



GIỚI THIỆU VUE 3X

III NỘI DUNG CHÍNH

I

GIỚI THIỆU

II

CÁC TÍNH NĂNG TRÊN VUE 3

II

CÀI ĐẶT

1. Giới thiệu

1. Vue.js là gì?

- Vue (phát âm là / vju: /, like view) là một **progressive framework** để xây dựng giao diện người dung, mã nguồn được xây dựng 100% JS
 - Thư viện cốt lõi chỉ tập trung vào layout View và dễ dàng lấy và tích hợp với các thư viện khác hoặc các dự án hiện có.
 - Mặt khác, Vue cũng hoàn toàn có khả năng cung cấp năng lượng cho các ứng dụng một trang phức tạp
 - Vue đơn giản và linh hoạt hơn, được xây dựng ứng dụng đơn trang
 - Vue cho phép bạn chỉ tạo các phần cụ thể trong ứng dụng của mình.
- Bạn chỉ học những gì cần thiết cho vấn đề bạn đang giải quyết.**

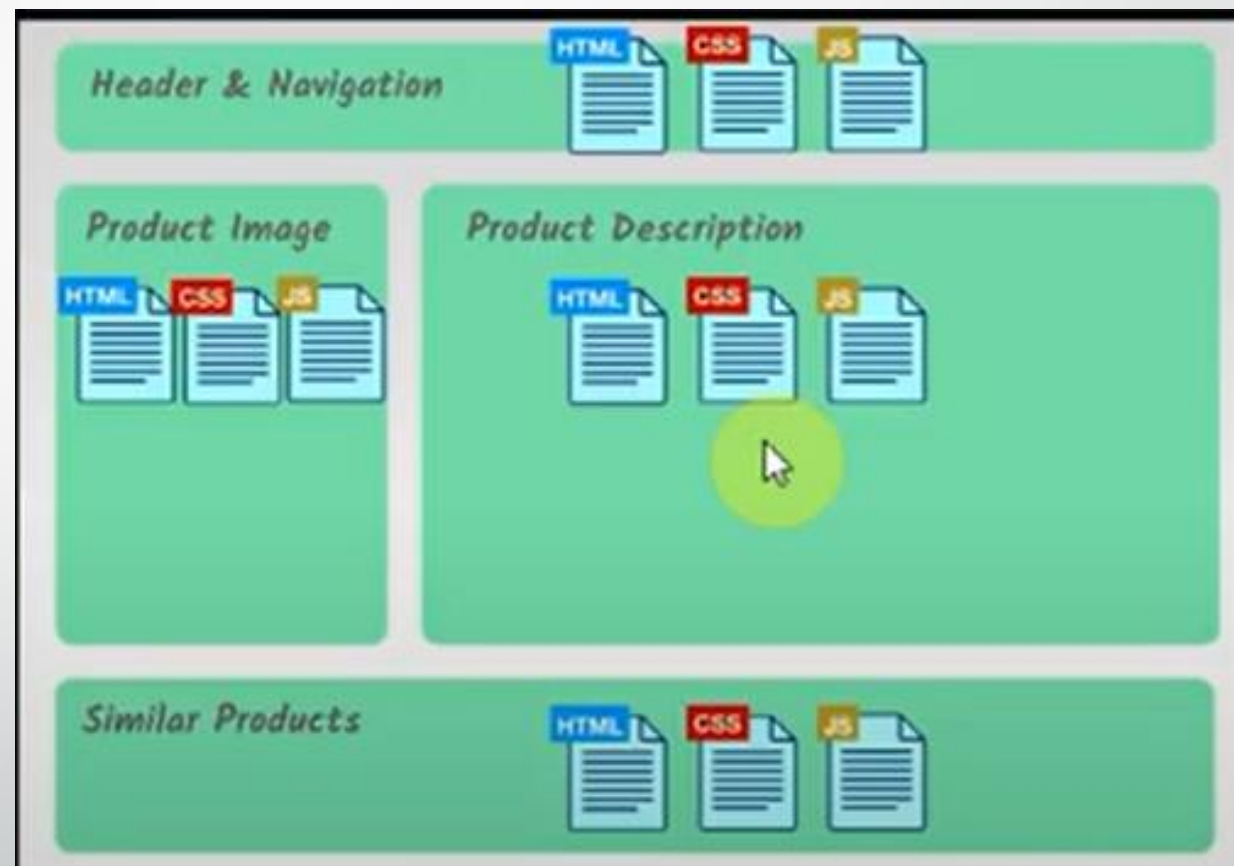
1. Giới thiệu

1. Vue.js là gì?



Châu Trần Trúc Ly

Trước



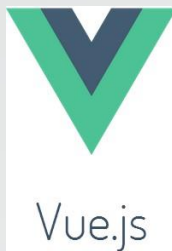
Sau

9/7/2024

4

1. Giới thiệu

2. Vue.js và Angular?



- **Vue** sử dụng Javascript, **Angular** dựa trên TypeScript
- **Vue** khá dễ học, đường cong học tập của **Angular** dốc hơn nhiều
- **Vue** được chỉ định nếu bạn đang làm việc một mình hoặc có một nhóm nhỏ, **Angular** dành cho các ứng dụng thực sự lớn.
- Trong khi Angular cung cấp ràng buộc dữ liệu để xử lý DOM mà không cần phải chạm trực tiếp vào nó, vẫn có những khái niệm bổ sung yêu cầu mã phải được cấu trúc theo một cách nhất định.
- Vue có những hiệu quả này, nhưng thậm chí còn nhẹ hơn.

2. Các tính năng trên Vue 3

1. Composition API (Now built-in)
2. Multiple root elements (Template syntax)
3. Suspense
4. Multiple V-models
5. Better Reactivity

2.1. Composition API (Now built-in)

Composition API là một trong những tính năng đáng chú ý nhất của Vue 3, cho phép các nhà phát triển tổ chức mã theo logic thay vì dựa trên các tùy chọn đối tượng. Điều này làm cho mã dễ tái sử dụng và bảo trì hơn.

Ví dụ, bạn có thể tạo ra các hàm riêng biệt để xử lý logic nghiệp vụ, sau đó tái sử dụng chúng trong nhiều thành phần khác nhau.

2.1. Composition API (Now built-in)

