**赵骰-回雪自定义插件帮助拓展文档**

"Zhao Dice"赵骰非官方插件交流群

2022年3月25日版

**目录**

[**赵骰-回雪自定义插件帮助拓展文档**](file:///D:\users\download\%E8%B5%B5%E9%AA%B0%E6%8F%92%E4%BB%B6%E9%80%9F%E6%88%90.%E4%BB%8E%E5%85%A5%E9%97%A8%E5%88%B0%E5%85%A5%E5%9C%9F(2022-3-26).html#赵骰-回雪自定义插件帮助拓展文档)[目录](file:///D:\users\download\%E8%B5%B5%E9%AA%B0%E6%8F%92%E4%BB%B6%E9%80%9F%E6%88%90.%E4%BB%8E%E5%85%A5%E9%97%A8%E5%88%B0%E5%85%A5%E5%9C%9F(2022-3-26).html#目录)[**零 碎 前 言**](file:///D:\users\download\%E8%B5%B5%E9%AA%B0%E6%8F%92%E4%BB%B6%E9%80%9F%E6%88%90.%E4%BB%8E%E5%85%A5%E9%97%A8%E5%88%B0%E5%85%A5%E5%9C%9F(2022-3-26).html#零-碎-前-言)[**0 . 准 备 工 作**](file:///D:\users\download\%E8%B5%B5%E9%AA%B0%E6%8F%92%E4%BB%B6%E9%80%9F%E6%88%90.%E4%BB%8E%E5%85%A5%E9%97%A8%E5%88%B0%E5%85%A5%E5%9C%9F(2022-3-26).html#0--准-备-工-作)[**1 . 输 入**](file:///D:\users\download\%E8%B5%B5%E9%AA%B0%E6%8F%92%E4%BB%B6%E9%80%9F%E6%88%90.%E4%BB%8E%E5%85%A5%E9%97%A8%E5%88%B0%E5%85%A5%E5%9C%9F(2022-3-26).html#1--输-入)[**1.1 通用处理**](file:///D:\users\download\%E8%B5%B5%E9%AA%B0%E6%8F%92%E4%BB%B6%E9%80%9F%E6%88%90.%E4%BB%8E%E5%85%A5%E9%97%A8%E5%88%B0%E5%85%A5%E5%9C%9F(2022-3-26).html#11-通用处理)[**1.2 Mirai码**](file:///D:\users\download\%E8%B5%B5%E9%AA%B0%E6%8F%92%E4%BB%B6%E9%80%9F%E6%88%90.%E4%BB%8E%E5%85%A5%E9%97%A8%E5%88%B0%E5%85%A5%E5%9C%9F(2022-3-26).html#12-mirai码)[**2 . 内 容 块**](file:///D:\users\download\%E8%B5%B5%E9%AA%B0%E6%8F%92%E4%BB%B6%E9%80%9F%E6%88%90.%E4%BB%8E%E5%85%A5%E9%97%A8%E5%88%B0%E5%85%A5%E5%9C%9F(2022-3-26).html#2--内-容-块)[**2.1 [[auto]]**](file:///D:\users\download\%E8%B5%B5%E9%AA%B0%E6%8F%92%E4%BB%B6%E9%80%9F%E6%88%90.%E4%BB%8E%E5%85%A5%E9%97%A8%E5%88%B0%E5%85%A5%E5%9C%9F(2022-3-26).html#21-auto)[**2.1.1 匹配词**](file:///D:\users\download\%E8%B5%B5%E9%AA%B0%E6%8F%92%E4%BB%B6%E9%80%9F%E6%88%90.%E4%BB%8E%E5%85%A5%E9%97%A8%E5%88%B0%E5%85%A5%E5%9C%9F(2022-3-26).html#211-匹配词)[**2.1.1.1 全词匹配**](file:///D:\users\download\%E8%B5%B5%E9%AA%B0%E6%8F%92%E4%BB%B6%E9%80%9F%E6%88%90.%E4%BB%8E%E5%85%A5%E9%97%A8%E5%88%B0%E5%85%A5%E5%9C%9F(2022-3-26).html#2111-全词匹配)[**2.1.1.2 模糊词匹配**](file:///D:\users\download\%E8%B5%B5%E9%AA%B0%E6%8F%92%E4%BB%B6%E9%80%9F%E6%88%90.%E4%BB%8E%E5%85%A5%E9%97%A8%E5%88%B0%E5%85%A5%E5%9C%9F(2022-3-26).html#2112-模糊词匹配)[**2.1.1.3 正则匹配**](file:///D:\users\download\%E8%B5%B5%E9%AA%B0%E6%8F%92%E4%BB%B6%E9%80%9F%E6%88%90.%E4%BB%8E%E5%85%A5%E9%97%A8%E5%88%B0%E5%85%A5%E5%9C%9F(2022-3-26).html#2113-正则匹配)[**2.1.1.4 多匹配词、多匹配模式**](file:///D:\users\download\%E8%B5%B5%E9%AA%B0%E6%8F%92%E4%BB%B6%E9%80%9F%E6%88%90.%E4%BB%8E%E5%85%A5%E9%97%A8%E5%88%B0%E5%85%A5%E5%9C%9F(2022-3-26).html#2114-多匹配词多匹配模式)[**2.1.2 执行内容**](file:///D:\users\download\%E8%B5%B5%E9%AA%B0%E6%8F%92%E4%BB%B6%E9%80%9F%E6%88%90.%E4%BB%8E%E5%85%A5%E9%97%A8%E5%88%B0%E5%85%A5%E5%9C%9F(2022-3-26).html#212-执行内容)[**2.1.3 其他项目**](file:///D:\users\download\%E8%B5%B5%E9%AA%B0%E6%8F%92%E4%BB%B6%E9%80%9F%E6%88%90.%E4%BB%8E%E5%85%A5%E9%97%A8%E5%88%B0%E5%85%A5%E5%9C%9F(2022-3-26).html#213-其他项目)[**2.2 [[define]]**](file:///D:\users\download\%E8%B5%B5%E9%AA%B0%E6%8F%92%E4%BB%B6%E9%80%9F%E6%88%90.%E4%BB%8E%E5%85%A5%E9%97%A8%E5%88%B0%E5%85%A5%E5%9C%9F(2022-3-26).html#22-define)[**2.3 [[timer]]**](file:///D:\users\download\%E8%B5%B5%E9%AA%B0%E6%8F%92%E4%BB%B6%E9%80%9F%E6%88%90.%E4%BB%8E%E5%85%A5%E9%97%A8%E5%88%B0%E5%85%A5%E5%9C%9F(2022-3-26).html#23-timer)[**时间调度器**](file:///D:\users\download\%E8%B5%B5%E9%AA%B0%E6%8F%92%E4%BB%B6%E9%80%9F%E6%88%90.%E4%BB%8E%E5%85%A5%E9%97%A8%E5%88%B0%E5%85%A5%E5%9C%9F(2022-3-26).html#时间调度器)[**2.4 [[lua]]**](file:///D:\users\download\%E8%B5%B5%E9%AA%B0%E6%8F%92%E4%BB%B6%E9%80%9F%E6%88%90.%E4%BB%8E%E5%85%A5%E9%97%A8%E5%88%B0%E5%85%A5%E5%9C%9F(2022-3-26).html#24-lua)[**2.4.1 table声明**](file:///D:\users\download\%E8%B5%B5%E9%AA%B0%E6%8F%92%E4%BB%B6%E9%80%9F%E6%88%90.%E4%BB%8E%E5%85%A5%E9%97%A8%E5%88%B0%E5%85%A5%E5%9C%9F(2022-3-26).html#241-table声明)[**2.4.2 函数定义**](file:///D:\users\download\%E8%B5%B5%E9%AA%B0%E6%8F%92%E4%BB%B6%E9%80%9F%E6%88%90.%E4%BB%8E%E5%85%A5%E9%97%A8%E5%88%B0%E5%85%A5%E5%9C%9F(2022-3-26).html#242-函数定义)[**2.4.3 指令定义**](file:///D:\users\download\%E8%B5%B5%E9%AA%B0%E6%8F%92%E4%BB%B6%E9%80%9F%E6%88%90.%E4%BB%8E%E5%85%A5%E9%97%A8%E5%88%B0%E5%85%A5%E5%9C%9F(2022-3-26).html#243-指令定义)[**2.4.3.1 msg内容**](file:///D:\users\download\%E8%B5%B5%E9%AA%B0%E6%8F%92%E4%BB%B6%E9%80%9F%E6%88%90.%E4%BB%8E%E5%85%A5%E9%97%A8%E5%88%B0%E5%85%A5%E5%9C%9F(2022-3-26).html#2431-msg内容)[**2.4.3.2 commands**](file:///D:\users\download\%E8%B5%B5%E9%AA%B0%E6%8F%92%E4%BB%B6%E9%80%9F%E6%88%90.%E4%BB%8E%E5%85%A5%E9%97%A8%E5%88%B0%E5%85%A5%E5%9C%9F(2022-3-26).html#2432-commands)[**2.4.3.3 msg\_order**](file:///D:\users\download\%E8%B5%B5%E9%AA%B0%E6%8F%92%E4%BB%B6%E9%80%9F%E6%88%90.%E4%BB%8E%E5%85%A5%E9%97%A8%E5%88%B0%E5%85%A5%E5%9C%9F(2022-3-26).html#2433-msgorder)[**2.4.3.4 replaces**](file:///D:\users\download\%E8%B5%B5%E9%AA%B0%E6%8F%92%E4%BB%B6%E9%80%9F%E6%88%90.%E4%BB%8E%E5%85%A5%E9%97%A8%E5%88%B0%E5%85%A5%E5%9C%9F(2022-3-26).html#2434-replaces)[**2.4.3.5 优先级**](file:///D:\users\download\%E8%B5%B5%E9%AA%B0%E6%8F%92%E4%BB%B6%E9%80%9F%E6%88%90.%E4%BB%8E%E5%85%A5%E9%97%A8%E5%88%B0%E5%85%A5%E5%9C%9F(2022-3-26).html#2435-优先级)[3 . 基 本 代 码](file:///D:\users\download\%E8%B5%B5%E9%AA%B0%E6%8F%92%E4%BB%B6%E9%80%9F%E6%88%90.%E4%BB%8E%E5%85%A5%E9%97%A8%E5%88%B0%E5%85%A5%E5%9C%9F(2022-3-26).html#3--基-本-代-码)[**3.1 【一天】**](file:///D:\users\download\%E8%B5%B5%E9%AA%B0%E6%8F%92%E4%BB%B6%E9%80%9F%E6%88%90.%E4%BB%8E%E5%85%A5%E9%97%A8%E5%88%B0%E5%85%A5%E5%9C%9F(2022-3-26).html#31-一天)[**3.2 【块】**](file:///D:\users\download\%E8%B5%B5%E9%AA%B0%E6%8F%92%E4%BB%B6%E9%80%9F%E6%88%90.%E4%BB%8E%E5%85%A5%E9%97%A8%E5%88%B0%E5%85%A5%E5%9C%9F(2022-3-26).html#32-块)[3.3 【计算】](file:///D:\users\download\%E8%B5%B5%E9%AA%B0%E6%8F%92%E4%BB%B6%E9%80%9F%E6%88%90.%E4%BB%8E%E5%85%A5%E9%97%A8%E5%88%B0%E5%85%A5%E5%9C%9F(2022-3-26).html#33--计算)[3.4 【变量】](file:///D:\users\download\%E8%B5%B5%E9%AA%B0%E6%8F%92%E4%BB%B6%E9%80%9F%E6%88%90.%E4%BB%8E%E5%85%A5%E9%97%A8%E5%88%B0%E5%85%A5%E5%9C%9F(2022-3-26).html#34--变量)[3.5 【赋值变量】](file:///D:\users\download\%E8%B5%B5%E9%AA%B0%E6%8F%92%E4%BB%B6%E9%80%9F%E6%88%90.%E4%BB%8E%E5%85%A5%E9%97%A8%E5%88%B0%E5%85%A5%E5%9C%9F(2022-3-26).html#35--赋值变量)[3.6 【常量】](file:///D:\users\download\%E8%B5%B5%E9%AA%B0%E6%8F%92%E4%BB%B6%E9%80%9F%E6%88%90.%E4%BB%8E%E5%85%A5%E9%97%A8%E5%88%B0%E5%85%A5%E5%9C%9F(2022-3-26).html#36--常量)[3.7 【换行】](file:///D:\users\download\%E8%B5%B5%E9%AA%B0%E6%8F%92%E4%BB%B6%E9%80%9F%E6%88%90.%E4%BB%8E%E5%85%A5%E9%97%A8%E5%88%B0%E5%85%A5%E5%9C%9F(2022-3-26).html#37--换行)[3.8 【随取】](file:///D:\users\download\%E8%B5%B5%E9%AA%B0%E6%8F%92%E4%BB%B6%E9%80%9F%E6%88%90.%E4%BB%8E%E5%85%A5%E9%97%A8%E5%88%B0%E5%85%A5%E5%9C%9F(2022-3-26).html#38--随取)[3.9 【比较】](file:///D:\users\download\%E8%B5%B5%E9%AA%B0%E6%8F%92%E4%BB%B6%E9%80%9F%E6%88%90.%E4%BB%8E%E5%85%A5%E9%97%A8%E5%88%B0%E5%85%A5%E5%9C%9F(2022-3-26).html#39--比较)[3.10【判断】](file:///D:\users\download\%E8%B5%B5%E9%AA%B0%E6%8F%92%E4%BB%B6%E9%80%9F%E6%88%90.%E4%BB%8E%E5%85%A5%E9%97%A8%E5%88%B0%E5%85%A5%E5%9C%9F(2022-3-26).html#310判断)[3.11【判空】](file:///D:\users\download\%E8%B5%B5%E9%AA%B0%E6%8F%92%E4%BB%B6%E9%80%9F%E6%88%90.%E4%BB%8E%E5%85%A5%E9%97%A8%E5%88%B0%E5%85%A5%E5%9C%9F(2022-3-26).html#311判空)[3.12【返回】](file:///D:\users\download\%E8%B5%B5%E9%AA%B0%E6%8F%92%E4%BB%B6%E9%80%9F%E6%88%90.%E4%BB%8E%E5%85%A5%E9%97%A8%E5%88%B0%E5%85%A5%E5%9C%9F(2022-3-26).html#312返回)[3.13【隐藏】](file:///D:\users\download\%E8%B5%B5%E9%AA%B0%E6%8F%92%E4%BB%B6%E9%80%9F%E6%88%90.%E4%BB%8E%E5%85%A5%E9%97%A8%E5%88%B0%E5%85%A5%E5%9C%9F(2022-3-26).html#313隐藏)[3.14【分割随取】](file:///D:\users\download\%E8%B5%B5%E9%AA%B0%E6%8F%92%E4%BB%B6%E9%80%9F%E6%88%90.%E4%BB%8E%E5%85%A5%E9%97%A8%E5%88%B0%E5%85%A5%E5%9C%9F(2022-3-26).html#314分割随取)[3.15【执行变量】](file:///D:\users\download\%E8%B5%B5%E9%AA%B0%E6%8F%92%E4%BB%B6%E9%80%9F%E6%88%90.%E4%BB%8E%E5%85%A5%E9%97%A8%E5%88%B0%E5%85%A5%E5%9C%9F(2022-3-26).html#315执行变量)[**3.16 【读项目】**](file:///D:\users\download\%E8%B5%B5%E9%AA%B0%E6%8F%92%E4%BB%B6%E9%80%9F%E6%88%90.%E4%BB%8E%E5%85%A5%E9%97%A8%E5%88%B0%E5%85%A5%E5%9C%9F(2022-3-26).html#316-读项目)[3.17【骰点计算】【详细骰点计算】](file:///D:\users\download\%E8%B5%B5%E9%AA%B0%E6%8F%92%E4%BB%B6%E9%80%9F%E6%88%90.%E4%BB%8E%E5%85%A5%E9%97%A8%E5%88%B0%E5%85%A5%E5%9C%9F(2022-3-26).html#317骰点计算详细骰点计算)[3.18【发送者QQ】](file:///D:\users\download\%E8%B5%B5%E9%AA%B0%E6%8F%92%E4%BB%B6%E9%80%9F%E6%88%90.%E4%BB%8E%E5%85%A5%E9%97%A8%E5%88%B0%E5%85%A5%E5%9C%9F(2022-3-26).html#318发送者qq)[3.19【随机数】](file:///D:\users\download\%E8%B5%B5%E9%AA%B0%E6%8F%92%E4%BB%B6%E9%80%9F%E6%88%90.%E4%BB%8E%E5%85%A5%E9%97%A8%E5%88%B0%E5%85%A5%E5%9C%9F(2022-3-26).html#319随机数)[3.20【当前群号】](file:///D:\users\download\%E8%B5%B5%E9%AA%B0%E6%8F%92%E4%BB%B6%E9%80%9F%E6%88%90.%E4%BB%8E%E5%85%A5%E9%97%A8%E5%88%B0%E5%85%A5%E5%9C%9F(2022-3-26).html#320当前群号)[3.21【现行日期】](file:///D:\users\download\%E8%B5%B5%E9%AA%B0%E6%8F%92%E4%BB%B6%E9%80%9F%E6%88%90.%E4%BB%8E%E5%85%A5%E9%97%A8%E5%88%B0%E5%85%A5%E5%9C%9F(2022-3-26).html#321现行日期)[3.22【执行脚本】](file:///D:\users\download\%E8%B5%B5%E9%AA%B0%E6%8F%92%E4%BB%B6%E9%80%9F%E6%88%90.%E4%BB%8E%E5%85%A5%E9%97%A8%E5%88%B0%E5%85%A5%E5%9C%9F(2022-3-26).html#322执行脚本)[3.23【10位时间戳】](file:///D:\users\download\%E8%B5%B5%E9%AA%B0%E6%8F%92%E4%BB%B6%E9%80%9F%E6%88%90.%E4%BB%8E%E5%85%A5%E9%97%A8%E5%88%B0%E5%85%A5%E5%9C%9F(2022-3-26).html#32310位时间戳)[3.24【设置固态变量】](file:///D:\users\download\%E8%B5%B5%E9%AA%B0%E6%8F%92%E4%BB%B6%E9%80%9F%E6%88%90.%E4%BB%8E%E5%85%A5%E9%97%A8%E5%88%B0%E5%85%A5%E5%9C%9F(2022-3-26).html#324设置固态变量)[3.25【设置变量针对个人】【设置变量针对群】](file:///D:\users\download\%E8%B5%B5%E9%AA%B0%E6%8F%92%E4%BB%B6%E9%80%9F%E6%88%90.%E4%BB%8E%E5%85%A5%E9%97%A8%E5%88%B0%E5%85%A5%E5%9C%9F(2022-3-26).html#325设置变量针对个人设置变量针对群)[3.26【设置变量有效期】](file:///D:\users\download\%E8%B5%B5%E9%AA%B0%E6%8F%92%E4%BB%B6%E9%80%9F%E6%88%90.%E4%BB%8E%E5%85%A5%E9%97%A8%E5%88%B0%E5%85%A5%E5%9C%9F(2022-3-26).html#326设置变量有效期)[3.27【正则匹配到的】](file:///D:\users\download\%E8%B5%B5%E9%AA%B0%E6%8F%92%E4%BB%B6%E9%80%9F%E6%88%90.%E4%BB%8E%E5%85%A5%E9%97%A8%E5%88%B0%E5%85%A5%E5%9C%9F(2022-3-26).html#327正则匹配到的)[**3.28 【到整数】**](file:///D:\users\download\%E8%B5%B5%E9%AA%B0%E6%8F%92%E4%BB%B6%E9%80%9F%E6%88%90.%E4%BB%8E%E5%85%A5%E9%97%A8%E5%88%B0%E5%85%A5%E5%9C%9F(2022-3-26).html#328-到整数)[**4 . 回 雪 编 写**](file:///D:\users\download\%E8%B5%B5%E9%AA%B0%E6%8F%92%E4%BB%B6%E9%80%9F%E6%88%90.%E4%BB%8E%E5%85%A5%E9%97%A8%E5%88%B0%E5%85%A5%E5%9C%9F(2022-3-26).html#4--回-雪-编-写)[**4.1 基本方法**](file:///D:\users\download\%E8%B5%B5%E9%AA%B0%E6%8F%92%E4%BB%B6%E9%80%9F%E6%88%90.%E4%BB%8E%E5%85%A5%E9%97%A8%E5%88%B0%E5%85%A5%E5%9C%9F(2022-3-26).html#41-基本方法)[**4.1.1 顺次**](file:///D:\users\download\%E8%B5%B5%E9%AA%B0%E6%8F%92%E4%BB%B6%E9%80%9F%E6%88%90.%E4%BB%8E%E5%85%A5%E9%97%A8%E5%88%B0%E5%85%A5%E5%9C%9F(2022-3-26).html#411-顺次)[**4.1.2 嵌套**](file:///D:\users\download\%E8%B5%B5%E9%AA%B0%E6%8F%92%E4%BB%B6%E9%80%9F%E6%88%90.%E4%BB%8E%E5%85%A5%E9%97%A8%E5%88%B0%E5%85%A5%E5%9C%9F(2022-3-26).html#412-嵌套)[**4.1.3 拼接**](file:///D:\users\download\%E8%B5%B5%E9%AA%B0%E6%8F%92%E4%BB%B6%E9%80%9F%E6%88%90.%E4%BB%8E%E5%85%A5%E9%97%A8%E5%88%B0%E5%85%A5%E5%9C%9F(2022-3-26).html#413-拼接)[**4.2 规范化插件**](file:///D:\users\download\%E8%B5%B5%E9%AA%B0%E6%8F%92%E4%BB%B6%E9%80%9F%E6%88%90.%E4%BB%8E%E5%85%A5%E9%97%A8%E5%88%B0%E5%85%A5%E5%9C%9F(2022-3-26).html#42-规范化插件)[**4.2.1 格式化**](file:///D:\users\download\%E8%B5%B5%E9%AA%B0%E6%8F%92%E4%BB%B6%E9%80%9F%E6%88%90.%E4%BB%8E%E5%85%A5%E9%97%A8%E5%88%B0%E5%85%A5%E5%9C%9F(2022-3-26).html#421-格式化)[**4.2.1.1 一次性输出**](file:///D:\users\download\%E8%B5%B5%E9%AA%B0%E6%8F%92%E4%BB%B6%E9%80%9F%E6%88%90.%E4%BB%8E%E5%85%A5%E9%97%A8%E5%88%B0%E5%85%A5%E5%9C%9F(2022-3-26).html#4211-一次性输出)[**4.2.1.2 输出整合**](file:///D:\users\download\%E8%B5%B5%E9%AA%B0%E6%8F%92%E4%BB%B6%E9%80%9F%E6%88%90.%E4%BB%8E%E5%85%A5%E9%97%A8%E5%88%B0%E5%85%A5%E5%9C%9F(2022-3-26).html#4212-输出整合)[**4.2.1.3 【】保护回车**](file:///D:\users\download\%E8%B5%B5%E9%AA%B0%E6%8F%92%E4%BB%B6%E9%80%9F%E6%88%90.