

Praat 脚本常用命令语句汇集解说

一 对话框中的命令语句

1.1 文本变量

1.1.1 sentence 语句

sentence 后面的 1 个单词会出现在对话框中。如需要使用多个单词，应该加下加线 “_”（下加线不会在对话框中显示。加短横线也可以，但短横线会显示在对话框中。总之就是 sentence 命令后另一个空格之前的内容可以显示，空格之后的内容不会显示）。

如：

```
sentence Sound_file_extension .wav
```

1.1.2 text 语句

text 后面的 1 个单词不会出现在对话框中。

```
text input_directory 请在此输入文件夹路径，如 C:\tmp\
```

1.2 数字变量

real, 实数, 如：

```
real Value 100
```

positive, 正实数（如输入的是 0 或负数会报错）

integer, 整数

1.3 选择变量

1.3.1 option 语句

一个 option 是一个选项，点击展开

如：

```
optionmenu Convert_from: 1
  option Hertz to Semitones
  option Semitones to Hertz
  option Hertz to Bark
  option Bark to Hertz
  option Hertz to Mel
  option Mel to Hertz
```

1.3.2 choice 语句

一个 option 是一个选项，直接显示

```
choice informant_character: 1
  button 男
  button 女
  button 小孩
```

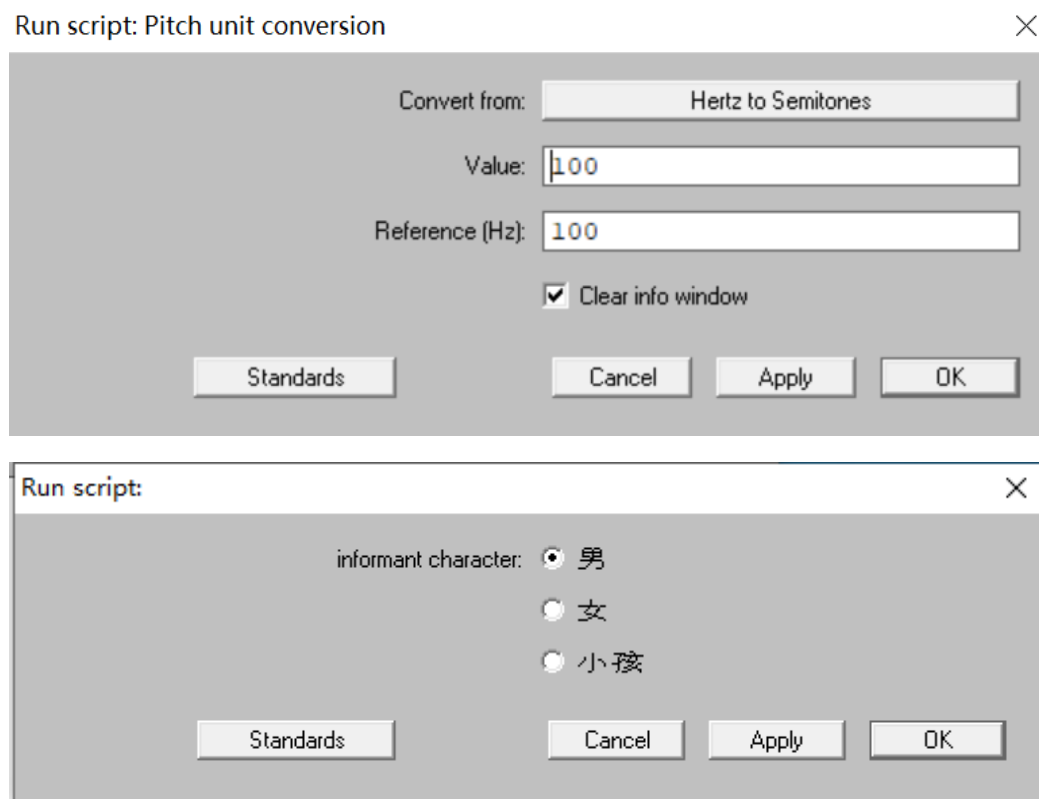
1.3.3 boolean 语句

boolean: 挑钩选项

如:

```
boolean Clear_info_window 1
```

上面几个对话框中的命令显示为:



二 主窗口相关的命令语句

2.1 创建序列（列表）的方法

```
Create Strings as file list... fileList 'input_directory$\'*.wav
```

变量最后需要加\$。

变量的外面需要加单引号”。

以哪种文件创建序列，前面要有星号*。

2.2 选取主窗口中的对象文件

```
select Strings fileList
```

```
select Sound 'simpName$'
```

选取对象窗口中的文件，select 首字母小写。

后面的文件类型首字母大写。

选取全部文件后，不选取其中某个文件;

```
select all
```

```
minus Strings fileList
```

选取多个对象文件:

```
selectObject: 1,2
myObject# = {1,2}
selectObject: myObject#
```

2.3 从目录中读取文件

Read from file...

