**回雪代码档案**

**前言**： 本文档改写自《代码档案》，参考《赵骰-回雪自定义插件帮助拓展文档》。加入了编者的个人想法，以助于阅读本文档的用户理解。

*编者:解九 2023.1.6*

**目录**

**[[[auto]]](file:///C:\\Users\\ylzy2\\AppData\\Local\\Temp\\Rar$EXa27340.11383\\%E5%9B%9E%E9%9B%AA%E6%95%99%E7%A8%8B.html" \l "[[auto]])**

* [全词匹配](file:///C:\\Users\\ylzy2\\AppData\\Local\\Temp\\Rar$EXa27340.11383\\%E5%9B%9E%E9%9B%AA%E6%95%99%E7%A8%8B.html" \l "全词匹配)
* [模糊词匹配](file:///C:\\Users\\ylzy2\\AppData\\Local\\Temp\\Rar$EXa27340.11383\\%E5%9B%9E%E9%9B%AA%E6%95%99%E7%A8%8B.html" \l "模糊词匹配)
* [正则匹配](file:///C:\\Users\\ylzy2\\AppData\\Local\\Temp\\Rar$EXa27340.11383\\%E5%9B%9E%E9%9B%AA%E6%95%99%E7%A8%8B.html" \l "正则匹配)
* [多匹配词](file:///C:\\Users\\ylzy2\\AppData\\Local\\Temp\\Rar$EXa27340.11383\\%E5%9B%9E%E9%9B%AA%E6%95%99%E7%A8%8B.html" \l "多匹配词)

**[[[define]]](file:///C:\\Users\\ylzy2\\AppData\\Local\\Temp\\Rar$EXa27340.11383\\%E5%9B%9E%E9%9B%AA%E6%95%99%E7%A8%8B.html" \l "[[define]])**

**[content / program](file:///C:\\Users\\ylzy2\\AppData\\Local\\Temp\\Rar$EXa27340.11383\\%E5%9B%9E%E9%9B%AA%E6%95%99%E7%A8%8B.html" \l "content / program)**

