# 风格指南-Vue

参考： <https://cn.vuejs.org/v2/style-guide/>

参考： <https://youzan.github.io/vant/#/zh-CN/style-guide>

**组件名为多个单词（component目录）**

组件名应该始终是多个单词，根组件App除外。

这样做可以避免跟现有以及未来的HTML元素相冲突，因为所有的HTML元素名称都是单个单词的。

考虑到每家公司都有前缀，因此我们自己写的组件，可以以Yilian作为前缀。

但view目录下，因为直接是页面，所以不需要是多个单词。

**单文件组件文件的名称**

单文件组件的文件名始终是单词大写开头，即PascalCase

**紧密耦合的组件名**

以父组件紧密耦合的子组件应该以父组件名作为前缀。

|  |
| --- |
| // bad  components/  |- TodoList.vue  |- TodoItem.vue  └─ TodoButton.vue  // good  components/  |- TodoList.vue  |- TodoListItem.vue  └─ TodoListItemButton.vue |

**组件数据**

组件的data必须是一个函数

当在组件中使用data属性的时候（除了new Vue外的任何地方），它的值必须是返回一个对象的函数。

详解：当data的值是一个对象时，它会在这个组件的所有实例之间共享。

**自闭合组件**

在单文件组件中没有内容的组件应该是自闭合的。

！！！DOM模板中不支持自闭合组件

如： <my-component />

**Prop定义**

Prop定义应该尽量详细，至少需要制定其类型。

- 写明组件的API，很容易看懂组件的用法

- 在开发环境下，如果向一个组件提供格式不正确的prop，Vue将会告警，以帮助你捕获错误来源

**Prop名大小写**

在声明prop的时候，其命名应该始终使用camelCase，而在模板中应该始终使用kebab-case。

**Props顺序 [ 尽量做到，不强制要求 ]**

标签的Props应该有统一的属性，依次为指令、属性和事件。

|  |
| --- |
| <my-component  v-if="if"  v-show="show"  v-model="value"  ref="ref"  :key="key"  :text="text"  @input="onInput"  @change="onChange"  /> |

**指令缩写**

用 : 表示v-bind: , 用 @ 表示 v-on:

**组件选项的顺序 [ 尽量做到，不强制要求 ]**

组件选项应该有统一的顺序。

|  |
| --- |
| export default {  name: '',  mixins: [], //混入对象列表。  props: {},  data() {},  computed: {},  watch: {},  methods: {}  created() {},  mounted() {},  destroyed() {},  components: {},  }; |

**为v-for设置键值**

总是用key配合v-for

**避免v-if和v-for用在一起**

永远不要把v-if和v-for同时用在一个元素上。

一般我们在两种常见的情况下会倾向于这样做：

- 为了过滤一个列表中的项目 (比如 v-for="user in users" v-if="user.isActive")。在这种情形下，请将 users 替换为一个计算属性 (比如 activeUsers)，让其返回过滤后的列表。

- 为了避免渲染本应该被隐藏的列表 (比如 v-for="user in users" v-if="shouldShowUsers")。这种情形下，请将 v-if 移动至容器元素上 (比如 ul, ol)。

**为组件样式设置作用域 [ 基本上都要用，统一的样式可以写到style中 ]**

对于应用来说，顶级 App 组件和布局组件中的样式可以是全局的，但是其它所有组件都应该是有作用域的。

- 使用scoped [个人推荐在组件里使用 ]

- 使用CSS Modules

- 使用前缀，比如.yilian- [ 个人推荐写全局样式 ]

**多个特性的元素[ 推荐，增加易读性 ]**

多个特性的元素应该分多行撰写，每个特性一行

|  |
| --- |
| <MyComponent  foo="a"  bar="b"  baz="c"  /> |