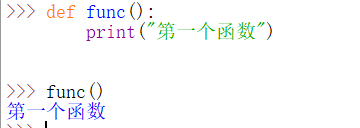
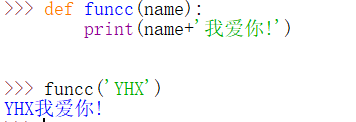
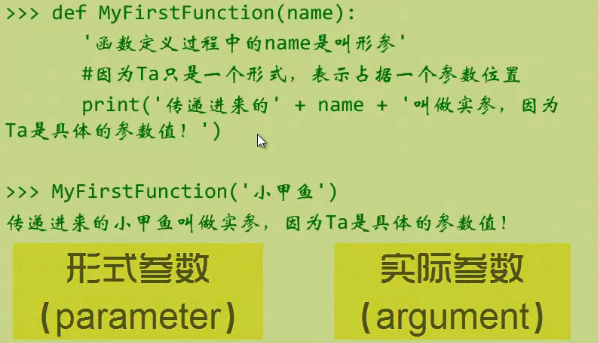
1. 函数

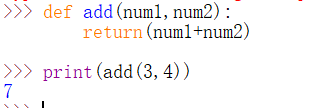


def 函数名（）：函数体

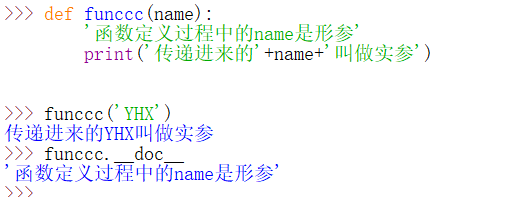
调用： 函数名（）

函数的参数name为参数



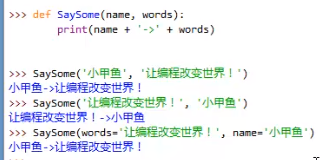
函数的返回值 

1. 函数文档

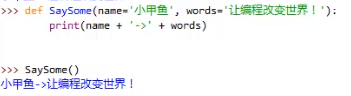


双下划线 文档实际是函数体中的注释 或 help（函数名）或 print.\_\_doc\_\_

1. 关键字参数

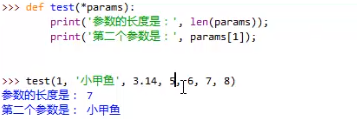


1. 默认参数

后续可调用函数改变参数

1. 收集参数

在参数前加 \* 号



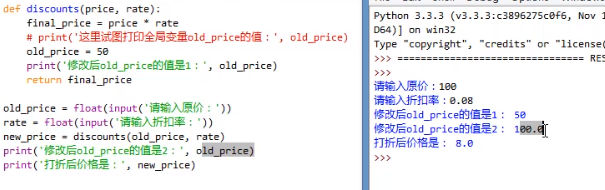
把所有参数打包成元组赋值给 收集参数

函数除收集参数外还有其他参数时 需要为其他参数设置默认值或初始值

1. 返回值

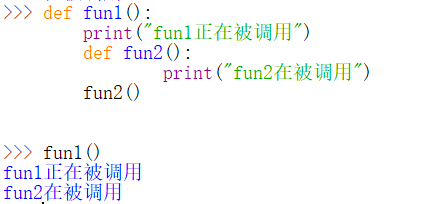
默认返回元组 返回多个值时实际是利用元组打包返回

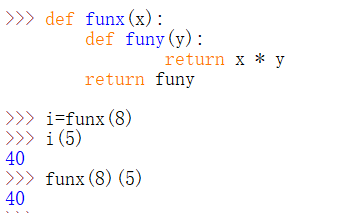
1. 局部变量 全局变量



若在函数中试图修改全局变量的值 py会生成一个名字一样的局部变量 进行修改

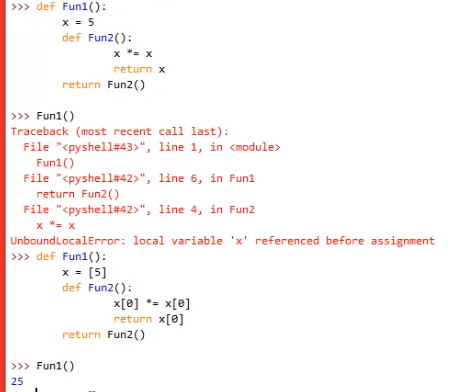
1. 内嵌函数和闭包

内嵌函数： 在全局中调用fun2（）会报错

闭包 ：

funy是一个闭包

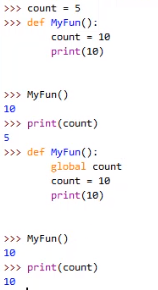
一个内部函数 引用了外部函数的变量 即可构成闭包