* [【块】](file:///C:\\Users\\ylzy2\\AppData\\Local\\Temp\\Rar$EXa27340.11383\\%E5%9B%9E%E9%9B%AA%E6%95%99%E7%A8%8B.html" \l "【块】)
* [【一天】](file:///C:\\Users\\ylzy2\\AppData\\Local\\Temp\\Rar$EXa27340.11383\\%E5%9B%9E%E9%9B%AA%E6%95%99%E7%A8%8B.html" \l "【一天】)
* [【隐藏】](file:///C:\\Users\\ylzy2\\AppData\\Local\\Temp\\Rar$EXa27340.11383\\%E5%9B%9E%E9%9B%AA%E6%95%99%E7%A8%8B.html" \l "【隐藏】)
* [【计算】](file:///C:\\Users\\ylzy2\\AppData\\Local\\Temp\\Rar$EXa27340.11383\\%E5%9B%9E%E9%9B%AA%E6%95%99%E7%A8%8B.html" \l "【计算】)
* [【变量】](file:///C:\\Users\\ylzy2\\AppData\\Local\\Temp\\Rar$EXa27340.11383\\%E5%9B%9E%E9%9B%AA%E6%95%99%E7%A8%8B.html" \l "【变量】)
* [【常量】](file:///C:\\Users\\ylzy2\\AppData\\Local\\Temp\\Rar$EXa27340.11383\\%E5%9B%9E%E9%9B%AA%E6%95%99%E7%A8%8B.html" \l "【常量】)
* [【换行】](file:///C:\\Users\\ylzy2\\AppData\\Local\\Temp\\Rar$EXa27340.11383\\%E5%9B%9E%E9%9B%AA%E6%95%99%E7%A8%8B.html" \l "【换行】)
* [【随取】](file:///C:\\Users\\ylzy2\\AppData\\Local\\Temp\\Rar$EXa27340.11383\\%E5%9B%9E%E9%9B%AA%E6%95%99%E7%A8%8B.html" \l "【随取】)
* [【比较】](file:///C:\\Users\\ylzy2\\AppData\\Local\\Temp\\Rar$EXa27340.11383\\%E5%9B%9E%E9%9B%AA%E6%95%99%E7%A8%8B.html" \l "【比较】)
* [【判断】](file:///C:\\Users\\ylzy2\\AppData\\Local\\Temp\\Rar$EXa27340.11383\\%E5%9B%9E%E9%9B%AA%E6%95%99%E7%A8%8B.html" \l "【判断】)
* [【判空】](file:///C:\\Users\\ylzy2\\AppData\\Local\\Temp\\Rar$EXa27340.11383\\%E5%9B%9E%E9%9B%AA%E6%95%99%E7%A8%8B.html" \l "【判空】)
* [【返回】](file:///C:\\Users\\ylzy2\\AppData\\Local\\Temp\\Rar$EXa27340.11383\\%E5%9B%9E%E9%9B%AA%E6%95%99%E7%A8%8B.html" \l "【返回】)
* [【分割随取】](file:///C:\\Users\\ylzy2\\AppData\\Local\\Temp\\Rar$EXa27340.11383\\%E5%9B%9E%E9%9B%AA%E6%95%99%E7%A8%8B.html" \l "【分割随取】)
* [【读项目】](file:///C:\\Users\\ylzy2\\AppData\\Local\\Temp\\Rar$EXa27340.11383\\%E5%9B%9E%E9%9B%AA%E6%95%99%E7%A8%8B.html" \l "【读项目】)
* [【赋值变量】](file:///C:\\Users\\ylzy2\\AppData\\Local\\Temp\\Rar$EXa27340.11383\\%E5%9B%9E%E9%9B%AA%E6%95%99%E7%A8%8B.html" \l "【赋值变量】)
* [【执行变量】](file:///C:\\Users\\ylzy2\\AppData\\Local\\Temp\\Rar$EXa27340.11383\\%E5%9B%9E%E9%9B%AA%E6%95%99%E7%A8%8B.html" \l "【执行变量】)
* [【执行脚本】](file:///C:\\Users\\ylzy2\\AppData\\Local\\Temp\\Rar$EXa27340.11383\\%E5%9B%9E%E9%9B%AA%E6%95%99%E7%A8%8B.html" \l "【执行脚本】)
* [【发送者QQ】](file:///C:\\Users\\ylzy2\\AppData\\Local\\Temp\\Rar$EXa27340.11383\\%E5%9B%9E%E9%9B%AA%E6%95%99%E7%A8%8B.html" \l "【发送者QQ】)
* [【随机数】](file:///C:\\Users\\ylzy2\\AppData\\Local\\Temp\\Rar$EXa27340.11383\\%E5%9B%9E%E9%9B%AA%E6%95%99%E7%A8%8B.html" \l "【随机数】)
* [【当前群号】](file:///C:\\Users\\ylzy2\\AppData\\Local\\Temp\\Rar$EXa27340.11383\\%E5%9B%9E%E9%9B%AA%E6%95%99%E7%A8%8B.html" \l "【当前群号】)
* [【现行日期】](file:///C:\\Users\\ylzy2\\AppData\\Local\\Temp\\Rar$EXa27340.11383\\%E5%9B%9E%E9%9B%AA%E6%95%99%E7%A8%8B.html" \l "【现行日期】)
* [【10位时间戳】](file:///C:\\Users\\ylzy2\\AppData\\Local\\Temp\\Rar$EXa27340.11383\\%E5%9B%9E%E9%9B%AA%E6%95%99%E7%A8%8B.html" \l "【10位时间戳】)
* [【设置固态变量】](file:///C:\\Users\\ylzy2\\AppData\\Local\\Temp\\Rar$EXa27340.11383\\%E5%9B%9E%E9%9B%AA%E6%95%99%E7%A8%8B.html" \l "【设置固态变量】)
* [【设置变量针对个人】](file:///C:\\Users\\ylzy2\\AppData\\Local\\Temp\\Rar$EXa27340.11383\\%E5%9B%9E%E9%9B%AA%E6%95%99%E7%A8%8B.html" \l "【设置变量针对个人】)
* [【设置变量针对群】](file:///C:\\Users\\ylzy2\\AppData\\Local\\Temp\\Rar$EXa27340.11383\\%E5%9B%9E%E9%9B%AA%E6%95%99%E7%A8%8B.html" \l "【设置变量针对群】)
* [【设置变量有效期】](file:///C:\\Users\\ylzy2\\AppData\\Local\\Temp\\Rar$EXa27340.11383\\%E5%9B%9E%E9%9B%AA%E6%95%99%E7%A8%8B.html" \l "【设置变量有效期】)
* [【正则匹配到的】](file:///C:\\Users\\ylzy2\\AppData\\Local\\Temp\\Rar$EXa27340.11383\\%E5%9B%9E%E9%9B%AA%E6%95%99%E7%A8%8B.html" \l "【正则匹配到的】)
* [【骰点计算】](file:///C:\\Users\\ylzy2\\AppData\\Local\\Temp\\Rar$EXa27340.11383\\%E5%9B%9E%E9%9B%AA%E6%95%99%E7%A8%8B.html" \l "【骰点计算】)
* [【详细骰点计算】](file:///C:\\Users\\ylzy2\\AppData\\Local\\Temp\\Rar$EXa27340.11383\\%E5%9B%9E%E9%9B%AA%E6%95%99%E7%A8%8B.html" \l "【详细骰点计算】)