```
import { ref, computed } from 'vue';

export default {
  setup() {
    const count = ref(0);
    const doubled = computed(() => count.value * 2);

    function increment() {
      count.value++;
    }

    return { count, doubled, increment };
  }
}
```


2.2. Multiple Root Elements

Trong Vue 2, mỗi thành phần chỉ có thể có một phần tử gốc. Vue 3 loại bỏ hạn chế này, cho phép các thành phần có nhiều phần tử gốc, giúp cấu trúc mã linh hoạt hơn.

```
<template>
  <header>Header</header>
  <main>Main Content</main>
  <footer>Footer</footer>
</template>
```

2.3. Suspense

Suspense hỗ trợ quản lý trạng thái chờ của các thành phần bất đồng bộ, cho phép hiển thị một UI dự phòng trong khi chờ dữ liệu được tải.

```
<template>
  <Suspense>
    <template #default>
      <AsyncComponent />
    </template>
    <template #fallback>
      Loading...
    </template>
  </Suspense>
</template>
```

2.4. Multiple v-models

Vue-2 cho phép sử dụng chỉ một v-model trên một thành phần. Trong Vue-3, có thể liên kết bất kỳ số lượng v-model nào với các thành phần tùy chỉnh của mình:

```
<template>
  <survey-form v-model:name="name" v-model:age="age"> </survey-form>
</template>
```

//SurveyForm.vue

```
<template>
  <div>
    <label>Name: </label>
    <input :value="name" @input="updateName($event.target.value)" />
    <label>Age: </label>
    <input :value="age" @input="updateAge($event.target.value)" />
  </div>
</template>
<script>
```

```
export default {
  props: {
    name: String,
    age: Number
  },
  setup(props, { emit }) {
    const updateName = value => {
      emit('update:name', value)
    }
    const updateAge = value => {
      emit('update:age', +value)
    }
    return { updateName, updateAge }
  }
}
</script>
```

2.5. Better reactivity

Vue 2 đã có khả năng phản ứng tuyệt vời và có thể không gặp phải bất kỳ trường hợp nào mà chúng ta thấy rằng phản ứng là thiếu. Tuy nhiên, đã có một vài trường hợp Vue 2 bị hụt.

