

1.在前端开发中，我们经常使用交互。

- 这个时候，我们就必须监听用户发生的事件，比如：点击、拖拽、键盘事件等等
- 在Vue中如何监听事件呢？使用v-on指令

一、v-on介绍

- **作用：**绑定事件监听器
- **缩写：**@
- **预期：**Function | Inline Statement | Object
- **参数：**event

二、v-on基础

1.这里我们用一个监听按钮的点击事件，来简单看看v-on的使用

- 下面代码中，我们使用了v-on:click="counter++"

```
<button v-on:click="counter++">+</button>
<button v-on:click="counter--">-</button>
```

- 另外，我们可以将事件指向一个methods中定义的函数：v-on:click="increment"

```
<button v-on:click="increment">+</button>
<button v-on:click="decrement">-</button>
</div>
<script>
  const app = new Vue({
    el: '#app',
    data: {
      counter: 0,
    },
    methods: {
      increment() {
        this.counter++;
      },
      decrement() {
        this.counter--;
      },
    },
  });
```

- 注：v-on也有对应的语法糖：

v-on:click可以写成@click

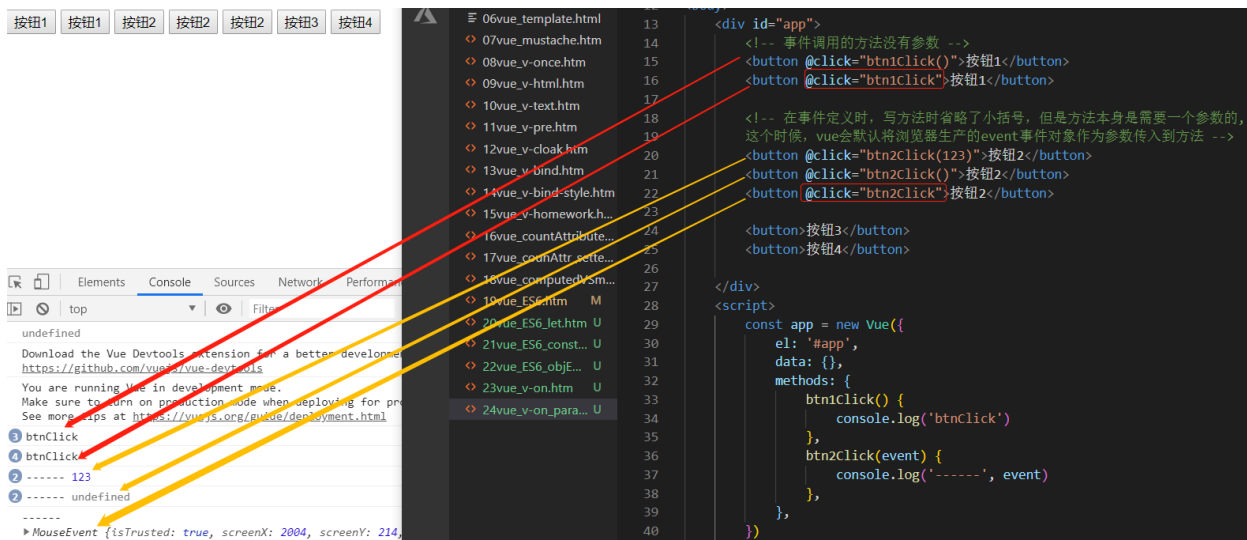
```
<button @click="increment">+</button>
<button @click="decrement">-</button>
```

三、v-on参数

1.当通过methods中定义方法，以供@click调用时，**需注意参数问题**：

- 情况一：如果该方法不需要额外参数，那么**方法后的()可以不添加**(满足条件：1.在事件监听的时候 2.方法不需要额外参数)。

但是注意：如果方法本身中有一个参数，那么会默认将原生事件event参数传递进去



- 情况二：如果需要同时传入某个参数，同时需要event时，可以通过\$event传入事件



四、v-on修饰符

1.在某些情况下，我们拿到event的目的可能是进行一些事件处。

2. Vue提供了修饰符来帮助我们方便的处理一些事件:

- .stop---调用event.stopPropagation(),停止冒泡 (在div还有其他内容时, 使用按钮)
- .prevent---调用event.preventDefault(), 阻止默认行为
- .{keyCode | keyAlias}---只当事件是从特定键触发时才触发回调, 监听某个键盘键帽
- .native---监听组件根元素的原生事件
- .once---只触发一次回调

The screenshot illustrates the use of Vue.js event modifiers in a web application. The UI consists of a form with the following elements:

- A button labeled "按钮" (Button).
- A button labeled "提交" (Submit).
- An input field with the value "dfdvaaa".
- A button labeled "按钮2" (Button 2).

The console shows the following log entries:

- div click
- btn click
- submitClick
- keyUp
- btn2Click
- fvfd

The code on the right shows the corresponding Vue.js template and script:

```
<div id="app">
  <!-- 1. .stop修饰符的使用: 停止冒泡 -->
  <div @click="divClick">
    aaaaa
    <button @click.stop="btnClick">按钮</button>
  </div>

  <!-- 2. .prevent修饰符的使用: 阻止默认行为 -->
  <form action="baidu">
    <input type="submit" value="提交" @click.prevent="submitClick">提交
  </form>

  <!-- 3. 监听某个键盘键帽: 键别名 -->
  <input type="text" @keyup.a="keyUp">

  <!-- 4. .once修饰符的使用: 点击回调实例触发一次 -->
  <button @click.once="btn2Click">按钮2</button>
</div>
<script>
  const app = new Vue({
    el: '#app',
    data: {
      counter: 0,
    },
    methods: {
      btnClick() {
        console.log("btnClick");
      },
      divClick() {
        console.log("divClick");
      },
      submitClick() {
        console.log("submitClick");
      },
      keyUp() {
        console.log("keyUp");
      },
      btn2Click() {
        console.log("btn2Click");
      }
    }
  });
</script>
```