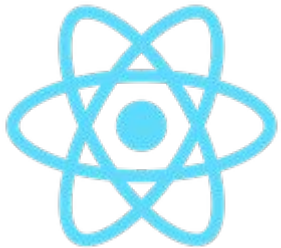
Desenvolvimento de Software para WEB

Aula 8 - Introdução ao Vue.js

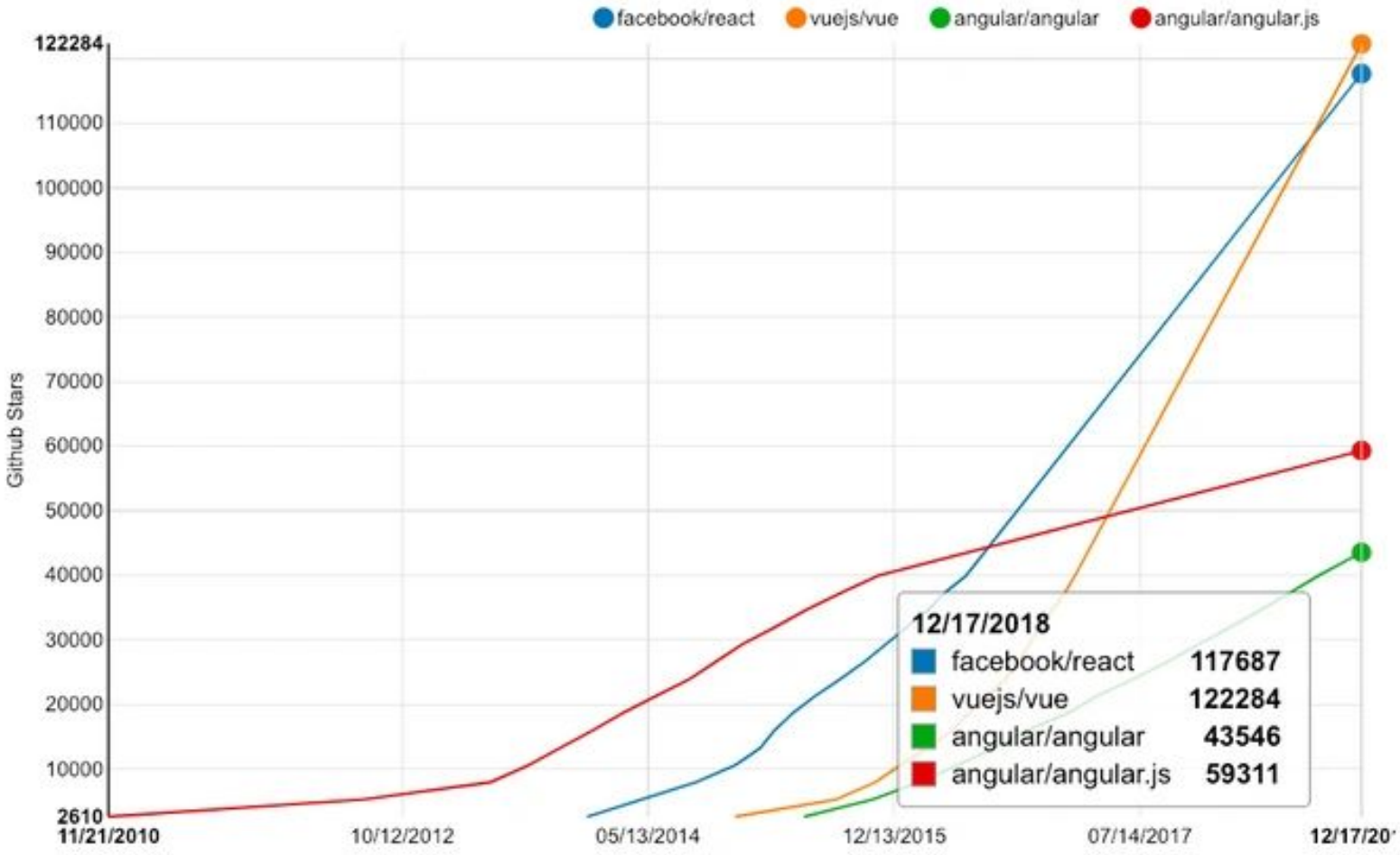
Professor: Anderson Almada

Frameworks front-end



Frameworks front-end

	Angular	React	Vue
Initial release	2010	2013	2014
Official site	angular.io	reactjs.org	vuejs.org
Approx. size (KB)	500	100	**80**
Current version	7	16.6.3	2.17
Used by	Google, Wix	Facebook, Uber	Alibaba, GitLab


facebook/react
vuejs/vue
angular/angular
angular/angular.js
Github Stars
122284
110000
100000
90000
80000
70000
60000
50000
40000
30000
20000
10000
2610
11/21/2010
10/12/2012
05/13/2014
12/13/2015
07/14/2017
12/17/20
12/17/2018
facebook/react 117687
vuejs/vue 122284
angular/angular 43546
angular/angular.js 59311

Introdução

- Instalação (no final do body)
  - `<script src="https://cdn.jsdelivr.net/npm/vue" ></script>`

    ou

  - `<script src="vue.js"></script>`

Introdução

- Hello World

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

<script>
  var app = new Vue({
    el: "#app",
    data: {
       title: "Aula WEB"
    }
  });
</script>
```

Reatividade

- Mudança do valor dos elementos em tempo de execução (reação)

- Teste:
  - Alterar o valor de app.title no console.

Diretivas

- Regras de realização de tarefas

- Na prática, entram como atributos nas tags HTML
  - Começam com **v-**

v-if

HTML