**[JavaScript](file:///C:\\Users\\ylzy2\\AppData\\Local\\Temp\\Rar$EXa27340.11383\\%E5%9B%9E%E9%9B%AA%E6%95%99%E7%A8%8B.html" \l "JavaScript)**

* [参数传递](file:///C:\\Users\\ylzy2\\AppData\\Local\\Temp\\Rar$EXa27340.11383\\%E5%9B%9E%E9%9B%AA%E6%95%99%E7%A8%8B.html" \l "参数传递)
* [JS输出](file:///C:\\Users\\ylzy2\\AppData\\Local\\Temp\\Rar$EXa27340.11383\\%E5%9B%9E%E9%9B%AA%E6%95%99%E7%A8%8B.html" \l "JS输出)
* [JS的Lib库](file:///C:\\Users\\ylzy2\\AppData\\Local\\Temp\\Rar$EXa27340.11383\\%E5%9B%9E%E9%9B%AA%E6%95%99%E7%A8%8B.html" \l "JS的Lib库)
* [JS对象](file:///C:\\Users\\ylzy2\\AppData\\Local\\Temp\\Rar$EXa27340.11383\\%E5%9B%9E%E9%9B%AA%E6%95%99%E7%A8%8B.html" \l "JS对象)
* [JSON](file:///C:\\Users\\ylzy2\\AppData\\Local\\Temp\\Rar$EXa27340.11383\\%E5%9B%9E%E9%9B%AA%E6%95%99%E7%A8%8B.html" \l "JSON)

**[插件Bug自查指南](file:///C:\\Users\\ylzy2\\AppData\\Local\\Temp\\Rar$EXa27340.11383\\%E5%9B%9E%E9%9B%AA%E6%95%99%E7%A8%8B.html" \l "插件Bug自查指南)**

**[拓展技巧](file:///C:\\Users\\ylzy2\\AppData\\Local\\Temp\\Rar$EXa27340.11383\\%E5%9B%9E%E9%9B%AA%E6%95%99%E7%A8%8B.html" \l "拓展技巧)**

**[鸣谢名单](file:///C:\\Users\\ylzy2\\AppData\\Local\\Temp\\Rar$EXa27340.11383\\%E5%9B%9E%E9%9B%AA%E6%95%99%E7%A8%8B.html" \l "鸣谢名单)**

**[[auto]]**

**[[auto]]是由关键词触发的内容块，分为关键词、执行内容两部分。 第一部分是下文的关键词匹配。**

**全词匹配**

即完全匹配，符号的全角半角也不得有差别。

你可以在星骰APP的设置界面里找到“匹配词回复”一项，这即为全词匹配。

[[auto]]

keywordFull="关键词"

content="回复词"

