

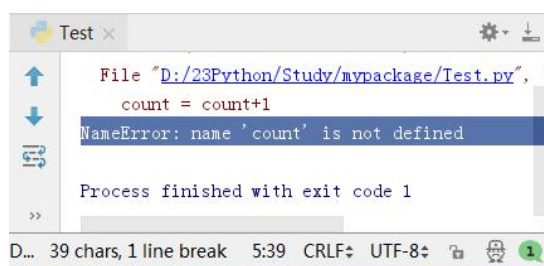
4. 变量的作用域

在 Python 中被创建访问时，都是在一个有限的范围内进行，我们称之为命名空间，也被称之为作用域。简而言之，就是变量的作用范围是一定的，并不是任何地方都能够访问的某个变量的。

例如，如下的代码片段就会报错。

```
for i in range(1,10):  
    count = count+1  
print('总计数: ',count)
```

报错信息如下图所示，提示 `count` 没有定义。



这段代码中，`count` 是在 `for` 循环中定义使用，而 `print()` 是在 `for` 循环外的操作，故而报错。也就是说，Python 变量的作用域由变量所在源代码中的位置决定。

（1）作用域类型

变量的作用域决定了变量的有效范围空间，也就是在程序的哪些部分可以被使用。

Python 的作用域一般划分成四类，如下：

➤ 局部作用域

局部作用域（Local, L）一般是定义在 `def` 内的变量，也就是说函数内部定义的变量，其只在该函数范围内有效，可以被访问。

➤ 嵌套作用域

嵌套作用域（enclosing,E），也包含在 `def` 关键字中，E 和 L 是相对的，E 相对于更上层的函数而言也是 L。与 L 的区别在于，对一个函数而言，L 是定义在此函数内部的局部作用域，而 E 是定义在此函数的上一层父级函数的局部作用域。主要是为了实现 Python 的闭包，而增加的实现。

➤ 全局作用域

全局作用域（glob, L）在模块中定义的变量的使用范围，可以认为每一个模块都是一

个全局作用域，也就是说每个.py 文件中顶层定义的变量，它的作用范围是整个文件。

➤ 内建作用域

内建作用域（built-in，B）也称为内置作用域，系统内固定模块里定义的变量。

如下代码段，清晰解释了四种类型的变量作用域的关系。

```
flag = int(1.8) #内建作用域
count = 0 #全局作用域
def fact(n):
    count = -1 #嵌套作用域
    def act(n):
        if n <= 0 :
            month = '五月'
            count = 1 #局部作用域
            print('%s 销售额: %d' % (month, count))
        else:
            month = '六月'
            count = 2 #局部作用域
            print('%s 销售额: %d' % (month, count))
    act(n)
```

（2）LEGB 法则

LEGB 法则： 当在函数中使用未确定的变量名时，Python 会按照优先级依次搜索四个作用域，以此来确定该变量名的意义。搜索变量名的顺序级别：局部作用域 > 嵌套作用域 > 全局作用域 > 内建作用域

动动手，大家在程序中试一试，如下代码段的输出结果是什么？仔细回味思考变量的作用域。

```

count = 0#全局作用域
def fact(n):
    count = -1#嵌套作用域
    def act(n):
        if n<= 0 :
            month = '五月'
            count = 1#局部作用域
            print('%s 销售额: %d'%(month, count))
        else:
            month = '六月'
            count = 2#局部作用域
            print('%s 销售额: %d'%(month, count))
    act(n)
fact(3)
print('销售额: ', count)

```

(3) global 和 nonlocal 关键字

在上面这段代码中，那该如何修改 count 的值，使得程序是我们想要的结果呢？需要用到 global 关键字。global 关键字用来在函数或其他局部作用域中使用全局变量。如果在局部要对全局变量进行修改等操作，需要在局部也要先声明该全局变量

```

count = 0#全局作用域
def fact(n):
    count = -1#嵌套作用域
    def act(n):
        global count
        if n<= 0 :
            month = '五月'
            count = 1#局部作用域
            print('%s 销售额: %d'%(month, count))
        else:
            month = '六月'
            count = 2#局部作用域
            print('%s 销售额: %d'%(month, count))
    act(n)
fact(3)
print('销售额: ', count)

```

小贴士

在局部如果不打算修改全局变量，则不声明全局变量，正常使用全局变量和各自的局部变量。

同理，如果要修改嵌套作用域中的变量则需要 `nonlocal` 关键字。动动手，体会一下如下。