从某文件夹中读取特定文件名的文件:

Read Strings from raw text file... 'soundpath\$'temp.txt

2.3.1 读取文本文件

Read Strings from raw text file... 'inputDirec\$'fileName\$'

2.4 设定已选的声音文件的名称为文件名

```
object_name$ = selected$ ("Sound")
```

2.5 在 TextGrid 的间隔中添加文本内容

Set interval text... 2 'interval' 'text\$'

在 TextGrid 的第 2 层, 第'interval'个间隔, 添加文本'text\$'

2.6 其他

2.6.1 移除文件, Remove 首字母大写。

2.6.2 正文中变量名首次出现, 最后也需要加\$。

例如:

```
filename$ = Get string... ifile
```

变量名首字母不能大写。

2.7 移除 TextGrid 文件中的某个间隔

Remove right boundary... 1 'ii'

移除第 1 层的第 ii 个间隔的右边边界

三 条件与循环

3.1 条件命令

if 后面直接跟条件; 条件结束必须有 endif。

例如:

```
if left$(openpath$,1)<>"\"
    openpath$=openpath$+"\"
endif
```

又如:

```
if output_format$ = "AIFF"
    Write to AIFF file... 'output_directory$'objectname$.aif
else
```

```
Write to WAV file... 'output_directory$'objectname$.wav
endif
```

3.2 多个并列条件命令

可用 or（或）或 and（并且）连接起来，例如：

```
if ctext2$="zh" or ctext2$="sh" or ctext2$="ch"
```

```
if (ctext1$="z" or ctext1$="s" or ctext1$="c") and ctext2$<>"zh" and ctext2$<>"sh" and
ctext2$<>"ch"
```

3.3 循环命令

循环命令有 for、repeat、while 等，最常用的是 for。

循环通常都是数字循环。

例如：

```
for i from 251 to 300
  Read from file: "C:\Users\ran\Desktop\6 壮语音系-声调\i.wav"
  View & Edit
  pause 检查'xu'
  Remove
endfor
```

又如：

```
for ifile from 1 to numberOfFiles
  select Strings fileList
  fileName$ = Get string... 'ifile'
  Read from file... 'openpath$'fileName$'
  View & Edit
  Remove
endfor
```

四 文件输出的方法

4.1 写入信息窗口

直接在信息窗口显示：

```
print Hello world!
```

如信息窗中已有内容，在新的一行显示：

```
printline 'f_one:0'
```

```
printtab 'x'
```

```
echo 'x'
```

清除信息窗口里面的所有信息：

```
clearinfo
```

4.2 将信息窗口的内容写入文本文件：

```
fappendinfo 'soundpath$temp.txt'
```

4.3 写入文件

4.3.1 创建并写入文本文件

地址和文件名分别用单引号，外面用双引号。后面写入的内容前有 1 个空格。各个数据之间没有空格。

‘tab\$’首字母为小写!! (变量首字母均必须为小写)

如:

```
fileappend "'openpath$textfile$" " speaker'tab$'vowel/frame'tab$'context'tab$'F1  
fileappend "'openpath$textfile$" 'f_one:0'tab$'
```

4.3.2 创建并写入表格文件

Write to table file... 'output_directory'\每隔 10 毫秒提取音高、强度、共振峰数据结果.xls

将文件写入脚本所在文件夹，直接写文件名即可:

```
Write to text file... new.TextGrid
```

4.3.3 直接创建文件并将文本内容写入

使用 “>” 命令:

```
header_row$ = "speaker" + tab$ + "vowel" + tab$ + "context" + tab$ + "F1" + tab$ + "F2" + tab$ +  
"F3" + tab$ + "gl F1" + tab$ + "gl F2" + tab$ + "gl F3" + newline$  
header_row$ > 'directory$'formantoutput.xls
```

>> 追加文本到已有文件中

4.4 删除文件

删除某目录中的文件，例如:

```
filedelete 'directory$'formantoutput.xls
```

删除脚本所在文件夹中的文件，直接写文件名即可，例如:

```
filedelete file.txt
```

4.5 另存文件的命令

Write to WAV/AIFF/text file...

