

# QRDicPro 变量大全

安♡玖特别版

作者： An!♡ 阿玖

如何在网页内进行搜索？

智能手机：  
一般是 菜单 → 页内查找 或者 菜单 → 工具箱 → 页内查找；  
你也可以使用 [百度](#) 搜索你所使用的浏览器的页内搜索方法。

Windows： `Ctrl + F`

macOS： `⌘command + F`

[[Time]] 新增&修改： [更新日志](#)

## 目录

### QRDicPro 变量大全安♡玖特别版

目录	
变量	
发言者相关变量	
%QQ%	
%昵称%	
%Title%	
发言内容相关变量	
%群号%	
%GroupName%	
%参数1%	
%括号1%	
%参数-1%	
%括号0%	
%撤回%	
%AT0%	
%IMG0%	
%FACE0%	
%FACENEW0%	
%FACEPRO0%	
%BUB0%	
%JSON0%	
%XML0%	
%FIMG0%	

%ATNUM%  
%IMGNUM%  
%FACENUM%  
%FACENEWNUM%  
%FACEPRONUM%  
%BUB0%  
%JSON0%  
%JSONNUM%  
%XML0%  
%XMLNUM%  
%FIMG0%  
%FIMGNUM%  
%JSONNUM%  
%XMLNUM%  
%FIMGNUM%

#### 系统变量

%主人%  
%管理员%  
%Robot%  
%RobotAuth%  
%Time%  
%NDTime%  
%RobotRunTime%

#### 其他变量

%Msgbar%  
%Status%  
%Adminuin%  
%Adminame%  
%Inviteename%  
%Skey%  
%Json%  
%Value%  
%Type%  
%代码运行时间%  
%Reqid%

#### 函数

##### 消息内容操作函数

±at %QQ%±  
±at 0±  
±rep %Msgbar%±±strmsg %参数-1%±  
±anony 匿名±  
±face 187±  
±facenew 277±  
±facepro 4194325±  
±bub 1± 授权专用  
±fmsg {"a": "com.tencent.randomwords", "resid": \$随机数 2000-2004\$, "m": "main", "prompt": "闪字"}±

## 文件操作函数

\$写 教程/%群号%/测试 m 0\$  
\$读 教程/%群号%/测试 m 0\$  
\$写文件 /sdcard/QR/example 内容\$  
\$读文件 /sdcard/QR/example\$  
\$下载 /sdcard/QR/example https://www.baidu.com\$  
\$下载 /sdcard/QR/example https://www.baidu.com {JSON格式请求头} A\$  
\$删除 /sdcard/QR/example\$

## 文本处理函数

\$替换 文本 被替换内容 替换后内容\$  
\$正则 文本 正则表达式 替换后内容\$  
\$字符串 replaceFirst 文本 正则表达式 替换后内容\$  
\$取中间 文本 开头 结尾\$  
\$取括号 内容 正则表达式\$  
\$转大写 %括号1%\$  
\$转小写 %括号1%\$  
\$字符串 length 文本\$  
\$字符串 substring 文本 开始位置\$  
\$字符串 substring 文本 开始位置 结束位置\$  
\$字符串 indexOf 文本 查找内容\$  
\$字符串 lastIndexOf 文本 查找内容\$  
\$概率随机 概率数组 内容数组\$

## 消息发送函数

\$发送\$  
\$发送 群 %群号%\$  
\$发送 好友 %QQ%\$  
\$发送 临时 %群号% %QQ%\$  
\$发送 群 msg %群号% 内容\$  
\$发送 临时 msg %群号% %QQ% 内容\$  
±ptt=音频路径或链接±  
\$发送 群 ptt %群号% 音频路径或链接\$  
card:1 xml卡片代码 授权专用  
\$发送 群 xml %群号% xml卡片代码\$ 授权专用  
\$发送 临时 xml %群号% xml卡片代码\$ 授权专用  
json: JSON卡片代码 授权专用  
\$发送 群 json %群号% json卡片代码\$ 授权专用  
\$发送 临时 json %群号% %QQ% json卡片代码\$ 授权专用  
±img=图片路径或链接或MD5±  
\$发送 群 img %群号% 图片路径或链接或MD5\$  
\$发送 临时 img %群号% %QQ% 图片路径或链接或MD5\$ 授权专用  
±fimg=图片路径或链接或MD5±  
\$发送 群 fimg %群号% 图片路径或链接或MD5\$  
±shake 1±

## QQ/QQ群操作函数

\$点赞 %QQ% 10\$  
\$禁 %群号% %QQ% 禁言时间\$

\$改 %群号% %QQ% 新群昵称\$  
\$群头衔 %群号% %QQ% 头衔\$  
\$踢 %群号% %QQ%\$  
\$全体禁言 开 %群号%\$  
\$全体禁言 关 %群号%\$  
\$撤回 %群号% %Msgbar%\$  
\$申请群 申请群号 理由\$ 授权专用  
\$退出群 退出群号\$ 授权专用  
\$获取群状态 %群号%\$  
\$进群审核 %群号% %QQ% 31 12 拒绝理由\$  
\$进群审核 %群号% %QQ% 2001 11 0\$  
\$获取群列表\$  
\$获取群成员 %群号%\$

#### 代码相关函数

\$调用 1000 内容\$  
\$回调 内容\$  
\$常调 内容\$  
\$回调内部 内容\$  
\$执行 %括号1%\$  
\$延迟执行 1000 %括号1%\$  
\$变量 A 123456\$  
\$全局变量 A Name\$  
\$取变量 A\$  
:lable  
\$jump :lable\$  
\$输出为 内容\$

#### 类变量函数

\$随机数 %a% %b%\$  
\$群昵称 %群号% %QQ%\$  
\$管理员 %QQ%\$  
\$图片链接 %IMG0%\$

#### 访问函数

\$访问 链接\$  
\$访问 GET 链接\$  
\$访问 SETGET {JSON格式请求头} 链接\$  
\$访问 SETGETFT {JSON格式请求头} A 链接\$  
\$访问 POST 网址 post参数\$  
\$访问 SETPOST {JSON格式请求头} 链接 post参数\$  
\$访问 SETPOSTFT {JSON格式请求头} A 链接 post参数\$

#### 时间函数

\$时间 yyyy-MM-dd\$  
\$时间格式 %NDTime% yyyy-MM-dd\$  
\$格式时间戳 2021-1-1 yyyy-MM-dd\$  
\$时间操作 毫秒级时间戳 操作类型 添加次数\$  
\$时间戳差 初始毫秒级时间戳 终止毫秒级时间戳\$

#### 运算函数

\$字符长度 %括号1%\$  
\$字节长度 %括号1%\$  
\$URLEncoder %括号1%\$  
\$URLDecoder %括号1%\$  
\$HexEncoder %括号1%\$  
\$HexDecoder %括号1%\$  
\$UnicodeEncoder %括号1%\$  
\$UnicodeDecoder %括号1%\$  
\$Base64Encoder %括号1%\$  
\$Base64Decoder %括号1%\$  
\$MD5 内容\$  
\$CRC32 内容\$  
\$GTK %Skey%\$  
\$BKN %Skey%\$  
\$SIG 内容\$  
\$随机字符 内容 长度\$  
\$数字转中文 数字\$

#### JSON 操作函数

@A[1]  
\$JSON 添加 A 1 100\$  
\$JSON 添加 A 100\$  
\$JSON 删除 A 1\$  
\$JSON 长度 A\$  
\$JSON 获取 A 1\$  
\$JSON 包含 A 100\$  
\$JSON 包含 A value 100\$  
\$JSON 包含 A key 100\$

#### 外部Java调用

\$BSH xxx.java %a%\$  
\$BSH xxx.java %a% %b%\$

#### 其他函数

\$设置群状态 %群号% 开\$  
\$设置群状态 %群号% 关\$  
\$添加管理员 %QQ%\$  
\$删除管理员 %QQ%\$  
\$词库操作 添加 词库文件名\$  
\$词库操作 删除 词库文件名\$

#### 画布

\$画布 声明 A 长 宽\$  
\$画布 颜色 A 16进制颜色码\$  
\$画布 文字 A 文字 X Y 16进制颜色码 size 字体文件路径（可选）\$  
\$画布 文字换行 A 文字 自动换行长度 X Y 16进制颜色码 size 字体文件路径（可选）\$  
\$图片参数 图片的绝对路径\$  
\$画布 图片 A 图片绝对路径 X Y W H X1 Y1 X2 Y2\$  
\$画布 保存 A 图片保存绝对路径\$

#### 转义字符

.....