%E4%BB%8E%E5%85%A5%E9%97%A8%E5%88%B0%E5%85%A5%E5%9C%9F(2022-3-26).html#4213-保护回车)[**4.2.2 模块化**](file:///D:\users\download\%E8%B5%B5%E9%AA%B0%E6%8F%92%E4%BB%B6%E9%80%9F%E6%88%90.%E4%BB%8E%E5%85%A5%E9%97%A8%E5%88%B0%E5%85%A5%E5%9C%9F(2022-3-26).html#422-模块化)[**4.3 反斜杠“\”**](file:///D:\users\download\%E8%B5%B5%E9%AA%B0%E6%8F%92%E4%BB%B6%E9%80%9F%E6%88%90.%E4%BB%8E%E5%85%A5%E9%97%A8%E5%88%B0%E5%85%A5%E5%9C%9F(2022-3-26).html#43-反斜杠)[**5 . 输 出**](file:///D:\users\download\%E8%B5%B5%E9%AA%B0%E6%8F%92%E4%BB%B6%E9%80%9F%E6%88%90.%E4%BB%8E%E5%85%A5%E9%97%A8%E5%88%B0%E5%85%A5%E5%9C%9F(2022-3-26).html#5--输-出)[**5.1 各类输出**](file:///D:\users\download\%E8%B5%B5%E9%AA%B0%E6%8F%92%E4%BB%B6%E9%80%9F%E6%88%90.%E4%BB%8E%E5%85%A5%E9%97%A8%E5%88%B0%E5%85%A5%E5%9C%9F(2022-3-26).html#51-各类输出)[**5.1.1 直接输出**](file:///D:\users\download\%E8%B5%B5%E9%AA%B0%E6%8F%92%E4%BB%B6%E9%80%9F%E6%88%90.%E4%BB%8E%E5%85%A5%E9%97%A8%E5%88%B0%E5%85%A5%E5%9C%9F(2022-3-26).html#511-直接输出)[**5.1.2 文件输出**](file:///D:\users\download\%E8%B5%B5%E9%AA%B0%E6%8F%92%E4%BB%B6%E9%80%9F%E6%88%90.%E4%BB%8E%E5%85%A5%E9%97%A8%E5%88%B0%E5%85%A5%E5%9C%9F(2022-3-26).html#512-文件输出)[**5.1.2.1 变量标签**](file:///D:\users\download\%E8%B5%B5%E9%AA%B0%E6%8F%92%E4%BB%B6%E9%80%9F%E6%88%90.%E4%BB%8E%E5%85%A5%E9%97%A8%E5%88%B0%E5%85%A5%E5%9C%9F(2022-3-26).html#5121-变量标签)[**5.1.2.2 变量有效期**](file:///D:\users\download\%E8%B5%B5%E9%AA%B0%E6%8F%92%E4%BB%B6%E9%80%9F%E6%88%90.%E4%BB%8E%E5%85%A5%E9%97%A8%E5%88%B0%E5%85%A5%E5%9C%9F(2022-3-26).html#5122-变量有效期)[**5.1.3 指定输出**](file:///D:\users\download\%E8%B5%B5%E9%AA%B0%E6%8F%92%E4%BB%B6%E9%80%9F%E6%88%90.%E4%BB%8E%E5%85%A5%E9%97%A8%E5%88%B0%E5%85%A5%E5%9C%9F(2022-3-26).html#513-指定输出)[**5.2 输出前的处理**](file:///D:\users\download\%E8%B5%B5%E9%AA%B0%E6%8F%92%E4%BB%B6%E9%80%9F%E6%88%90.%E4%BB%8E%E5%85%A5%E9%97%A8%E5%88%B0%E5%85%A5%E5%9C%9F(2022-3-26).html#52-输出前的处理)[**5.2.1特殊文本**](file:///D:\users\download\%E8%B5%B5%E9%AA%B0%E6%8F%92%E4%BB%B6%E9%80%9F%E6%88%90.%E4%BB%8E%E5%85%A5%E9%97%A8%E5%88%B0%E5%85%A5%E5%9C%9F(2022-3-26).html#521特殊文本)[**5.2.2 Mirai图片/at码**](file:///D:\users\download\%E8%B5%B5%E9%AA%B0%E6%8F%92%E4%BB%B6%E9%80%9F%E6%88%90.%E4%BB%8E%E5%85%A5%E9%97%A8%E5%88%B0%E5%85%A5%E5%9C%9F(2022-3-26).html#522-mirai图片at码)[**5.2.3 句首处理**](file:///D:\users\download\%E8%B5%B5%E9%AA%B0%E6%8F%92%E4%BB%B6%E9%80%9F%E6%88%90.%E4%BB%8E%E5%85%A5%E9%97%A8%E5%88%B0%E5%85%A5%E5%9C%9F(2022-3-26).html#523-句首处理)[6 . 插 件 调 试](file:///D:\users\download\%E8%B5%B5%E9%AA%B0%E6%8F%92%E4%BB%B6%E9%80%9F%E6%88%90.%E4%BB%8E%E5%85%A5%E9%97%A8%E5%88%B0%E5%85%A5%E5%9C%9F(2022-3-26).html#6--插-件-调-试)[**6.1 装载插件**](file:///D:\users\download\%E8%B5%B5%E9%AA%B0%E6%8F%92%E4%BB%B6%E9%80%9F%E6%88%90.%E4%BB%8E%E5%85%A5%E9%97%A8%E5%88%B0%E5%85%A5%E5%9C%9F(2022-3-26).html#61-装载插件)[**6.2 执行报错**](file:///D:\users\download\%E8%B5%B5%E9%AA%B0%E6%8F%92%E4%BB%B6%E9%80%9F%E6%88%90.%E4%BB%8E%E5%85%A5%E9%97%A8%E5%88%B0%E5%85%A5%E5%9C%9F(2022-3-26).html#62-执行报错)[**6.3 插件debug**](file:///D:\users\download\%E8%B5%B5%E9%AA%B0%E6%8F%92%E4%BB%B6%E9%80%9F%E6%88%90.%E4%BB%8E%E5%85%A5%E9%97%A8%E5%88%B0%E5%85%A5%E5%9C%9F(2022-3-26).html#63-插件debug)[7 . J a v a S c r i p t](file:///D:\users\download\%E8%B5%B5%E9%AA%B0%E6%8F%92%E4%BB%B6%E9%80%9F%E6%88%90.%E4%BB%8E%E5%85%A5%E9%97%A8%E5%88%B0%E5%85%A5%E5%9C%9F(2022-3-26).html#7--j-a-v-a-s-c-r-i-p-t)[7.1 参数传递](file:///D:\users\download\%E8%B5%B5%E9%AA%B0%E6%8F%92%E4%BB%B6%E9%80%9F%E6%88%90.%E4%BB%8E%E5%85%A5%E9%97%A8%E5%88%B0%E5%85%A5%E5%9C%9F(2022-3-26).html#71-参数传递)[7.2 JS输出](file:///D:\users\download\%E8%B5%B5%E9%AA%B0%E6%8F%92%E4%BB%B6%E9%80%9F%E6%88%90.%E4%BB%8E%E5%85%A5%E9%97%A8%E5%88%B0%E5%85%A5%E5%9C%9F(2022-3-26).html#72-js输出)[7.3 JS的Lib库](file:///D:\users\download\%E8%B5%B5%E9%AA%B0%E6%8F%92%E4%BB%B6%E9%80%9F%E6%88%90.%E4%BB%8E%E5%85%A5%E9%97%A8%E5%88%B0%E5%85%A5%E5%9C%9F(2022-3-26).html#73-js的lib库)[7.4 JS对象](file:///D:\users\download\%E8%B5%B5%E9%AA%B0%E6%8F%92%E4%BB%B6%E9%80%9F%E6%88%90.%E4%BB%8E%E5%85%A5%E9%97%A8%E5%88%B0%E5%85%A5%E5%9C%9F(2022-3-26).html#74-js对象)[7.5 JSON](file:///D:\users\download\%E8%B5%B5%E9%AA%B0%E6%8F%92%E4%BB%B6%E9%80%9F%E6%88%90.%E4%BB%8E%E5%85%A5%E9%97%A8%E5%88%B0%E5%85%A5%E5%9C%9F(2022-3-26).html#75-json)[7.6 JS进阶](file:///D:\users\download\%E8%B5%B5%E9%AA%B0%E6%8F%92%E4%BB%B6%E9%80%9F%E6%88%90.%E4%BB%8E%E5%85%A5%E9%97%A8%E5%88%B0%E5%85%A5%E5%9C%9F(2022-3-26).html#76-js进阶)[**8 .** **L** **u** **a**](file:///D:\users\download\%E8%B5%B5%E9%AA%B0%E6%8F%92%E4%BB%B6%E9%80%9F%E6%88%90.%E4%BB%8E%E5%85%A5%E9%97%A8%E5%88%B0%E5%85%A5%E5%9C%9F(2022-3-26).html#8--l-u-a)[8.1 基本格式](file:///D:\users\download\%E8%B5%B5%E9%AA%B0%E6%8F%92%E4%BB%B6%E9%80%9F%E6%88%90.%E4%BB%8E%E5%85%A5%E9%97%A8%E5%88%B0%E5%85%A5%E5%9C%9F(2022-3-26).html#81-基本格式)[8.2 插件示例](file:///D:\users\download\%E8%B5%B5%E9%AA%B0%E6%8F%92%E4%BB%B6%E9%80%9F%E6%88%90.%E4%BB%8E%E5%85%A5%E9%97%A8%E5%88%B0%E5%85%A5%E5%9C%9F(2022-3-26).html#82-插件示例)[8.2.1 【官方补丁】暗骰屏蔽](file:///D:\users\download\%E8%B5%B5%E9%AA%B0%E6%8F%92%E4%BB%B6%E9%80%9F%E6%88%90.%E4%BB%8E%E5%85%A5%E9%97%A8%E5%88%B0%E5%85%A5%E5%9C%9F(2022-3-26).html#821-官方补丁暗骰屏蔽)[8.2.2 （群文件）示范屏蔽某个赵骰功能的Lua](file:///D:\users\download\%E8%B5%B5%E9%AA%B0%E6%8F%92%E4%BB%B6%E9%80%9F%E6%88%90.%E4%BB%8E%E5%85%A5%E9%97%A8%E5%88%B0%E5%85%A5%E5%9C%9F(2022-3-26).html#822-群文件）示范屏蔽某个赵骰功能的lua)[8.2.3 【官方扩展】赵骰DND特化模块](file:///D:\users\download\%E8%B5%B5%E9%AA%B0%E6%8F%92%E4%BB%B6%E9%80%9F%E6%88%90.%E4%BB%8E%E5%85%A5%E9%97%A8%E5%88%B0%E5%85%A5%E5%9C%9F(2022-3-26).html#823-官方扩展赵骰dnd特化模块)[8.2.4 （群文件）luafile Lua进行的文件io](file:///D:\users\download\%E8%B5%B5%E9%AA%B0%E6%8F%92%E4%BB%B6%E9%80%9F%E6%88%90.%E4%BB%8E%E5%85%A5%E9%97%A8%E5%88%B0%E5%85%A5%E5%9C%9F(2022-3-26).html#824-群文件）luafile-lua进行的文件io)[**9** **.** **Mirai插件**](file:///D:\users\download\%E8%B5%B5%E9%AA%B0%E6%8F%92%E4%BB%B6%E9%80%9F%E6%88%90.%E4%BB%8E%E5%85%A5%E9%97%A8%E5%88%B0%E5%85%A5%E5%9C%9F(2022-3-26).html#9--mirai插件)[Mirai是什么？](file:///D:\users\download\%E8%B5%B5%E9%AA%B0%E6%8F%92%E4%BB%B6%E9%80%9F%E6%88%90.%E4%BB%8E%E5%85%A5%E9%97%A8%E5%88%B0%E5%85%A5%E5%9C%9F(2022-3-26).html#mirai是什么)[如何在赵骰里放一个Mirai插件](file:///D:\users\download\%E8%B5%B5%E9%AA%B0%E6%8F%92%E4%BB%B6%E9%80%9F%E6%88%90.%E4%BB%8E%E5%85%A5%E9%97%A8%E5%88%B0%E5%85%A5%E5%9C%9F(2022-3-26).html#如何在赵骰里放一个mirai插件)[**一些几乎是必备的插件！**](file:///D:\users\download\%E8%B5%B5%E9%AA%B0%E6%8F%92%E4%BB%B6%E9%80%9F%E6%88%90.%E4%BB%8E%E5%85%A5%E9%97%A8%E5%88%B0%E5%85%A5%E5%9C%9F(2022-3-26).html#一些几乎是必备的插件)[**如何加载控制台与调试Mirai插件**](file:///D:\users\download\%E8%B5%B5%E9%AA%B0%E6%8F%92%E4%BB%B6%E9%80%9F%E6%88%90.%E4%BB%8E%E5%85%A5%E9%97%A8%E5%88%B0%E5%85%A5%E5%9C%9F(2022-3-26).html#如何加载控制台与调试mirai插件)[如何给自己权限](file:///D:\users\download\%E8%B5%B5%E9%AA%B0%E6%8F%92%E4%BB%B6%E9%80%9F%E6%88%90.%E4%BB%8E%E5%85%A5%E9%97%A8%E5%88%B0%E5%85%A5%E5%9C%9F(2022-3-26).html#如何给自己权限)[**关于Mirai-Native插件的加载**](file:///D:\users\download\%E8%B5%B5%E9%AA%B0%E6%8F%92%E4%BB%B6%E9%80%9F%E6%88%90.%E4%BB%8E%E5%85%A5%E9%97%A8%E5%88%B0%E5%85%A5%E5%9C%9F(2022-3-26).html#关于mirai-native插件的加载)[**关于Lua Mirai**](file:///D:\users\download\%E8%B5%B5%E9%AA%B0%E6%8F%92%E4%BB%B6%E9%80%9F%E6%88%90.%E4%BB%8E%E5%85%A5%E9%97%A8%E5%88%B0%E5%85%A5%E5%9C%9F(2022-3-26).html#关于lua-mirai)[附 录 1 自 我 提 高](file:///D:\\users\\download\\%E8%B5%B5%E9%AA%B0%E6%8F%92%E4%BB%B6%E9%80%9F%E6%88%90.%E4%BB%8E%E5%85%A5%E9%97%A8%E5%88%B0%E5%85%A5%E5%9C%9F(2022-3-26).html" \l "附-录-1-作-业--雾-）-自-我-提-高)[试一试](file:///D:\users\download\%E8%B5%B5%E9%AA%B0%E6%8F%92%E4%BB%B6%E9%80%9F%E6%88%90.%E4%BB%8E%E5%85%A5%E9%97%A8%E5%88%B0%E5%85%A5%E5%9C%9F(2022-3-26).html#试一试)[1.关键词回复](file:///D:\users\download\%E8%B5%B5%E9%AA%B0%E6%8F%92%E4%BB%B6%E9%80%9F%E6%88%90.%E4%BB%8E%E5%85%A5%E9%97%A8%E5%88%B0%E5%85%A5%E5%9C%9F(2022-3-26).html#1关键词回复)[2.定时回复](file:///D:\users\download\%E8%B5%B5%E9%AA%B0%E6%8F%92%E4%BB%B6%E9%80%9F%E6%88%90.%E4%BB%8E%E5%85%A5%E9%97%A8%E5%88%B0%E5%85%A5%E5%9C%9F(2022-3-26).html#2定时回复)[3.关键词回复群开关（以关键词回复为基础）](file:///D:\users\download\%E8%B5%B5%E9%AA%B0%E6%8F%92%E4%BB%B6%E9%80%9F%E6%88%90.%E4%BB%8E%E5%85%A5%E9%97%A8%E5%88%B0%E5%85%A5%E5%9C%9F(2022-3-26).html#3关键词回复群开关以关键词回复为基础）)[4.快速调试](file:///D:\users\download\%E8%B5%B5%E9%AA%B0%E6%8F%92%E4%BB%B6%E9%80%9F%E6%88%90.%E4%BB%8E%E5%85%A5%E9%97%A8%E5%88%B0%E5%85%A5%E5%9C%9F(2022-3-26).html#4快速调试)[5.关键词监控](file:///D:\users\download\%E8%B5%B5%E9%AA%B0%E6%8F%92%E4%BB%B6%E9%80%9F%E6%88%90.%E4%BB%8E%E5%85%A5%E9%97%A8%E5%88%B0%E5%85%A5%E5%9C%9F(2022-3-26).html#5关键词监控) [6.漂流瓶](file:///D:\users\download\%E8%B5%B5%E9%AA%B0%E6%8F%92%E4%BB%B6%E9%80%9F%E6%88%90.%E4%BB%8E%E5%85%A5%E9%97%A8%E5%88%B0%E5%85%A5%E5%9C%9F(2022-3-26).html#6漂流瓶)[7.备忘录](file:///D:\users\download\%E8%B5%B5%E9%AA%B0%E6%8F%92%E4%BB%B6%E9%80%9F%E6%88%90.%E4%BB%8E%E5%85%A5%E9%97%A8%E5%88%B0%E5%85%A5%E5%9C%9F(2022-3-26).html#7备忘录)[8.好感度](file:///D:\users\download\%E8%B5%B5%E9%AA%B0%E6%8F%92%E4%BB%B6%E9%80%9F%E6%88%90.%E4%BB%8E%E5%85%A5%E9%97%A8%E5%88%B0%E5%85%A5%E5%9C%9F(2022-3-26).html#8好感度)[9.安科](file:///D:\users\download\%E8%B5%B5%E9%AA%B0%E6%8F%92%E4%BB%B6%E9%80%9F%E6%88%90.%E4%BB%8E%E5%85%A5%E9%97%A8%E5%88%B0%E5%85%A5%E5%9C%9F(2022-3-26).html#9安科) [10.广播](file:///D:\users\download\%E8%B5%B5%E9%AA%B0%E6%8F%92%E4%BB%B6%E9%80%9F%E6%88%90.%E4%BB%8E%E5%85%A5%E9%97%A8%E5%88%B0%E5%85%A5%E5%9C%9F(2022-3-26).html#10广播)[11.复读机](file:///D:\users\download\%E8%B5%B5%E9%AA%B0%E6%8F%92%E4%BB%B6%E9%80%9F%E6%88%90.%E4%BB%8E%E5%85%A5%E9%97%A8%E5%88%B0%E5%85%A5%E5%9C%9F(2022-3-26).html#11复读机)[12.反复读机](file:///D:\users\download\%E8%B5%B5%E9%AA%B0%E6%8F%92%E4%BB%B6%E9%80%9F%E6%88%90.%E4%BB%8E%E5%85%A5%E9%97%A8%E5%88%B0%E5%85%A5%E5%9C%9F(2022-3-26).html#12反复读机)[13.闹钟](file:///D:\users\download\%E8%B5%B5%E9%AA%B0%E6%8F%92%E4%BB%B6%E9%80%9F%E6%88%90.%E4%BB%8E%E5%85%A5%E9%97%A8%E5%88%B0%E5%85%A5%E5%9C%9F(2022-3-26).html#13闹钟) [14.资讯](file:///D:\users\download\%E8%B5%B5%E9%AA%B0%E6%8F%92%E4%BB%B6%E9%80%9F%E6%88%90.%E4%BB%8E%E5%85%A5%E9%97%A8%E5%88%B0%E5%85%A5%E5%9C%9F(2022-3-26).html#14资讯) [15.赵骰](file:///D:\users\download\%E8%B5%B5%E9%AA%B0%E6%8F%92%E4%BB%B6%E9%80%9F%E6%88%90.%E4%BB%8E%E5%85%A5%E9%97%A8%E5%88%B0%E5%85%A5%E5%9C%9F(2022-3-26).html#15赵骰)[附 录 2 若 干 插 件 代 码 技 巧](file:///D:\users\download\%E8%B5%B5%E9%AA%B0%E6%8F%92%E4%BB%B6%E9%80%9F%E6%88%90.%E4%BB%8E%E5%85%A5%E9%97%A8%E5%88%B0%E5%85%A5%E5%9C%9F(2022-3-26).html#附-录-2-若-干-插-件-代-码-技-巧)[**附2.1 回雪循环**](file:///D:\users\download\%E8%B5%B5%E9%AA%B0%E6%8F%92%E4%BB%B6%E9%80%9F%E6%88%90.%E4%BB%8E%E5%85%A5%E9%97%A8%E5%88%B0%E5%85%A5%E5%9C%9F(2022-3-26).html#附21-回雪循环)[**附2.2 常量“头文件”**](file:///D:\users\download\%E8%B5%B5%E9%AA%B0%E6%8F%92%E4%BB%B6%E9%80%9F%E6%88%90.%E4%BB%8E%E5%85%A5%E9%97%A8%E5%88%B0%E5%85%A5%E5%9C%9F(2022-3-26).html#附22-常量头文件)[**附2.3 回雪数组及对象**](file:///D:\users\download\%E8%B5%B5%E9%AA%B0%E6%8F%92%E4%BB%B6%E9%80%9F%E6%88%90.%E4%BB%8E%E5%85%A5%E9%97%A8%E5%88%B0%E5%85%A5%E5%9C%9F(2022-3-26).html#附23-回雪数组及对象)[**附2.4 回雪与JS间的数组传递**](file:///D:\users\download\%E8%B5%B5%E9%AA%B0%E6%8F%92%E4%BB%B6%E9%80%9F%E6%88%90.%E4%BB%8E%E5%85%A5%E9%97%A8%E5%88%B0%E5%85%A5%E5%9C%9F(2022-3-26).html#附24-回雪与js间的数组传递)[附 录 3 官 方 参 考 资 料](file:///D:\users\download\%E8%B5%B5%E9%AA%B0%E6%8F%92%E4%BB%B6%E9%80%9F%E6%88%90.%E4%BB%8E%E5%85%A5%E9%97%A8%E5%88%B0%E5%85%A5%E5%9C%9F(2022-3-26).html#附-录-3-官-方-参-考-资-料)[附 录 4 杂 物 、 资 料 与 工 具](file:///D:\users\download\%E8%B5%B5%E9%AA%B0%E6%8F%92%E4%BB%B6%E9%80%9F%E6%88%90.%E4%BB%8E%E5%85%A5%E9%97%A8%E5%88%B0%E5%85%A5%E5%9C%9F(2022-3-26).html#附-录-4-杂-物--资-料-与-工-具)[**编 写 者 们**](file:///D:\users\download\%E8%B5%B5%E9%AA%B0%E6%8F%92%E4%BB%B6%E9%80%9F%E6%88%90.%E4%BB%8E%E5%85%A5%E9%97%A8%E5%88%B0%E5%85%A5%E5%9C%9F(2022-3-26).html#编-写-者-们)[**鸣 谢 名 单**](file:///D:\users\download\%E8%B5%B5%E9%AA%B0%E6%8F%92%E4%BB%B6%E9%80%9F%E6%88%90.%E4%BB%8E%E5%85%A5%E9%97%A8%E5%88%B0%E5%85%A5%E5%9C%9F(2022-3-26).html#鸣-谢-名-单)

