第5章 函数和代码复用

目录

CONTENTS

5.3 变量的作用域

目录

CONTENTS

根据程序中变量所在的位置和作用范围，变量分为局部变量和全局变量。局部变量仅在函数内部，且作用域也在函数内部，全局变量的作用域跨越多个函数。

5.3.1 局部变量

局部变量指在函数内部使用的变量，仅在函数内部有效，当函数退出时变量将不再存在。

```
>>>def func(x, y = 10):
        z = x*y    # z是函数内部的局部变量
        return z
>>>s = func(99, 2)
>>>print(s)
198
>>>print(z)
Traceback (most recent call last):
  File "<pyshell#11>", line 1, in <module>
    print(z)
NameError: name 'z' is not defined
```

变量z是函数func()内部使用的变量，当函数调用后，变量z将不存在。

5.3.2 全局变量

全局变量指在函数之外定义的变量，在程序执行全过程有效。全局变量在函数内部使用时，需要提前使用global保留字进行声明，语法形式如下：

global <全局变量>

```
>>>n = 1      #n是全局变量
>>>def func(a, b):
        c = a * b    #c是局部变量，a和b作为函数参数也是局部变量
        return c
>>>s = func("knock~", 2)
>>>print(c)
Traceback (most recent call last):
  File "<pyshell#6>", line 1, in <module>
    print(c)
NameError: name 'c' is not defined
```

这个例子说明，当函数执行完退出后，其内部变量将被释放。

```
>>>n = 1       #n是全局变量
>>>def func(a, b):
          n = b   #n是函数内存中新生成的局部变量，不是全局变量
          return a*b
>>>s = func("knock~", 2)
>>>print(s, n)    #测试一下n值是否改变
knock~knock~ 1
```

函数func()内部使用了变量n，并且将参数b赋值给变量n，为何全局变量n值没有改变？如果未使用保留字global声明，即使名称相同，也不是全局变量。

如果希望让func()函数将n当作全局变量，需要在变量n使用前显式声明该变量为全局变量，代码如下。

```
>>>n = 1      #n是全局变量
>>>def func(a, b):
        global n
        n = b      #将局部变量b赋值给全局变量n
        return a*b
>>>s = func("knock~", 2)
>>>print(s, n)   #测试一下n值是否改变
knock~knock~ 2
```

总结

总结

- 简单数据类型变量无论是否与全局变量重名，仅在函数内部创建和使用，函数退出后变量被释放。
- 简单数据类型变量在用global保留字声明后，作为全局变量使用。

THANKS