## 特殊关键词前缀

### 特殊事件

[系统]

[退群]

[撤回]xxx 授权专用

[红包]

[戳一戳]

[上下管理]

### 消息来源

[群]xxx

[好友]xxx

[临时]xxx

### 其他前缀

[内部]xxx

[初始化]

[群初始化]

[总处理]

## 运算符

### 数学运算符

算术运算格式

基础运算符

其他运算符

高级运算模式

舍入办法

### 逻辑运算符与条件语句

关系运算符

逻辑运算符

条件语句关键字

如果:

正则:

如果尾

正则尾

### 其他关键字

:

±

### 其他知识

#### 正则表达式基础

简介

元字符

匹配某些字符

数量限制

正则表达式触发词实例

转义符 \

完整的正则表达式教程

#### QR 中的 JSON

关于 JSON

JSON 的替代方案

JSON 操作代码示例

[\\$JSON 添加 A 1\\$](#)

[\\$JSON 添加 A 1 1\\$](#)

[\\$JSON 获取 A 1\\$](#)

[\\$JSON 删除 A 1\\$](#)

[\\$JSON 长度 A\\$](#)

QQ 官方 API

[±img=http://q2.qlogo.cn/headimg\\_dl?dst\\_uin=%QQ%&spec=5±](#)

[±img=http://p.qlogo.cn/gh/%群号%/%群号%/±](#)

时间变量详解

[基础时间变量](#)

[更多时间变量](#)

未整理&杂项

[红包/转账相关](#)

[转账检测](#)

[查红包](#)

[卡片代码示例](#)

[JSON 卡片](#)

[XML 卡片](#)

废弃&失效&不明意义部分

[±xingtype=%随机数40000-40005%±](#)

[%Number%](#)

[%Pid%](#)

[%Strmsg%](#)

[±nick-测试昵称±](#)

[±title-测试±](#)

[抢红包](#)

[发红包](#)

关于

[关于变量大全编写](#)

[作者](#)

[编辑工具](#)

[关于共享](#)

[许可协议](#) □

想要制作 [变量大全安♡玖特别版](#) 的网页镜像?

想要以文件形式分享 [变量大全安♡玖特别版](#) ?

---

## 变量

---

# 发言者相关变量

## %QQ%

获取发言者的 QQ 号。

备选方案： %Code% %Uin%

所有匿名用户的发言， %QQ% 为 80000000。

其中 %Code% 可以获取部分特殊触发QQ

## %昵称%

获取发言者的 群昵称。

备选方案： %UinName%

要获取真正的「昵称」， 可以参考 [\\$群昵称 \\$](#) 函数。

## %Title%

获取群内 专属头衔。

无专属头衔时返回空值。

# 发言内容相关变量

## %群号%

获取本群的 群号。

备选方案： %群% %Groupid%

临时会话中， %群号% 为本群群号；

好友会话中， %群号% 为 0。

## %GroupName%

获取本群的 群名。

在 [系统] [撤回] 与 [上下管理] 前缀的代码中无法正常获取。



## %参数1%

触发词中第一个空格以后，第二个空格或结尾以前的内容。

%参数2 %参数3% 以此类推。

自从有子 [%括号1%](#)，谁还用 [%参数1%](#) 啊

---

## %括号1%

触发词中，从左到右第一对括号中的内容。括号顺序以前半括号 ( 的出现顺序为准。

%括号2 %括号3% 以此类推。

你可以尝试以下代码。

```
(测试(测测)和(试试))
```

```
括号1为: %括号1%\n
```

```
括号2为: %括号2%\n
```

```
括号3为: %括号3%
```

发送 `测试测测和试试` 以触发。

想要具体了解？请学习 [正则表达式-捕获组](#)。

---

## %参数-1%

## %括号0%

获取发言的全部文本内容。

---

## %撤回%

获取撤回的全部文本内容

---

## %AT0%

获取被 @ 的第一个成员的 QQ 号。

第二个成员的 QQ 号为 [%AT1%](#)，以此类推。

获取不到 [%AT0%](#)，[%AT1%](#) ...时，对应变量值为0；

@全体成员时，[%AT0%](#)，[%AT1%](#) ...均为0。

---

## %IMG0%

## %IMG0%

获取「图片代码<sup>①</sup>」，可以在发图时使用。

第二张图片的代码为 `%IMG1%`，以此类推。

`%IMG0%` 的核心是一串 32 位的大写字母 + 数字，即图片的 MD5<sup>1</sup> 码。

手机、电脑；群聊、私聊发图得到的 `%IMG0%` 有一定差异。

因此，想要使用 MD5 对比图片是否一致，可以使用以下方法处理 `%IMG0%`：

\$正则 @ %IMG0%@/.\*-|[\dA-Z]@\$

它可以提取任意「图片代码」中的 MD5 部分。

① 变量大全中指图片 MD5 码经过一定程度修饰后的结果。具体形式可能为：

MD5.图片扩展名

{UUID格式的MD5}.图片扩展名

/发送者QQ号-一串数字-MD5

## %FACE0%

获取消息中第一个普通表情的编号。

第二个普通表情的编号为 `%FACE1%`，以此类推。

只能获取较为早期的 QQ 表情，较晚出现的 QQ 表情（如：`/元宝` `/菜狗` 等）无法获取。

## %FACENEW0%

同上，但只能获取 `%FACE0%` 获取不到的、较晚出现的 QQ 表情。

## %FACEPRO0%

同上，但只能获取表情页无法直接找到的、被称为「会员小表情」的 QQ 表情（如：`/流星` `/猫爪`）。

## %BUB0%

获取发言者 气泡 ID<sup>2</sup>。

## %JSON0%

获取 [JSON 卡片](#) 代码。代码不存在时为空。

## **%XML0%**

获取 [XML 卡片](#) 代码。代码不存在时为空。

---

## **%FIMG0%**

获取闪照的「图片代码」，类似 `%IMG0%`。

---

## **%ATNUM%**

获取消息中 @ 操作的总数量。

---

## **%IMGNUM%**

获取消息中图片的总数量。

---

## **%FACENUM%**

## **%FACENEWNUM%**

## **%FACEPRONUM%**

获取消息中对应种类表情的总数量。

---

## **%BUB0%**

获取发言者 气泡 ID<sup>2</sup>。

---

## **%JSON0%**

获取 [JSON 卡片](#) 代码。代码不存在时为空。

---

## **%JSONNUM%**

## **%XML0%**

获取 [XML 卡片](#) 代码。代码不存在时为空。

---

## **%XML NUM%**

## %FIMG0%

获取修饰后的闪照的 MD5 <sup>1</sup> 码，类似 `%IMG0%`。

## %FIMGNUM%

部分用法示例：

```
[\\s\\S]*
如果:%JSON0%!=
捕获到JSON卡片： %JSON0%
如果尾
如果:%XML0%!=
捕获到XML卡片： %XML0%
如果尾
如果:%FIMG0%!=
捕获到闪照： %FIMG0%
如果尾
```

## %JSONNUM%

## %XMLNUM%

获取消息中对应类型卡片代码的数量。

## %FIMGNUM%

获取消息中闪照的数量。

部分 NUM 变量（如 `%FIMGNUM%`）一般情况下只能为 0 或 1，但是也允许其他值出现。

比如 QR 在一条消息中添加多条闪照内容，其他 QR 机器人就可以得到超过 1 的 `%FIMGNUM%` 并获取未被显示的图片的「图片代码」。

部分用法示例：

```
[\\s\\S]*
如果:%JSON0%!=
捕获到JSON卡片： %JSON0%
如果尾
如果:%XML0%!=
捕获到XML卡片： %XML0%
如果尾
```

2017/10/17

如果:%FIMG0%!=  
捕获到闪照: %FIMG0%  
如果尾

---

## 系统变量

写了一年了我都没给它们找到一个好名字。

---

### %主人%

获取 QR 词库插件中设置的「主人」QQ 号。

---

### %管理员%

获取 QR 词库插件中设置的「管理员」，返回数组形式。

不建议萌新使用，萌新可以参考 [\\$管理员 \\$](#) 函数。

---

### %Robot%

获取当前登录的机器人的 QQ 号。

---

### %RobotAuth%

判断本机器人 QQ 是否在 QR 授权。

`%RobotAuth%` 为 0，说明未授权；

`%RobotAuth%` 为 1，说明已授权。

---

### %Time%

获取当前服务器 UNIX 时间戳<sup>3</sup>。

网络时间，不受系统时间影响。

---

### %NDTime%

## %NDTime%

获取当前系统 毫秒级 时间戳。

本地时间，不联网。因此，修改系统时间会引起 %NDTime% 变化。

## %RobotRunTime%

获取QR主程序本次打开时的 毫秒级 时间戳。

可用于计算连续运行时间。

## 其他变量

### %Msgbar%

消息的群内唯一 ID，每群单独计算，每一条消息的 %Msgbar% 是上一条消息的 %Msgbar% + 1。

### %Status%

消息的状态码。

常用 %Status%

- %Status% == 33 进群；
- %Status% == 34 被邀请的人进群；
- %Status% == 84 申请进群；
- %Status% == 87 被邀请的人申请进群；
- %Status% == 82 群聊消息；
- %Status% == 166 好友消息；
- %Status% == 141 群临时消息。

关于邀请和进群的 %Status% 可能并不准确。

### %Adminuin%

进行群管理操作的管理员的 QQ 号。

### %Adminame%

进行群管理操作的管理员的群昵称。

## 获取退群者昵称

获取机器人 Skey。

需要在 **QR主程序→插件列表→长按词库插件** 弹出的界面勾选 Skey 授权。

获取机器人 Pskey, [JSON](#) 格式。

需要在 **QR主程序→插件列表→长按词库插件** 弹出的界面勾选 Pskey 授权。

在 **[上下管理]** 中使用。

%Value% 为 1 时, %QQ% 被任命群管理员;

