**[python 全局变量引用与修改](http://www.cnblogs.com/yanfengt/p/6305542.html)**

**一、引用**

使用到的全局变量只是作为引用，不在函数中修改它的值的话，不需要加global关键字。如：

[复制代码](javascript:void(0);)

#! /usr/bin/python

a = 1

b = [2, 3]

def func():

if a == 1:

print("a: %d" %a)

for i in range(4):

if i in b:

print("%d in list b" %i)

else:

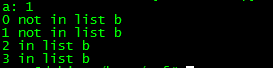
print("%d not in list b" %i)

if \_\_name\_\_ == '\_\_main\_\_':

func()

[复制代码](javascript:void(0);)

输出结果：



可以看出，无论是列表还是变量，都是可以直接引用的。

**二、修改**

使用到的全局变量，需要在函数中修改的话，就涉及到歧义问题，如：

[复制代码](javascript:void(0);)

#! /usr/bin/python

a = 1

b = [2, 3]

def func():

a = 2

print "in func a:", a

b[0] = 1

print "in func b:", b

if \_\_name\_\_ == '\_\_main\_\_':

print "before func a:", a

print "before func b:", b

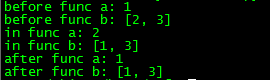
func()

print "after func a:", a

print "after func b:", b

[复制代码](javascript:void(0);)

输出结果：



可以看出，对于变量a，在函数func中"a = 2"，因为存在既可以表示引用全局变量a，也可以表示创建一个新的局部变量的歧义，所以python默认指定创建一个新的局部变量来消除这一歧义，但对于列表b而言，"b[0] = 1"不存在这种歧义，因此直接修改了全局变量，但是如果改成了"b = [3, 4]"，那么b也会变成局部变量。特别地，当在func中a = 2之前加入"if a == 1:"这一语句，脚本运行出错，因为这一语句引入了全局变量，导致了"a = 1"这一语句无法创建同名的局部变量。

因此，需要修改全局变量a，可以在"a = 2"之前加入global a声明，如：

[复制代码](javascript:void(0);)

#! /usr/bin/python

a = 1

b = [2, 3]

def func():

global a

a = 2

print "in func a:", a

b[0] = 1

print "in func b:", b

if \_\_name\_\_ == '\_\_main\_\_':

print "before func a:", a

print "before func b:", b

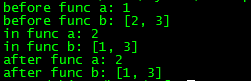
func()

print "after func a:", a

print "after func b:", b

[复制代码](javascript:void(0);)

输出结果：



**结论：引用全局变量，不需要golbal声明，修改全局变量，需要使用global声明，特别地，列表、字典等如果只是修改其中元素的值，可以直接使用全局变量，不需要global声明。**