# 一、命名规范

* 组件命名：公共组件components文件下使用大驼峰式命名（PascalCase），Views下业务组件使用小驼峰式命名（pascalCase）并确保其简洁且易于理解。
* class命名：单词间使用-分隔符分开，小写，例（app-container）。
* 接口文件命名：一个view页面文件对应一个api文件，命名保持一致，页面复杂接口多的可以拆分多个文件，公共使用的api放common.js文件中。
* api命名：使用 camelCase 命名，小驼峰。
* Vue页面命名：页面目录使用 camelCase 命名，小驼峰。
* Hooks使用use+业务单词命名，例（useUser.js）。
* 方法名、参数名、成员变量、局部变量都统一使用 lowerCamelCase 风格，必须遵从驼峰形式。

# 二、vue模版规范

* 尽量使用计算属性computed而不是在模板中写过多的逻辑。
* 使用v-for时指定key，并使用唯一的值。
* 使用refs引用DOM元素或子组件实例，避免直接操作DOM。
* 若需要一个基本类型的响应式数据，必须使用ref。若需要一个响应式对象，层级不深，ref、reactive都可以。需要一个响应式对象，且层级较深，推荐使用reactive。
* 禁止使用var定义变量，使用let定义变量，const定义常量。
* 禁止v-if 和 v-for 放在同一个元素上。Vue3中: 当它们同时存在于一个节点上时，v-if 比 v-for 的优先级更高。
* 指令缩写：

用 : 表示 v-bind

用 @ 表示 v-on

用 # 表示 v-slot

* script标签内部加上name属性, 用于定义组件名称，使用大驼峰。

# 编码风格

* 对关键代码段进行注释，‌以提高代码的可读性和可维护性。
* 字段文件等命名要语义化，禁止使用a,b,c,d；name1,name2,name3等命名。
* 缩进换行请使用两个空格。
* js代码结尾使用分号。
* 代码块之间不允许多个空行。
* 禁止定义未使用的变量和空函数。
* 命名函数使用const定义箭头函数。例：const function =()=>{}。
* 上线应去掉console.log，谨慎使用console.log，‌以避免性能问题。‌

‌