# Vue代码开发规范

1. 基础规范
2. 命名规范
   1. 命名尽量简洁、明确，不超过三个单词。
   2. 基于文件夹，有层次概念。例：

user------Login.vue、Register.vue

device------List.vue、Detail.vue

* 1. 文件夹命名：

公共组件中的文件夹采用首字母大写的驼峰命名：GlobalHeader

其他文件夹统一采用首字母小写的驼峰命名：app

* 1. 文件命名：

Vue组件采用首字母大写的驼峰命名：Login.vue

html、js文件采用首字母小写的驼峰命名：main.js

资源文件使用小写字母+横线命名：icon-next.png

* 1. 其他命名：

组件名称使用首字母大写驼峰命名：TopMenu

方法、变量、自定义参数首字母小写驼峰命名：getName、userInfo

事件处理的方法用handle开头：handleClick

css样式小写字母+横线命名：normal-style

1. 格式规范
   1. 代码缩进统一为2空格
   2. 文件头、方法、关键代码添加注释
   3. 符合eslint检查规范（解决eslint相关报错）
   4. 公共方法统一提取，放入utils中
   5. 公共组件统一提取，放入components中
2. vue文件
3. 整体文件格式如下：

<template>

<div>

</div>

</template>

<script>

export default {}

</script>

<style lang=”less” scoped>

</style>

1. template
   1. 标签统一为小写字母+横线：div、app-info
   2. 无子元素的标签采用自闭写法：<div />
   3. 元素属性超过三个，分行编写，每行一个属性
   4. 使用Vue语法中的缩写方式：

v-bind:key--->:key

v-on:click--->@click

自定义的方法也使用@fun

* 1. v-for必须设置key属性，且一般不为index。
  2. 可复用代码，提取公共组件

1. script
   1. 书写顺序：

import：引入统一使用@/path，部分组件内部引用例外

export defalut {

name：首字母大写的驼峰命名

components：引用的组件

props：组件可变参数，必须声明type和default

data：组件内数据变量

computed：计算属性

watch：侦听属性

声明周期方法：必须按照生命周期顺序书写beforeCreate-created...

网络请求放在created方法中

methods：事件处理，js方法，注意公共方法的提取

}

1. style
   1. 尽量使用scope方式
   2. 命名使用小写字母+横线：top-div
   3. 每个css样式之间插入空行
   4. 最外部样式使用与组件名称命名相同的样式包裹：.app-info{}

避免组件间的样式冲突，scope方式下不会影响

* 1. 尽量使用精准样式

background---->background-image、background-color

* 1. 保持每行一个属性
  2. color值统一小写，且尽量简写，#fafcfb，#ccc
  3. 公共样式可写入App.vue
  4. 注意less中样式嵌套的使用
  5. 伪类样式书写：

.top {

&:hover {}

}

1. modules.js文件
   1. state：首字母小写驼峰命名，添加必要注释，给出默认值
   2. mutations：大写字母+下划线命名：SET\_NAME
   3. actions：驼峰命名，需要返回或处理resolve和reject两种结果
   4. getters：按module编写，每个module的参数写在一起，给出必要的注释