**Índice da Seção 2:**

* **Primeiros Passos**
* **Ferramentas de Desenvolvimento Vue**
* **Trabalhando com Dados**
* **Múltiplas Instâncias Vue**
* **Acessando os dados da instância**
* **Métodos**
* **Diretivas**
* **Vinculação de dados bidirecional**
* **Atributos Vinculativos**

Este é um exemplo de inicialização do Vue.js. Onde no data contém a informação que será colocada dentro do elemento especificado em .mount().

Vue.createApp({

data() {

return {

firstName: 'John'

}

}

}).mount('#app')

Dentro do data, deve conter informações do tipo expressão. Por exemplo:

// Number Expression

2 + 2; //Evalutes to 4

//String Expression

“hello”.toUpperCase(); // Evalutes to HELLO WORLD

// Logical Expressions

100 === 100; // Evaluates to true

Não deve conter o seguinte tipo de informação:

//Declarations

var a;

// Function Declarations

function greet(message) {

console.log(message);

}

// Conditional Statements

if (a === b) {

// Do something

}

Para checkar quais informações a instância do Vue está considerando, basta abrir o vue tools, clicar em <Root> e analisar no canto direito as informações que estão presentes em data.

Para acessar os dados da instância e alterá-los, você pode fazer o seguinte:

const vm = Vue.createApp({

data() {

return {

firstName: 'John',

lastName: 'Doe'

}

}

}).mount('#appOne')

setTimeout(() => {

vm.firstName = 'Bob';

}, 2000);

Perceba que colocamos a instância dentro de uma variável e a partir dessa variável, conseguimos acessar a propriedade contida dentro do data desejada. Essa maneira de acesso é chamada de Proxy. Para acessar sem um Proxy, pode ser feito o seguinte:

vm.$data.firstName

Escrevemos o nome e o sobrenome utilizando duas linhas, uma para first e outra para last. É possível agrupar tudo em uma única linha da seguinte forma:

{{ `${firstName} ${lastName.toUpperCase()}` }}

O que ocorre é que essa é considerada uma expressão complexa.

Diretivas sempre começam com v-nomedadiretiva. As diretivas são atributos especiais que você adiciona a elementos HTML, permitindo que o Vue.js manipule o DOM de maneira declarativa.

v-cloak => só mostra o elemento após a página ter sido carregada por completo.

v-model => conecta o valor de um data a um campo de formulário, por exemplo.

Reatividade é quando qualquer mudança no data é imediatamente refletida na página.

**Diretivas**

Diretivas podem ser escritas assim: v-nomedadiretiva ou somente :nomedadiretiva

Diretivas do tipo v-on: podem ser escritas com o @ no lugar.

**V-MODEL**

A diretiva v-model é classificada como two-way binding, ela é bidirecional. De qualquer um dos dois lados que alterar o valor, ele será atualizado no outro instantaneamente. Essa diretiva recebe uma variável, e por isso tem que declarar ela dentro do data no js.

A diretiva model possui dois tipos: v-model.number, v-model.lazy.trim

**V-BIND**

**Em termos gerais, nós poderemos usar o v-bind em qualquer atributo html. Você também pode usar em atributos do HTML5 como por exemplo data-alguma-coisa Exemplos: data-user-id, data-validate, etc. Uma dica é não usar a palavra v-bind, porque tem um caminho mais curto, que é só colocar o sinal de dois pontos :.**

**V-ON:KEYUP OU @KEYUP.ENTER**

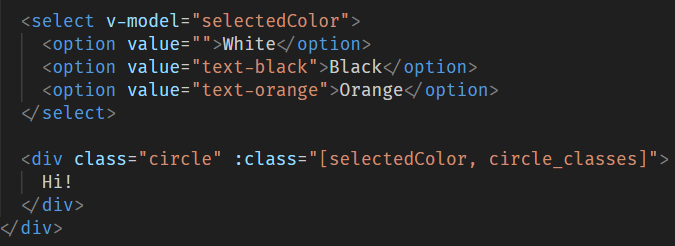
**É uma diretiva Vue.js usada para adicionar um ouvinte de evento a um botão no teclado.**

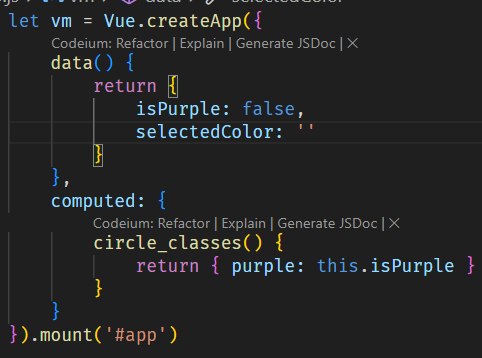
**IDENTIFICANDO O TIPO DE DADO DE UMA VARIÁVEL DO DATA:**

Lá no HTML, quando for chamar no html a variável que está dentro do data do js, usando como exemplo: {{ age }}, para saber o tipo de dado, basta: {{ typeof age }}

**ALTERANDO A COR DE UM ELEMENTO HTML COM VUE.JS**

Podemos adicionar um v-model, que trabalha em todos os elementos de formulários, incluindo selects e textareas.





**Use v-show se o elemento irá constantemente aparecer e desaparecer da página.**

**Use v-if se o elemento irá aparecer e desaparecer ocasionalmente, ou raramente.**

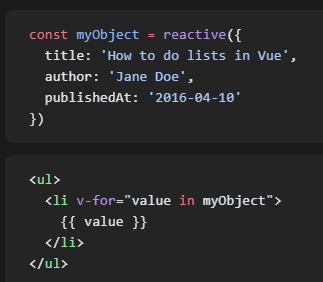
**CONDITIONAL RENDERING**

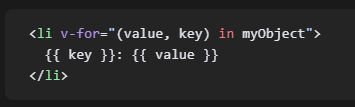
* **A v-else element must immediately follow a v-if or a v-else-if element - otherwise it will not be rec**

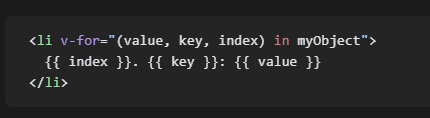
**LIST RENDERING**

* **A diretiva v-for pode ser usada para renderizar uma lista de items baseada em um array. Essa diretiva requer uma sintaxe especial na form de item in items, onde items é a matriz de dados de origem e item é um apelido para o elemento do array que está sendo iterado.**

**Exemplos:**

****

****

****