```
<div id="app">
  <p v-if="status">{{ title }}</p>
</div>
```

JS

```
data: {
  status: false,
  title: "Aula WEB"
}
```

O **v-if** funciona como o if em linguagens de programação: dado uma condição, mostre algum elemento HTML

v-show

HTML

```
<div id="app">
  <p v-show="status">{{ title }}</p>
</div>
```

JS

```
data: {
  status: false,
  title: "Aula WEB"
}
```

O **v-show** funciona parecido com o v-if. A diferença é que no v-show se caso a condição não seja satisfeita, então o elemento html será coloca em **display: none**. No v-if, se a condição não for satisfeita, o elemento HTML ficará como comentário

v-else

HTML

```
<div id="app">
  <p v-if="status">{{ title }}</p>
  <p v-else>{{ title }}</p>
</div>
```

JS

```
data: {
  status: false,
  title: "Aula WEB"
}
```

O **v-else** é uma diretiva para que caso a condição do v-if não seja satisfeita, então ela deve mostrar algum outro elemento HTML para o usuário.

Exemplo v-if e v-else

HTML

```
<div id="app">
  <div v-if="usuario.id == 1">
    Id: {{ usuario.id }} <br>
    Nome: {{ usuario.nome }} <br>
    Sobrenome: {{ usuario.sobrenome
}}<br>
  </div>
  <div v-else>
    <p>Usuário não existe !!</p>
  </div>
</div>
```

JS

```
data: {
  usuario: {
    id : 1,
    nome: "Chico",
    sobrenome: "Silva"
  }
}
```

v-for

HTML

```
<ul>
  <li v-for="carro in carros">{{ carro.nome }}</li>
</ul>
```

JS

```
data: {
  carros: [
    { nome: "gol"},
    { nome: "palio"},
    { nome: "siena"}
  ]
}
```

O **v-for** serve para realizar uma iteração em um array, por exemplo

v-model

HTML

```
<form action="">
  Nome: <input type="text" v-model="nome"
name="nome"> <br>
  Sobrenome: <input type="text"
v-model="sobrenome" name="sobrenome">
<br>
  Idade: <input type="text"
v-model="idade" name="idade"> <br>
  Aceita: <input type="checkbox"
v-model="aceita" value="sim"> <br>
</form>
```

HTML

```
{{ nome }} <br>
{{ sobrenome }} <br>
{{ idade }} <br>
{{ aceita }} <br>
```

O **v-model** serve para linkar um determinado input com uma chave do objeto JavaScript no data Vue.js. Assim, se o valor no campo mudar, a chave também tem o valor alterado.

v-model

JS

```
data: {
  nome: "",
  sobrenome: "",
  idade: "",
  aceita: ""
}
```

v-bind

HTML

```
<img v-bind:src="nomeImg" v-bind:alt="alt">
```

Ou

```
<img :src="nomeImg" :alt="alt">
```

JS

```
data: {
  nomeImg: "carro.jpeg",
  alt: "Carro vermelho"
}
```

O **v-bind** serve para linkar um determinado atributo do elemento HTML com alguma chave do objeto JavaScript no data Vue.js. Assim, se alterar o valor de uma determinada chave, muda o conteúdo do atributo linkado.

v-bind em class

CSS

```
<style>
    .red1 {
        color: red
    }

    .green2 {
        color: green
    }
</style>
```

HTML

```
<p v-bind:class="classeCor1">OK</p>
<p v-bind:class="classeCor2">OK</p>
```

JS

```
data: {
    classeCor1: "red1",
    classeCor2: "green2"
}
```

v-bind em style

HTML

```
<p :style="ok">OK</p>
```

JS

```
ok: {
    'font-size': '20px',
    'text-transform': 'lowercase'
}
```

methods

JS

```
var app = new Vue({
    el: "#app",
    data: {
        title: "Aula WEB"
    },
    methods: {
        enviar() {
            alert("Enviado!");
        },
    }
});
```

Os **methods** funcionam como funções. Nos methods, pode ser alterado alguma chave do objeto JavaScript no data. Para isso, utilize o **this**

Exemplo: **this.title**

v-on

HTML

```
<button type="button" v-on:click="enviar()">OK</button>
```

O **v-on** serve linkar algum evento com determinado elemento. Nesse exemplo, temos que sempre quando o botão for clicado, o **method** chamado **enviar**() será chamado.

filters

JS

```
var app = new Vue({
    el: "#app",
    data: {
        usuarios: [
            { nome: "Jose" },
            { nome: "Chico" },
            { nome: "Francisco" },
        ]
    },
```

JS

```
    filters: {
        setMaiusculo(str) {
            return str.toUpperCase();
        }
    }
});
```

Os **filters** também tem o objetivo de manipulação, mas dado uma determinada entrada para ele. Usado geralmente em v-for

filters

HTML

```
<ul>
    <li v-for="usuario in usuarios" >{{ usuario.nome | setMaiusculo }} </li>
</ul>
```

filters - parâmetros

JS

```
filters: {
     setMaiusculo(str) {
          return str.toUpperCase();
     },
     truncar(str, length) {
          if(str.length > length) {
               str = str.substring( 0, length) + " ...";
          }
          return str;
     }
}
```

filters - parâmetros

HTML

```
<ul>
<li v-for="usuario in usuarios">{{ usuario.nome | setMaiusculo | truncar(3) }}
</li>
</ul>
```

Links importantes

- https://br.vuejs.org/

- https://www.youtube.com/watch?v=07-TvnH7XNo&list=PLcoYAcR89n-qq1vGRbaUiV6Q9puy0qigW

Dúvidas??

E-mail: almada@crateus.ufc.br