%Value% 为 0 时, %QQ% 被取消群管理员。

消息类型。官方插件示例中有详细介绍。

词库中很少用到。

部分反馈类型 0为群, 1为好友, 2为讨论组, 3为系统, 4为临时会话, 19为群撤回。

本触发词的代码从开始运行到现在所经过的时间。

返回一个 58 位小数，但小数部分至多只有小数点后 9 位以内的数字有意义。

因为手机CPU的时钟周期才  $10^{-10}$  数量级。

**%Renid%**

获取特殊消息ID。

如戳一戳/群签到之类的通知。

## 函数

DicPro 的大函数名不再区分大小写。

以 `$jump :lable$` 函数为例，现在你可以写成：

`$Jump :lable$` `$JUMP :lable$` 甚至 `$jUmP :lable$` 而仍然可以正常执行。

但仍建议你使用变量大全中的大小写方式。因为字母作为函数参数（如：`$访问 GET 链接$` 中的 GET）时仍然区分大小写。

函数中的 `%QQ%` `%群号%` 等是作为示例的变量，并非指在参数前后加上 `%`。

如：

✔ `$发送 群 msg %群号% 你好! $`

✔ `$发送 群 msg 12345678 你好! $`

✘ `$发送 群 msg %12345678% 你好! $`

## 消息内容操作函数

此处的函数只是对消息内容进行特殊操作，不会增加消息内容。

因此，这些函数需要搭配文字才能成功发送消息并产生效果。

### ±at %QQ%±

在消息中插入 @ 代码。

关于@：

`±at %QQ%±` 是「@」代码，可以产生「@」效果，即对方会收到「有人@你」黄字通知，但消息中不会显示「@xxxx」文字。

而「@昵称」只会在消息中显示@文本，而没有通知提示。

因此，QR中正确的「@」方式为：

`±at %QQ%±@%昵称%`

特别地，macOS 版本的 QQ 可以看到 @ 代码，因此会看到两次 @。此问题暂时无法解决。

另外，消息内容操作函数 部分的其他函数，macOS QQ 也可以看到空消息。

QR 也能收到这种消息，所以可以用来机器人之间在群里偷偷通讯。虽然没什么大用。



## ±at 0±

@全体成员，同样不显示「@全体成员」的文字。

---

## ±rep %Msgbar%±±strmsg %参数-1%±

「回复」ID 为 %Msgbar% 的消息。

当前版本的 QR 中无法选定消息进行回复，只能回复触发指令的消息。疑似 bug。

±strmsg=%参数-1%± 中， %参数-1% 处为回复框里面的内容，可以自行修改。

±strmsg=%参数-1%± 非必选，若不写则回复框引用内容为空。

若回复内容为非文本类型（图片、文件等），则 ±strmsg=%参数-1%± 无效。

---

## ±anony 匿名±

使本条消息匿名发送。

机器人匿名发送的消息也可以触发它自己。

---

## ±face 187±

发送编号为 187 的普通（以 %FACE0% 获取）表情。

## ±facenew 277±

发送编号为 277 的新（以 %FACENEW0% 获取）表情。

## ±facepro 4194325±

发送编号为 4194325 的会员（以 %FACEPRO0% 获取）小表情。

不需要机器人本身开通 QQ VIP 或 SVIP。

实际上所谓「会员小表情」只在添加表情包时有限制。

---

## ±bub 1± 授权专用

使用气泡 ID <sup>2</sup> 为 1 的气泡发言。

不受 VIP 限制；仅作用于当前一条消息，后续消息仍为原气泡。

仅当新版 QRSPEED 设置中开启「所有消息走长消息」时可用。

谨慎勾选：此项勾选会导致部分消息类型不可用。

---

```
±fmsg {"a":"com.tencent.randomwords","resid":$随机数 2000-2004$,"m":"main","prompt":"闪字"}±
```

将发送的文字转换为闪字。

自然，你可以将其中的 `$随机数 2000-2004$` 改为 2000 - 2004 之间的固定数字，以期得到固定的闪字效果。

## 文件操作函数

### \$写 教程/%群号%/测试 m 0\$

向 `/sdcard/QR/QRDlc/data/教程/%群号%/测试` 中写入 `m=0`。

我们将这里的「m」称为「key」。key 可以为任意数目的任意字符。

一个文件中，可以写入多个 key 不同的数据。

示例：，你可以使用 `$写 教程/%群号%/测试 mass 100$` 和 `$写 教程/%群号%/测试 火锅 100$` 在文件「测试」中写入两个数据。

之后，你可以使用下方的 `$读$` 函数读取这些数据。

### \$读 教程/%群号%/测试 m 0\$

读取 `/sdcard/QR/QRDlc/data/教程/%群号%/测试` 中 m 对应的值，不存在则返回默认值 0。

你可以修改默认值「0」。

若将代码修改成 `$读 教程/%群号%/测试 m 100$`，则对应值不存在时返回默认值「100」。

### \$写文件 /sdcard/QR/example 内容\$

以 UTF-8<sup>4</sup> 编码向文件 `/sdcard/QR/example` 中写入「内容」。

若上层文件夹不存在，则会尝试自动创建文件夹。

### \$读文件 /sdcard/QR/example\$

以 UTF-8 编码读取文件 `/sdcard/QR/example` 中的内容。

与 `$访问 file://路径$` 不同，无法获取文件夹内容。

### \$下载 /sdcard/QR/example https://www.baidu.com\$

将 `$访问 https://www.baidu.com$` 得到的结果下载到 `/sdcard/QR/example`。

示例：

```
$下载 /QR/QRDlc/data/百度.html https://www.baidu.com$
```

```
$下载 /QR/QRDlc/data/backup.txt file:///sdcard//QR/QRDlc/dic.txt$
```

无法自动创建文件夹，需要使用已有文件夹或手动创建路径文件夹才能成功下载文件。

---

## \$下载 /sdcard/QR/example https://www.baidu.com {JSON格式请求头} A\$

与 [上方的函数](#) 类似。可以自定义请求头为 `{JSON格式请求头}`。响应头将存储于变量 `A`。

---

## \$删除 /sdcard/QR/example\$

删除任意文件或文件夹，其路径为 `/sdcard/QR/example`。

**注意！** `$删除 /sdcard/QR/QRDlc/dic.txt$` 或类似操作将可能直接删除你的词库。

请不要在代码中留下被删除词库的隐患。

---

## 文本处理函数

文本处理函数中符号 `%` 可以随意更换，但要保证它不会在参数中出现。

如：`$替换 QR 文本QR原内容QR新内容$`

在所有参数均不含空格，或使用变量而不是文本作为参数的情况下，可以使用类似 `$替换 文本 原内容 新内容$` 的方法，下面不做额外说明。

---

## \$替换 % 文本 %被替换内容 %替换后内容\$

将「文本」中的「被替换内容」全部替换为「替换后内容」。

备选方案：`$字符串 replace 文本 被替换内容 替换后内容$`

---

## \$正则 % 文本 %正则表达式 %替换后内容\$

以 [正则表达式](#) 匹配「文本」中的「正则表达式」，并将匹配结果全部全部替换为「替换后内容」。

备选方案：`$字符串 replaceAll 文本 正则表达式 替换后内容$`

注意：`$正则` 函数中，`[0-9]` 将不会匹配数字 0 - 9 中的任意一个，而是被当作 [计算格式](#) 运算后匹配 `-9`。

---

## \$字符串 replaceFirst 文本 正则表达式 替换后内容\$


以 [正则表达式](#) 匹配「文本」中的「正则表达式」，并将首个匹配结果全部替换为「替换后内容」。

---


## \$取中间 文本 开头 结尾\$

截取「文本」从首个「开头」到「开头」后首个「结尾」的中间部分。

如果要从文本的开头开始取，则参数「开头」留空。

如：\$取中间  12345  3\$ 结果为：12

如果要一直取到文本的结尾，则参数「结尾」留空。

如：\$取中间  12345  3  \$ 结果为：45

---

## \$取括号 内容 正则表达式\$

以 [正则表达式](#) 匹配「内容」。

将匹配成功的内容输出为多维数组

示例：

\$取括号 你好啊猪 你(.\*)啊\$

输出[["你好啊","好"]]

