

JavaScript 命名规范

1) 采用小写驼峰命名 **lowerCamelCase**，代码中的命名均不能以下划线，也不能以下划线或美元符号结束

反例： `_name` / `name_` / `name$`

2) 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。

正例： `localValue` / `getHttpMessage()` / `inputUserId`

****其中 method 方法命名必须是 动词 或者 动词+名词 形式****

正例： `saveShopCarData` / `openShopCarInfoDialog`

反例： `save` / `open` / `show` / `go`

****特此说明，增删查改，详情统一使用如下 5 个单词，不得使用其他（目的是为了统一各个端）****

`add` / `update` / `delete` / `detail` / `get`

附： 函数方法常用的动词:

```
get 获取/set 设置,  
add 增加/remove 删除  
create 创建/destory 移除  
start 启动/stop 停止  
open 打开/close 关闭,  
read 读取/write 写入  
load 载入/save 保存,  
create 创建/destroy 销毁  
begin 开始/end 结束,  
backup 备份/restore 恢复  
import 导入/export 导出,  
split 分割/merge 合并  
inject 注入/extract 提取,  
attach 附着/detach 脱离  
bind 绑定/separate 分离,  
view 查看/browse 浏览  
edit 编辑/modify 修改,  
select 选取/mark 标记  
copy 复制/paste 粘贴,  
undo 撤销/redo 重做  
insert 插入/delete 移除,  
add 加入/append 添加  
clean 清理/clear 清除,  
index 索引/sort 排序  
find 查找/search 搜索,  
increase 增加/decrease 减少  
play 播放/pause 暂停,  
launch 启动/run 运行  
compile 编译/execute 执行,  
debug 调试/trace 跟踪
```

observe 观察/listen 监听,
build 构建/publish 发布
input 输入/output 输出,
encode 编码/decode 解码
encrypt 加密/decrypt 解密,
compress 压缩/decompress 解压缩
pack 打包/unpack 解包,
parse 解析/emit 生成
connect 连接/disconnect 断开,
send 发送/receive 接收
download 下载/upload 上传,
refresh 刷新/synchronize 同步
update 更新/revert 复原,
lock 锁定/unlock 解锁
check out 签出/check in 签入,
submit 提交/commit 交付
push 推/pull 拉,
expand 展开/collapse 折叠
begin 起始/end 结束,
start 开始/finish 完成
enter 进入/exit 退出,
abort 放弃/quit 离开
obsolete 废弃/depreciate 废旧,
collect 收集/aggregate 聚集

3) 常量命名全部大写，单词间用下划线隔开，力求语义表达完整清楚，不要嫌名字长。

正例： `MAX_STOCK_COUNT`

反例： `MAX_COUNT`