**Vue命令：**

指令带有前缀 v-，以表示它们是 Vue 提供的特殊属性。

1. v-bind  主要用于属性绑定，Vue官方提供了一个简写方式 **:bind** ，标签：[vue.js指令 v-bind :src :class   :style 属性绑定](http://www.mamicode.com/so/1/vue.js%e6%8c%87%e4%bb%a4_2_v-bind_2__2_%3asrc_2__2_%3aclass_2_%26nbsp%3b_2_%3astyle_2__2_%e5%b1%9e%e6%80%a7%e7%bb%91%e5%ae%9a)
2. v-for 指令可以绑定数组的数据来渲染一个项目列表
3. v-if 指令可以判断数据来渲染一个项目列表
4. v-else-if，顾名思义，充当 v-if 的“else-if 块”，可以连续使用：
5. v-else 元素必须紧跟在带 v-if 或者 v-else-if 的元素的后面，否则它将不会被识别。
6. v-on 指令绑定一个事件监听器，它用于监听 DOM 事件，通过它调用我们 Vue 实例中定义的方法
7. v-model 指令，它能轻松实现表单输入和应用状态之间的双向绑定。
8. [v-once 指令](https://cn.vuejs.org/v2/api/#v-once)，你也能执行一次性地插值，当数据改变时，插值处的内容不会更新。但请留心这会影响到该节点上所有的数据绑定
9. v-html 指令，输出真正的 HTML，双大括号会将数据解释为普通文本，而非 HTML 代码
10. v-bind 和 v-on 这两个最常用的指令，提供了特定简写： v-bind缩写，v-bind:href => :href; v-on 缩写，v-on:click => @click;
11. 修饰符是以半角句号 . 指明的特殊后缀，用于指出一个指令应该以特殊方式绑定。例如，.prevent 修饰符告诉 v-on 指令对于触发的事件调用 event.preventDefault()：
12. 用key管理可复用的元素，Vue 会尽可能高效地渲染元素，通常会复用已有元素而不是从头开始渲染。这么做除了使 Vue 变得非常快之外，还有其它一些好处。例如，如果你允许用户在不同的登录方式之间切换：（1）在代码中切换 loginType 将不会清除用户已经输入的内容。因为两个模板使用了相同的元素，<input> 不会被替换掉——仅仅是替换了它的 placeholder。（2） Vue 为你提供了一种方式来表达“这两个元素是完全独立的，不要复用它们”。只需添加一个具有唯一值的 key 属性即可。每次切换时，输入框都将被重新渲染。 代码区别:<input placeholder:”Enter your username”> <input placeholder:”Enter your email address”> <input placeholder=”Enter your username” key=”username-input”> <input placeholder=”Enter your email address” key=”email-input”>
13. V-show另一个用于根据条件展示元素的选项是 v-show 指令。用法大致一样：

|  |
| --- |
| <h1 v-show="ok">Hello!</h1>  不同的是带有 v-show 的元素始终会被渲染并保留在 DOM 中。v-show 只是简单地切换元素的 CSS 属性 display。  注意，v-show 不支持 <template> 元素，也不支持 v-else。 |

1. 当 v-if 与 v-for 一起使用时，v-for 具有比 v-if 更高的优先级。
2. v-if vs v-show: 一般来说，v-if 有更高的切换开销，而 v-show 有更高的初始渲染开销。因此，如果需要非常频繁地切换，则使用 v-show 较好；如果在运行时条件很少改变，则使用 v-if 较好。