Trong Vue-3, chúng hoạt động trực tiếp mà không cần bất kỳ chức năng trợ giúp nào:

2.5. Better reactivity

```
export default {  
  setup() {  
    let list = ref([1, 2])  
    let a = ref(0)  
    let myObj = ref({ name: 'Preetish' })  
    function myFun() {  
      list.value[3] = 3  
      myObj.value.last = 'HS'  
      delete myObj.value.name  
    }  
    return { myFun, list, myObj }  
  }  
}
```

2.6. Ưu điểm của Vue3

Hiệu suất tốt hơn: Vue 3 nhẹ hơn và nhanh hơn Vue 2 nhờ vào cơ chế phản ứng cải tiến và cây ảo (Virtual DOM) tối ưu hơn.

Khả năng mở rộng cao: Composition API và multiple root elements làm cho Vue 3 linh hoạt hơn trong việc xây dựng các ứng dụng phức tạp.

Hỗ trợ TypeScript tốt hơn: Vue 3 được thiết kế để tương thích tốt hơn với TypeScript, mang lại trải nghiệm phát triển mạnh mẽ hơn cho các nhà phát triển sử dụng TypeScript.

2.7. Nhược điểm của Vue3

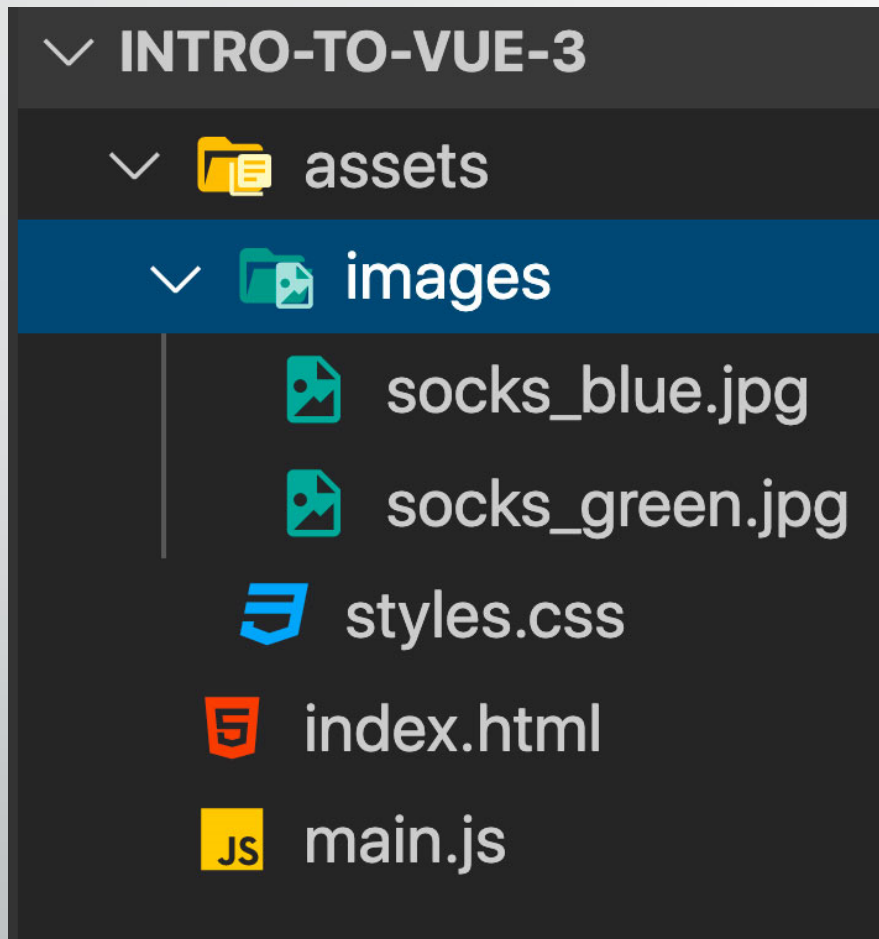
Đường học tập: Những người mới làm quen với Vue có thể thấy khó khăn khi tiếp cận với các khái niệm như Composition API.

