

Vue.JS

Instância & Sintaxe de Templates





Revisão

Instalação

Desenvolvimento

```
<script src="https://cdn.jsdelivr.net/npm/vue/dist/vue.js"></script>
```

Produção

```
<script src="https://cdn.jsdelivr.net/npm/vue"></script>
```

Iniciando no Vue.js

HTML

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

JavaScript

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

Olá Vue!

Iniciando no Vue.js

HTML

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

JavaScript

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

Olá Vue!

A instância Vue

A Instância Vue

```
var vm = new Vue({  
  // opções  
});  
  
var vm = new Vue({  
  el: '#app',  
  data: {  
    message: 'Olá Vue!'  
  }  
});
```

Como convenção, muitas vezes usamos a variável **vm** (abreviação de ViewModel) para se referir à instância **Vue**.

Uma aplicação **Vue** consiste em uma instância Vue raiz criada com **new Vue**.

Quando você cria uma instância Vue, é necessário passar um objeto de opções.

Dados e Métodos

```
// Objeto de dados
var data = { a: 1 }

// O objeto é adicionado à
instância Vue
var vm = new Vue({
  data: data
})

// A instância cria uma referência
ao mesmo objeto
vm.a === data.a // => true
```

Quando uma instância Vue é criada, ela adiciona todas as propriedades encontradas no objeto **data** ao sistema de reatividade do **Vue**.

Quando algum dado for modificado, a camada visual irá **re-renderizar**.

Dados e Métodos

```
// Não irá causar mudanças
vm.b = 'propriedade b'

// Exemplo válido
data: {
  newTodoText: '',
  visitCount: 0,
  hideCompletedTodos: false,
  todos: [],
  error: null
}
```

Importante:

Se você sabe que precisará de uma propriedade no futuro, você precisará especificar algum valor inicial no objeto `data`.

Dados e Métodos

```
// Prefixo $
var data = { el: 1 }
var vm = new Vue({
  el: '#app',
  data: data
})

vm.$data === data // => true
// $el === '#app'
vm.$el === v.el // => false
```

Importante:

Os métodos das instâncias Vue são diferenciados pelo prefixo **\$** para não confundi-los com propriedades definidas pelo usuário.

Instance Lifecycle

```
new Vue({  
  data: {  
    a: 1  
  },  
  created: function () {  
    // `this` aponta para a  
    // instância  
    console.log('a é: ' + this.a)  
  }  
})
```

Ao longo da inicialização da instância Vue, serão invocados **gatilhos de ciclo de vida** (lifecycle hooks), oferecendo a oportunidade de executar lógicas personalizadas em etapas específicas.

O gatilho **created** pode ser utilizado para executar código logo após a instância ser criada.

created **mounted**

updated **destroyed**

Syntaxe de templates

Interpolações

#Texto

O mais básico data binding, interpolando texto com chaves duplas

```
var app = new Vue({
  el: '#app',
  data: {
    nome: 'João'
  }
})

// Reativo
<span>Olá, {{ nome }}</span>

// Interpolação única
<span v-once>

Valor fixo: {{ nome }}

</span>
```

Interpolações

#HTML

Para que você exiba HTML, utilize a diretiva

v-html

```
var app = new Vue({
  el: '#app',
  data: {
    rawHtml: '<span
style="color:red">Texto
vermelho</span></p>'
  }
})

<p>Interpolação textual: {{
rawHtml }}</p>

<p>Diretiva v-html: <span
v-html="rawHtml"></span></p>
```

Interpolações

#Atributos

Chaves duplas não podem ser usadas em atributos HTML. Para isso, utilize a diretiva

v-bind

```
var app = new Vue({
  el: '#app',
  data: {
    dynamicId: 'secao',
    buttonDisabled: true
  }
})

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

<button
  v-bind:disabled="buttonDisabled">B
otão</button>
```

Interpolações

#Expressões

Chaves duplas não podem ser usadas em atributos HTML. Para isso, utilize a diretiva

v-bind

```
{{ number + 1 }}
```

```
{{ ok ? 'SIM' : 'NÃO' }}
```

```
<div v-bind:id="'list-' +  
id"></div>
```

```
// Apenas uma expressão!!
```

```
<!-- atribuição -->
```

```
{{ var a = 1 }}
```

```
<!-- controle de fluxo -->
```

```
{{ if (ok) { return message } }}
```


Diretivas

Diretivas são atributos especiais com o prefixo **v-**. O trabalho de uma diretiva é aplicar reativamente efeitos colaterais ao DOM, ou seja, realizar algum efeito quando o valor da expressão é modificado.

v-if

v-on

v-model

v-for

v-bind

```
<p v-if="showElem">Visível</p>

// Algumas diretivas usam atributos
<a v-bind:href="url"> ... </a>
<a v-on:click="doSomething"> ...
</a>

// Atributos dinâmicos
<a v-bind:[attributeName]="url">
... </a>
```

Abreviações

v-bind

```
<!-- sintaxe completa -->  
<a v-bind:href="url"> ... </a>
```

```
<!-- abreviação -->  
<a :href="url"> ... </a>
```

v-on

```
<!-- sintaxe completa -->  
<a v-on:click="doSomething"> ... </a>
```

```
<!-- abreviação -->  
<a @click="doSomething"> ... </a>
```

The background features a dark blue-grey field with several large, overlapping, semi-transparent geometric shapes in a slightly lighter shade of blue. These shapes are angular and layered, creating a sense of depth. Horizontal bands of a muted green color are positioned at the top and bottom of the frame.

Obrigado!