

Vue.js 组件基础

主讲人：和凌志

主要内容

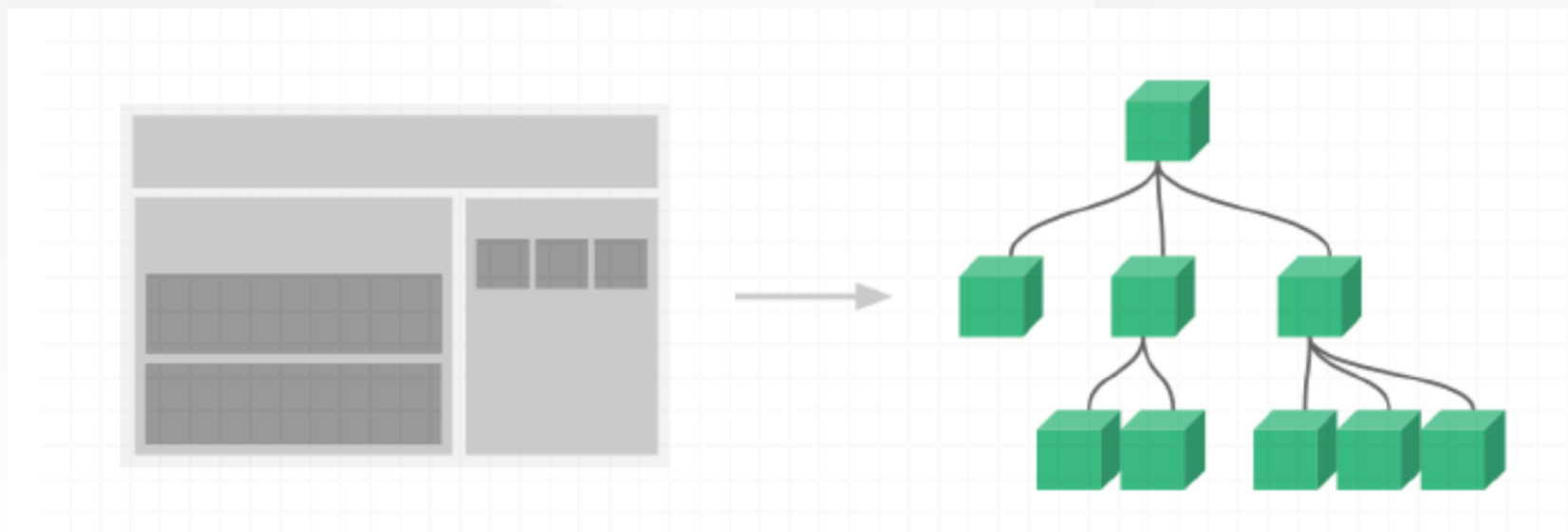
- 如何创建一个组件
- 组件注册：全局注册、局部注册
- 组件的应用



Vue 组件系统

Vue.js 组件系统

- 组件系统是Vue.js的一个重要概念，因为它提供了一种抽象，让我们可以用独立可复用的**小组件**来构建**大型应用**。



Vue.js 组件系统 示例

```
<div id="app">
```

```
<app-nav> </app-nav>
```

```
<app-view>
```

```
<app-sidebar> </app-sidebar>
```

```
<app-content> </app-content>
```

```
</app-view>
```

```
</div>
```

组件的注册与调用

组件名的大小写

- 在注册一个组件时，需要给它一个名字。比如：

```
Vue.component('my-component-name', { /* ... */ })
```

- 该组件名就是 Vue.component 的第一个参数。
- 组件名 (字母全小写且必须包含一个连接符)。这会帮助你避免和当前以及未来的 HTML 元素相冲突。

组件： 全局注册

- 用 `Vue.component` 来创建组件：

```
Vue.component('my-component-name', {  
  // ... 选项 ...  
})
```

- 这些组件是全局注册的。也就是说，它们在注册之后可以用在任何新创建的 Vue 根实例 (`new Vue`) 的模板中。比如：

```
Vue.component('component-a', { /* ... */ })  
Vue.component('component-b', { /* ... */ })  
Vue.component('component-c', { /* ... */ })  
  
new Vue({ el: '#app' })
```


组件的调用

- 在所有子组件中也是如此，也就是说，这三个组件在各自内部也都可以相互使用。

```
<div id="app">  
  <component-a></component-a>  
  <component-b></component-b>  
  <component-c></component-c>  
</div>
```

组件： 局部注册

- 全局注册往往是不够理想的。比如，如果使用一个像 webpack 这样的构建系统，全局注册所有的组件意味着即便已经不再使用一个组件了，它仍然会被包含在你最终的构建结果中。这造成了用户下载的 JavaScript 无谓增加。
- 在这些情况下，可以通过一个普通的 JavaScript 对象来定义组件。

组件：局部注册

- 通过一个普通的 JavaScript 对象来定义组件：

```
var ComponentA = { /* ... */ }  
var ComponentB = { /* ... */ }  
var ComponentC = { /* ... */ }
```

- 然后在 components 选项中定义你想要使用的组件：

```
new Vue({  
  el: '#app',  
  components: {  
    'component-a': ComponentA,  
    'component-b': ComponentB  
  }  
})
```

组件：局部注册

- 注意局部注册的组件在其子组件中 *不可用*。例如，如果你希望 ComponentA 在 ComponentB 中可用，则需要这样写：

```
var ComponentA = { /* ... */ }  
  
var ComponentB = {  
  components: {  
    'component-a': ComponentA  
  },  
  // ...  
}
```

动手实验

小结

✓ 掌握 Vue 组件概念