**前端代码规范**

1. **命名规范**
2. 请确保文件命名总是以字母开头而不是数字。而以特殊字符开头命名的文件，一般都有特殊的含义与用处
3. 资源的字母名称必须全为小写，这是因为在某些对大小写字母敏感的操作系统中，当文件通过工具压缩混淆后，或者人为修改过后，大小写不同而导致引用文件不同的错误，很难被发现。
4. JS命名规范
5. JS函数命名请使用驼峰命名方式，如：clickButton
6. JS 类命名请使用 大写字母开头
7. Js不可改变的变量请使用常量命名
8. CSS 命名规范
9. css最好用class来命名，js用id来命名，已做区分

注意事项

* 规则命名中，css使用小写加中横线的方式，js使用驼峰命名方式
* 命名中尽量避免使用中文拼音，应该采用更简明有语义的英文单词进行组合
* 命名注意缩写，但是不能盲目缩写
* 不允许通过1、2、3等数字号进行命名
* 避免class与id重名
* id注意用于标识模块或页面的某一个父容器区域，名称必须唯一，不要随意新建id
* class用于标识某一个类型的对象，命名必须言简意赅
* 尽可能提高代码模块的复用，样式尽量用组合的方式
* 规则名称中不应该包含颜色、定位等与具体显示效果相关的信息，应该用意义命名，而不是结果名称

1. **协议**

不要指定引入资源所带的具体协议，当引入图片或其他媒体文件，还有样式和脚本时，URLs 所指向的具体路径，不要指定协议部分（http:, https:），除非这两者协议都不可用。

不指定协议使得 URL 从绝对的获取路径转变为相对的，在请求资源协议无法确定时非常好用，而且还能为文件大小节省几个字节。

1. **文本缩进**

使用4空格，tab 键进行缩进

1. **项目规范**

除 公共组件之外，所有的vue 都应当对应一个js文件和css文件。如果css内容较少方可写在vue里面。Css 的请尽量使用less 来写。公共组件请挂载在vue上面。避免重复使用

所有的回跳页面，若无参数，禁止使用 view.$fire, 请使用view.$emit(‘back’ )

相同的页面样式，应使用公共组件来写，避免太多重复代码

页面操作尽量使用vue的api去操作dom，不得以使用js操作dom，则需使用js原生操作，不可引大型的jquery大型的操作dom操作库件。

1. **其它规范 （ 包含性能优化 ）**

* 合并margin、padding、border的-top/-right/-bottom/-left的设置，尽量使用短名称
* 选择器应该在满足功能的基础上尽量简短，减少选择器嵌套，查询消耗。但是一定要避免覆盖全局样式设置
* 禁止在css中使用\*选择符
* 0后面不需要单独，比如0px可以省略成0，0.8px可以省略成.8px
* 如果没有边框时，不要写成border:0;应该写成border:none
* 在保存代码解耦的前提下，尽量合并重复的样式
* background、font等可以缩写的属性，尽量使用缩写形式
* 能以背景形式呈现的图片，尽量都写入CSS样式中
* Js中公共的函数，后面可重复使用的请放在公共js里面。
* Js结果运算请尽可能的使用三元运算，避免使用if 实现。
* Js中的运算请尽量使用 **简易** 逻辑实现，避免复杂逻辑创建垃圾代码。
* 不要轻易改动全站级CSS和通用CSS库，改动后，要经过全面测试
* 避免过小的背景图片平铺
* 绝对不要在CSS中使用"\*"选择符
* 层级(z-index)必须清晰明确，页面弹窗、气泡为最高级(最高级为999)，不同弹窗气泡之间可在三位数之间调整，普通区块为10-90内10的倍数；区块展开、弹出为当前父层级上个位增加，禁止层级间盲目攀比
* 背景图片在情况允许，尽可能使用sprite技术，减小http请求，考虑到多人协作开发，sprite按照模块、业务、页面来划分
* 页面内部尽量避免使用style属性，CSS放在head标签中，由link标签引入，使页面的结构与表现分离
* 尽量减少使用float、position等影响性能的属性，这样可以避免新手在布局时出现的混乱
* 尽量少使用<br />来断行
* 不要连续出现多个 (空格)，也尽量少使用全角空格（英文字符集下，全角空格会变成乱码），空白应该尽量使用text-indent、maring/padding等方法来实现
* 排版如果遇到需要首行缩进的处理，可以使用text-indent:2em;
* 图片如果需要加载就在页面上用img标签写出，重要的图片必须加上alt属性，给重要的元素和截断的元素上加上title
* 分清楚什么情况下jpg/gif/png图片