---

## \$转大写 %括号1%\$

将 %括号1% 中的所有英文字母转换为大写字母。

## \$转小写 %括号1%\$

将 %括号1% 中的所有英文字母转换为小写字母。

---

## \$字符串 length 文本\$

获取 %括号1% 的字符数量。

以 UTF-16 BE 编码后的长度。因此，少部分生僻字、特殊符号可能被判断为 2 个字符。

---

## \$字符串 substring 文本 开始位置\$

取「文本」从「开始位置」至最后的部分。包含开始位置的字符。

第一个字符的「位置」是 0，第二个字符的「位置」是 1，以此类推。

这和数组的索引是一样的。「位置」实际上就是字符在字符串中的索引。

## \$字符串 substring 文本 开始位置 结束位置\$

取「文本」从「开始位置」至「结束位置」的部分。包含开始位置的字符但不包含结束位置的字符。

## \$字符串 indexOf 文本 查找内容\$

若 查找内容 存在，返回首个 查找内容 的首个字符的位置；

若 查找内容 不存在，返回 -1。

对位置的定义见 [\\$字符串 substring 文本 开始位置\\$](#) 函数的注释。

## \$字符串 lastIndexOf 文本 查找内容\$

若 查找内容 存在，返回最后一个 查找内容 的首个字符的位置；

若 查找内容 不存在，返回 -1。

## \$概率随机 概率数组 内容数组\$

概率数组的长度必须与内容数组的长度相等，不是字符长度。

示例：

随机

```
$概率随机 [1,2] ["你好","猪"]$
```

## 消息发送函数

### \$发送\$

将上方输出的文本直接发送。

示例：

测试发送

1

23

\$发送\$

456

用于分段发送时可能由于网络波动导致顺序颠倒。

下面六个函数若在同一触发词代码中出现且互相冲突，以最后一个出现的函数为准。

## **\$发送 群 %群号%\$**

本次触发中，将消息发送至群 **%群号%**。

若不指定 **%群号%**（即：**\$发送 群\$**），则自动填充当前 **%群号%**。

## **\$发送 好友 %QQ%\$**

本次触发中，将消息发送至好友 **%QQ%**。

若不指定 **%QQ%**（即：**\$发送 好友\$**），则自动填充触发者 **%QQ%**。

## **\$发送 临时 %群号% %QQ%\$**

本次触发中，将消息作为临时消息发送至群 **%群号%** 中的成员 **%QQ%**。

备选方案：**\$发送 私聊 %群号% %QQ%\$**

若不指定 **%群号%** 与 **%QQ%**（即：**\$发送 临时\$**），则自动填充当前 **%群号%** 与触发者 **%QQ%**。

临时会话权限可能被群主关闭。

## **\$发送 群 msg %群号% 内容\$**

向群 **%群号%** 发送文本消息。

## **\$发送 临时 msg %群号% %QQ% 内容\$**

向群 **%群号%** 的群成员 **%QQ%** 发送文本临时消息。

无法带图。

## **±ptt=音频路径或链接±**

向当前群发送语音消息。

## **\$发送 群 ptt %群号% 音频路径或链接\$**

向指定群发送语音消息。

大于 10MB 的音频文件无法被发送。

无法向单独的某人（好友、临时消息）发送语音消息。

## card:1 xml卡片代码 授权专用

向当前聊天发送 [XML 卡片](#)。

## \$发送 群 xml %群号% xml卡片代码\$ 授权专用

向指定群发送 XML 卡片。

## \$发送 临时 xml %群号% xml卡片代码\$ 授权专用

向指定群成员发送 XML 卡片。

---

## json: JSON卡片代码 授权专用

向当前聊天发送 [JSON 卡片](#)。

## \$发送 群 json %群号% json卡片代码\$ 授权专用

向群 %群号% 发送 JSON 卡片。

## \$发送 临时 json %群号% %QQ% json卡片代码\$ 授权专用

向群 %群号% 中成员 %QQ% 发送 JSON 卡片。

---

## ±img=图片路径或链接或MD5±

向当前群发送图片消息。

可以搭配文字。

私聊发图为 授权专用 功能。

## \$发送 群 img %群号% 图片路径或链接或MD5\$

向指定群发送图片消息。

## \$发送 临时 img %群号% %QQ% 图片路径或链接或MD5\$ 授权专用

向指定群的群成员发送图片临时消息。

---

## ±fimg=图片路径或链接或MD5±

向当前群发送闪照。

## \$发送 群 fimg %群号% 图片路径或链接或MD5\$

向指定群发送闪照。

---

## ±shake 1±

发送窗口抖动。

---

## QQ/QQ群操作函数

---

### \$点赞 %QQ% 10\$

点赞 `%QQ%` 10 次。只能给机器人的好友点赞。

当然，如果你的机器人开通了 SVIP，你可以将 10 改为 20。

---

### \$禁 %群号% %QQ% 禁言时间\$

禁言群成员。机器人需有 QQ 群管理员权限。

禁言时间以 秒 为单位。

禁言时间为 0 时，表示解除禁言。

---

### \$改 %群号% %QQ% 新群昵称\$

修改群成员的群昵称。修改他人群昵称时，机器人需有 QQ 群管理员权限。

即使没有管理员权限，机器人也可以修改自己的群昵称。

---

### \$群头衔 %群号% %QQ% 头衔\$

修改群成员的群头衔，需要有群主权限。

最多19个非汉字字符。

---

### \$踢 %群号% %QQ%\$

移除群成员。机器人需有 QQ 群管理员权限。

---

### \$全体禁言 开 %群号%\$



开启指定群的全员禁言。

**\$全体禁言 关 %群号%\$**

关闭指定群的全员禁言。

---

**\$撤回 %群号% %Msgbar%\$**

撤回 ID 为 %Msgbar% 的消息。

---

**\$申请群 申请群号 理由\$** 授权专用

申请入群。

由于 腾讯风控<sup>5</sup> 的原因，机器人发出的入群申请可能被 屏蔽 或 过滤。

---

**\$退出群 退出群号\$** 授权专用

退出群。

---

**\$获取群状态 %群号%\$**

获取 %群号% 在 QR 主程序中的开关状态，返回 开 或 关。

---

**\$进群审核 %群号% %QQ% 31 12 拒绝理由\$**

拒绝 %QQ% 入群。

很可能仅在 [系统] 前缀的触发词代码中有效。

**\$进群审核 %群号% %QQ% 2001 11 0\$**

同意 %QQ% 入群。

很可能仅在 [系统] 前缀的触发词代码中有效。

---

**\$获取群列表\$**

获取机器人加入的群数组

---

## \$获取群成员 %群号%\$

获取机器人加入的群聊的群成员数组

---

## 代码相关函数

### \$调用 1000 内容\$

延迟 1000 毫秒后，触发关键词「内容」。

示例：

说话

\$调用 1000 说你好\$

\$调用 2000 说再见\$

\$调用 3000 说走开走开\$

[内部]说(.\*)

%括号1%

1 秒 = 1000 毫秒。

---

### \$回调 内容\$

触发关键词「内容」，并将执行结果返回到 `$回调 内容$` 处。

示例：

说话

a:\$回调 说你好\$

我说了个「%a%」 \n

我又说了个「\$回调 说再见\$」 \n

\$回调 说走开走开\$

[内部]说(.\*)

%括号1%

---

### \$常调 内容\$

触发不含 `[内部]` 前缀的关键词「内容」，并将执行结果返回到 `$常调 内容$` 处。

---

## **\$回调内部 内容\$**

触发含有 `[内部]` 前缀的关键词「内容」，并将执行结果返回到 `$回调内部 内容$` 处。

---

## **\$执行 %括号1%\$**

将 `%括号1%` 作为 QRSpeed 词库代码执行。可以执行多行。

建议不要在私聊中执行，否则可能会出现 bug。

如果你需要用到它，记得加上权限或防范措施，以免被恶意执行（如被 [删除词库](#)）。

---

## **\$延迟执行 1000 %括号1%\$**

延迟延迟 1000 毫秒后，将 `%括号1%` 作为 QRSpeed 词库代码执行，并发送执行结果。

---

## **\$变量 A 123456\$**

将变量 A 赋值为 123456，支持多字变量。

示例：

我的金币

`$变量 金币 10000$`

你现在有%金币%枚金币。

## **\$全局变量 A Name\$**

可在这个词库定义全局变量，任何一个指令都可使用。

静态全局，不会随着时间改变，且重启一次词库或主程序就要重新赋值一次，较为麻烦

## **\$取变量 A\$**

获取变量 A 的值。

---

**:lable**

## \$jump :lable\$

`:lable` 指定一个跳行标签。你可以使用 `$jump :lable$` 跳行到对应标签处。

你可以指定多个标签，其中的 lable 可以改为其他字符串。

备选方案： `$跳 :lable$`

示例1：

循环示例

a:0

:here!