输入=关键词

输出=回复词

**模糊词匹配**

即不需要消息与关键词完全相同，只需要消息中完全包含关键词，且为连续、有序的。

连续的：

模糊词匹配“竹子”

“后院长了竹子”可以匹配

“竹林地上有叶子”不可以匹配

有序的：

模糊词匹配“球拍”

“给我一个球拍”可以匹配

“我要拍球”不可以匹配

[[auto]]

keywordContained="模糊关键词"

content="回复词"

输入=模糊关键词

输出=回复词

输入=测试模糊关键词

输出=回复词

输入=关键模糊词

输出= //此时无输出，因为并不符合Contained的要求

**正则匹配**

即使用正则表达式进行匹配，并可以捕获匹配内容作为输出的一部分或参与下一个部分的运行等。

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

[[auto]]

keywordRegexp="^复读(.\*)$"

program="【正则匹配到的0】！" //在content/program的代码部分会提到【正则匹配到的】使用方法

输入=复读早安

输出=早安！

输入=骰娘给我复读早安

输出= //由于^的特殊性，只会匹配以复读开头的语句，因此无输出

**多匹配词**

即多个匹配词触发同一个回复词。

[[auto]]

keywordFull=["关键词1","关键词2","关键词n"] //除引号内的内容外，均使用英文半角符号书写

content="回复词"

**[[define]]**

与变量对应，常量就是“不变量”，只能通过直接修改插件来修改的*不可变量*。常量本身只有储存字符串内容的功能。 特殊的，[[define]]下的内容块必须为content，这意味着单独运行不会解析执行任何代码。即使将content改成program也无济于事，反而会使程序识别错误。

[[define]]

name="常量名"

content="""

常量内容

"""

**content / program**

**content和program是[[auto]]下的执行内容。特别的，content会将其中的内容原封不动地输出，而program会执行书写正确的代码。**

而content与program需要用引号包裹住内容。当代码只有一行时，使用一对英文引号即可。当代码需要多行时，则需要三对引号来包裹代码。

[[auto]]

keywordxxx="关键词"

content="回复词"

[[auto]]

keywordxxx="关键词"

program="""

【赋值变量 a,【一天】】

【计算 【变量 a】-10000】

【变量 a】

"""

**【块】**

空值。在格式化你的代码时有空格或许会令代码观感不好、难以debug，用【块】来替代空的位置吧。

输入=【块】

输出= //此时的输出为空，如单独显示则为无回复

输入=早上好【块】晚上好

输出=早上好晚上好 //【块】并不是一个空格

**【一天】**

若通过program输出为86400，即一天的秒数。

输入=【一天】

输出=86400

**【隐藏】**

执行其中的代码，但取消隐藏内的所有输出。

输入=【隐藏 【一天】】

输出= //此时的输出为空，因为隐藏了所有输出。如单独显示则为无回复

输入=【一天】

输出=86400 //没有【隐藏】的包裹后，输出结果为一天的秒数

**【计算】**

进行四则运算，可包含括号。

输入=【计算 3\*5+2】

输出=17

特殊的，如果你输入了一串非纯数字的字符并且出现了不同级的运算，将会导致未知的计算错误。

输入=【计算 a1\*b2\*3c】

输出=6 //此时的计算代买会挑选出符合条件的数字并进行计算

输入=【计算 a1+b2\*3c2】

输出=8 //此时的计算便会出现未知错误

**【变量】**

存储内容并输出，如果没有此变量名则不输出，等效于【块】。

输入=【变量 a】

输出= //在没有赋值的情况下，输出为空，等效于【块】

输入=【赋值变量 a,喵喵】【变量 a】

输出=喵喵 //有了内容的变量再声明就会有输出

**【常量】**

调用同名[[define]]模块的内容。

[[define]]

name="调用"

content="喵喵喵"

输入=【常量 调用】

输出=喵喵喵

而执行define下的内容可以搭配【执行脚本】或【执行变量】使用。

[[define]]

name="test"

content="""

【赋值变量 a,9】【计算 【变量 a】+1】

"""

