual**  
Manually select features

☐ Remote preset  
Fetch a preset from a git repository

← Previous

Next →

/mnt/c/Users/Renato Henriques/Documents/HackerSchool/Workshop NodeJS

Ready on http://localhost:8000

21/10/2018, 10:29:32



# Vue CLI:

Create a new project

Details

Presets

Features

Configuration

You will be able to add features after the project is created by installing plugins.

Enable features

Babel

Transpile modern JavaScript to older versions (for compatibility) [More Info](#)

TypeScript

Add support for the TypeScript language [More Info](#)

Progressive Web App (PWA) Support

Improve performances with features like Web manifest and Service workers [More Info](#)

Router

Structure the app with dynamic pages [More Info](#)

Vuex

Manage the app state with a centralized store [More Info](#)

CSS Pre-processors

Add support for CSS pre-processors like Sass, Less or Stylus [More Info](#)

Linters / Formatter

Check and enforce code quality with ESLint or Prettier [More Info](#)

Unit Testing

Add a Unit Testing solution like Jest or Mocha [More Info](#)

E2E Testing

Add an End-to-End testing solution to the app like Cypress or Nightwatch [More Info](#)

Use config files

Use specific configuration files (like '.babelrc') instead of using 'package.json'.

Previous

Create Project

/mnt/c/Users/Renato Henriques/Documents/HackerSchool/Workshop NodeJS

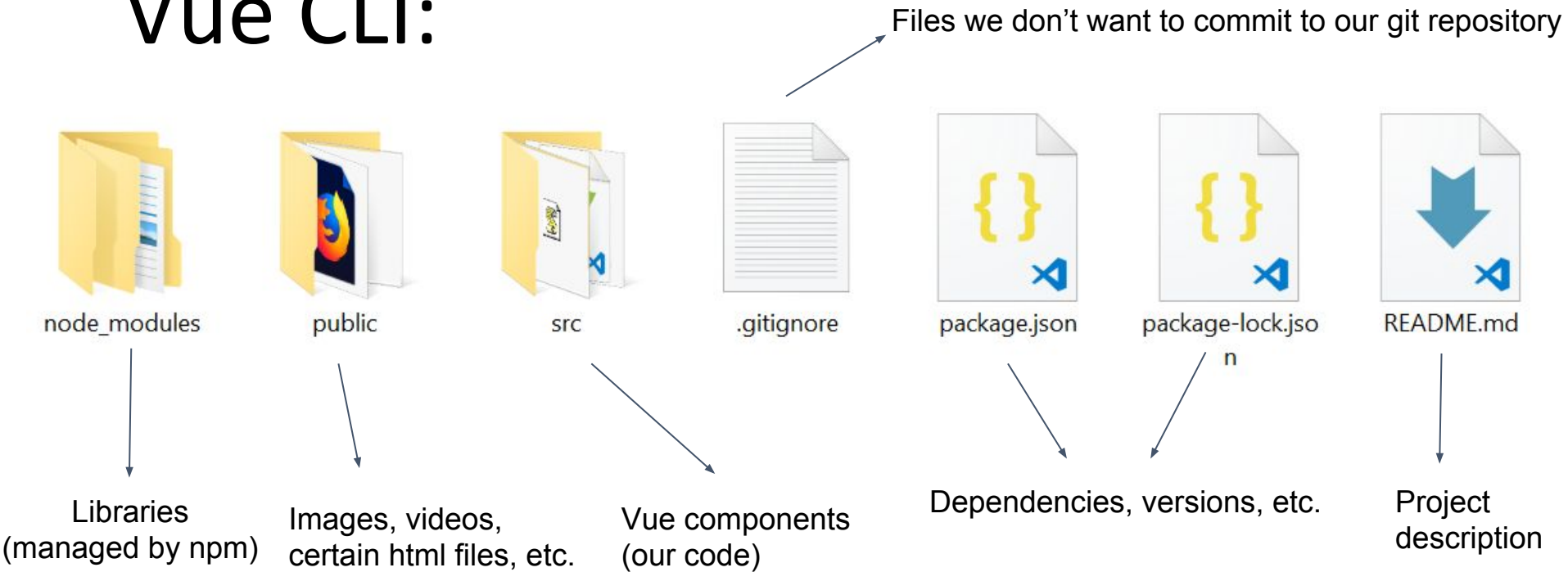