如果:%a%<10

%a%\n

a:[%a%+1]

\$jump :here!\$

如果尾

示例2：

跳行示例

1\$jump :lable3\$

2

:lable1

3\$jump :lable2\$

:lable3

4\$jump :lable1\$

:lable2

5

对「示例2」看晕了对吗？请尽量不要在一段代码中使用过多跳行，除非你确认以后不会再更改它。

---

## \$输出为 内容\$

Pro版本的词库进行了高强度转义，想要直接输出链接或回调输出的±类关键词和卡片类关键词，就使用输出为吧

---

## 类变量函数

写了一年了我也没给它们找到一个好名字。

## \$随机数 %a% %b%\$

获取 %a% 至 %b% 之间（包括 %a% 与 %b%）的一个随机数。其中 %a% 不必小于 %b%。

备选方案：\$随机数 %a%-b%\$

若使用 \$随机数 %a%-b%\$，%a% 与 %b% 只能为自然数，否则输出0。

若使用 \$随机数 %a% %b%\$，%a% 与 %b% 只能为整数，否则没有输出。

---

## \$群昵称 %群号% %QQ%\$

获取 %QQ% 在 %群号% 的群名片。

当机器人或 %QQ% 不在所查询群时，返回 %QQ% 的真实昵称。

因此，可以使用 \$群昵称 0 %QQ%\$ 获取某人真实昵称。

---

## \$管理员 %QQ%\$

检测 %QQ% 是否为词库插件中设置的管理员。

若是，返回 %QQ%；

否则，返回 0。

使用示例：

如果:\$管理员 %QQ%\$==%QQ%  
是管理员

---

## \$图片链接 %IMG0%\$

将「图片代码」为 %IMG0% 的图片上传到 QR 服务器并获取链接。

链接获取失败时，返回传入的 %IMG0%。这种情况一般会在你传入的并不是合法的 %IMG0% 格式时出现。

---

## 访问函数

访问结果以 UTF-8<sup>4</sup> 编码。

---

## \$访问 链接\$

### \$访问 GET 链接\$

向链接发送 GET 请求，可以理解为普通的访问链接。

你可以使用 `$访问 file://路径$` 访问本机文件或文件夹，获取文件内容或文件夹内所有文件名。

---

以下的访问方法，如果看不懂可以忽略。

---

### \$访问 SETGET {JSON格式请求头} 链接\$

向链接发送自定义请求头的 GET 请求。

---

### \$访问 SETGETFT {JSON格式请求头} A 链接\$

向链接发送自定义请求头的 GET 请求，并将响应头赋值给变量 A。

---

### \$访问 POST 网址 post参数\$

向链接发送 POST 请求。

---

### \$访问 SETPOST {JSON格式请求头} 链接 post参数\$

向链接发送自定义请求头的 POST 请求。

---

### \$访问 SETPOSTFT {JSON格式请求头} A 链接 post参数\$

向链接发送自定义请求头的 POST 请求，并将响应头赋值给变量 A。

---

## 时间函数

时间格式化函数（`$时间 yyyy-MM-dd$` `$时间格式 %NDTime% yyyy-MM-dd$`）中，`yyyy-MM-dd` 这部分内容被称为格式化字符串。

由于格式化字符串中的许多关键字很少用到且规则复杂，这里只介绍一些简单的时间格式。想具体了解，请点击 [时间格式化字符串](#) 本章

[时间戳转换](#) 页面。

使用示例：

测试time

\$时间 HHmm\$\n

\$时间 yyyy-MM-dd HH:mm:ss.SSS E\$\n

\$时间 时间现在时间ahh时mm分ss秒\$

以下介绍一些常用的关键字。将关键字放入格式化字符串中，就会被替换为对应的时间。

`yyyy` 年

`MM` 月

`dd` 日

`HH` 时：24 小时制

`hh` 时：12 小时制

`mm` 分

`ss` 秒

`E` 星期

`a` 半日（上午/下午）

`$时间 $` 等函数实际上是调用 *Java SimpleDateFormat* 类，处理时间并输出。

由于作者并没有找到足够详尽的 *SimpleDateFormat* 中文介绍，因此在下方按照自己的理解与实践编纂了[详解](#)。

如果你发现了[详解](#)中的疏漏，或者可以找到完整的中文介绍，希望可以联系作者修正或补充链接。

## \$时间 yyyy-MM-dd\$

将当前时间转换为一般时间格式。

具体格式定义方法见下面的函数介绍。

## \$时间格式 %NDTime% yyyy-MM-dd\$

将毫秒级时间戳 `%NDTime%` 转换为一般时间格式。

---

## \$格式时间戳 2021-1-1 yyyy-MM-dd\$

将一般时间格式转换为毫秒级时间戳。

---

## \$时间操作 毫秒级时间戳 操作类型 添加次数\$

操作类型

YEAR=一年

MONTH=一个月

DAY\_OF\_MONTH=一天

HOUR\_OF\_DAY=一小时

MINUTE=一分钟

SECOND=一秒

WEEK\_OF\_YEAR=一周

示例：

\$时间操作 %NDTime% YEAR 1\$

---

## \$时间戳差 初始毫秒级时间戳 终止毫秒级时间戳\$

计算两个毫秒级时间戳的差

---

## 运算函数

使用本部分函数时，如果你需要手动添加参数（如：\$MD5 测试\$），你需要注意参数中的空格，它在许多函数中可能引起错误。

使用变量作为参数（如：\$MD5 %括号1%\$ \$MD5 %a%），则无需担心空格会引起错误。

---

## \$字符长度 %括号1%\$

获取 %括号1% 的字符数量。

以 UTF-16 BE 编码后的长度。因此，少部分生僻字、特殊符号可能被判断为 2 个字符。

---

## \$字节长度 %括号1%\$

获取 %括号1% 的字节数量。

以 UTF-8 以编码后的长度。一个汉字包含 3-4 个字节，而不是某些上古书籍中的 2 个。

---

## \$URLEncoder %括号1%\$



URL 编码：将文本使用 UTF-8 百分号编码方式进行转码。

QR 的 URL 编码中，除了数字 0-9、大小写形式的 26 字母 a-z A-Z、句点 .、减号/连字符 - 以及下划线 \_ 之外，其他符号都会被编码。

## \$URLDecoder %括号1%\$

URL 解码：将 UTF-8 百分号编码转换为文本。

---

## \$HexEncoder %括号1%\$

Hex 编码：将文本使用 UTF-8 十六进制编码方式进行转码。

实际上就是将 所有 字符 URL 编码之后去掉百分号。

## \$HexDecoder %括号1%\$

Hex 解码：将 UTF-8 十六进制编码转换为文本。

---

## \$UnicodeEncoder %括号1%\$

Unicode 编码：将文本使用 UTF-16 BE \u 编码方式进行转码。

## \$UnicodeDecoder %括号1%\$

Unicode 解码：将 UTF-16 BE \u 编码转换为文本。

---

## \$Base64Encoder %括号1%\$

Base64 编码：将文本使用 Base64 编码方式进行转码。

## \$Base64Decoder %括号1%\$

Base64 解码：将 Base64 编码转换为文本。

---

## \$MD5 内容\$

计算一段文字的 MD5 <sup>1</sup>。

可以用来校验文件是否被更改，或者作为文件的唯一 ID。

---

## \$CRC32 内容\$

计算一段文字的 CRC32。

CRC32 和 MD5 一样，都算一种摘要算法。CRC32 计算的结果更短，也更容易重复

---

## **\$GTK %Skey%\$**

## **\$BKN %Skey%\$**

这两个函数完全相同，将 `%Skey%` 转换为 `bkn/g_tk`<sup>6</sup>。

---

## **\$SIG 内容\$**

QR 官方解释为：固定的数字签名算法。

似乎在 QQ 扫码登录时有用。

---

## **\$随机字符 内容 长度\$**

随机一串定义长度的内容

如果【内容】为空，则使用函数内置内容

示例：

`$随机字符 20$`

`$随机字符 Riko是猪啦 20$`

---

## **\$数字转中文 数字\$**

将阿拉伯数字转为中文数字

示例：

`$数字转中文 3248$`

---

## **JSON 操作函数**

注意，QR中所有JSON操作中都直接使用变量名，如「A」而不是「%A%」。

---

## **@A[1]**

- 获取 JSON 数组 `A` 中索引 `1` 对应的值;
- 获取 JSON 对象 `A` 中, 键 `"1"` 对应的值。

---

## \$JSON 添加 A 1 100\$

- 将 JSON 数组 `A` 中索引 `1` 对应的值修改为字符串 `"100"` , 如果前方有索引没有对应值, 将会补充空值 `null` ;
- 将 JSON 对象 `A` 中键 `"1"` 对应的值修改为字符串 `"100"` , 如果不存在键 `"1"` , 则创建一个新的键值对 `"1": "100"` 。

---

## \$JSON 添加 A 100\$

- 在 JSON 数组 `A` 的末尾添加字符串 `"100"` 。

---

## \$JSON 删除 A 1\$

- 删除 JSON 数组 `A` 中索引为 `1` 的值;
- 删除 JSON 对象 `A` 中键为 `"1"` 的键值对。

---

## \$JSON 长度 A\$

- 获取 JSON 数组 `A` 中元素的数量。
- 获取 JSON 对象 `A` 中元素的数量。

---

## \$JSON 获取 A 1\$

可以在 `@A[1]` 函数无法成功获取时使用

- 获取 JSON 数组 `A` 中索引为 `1` 的值;
- 获取 JSON 对象 `A` 中键 `"1"` 对应的值。

---

## \$JSON 包含 A 100\$

- 检测 JSON 数组 `A` 中是否包含字符串 `"100"` 或数值 `100` 。  
如果包含, 返回被检测值 (这里为 `100`) ; 否则返回空值。

