





=====》





主要是因为嵌套v-for循环中第二层循环的属性值，需要取决于第一层循环的传参值；

计算属性不能传值，只能利用已有参数做出相应的某种变化，想使用灵活需要内部依赖的参数变化来调整；  
使用methods来做，根据传参返回不同的值。

.native：原生事件绑定到组件的根元素（js原生事件如：click，focus等），如：v-on:focus.native；绑定的是js自己的原生事件而不是组件内自定义的事件（注意：如果根元素是<label>标签等 ，label 元素不会向用户呈现任何特殊效果。当用户选择该标签时，浏览器就会自动将焦点转到和标签相关的表单控件上，就是说本身不会触发click等事件）

vue开发请注意参数的命名，当使用Vue组件时，不能用驼峰式命名v-on监听的事件（Note that HTML attributes are case-insensitive and you cannot use v-on to listen to camelCase events when using in-DOM templates. ）

即使是组件模板也只能有一个根元素包含所有子元素（类似于Vue实例）（Component template should contain exactly one root element. If you are using v-if on multiple elements, use v-else-if to chain them instead.）

组件的默认属性：key

参数使用kebab-case方式命名记得用单引号括起来；

一个对象的 v-for ：v-for="(value, key, index) in object”

**优雅的使用 icon踩坑日记**

1. 所有的在别处定义的全局定义都必须导入到main.js，才能生效;
2. @/这是webpack设置的路径别名。代表着到src这个文件夹的路径，与vue无关
3. import默认导入index命名的文件

**Vuex 插件推荐**

1. **状态持久化**

vuex-persistedstate  使用浏览器的本地存储（ local storage ）对状态（ state ）进行持久化。这意味着刷新页面或关闭标签页都不会删除你的数据。一个很好的例子就是购物车：如果用户不小心关闭了一个标签，他们可以重新打开并回到之前页面的状态。

**2. 同步标签页、窗口**

vuex-shared-mutations  可在不同的标签页之间同步状态。它通过 mutation 将状态储存到本地存储（local storage）来实现。选项卡、窗口中的内容更新时触发储存事件，重新调用 mutation ，从而保持状态同步。

**3. 语言本地化**

vuex-i18n  允许你轻松地用多种语言存储内容。让你的应用切换语言时更容易。一个很酷的功能是你可以存储带有标记的字符串，比如"Hello {name}, this is your Vue.js app."。所有的翻译版本都会在标记的地方使用相同的字符串。

**4. 管理多个加载状态**

vuex-loading  有助于你管理应用中的多个加载状态。这个插件适用于状态变化频繁且复杂的实时应用程序。

**5. 缓存操作**

vuex-cache  可以缓存 Vuex 的 action。例如，如果你从服务器检索数据，这个插件将在第一次调用该 action 时缓存结果，然后在之后的dispatch中，直接返回缓存的值。必要时清除缓存也很简单。