

## 【规范】31 Vue项目规范

所有软件项目都避免不了，随着人员和模块的增加，代码会渐渐“腐烂”。——  
《CodeComplete》

### ✳ 规范本身的规范

1. 简洁第一：规则不能超过 3 点
2. 统一：本规范与易仓的总架构项目规范一致
3. BEM 思想：模块-元素-特性的思想，贯穿各个种类的规范

### 📁 各个种类的规范

#### 文件夹

1. 菜单式编程：文件夹以 index 作为唯一接口。index 调用各种模块，把耦合度降到最低

比如路由，index.js 作为对外接口，config 是配置，routes 是数据

/router

|--index.js // 导出一个对象，让 Vue 使用 Vue.use(VueRouter)

|--config.js // 导出一些配置给 index.js 使用，比如 scrollBehavior(), 尽量以装饰函数的形式

|--routes.js // 导出一个路由数组，让 index.js 使用

2. 自说明：各个文件夹会有一份 README.md，用于说明当前文件夹的管理规则
3. 复杂度控制：每个文件尽量不要超过 200 行

#### HTML

1. 遵守《Vue 风格指南》
2. 组件化：要做到“所见即所得”，不允许堆砌 div

比如页面由 3 部分组成：用户信息、商铺列表，那这个页面就是以下结构

```
<div class="home">
```

```
<user-info/> // 内部获取用户信息并渲染
```

```
<shop-list/> // 内部获取商铺列表并渲染
</div>
```

然后各个子组件处理自己的逻辑

### 3. 组件的属性有序排列：从上到下编写，原生属性-自定义方法-绑定方法

比如

```
< user-info
  class="user-info"
  avatar="/assets/img/avatar.png"
  @click="onClickUserInfo" // 绑定方法以“on-事件-对象”的格式命名，与 VantUI 一致
/>
```

## CSS

1. 遵守 BEM 规范
2. 所有类名都需要挂上 "ec-erp"，这是外层乾坤架构要求的，目的是做样式隔离
3. 遵守 W3C 协会建议，样式从外到内编写，提高精准度、方便维护

比如

```
.user-info{
  position: absolute;
  top: 0;
  left: 0;
  margin: 20px;
  background: blue;
  font-size: 14px;
}
```

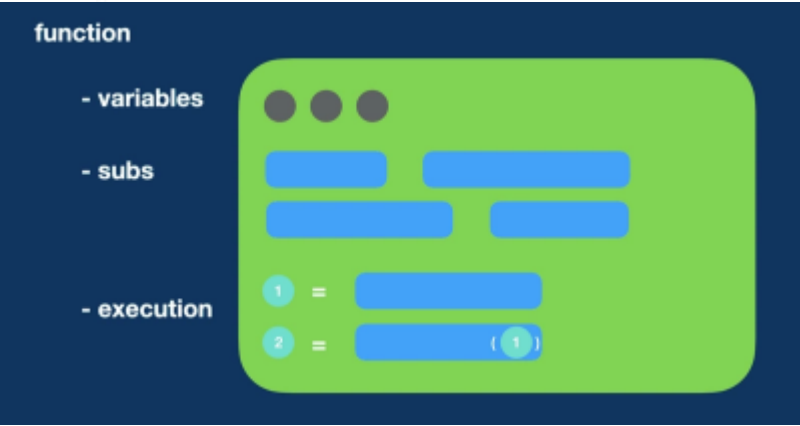
## JS

1. 遵守《CleanCode》：单一职责，精确命名，少输入，单一的输出

比如对一个数据列表的一些属性进行加工，那就写成

```
const list = itemsPropsHandle( dataList )
```

2. 菜单式编程 <<https://segmentfault.com/a/1190000038247789>>：遵循百度首席前端的写法，提升效率和可扩展性



2. 编写文件头部信息，难理解的地方一定要加注释



## 提交规范

1. 遵守《GitEmoji 规范》

- 🌟：新增代码、功能、特性
- 🐛：修复问题
- 💎：版本发布
- 📄：修改文档，比如 /doc 文档修改、README.md 改动
- 🏠：项目结构改动，比如公共文件夹移动、新增大量图片

2. 公式：Emoji + 【模块】 + 描述。比如“🌟：【首页】新增主动监听功能”。

建议加入到输入法，方便快速编写