输入=【赋值变量 测试,【常量 test】】【执行变量 测试】

输出=10

**【换行】**

等效于\n和Enter回车。

输入=早【换行】安

输出=早

安

**【随取】**

取其中的随机内容并输出。分割符号为英文半角逗号。

输入=【随取 早上好,中午好,晚上好,你好】

输出=你好

输入=【随取 早上好,中午好,晚上好,你好】

输出=早上好

输入=【随取 早上好,中午好,晚上好,你好】

输出=晚上好

**【比较】**

比较其中的参数大小（参数应为数字或计算表达式），并根据比较结果执行内容。

【比较 a,b,参数1,参数2】

其中a≥b则会输出参数1，a＜b会输出参数2。

【比较 a,b,

若a≥b返回该语句

,

若a＜b返回该语句

】

输入=【比较 【变量 a】,1,a大于1！,a小于1！】

输出=a小于1！ //由于此时变量a没有被赋值，内容为空。在计算中等同于0

特殊的，一切等效于块或非数字类型的内容被视作0。如该字符串内包含数字，参考【计算】一栏的特殊运算。

**【判断】**

判断参数是否相同，并根据判断结果执行内容。

【判断 a,b,参数1,参数2】

其中a≠b会输出参数1，a=b会输出参数2.

【判断 a,b,

若a≠b返回该语句

,

若a＝b返回该语句

】

输入=【判断 九九,九九,九九不是九九是什么？,九敏！】

输出=九敏！

**【判空】**

判断参数1是否为空，不为空输出该参数1，为空输出参数2.

输入=【判空 【变量 a】,变量a为空值！】

输出=变量a为空值！

输入=【赋值变量 a,该吃饭了】【判空 【变量 a】,变量a为空值！】

输出=该吃饭了

**【返回】**

取消既有输出，返回指定内容并停止该内容块的执行（即，若返回，后续内容不运行）。可类比return。

输入=123【返回 456】789

输出=123456

**【分割随取】**

利用自定义的分割符进行分割，等效于随取。

输入=【分割随取 分割符,1分割符2分割符3】

输出=1

输入=【分割随取 \*,内容1\*内容2\*内容3】

输出=内容2

**【读项目】**

读取项目中的文本内容并直接输出，不会运行其中的代码。

good="白嫖打咩！"

输入=【读项目 good】

输出=白嫖打咩！

输入=【读项目 略略略】

输出= //在读取一个并未被赋值甚至不存在的项目时，返回空值

today="【现行日期】"

输入=今天是【赋值变量 代码,【读项目 today】】【执行变量 代码】

输出=今天是2022\_12\_26 //读项目不会执行其中的代码，但搭配【执行变量】就可以解析并执行其中的代码

特殊的，program和keywordxxx也可以被读取，但若该keyword项目为多匹配词模式，则会输出空值

[[auto]]

keywordFull="测试"

program="喵喵喵"

输入=【读项目 keywordFull】

输出=测试

输入=【读项目 program】

输出=喵喵喵

[[auto]]

keywordFull=["测试","调试"]

program="喵喵喵"