只能检测是否包含字符串或数值; 无法检测 `true` `false` `null` 。

---

## \$JSON 包含 A value 100\$

- 检测 JSON 对象 `A` 中是否存在键的对应值为字符串 `"100"` 或数值 `100`。

如果包含，返回被检测值（这里为 `100`）；否则返回空值。

同上，只能检测是否包含字符串或数值。

## **\$JSON 包含 A key 100\$**

- 检测 JSON 对象 `A` 中是否存在键为字符串 `"100"`。
- 如果包含，返回被检测值（这里为 `100`）；否则返回空值。

同上，只能检测是否包含字符串。因为数值不能作为 JSON 的键。

## 外部Java调用

QR 的外部 Java 调用使用轻量级 Java 解释器 BeanShell，一些 Java 语言特性可能无法使用。

### **\$BSH xxxx.java %a%\$**

调用 `/sdcard/QR/QRDic/BSH/xxxx.java` 中的函数 `「%a%」`。

### **\$BSH xxxx.java %a% %b%\$**

调用 `/sdcard/QR/QRDic/BSH/xxxx.java` 中的函数 `「%a%」`，参数为 `「%b%」`。

## 其他函数

### **\$设置群状态 %群号% 开\$**

### **\$设置群状态 %群号% 关\$**

设置 QR 主程序中的开关。

这样一来，你无法在当前群触发任何消息。因此，你将难以在当前群内重新打开此开关。

### **\$添加管理员 %QQ%\$**

### **\$删除管理员 %QQ%\$**

添加与删除 QR 词库插件中的「管理员」。

不是添加群管理的 管理员。

---

## \$词库操作 添加 词库文件名\$

## \$词库操作 删除 词库文件名\$

添加与删除 QRDICPro 词库插件中的 加载词库。

---

## 画布

### \$画布 声明 A 长 宽\$

定义 [A](#) 画布的长和宽。

长和宽可以是浮点数/整数型

### \$画布 颜色 A 16进制颜色码\$

[十六进制颜色码](#)

定义 [A](#) 画布的背景颜色

### \$画布 文字 A 文字 X Y 16进制颜色码 size 字体文件路径（可选）\$

在 [A](#) 画布打印文字

参数说明：

【文字】 需要打印的文字

【X】 文字打印的X轴

【Y】 文字打印的Y轴

【[16进制颜色码](#)】 画布的颜色

【size】 文字打印的大小

【字体文件路径（可选）】 字体文件的绝对路径

示例：\$画布 文字 A Riko 100 100 FFB6C1 100\$

---

### \$画布 文字换行 A 文字 自动换行长度 X Y 16进制颜色码 size 字体文件路径（可选）\$

在 **A** 画布打印文字

参数说明：

【文字】 需要打印的文字（可携带换行符\n）

【自动换行长度】 自动换行的长度

【X】 文字打印的X轴

【Y】 文字打印的Y轴

【16进制颜色码】 画布的颜色

【size】 文字打印的大小

【字体文件路径（可选）】 字体文件的绝对路径

示例：\$画布 文字换行 A Riko\n是猪 400 100 100 FFB6C1 100\$

## \$图片参数 图片的绝对路径\$

获取图片的长和宽数组

示例：

A:\$图片参数 /mnt/sdcard/QR/tu.png\$

@A[Width]\n

@A[Height]

没有图片的话会报错

## \$画布 图片 A 图片绝对路径 X Y W H X1 Y1 X2 Y2\$

在 **A** 画布上打印图片

参数说明：

【图片绝对路径】 需要打印的图片路径

【X】 图片打印的X轴

【Y】 图片打印的Y轴

【W】 图片的宽度

【H】 图片的高度

【X1】 图片在画布的左上角X轴坐标

【Y1】 图片在画布的左上角Y轴坐标

【X2】 图片在画布的右下角X轴坐标

【Y2】 图片在画布的右下角Y轴坐标

示例：

\$画布 图片 A /mnt/sdcard/QR/tu.png 0 0 100 100 50 50 650 650\$

## \$画布 保存 A 图片保存绝对路径\$

将  画布图像保存

无法自行创建路径

只能保存在已存在文件夹

## 转义字符

`\r` `\n` `\r\n` 换行符

`\%0A` 换行符

`\%25` 百分号

`\%20` 空格

「\%xx」为 UTF-8 百分号编码的字符。

## 特殊关键词前缀

注：

[xx]xxx

表示前缀后需要加触发词；

[xx]

表示单独使用前缀即可。

## 特殊事件

### [系统]

进群等系统通知触发。

### [退群]

成员退群触发。

**[撤回]xxx** 授权专用

有人（包括机器人自己）撤回消息时触发。

**[红包]**

有人发送红包时触发。

QR 甚至可以撤回红包。（不过好像还是可以在群未领红包里领到？）

**[戳一戳]**

检测戳一戳消息

%QQ%获取戳一戳的戳方

%Code%获取戳一戳的被戳方

使用此标签请勿使用 Android Pad 、 Android Watch 协议登入

部分小标签消息和群打卡消息也会触发戳一戳标签，但%QQ%/ %Code%获取不到QQ

**[上下管理]**

检测QQ群管理变动。

示例:

[上下管理]  
如果:%Value%==1  
%昵称% 成为了管理员。  
返回  
如果尾  
如果:%Value%==0  
%昵称% 不再是管理员了。

**消息来源**



**[群]xxx**

**[好友]xxx**

**[临时]xxx**

只会被对应来源的消息触发。

---

## 其他前缀

**[内部]xxx**

只能通过 `$调用 xxx$` `$回调 xxx$` `$回调内部 xxx$` 触发，不会被 QQ 消息触发。

---

**[初始化]**

*QRSpeed词库插件* 每次开始运行（重启、重连）时触发一次。

因为不是在特定群执行，所以许多函数、变量无法使用。

**[群初始化]**

*QRSpeed词库插件* 每次开始运行（重启、重连）时每个群触发一次。

---

**[总处理]**

所有消息都会经过这里进行处理

【放至词库文件初始位置既所有消息均会触发 `[总处理]` 进行处理】

【放至词库文件末尾既除指令外的所有消息均会触发 `[总处理]` 进行处理】

约等于一个通配符【但是请不要把通配当 `[总处理]` 使用，会裂开的】

---

## 运算符

### 数学运算符

### 算术运算格式

—— 运算符与运算符 —— 运算符与运算符 ——



[decimal-最低显示位数-最高显示的位数（试到320）-舍入方法-舍入执行的位数个填默认为显示位数:算式]

示例：

[decimal-0-16-DOWN-3:1+1+1]

## 舍入办法

### UP

舍入远离零的舍入模式。

在丢弃非零部分之前始终增加数字(始终对非零舍弃部分前面的数字加1)。

此舍入模式始终不会减少计算值的大小。

### DOWN

接近零的舍入模式。

在丢弃某部分之前始终不增加数字（从不对舍弃部分前面的数字加1，即截短）。

此舍入模式始终不会增加计算值的大小。

### CEILING

接近正无穷大的舍入模式。

如果 计算结果 为正，则舍入行为与 UP 相同。

如果 计算结果 为负，则舍入行为与 DOWN 相同。

此舍入模式始终不会减少计算值。

### FLOOR

接近负无穷大的舍入模式。

如果 计算结果 为正，则舍入行为与 DOWN 相同。

如果 计算结果 为负，则舍入行为与 UP 相同。

此舍入模式始终不会增加计算值。

### HALF\_UP

向最接近的数字舍入，如果与两个相邻数字的距离相等，则为向上舍入的舍入模式。

如果舍弃部分  $\geq 0.5$ ，则舍入行为与 UP 相同；否则舍入行为与 DOWN 相同。

此舍入模式是我们大多数人在小学时就学过的（四舍五入）。

### HALF\_DOWN

向最接近的数字舍入，如果与两个相邻数字的距离相等，则为上舍入的舍入模式。

如果舍弃部分  $> 0.5$ ，则舍入行为与 UP 相同；否则舍入行为与 DOWN 相同（五舍六入）。

舍弃部分为 0.5，则舍入行为与 `HALF_UP` 相同；舍弃部分为 0.5 以下（含 0.5）。

`HALF_EVEN`

向最接近的数字舍入，如果与两个相邻数字的距离相等，则向相邻的**偶数**舍入。

如果舍弃部分左边的数字为**奇数**，则舍入行为与 `HALF_UP` 相同；

如果为**偶数**，则舍入行为与 `HALF_DOWN` 相同。

注意，在重复进行一系列计算时，此舍入模式可以将累加错误减到最小。

此舍入模式也称为“**银行家舍入法**”，主要在美国使用。四舍六入，五分两种情况。

如果前一位为奇数，则入位，否则舍去。

`UNNECESSARY`

断言请求的操作具有精确的结果，因此不需要舍入。

## 逻辑运算符与条件语句

### 关系运算符

`<` 小于

`>` 大于

`<=` 小于等于

`>=` 大于等于

`==` 等于

`!=` 不等于

### 逻辑运算符

`|` 逻辑或

`&` 逻辑与

QR 中，逻辑运算从左到右依次运行，不支持括号。

因此

```
%a==1&%b==2 | %c==3&%d==4
```

运算顺序为

```
((%a==1&%b==2) | %c==3) & %d==4
```

不知道什么是逻辑运算？请查阅高中数学课本或 [百度一下](#)。

### 条件语句关键字

如图。

如未。

判断其后的表达式是否成立。

如果成立则执行接下来的语句；否则跳转到对应的「如果尾」继续执行。

正则:

判断其后的表达式是否成立。如果表达式中的关系运算符为 `==` 或 `!=`，则以 `==` 或 `!=` 后的字符串作为正则表达式匹配前面的字符串。

如果成立则执行接下来的语句；否则跳转到对应的「如果尾」继续执行。

```
如果以文本形式作为参数的正则表达式中包含 |，会被识别成逻辑运算符。建议使用变量避免这个问题。如：

测试reg
a:(123|456)
正则:456==%a%
成功匹配
如果尾
```

如果尾

正则尾

「如果尾」已经在「如果:」「正则:」中介绍。

「正则尾」与「如果尾」完全等价，二者都可以用于「如果:」与「正则:」语句。

Pro 新增的正则尾大概是怕如果尾被写错...

