

变量的作用域

在Python程序中创建、改变、查找变量名时，都是在一个保存变量名的空间中进行，我们称之为命名空间，也被称之为作用域。python的作用域是静态的，在源代码中变量名被赋值的位置决定了该变量能被访问的范围。即Python变量的作用域由变量所在源代码中的位置决定。

只有当变量在Module(模块)、def(函数)中定义的时候，才会有作用域的概念。在作用域中定义的变量，一般只在作用域中有效。需要注意的是：在if-elif-else、for-else、while、try-except\try-finally等关键字的语句块中并不会产生作用域。

注：

1. 在Python中，scope是由namespace按特定的层级结构组合起来的。
scope一定是namespace，但namespace不一定是scope。

来自 <http://python.jobbole.com/81367/>

1. class没有作用域(scope)，但有一个局部的名空间(namespace)，它并不构成一个作用域。

这意味着在类定义中的表达式可以访问该名空间。

搜索变量名的优先级：局部作用域 > 嵌套作用域 > 全局作用域 > 内置作用域

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4个作用域，以此来确定该变量名的意义。首先搜索局部作用域(L)，之后是上一层嵌套结构中def或lambda函数的嵌套作用域(E)，之后是全局作用域(G)，最后是内置作用域(B)。按这个查找原则，在第一处找到的地方停止。如果没有找到，则会出发NameError错误。

1. L(local)局部作用域

局部变量：包含在def关键字定义的语句块中，即在函数中定义的变量。每当函数被调用时都会创建一个新的局部作用域。Python中也有递归，即自己调用自己，每次调用都会创建一个新的局部命名空间。在函数内部的变量声明，除非特别的声明为全局变量，否则均默认为局部变量。有些情况需要在函数内部定义全局变量，这时可以使用global关键字来声明变量的作用域为全局。局部变量域就像一个 栈，仅仅是暂时的存在，依赖创建该局部作用域的函数是否处于活动的状态。所以，一般建议尽量少定义全局变量，因为全局变量在模块文件运行的过程中会一直存在，占用内存空间。

注意：如果需要在函数内部对全局变量赋值，需要在函数内部通过global语句声明该变量为全局变量。

1. E(enclosing)嵌套作用域

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，对于一个函数而言，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。

1. G(global)全局作用域

即在模块层次中定义的变量，每一个模块都是一个全局作用域。也就是说，在模块文件顶层声明的变量具有全局作用域，从外部开来，模块的全局变量就是一个模块对象的属性。

注意：全局作用域的作用范围仅限于单个模块文件内

1. B(built-in)内置作用域

系统内固定模块里定义的变量，如预定义在builtin 模块内的变量。

不要使用 `from xx import *`，会把xx模块中所有东西(变量, 函数, 类, 等等)(以 `_` 开头的变量, 函数不会引入)引入变为全局变量, 就不知道引入了什么, 怕 会引起作用域的冲突