

## 课程目标

01

事件

02

双向绑定

03

样式

## 事件

### 事件

```
1 <template>
2   <div>
3     {{ count }}
4     <button v-on:click="add">add</button>
5   </div>
6 </template>
7 <script>
8   export default {
9     data() {
10      return {
11        count: 9
12      };
13    },
14    methods: {
15      add(event) {
16        // `this` 在方法里指向当前 Vue 实例
17        this.count += 1;
18        // `event` 是原生 DOM 事件
19        if (event) {
20          alert(event instanceof MouseEvent);
21        }
22      }
23    }
24  };
25 </script>
```

@click 现在可这么写

1016663971

## 事件修饰符

为了保证methods方法只有纯粹的数据逻辑(和DOM解耦, 易于单元测试), 不去处理DOM相关操作。常见事件修饰符

```
<!-- 阻止单击事件继续传播 -->
<a v-on:click.stop="doThis"></a>

<!-- 提交事件不再重载页面 -->
<form v-on:submit.prevent="onSubmit"></form>

<!-- 修饰符可以串联 -->
<a v-on:click.stop.prevent="doThat"></a>

<!-- 只有修饰符 -->
<form v-on:submit.prevent></form>

<!-- 添加事件监听器时使用事件捕获模式 -->
<!-- 即内部元素触发的事件先在此处理, 后才交由内部元素进行处理 -->
<div v-on:click.capture="doThis">...</div>

<!-- 只当在 event.target 是当前元素自身时触发处理函数 -->
<!-- 即事件不是从内部元素触发的 -->
<div v-on:click.self="doThat">...</div>

<!-- 点击事件将只会触发一次 -->
<a v-on:click.once="doThis"></a>

<!-- 滚动事件的默认行为 (即滚动行为) 将会立即触发 -->
<!-- 而不会等待 `onScroll` 完成 -->
<!-- 这其中包含 `event.preventDefault()` 的情况 -->
<div v-on:scroll.passive="onScroll">...</div>

<!-- 只有在 `key` 是 `Enter` 时调用 `vm.submit()` -->
<input v-on:keyup.enter="submit" />
```

## 自定义事件



\$emit( )



```

1  <template>          父组件
2    <div>
3      {{msg}}
4      <event-parent-child
5        :msg="msg"      实时监听msg变化，并把子
6        @update:msg="msg = $event" 组件传递
7      ></event-parent-child> 过来值更
8    </div>              新给到
9  </template>           msg

```

```

1  <template>          子组件
2    <button
3      @click="$emit('update:msg', 'hi')"
4      say {{ msg }}    更新父组件传递过来的msg
5    </button>
6  </template>
7  <script>
8    export default {
9      props: ["msg"]
10   };
11  </script>

```

同时也可直接调用父组件方法

```

this.$emit( event: 'after-confirm', Object.assign(this.form, this.extraParameter))

