

v-on 指令常用修饰符：

- `.stop` - 调用 `event.stopPropagation()`，禁止事件冒泡。
- `.prevent` - 调用 `event.preventDefault()`，阻止事件默认行为。
- `.capture` - 添加事件侦听器时使用 `capture` 模式。
- `.self` - 只当事件是从侦听器绑定的元素本身触发时才触发回调。
- `.{keyCode | keyAlias}` - 只当事件是从特定键触发时才触发回调。
- `.native` - 监听组件根元素的原生事件。
- `.once` - 只触发一次回调。
- `.left` - (2.2.0) 只当点击鼠标左键时触发。
- `.right` - (2.2.0) 只当点击鼠标右键时触发。
- `.middle` - (2.2.0) 只当点击鼠标中键时触发。
- `.passive` - (2.3.0) 以 `{ passive: true }` 模式添加侦听器

注意：如果是在自己封装的组件或者是使用一些第三方的UI库时，会发现并不起效果，这时就需要用`.native`修饰符了，如：

//使用示例：

```
<el-input
  v-model="inputName"
  placeholder="搜索你的文件"
  @keyup.enter.native="searchFile(params)"
>
</el-input>
```

v-bind 指令常用修饰符：

- `.prop` - 被用于绑定 DOM 属性 (property)。 (差别在哪里?)
- `.camel` - (2.1.0+) 将 kebab-case 特性名转换为 camelCase. (从 2.1.0 开始支持)
- `.sync` (2.3.0+) 语法糖，会扩展成一个更新父组件绑定值的 `v-on` 侦听器。

v-model 指令常用修饰符：

- `.lazy` - 取代 `input` 监听 `change` 事件
- `.number` - 输入字符串转为数字
- `.trim` - 输入首尾空格过滤