

Vue3 基礎

Vue 3 是一個流行的前端JavaScript框架，用於建構用戶界面和單頁應用程序。相比於 Vue 2，它帶來了許多新特性和改進，包括更快的渲染速度、更小的包大小以及Composition API的引入。以下是Vue 3的一些基礎知識，包括實例創建、模板語法、數據綁定、指令和事件處理的介紹和示例。

Vue3 實例

在Vue 3中，創建一個應用實例通過使用createApp方法，然後使用mount方法將其掛載到一個DOM元素上。例如：

```
import { createApp } from 'vue';
import App from './App.vue'; // 引入App組件

const app = createApp(App);
app.mount('#app');
```

這裏App.vue是一個Vue組件，#app是DOM中的一個元素的ID，Vue應用會在這個元素內渲染。

Vue3 模板語法

Vue模板提供了聲明式地將數據渲染進DOM的機制。模板語法包括插值、指令和JavaScript表達式等。

插值：

```
<span>{{ message }}</span>
```

在這個例子中，message是一個與Vue實例相關聯的數據屬性，它的值將被渲染到標籤內。

指令：

Vue指令是帶有v-前綴的特殊屬性，用於在模板上應用反應式行為。

```
<p v-if="seen">現在你看到我了</p>
```

v-if指令根據表達式seen的真偽值決定元素是否渲染。

數據綁定和插值

Vue提供了一種機制來實現數據和DOM元素之間的雙向綁定，最常見的是使用v-model指令。

```
<input v-model="message" placeholder="請輸入些什麼">
<p>消息: {{ message }}</p>
```

這裏，input元素的值和message數據屬性被雙向綁定。當輸入框的值發生改變時，message的值也會更新，反之亦然。

指令和事件處理

除了內置指令，Vue還允許使用v-on指令添加事件監聽器。

```
<button v-on:click="doSomething">點擊我</button>
```

這裏，當按鈕被點擊時，會調用與Vue實例關聯的doSomething方法。

結合上述基礎，你可以開始使用Vue 3建立動態的、互動豐富的Web應用。這只是Vue 3豐富特性的一小部分，隨著你對框架的深入了解，你將能夠開發出更加複雜和高效的應用。