前端编码规范

**命名规范**

\* 文件夹名采用大驼峰

\* Html文件名采用全小写，使用 \_ 分割

\* JS文件名采用大驼峰

\* CSS文件名采用全小写，使用 \_ 分割

\* 类名首字母大写

\* 变量、属性和方法名首字母采用小驼峰

\* 常量所有字母大写，两个词之间采用 \_ 分割

\* 临时变量或循环计数变量单个小写字母( 例如SaleInfo使用s来表示)

\* 命名使用英文含义单词或缩写

\* Html中的ID、Class、Tag等均采用全小写字母，两词之间使用 - 分割

**使用规范**

\* 变量必须先声明再使用

\* 变量类型固定，运行时不改变变量类型

\* 类的私有属性或方法有一定区分( 例如：加入 \_ 前缀 )

\* 尽可能避免使用多维大数组，即使存在也要第一时间拆分

\* 尽可能减少全局变量、使用有意义的统一全局命名空间

**编码风格**

\* 单行长度小于等于120个字符

\* 缩进使用2个空格、4个空格或1个tab

\* 代码行中保留适当的空格，不同功能代码之前保留适当的空行

\* 代码循环或回调嵌套层次避免超过4层

\* 类不要超过1000行，方法不要超过200行

\* 函数参数小于等于3个，多于3个参数使用对象传递

\* 功能逻辑或模块的内部代码使用闭包保护，减少冲突可能

\* 增加必要的注释，包括文件头部的文件功能说明，变量定义说明，函数方法说明

\* 变量定义说明在右侧；函数方法说明在头部 ( 包括用途、参数和返回值说明)；条件语句说明在判断内部

**编码逻辑**

\* 尽可能使用异步替代同步 ( 例如：ajax异步，js加载异步)

\* API设计风格统一，且均包含错误回调或异常抛出

\* 避免直接给dom对象附加属性或方法

\* 避免逻辑代码重复，相同功能代码出现次数大于等于3次时，应抽象为函数或类

\* 代码执行阻塞时间不超过500ms

**缓存和存储**

\* 合理使用浏览器userData、Cookie、localStorage、WebSQL、IndexDB等存储，对于每个存储的内容都有长度控制

\* 内存缓存及存储缓存都有使用空间限定，且都有过期清理机制，原则上每次初始化时需清空所有缓存

\* 合理使用http请求缓存，静态资源后不带动态参数 ( 例如：时间戳 )

**日志和监控**

\* 关键逻辑、关键数据、外部交互点和页面性能必须有监控埋点 ，并记录埋点Key

\* 适当的使用Console输出信息，避免污染控制台信息，输出信息能明确表达其含义

\* 合理回传必要的日志

\* 针对初始化、外部交互 ( 数据、Native、RN与Hybrid )、输出结果和页面性能的埋点建立定时Job

**环境和兼容性**

\* 代码中尽可能不要出现判断公司内部环境的逻辑，使用编译系统的ConfigGen等替代

\* 根据程序的具体使用场景，代码内必须做兼容处理，并需将相关代码摘出模块化处理

\* 避免使用userAgent来判断兼容性，尽可能判断属性或方法有效性来处理兼容

**技术选型**

\* 尽可能使用公司标准，避免重复造轮子

\* 如果该组件或方案公司已经提供，请采用公司方案

\* 如果公司方案不能满足自己的需求，可提需求改进

\* 如果提需求后仍然未改进，则做升级处理，最终成果反馈给公司

\* 如果公司没有现成方案，但是有开源方案，应该采用开源方案

**其他**

\* 注释需有针对性，提高代码可读性为主，注释为辅

\* 说明文档使用浅显易懂的文字表达，配以流程图或示意图

\* 每次HotFix需注明修复日期及问题原因(可采用CP4 或 Conf 链接)