Python的作用域顺序是局部变量，非局部变量，然后是全局变量。当我们通过调用函数去修改变量时，是涉及此方便的问题。

Example 1:

def modify():

testString =”123”

test\_dict = {“a”:2}

testString = “abc”

test\_dict = {“a”:1}

modify()

执行完成后，字符串testString还是abc, test\_dict还是 {“a”:1}。这是因为在modify()函数里面，申明了局部变量testString,局部变量test\_dict, 所以python局部变量开始找，都认为是局部变量，所以全部变量是不会变的。如果要变要则要特殊申明,通过global关键字，告诉python我用的是全局变量。

Example 2:

def modify2():

global testString

testString = “123”

global test\_dict

test\_dict = {“a”:2}

testString = “abc”

test\_dict = {“a”:1}

modify2()

执行完成后，字符串testString和 test\_dict 都将被改变。因为在这我们显示表明使用全局变量，所以函数将改变全局变量。

Example 3:

def modify3():

testString = “123”

test\_dict[“a”] = 2

testString = “abc”

test\_dict = {“a”:1}

modify3()

执行完成后，字符串testString 不会被改变，因为pytho优先找到了局部变量testString， 而test\_dict将会被改变。因为test\_dict[“a”]与example1中test\_dict = {“a”:2} 不一样。 test\_dict[“a”]是要有test\_dict 去改变其中的key = “a”的值，python会去找局部变量，找不到然后找到global的变量改变，如果还没有则报错。 而test\_dict = {“a”:2}是申明变量并且赋值，所以python永远首先找到的局部变量的test\_dict

Example 4:

def clear\_dict1():

test\_dict = {}

def clear\_dict2():

test\_dict.clear()

def clear\_dict3(pass\_dict):

pass\_dict.clear()

test\_dict = {“a”:1}

clear\_dict1()

clear\_dict2()

clear\_dict3(test\_dict)

exampl4中，我要清除test\_dict中的所有键值对。通过clear\_dict1()是无法清除的，因为在函数申明了局部变量test\_dict,python只会改动局部变量。而clear\_dict2是可以的，因为没有申明，python按顺序 只能使用全局的test\_dict, 所以可以清空字典。Clear\_dict3也是可以的，对与python来说，此时pass\_dict指向的是test\_dict全局变量，所以可以清空字典。

P.S Python查找变量本质是局部变量，非局部变量，全局变量的顺序。对于字符串，数字，由于赋值和申明一样的形式, 所以要改变函数外的值要使用global. 对于字典，列表， 如果我们要改变全局变量的值，只要不要申明变量，python将使用全局变量，从而改变。