**零 碎 前 言**

回雪自定义，原名类铃心自定义。是赵骰开发者“赵喵喵V2.0”基于Java语言构建的中文简易编程工具，本文档教你如何编写一个赵骰插件，并从回雪进阶至js，Lua，乃至Mirai的插件。

在学习如何编写一个插件前，你应当先知道如何装载及使用一个赵骰插件。如果你不知道，那么你应当先去看搭骰教程。不知道如何使用插件就尝试着学习编写插件在我看来是一个本末倒置的行为。

本文档是由非官方赵骰插件爱好者学习交流互助群“Zhao Dice”(871504232)的各位群员参考官方文档并结合自身经验共同编撰而成的非官方教程。阅读过程中请注意实践及既有知识的巩固。如有错误或争议之处，可联系[275807554@qq.com](file:///D:\users\download\275807554@qq.com)。

本文档以Mirai框架赵骰为准，文档中的Mirai相关内容不适用于GOCQ-Http框架下的GDice版本赵骰。

文档共有九个章节，分别为准备工作、内容块、基本回雪代码、回雪格式、输出、插件调试、JavaScript、Lua、Mirai插件。另有四个附录，以及编写者名单、鸣谢名单。文档内容并不完全遵循赵骰插件学习、入门、进阶的过程，遇到不懂的内容可以先尝试跳过。

文档存在主观、不专业的理解内容，特别是我参与编撰的部分，请您自行评判。

在编写文档时，由于深受高考作文习惯的影响，我所参与编撰的部分很可能会出现措辞冗长、风格八股、官话套话的情况，敬请谅解。

请勿凭此文档拉踩、贬低官方的《类铃心帮助文档》及《代码档案》。《类铃心帮助文档》与《代码档案》是几乎所有群员入门回雪自定义时除成品插件外仅有的参考文件，本文档大量内容也是参考此二文档进行编写，其贡献与意义不可磨灭。

当您编写插件时，应当仔细审视您所编写的插件的功能特点。若您编写的插件存在明确或潜在的恶搞性、破坏性、骚扰性、违法违规或其他不宜、有风险、有争议的功能、特性或内容，例如群发、刷屏、循环发送、复读、周期发送、私信骚扰、私信轰炸、广告、宣传推广、长消息、辱骂、违规违法内容等，您应当谨慎地使用您的插件，并对它的传播严加管控。本文档及编写者不为任何人凭此文档编写的任何插件所造成的任何不良后果承担任何责任。

联的冮山

本文档以在线文档方式存在并持续更新，如有问题，进QQ群：871504232 反馈。

另外，你需要明确的一些关系：

1.我们使用的就是所谓的第三方客户端，一般被称为QQ机器人而不是骰子。

2.机器人（赵骰）的构成：

**--框架：**

种类繁多，包括但不限于CQ、ono、qr、myqq、nonebot、OlivOS、qtm……

* 关于这一点，你只需要知道酷Q（CoolQ）是老大哥，他死了，但没完全死——机器人标准OneBot就是在CQHTTP 的基础之上确定的，因此你会见到很多插件是从CQ迁移的，甚至有些框架和插件就是为了迁移CQ插件而出现的。
* 赵骰的框架

过去的框架是先驱，已经寄了。

赵骰目前的框架是 mirai，以及使用 mirai 以及 MiraiGo 开发的go-cqhttp（GDice）

**--插件：**

* 赵骰是插件，也可以是插件的插件（指QTM的插件虚拟机），但不是框架，不要用拿插件问题去找框架开发者
* 铃心是功能强大又知名的插件，但本质上与赵骰平级，不要用一个插件的问题去问另一个插件
* mirai的其他插件同理（ 铃心适配框架相当多，不一定是mirai插件 ）

**--赵骰的插件（回雪）**

* 支持的语言

回雪：也就是类铃心，【】的形式，类，也就是与铃心相似，但功能不及铃心。

js（JavaScript）：JavaScript语言，有专门的lib库，不要疑惑你的插件为什么在在线编辑器下报错

lua：Lua语言，同上，有SDK。

* 三种语言的变量名不互通。

NAO

**0 . 准 备 工 作**

NAO&弥

回雪自定义文件只支持UTF-8编码，文件后缀txt或者toml均可。事实上没有后缀也可以，内容正确即可。然而若因此故意找茬，加上zip, doc, jpg, mp3等后缀，出错实属自找。

新建一个UTF-8编码，后缀为txt或toml的文件。

手机建议用MT管理器，在赵骰官方群文件里可以找到。你将在文本编辑界面的右上角看到“UTF-8”字样。

电脑用自带的记事本即可，你会在记事本右下角看到“UTF-8”字样。//另外，非常建议您使用VS code、epk、notepad++等更高级的文本编辑器！

将以下几个代码块复制或抄进你刚刚新建的文件里。

注意：请将第4行、第12行与第24行的“userQQ”改为你的QQ号。

[[auto]]

keywordRegexp="^&([\\s\\S]\*)"

program="""

【判断 【发送者QQ】,userQQ,【块】,

【赋值变量 test,【正则匹配到的】】【执行变量 test】

&运行已完成。】

"""

[[auto]]

keywordRegexp="^js&([\\s\\S]\*)"

program="""

【判断 【发送者QQ】,userQQ,【块】,

【执行脚本 【正则匹配到的】】

js&运行已完成。】

"""

[[lua]]

version=1

name="lua console"

help="lua调试"

lua="""

msg\_order = {}

function debug(msg)

   if(msg.fromQQ == "userQQ")

   then

   local i = ZhaoDiceSDK.eval(string.gsub(msg.fromParams,"lua&","",1))

   return tostring(i) .. "\\nlua&运行已完成。"

   end

end

msg\_order["lua&"] = "debug"

"""

注意除了【】以外的符号都是英文半角符号。

现在你写成了你的第一个插件，它的功能是快速便捷地调试、演示代码，把这个插件装进你的骰子。

向你的骰子发送“&123”，如果你得到了

123

&运行已完成。

的回复，说明插件能够正常使用。

\*三个调试词是“&”、“js&”、“lua&”

\*调试具有潜在的危险性，JS调试可以使您的骰子暂时停摆（无上限），而Lua调试甚至可以使您的骰子被删除！

**1 . 输 入**

联的冮山

输入并不简单地指赵骰接收到的消息，而是接收到的消息经过处理后用于触发指令与插件的内容。输入对于几乎所有赵骰插件来说都是必须的，所有的关键词都由输入触发，而非直接由消息触发。因此，必须掌握消息与输入的不同，才能更好地编写插件。

骰子的控制台所显示的就是实际的输入内容。你可以一边阅读本章一边向骰子发送消息并查看控制台中的实际输入内容，增进理解。

**1.1 通用处理**

消息开头的空白符会被过滤。手机赵骰会删去句首的空格，电脑赵骰则删去空格与制表符（Tab）。句首换行不受影响。

信息中所有有效的@单人都会在输入中被转化为被@者的QQ号。例如“写教程@联的冮山”在控制台中即显示为“写教程275807554”。但是，如果被@者为骰子自身，且该@位于句首，那么该@会被直接截去，不会进入输入中。

一些特殊消息，如匿名聊天、双击头像、入场特效等消息不会被识别；厘米秀动作会被识别为“[动作消息]动作名称”；红包会被识别为“[QQ红包]请使用新版手机QQ查看红包。”；emoji不会被进行处理，而是直接进入输入内容；视频短片可能会被识别为“你的QQ暂不支持查看视频短片，请期待后续版本。”，也可能会被识别为空白；另外一些消息，比如群公告，会识别为“不支持的消息类型”。

QQ表情处理被分为三种情况：投入使用较早的表情，如微笑，会被识别为空白；投入使用较晚的表情，如摸鱼，会被识别为“/摸鱼”；一些单独发送会有特殊效果的表情（官称“超级表情”），在单独发送时会显示“[表情名称]请使用最新版手机QQ体验新功能”，在非单独发送时则参考前两种情况。

练习1.1：思考一下赵骰是怎么区分@自己与@其他人的。

**1.2 Mirai码**

你会发觉控制台里只有文字，而没有其他的任何东西，这是因为非纯文字消息未被识别，或被处理为文本内容。其中一些常用、常见的非纯文字消息，如图片表情、语音、回复、聊天记录、@全体成员，则会被转化为Mirai码的形式。

图片：[mirai:image:{图片码}.后缀名]

语音：[mirai:audio:音频码.后缀名]

回复：[mirai:quote:[ID1],[ID2]]

聊天记录：[mirai:origin:FORWARD,编码]ForwardMessage(preview=[消息预览],title=卡片第一行,brief=[聊天记录],sourse=聊天记录,summary=查看n条消息记录,nodeList=[Node(senderId=发送者QQ,time=时间戳,senderName=群昵称,messageChain=消息),Node(……),……])