输入=【读项目 keywordFull】

输出= //此时返回为空值，所以不会有输出

**【赋值变量】**

给一个变量赋予内容，可以是任何类型的内容。在赋值之前，这个变量的内容为空，等效于【块】。

输入=【变量 a】

输出= //此时的变量等效为块，所以没有输出

输入=【赋值变量 a,123】【变量 a】

输出=123

**【执行变量】**

读取并执行变量的内容，等效于program。常与[[define]]搭配使用。

输入=【变量 a】【赋值变量 a,【一天】】【执行变量 a】

输出=86400

**【执行脚本】**

执行既定的js脚本，输入的参数会以v1、v2、v3……的变量的形式传入脚本。

【执行脚本 脚本,参数1,参数2,参数3,……】

Lib类为星骰专用的js类，在JavaScript部分你会看到。

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：99

输入=【执行脚本 v1-v2,99,66】 //js脚本中的函数分割也使用英文半角逗号，这与回雪的传入参数格式冲突。

因此，当使用 含有英文逗号的脚本时，可以将脚本储存为常量或项目，通 过【常量】或【读项目】调用

输出=33

**【发送者QQ】**

获取触发这个[[auto]]内容块的QQ号。

输入=【发送者QQ】

输出=1234567 //此时输出应为你的QQ号

**【当前群号】**

获取触发这个[[auto]]内容块的所在群群号。若在私聊触发则显示为空值，等效于【块】。

//在群里执行该代码时

输入=【当前群号】

输出=1234566 //此时输出应为所在群的群号

//在私聊里执行该代码时

输入=【当前群号】

输出= //此时输出为空值

**【随机数】**

随机取一个最大值与最小值范围内的数。特殊的，随机数的最大与最小值可以是一个变量。

输入=【随机数 1,100】

输出=23

输入=【赋值变量 a,20】【随机数 1,【变量 a】】

输出=15

**【现行日期】**

获取触发骰子所在设备的日期，输出格式为“年*月*日”。

【现行日期-年】获取年份数字

【现行日期-月】获取月份数字

【现行日期-日】获取日期数字

输入=【现行日期】\n【现行日期-年】\n【现行日期-月】\n【现行日期-日】

输出=2022\_12\_26

2022

12

26

**【10位时间戳】**

获取触发骰子从1970年1月1日0点开始到触发时间所经过的秒数。

输入=【10位时间戳】

输出=1672021551

**【设置固态变量】**

即将这个变量长期储存，并非临时性的。这将通过文件操作来实现。

输入=【设置固态变量 长期】【赋值变量 长期,123456】【变量 长期】

//2 thousands years later

输出=123456

**【设置变量针对个人】**

将变量操作设置为针对触发者，就像打标签一样。这可以使得这个同名变量储存的内容在不同人触发时呈现不一样的结果。可类比QQ号。

输入=【设置固态变量 好感度】【设置变量针对个人 好感度】【赋值变量 好感度,【随机数 1,10】】【变量 好感度】

//当九九触发时

输出=9

//当小無触发时

输出=1

**【设置变量针对群】**

将变量操作设置为针对触发群，就像打标签一样。这可以使得这个同名变量储存的内容在不同的群里触发时呈现不一样的结果。可类比QQ群号。

输入=【设置固态变量 群】【设置变量针对群 群】【赋值变量 群,【当前群号】】

//在AD钙奶插件交流群触发时

输出=703223775

//在星骰的研究群触发时

输出=584605084

**【设置变量有效期】**

单位为秒，到期后该变量为空值，并在下一次被触发时清除。

输入=【设置固态变量 a】【赋值变量 a,123456】【设置变量有效期 60】【变量 a】

//在10s时触发

输出=123456

//在100s时触发

输出= //此时输出为空

**【正则匹配到的】**

与正则匹配搭配使用，详细运用可见示例。

【正则匹配到的0】会获取正则匹配中顺次到的第一个结果

【正则匹配到的1】会获取正则匹配中顺次到的第二个结果

正则匹配到的（）原则上可以写到百位以上，但你真的需要吗？

[[auto]]

keywordRegexp="^复读(.\*)"

program="【正则匹配到的0】！"

输入=复读早安

输出=早安！

**【骰点计算】**

骰娘的本职，等效于.r。

注：骰点计算式需为 **xdy** 或 **xDy** 的格式。

输入=【骰点计算 3d6\*5】

输出=55

**【详细骰点计算】**

类比骰点计算，但是输出计算过程。

输入=【详细骰点计算 1d20+3d6\*5】

输出=11+{1+5+1}(7)\*5=11+35=46

**JavaScript**

请确保你拥有基础的JavaScript知识。 如果你无法理解 "var x = 1" ，请去[菜鸟教程-JavaScript变量](https://www.runoob.com/js/js-variables.html)

**参数传递**

回雪向js传递参数的数量没有限制，但参数类型均为字符串。因此当你需要从回雪传递一个数字参数（例如十位时间戳）并需要进行数字运算时，需要使用Number()与String()转换参数类型。

不过，js有更为方便获取十位时间戳的方法。

[[define]]

name="脚本"

content="""

v1 //这里是你的js脚本，v1和v2是传的参数1和参数2

v2

"""

[[define]]

name="模块"

content="""

【赋值变量 句子,【执行脚本 【常量 脚本】,【参数1】【参数2】】】

【执行变量 句子】

"""

[[auto]]

keywordxxx=""

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

当两个参数的类型为Number（数字）时，这两个参数相加便相当于加法计算，最后的结果为2.