ready on: http://localhost:8080

10/10/2019, 15:16:33

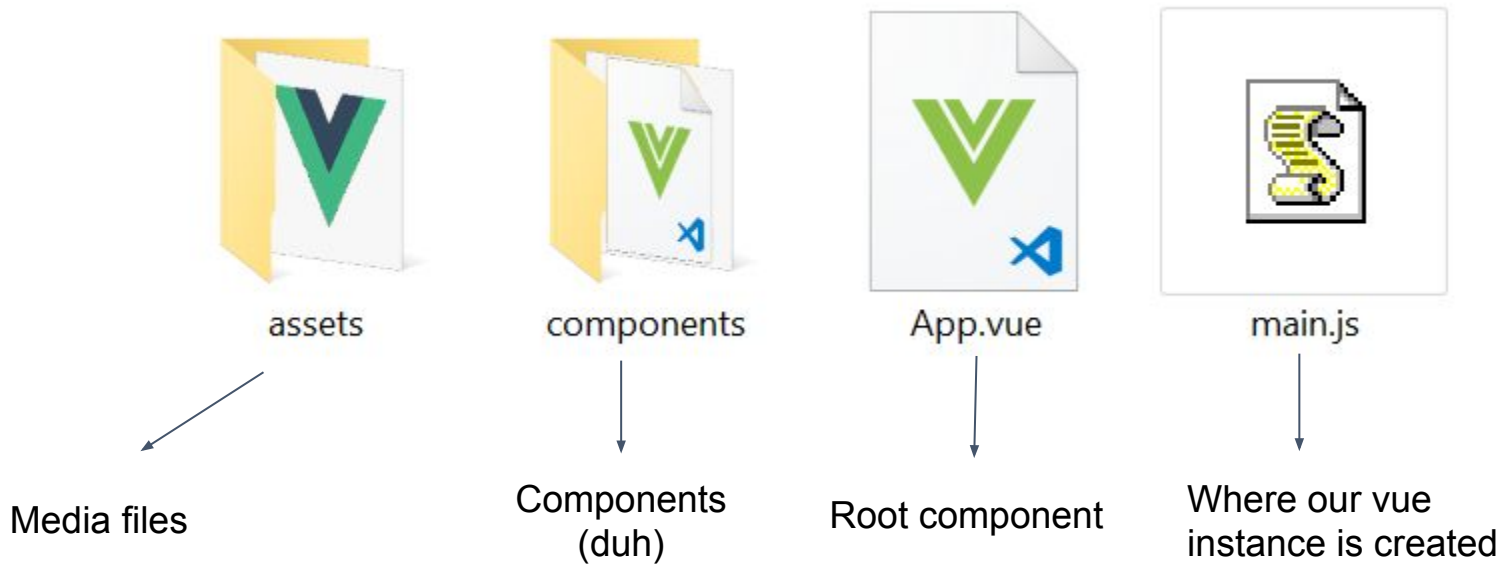




# Vue CLI:



# Vue CLI:





# Template syntax:

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

DOM (html)

```
<script>
export default {
  name: 'HelloWorld',
  data() {
    return {
      msg: "Hello"
    };
  }
}
</script>
```

Methods, data, etc. (Javascript)

```
<!-- Add "scoped" attribute to limit CSS to this component only -->
<style scoped>
  h3 {
    margin: 40px 0 0;
  }
  ul {
    list-style-type: none;
    padding: 0;
  }
  li {
    display: inline-block;
    margin: 0 10px;
  }
  a {
    color: #42b983;
  }
</style>
```

Styling the component (CSS)



# Template syntax:

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

```
<script>
export default {
  name: 'HelloWorld',
  data() {
    return {
      msg: 'Hello'
    };
  }
}
```

```
<!-- Add "scoped" attribute to limit CSS to this component only -->
<style scoped>
  h3 {
    margin: 40px 0 0;
  }
  ul {
    list-style-type: none;
    padding: 0;
  }
  a {
    color: #42b883;
  }
</style>
```

## SEPARATION OF CONCERNS, Ladies and Gentleman

DOM (html)

Methods, data, etc. (Javascript)

Styling the component (CSS)



# Let's try making a component:

- Create a new component called Person.vue
- The component has a *prop* called *person* which is an Object
- Present in paragraphs (<p>) the *person's* three properties (*name*, *age* and *single*).