@全体成员：[mirai:atall]

练习1.2：从控制台得到一个图片码，并利用调试词“&”让骰子把图片发送出来。（考虑到可能被tx屏蔽，建议私

**2 . 内 容 块**

联的冮山&竹林&弥

在接触回雪代码前，需要先明白赵骰插件的基本格式。

赵骰插件由内容块组成。这些内容块以[[auto]]、[[define]]、[[timer]]、[[lua]]开头。我们会进行一一讲解。

**2.1 [[auto]]**

[[auto]]是由关键词触发的内容块。它分为关键词、执行内容、其他项目三部分。

[[auto]]

keywordxxx="匹配词"

content/program="执行内容"

xxx="其他项目"

骰子每收到一条消息，便会将此消息与装载的所有插件中的所有[[auto]]内容块的关键词进行匹配，并执行匹配成功的内容，然后处理下一条消息。在同一个插件文件中，骰子按照从上向下的顺序与所有[[auto]]内容块中的关键词一一匹配并执行。如果同一条消息在同一插件文件中触发执行了多个有内容输出的[[auto]]内容块，则只会输出最先执行的[[auto]]内容块的输出内容。

**2.1.1 匹配词**

keyword是该内容块的关键词匹配方式，分为三种：keywordFull全词匹配、keywordContained模糊词匹配、keywordRegexp正则匹配。

**2.1.1.1 全词匹配**

全词匹配，即消息与关键词完全相同，不多不少，分毫不差，连符号的半角全角都不能有差别。在赵骰APP、软件直接设置的关键词回复就是全词匹配。

练习2.1.1.1：现在你可以把你在赵骰APP/程序的设置界面设置的关键词回复搬到插件里了。

**2.1.1.2 模糊词匹配**

模糊词匹配，即只要消息中完全包含了关键词，且是连续的、有序的，即视为匹配成功。

连续的：设模糊词为“**竹子**”，“后山长了**竹子**”能够匹配，而“**竹**林地上全是叶**子**”不能匹配。

有序的：设模糊词为“**蜜蜂**”，“一窝**蜜蜂**”能够匹配，而“优质**蜂蜜**”不能匹配。

**2.1.1.3 正则匹配**

正则匹配，使用正则表达式对消息进行匹配，并具有捕获匹配内容的功能。

至于正则表达式的学习与运用，请参考[正则表达式 – 语法 | 菜鸟教程](https://www.runoob.com/regexp/regexp-syntax.html)及其相关篇目内容。

需要提醒的是，不同编程语言的正则表达式语法存在微小差别。以及，网页中的正则表达式格式为

/正则表达式/修饰符

而回雪自定义的正则表达式格式为

keywordRegexp="(?修饰符)正则表达式"

这些都是应当注意的地方。

**2.1.1.4 多匹配词、多匹配模式**

一个[[auto]]内容块下可以设置多个匹配词，也可以设置多个匹配词模式。

--以设置多个全词匹配为例

keywordFull=["测试","调试","尝试"]    --keywordContained与keywordRegexp同样可行。

content="测试成功"

--多个匹配词模式

keywordFull="关键词测试"

keywordContain="测试插件"

keywordRegexp=[".\*插件$","^调试.\*"]

content="触发匹配词"

如果你有编程基础，你应该很容易就能看出这是一个字符串数组。

**2.1.2 执行内容**

执行内容有两种，program和content.

program会解析自己内容中的代码并执行，而content则直接将自己内容原封不动地输出。

除非没有回雪代码需要执行，一般都使用program。

与keyword一样，同一[[auto]]下可以设置多个content，效果为在多个content中随机发送一个。

同一[[auto]]下不能设置多个program.

练习2.1：优化一下骰子的回复词吧。

**2.1.3 其他项目**

--打开你的调试插件，在第一个[[auto]]下写入以下内容

blb="欢迎绑架赵赵"

zhao="【发送者QQ】"

--这是两个项目，你在学习第三章时可以用它调试

其他项目的形式皆为

自定义项目名="项目内容"

使用回雪【读项目】代码可以读取项目内容。项目内容是字符串形式，【读项目】并不会解析并执行其中可能存在的代码。这需要额外的步骤。

项目的有效范围仅限于其所处的[[auto]]内容块内。

**2.2 [[define]]**

[[define]]是常量，或者说“不可变量”。分为常量名和常量内容两部分。

[[define]]

name="常量名"

content="常量内容"

--例如，打开你的调试插件，写入以下内容。

[[define]]

name="def"

content="Hello world"

--这就是一个完整的常量，你在学习第三章时可以用它调试。

常量是固定的、只能通过直接修改插件来修改的“不可变量”。常量本身只有存储字符串内容的功能。它需要被[[auto]]或[[timer]]内容块使用【常量】代码调用。与【读项目】相同的是，【常量】并不会解析并执行常量中可能存在的代码。它们只是字符串。耍小聪明将常量的content改为program并不会起到作用，反而可能会导致程序出错——程序只会识别[[define]]内容块下的name和content.

在[[define]]内容块下写任何其他项目都是没有意义的。因为常量只会被其他内容块调用，常量中的代码是也只能是在调用常量的内容块中执行。【读项目】也同样是在调用常量的内容块中执行。

[[define]]的有效范围为全局——整个骰子。也就是说，常量可以跨插件调用。当然，你便需要考虑各插件间常量名冲突的潜在问题。

**2.3 [[timer]]**

[[timer]]依靠时间调度器，以精确到秒的时间设置触发后执行js脚本。也就是说[[timer]]无法使用回雪代码编写。分为名称、时间配置、js代码、其他项目四部分。

[[timer]]

name="调度器名称"

expression="时间调度器"

js="脚本代码"

**时间调度器**

时间配置的格式为“秒 分 时 日 月 周”。

例如设置每月23日16时33分整的格式为：

"0 33 16 23 \* ?"

设置每周一12点整的格式为：

"0 0 12 \* \* MON"

"0 0 12 \* \* 1"

设置以5秒为周期的格式为：

"0/5 \* \* \* \* ?"

如果你不明白，可以使用：[表达式生成器](https://cron.qqe2.com/)

练习2.3：让孩子跟你说声晚安。

**2.4 [[lua]]**

[[lua]]是赵骰在2021末、2022年初更新的，能够兼容执行Lua语言代码的内容块。[[lua]]内容块定义的指令优先级大于赵骰内置指令，但也比其他内容块复杂。本节讲解[[lua]]内容块中的基本构成。

至于具体的Lua代码编写，我们给出一个网页供自行学习：[Lua 教程 | 菜鸟教程](https://www.runoob.com/lua/lua-tutorial.html)

[[lua]]内容块由名称、注释、主体三部分构成，同时，主体由table声明、函数定义、指令定义三部分构成。

--注释

[[lua]]

name="名称"

help="注释"

lua="""

   --table声明

commands = {}

replaces = {}

msg\_order = {}

   --指令定义与函数编写

sdk = ZhaoDiceSDK

function function\_name(msg)

   return xxx

end

commands["command\_name"] = "function\_name"

replaces["command\_name"] = "function\_name"

msg\_order["keyword"] = "function\_name"

"""

名称、注释无需多言，重点在于主体。

**2.4.1 table声明**

table是表，一种数据类型，但我们不打算在这里详细讲述，我们简要讲明这三个表分别的作用。

commands是定义指令。即指令前缀（如果你没有改过，那就是“.” “。” “!” “！”）+自定义指令。

replaces是重定向指令，收到指令后将指令重定向再次处理。

msg\_order是头关键词匹配指令，无需指令前缀。

commands和replaces受.bot on/off控制，msg\_order受.reply on/off (this)控制。

当你需要其中的一项或多项时，必须先定义对应的table，用不到的则无需定义。

**2.4.2 函数定义**

function func\_name(argument)

   --code

   return xxx   -输出

end

这是Lua函数的基本格式，代码执行的场所。

func\_name是函数名。

argument是函数参数，多数为msg，即接受到的消息，具体包含发送者名称、发送者QQ、来自群聊/私聊、来源群号（如果有）、群昵称（如果为私聊，则与发送者名称一致）、消息内容、指令参数、指令参数是否为空等可以调用处理的数据。

code是具体代码，自行编写。

return是返回值，输出内容。

**2.4.3 指令定义**

学习Lua后你会发现指令定义其实是对table赋值的过程，但我们暂不讨论。

以由易到难的顺序讲述。

**2.4.3.1 msg内容**

当触发插件时,所传入的信息远远不止你发送的内容.

这是一个读取传入内容的插件:

[[lua]]

version=1

name="luamsg"

help="luamsg"

lua="""

commands = {}

local sdk = ZhaoDiceSDK

function luamsg(msg)

i = json.encode(msg)

j = i.."#{SPLIT}msg获取结束"

return j

end

commands["text"] = "luamsg"

"""

#要注意的是,msg仅作为形参存在,变量名取决于function时填入的参数名,因此

function luamsg(a)

i = json.encode(a)

j = i.."#{SPLIT}msg获取结束"

return j

end

#这样也是可以正常运行的

使用此插件可以获取msg内容,即:

//发送:.text Hello World可得

{

   "fromNick":"此处是触发者昵称",

   "fromGroup":"此处是触发群号",

   "isGroup":true,

   "regexStructure":{

       "middle":"Hello World"

  },

   "isEmpty":false,

   "fromQQ":"此处是触发者QQ",

   "fromGroupNick":"此处是触发者群昵称",

   "fromMsg":".text Hello World",

   "fromParams":"Hello World"

}

以上内容全都可以由Lua插件调用.

练习2.4.3.1：看看私聊的msg内容，与上面给出的有何异同。

**2.4.3.2 commands**

commands["指令"]="函数名"

这行代码意为定义一个指令“.指令 参数”，该指令会执行名为“函数名”的函数。

例如：

function hear(msg)

   return "QQ:" .. msg.fromQQ

end

commands["test"]="hear"

这段代码会新建一个“.test”指令，并且返回“QQ：<发送者QQ>”的消息。

commands可以覆盖赵骰原有指令的输出，但如果commands定义的函数没有输出，则原有指令会照常执行。

部分指令，如“.bot on/off”和“.reply on/off”指令，不能被覆盖。

练习2.4.3.2：写一个“.try”指令，它能返回触发关键词的群号。

**2.4.3.3 msg\_order**

msg\_order["关键词"]="函数名"

这行代码意为捕获以关键词开头的消息，相当于keywordRegexp="^(关键词[\s\S]\*)"，该指令会执行对应的函数。

例如：

function name(msg)

   return "You are " .. msg.fromGroupNick

end

msg\_order["me"]="name"

这段代码会匹配以“me”开头的消息（无需指令前缀），并回复“You are <发送者群昵称>”的消息。

练习2.4.3.3：仍然是me，让骰子回复“<发送者群昵称>发送了：<消息全文>”

**2.4.3.4 replaces**

replace["指令"]="函数名"

这行代码意为定义一个“.指令 参数”的指令，执行对应的函数，并将指令重定向至函数执行结果再次匹配指令，若为无效指令，则停止执行。

例如：

function pointer(msg)

   return "rd" .. msg.fromParams

end

replaces["roll"]="pointer"

这段代码会新建一个“.roll”指令，并将其重定向至“rdxx”。通俗地讲，你发送指令“.roll 20”相当于你发送指令“.rd20”。

与commands相同，replaces可能覆盖赵骰原有指令。

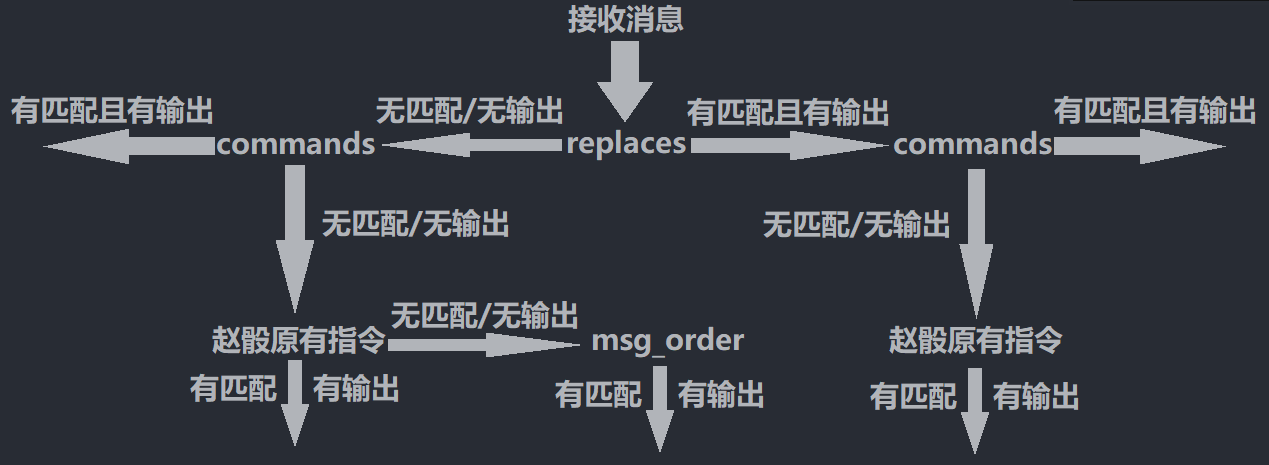
练习2.4.3.3：写一个指令“.crazy”，重定向为san check 1d10/1d100.

**2.4.3.5 优先级**

当Lua插件中同时存在多个指令定义时，便难免讨论它们的优先级与执行顺序。

经过实践，三类指令定义的顺序如下：消息接收后，优先尝试匹配replace重定向指令。若有匹配，进行重定向完毕后再尝试匹配commands指令，有匹配则执行指令，无匹配则无反应。

若没有匹配replace指令，则尝试匹配commands指令。有匹配则执行指令，无匹配则尝试匹配msg\_order关键词。有匹配则执行，无匹配则无反应。



**3 . 基 本 代 码**

NAO

***建议：使用"准备工作"中的插件代码跟进实践以便于更好地记忆、理解、掌握。***

**3.1 【一天】**

参数及格式：无。

返回86400，即一天的秒数

输入=【一天】

输出=86400

一点补充：关于测试插件的使用——向你的骰子发送“&【一天】”

//向骰子发送消息

&【一天】               //输入=【一天】

//骰子回复消息

86400

&运行已完成。            //输出=86400

**3.2 【块】**

参数及格式：无。

空值。【块】并不是null,而是一个空字符串""。 因为回雪不允许空参数，所以需要为空的地方都应填入【块】。

输入=【块】

//没有输出 既然是空字符串，那必然是什么输出都没有的。

输出=

//如果你在【块】里写了东西，那这个【块】便会形同虚设。

输入=【块阿巴阿巴】

输出=阿巴阿巴

**3.3 【计算】**

参数及格式：含有加减乘除及括号的运算式。

计算内部内容的结果并输出。

输入=【计算 500+20】

输出=520

//如果你输入了一串并不符合要求的字符。【计算】则会从这串字符中挑出符合条件的、最接近字符串末尾的、最长的（三个条件的优先级从高到低排列。）字符串作为计算式。

输入=【计算 a12+24b3\*3c】

输出=9

//【计算】是少有的输出结果为数字而非字符串的代码块。因此，【计算】可以用于将较大的数字转化为科学计数法

**3.4 【变量】**

参数及格式：【变量 变量名】。

输出对应变量的内容，如果不存在此变量则不输出。此时变量内容等于【块】

输入=【变量 A】

输出=        //没有输出。

输入=【赋值变量 A,欢迎使用赵赵类铃心】【变量 A】

输出=欢迎使用赵赵类铃心

**3.5 【赋值变量】**

参数及格式：【赋值变量 变量名,内容】。

赋值一个变量进行使用。任何变量在被赋值前都是空值，即【块】。同样地，也可以通过将变量赋值为【块】来将之清除。注意回雪自定义中的","为英文符号，下同

输入=【赋值变量 A,绑架赵赵】【变量A】

输出=绑架赵赵

**3.6 【常量】**

参数及格式：【常量 常量名】

调用对应的[[define]]内容块的内容。

输入=【常量 def】

输出=HELLO WORLD

常量和变量可以同名，不会发生冲突。

**3.7 【换行】**

参数及格式：无。

字面意思，相当于回车

输入=欢迎【换行】绑架【换行】赵赵

输出=欢迎

绑架

赵赵

**3.8 【随取】**

参数及格式：【随取 内容1,内容2,内容3,......】

随机取一个内容。

输入=【随取 绑,架,赵,喵,南,乔】

输出=赵

输入=【随取 绑,架,赵,喵,南,乔】

输出=南

输入=【随取 绑,架,赵,喵,南,乔】

输出=乔

**3.9 【比较】**

参数及格式：【比较 参数A,参数B,A≥B则执行此内容,A<B则执行此内容】

比较参数A与参数B大小（它们应当都是数字或计算表达式），并根据结果执行相应参数。

输入=【比较 2020,2021,赵赵大,evolution！】

输出=evolution！

Ps.在【比较】中，【块】被视为0.

