

RSL脚本语言记法

- RSL脚本语言记法
 - 关键字
 - 变量
 - 字符串
 - 注释
 - 输出
 - 输入
 - 函数
 - 分支跳转
 - 静音
 - 默认
 - 退出

关键字

Step	Speak	Branch
Listen	Silence	Default
Exit		

变量

RSL脚本中变量类型设置默认为字符串型

```
$arg1
```

arg1表示变量名称

eg:

```
$name #引用name变量
```

字符串

在该语言中，想要表达字符串即用双引号引用起来即可。

```
"..."
```

eg:

```
Speak $name + "您好，请问有什么可以帮助您？请在听到滴的一声时开始说话"
Speak "请问您有什么想与我们交流的吗？请提出你宝贵的建议。请在听到滴的一声时开始说话"
```

注释

脚本支持单行注释

```
#...
```

eg:

```
# 这是一个注释  
Step Proc #这也是一个注释
```

输出

脚本输出Speak表示，计算表达式合成一段文字，打印字符串通过双引号引用，不同部分之间用+号连接

```
Speak arg1+arg2
```

arg1、arg2表示字符串或者变量

eg:

```
Speak $name + "您好，请问有什么可以帮助您？请在听到滴的一声时开始说话"  
Speak "您的语文成绩为" + $chinese + "分，您的数学成绩为" + $math + "分。祝你能收获一个满意的  
成绩。再见！"
```

输入

输入需要通过Listen函数

```
Listen arg1 arg2
```

arg1表示开始能够输入的计时器，arg2表示结束计时器

eg:

```
Listen 5 20 #表示5s后才能输入，在20s时停止输入（即输入时间为20-5=15s）
```

函数

用Step创建一个步骤，表明建立一个函数，完整表示一个步骤的所有行为。其中第一个声明建立的函数默认为程序的开始位置

```
Step arg1
```

arg1表示新建的step的名称

eg:

```
#起始步骤
Step welcome
Speak $name + "您好, 请问有什么可以帮助您? 请在听到滴的一声时开始说话"
Listen 5 20
Branch "查成绩" scoreProc
Branch "提建议" adviceProc
Silence silenceProc
Default defaultProc
```

分支跳转

Branch对不同的情况进行不同的跳转，连接各个step步骤

```
Branch "arg1" arg2
```

arg1表示用户输入的回答，arg2表示该回答需要跳转的step的名称。

eg:

```
Branch "查成绩" scoreProc
Branch "提建议" adviceProc
```

静音

在没有回答的情况下，即静音条件下，我们也需要跳转。并且如果连续三次静音操作，则会退出程序。

```
Silence agr1
```

agr1表示静音条件下需要跳转的step名称。

eg:

```
Silence silenceProc
```

默认

在没有回答匹配的情况下，我们也需要跳转。

```
Default arg1
```

arg1表示在默认条件下，需要跳转的step名称

```
Default defaultProc
Default thanks
```

退出

如何退出程序，我们用Exit，当执行到这一句时则退出

```
Exit
```

eg:

Step thanks

Speak "感谢您的来电，再见"

Exit