当两个参数的类型为String（字符串）时，这两个参数相加便会像积木一样摆在一起。如“1”+“1”最后的结果为“11”；“早上”+“好”最后的结果为“早上好”。

**JS输出**

为了代码的可读性与完整性，建议将JS脚本包装为一个函数：

function cat(){

var word = "Hello world!";

return word;

}

cat();

执行这个脚本后会输出Hello world!

这里我们通过function函数名的方式构建了一个函数，因此需要用return来返回所需的结果。

当然，仅构建函数还不够，我们需要执行函数，也就是最后一行cat()。

而上例中的变量cat是String字符串类型，因此你可以通过cat += "新字符串"来增加你输出的内容。

function cat(){

var word = "Hello world!";

word += "Good morning!";

return word;

}

cat();

此时输出就会变成Hello world!Good morning!

但这样并不好看，因此需要在word里加入换行符。

有如下三种方式可以选择：

1 "【换行】"

很明显，这是一个回雪函数。因此如果想要使用这种方式，这个脚本的执行结果必须能够回归到回雪中被执行。

2 "\n"

在字符串中，\n就是换行符的意思，这是通用的写法。

3 String.fromCharCode(10)

这也是一种换行方法，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。因此使用该方式时不需要双引号包裹代码。

**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.sender//相当于【发送者QQ】

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")//roll点,返回数字

例:Lib.rollDetail("1d3" + "+2" + "+1d5")返回:1d3+2+1d5=3+2+1=5+1=6

Lib.getConfig("keywordxx")//获取该[[auto]]下的配置节，相当于【读项目 keywordxx】

Lib.getConstant("常量名")//获取[[define]]下的常量，相当于【常量 xx】

Lib.draw("牌堆关键词")//抽取牌堆

Lib.jrrp("盐")//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

//这种发送方式为临时会话，很容易导致骰娘暴毙

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

\*\*高危慎用\*\*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

**JS对象**

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，可以包含若干属性和方法。

通过赋值语句，我们可以在声明一个空对象的基础上添加属性与方法。

如：向obj这个对象中添加一个值为meow的cat。

obj.cat = meow;

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，分别为name、model和color，值为"Fiat"、500和"white"。

那么如何读取对象中的属性呢？

如下：

var model = car.model

var model = car["model"]

//两者没有区别

这样model就被赋值为500.

书接上回，对象还可以包含方法。那声明是方法呢？

对象方法的本质上是一个函数，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()

//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"}'

**JSON**

有对象已经很好了，但回雪只能储存字符串，那怎么办？

以下两个函数是回雪的救星

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。

JSON.parse()函数可以将字符串类型的JSON转变成对象。

注：很多api的返回都是JSON格式

再注：有些api会返回的JSON中下毒*（此处为弥老师的形象比喻）*

//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?

//那个点是一个看不见的空白符,感谢Typora让阿弥老师看到这个东西。九没钱，所以没有Typora，所以看不到（悲）

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]

//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

**插件Bug自查指南**

1. 文件是否为UTF-8代码
2. 【】是否前后对应
3. 分割符是否为英文半角
4. 英文单词拼写正误
5. content/program内容块的引号是否对应
6. 回雪代码的单词拼写正误
7. 插件是否放在custom文件夹
8. 载入插件后重启骰娘了吗（.system load）

**拓展技巧**

跳转<回雪拓展档案>进行学习。在此之前，请确保你对本文档内的内容掌握程度较高，并且能够自行编写基本的回雪插件。*如：关键词回复、好感度插件*

**鸣谢名单**

**原教程编写者们：**

NAO

弥

联的冮山

竹林

Sern·Spinix

**鸣谢名单的鸣谢名单：**

赵喵喵V2.0

蛇🐍

恒萧·魔骨

羊山羊

**真的鸣谢名单：**

阿弑

蛇🐍

联的冮山

青白

弥

NAO

豆