Ps2.如果输入A、B不符要求，参见【计算】。

**3.10【判断】**

参数及格式：【判断 参数1,参数2,AB不同则执行此内容,AB相同则执行此内容】

判断参数1与参数2是否相同，并根据结果执行相应参数。

输入=【判断 赵赵,赵赵,just much,best much】

输出=best much

输入=【判断 南乔,赵赵,just much,best much】

输出=just much

**3.11【判空】**

参数及格式：【判空 参数1,参数2】

判断参数1是否为空，为空返回参数2。 // 赋值尝试一下吧

输入=【判空【变量 A】,变量A为空！】

//若【变量 A】为空则

输出=变量A为空！

//若【变量 A】不为空则

输出=xxx            //变量A的值

**3.12【返回】**

参数及格式：【返回 内容或代码】

取消既有输出，直接返回指定内容，然后立刻强制停止此次[[auto]]执行。

如果你有编程基础，那么可以尝试将其与函数中的return做类比。

输入=赵骰【返回 赵喵喵】Dice

输出=赵喵喵

若要仅取消输出，可以这样：

输入=赵骰 【返回 【块】】

输出=没有输出

**3.13【隐藏】**

参数及格式：【隐藏 内容或代码】

照常执行代码，但取消代码块内所有直接输出。

输入=123【赋值变量 a,456】【变量 a】

输出=123456

输入=【隐藏 123【赋值变量 a,456】】【变量 a】

输出=456

//其实不是代码的【】也能起到【隐藏】的作用，但代码在【】中无法执行，所以【】仅仅在调试与debug时会使用，并且debug或调试完成后应当及时清除。

输入=12【34】56

输出=1256

输入=12【【赋值变量 a,34】】【变量 a】56

输出=1256

**3.14【分割随取】**

参数及格式：【分割随取 分隔符,内容1分隔符内容2分隔符内容3......】

分隔内容并进行随取

输入=【分割随取 |,赵|喵|N|B】

输出=喵

输入=【分割随取 |,赵|喵|N|B】

输出=B

输入=【分割随取 |,赵|喵|N|B】

输出=赵

**3.15【执行变量】**

参数及格式：【执行变量 变量名】

解析并执行变量内容。相当于将变量内容当做一个program执行。

【执行变量】通常的用途就是执行通过【正则匹配到的】、【常量】、【读项目】等代码引用而不能直接执行的代码文本，用法是先将代码文本存入一个变量再执行。

//变量：bla=“【随取 1,2,3】”

输入=【变量 bla】

输出=【随取 1,2,3】

输入=【执行变量 bla】

输出=3

**3.16 【读项目】**

参数及格式：【读项目 项目名】

读取项目中的文本内容并进行输出。

输入=【读项目 blb】

输出=欢迎绑架赵赵

//项目内的文本只会被视为字符串，即使是代码也会原样输出，但可以使用【执行变量】执行。

输入=我是【赋值变量 代码,【读项目 zhao】】【执行变量 代码】

输出=我是12345678 //此时应是你的QQ

//读取不存在的项目，返回空值。

输入=【读项目 阿巴阿巴】

输出=            //没有输出

//program与keyword同样是项目，可以被读取。

输入=【读项目 keywordRegexp】

输出=^&([\\s\\S]\*)        //如果发现输出中出现了“\”的减少，此为正常现象。

//但是，如果同一个keyword项目有着多个匹配，这个项目将无法被读取。

[[auto]]

keywordFull=["测试","调试"]

program="【判空 【读项目 keywordFull】,空值！】"

//发送关键词“测试”或者“调试”，骰子回复消息：

空值！

**3.17【骰点计算】【详细骰点计算】**

参数及格式：【骰点计算 骰点表达式】【详细骰点计算 骰点表达式】

解析执行输入的骰点表达式。类似于.r[表达式]，不同之处在于：当骰点表达式输入不正确时，.r指令会骰出1D100，而【骰点计算】则会输出0.

不输入任何表达式也算骰点表达式输入不正确，而不是缺少参数。表达式可以含有代码。

输入=【骰点计算 1d520】

输出=79

输入=【详细骰点计算 1d520】

输出=1d520=99

输入=【骰点计算】

输出=0

输入=【详细骰点计算】

输出=骰点表达式计算失败=0

输入=【赋值变量 A,2】【骰点计算 1d【变量 A】】

输出=2

**3.18【发送者QQ】**

参数及格式：无。

输出触发关键词的人的QQ

输入=【发送者QQ】

输出=123456789 //此时应是你的QQ

**3.19【随机数】**

参数及格式：【随机数 [最小值]-[最大值]】

返回给出区间内的随机数，包含最大与最小值。该代码块的详细执行过程是：如果给出两数中含有小数或负数，则先将小数和负数取零，再返回所得区间的随机数。

任何代码都不能作为【随机数】的参数。【随机数】并不会执行内部代码，而是将代码当做普通的字符串处理——识别为0。

最大值不能小于最小值，否则会报错。最大值与最小值是指处理完毕后的最大值与最小值。

输入=【随机数 1-10】

输出=3

输入=【随机数 -3--1】

输出=0 //范围0~0

输入=【随机数 2.3-4】

输出=1 //范围0~4

//变量：最小值=“49”

输入=【随机数 【变量 最小值】-50】

输出=23 //【变量 最小值】被识别为0

3月24日消息：【随机数】将被更新为【随机数 最小值,最大值】的格式，届时将允许使用代码作为【随机数】的参数。同时，原有的【随机数】格式仍然有效。

**3.20【当前群号】**

参数及格式：无。

触发关键词的群号，如果为私聊则输出【块】。

输入=【当前群号】

输出=123456789 //此时应是你的QQ群号

**3.21【现行日期】**

参数及格式：无。

触发时的日期，日期直接取自骰子所在设备，所以更改骰子设备的时间设置能够影响结果。

输入=【现行日期】

输出=2022\_2\_20

输入=【现行日期-日】 //年、月、日，只此三项。

输出=20

**3.22【执行脚本】**

参数及格式：【执行脚本 脚本,参数1,参数2,参数3,......】

执行给定的js脚本，输入的参数会以名为v1,v2,v3......的变量的形式传入脚本。

Lib为赵骰专门的js类，你会在第七部分看到。

输入=你的今日人品为：【执行脚本 Lib.jrrp("人品")】

输出=你的今日人品为：55

输入=【执行脚本 v1+v2,23,14】 //js脚本中函数参数分隔也用半角英文逗号，这与回雪冲突，使脚本容易被回雪分割，造成脚本不完整，最终报错。因此，可以将脚本存为常量进行引用。

输出=37

**3.23【10位时间戳】**

参数及格式：无。

得到10位数的时间戳。它的意义是从1970年1月1日00:00:00到当前时间所经过的秒数。与【现行日期】相同，这个时间戳取自本地设备。

输入=【10位时间戳】

输出=1645292949

**3.24【设置固态变量】**

参数及格式：【设置固态变量 变量名】

将变量设置为固态，即能够长期存储，而非临时性的。这将通过文件操作来实现。详见5.1.2 文件输出。

输入=【赋值变量 a,3】【变量 a】

输出=3

输入=【变量 a】

输出=                //没有输出

输入=【设置固态变量 a】【赋值变量 a,3】【变量 a】

输出=3

输入=【变量 a】

输出=3

**3.25【设置变量针对个人】【设置变量针对群】**

参数及格式：【设置变量针对个人 变量名】【设置变量针对群 变量名】

将变量操作设置为针对插件触发者/触发群。该变量操作被附加了一个内容为“person*触发者QQ”/“group*群号”的标签。附加标签可以用于区分同名变量。

一次声明在同一个代码块的同一次执行过程中有效。之后必须重新声明。

更详细的讲解参见5.1.2.1 变量标签

//用户a

输入=【设置固态变量 a】【设置变量针对个人 a】【赋值变量 a,a】

//用户b

输入=【设置固态变量 a】【设置变量针对个人 a】【赋值变量 a,b】

//用户a

输入=【设置变量针对个人 a】【变量 a】

输出=a

//用户b

输入=【设置变量针对个人 a】【变量 a】

输出=b

//群a

输入=【设置固态变量 a】【设置变量针对群 a】【赋值变量 a,a】

//群b

输入=【设置固态变量 a】【设置变量针对群 a】【赋值变量 a,b】

//群a

输入=【设置变量针对群 a】【变量 a】

输出=a

//群b

输入=【设置变量针对群 a】【变量 a】

输出=b

**3.26【设置变量有效期】**

参数及格式：【设置变量有效期 变量名,有效期】

设置固态变量的有效期（以秒计）。到期后该变量失效并在下一次对应插件被触发时被清除。

更详细的讲解请看5.1.2.2 变量有效期

输入=【设置固态变量 a】【设置变量有效期 a,10】【赋值变量 a,1】【变量 a】

输出=1

7秒后

输入=【变量 a】

输出=1

15秒后

输入=【变量 a】

输出=没有输出。

**3.27【正则匹配到的】**

参数及格式：无参数或【正则匹配到的0】【正则匹配到的1】......

该代码块仅限于关键词匹配为正则匹配且存在捕获时使用。【正则匹配到的】与【正则匹配到的0】等效。不存在的捕获会导致报错。

你在"准备工作"里写的、正在用来调试代码的"&"插件就是正则匹配关键词。

使用此代码块应当先学习运用正则关键词匹配。

keywordRegexp="^(.\*)abcd(.\*)$"

program="【正则匹配到的1】"

输入=abcdefg

输出=efg

**3.28 【到整数】**

参数及格式：【到整数 数字】

将数字取整，取整方向为向0取整。

//正数向下取整

输入=【到整数 3.2】

输出=3

//负数向上取整

输入=【到整数 -2.5】

输出=-2

//可以与【计算】结合使用

输入【计算 37/5】

输出=7.4

输入=【到整数 【计算 37/5】】

输出=7

**4 . 回 雪 编 写**

联的冮山

回雪作为赵骰开发者构建，仅用于编写赵骰插件拓展的语言，或者说工具，具有非专业、易理解、易上手、上限低的特点。它能够满足绝大部分骰主的大部分插件需求，而js和Lua也补足了回雪自定义的部分缺陷——或者说，力所不能及之处——以更大程度地满足骰主的需求。因此，只需教以最基本的思路与简单的规范化习惯，即可投入实践编写插件。

虽然如此，我仍然建议初学者应耐心地从简单的插件开始尝试，积累经验，然后再逐步加大难度。

**4.1 基本方法**

回雪的基本编写方法是顺次、嵌套与拼接。任何回雪代码技巧都是由这三个基本方法组合而成。

**4.1.1 顺次**

顺次指将若干代码按一定顺序摆列以达到目的。如很多插件都有类似这样的模块：

[[define]]

name="前置"

content="""【设置固态变量 攻击】【设置变量针对个人 攻击】【设置固态变量 防御】【设置变量针对个人 防御】……"""

这就是一个以顺次方法写的代码块。不过，仅使用顺次方法能够实现的功能非常少。

**4.1.2 嵌套**

嵌套是指回雪代码可以作为其它代码的参数进行嵌套。嵌套是编写插件时最常用的方法。

【赋值变量 金币,【计算 【变量 金币】+【随机数 5-10】】】

纯回雪插件里往往容易存在大量、多层的嵌套，俗称“套娃”，例如写一个普通的检定，在判断成功等级时，就会出现“套娃”。

【比较 【变量 出目】,【比较 【变量 目标值】,50,100,96】,大失败,【比较 1,【变量 出目】,大成功,【比较 【计算 【变量 目标值】/5】,【变量 出目】,极难成功,【比较 【计算 【变量 目标值】/2】,【变量 出目】,困难成功,【比较 【变量 目标值】,【变量 出目】,成功,失败】】】】】

“套娃”不论是编写还是debug都比较耗费精力，因此“套娃”不宜过多，除非有代码的格式可读性强，或者编写者本人比较善于阅读套娃。同时，也可以尝试妥善利用常量和项目来拆解套娃。

**4.1.3 拼接**

拼接指文本拼接，基于回雪高度自由且毫无难度的文本拼接。

我们从一个最简单的拼接开始。

[[auto]]

keywordFull="骰1d100"

program="1d100=【随机数 1-100】"

输出的结果你应当已经猜到了。在回雪中，想要将骰点结果接在“1d100=”后面时，直接拼接即可。而在其它编程语言中，如js，需要这么做：

var roll=Math.ceil(Math.random()\*100)    //这是【随机数 1-100】

"1d100="+roll    //这是字符串拼接

"1d100="+Math.ceil(Math.random()\*100)    //合在一起

实际上难度差异并不大，但回雪的拼接方式极其方便，并且与嵌套结合使用可以发挥更大的价值。例如数组、对象、合并同类项等。

[[auto]]

keywordRegexp="^剪刀石头布，我出(.\*)$"

program="【常量 【正则匹配到的0】对战【随取 石头,剪刀,布】】"

[[define]]

name="剪刀对战石头"

content="我出石头！好耶！赢了！"

[[define]]

name="剪刀对战剪刀"

content="我出剪刀！嗯……平局，再来一次！"

--（以下略）--

**4.2 规范化插件**

诚然，我们既不是回雪开发者，也不是专业人士，因此使用“规范化”这个词有欠妥当。我们使用这个词只是为了表达规整、有序的意味。

规范化插件并不是必修内容，如果你认为自己不需要学习规范化，那么可以尽管跳过此节内容，直接阅读第五章。

如果你苦恼于插件混乱理不清代码导致debug困难，那么我更建议你阅读本节，从源头上解决debug难的问题。我们把规范化拆为格式化与模块化两方面，分别进行讲述。

**4.2.1 格式化**

当你开始编写更加复杂的回雪插件，亦或是学习别人的较为复杂的回雪插件时，你将不可避免地面对可读性差、容易出现错误的“套娃”。为了代码的可读性，也为了debug方便，便需要通过规整格式来让“套娃”更加整洁易读。常用的方法是利用回车键（Enter换行）与制表键（Tab缩进）来规整代码格式。

//推荐格式之一

【代码

参数1

,

参数2

,

……

】

//举例，判断一次检定的成功等级

【比较

   【变量 出目】

,

   【比较

       【变量 目标值】

  ,

       50

  ,

       100

  ,

       96

   】

,

   大失败

,

   【比较

       1

  ,

       【变量 出目】

  ,

       大成功

  ,

       【比较

           【计算 【变量 目标值】/5】

      ,

           【变量 出目】

      ,

           极难成功

      ,

           【比较

               【计算 【变量 目标值】/2】

          ,

               【变量 出目】

          ,

               困难成功

          ,

               【比较

                   【变量 目标值】

              ,

                   【变量 出目】

              ,

                   成功

              ,

                   失败

               】

           】

       】

   】

】

//具体的格式以你自己的判断为准。格式主要是为你自己服务的，它应当是最适合你的那种。

//例如这种示例格式其实过度分散了，反而不利于阅读。

除“【”与开头标识词不能被中断外（如“【赋值1变量”，空格也不行），回雪代码可以自由地使用回车与制表。但是，这些回车与制表符会被代码忠实地识别并执行。因此，如果不能合理地处理，就容易出现输出结果格式崩溃的悲剧。例如这个悲剧的惩罚骰检定模拟：

1D100=

1

{

4

,

5

,

1}=51/47

失败。

虽然这个问题可以被一次性输出、输出整合、【】保护回车等方法解决，但这又会引起代码的臃肿或混乱。这意味着回雪有着输出美观、代码美观与代码精简间极易冲突的缺陷。不过，我们还是简单讲述下这三个解决输出格式问题的方法。

**4.2.1.1 一次性输出**

代码中无输出，而将输出内容装入项目或常量中，输出时直接读取即可。

--在好感度插件中写一个告白的功能

--此为错误示范。

[[auto]]

keywordFull="我喜欢你"

program="唔……

【赋值变量 目标值,【变量 好感度】】

【赋值变量 取悦检定,【常量 检定】】

【赋值变量 结果,【执行变量 取悦检定】】

【判断 【变量 结果】,失败,

   【判断 【变量 结果】,大失败,

       真……真的吗？

       【赋值变量 增值,【随机数 5-10】】

       （好感度增加【变量 增值】）

       【赋值变量 好感度,【计算 【变量 好感度】+【变量 增值】】】

  ,

       什么……你？你以为你是谁？

       （好感度减半）

       【赋值变量 好感度,【计算 【变量 好感度】/2】】

   】

,

   我们还是做朋友比较好……

   【赋值变量 减值,【随机数 5-10】】

   （好感度减少【变量 减值】）

   【赋值变量 好感度,【计算 【变量 好感度】-【变量 减值】】】

】"

输出=唔……

我们还是做朋友比较好……

   （好感度减少10）

--一些换行和缩进被视为回复的一部分，我们可以采用一次性输出的方法。

[[auto]]

keywordFull="我喜欢你"

program="【赋值变量 目标值,【变量 好感度】】

【赋值变量 取悦检定,【常量 检定】】

【赋值变量 结果,【执行变量 取悦检定】】

【判断 【变量 结果】,失败,

   【判断 【变量 结果】,大失败,

       【赋值变量 增值,【随机数 5-10】】

       【赋值变量 好感度,【计算 【变量 好感度】+【变量 增值】】】

       【赋值变量 回复,【常量 告白成功】】

       【执行变量 回复】

  ,

       【赋值变量 好感度,【计算 【变量 好感度】/2】】

       【赋值变量 回复,【常量 告白大失败】】

       【执行变量 回复】

   】

,

   【赋值变量 减值,【随机数 5-10】】

   【赋值变量 好感度,【计算 【变量 好感度】-【变量 减值】】】

   【赋值变量 回复,【常量 告白失败】】

   【执行变量 回复】

】"

[[define]]

name="告白大失败"

content="唔……

什么……你？你以为你是谁？

（好感度减半）"

[[define]]

name="告白失败"

content="唔……

我们还是做朋友比较好……

（好感度减少【变量 减值】）"

[[define]]

name="告白成功"

content="唔……

真、真的吗？

（好感度增加【变量 增值】）"

输出=唔……

我们还是做朋友比较好……

（好感度减少10）

**4.2.1.2 输出整合**

输出整合是指，利用一个变量将执行过程中所有要输出的内容整合起来，并在插件执行结尾一次输出。使用输出整合时需要注意换行。可以使用“\n”或者“换行”来保证代码规整的同时做到输出美观。

--我们仍然以上面的好感度为例，直接放出使用输出整合方法的代码。

[[auto]]

keywordFull="我喜欢你"

program="【赋值变量 回复,唔……】

【赋值变量 目标值,【变量 好感度】】

【赋值变量 取悦检定,【常量 检定】】

【赋值变量 结果,【执行变量 取悦检定】】

【判断 【变量 结果】,失败,

   【判断 【变量 结果】,大失败,

       【赋值变量 回复,【变量 回复】\n真……真的吗？】

       【赋值变量 增值,【随机数 5-10】】

       【赋值变量 回复,【变量 回复】\n（好感度增加【变量 增值】）】

       【赋值变量 好感度,【计算 【变量 好感度】+【变量 增值】】】

  ,

       【赋值变量 回复,【变量 回复】\n什么……你？你以为你是谁？】

       【赋值变量 回复,【变量 回复】\n（好感度减半）】

       【赋值变量 好感度,【计算 【变量 好感度】/2】】

   】

,

   【赋值变量 回复,【变量 回复】\n我们还是做朋友比较好……】

   【赋值变量 减值,【随机数 5-10】】

   【赋值变量 回复,【变量 回复】\n（好感度减少【变量 减值】）】

   【赋值变量 好感度,【计算 【变量 好感度】-【变量 减值】】】

】

【变量 回复】"