Name: Renato Henriques

Age: 22

Single: true



# Acquiring data in App component:

- Until now, we only inserted data into the DOM
- The directive **v-model** let us directly change our variables using input elements of HTML

```
<input v-model="information.name" placeholder="name">
```

```
<input v-model="information.age" placeholder="age">
```

```
<input type="checkbox" id="checkbox" v-model="information.single">  
<label for="checkbox">Single</label>
```



## Passing the data to a child component:

- Import *Person* component from the `/components`
- Declare in *App* component that it has a child component called *Person*
- Insert the *Person* component in the DOM
- `<Person></Person>`



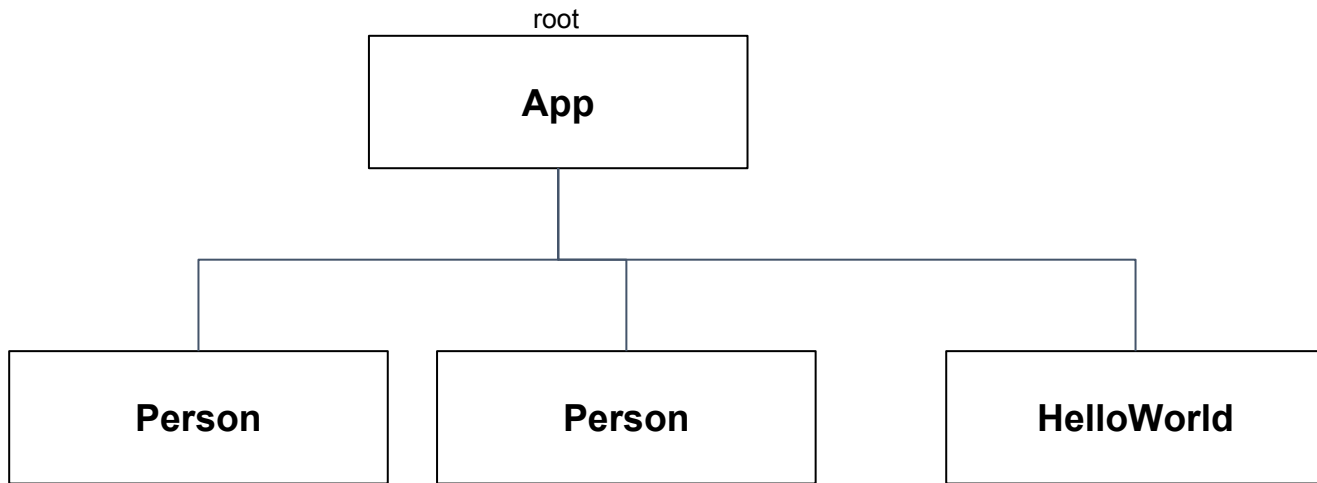
## Passing the data to a child component:

- Use the directive **v-bind** to bind the prop from *Person* to the *information* object from *App*
- `<Person v-bind:person="information"></Person>`





# Final state of our components tree:





# When we need it beautiful and quickly

- Vuetify!



# Cool things we haven't talked today:

- How to connect to Backend (Axios library)
- Vue router
- Components in depth
- Vuex
- ...



# But this is only the beginning...

Vue.js documentation:

<https://cli.vuejs.org/guide>

The Vue.js cheat sheet:

<https://vuejs-tips.github.io/cheatsheet/>

Compilation of guides, libraries, “everything” about Vue.js:

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

A very cool tutorial about making a Full Stack Web App using Vue.js and Node.js:

<https://www.youtube.com/watch?v=Fa4cRMaTDUI>



*Hope you enjoyed!*  
**Do you have any questions?**



**Presentation by:** Renato Afonso Henriques

**Contact:** [renato.henriques@hackerschool.tecnico.ulisboa.pt](mailto:renato.henriques@hackerschool.tecnico.ulisboa.pt)

**GitHub:** <https://github.com/rena2henriques>