**答案**：

[0, 1]

[3, 2, 1, 0, 1, 4]

[0, 1, 0, 1, 4]

第一个函数调用十分明显，for循环先后将0和1添加至了空列表l中。l是变量的名字，指向内存中存储的一个列表。

第二个函数调用在一块新的内存中创建了新的列表。l这时指向了新生成的列表。之后再往新列表中添加0、1、2和4。

第三个函数调用的结果。它使用了之前内存地址中存储的旧列表。这就是为什么它的前两个元素是0和1了。

不明白的话就试着运行下面的代码吧：

l\_mem = []

l = l\_mem # the first call

for i in range(2):

l.append(i\*i)

print l # [0, 1]

l = [3,2,1] # the second call

for i in range(3):

l.append(i\*i)

print l # [3, 2, 1, 0, 1, 4]

l = l\_mem # the third call

for i in range(3):

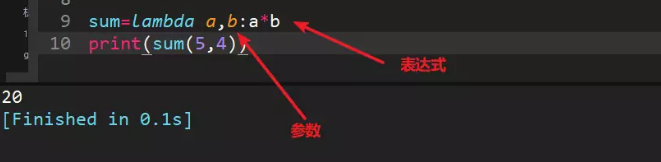
l.append(i\*i)

print l # [0, 1, 0, 1, 4]

**如何在一个函数内部修改全局变量**

**利用global 修改全局变量**

**用lambda函数实现两个数相乘**



**列表推导式求列表所有奇数并构造新列表，a =  [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]**