**4.2.1.3 【】保护回车**

这其实是一个不被推荐使用的方法。【】保护回车的具体做法为利用无效【】会隐藏内部内容的性质，将所有影响输出格式的换行和缩进都用【】括起来，使其不会出现在输出中。

这种方法会极大地影响代码的美观性，同时还需要自行判断究竟哪些换行/缩进影响了输出的格式，哪些需要保留，因此并不推荐。

--使用【】保护回车方法的代码。

[[auto]]

keywordFull="我喜欢你"

program="唔……

【赋值变量 目标值,【变量 好感度】】【

】【赋值变量 取悦检定,【常量 检定】】【

】【赋值变量 结果,【执行变量 取悦检定】】【

】【判断 【变量 结果】,失败,

   【判断 【变量 结果】,大失败,

       真……真的吗？

       【赋值变量 增值,【随机数 5-10】】【

       】（好感度增加【变量 增值】）

       【赋值变量 好感度,【计算 【变量 好感度】+【变量 增值】】】

  ,

       什么……你？你以为你是谁？

       （好感度减半）

       【赋值变量 好感度,【计算 【变量 好感度】/2】】

   】

,

   我们还是做朋友比较好……

   【赋值变量 减值,【随机数 5-10】】【

   】（好感度减少【变量 减值】）

   【赋值变量 好感度,【计算 【变量 好感度】-【变量 减值】】】

】"

**4.2.2 模块化**

模块化是指，通过增设常量，将代码按照功能、形式等拆分为若干个模块，方便解析与修改。

模块化插件中的模块会被集中为几个区。区的划分并不是以触发关键词的异同为标准，而是按照模块功能大致分为触发模块、执行模块、回复模块三类。（Ps.这意味着模块化插件往往使用一次性输出）为了方便对插件编写不熟悉的骰主更改插件的回复词和触发词，模块化插件中的模块一般以回复模块、触发模块、执行模块的顺序排列。

我们按照插件执行的顺序快速讲述这三种模块。

--模块化插件中的触发模块往往很短，因为其中的代码都被转移至执行模块，program中只剩下对自己所对应执行模块的声明。

[[auto]]

keywordxxx="xxx"

program="

【赋值变量 执行,【常量 xxx模块】】

【执行变量 执行】"

--执行模块装载了插件基本所有代码，并按照功能拆分，使用【赋值变量】【执行变量】相互连接执行。

[[define]]

name="xxx"

content="【xxx】……"

--回复模块参考4.2.1.1节一次性输出，里边只装载了回复语句，可能存在变量。

[[define]]

name="xxx"

content="xxx【xx】xxx……"

**4.3 反斜杠“\”**

反斜杠“\”用于特殊字符的输入与输出。从插件载入开始，到每一次执行、输出等，都包含了编译文本的过程。而每一次文本编译都会使得文本中的反斜杠被消耗。因此，在编写插件时，常常需要2n个反斜杠才能达到预期的效果。至于具体的数量，就要根据具体情况结合自身经验通过不断试错来判断。

例如，[[auto]]内容块的keywordRegexp正则匹配，就需要使用两个反斜杠“\\”来起到反斜杠的作用。

**5 . 输 出**

联的冮山

插件可以存在没有输出的隐蔽执行的代码、程序、内容块。但整个插件没有任何输出未免过于不可理喻且毫无意义。一个只会输入却不会给出任何结果、任何输出的插件只是一个没有任何意义的黑箱。输出是赵骰插件的重要一环，也是本节主要内容。

**5.1 各类输出**

我们对赵骰插件的输出进行分类。

赵骰插件输出可以分为文件输出和消息输出，而消息输出又可以分为指定输出与直接输出。

由易到难讲解。

**5.1.1 直接输出**

直接将运行结果发送给触发来源。触发来源是群，即将运行结果发送给触发群。触发来源是私信，则将运行结果发送给相应私信。

直接输出是最常用的输出，尤其是对于插件入门者而言，这一般是唯一的输出。

[[auto]]内容块中的content就是将内容直接输出，而program则是将内容解析、执行完毕后，将所得结果直接输出。

在program中，被解析为非代码的部分，例如“123【执行变量 a】789”中的“123”，会被直接存入缓存。被解析为代码的部分则经过执行，所得结果（可能为空）也会被存入缓存。例如，设“【执行变量 a】”的执行结果为“456”，则缓存由“123”变为“123456”，再变为“123456789"。在消息所触发的所有内容块执行完毕后，缓存存储的内容才会被直接输出。

若一条消息触发了多个[[auto]]内容块，且存在至少两个[[auto]]内容块有直接输出，那么就会产生输出冲突。若同一插件文件中的两个或多个[[auto]]内容块产生了输出冲突，则相对位于最上方的[[auto]]成功输出，其他内容块的输出被取消。

**5.1.2 文件输出**

说得花里胡哨，其实“文件输出”就是存储变量的意思。直到你学会Lua的文件操作前，文件输出都是这个意思。在custom文件夹里有一个data文件夹，里面存放着插件名（包括后缀）.json文件，即为相应插件的数据。

赵骰插件变量分为固态变量和临时变量，只有固态变量会被存入data文件，并能够在下一次插件触发或同一插件下一个内容块运行时被调用，而临时变量则会在内容块执行完毕后被立即清除。

所有被存储的变量都是字符串的形式，再加上#{}系列代码（详见5.2节），变量便可以存储文字、网址、数字、图片链接、音频链接等。

{

   "value":[

      {

           "value":"变量内容"

           "expiretime":0    //变量有效期【设置变量有效期】

           "tag":""    //变量标签【设置变量针对个人/群】

           "key":"变量名"

      },

      {

           "value":"变量2内容"

           "expiretime":1645261020282    //到期时间-13位时间戳

           "tag":"group\_871504232"

           "key":"变量名2"

      }

  ]

}

使用【设置固态变量】代码声明一个固态变量。这个变量不会被立即存入data文件——文件是间歇更新。但你可以使用.system save指令手动更新所有data文件，特别是在你需要暂停、重启骰子时。

值得注意的是，如果所在内容块结束时，该固态变量为空值，即【块】，那么该固态变量同样会被予以清除，下一次调用必须重新设置固态变量。

固态变量会在骰子缓存暂时存储，稍后在data文件永久存储。在读取数据时，缓存的优先度大于data文件的优先度。system save与system load可以手动将缓存存入data文件。

**5.1.2.1 变量标签**

使用【设置变量针对个人/群】代码为变量设置标签，这在data文件中展现为变量的tag。

【设置变量针对个人/群】并不是对变量赋值，而是对于变量读写的设置。在设置之后的内容块执行过程中，该变量的读取、赋值、设置为固态，始终针对持有该tag的变量，而忽视同名的、不同tag的变量。

输入=【赋值变量 a,1】【设置变量针对个人 a】【变量 a】

输出=没有输出。

在没有设置变量针对个人/群时，并不意味着变量的读写仅针对无标签的变量，而是恰恰相反，变量的读写不会针对标签，而是对检索变量时最先遇到的同名变量进行读写。这很容易造成数据串通、出现bug。

在js脚本中，可以使用Lib.setTag(key,tag)函数进行更多、自定义tag的设置。

**5.1.2.2 变量有效期**

使用【设置变量有效期】为变量设置有效期限，这在data文件中展现为变量的expiretime.

设置变量有效期只对固态变量有意义，因为临时变量会被立即清除。

【设置变量有效期】中的参数为变量持续有效时间的秒数，而data文件中的expiretime则是变量的到期时间的、精确到毫秒的13位时间戳。

插件被触发时，所有expiretime大于0、小于当时时间戳的固态变量会被清除。

练习5.1.2：这些知识足够你写出好感度插件的最基本功能了。

**5.1.3 指定输出**

指定输出是js脚本的功能。使用如下四行代码对指定的群或个人发送指定内容。

//通过指定群向指定个人发送信息（可能发起临时对话）。

Lib.sendPrivateMessage("内容","群号","QQ号")

//向指定好友发送消息。

Lib.sendFriendPrivateMessage("内容","QQ号")

//将内容发送至指定群。

Lib.sendGroupMessage("内容","群号")

//将内容发送至触发来源

Lib.reply("内容")

除了能够指定发送目标、必须使用js外，指定输出与直接输出的另一处不同是：指定输出是实时的。只要代码执行到这一行，指定输出就会立刻执行。而直接输出则到消息处理结束时才会执行。

另外，指定输出也不会出现输出冲突的情况。因为所有的指定输出都是分别执行的，即使是向同一群聊/个人指定输出，也一定存在着先后顺序，被分为两条或多条信息发送。

不过这些都必须使用js脚本。至于js的编写，参见后面的章节：第七章 JavaScript.

练习5.1.3：你可以做一个具有最基本功能的监视器。它会将含有特定关键词的消息转发给你。

**5.2 输出前的处理**

如同输入那样，在直接输出与指定输出时，赵骰与mirai会先对输出内容进行一定的处理，使得输出与实际发出的消息有所不同。处理具体可以分为三种：特殊文本、Mirai码、句首处理。

**5.2.1特殊文本**

赵骰会对输出内容中的一些特殊文本进行处理。这些特殊文本的格式为“{xxx}”或“#{xxx}”。

多数特殊文本的作用是调用数据，例如#{PICTURE}或{user}。但也有的是对现有输出进行处理，例如#{AT}和#{SPLIT}。

具体可见[赵骰官网|特殊文本](https://trpgbot.com/specialCodes)。

练习5.2.1：第零章的调试插件中，调试结果后会另起一行提示“&运行已完成”。让它另起一条信息发送。

**5.2.2 Mirai图片/at码**

Mirai图片码的处理是Mirai框架所为。图片码是字符串，可以被分割处理，也可以被拼接而成。输出时，Mirai框架会识别Mirai图片码，并将其转化为相应的图片。

Mirai图片码可以跨骰调用。

Mirai at码是输出限定：骰子@他人时，控制台会显示骰子的输出：[mirai:at:QQ号]，使用此码即可达到与#{AT}相同的效果，即@他人。

练习5.2.2：试试让骰子复读图片。（小心误刷屏）

**5.2.3 句首处理**

特殊文本处理完成后，赵骰还会对输出消息开头进行处理。例如，赵骰会除去内容开头所有的空白符。

另外，赵骰还会除去所有的点“.”、句号“。”、中文感叹号“！”、英文感叹号“!”，其目的是防止输出文本有意或无意地被其他骰识别为骰点指令导致炸骰。这对骰子的安全性危害很大，特别是当这种情况发生在大群时。（这也是跑团或骰娘大群都有“禁止裸指令”这条群规的原因）

**6 . 插 件 调 试**

联的冮山&竹林

能够一次性写出成功完备的插件可能性很小。测试是一定需要的，反复出现bug也是难免的。但是，掌握一些技巧与经验显然可以大大缩减你的痛苦。

**6.1 装载插件**

回雪自定义测试第一步是插件能否装载成功。如果连装载都无法成功，那么一般说明在内容块的格式上出现了问题。

system load 重装插件后，请及时查看控制台提示是否装载成功，不要盲目调试。

1.确定文件编码是否为UTF-8

2.寻找成对引号：重点排查program和content的末尾三引号是否缺失

·program及content下属的成对引号(""" """ 或 " ")在编写、复制、剪贴过程中丢失了或者多了，总而言之，引号未能呈现成对匹配，导致整个插件装载失败。

3. content 、program等 未被成对引号(""" """ 或 " ")包裹的关键词有拼写（注意大小写）错误，甚至根本没写。

4.content、program等标准代码块外部存在其他不明字符

#在行首加井号为回雪单行代码注释方法

//加双斜为js单行代码注释方法

--加两短横为Lua单行代码注释方法

ANSI编码插件能够装载成功，但不会有任何反应。

**6.2 执行报错**

回雪自定义测试第二步才是正式测试。当出现了代码错误时，骰子会返回中文报错（如果是js部分出现代码错误，则为js\_error）。你并不会看到具体的报错类型，回雪只会告诉你"你这里写错了"并把它认为有错的代码标出来，但你可以凭借给出的代码及编写插件的经验来判断具体错误类型。根据经验，回雪自定义报错的类型主要有四种：

1. 参数缺失。

此类报错所给出的代码一般为【比较】、【判断】等嵌套代码。

此类报错一般表明某代码块里缺失了必要的参数。但是相比真正忘了写参数的情况，大多数情况下这都是格式出错、代码块嵌套失误的间接后果。

1. 未找到常量。

所给出的代码块为未找到的【常量】。

此类报错一般是因为常量名写错了，或者真的没有这个常量。

1. 执行超时。

骰子无响应一段时间，然后才回复报错，并给出含有【执行变量】或【执行脚本】的代码。

一般是因为代码陷入了死循环。

1. 自我保护。

一般给出一段包含了【执行变量】代码块的代码。

这是赵骰自带的有限的自我保护机制，能够防止一些简单的死循环，但也会阻止部分简单的有限循环代码。

另外，向回雪代码块中传递过量参数并不会引起报错，回雪会自动忽略后来的多余参数。这同样需要自行打开代码debug.

**6.3 插件debug**

如果你的插件没有报错，但你对运行结果不满意的话，恭喜你，开始自行debug吧。在debug时，你需要尽快缩小bug所在范围，但又不能将其遗漏。因此，一点小技巧可以帮助你。

注意，这些都是来自群员的经验总结，它并不总是正确的。

1. 格式整理。这是在bug出现前就应做的事。你的插件代码应当始终保持整洁易读的格式。“整洁易读”是对你而言，这样不仅能提高debug的效率，也能减少bug的产生。
2. 预先推断。如果你的插件有输出内容，你应当根据输出内容判断可能的bug：什么问题会让它变成这个样子？或者，如果你自认为没有那个能力或耐心，也应当根据输出内容中正常与异常的部分来大致判断bug的大概位置。
3. 顺序遍历。跟随代码的执行顺序寻找bug，可以保证你的debug过程没有遗漏。
4. 断点测试。如果你仍未能找到bug，就需要自行梳理插件运行的逻辑与机制，并使用断点进行测试。

//js

Lib.reply(string)

//回雪

【执行脚本 Lib.reply(string)】

将代码塞进需要的地方，system load，测试。根据断点反馈判断bug或调整断点位置。

断点代码属于指定输出代码，只要被执行就会立即输出。使用断点代码以测试插件某一代码段是否正常运行。

//仅确认是否某一代码段是否执行

Lib.reply("123")

//确认特定变量是否正确

Lib.reply("value\_name="+String(Value))

1. 高效求助。无能为力的你选择求助群友。你打算如何向他们描述你的问题？

“救命”？“有人吗”？“我插件出bug了”？

这样的求助没有给出任何有效的信息。在学校找老师问题可以先提一句“老师我想问一道题”，因为你知道你会立刻（一秒）得到回复，也确实是这样，但网络上呢？打开百度知道你也不会看到哪个人的问题是“有人吗”。

向群友求助时，你应尽可能给出详细的信息。插件的功能、你的预期效果、实际效果（截图）、相关代码（截图）。这才是高效的求助。

**7 . J a v a S c r i p t**

弥

你试图踏入一片未知的领地,在准备好之前请谨慎入内

请确保自己掌握最基础的JavaScript知识

如果你无法理解"var x = 1"是什么含义,请去[JavaScript 变量|菜鸟教程](https://www.runoob.com/js/js-variables.html)

**7.1 参数传递**

[[define]]

name="脚本"

content="""

\*\*\*\*\*这是你的JS脚本\*\*\*\*\*

v1 //这就是你传入的 参数1

v2 //这就是你传入的 参数2

\*\*\*\*\*这是你的JS脚本\*\*\*\*\*

"""

[[define]]

name="模块"

content="""

【赋值变量 模块句子,【执行脚本

【常量 脚本】,

【\*\*参数1\*\*】,

【\*\*参数2\*\*】

】】

【执行变量 模块句子】

"""

[[auto]]

keywordFull=""

program="""

【赋值变量 执行句子,【常量 模块】】

【执行变量 执行句子】

"""

//这是一个典型的"脚本-模块-执行"三段模板

注意:由回雪向JS传递参数的数量没有限制,但所有传递的参数都为字符串,因此当你需要数字类型的参数,譬如"十位时间戳"(当然,JS可以更方便的获得时间戳)并需要对时间戳进行数字运算时,需要令对应参数改变类型,这时需要使用Number()与String().

简单介绍两种类型:

数字类型:数字,可以进行数字运算的数字

字符串类型:字符串比起数字更像是"积木"

当两个数字类型的变量相加时,如1+1,最后输出的结果为2

但当两个字符串类型的变量相加时,如"1" + "1",最后的结果是"11",两个积木累在了一起.

**7.2 JS输出**

为了可读性与完整性,我们建议您将自己的JS脚本包装为一个函数,譬如:

function qipa(){

var said = "Hello world!";

return said;

}

qipa();

这个脚本,在执行后会输出Hello world!

我们通过function 函数名的方式构建了一个函数,因此我们需要使用return来返回我们需要的结果.

当然,只是构建函数还不够,我们需要执行函数,也就是最后一行.

显然 上例中的变量said是字符串类型,因此你可以通过said += "新字符串"来增加你输出的内容.

function qipa(){

var said = "Hello world!";

said += "Nice to meet you!";

return said;

}

qipa();

这样,输出就会变成Hello world!Nice to meet you!

但这样并不好看 因此我们需要在said中加入换行符.

有如下几种换行方式可供选择:

1. "【换行】":这是一个回雪函数,作用就是在输出的时候将它自己替换成换行符.（如果使用这种方式，那么这个脚本的执行结果必须能够回归回雪中被执行，否则"【换行】"最终会以一对括号里写着"换行"俩字的形式输出。）
2. "\n":在字符串中，\n就是换行符的意思，这是通用写法。
3. String.fromCharCode(10):这也是一种换行方法，但使用上与以上两种有所不同，因为虽然它的输出结果是字符串形式，但它本身不是字符串形式。

said += "【换行】";

said += "\n";

said += String.fromCharCode(10);

只有第三种不需要用引号.