4.6 新建文件夹命令

如:

```
createDirectory: mfccpath1$
```

4.7 文本变量与数字变量表示的差异

用文本表示的数字变量，直接用文本。

例如:

```
numberOfFiles
```

用文本表示的文本变量，在最后需要加\$。

例如:

```
fileName$
```

4.8 检查某文件夹中是否已存在某文件

例如:

```
if fileReadable ("folder$'soundname$'.TextGrid")
    pause The file 'folder$'soundname$'.TextGrid already exists. Do you want to overwrite it?
endif
```

又如:

```
if fileReadable(numFile$)
    num$<'numFile$'
    numfil=extractNumber(num$,"")
endif
```

又如:

```
if fileReadable("2.wav")
    Read from file... 2.wav
endif
```

如果脚本所在文件夹中存在名称为“2.wav”的文件，则读取该文件。

五 变量处理与公式

5.1 求某个变量的长度

```
length(directory_manual$)
```

例如:

```
if length(directory_manual$) > 0
    directory$ = directory_manual$
endif
```

5.2 某个变量右侧的若干个字符

```
right$ (filename$, 9)
```

例如:

```
if right$ (filename$, 9) = ".TextGrid"
    Read from file... 'directory$'filename$'
endif
```

5.3 将数字转换为字符串

```
string$(56%) → 0.56
```

5.4 数据保留小数

5.4.1 数据取整（去掉小数，只保留整数）

```
f_one:0
```

5.4.2 使用 fixed 命令，如对 x 保留 3 位小数:

fixed\$(x,3)

5.5 从字符串中提取数字，提取的是 b\$后面的数字

extractNumber(a\$,b\$)

如 extractNumber("I am 32.,"am"): 提取的是 "am" 之后的 32。

5.6 把其他数字表达转换为纯数字

string\$(5e6) = 5000000

string\$(56%) = 0.56

将字符串变量转换为数字

number(a\$)

5.7 求字符串中是否包含某些字符串

index(a\$,b\$)

如有则返回首次出现在第几个字符位置，没有则返回 0

rindex(a\$,b\$)

如有则返回最后一次出现在的第几个字符位置，没有则返回 0

5.8 字符串变量的计算

字符串变量进行计算时，不能加单引号，否则会报错。例如：

```
rName$ = 'simpNameec$' + 'numName2$'
```

必须直接写不加单引号的变量：

```
rName$ = simpNameec$ + numName2$
```

5.9 对字符串和数字之间进行比较和计算，但是后台不能直接对二者进行计算，例如：

```
if numName1$ = 1
```

```
    numName2$ = 01
```

```
endif
```

会报错：

(1) numName1\$ = 1 会报字符串和数字不能比较；

(2) numName2\$ = 01 也会报错。

这时，将变量名外面加单引号，即可与数字相比较；同时，在数字外面加双引号，可以将数字和字符串进行比较。按下面这样写即可：

```
if 'numName1$' = 1
```

```
    numName2$ = "01"
```

```
endif
```

5.10 有关公式

abs(x) 求绝对值

round(x) 求与 x 最接近的整数，四舍五入

floor(x) 求小于或等于 x 的整数

ceiling(s) 求大于或等于 x 的整数

sqrt(x) 求 x 的平方根，x 须 ≥ 0

`min(x,...)` 求括号内的一组数的最小值，各个数值用逗号隔开

六 其他

6.1 给不带“\”的文件夹地址最右侧添加“\”

如：

```
if left$(input_Directory$,1)<>"\"
    input_Directory$=input_Directory$+"\\"
endif
```

上面的脚本命令中，原则上应该写“`right$`”；但此时写“`left$`”也可。

6.2 一行命令太长换行

1 行如果太长，可以加“...”转换为 2 行

6.3 某操作执行过程中不出现警告信息、不出现检测信息等

`nowarn Save as WAV file: "hello.wav"`

`nocheck Insert boundary... 3 'addTi'`

6.4 脚本运行中暂停

例如：

`pause` 开始手动标注

`pause` 选择需要比较距离的两个 MFCC 文件

6.5 调用 `procedure` 步骤

`procedure playOctave`

`endproc`

6.6 调用 `procedure` 步骤中的赋值功能

例如：

`call AddBarkScale f1_minimum f1_maximum f2_minimum f2_maximum 1`

`procedure AddBarkScale f1min f1max f2min f2max garnish`

步骤 `AddBarkScale` 之后的数字变量，`call` 步骤之后的数值，都是对下面 `procedure` 步骤之后变量的赋值。

6.7 调用系统命令

`runSystem: "del", directory$, "*.wav"`

6.8 在脚本中调用其他脚本

`include square.praat`

6.9 脚本中执行其他脚本

`execute "C:\Users\rans\Desktop\珠海试验\珠海语音语料库二字（普通话）\reverse.praat"`

6.10 暂停脚本

pauseScript: message

6.11 退出脚本

exitScript()

exitScript: *error-message*

6.12 结束脚本

exit OK! 所有文件已经处理。

6.13 将脚本添加到固定菜单

脚本窗口: Add to fixed menu...

6.14 将脚本添加到动态菜单

脚本窗口: Add to dynamic menu...

6.15 退出 Praat

Quit