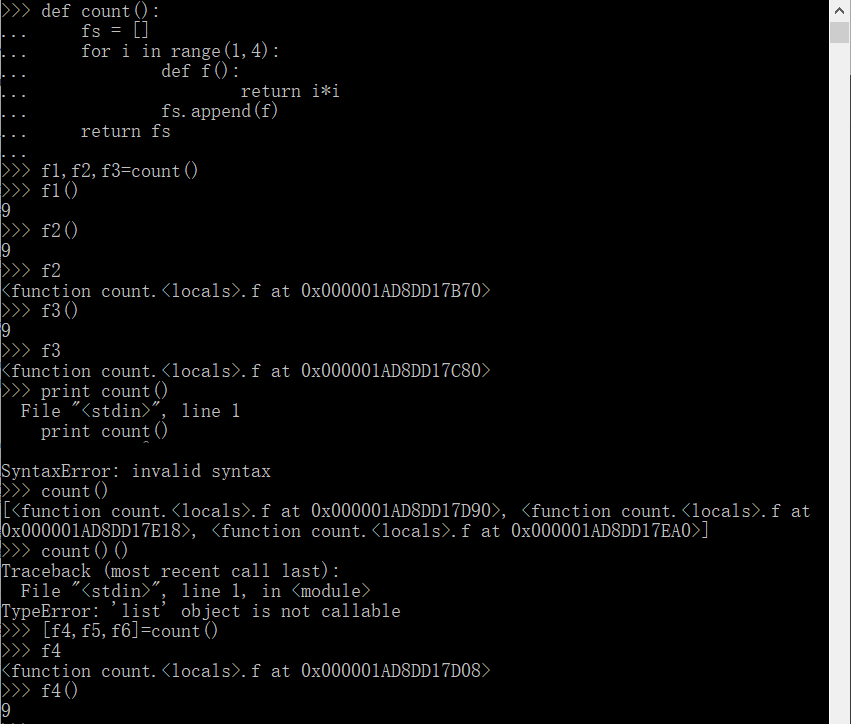
一、错误代码解析：

1. 首先，return fs是返回一个fs（list来的），list里面有3个元素，都是f的地址（这里定义了三次f函数，所以地址都不同），所以如果这么输入：

>>>count()

[<function count.<locals>.f at 0x000001AD8DD17D90>,<function count.<locals>.f at 0x000001AD8DD17E18>,<function count.<locals>.f at 0x000001AD8DD17EA0>]

1. 再来看看代码f1,f2,f3=count()，其实这相当于[f1,f2,f3]=count()，不信你可以试试，效果是一样的
2. 最后我们来讨论一下为什么三个f1(),f2(),f3()返回的值都是9？

由于当我们把函数f加入到fs这个list上时，是按照当i=1时，fs=[f]；i=2时，fs=[f,f]；i=3时，fs=[f,f,f]。然后就停止了，i定个在3这个值里面。后来当我们调用f1()f2()f3()时，用的都是i=3.意思是i分别等于123时，是我们在list的末尾加入f函数的过程，后来调用f()的时候，i已经停止循环了

二、正确代码解析：

1. 更改后的代码中f(j)的功能时返回一个能够计算j\*j的函数（由于后面的append需要加入的是函数而非函数值，这样就巧妙地解决了）
2. 而r=f(i)已经提前把i传入f函数里面，这样就确保不会因循环变量的改变而改变了g()的值了

