事件监听可以使用 v-on 指令：

<template>

  <div id="app">

    <!-- `greet` 是在下面定义的方法名 -->

    <button v-on:click="greet">G</button>

  </div>

</template>

<script>

export default {

  name: "app",

  data() {

    return {

      name: "Vue.js"

    };

  },

  // 在 `methods` 对象中定义方法

  methods: {

    greet: function(event) {

      // `this` 在方法里指当前 Vue 实例

      alert("Hello " + this.name + "!");

      // `event` 是原生 DOM 事件

      if (event) {

        alert(event.target.tagName);

      }

    }

  }

};

</script>

除了直接绑定到一个方法，也可以用内联 JavaScript 语句：

<button v-on:click="say('hi')">Say hi</button>

**事件修饰符**

Vue.js 为 v-on 提供了事件修饰符来处理 DOM 事件细节，如：event.preventDefault() 或 event.stopPropagation()。

Vue.js通过由点(.)表示的指令后缀来调用修饰符。

    <!-- 阻止单击事件冒泡 -->

    <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>

    <!-- 添加事件侦听器时使用事件捕获模式 -->

    <div v-on:click.capture="doThis">...</div>

    <!-- 只当事件在该元素本身（而不是子元素）触发时触发回调 -->

    <div v-on:click.self="doThat">...</div>

    <!-- click 事件只能点击一次，2.1.4版本新增 -->

    <a v-on:click.once="doThis"></a>

**按键修饰符**

Vue 允许为 v-on 在监听键盘事件时添加按键修饰符：

<!-- 只有在 keyCode 是 13 时调用 vm.submit() -->

<input v-on:keyup.13="submit">

记住所有的 keyCode 比较困难，所以 Vue 为最常用的按键提供了别名：

<!-- 同上 -->

<input v-on:keyup.enter="submit">

<!-- 缩写语法 -->

<input @keyup.enter="submit">

全部的按键别名：

.enter

.tab

.delete (捕获 "删除" 和 "退格" 键)

.esc

.space

.up

.down

.left

.right

.ctrl

.alt

.shift

.meta