```

同时也可子组件直接调用父组件中的某个方法，后面是这个调用这个方法需要的参数。

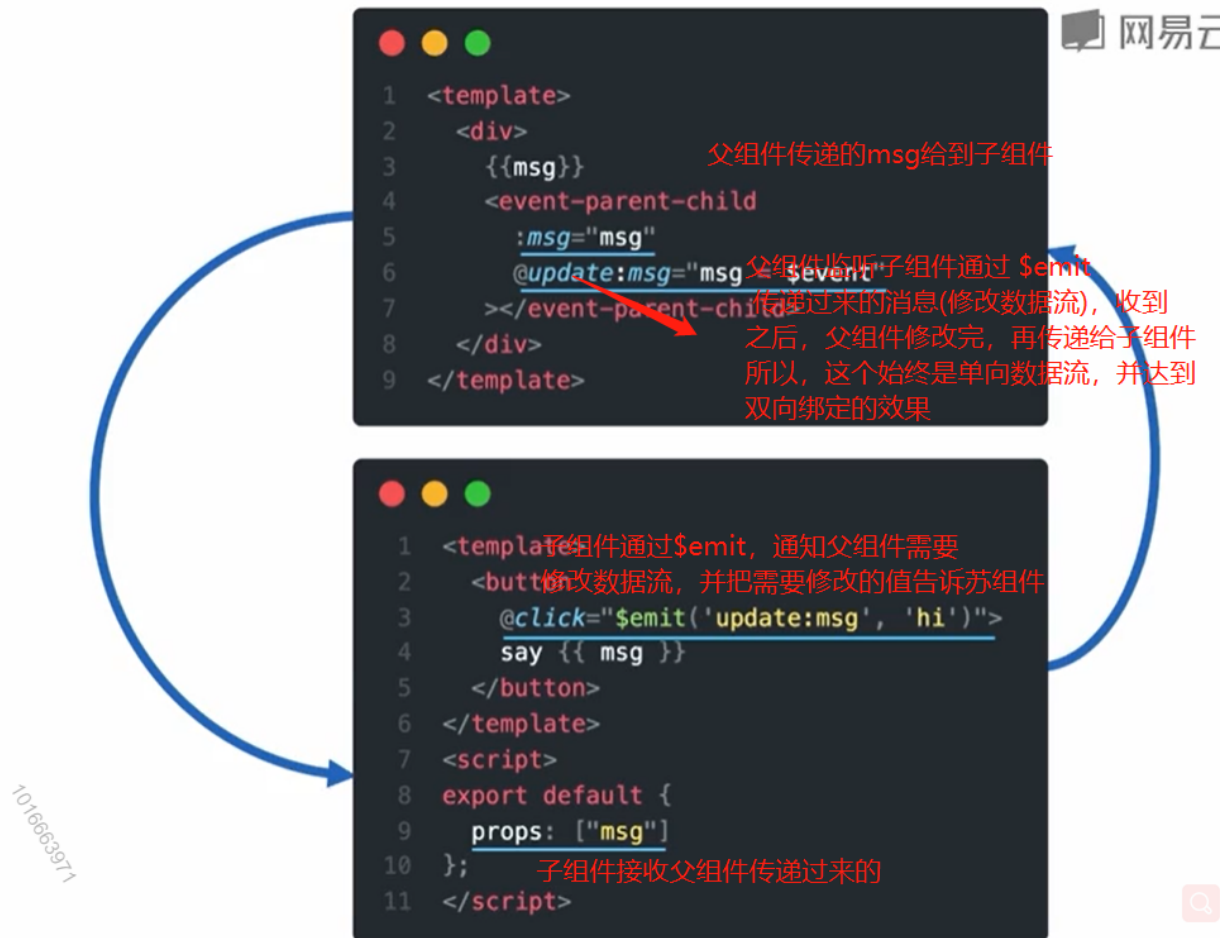
t>

```
<!-- 更换负责人 -->
<ChangePrincipal :visible.sync="dialog.change" :extraParameter.sync="contract" :inAction.sync="inAction.change" @after-confirm="afterChangePrincipal"/>
<!-- 二次弹窗 -->
<ConfirmDialog
  :visible.sync="dialog.confirm"
  @after-confirm="afterChangePrincipal"/>
```

监听子组件相关的方法调用

## 双向绑定

vue是单向数据流，为什么能进行数据双向绑定呢？



## 双向绑定

```
1 <template>
2   <div>
3     {{msg}}
4     <event-parent-child
5       :msg.sync="msg"
6     ></event-parent-child>
7   </div>
8 </template>
```

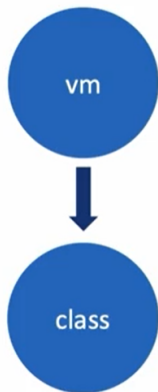
直接简写成sync

```
1 <template>
2   <div>
3     {{msg}}
4     <event-parent-child
5       :msg="msg"
6       @update:msg="msg = $event"
7     ></event-parent-child>
8   </div>
9 </template>
```

```
1 <template>
2   <button
3     @click="$emit('update:msg', 'hi')"
4     say {{ msg }}
5   ></button>
6 </template>
7 <script>
8   export default {
9     props: ["msg"]
10  };
11 </script>
```

## 样式

### class



```
1 <template>
2   <div
3     class="ui-demo"
4     v-bind:class="{ 'ui-demo_status--active': isActive,
5                     'ui-demo_status--error': hasError }"
6   ></div>
7 </template>
8 <script>
9   export default {
10     data() {
11       return {
12         isActive: false,
13         hasError: false
14       };
15     }
16   };
17 </script>
```

网易云课堂

```
1 if (true)
2 if ([])
3 if ([])
4 if (42)
5 if ("foo")
6 if (new Date())
7 if (-42)
8 if (3.14)
9 if (-3.14)
10 if (Infinity)
11 if (-Infinity)
```

truthy

```
false
0
""
null
undefined
NaN
```

## | style

```
1 <template>
2   <div
3     class="ui-demo"
4     v-bind:style="{ 'width': `${width}px`,
5                     'height': `${height}px`,
6                     'backgroundColor': bgColor }"
7   ></div>
8 </template>
9 <script>
10 export default {
11   data() {
12     return {
13       width: "100",
14       height: "100",
15       bgColor: "#fff"
16     };
17   }
18 };
19 </script>
```

## Scoped

 网易云课堂

## BEM

```
1 <template>
2   <div class="test">hi</div>
3 </template>
4
5 <style scoped>
6   .test {
7     color: red;
8   }
9 </style>
```

```
<div id="app">
  <div data-v-469af010 class="test">hi</div> == $0
</div>
```

```
<style type="text/css">
.test[data-v-469af010] {
  color: red;
}
</style> == $0
```