**7.3 JS的Lib库**

Lib.setAsSolidValue("变量名")//相当于回雪的【设置固态变量 变量名】

Lib.setTag("变量名","tag")    //当tag为"person\_QQ号"时等同于【设置变量针对个人】

                            //当tag为"group\_群号"时等同于【设置变量针对群】

Lib.setValue("变量名","值")//相当于【赋值变量】

Lib.getValue("变量名")//相当于【变量】

Lib.getSender()//相当于【发送者QQ】

Lib.getGroup()//相当于【当前群号】

Lib.wget("网页链接")//获取网页get返回,常用于调用API

Lib.roll("骰点字符串")//roll点,返回数字.例:Lib.roll("1d100")返回:24

Lib.rollDetail("骰点字符串")//rool点,返回详细过程.例:Lib.rollDetail("1d3" + "+2" + "+1d5")返回:1d3+2+1d5=3+2+1=5+1=6

Lib.getConfig(key:String)//获取[[auto]]下的配置节

Lib.getConstant(key:String)//获取[[define]]下的常量

Lib.draw(name:String)//抽取牌堆

Lib.jrrp(salt:String)//今日人品

Lib.sendGroupMessage("消息","群号")//向group发送群消息

\*\*高危慎用\*\*Lib.sendPrivateMessage("消息","群号","QQ号")//向id通过group发送群私聊消息

Lib.sendFriendPrivateMessage("消息","QQ号")//向id发送私聊消息

Lib.reply("消息")//直接回复,回复的内容不会经过回雪执行,需要注意.

Lib.groupChat()//返回true/false,判断是否为群聊

Lib.sleep(time)//延迟(ms),最高延迟5s(4999),注意:在延迟期间骰无法做出任何反应

Lib.getStringMiddle("字符串","左字符串","右字符串")//从"字符串"中取出在"左字符串"与"右字符串"之间的内容

练习7.1~7.3：如果你写了好感度插件，尝试让自己能够查看他人的好感度。

**7.4 JS对象**

你已经做好在情人节为自己new一个对象的准备了吗?

如何new一个空对象:

var obj = new Object();

var obj = {};

//两者没有区别

那对象有什么作用呢?

一个对象,可以包含若干个属性与方法.

在上面我们声明一个空对象的基础上,你可以向其中添加属性与方法.

通过赋值语句,我们可以向obj这个对象中添加一个值为value的name

obj.name = value;

下面这种声明对象的方法,在声明时就设置了三个属性:

var car = {name:"Fiat", model:500, color:"white"};

这是一个包含三个属性的对象,name分别为name,model和color,值为字符串"Fiat",500和"white"

那怎么读取对象中的属性呢?

如果我们要取出name为model的属性,可以:

var model = car.model

var model = car["model"]

//两者没有区别

这样model就被赋值为500.

上面说到,对象不止包含属性,还包含方法,那方法是什么呢?

对象方法本质上是一个函数,譬如声明对象时添加方法:

var obj = {methodName:function(){return "1"}}

或声明后添加新方法

obj.methodName=function(){return "1"}

在拥有方法后,方法也类似属性存在对象中,因此方法有两种调用方法,同时也有两种不同的结果

obj.methodName()

//这样调用会执行该方法,因此将返回字符串'1'

obj.methodName

//这样调用类似于属性,因此将返回字符串'function(){return "1"}'

**7.5 JSON**

对象是好东西,但回雪只能储存字符串,那怎么办呢?

让我们隆重欢迎两个函数

json = JSON.stringify(obj)

obj = JSON.parse(json)

JSON.stringify()函数可以将对象转变成字符串类型的JSON,与此相反

JSON.parse()函数可以将字符串类型的JSON转变成对象.

如果你不会存储JSON,那请看7.3的Lib库

\*注意:很多API的返回都是JSON格式,但要仔细观察

\*\*注意:有些API会在返回的JSON中下毒(x),遇到这种请在wget得到JSON后使用trim()方法清除空白部分

//例1:

[

    {

        "breeds":[

        ],

        "id":"\_KGLkkogN",

        "url":"https://cdn2.thecatapi.com/images/\_KGLkkogN.jpg",

        "width":3278,

        "height":2502

    }

]

//例2:

{

"pic":"https://tvax3.sinaimg.cn/large/004kfMibgy1guucknfx9wj629s4547wp02.jpg"

}

//例3:

•{

"pic":"https://tvax3.sinaimg.cn/large/004kfMibgy1guucknfx9wj629s4547wp02.jpg"

}

//你能看出例3与例2有什么区别吗?

//那个红点是一个看不见的空白符,感谢Typora让我看到这个东西

有些API的返回会在最外层套一个[](例1),[]表示这是一个数组,因此你需要在最外层添加[0],下面我们来看看:

//例1:

var json = Lib.wget(API)

var obj = JSON.parse(json)

obj[0].url

//例2

var json = Lib.wget(API)

var obj = JSON.parse(json)

obj.pic

//例3(毒)

var json = Lib.wget(API).trim()

var obj = JSON.parse(json)

obj.pic

练习7.5：上网上找点api，自己试试解析应用。

**7.6 JS进阶**

如果你想继续学习JS的话,欢迎使用[JavaScript教程|菜鸟教程](https://www.runoob.com/js/js-tutorial.html)

在此仅列出一些对插件编写帮助较大的内容:

[JavaScript 运算符|菜鸟教程](https://www.runoob.com/js/js-operators.html)

[JavaScript 逻辑运算|菜鸟教程](https://www.runoob.com/js/js-comparisons.html)

[JavaScript If...Else语句|菜鸟教程](https://www.runoob.com/js/js-if-else.html)

[JavaScript switch语句|菜鸟教程](https://www.runoob.com/js/js-switch.html)

[JavaScript for循环|菜鸟教程](https://www.runoob.com/js/js-loop-for.html)

进阶Plus:

[JavaScript 对象|菜鸟教程](https://www.runoob.com/js/js-obj-intro.html)

[JavaScript 正则表达式|菜鸟教程](https://www.runoob.com/js/js-regexp.html)

[JavaScript JSON|菜鸟教程](https://www.runoob.com/js/js-json.html)

**8 . L u a**

竹林&弥

**8.1 基本格式**

--lua的基本格式（在行首加--为lua的注释方法）

[[lua]]

name="名称"

help="提示"

lua="""

   --前置内容，赋值以下为空table（懂了以后可以自行删减）

   commands = {}

replaces = {}

   msg\_order = {}

sdk = ZhaoDiceSDK

   --函数模块(用end结尾)

   function func\_name(msg)

--此处输入函数具体代码，比如if then语句，示例如下：

       if(msg.fromQQ == "568693644") then    --此处引号内QQ号请改成自己的QQ号

           --此处输入函数具体代码，比如return语句，示例如下：

           return "我听到了主人的呼唤"          --字符串请用引号包裹

       end

end

   function func\_nameRep(msg)

return "指令1"

end

--指令具体

commands["指令1"] = "func\_name"

       --输入： .指令1

       --返回： 运行所指向的函数名

       --commands为新增一个bot指令至一个函数，受bot on/off 控制，指令前要加点

replaces["指令2"] = "func\_nameRep"

       --输入： .指令2

       --返回： 该函数返回的指令（指令1）

       --replaces为重定向一个bot指令至另一个bot指令，受bot on/off 控制，指令前要加点

   msg\_order["指令3"] = "func\_name"

       --输入： 指令3

       --返回： 运行所指向的函数名

       --msg\_order为新增一个reply指令至一个函数，受reply on/off 控制，指令前不加点

"""

\*可能有误，随时更新（闭目）

**8.2 插件示例**

以下仅列出插件列表供大家学习参考，举一反三，推荐按顺序看

**8.2.1 【官方补丁】暗骰屏蔽**

因腾讯于2022年处加强了对于群临时对话审核，我们通常建议骰主装载此补丁以屏蔽暗骰功能，同时也是一个可以着手学习Lua的方向。

参照此插件举一反三，能对让赵骰对任何bot指令做出你想要的固定回复。

[[lua]]

name="暗骰屏蔽"

help="这个插件会干掉。rh功能，返回自定义的字符串。本插件中返回的是『{nick}已经关闭暗骰功能，请{player}在加好友后小窗手动掷骰。』"

lua="""

   commands ={}

   commands["rh"]="rhDisable"

   commands["h"]="rhDisable"

   commands["rah"]="rhDisable"

   commands["psy"]="rhDisable"

   commands["team desc"]="rhDisable"

   commands["teamdesc"]="rhDisable"

function rhDisable(msg)

return "{nick}已经关闭暗骰功能，请{player}在加好友后小窗手动掷骰。"

end

"""

**8.2.2 （群文件）示范屏蔽某个赵骰功能的Lua**

本插件展示了如何彻底无效化一个赵骰bot功能，与『暗骰屏蔽补丁』的区别在于彻底无效化之后不会出现指令冲突的问题。现在，你可以另起文件，使用你所熟悉的类铃心或js语言来自定义该功能了！

（因为bot功能始终优先于reply，所以你会发现在编写插件时，如果你的keyword设置了与bot功能一样的指令，该插件是无法输出的，而彻底无效化可以解决这种冲突。）

[[lua]]

name="deck屏蔽"

help="这个插件会彻底无效化。deck功能"

lua="""

commands = {}

replaces = {}

sdk = ZhaoDiceSDK

function deckDisable(msg)

return "这串字符是返回不了的，因为replaces必须指向一个能返回另一个有效commands的函数"

end

replaces["deck"] = "deckDisable"

"""

**8.2.3 【官方扩展】赵骰DND特化模块**

此插件详细展示了使用Lua编写一个完整掷骰功能的方法，如果你能把这个拆出来，意味着可以真正重写赵骰某个功能了，快为bot添加一些花活吧！

（例如给你的好感商店添加“黑曜石骰”／“水晶骰”／“情侣绑定骰”／“万圣节限定”等皮肤……让购买了该皮肤的玩家在日常的跑团骰点过程中也可以享受到至尊会员的优越体验（雾

**8.2.4 （群文件）luafile Lua进行的文件io**

本插件展示了如何使用Lua进行文件的io,对系统内文件操作给赵骰带来了无限的可能,但也同样伴随着极大的风险,因此当你编写包含io功能的Lua插件时,一定切记添加权限判定,或严格限制io路径,以防造成严重后果,截止2022.3.21,赵骰仅PC端支持文件io.

#以下文件io路径都限制为custom\data,也就是插件数据保存路径

[[lua]]

version=1

name="lua\_write"

help="lua\_write"

lua="""

commands = {}

function luawrite(msg)

if (msg.fromQQ == "权限QQ")

then

local i = msg.fromParams

local fileNumber = string.find(i," ")

local fileName = string.sub(i,0,fileNumber)

local fileContent = string.sub(i,fileNumber+1)

local function writeFile(fileName,fileContent)

local f = assert(io.open(fileName,"a+"))

f:write(fileContent)

f:close()

end

writeFile("data\\\\\\\\zhao.dice\\\\\\\\cocdata\\\\\\\\custom\\\\\\\\data\\\\\\\\"..fileName,fileContent)

return "已于"..fileName.."写入："..fileContent

end

end

commands["filewrite"]="luawrite"

"""

#以上内容为Lua的文件写入,触发后会找寻fileName这个文件,存在则在其末尾添加fileContent的内容,不存在则新建文件后添加.

[[lua]]

version=1

name="lua\_read"

help="lua\_read"

lua="""

commands = {}

function luaread(msg)

if (msg.fromQQ == "权限QQ")

then

local fileName = msg.fromParams

local function readFile(fileName)

local f = assert(io.open(fileName,"r"))

local fileRead = f:read("\*a")

f:close()

return fileRead

end

local fileRead = readFile("data\\\\\\\\zhao.dice\\\\\\\\cocdata\\\\\\\\custom\\\\\\\\data\\\\\\\\"..fileName)

return "已于"..fileName.."读取："..fileRead

end

end

commands["fileread"]="luaread"

"""

#以上内容为Lua的文件读取,触发后会寻找fileName这个文件并以文本格式读取其全部内容

**9 . Mirai插件**

NAO&弥&Sern

在这一章，不教您写，只会教你白嫖，因此您可以跳过。

**Mirai是什么？**

解释这个问题，其实Mirai官方给的教程就足够了，在这里我放一些官方网站

[MiraiForum 论坛](https://mirai.mamoe.net/)

[Mirai 开发文档](https://docs.mirai.mamoe.net/)

[Mirai - UserManual Mirai用户手册](https://docs.mirai.mamoe.net/UserManual.html)

记住，赵骰是Mirai的一个插件，不要在Mirai的社区询问赵骰问题！！！！

**（ 骰子在铃心那边名声大概就是这样坏的 ）**

**如何在赵骰里放一个Mirai插件**

首先，需要保证你的赵骰是Mirai版本的(截止2022.2.21,赵骰仅有PC版可以添加Mirai插件)

找到你的MiraiZhao\plugins文件夹,将下载的Mirai插件放入,然后重新启动赵骰

(使用system reload也可)

*\*注意:使用system load仅能重载赵骰内插件,Mirai插件与赵骰平行,仅能通过system reload来变相重启*

**一些几乎是必备的插件！**

chat-command ：允许在聊天环境执行指令（绝对需要安装啊）

Mirai-api-http ：提供 HTTP 支持，允许使用其他编程语言的插件（ 看情况 ）

LuckPerms-Mirai : 易用的高级高效率权限组插件, 适合权限分配模型比较复杂的情况

mirai-native ：与 CoolQ 对接的一个通道，主要用于加载CoolQ插件

*Mirai用户手册里有下载链接，不过是Github网站，如果不太擅长魔法，您可以从群文件里获取*

[*未来控制台 用户手册*](https://github.com/mamoe/mirai/blob/dev/docs/UserManual.md)

**如何加载控制台与调试Mirai插件**

1. 下载群文件debug.bat，并将其放置于Laucher.bat旁边。
2. 关闭骰子程序后，启动debug.bat，将会弹出Mirai的操作台，并且开启骰子。

在给予自身权限后，可通过在Mirai操作台发送/help来查看你在私聊发送/help的全部内容。关闭这个操作台后，骰子也会一同关闭（赵：本意是用于开发者模式。）

1. 自动登录时，会加载你放入位置正确的Mirai插件，并且于操作台上显示加载成功提示。

如：I/PatPat: 摸头插件已加载

1. 确认已加载后，一些简单的一站式服务插件便可以正常使用了！恭喜你！剩下的所有加载情况和报错都会显示在屏幕上，你就可以去上面对着看你到底哪个文件放错了。

*有一些插件并不是一站式，他们甚至需要你多次重启操作台，并且要求改一些文件，或者在固定位置放置文件，这些所有的修改以及确认该放在什么地方则要去Mirai论坛上查看对应的插件说明，需要去github，或者在操作台查看报错内容进行修改。*

注：以上操作均为直接套用成型插件，能用万岁，不能用别灰心，相信你自己，可以试试看自己写出来一个你想要的功能（？）

**如何给自己权限**

\***由于有了debug.bat，以下内容废弃，在控制台输入 如下内容即可**

/perm permit u你的QQ \*:\*

\***更多指令见** [Mirai Console 内置指令](https://github.com/mamoe/mirai/blob/dev/mirai-console/docs/BuiltInCommands.md)

\***不过使用Mirai插件不少需要改配置文件，所以留下这个当例子。**

由于赵骰高度包装，你可能找不到使用命令行控制台方法

所以，直接改配置文件！

找到你的MiraiZhao\config\Console文件夹，用记事本(或别的文本编辑软件) 打开PermissionService文件

这是一个yml文件，理所当然，全英文符号。

grantedPermissionMap:

 '\*:\*':

   - console

在文本最后添加 - u+你的QQ号 就可以获得Mirai权限

\*注意:完全关闭赵骰后再尝试修改!

\*\*注意:没有加号!没有加号!没有加号!

\*\*\*注意:记得加u!记得加u!记得加u!

grantedPermissionMap:

 '\*:\*':

   - console

   - u123456789

如何确定自己成功获取权限:尝试对bot发送/help,如果有回应则表明拥有权限

如果你阅读了上述官方文档，你应该明白以上操作的具体含义。

**关于Mirai-Native插件的加载**

首先下载Mirai-Native插件:[GitHub:Mirai-Native](https://github.com/iTXTech/mirai-native)

Mirai-Native也是Mirai的一个插件,因此将其放置于MiraiZhao\plugins文件夹内,重启bot.

查看控制台可以得到:

I/MiraiNative:正在加载Mirai Native Bridge C:\Users\Administrator\Desktop\MiraiZhao\data\org.itxtech.mirainative\CQP.dll

此时就成功加载了Mirai-Native插件,要注意:CQ插件将作为Mirai-Native插件的插件存在,因此不能将其放置于MiraiZhao\plugins文件夹内,而要将其放置于MiraiZhao\data\org.itxtech.mirainative\plugins文件夹.

若插件有相关JSON需要配置,则类似的将其按照要求放置于MiraiZhao\data\org.itxtech.mirainative\data对应文件夹内.