Cộng đồng và tài nguyên: Mặc dù Vue 3 đang ngày càng phổ biến, nhưng số lượng tài nguyên và thư viện hỗ trợ vẫn còn ít hơn so với Vue 2.

Cần cập nhật: Các dự án hiện tại sử dụng Vue 2 có thể cần phải cập nhật và thích ứng với các API mới của Vue 3, điều này có thể đòi hỏi công sức và thời gian đáng kể.

2. Tạo project với CDN

Tìm hiểu thư mục



2. Tạo project với CDN

```
<head>
  <meta charset="UTF-8" />
  <title>Vue Mastery</title>
  <!-- Import Styles -->
  <link rel="stylesheet" href="./assets/styles.css" />
  <!-- Import Vue.js -->
  <script src="https://unpkg.com/vue@3.0.11/dist/vue.global.js"></script>
</head>
<body>
  <div id="app">
    <h1>{{ product }}</h1>
    <!-- Solution -->
    <p>{{ description }}</p>
    <!-- Solution -->
  </div>

  <!-- Import App -->
  <script src="./main.js"></script>

  <!-- Mount App -->
  <script>
    const mountedApp = app.mount('#app')
  </script>
</body>
```

2. Tạo project với CDN

Tạo ứng dụng với hàm createApp

- Để hiển thị dữ liệu của trong HTML, trước tiên phải tạo một ứng dụng Vue.
- Tạo file main.js

```
const app = Vue.createApp({})
```

Hoặc



```
const app = Vue.createApp({  
  data() {  
    return {  
      product: 'Socks'  
    }  
  }  
})
```

2. Tạo project với CDN

Tạo ứng dụng với hàm createApp

- Import ứng dụng Vue vào trong HTML
- Index.html

```
<!-- Import App -->  
<script src="./main.js"></script>
```

2. Tạo project với CDN

Tạo ứng dụng với hàm `createApp`

Các tùy chọn được chuyển đến `createApp()`, được sử dụng để chuyển đến thành phần gốc. Thành phần được sử dụng làm điểm bắt đầu cho việc kết xuất khi chúng ta **mount** ứng dụng

Mount: chèn 1 ứng dụng **vue** vào document, tức là nó được gắn vào phần tử nào

Index.html

```
<!-- Mount App -->
<script>
  const mountedApp = app.mount('#app')
</script>
```

2. Tạo project với CDN

Tạo ứng dụng với hàm createApp

- Bây giờ chúng ta phải chỉ ra một phần của HTML mà Vue.js sẽ xem xét và sử dụng.
- Chúng tôi sẽ tạo div với id “app”

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <title>Hello World - Vue.js</title>
    <script type="text/javascript" src="js/vue.js"></script>
  </head>
  <body>
    <div id="app">    </div>
  </body>
</html>
```

Châu Trần Trúc Ly

3. Ví dụ 1

```
<head>
  <meta charset="UTF-8" />
  <title>Vue Mastery</title>
  <!-- Import Styles -->
  <link rel="stylesheet" href="./assets/styles.css" />
  <!-- Import Vue.js -->
  <script src="https://unpkg.com/vue@3.0.11/dist/vue.global.js"></script>
</head>
<body>
  <div id="app">
    <h1>{{ product }}</h1>
    <!-- Solution -->
    <p>{{ description }}</p>
    <!-- Solution -->
  </div>

  <!-- Import App -->
  <script src="./main.js"></script>

  <!-- Mount App -->
  <script>
    const mountedApp = app.mount('#app')
  </script>
</body>
```

File HTML

Châu Trần Trúc Ly

```
const app = Vue.createApp({
  data() {
    return {
      product: 'Socks',
      // Solution
      description: 'A warm fuzzy pair of socks.'
      // Solution
    }
  }
})
```

File JS

9/7/2024

22

3. Ví dụ 1

```
<body>
  <div id="app">
    <p>Hello</p>
    <script src="./main.js"></script>
  </div>
</body>
</html>
```

File HTML

```
const app=Vue.createApp({
  data(){
    return{
    }
  }
}).mount( '#app' );
```

File JS



The
end