一．PythoninnV使用方法

1）初始化，需要使用一个脚本pythtoninnv.u，该脚本在D:\project\jiangshu\nV\trunk\Link\Python目录下，所以在nV下输入的命令是<<"D:\project\jiangshu\nV\trunk\Link\Python\pythoninnv.u"，执行该脚本后，会引入全局的名字空间Python，并在该名字空间下，挂载某些函数功能。

2）在Python下提供的函数有execute，eval,，locals，globals，builtins，import，close。

a) execute，eval参数为一个字符串，可以用于直接以Python的方式输入命令，比如Pyton`execute[“b = 1”]等等。

b) import函数，参数为一个字符串，引入一个python module，比如str = Python`import[“string”]，可以通过str@lower[“SDFdsfd”]访问函数的功能

c)locals,globals,builtins函数返回Pyton的局部命名空间、全局命名空间和builtins命名空间，命令如下:pl = Python`locals[], pg = Python`globals[], pt = Python`builtins[]。然后可以访问其功能，比如pt@dir[pt]。

**注意：**有这样一个问题，访问全局空间中的属性，比如\_\_builtins\_\_，pg@\_\_builtins\_\_和pg[\_\_builtins\_\_]这样的方式就不行。而要pg[“\_\_builtins\_\_”].

d) 调用Python`close[]可以关掉引入的python

**注意：**是不是考虑不提供这个接口，可能会带来写麻烦，比如有某个对象，指向了python对象，关掉了就不好了

二．PythoninnV中数据结构

1）PythonObject是继承自Object的一个类，它包装了一个PyObject，提供了一些常用的成员函数Init、Close、构造、析构、compare、equal、hash、print等函数。

2）py\_convert函数用于将python中的数据结构转换到var中的数据结构，对于Py\_None、Py\_True、Py\_False、Pyring、PyInt、PyLong、PyFloat等基本的数据结构。nV中都有相应的数据结构对应，对于PyList和PyTuple，都转换到nV中的Tuple List，对于nVObject，可直接返回其包装的var，其它类型，转换为PythonObject.