**关于Lua Mirai**

LM是mirai生态中比较特殊的一员，其SDK 直接基于 mirai-core 开发，可以单独运行，虽然我们一般只会用mirai插件版的LM。

如果你想玩一下奇奇怪怪的玩法。比如，引用回复、合并转发……建议你使用这个。

官方文档：[LM 官方文档](https://ooooonly.gitee.io/lua-mirai-doc/)

因为官方文档一般比较枯燥，也没有例子，我随便写一个。

-- LuaMiraiScript --

-- name: roll

-- version: 1.0

-- description: 这是一个简单的脚本

-- author: nao

-- /LuaMiraiScript --

--上面的是脚本头，说明脚本的信息 print会把信息打印到控制台上 方便debug

print('roll载入脚本成功')

--监听群聊消息 GroupMessageEvent即是群聊消息事件

--作为使用者 我们无须清楚它的原理 套用格式即可

--也就是 Event.subscribe([<过滤条件>..],(<监听器>))

Event.subscribe("GroupMessageEvent",function(event)

--event对象在不同事件中不同，你可以print(event)后在控制台看看 或者看看官方文档

--在这里只需要 消息（Meesage）

local msg = event.message

--不过message也有复杂的结构 使用tostring函数获得message中的文本部分

local msgt = tostring(msg)

--如果文本是以\*r开头的 就是需要处理的

if msgt:find("\*r")==1 then

--用来debug时设置的断点 下面所有的print都一样

print('匹配到\*rd前缀,开始计算。')

--从event中引用来源群号

local sender = event.group

--创建回复文本变量

local said

--两个1d100的情况

if  msgt == '\*r' or msgt =='\*rd' then

--生成随机数 填充回复文本

r = math.random(1, 100)

said = '出目是：1d100='..r

--引用回复的方法 详见官方文档

sender:sendMessage(Quote(msg) .. said)

print("输出"..said)

else

--其实这里有一种情况没考虑到，但我懒得改了 只会if+if的废物是我了

--从消息文本中截取\*rd后面的东西并确定是数字

--tonumber函数 和sub

local d = tonumber(msgt:sub(4))

--不是数字的话 打回去 否则复制粘贴并传参

if d == nil then

said = '参数错误，请输入纯数字。'

sender:sendMessage(Quote(msg) .. said)

print("输出"..said)

else

r = math.random(1,d)

said = '出目是：1d'..d..'='..r

sender:sendMessage(Quote(msg) .. said)

print("输出"..said)

end

end

end

end)

--看着这4个end我就知道自己多烂了，扔作业，优化代码

--顾名思义 就是卸载脚本时会出现的打印。

Event.onFinish = function() print("脚本roll被卸载！") end

**附 录 1 自 我 提 高**

NAO

***建议新群员尝试写写这些插件来锻炼自己的写插件能力。***

***插件按由易到难的整体趋势排列。 （★为基本功能，☆为额外功能）***

***------冮山原拟群公告***

***别真当作业写，咱写插件不是上班。***

**试一试**

**1.关键词回复**

★把自己在赵骰APP/应用程序上设置的回复词搬进插件。

☆一个关键词有随机多种回复。

☆多个关键词匹配同一回复。

☆使用正则表达式匹配关键词。

☆如果是随机回复，尝试各回复不同概率。

☆能够通过指令而非修改插件设置关键词、回复词。

☆如果是随机回复，尝试使用一种算法，令回复分布更加均匀。

**2.定时回复**

★能够在指定时间向指定群聊/个人发送指定内容消息。

★可以是反复的，即每n秒/分/时/日/周/月/年发送一次。（你不应当拿来恶意刷屏，否则后果自负。）

☆尝试一次向多个群/个人发送消息，或向同一群聊/人发送多条信息，这很简单。

☆尝试在消息中@他人/调用牌堆/调用今日人品/调用图片，诸如此类。

**3.关键词回复群开关（以关键词回复为基础）**

★能通过指令打开、关闭指定群聊的日常用语（草，早，晚安）回复。

☆能够通过指令查看屏蔽群名单。

☆能够通过指令屏蔽指定个人。

☆能够通过指令在指定群屏蔽指定个人。

**4.快速调试**

★你应当已经写了“准备工作”里的插件。它可以让你调试代码的时候省去大量时间，提高代码的运行效率。现在，你应当为插件换一个关键词，以及执行完毕的输出提示，避免你的骰子与别的骰子撞了台词。

☆显然这个插件的某些部分只是用来学习回雪的，在文档以外的调试中毫无作用。如果你认为自己足够熟悉回雪了，就删掉它们吧。

☆执行完毕的输出提示可以分作单独一条消息，而不是放在输出最后一行。

☆可以利用固态变量设置伪常量，加强调试能力。

**5.关键词监控**

★骰检测到含有特定关键词的消息时，向骰主发送至少含有消息原文、发送人、发送场所（群或私聊）三项基本信息的消息。

☆可以通过指令设置屏蔽指定群聊、个人。

☆可以通过指令特别关心指定群聊、个人。

☆可以通过指令增减监控词。

☆可以通过指令或修改插件设置监控屏蔽词。

☆可以通过指令查看关键词、屏蔽群/个人、特别关心群/个人、屏蔽词列表。

☆可以尝试调用api获取触发者的QQ昵称、所在群名称。

**6.漂流瓶**

★可以写点东西装进漂流瓶扔进大海。

★可以随机捞漂流瓶查看内容。

☆可以选择漂流瓶是否扔回大海。

☆设置审核环节，防止不宜内容。

☆设置后台，查看漂流瓶数量、漂流瓶列表。查看指定漂流瓶内容、发送者、发送来源，删除、修改指定漂流瓶。

**7.备忘录**

★添加一条备忘录

★查看备忘录列表

★删除指定备忘录

☆备忘录分为标题和内容，查看列表只查看标题，查看指定备忘录则查看标题和内容。

☆账号分离，不同人的备忘录不同。

☆设置后台，能够查看用户列表、查看别人的备忘录、拉黑在备忘录里存写不宜内容的人。

☆列表单次显示一定数量备忘录，使用上一页、下一页功能翻页查看列表。

**8.好感度**

★特定关键词增加/减少好感度

★查询好感度

★好感度不同，台词不同。

☆增加/减少好感度的量是随机的。

☆某些关键词触发检定，目标值与奖惩骰由好感度决定，检定结果决定好感度增减。

☆既有好感度可以影响好感度增减量。

☆好感度对奖惩骰与目标值的影响效果不是线性的。

☆没有后台的插件不是好插件。（雾）

☆其他你想要的功能，比如商店。

**9.安科**

★新建一个节点。

★添加选项。

★查看已有选项。

★安科。

☆删除选项。

☆清除选项。

☆更改选项。

☆仅包含最终成型的故事log

☆带节点、选项等具体信息的详细log

**10.广播**

★添加广播对象

★查看广播对象列表

★删除广播对象

★对所有对象广播

★对所有群聊对象广播

★对所有个人对象广播

☆点对点消息

☆给予/撤销他人的操作权限

☆查看权限列表

☆备注对象

☆有限地自动添加、删除对象

☆分组功能。

★添加、删除分组。

★将对象移入、移出指定分组。

★广播指定分组。

☆命名、重命名分组

☆合并分组

☆设置默认分组，新添加的对象自动移入该分组。

☆查找对象

☆通过修改插件设置定时广播

☆广播操作记录

☆查询指定时间的广播记录

☆查询含有指定关键词的广播记录

**11.复读机**

★检测到连续n条相同消息时复读。

★有群开关

☆不同群的n独立设置。

☆多骰互刷识别反制。

**12.反复读机**

★检测到连续n条相同消息时打断。

★有群开关。

☆不同群的n独立设置。

☆复读"打断"识别反制。

**13.闹钟**

★通过指令设置闹钟在指定时刻提醒您。

★查看已经设置的闹钟。

★删除指定闹钟。

☆多类参数，既可设置年月日时分秒，也可仅给出时分秒。

☆一次性闹钟/非一次性闹钟。

☆若闹钟在群聊设置，则在该群聊@并提醒您。

☆可设置指定闹钟开关上一行所述功能。

☆可以多人使用。

☆可以设置闹钟的提醒人。

☆通过指令设置定时广播。

**14.资讯**

★api调用练习。获取群友头像和昵称、查询新闻或天气、查询疫情数据、搜索美图、翻译、查询BV号......

**15.赵骰**

★无效化赵骰原有指令

★自写代码覆盖原有指令

**附 录 2 若 干 插 件 代 码 技 巧**

**附2.1 回雪循环**

联的冮山

我们参考这个结构搭建回雪循环体代码：

for(form;judge;after){

   code

}

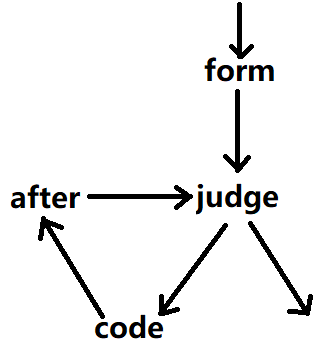
这是C++的for循环代码结构，其流程为：

执行form语句，判断judge语句的值是否为true，若是则执行code语句，否则跳过循环。执行code语句完毕后，执行after语句，再次判断judge语句的值是否为true，如此循环。

四个参数都可以为空。若judge为空，则视为true。

接下来，开始尝试搭建回雪循环体。

我们可以很容易看出form可以排除循环之外，code与after可以合并。而回雪的【判断】【比较】等可以用于judge。如果你不明白，我们可以画示意图。



这样应该比较浅显了。于是，我们就可以开始尝试使用回雪搭建循环体。

java[[define]]

name="循环"

content="【code和after代码】【比较/判断 参数1,参数2,【执行变量 循】,【块】】"

[[auto]]

keywordxxx="xxx"

program="【前续代码】【form代码】【赋值变量 循,【常量 循环】】【执行变量 循】【后续代码】"

这就是一个基本的循环体，根据实际情况可以进行改装。例如，循环常量中的【块】可以替换为文本内容

[[define]]

name="循环"

content="【code和after代码】【比较/判断 参数1,参数2,【执行变量 循】,输出内容】"

亦或者，循环体结束后仍有代码执行，便可以用代码取代【块】的位置。

[[define]]

name="循环"

content="【code和after代码】【比较/判断 参数1,参数2,【执行变量 循】,输出内容】"

如果整个program只有一个循环体，没有前续代码及form代码，那么只用program即可运行循环体。

[[auto]]

keywordxxx="xxx"

program="【code和after代码】【比较/判断 参数1,参数2,【赋值变量 循,【读项目 program】】【执行变量 循】,【后续代码】】"

其实后两种将后续代码装入循环体的做法并不推荐。因为回雪循环的本质是递归，将后续代码装入循环体意味着代码在最深一层递归执行。简而言之，耗内存。

**附2.2 常量“头文件”**

“头文件”一名引用自其它编程语言，主要内容为公共的、可以被其它文件引用的函数或方法（或其它头文件）。

在回雪中，只有常量能够被跨文件调用，而回雪的“头文件”就是指由常量集合而成的插件文件，具有精简代码、提升debug效率等优点。

//头文件

[[define]]

name="规则书检定"

content="【赋值变量 结果,【随机数 1-100】】【比较 【变量 结果】,【比较 50,【变量 目标值】,96,100】,大失败,【比较 1,【变量 结果】,大成功,【比较 【变量 目标值】,【变量 结果】,【比较 【计算 【变量 目标值】/2】,【变量 结果】,【比较 【计算 【变量 目标值】/5】,【变量 结果】,极难成功,困难成功】,成功】,失败】】】"

//插件1

"……【赋值变量 目标值,【变量 好感度】】【赋值变量 好感度检定,【常量 规则书检定】】【执行变量 好感度检定】……"

//插件2

"……【赋值变量 目标值,【计算 【计算 【变量 攻击力】-【变量 防御力】】\*10】】【赋值变量 战斗检定,【常量 规则书检定】】【执行变量 战斗检定】……"

**附2.3 回雪数组及对象**

回雪变量连续稳定的数据类型只有一种：字符串。

在需要数组形式的变量时，回雪无法像常规编程语言一样定义一个“数组名[]”。但我们可以利用回雪无需提前定义变量、字符串直接拼接参数等优势定义一组变量，将之视为一个“数组”。

数组变量名有且只有通名与序号两部分，一般采取“变量名\_序号”的格式。

联系人\_1="13322446688"

联系人\_2="15523456789"

联系人\_3="16666668888"

使用“变量名\_序号”的规范化格式有利于记忆、识别、管理、操作。同时，也为多维数组留出空间。

这就是回雪数组。

坐标\_2\_5="码头"

坐标\_4\_6="城区"

坐标\_8\_9="山区"

如果你悟性很高，你便可以想到序号不一定是数字——它不一定是序号，也可以是关键字词。同时，多维数组也不要求必须是规整的矩阵形式。

恭喜你，你学会了回雪对象。

嫌疑人\_姓名="张三"

嫌疑人\_身高="178"

嫌疑人\_行踪\_1="沈阳"

嫌疑人\_行踪\_2="延安"

嫌疑人\_行踪\_3="兰州"

嫌疑人\_行踪\_距离="2568km"

**附2.4 回雪与JS间的数组传递**

回雪向JS传递变量有两种方法：

1.在【执行脚本】中加入参数传递变量。

【执行脚本】只能传递固定数量的变量，并且回雪变量均为字符串，因此无法直接传递数组。我们可以借助JS的字符串处理函数split()完成传递。

//value:String.split(separator:String)

var str="a|b|c|d|e"

var arr=str.split("|")

arr[2]    //输出 = c

将回雪数组存储为示例中的字符串方式，其中分隔用的字符必须是不会与变量内容重复的。例如，你不能用字母去分隔单词数组。

2.在JS脚本中使用Lib.getValue()读取固态变量。

这个方法要求数组变量必须是固态变量，绝大多数情况下数组遍历都能符合这个要求。在JS脚本中定义一个数组，并使用Lib.getValue()配合for循环读取一个完整的回雪数组。

JS脚本向回雪传递变量有两种方法：

1.【执行脚本】的返回值。

【执行脚本】只能返回一个值，但我们可以采用回雪向JS直接传递变量的方法。使用join()函数处理后再传递。

//value:String.join(separator:String)

var arr=["a","b","c","d","e"]

arr.join("-")    //输出 = a-b-c-d-e

同样地，不应使用数组元素可能包含的字符作为分隔符。

2.使用Lib.setAsSolidValue()与Lib.setValue()设置并赋值固态变量。

使用Lib.setValue()配合for循环将数组元素一一赋值为回雪变量。注意，Lib.setValue()并没有设置固态变量的功能，如果需要声明，则应使用Lib.setAsSolidValue()函数。

**附 录 3 官 方 参 考 资 料**

《类铃心帮助文档》------恒萧·魔骨

《代码档案》------蛇𓆗

**附 录 4 杂 物 、 资 料 与 工 具**

联的冮山

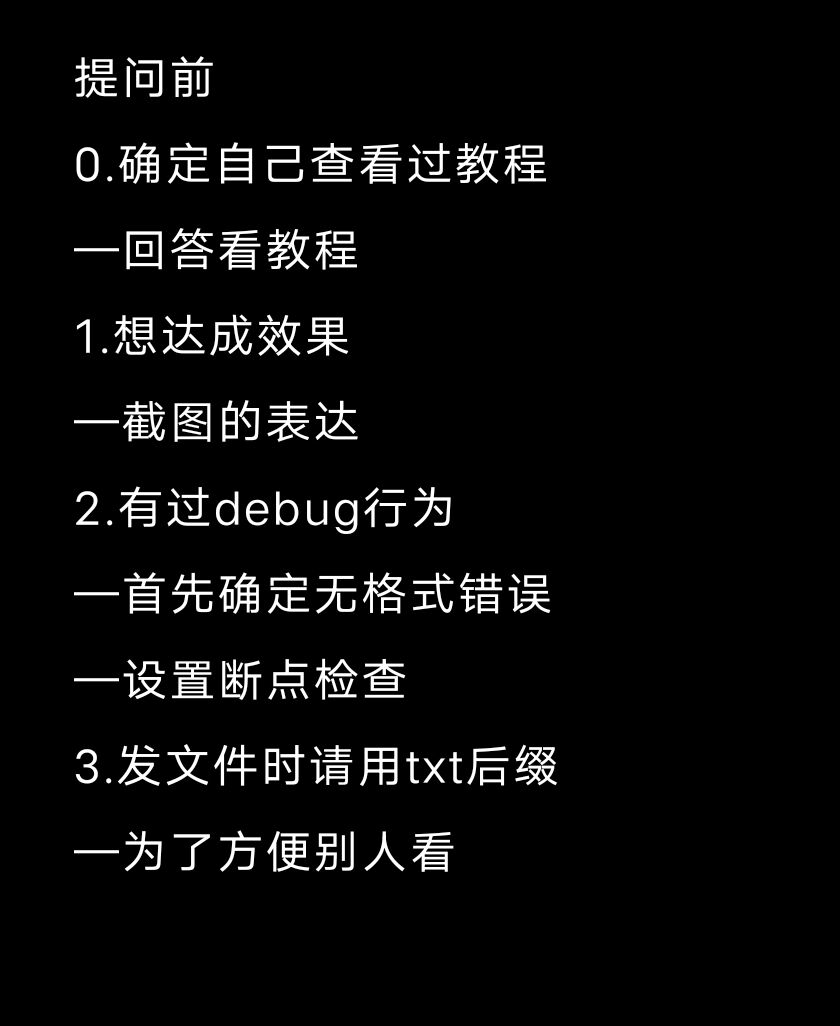
这些杂物、资料和工具是整理本群精华信息而来，如果你有任何可能用到的资料、工具，也欢迎写入本篇附录。

在线数学模拟演示工具：<https://www.geogebra.org/calculator>

一个表情包



两张图片



铃心变量： <http://myepk.club/variable.html>

正则表达式调试网站：<https://regexr.com/>

缺点是英文，浏览器自带翻译插件即可解决。

在线文档文件转换器：<https://convertio.co/zh/>

emoji官网：<https://www.emojiall.com/zh-hans>

Cron表达式（时间调度器）在线生成器：<https://cron.qqe2.com/>

Mirai官方插件发布地址：<https://mirai.mamoe.net/category/11>

一则低劣的传销宣讲：

学写插件是件有趣快乐的事，其乐趣不仅出于写出插件的成就感，也蕴于搭建代码逻辑结构的过程[。](file:///D:\users\download\%E8%B5%B5%E9%AA%B0%E6%8F%92%E4%BB%B6%E9%80%9F%E6%88%90.%E4%BB%8E%E5%85%A5%E9%97%A8%E5%88%B0%E5%85%A5%E5%9C%9F(2022-3-26).html)

学写插件能够锻炼自己的思维能力，看着自己的插件如积木般有条不紊地搭建起来，你也会不禁惊叹于精妙的作品与自己潜藏的能力。学习插件也能自力更生，让自己的需求得到更好的满足，每一处细节，每一个功能，都由你亲自定制，只有你自己才是最能满足你的人。越来越多的人参与到插件创作当中，可以使大家共同学习、共同进步，也可以为赵骰提供丰富的拓展，展现赵骰广阔的前景与不灭的活力。

诚然，刚刚入门的你会认为自己写不出好的插件，但又有几个人是一步登天的天才呢？耐心学习，掌握方法，总结经验，热心交流，你一定会得到提升。不是可能，而是一定！只要你愿意持续投入一点时间，一点动力，努力的果实终会结出，你终将拥有自己的代表作！

------联的冮山

**编 写 者 们**

NAO

弥

联的冮山

竹林

Sern·Spinix

**鸣 谢 名 单**

赵喵喵V2.0

蛇𓆗

恒萧·魔骨

羊山羊