其他关键字

本来想单独开一个关键字分区，但是许多关键字似乎更适合与相关内容一起介绍...

:

当 `:` 为非触发词的某一行的第二个字符时，则将 `:` 后面的值赋值给 `:` 前方的变量。

```
测试var
a:100
a的值为%a%。
```

±

用忘的刀隔付。 前后的又半云刀刀放八 删山缓中 中， 取终一起删山。

\*实际上我也不知道该叫它什么。

实际上 [消息内容操作函数](#) [消息发送函数](#) 中的部分函数所带有的 `±` 也是起到分隔与识别的作用。

## 其他知识

### 正则表达式基础

#### 简介

QR 中的触发词使用正则表达式进行匹配。即使你不知道什么是正则表达式，你也很可能在 QR 中使用过它。

如：`. *早上好. *` `我想. *`

你可能已经知道，`. *` 指代 *任意数量的任意字符*<sup>①</sup>。实际上，它就是简单的正则表达式。

① 严格来说并非如此。我们将在下面讲到，`. *` 并不能匹配换行符 `\n`。

在正则表达式中，具有特殊意义的字符被称为「元字符」。接下来将简单介绍常用的元字符的用法。

#### 元字符

##### 匹配某些字符

`\w` 匹配 文字、数字、下划线。

`\W` 匹配 `\w` 不能匹配的字符。

`\d` 匹配 数字。

`\D` 匹配 `\d` 不能匹配的字符。

`\s` 匹配 空白字符（包括空格、换行等）。

`\S` 你猜得没错，匹配 `\s` 不能匹配的字符。

`\n` 匹配 换行符。

`\r` 匹配 回车符。

QQ 中，手机发送的换行为换行符 `\n`，而电脑发送的换行为回车符 `\r`。

因此，防止电脑端无法触发指令，我们更建议使用 `[\n\r]` 来匹配换行。

`[]` 根据方括号的内容匹配 *部分* 字符。

如：

`[abcde]` 匹配 `a` `b` `c` `d` `e` 中的任意一个字符。它可以匹配 `a` `b` `c` `d` 或 `e`，而不能匹配 `f` `abc`。

`[^xyz]` 匹配除 `x` `y` `z` 之外的任意一个字符。它可以匹配 `a` `3` `/` `之` 或 `乙` 等，而不能匹配 `x` `abc`。

`.` 匹配除换行符 `\n` 外的 *任意* 字符。

`.` 与数量限制元字符 `*` 组合成 `. *` 被一些人称为「通配符」，它可以匹配任意不带换行的文本；但是，更加通配的明显是 `[\s\S]*` `[\d\D]*` 或 `[\w\W]*` 等，它们可以匹配任意文本。

当单独出现的「通配符」如 `.*` `[\s\S]*` 等作为触发词时，该触发词下方的触发词几乎都不会被触发。因此，词库中最好只出现至多一次的「通配符」且仅将它放在词库的最下方。

## 数量限制

`{4}` 限制前方元素出现 4 次。如： `\d{4}` 匹配 4 个数字； `(abc){2}` 匹配 `abccabc`。

`{1,4}` 限制出现 1-4 次。如： `\d{1,4}` 匹配 1-4 个数字； `[abc]{2}` 匹配 `aa` `bb` `ac` `bc` 等。

`{4,}` 限制出现 4 次或更多次。如： `\d{4,}` 四位及以上数字； `(苹果|apple){2,}` 匹配 `apple苹果` `appleapple苹果` `苹果apple苹果苹果apple` 等。

`?` 限制出现 0 次或 1 次。相当于 `{0,1}`

`+` 限制出现 1 次或更多次。相当于 `{1,}`

`*` 限制出现 0 次或更多次。相当于 `{0,}`

## 正则表达式触发词实例

你可以尝试以下代码：

`(无路赛){2,}`

测试成功了，你个傲娇。

`(\w)\n是\n笨\n蛋`

如果:%括号1%==你

虽然很不情愿，但是

如果尾

测试成功了。

`([你我他不?])是笨蛋\W*`

如果:%括号1%==你不| %括号1%==我| %括号1%==他

那当然！

返回

如果尾

这怎么可能呢？

戳了你`(\d+)`次`(\W*)`

捶了你%括号1%天%括号2%

点歌`\s*(.+)`

你点了一首「%括号1%」

## 转义符 \

转义符本应放在 元字符 中介绍的，但是为了不打断学习的连续性，我们把转义符从 元字符 中独立了出来。

输入命令应该在正则表达式下进行的。但是为了保持脚本的连续性，我们将输入从正则表达式中独立出来。

我们先来看一个例子：

一只小可爱 A 想写一个骰子。他选择了 `.r` 来作为掷骰子的触发词。

测试之后，他发现骰子没有问题，便在群内开放使用了。

第二天，大家在群里聊天：

B：「A 劳斯您好，您家崽人设是您自己画的对叭？！太神仙了呜呜呜，请问您接单吗？」

A：「接。」

B：「多少钱呀？」

A：「1r」

机器人：「@A 1d100 = 32」

A：「？」

内容完全虚构，若有雷同，纯属我太聪明了

在正则表达式中，`.` 表示的是一个任意非换行字符。因此，`1r` 可以被 `.r` 匹配，触发了指令。

一些字符（如 `.` `(` `+` 等）由于成为了 元字符 而被赋予了特殊含义，从而不再代表这些字符本身。

那么如何匹配这些字符本身呢？答案很简单，我们只需要在元字符前加上 转义符 `\`。

在这个例子中，如果将词库中的触发词改为 `\.r`，就只有 `.r` 才可以触发骰子了。

## 完整的正则表达式教程

[点我查看。](#)

## QR 中的 JSON

### 关于 JSON

[百度百科](#)

[JSON 在线格式化工具](#)

当你拿到一串 JSON 时，给人的直观感受很可能是「乱糟糟的一团」；  
这时，你可以使用 JSON 格式化工具 将其格式化，以直观地看到 JSON 的层次结构。

同时，你也可以用 JSON 格式化工具来检查自己的 JSON 是否包含语法错误。

## JSON 操作代码示例

\$JSON 添加 A 1\$



JSON 添加 A 1 \$

```
JSON添加1
A:[1,2,3]
$JSON 添加 A 9$
%A%
```

\$JSON 添加 A 1 1\$

```
JSON添加2
A:[1,2,3]
B:{"玖":"1","安":"an"}
$JSON 添加 A 1 a$
$JSON 添加 B 玖 9$
$JSON 添加 B 玖玖 安安$
A: %A%\n
B: %B%
```

\$JSON 获取 A 1\$

```
JSON获取
A:[1,"8",3]
B:{"玖":"9","安":"an","玖玖":"安安"}
$JSON 获取 A 1$\n
$JSON 获取 B 玖$
```

\$JSON 删除 A 1\$

```
JSON删除
A:[1,"8",3]
B:{"玖":"9","安":"an","玖玖":"安安","an":"annie"}
$JSON 删除 A 1$
$JSON 删除 B an$
A: %A%\n
B: %B%
```

\$JSON 长度 A\$

```
JSON长度
```

```
A:[1,"8",3]
B:{"玖":"9","安":"an","玖玖":"安安","an":"annie"}
$JSON 长度 A$\\n
$JSON 长度 B$
```

## QQ 官方 API

`±img=http://q2.qlogo.cn/headimg\_dl?dst\_uin=%QQ%&spec=5±`

获取发言人头像。

`±img=http://p.qlogo.cn/gh/%群号%/%群号%/±`

获取群头像。

## 时间变量详解

从 `%时间%` 过来却看晕了? [点我返回](#) `%时间%` 变量。(劝退?)

