

## 09. 事件处理能力

学习要点：

1. 事件处理
2. 事件修饰符

本节课我们来开始学习 Vue 的事件处理以及事件修饰符功能。

### 一. 事件处理

1. 先来个简单的事件处理，热热身，比如：每次点击按钮增加统计+1；

```
<button v-on:click="counter += 1">增加</button>
<div>counter : {{counter}}</div>
```

```
const dataObj = {
  counter : 0,
};
```

2. 我们可以通过点击执行方法，并且处理事件对象；

```
<button v-on:click="greet">执行事件</button>
```

```
methods : {
  greet(event) {
    alert(this.message + event.target.innerText);
  }
}
```

PS: (1).event 参数可以不用显式的接受，也可以获取到。

(2).如果执行没有参数，而接受任意参数，也会当作 event 对象；

3. 我们也可以实现执行方法中传递参数，如果要传递 event，请\$event；

```
<button v-on:click="say('world!', $event)">传递参数</button>
```

```
say(info, event) {
  alert('Hello, ' + info);
}
```

4. 事件对象有很多作用，比如阻止冒泡；

```
<div v-on:click="say('div', $event)">
  <button v-on:click="say('world!')">传递参数</button>
</div>
```

## 二. 事件修饰符

1. 系统提供了常用的事件对象修饰符，让你直接定义视图中事件的执行方式：

```
<button v-on:click.stop="say('world!')">传递参数</button>
```

PS：以下是手册提供的常用对象事件修饰符，自行查阅使用；

```
<!-- 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，然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>
```

```
<!-- 2.1.4 新增点击事件将只会触发一次 -->
```

```
<a v-on:click.once="doThis"></a>
```

```
<!-- 2.3.0 新增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>
```

2. 再选择两个常用的比如：只运行执行一次 `once`，取消默认事件 `prevent`；

```
<button v-on:click.once="greet">执行事件</button>
```

```
<form action="http://www.baidu.com">
```

```
  <button type="submit" v-on:click.prevent="">提交</button>
```

```
</form>
```

3. 系统还提供了按键事件修饰符，比如回车键按下触发；

```
<input v-on:keyup.enter="submit">
```

.enter / .tab / .delete / .esc / .space / .up / .down / .left / .right