`%时间%` 变量里有很多保留字符。使用 `%时间%` 变量时, 保留字符会被替换为对应值, 而其他字符保持不变。

这样说可能有些抽象, 这里先放示例。

你可以通过尝试下方的代码, 了解 `%时间%` 变量的用法。

时间变量测试

```
%时间HHmm%\\n
```

```
%时间yyyy年MM月dd日 HH:mm:ss%\\n
```

```
%时间现在时间ah时m分s秒%
```

接下来将具体说明 `%时间%` 变量中的保留字符。

如果 `%时间%` 变量中带有保留字符之外的 英文字母 (如: `%时间AAHHmmSS%`), 会报错并使整段代码不可用。

在这种情况下, 你可以使用单引号 `'` 包裹住不想被转换为时间的部分, 以期其不会被格式化解释; 而连续的两个单引号 `''` 则会被转换为输出结果中的一个单引号 `'`。这种方法也适用于保留字符。(如: `%时间'AA'HH' 'mm' 'SS' %`)

## 基础时间变量

时间格式化函数 ( `$时间 vvvv-MM-dd$` `$时间格式 %NDTime% vvvv-MM-dd$` ) 中, `vvvv-MM-dd` 这部分内容被称为

格式化字符串。

由于格式化字符串中的许多关键字很少用到且规则复杂，这里只介绍一些简单的时间格式。想具体了解，请点击 [时间格式化详解](#) 查看。

使用示例：

测试time

\$时间 HHmm\$\n

\$时间 yyyy-MM-dd HH:mm:ss.SSS E\$\n

\$时间 时间现在时间ahh时mm分ss秒\$

以下介绍一些常用的关键字。将关键字放入格式化字符串中，就会被替换为对应的时间。

yyyy 年

MM 月

dd 日

HH 时：24 小时制

hh 时：12 小时制

mm 分

ss 秒

E 星期

a 半日（上午/下午）

## 更多时间变量

部分时间变量存在 *补 0 规则*。

你可以通过以下示例来了解补 0 规则：

%时间y% %时间yy% %时间yyy% 值为 2021

%时间yyyy% 值为 02021

%时间yyyyyy% 值为 002021

除了少部分特殊字符组，大部分结果为 *纯数字* 的时间变量均具有 *补 0 规则*。

为方便学习，具有 *补 0 规则* 的时间变量，将在「结果示例」中标注 **补0**。

未标注 **补0** 的变量，无论重复多少次字母均不会使结果变化。

如：%时间GGG% %时间GGGG% 甚至 %时间GGGGGGGGGGGGGGGG% 结果均为 公元。

以下所有 *结果示例* 使用的示例时间为：

2021 年 2 月 19 日 18:03:25.876 GMT+8:00

**G** 历法

**yy** 年

结果示例：21

**y** 年

结果示例：2021 补0

**Y** 年，以本周最后一天的年份为准

结果示例：2021 补0

**M** **L** 月

结果示例：2 补0

**d** 日

结果示例：19 补0

**H** 小时：24 小时制（午夜为 0 点，没有 24 点）

结果示例：18 补0

**k** 小时：24 小时制（午夜为 24 点，没有 0 点）

结果示例：18 补0

**h** 小时：12 小时制（午夜为 0 点，正午为 12 点）

结果示例：6 补0

**K** 小时：12 小时制（午夜为 12 点，正午为 0 点）

结果示例：6 补0

**m** 分

结果示例：3 补0

**s** 秒

结果示例：25 补0

**a** 上午/下午

结果示例：下午

**cccc** **EEEE** 星期

结果示例：星期五

**ccccc** **EEEE** 星期

结果示例：五

**c** **E** 其他数量**c** 其他数量**E** 星期

结果示例：周五

**u** 星期

结果示例：5 补0

**D** 今年第x天

结果示例：50 补0

**w** 今年第x个星期 日/一/二 ...

结果示例：8 补0

类似于 **Y** **YY**，以每周最后一天为年份标准。  
例如，在 2021 年 12 月 29 日，该值为 1。

**w** 本月第x个星期 日/一/二 ...

结果示例：3 补0

**F** 本月的第x个七天

结果示例：3 补0

**G** 历法

结果示例：公元

**s** 十分之一秒

结果示例：8

**ss** 百分之一秒

结果示例：87

**sss** 千分之一秒（毫秒）

结果示例：876 补0

**z** **Z** **X** 系列：当前时区

**z** 结果示例：+0800

**zz** **zzz** 结果示例：GMT+08:00

**zzzz** 更多**z** 结果示例：中国标准时间

**z** **zz** **zzz** 结果示例：+0800

**zzzz** 结果示例：GMT+08:00

更多**z** 结果示例：+08:00

z、Z 和 X 似乎是一般时区、RFC 822 时区和 ISO 8601 时区的区别。其中「一般时区」原文为「General time zone」，翻译可能有误。虽然我也不知道他们之间有什么区别，但我先把它写在这。

## 未整理&杂项

## 红包/转账相关

## 转账检测

[转账](\S\S+)  
如果:%Tfuin%!=  
%Tfuin%\n  
%Tfurl%\n  
%Tftip%\n  
%Tftitle%\n  
%Tftext%\n  
%TfAmounts%\n  
如果尾

## 查红包

```
$查红包 %群% %Skey% %Authkey% %Listid% A$
%A%
```

## 卡片代码示例

你需要使用 [这些](#) 函数来发送卡片。

卡片可能失效（屏蔽），因此这里不保证作为示例的卡片代码可用。

## JSON 卡片

[illegible]

```
{ "app": "com.tencent.miniapp", "view": "notification", "meta": { "notification": { "appinfo": { "appName": "优质解答", "iconUrl": "https://icean.moe/?req=0" }, "button": [ { "name": "一种植物" } ] } } }
```

## XML 卡片

```
<?xml version='1.0' encoding='UTF-8' standalone='yes' ?><msg  
servicetemplateaction="viewMultiMsg" brief="[聊天记录]"  
m_resm_fileName="6931536343294312393" tSum="1" sourceMsgurl="" flag="3" adverSign="0"  
multiMsgFlag="0"><item layout="1" advertiser_amaxLines="2" lineSpace="12">群聊的聊天记  
录</title><title size="26" color="#777777" maxLines="2" lineSpace="12">An!: 这是一张卡  
片</title><hr hidden="false" style="0" /><summary size="26" color="#777777">查看1条转发消  
息</summary></item><source name="聊天记录" icon="" action="" app/></msg>
```

## 废弃&失效&不明意义部分

~~±imgtype=%随机数40000-40005%±~~

定义秀图类型；已失效

好像又能用了？我懒

~~%Number%~~

目前已知在红包消息中为 1，其他消息中为 0。QR 官方解释是红包金额，可能已废弃。

~~%Pid%~~

未知；似乎是消息文本的「唯一」ID。相同的文本 `%Pid%` 相同，不同的文本 `%Pid%` 认为不同。

~~%Strmsg%~~

废弃

~~±nick 测试昵称±~~

废弃

~~±title 测试±~~

自定义头衔为「测试」，仅作用于当前一条消息。

失效。

与群主设置的专属头衔不同，QR 设置的头衔可以超过 6 字限制。

## 抢红包

\$抢红包 %Groupid% %Skey% %Authkey% %Listid% %Number%\$  
\$抢红包 %Groupid% %Skey% %Authkey% %Listid% %Number% A\$

---

## 发红包

##指定专属红包 8不指定 32口令红包 1024指定 65536语音红包

A:1024

##红包类型 1普通红包 2拼手气红包

B:2

##发送类型 1好友 3群 4非好友

C:3

##红包金额 分为单位

D:%随机数5-10%

##红包数量

E:1

##接收红包 群号或好友QQ号

F:%群号%

##红包标题 或 口令

G:我是红包

##指定发红包用|分割

H:%Code%

\$红包 %A% %B% %C% %D% %E% %F% %G% %H%\$

示例:

我要红包

\$红包 1024 1 3 1 1 %群号% 你的红包 %QQ%\$

---

## 关于

### 关于变量大全编写

#### 作者

- An!
- 阿玖
- 思思

#### 编辑工具

[Typora](#)



---

## 关于共享

### 许可协议

本作品采用 [知识共享 署名-相同方式共享 4.0 国际许可协议](#) 进行许可。

如果你认为本文有不妥之处，可以对下载后的文件进行适当修改。修改后的文章仍应以 [此许可协议](#) 发布。

感谢你对知识共享的支持与对 QR 新人的帮助。

### 想要制作 **变量大全安♡玖特别版** 的网页镜像？

你可以 [点此下载 HTML 文件](#)，并在稍后上传到你的服务器。

如果想用其他 Markdown 编辑器重新渲染网页，你也可以 [点此下载 Markdown 文件](#)。

### 想要以文件形式分享 **变量大全安♡玖特别版**？

你可以选择下载以下一个或多个文档并上传到任何平台：

[PDF 文档 \(推荐\)](#) [HTML 文档 \(推荐\)](#) [Word 文档](#) [Epub 文档](#) [RTF 文档](#)

HTML 以外格式的文档不保证最新版本；其中 PDF 格式的文档较常更新，但仍可能会落后于最新版本 1-2 次修改。

---

1. 一种摘要算法。↩↩↩

2. 气泡 ID 手动获取方法：在聊天界面双击想要获取 ID 的气泡→点按右下角「装扮」→点按气泡名称→点击右上角的「…」展开菜单→手指放在菜单第三行向左轻扫→轻触「复制链接」成功复制链接后，找合适的输入框粘贴刚刚复制的内容，可以在链接文本找到 `bubbleId=`，其后紧跟的数字就是气泡 ID。↩↩↩

3. UNIX 时间戳：1970-01-01 00:00:00 GMT 至当前的总秒数。↩

4. 一种常见编码方式。编码方式决定字符在计算机中的存储方式，使用错误的编码方式读取文件会出现乱码。↩↩

5. 腾讯对「风险」QQ 号的一系列限制。↩

6. QQ 网页 cookie 的一部分，由 skey 通过某种摘要算法计算得出。↩