fun1里面的X被当作全局变量  fun2中要访问X并修改全局变量的值是不可以的 产生了屏蔽

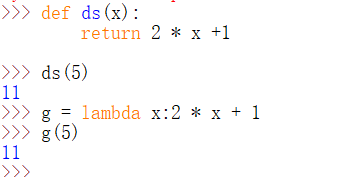
列表 是 全局变量 可以直接使用

1. 可以使用 global 关键字在函数中修改全局变量的值



1. 匿名函数

Lambda 表达式

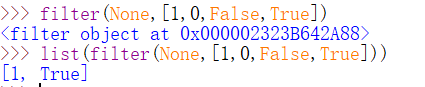


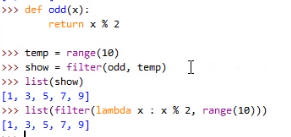
lambda x:2 \*x +1 前面是函数的参数 后面是返回值表达式

多个参数： lambda x,y : x+y

1. 两个常用BIF

Filter（） 过滤器 筛选掉非True的元素

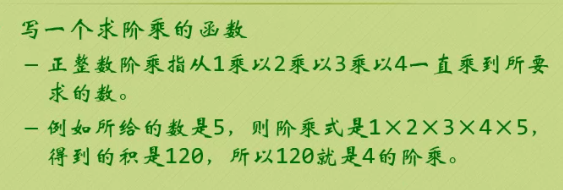




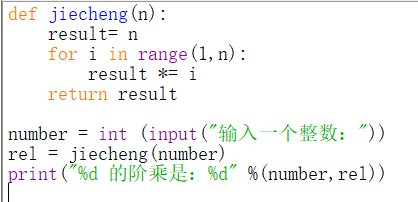
Map（） 将后一个生成的数字 赋给前面的参数 进行函数操作后打印出来



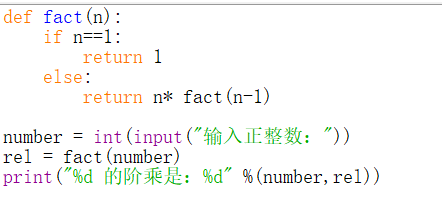
1. 递归



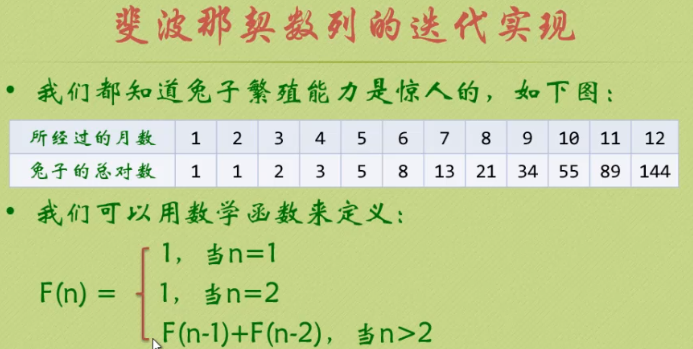
非递归形式：

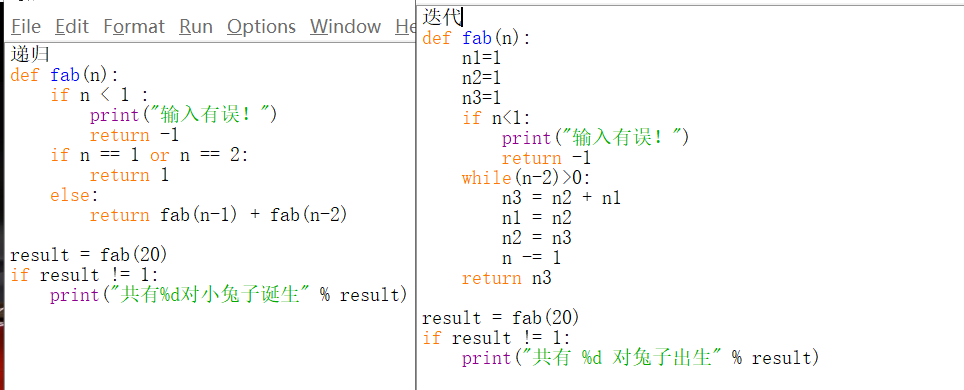


递归： 正确的返回形式 自身调用自身 默认递归最多100层

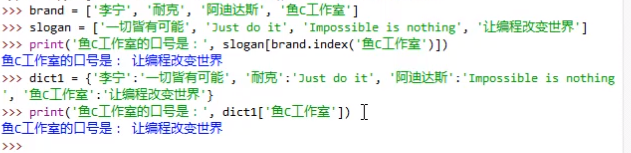


Eg：斐波那契数列



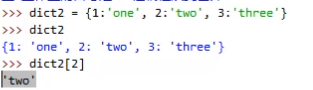


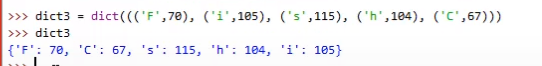
1. 汉诺塔
2. 字典



上面是列表的对应关系 下面使用字典 字典={‘key’：‘value’}

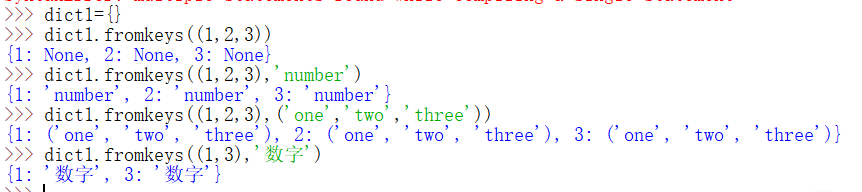
字典的创建：





Dict（） 函数创建字典

Fromkeys()： 创建字典 若想修改值时 是创建一个新的字典第二个参数不指定时 默认为None

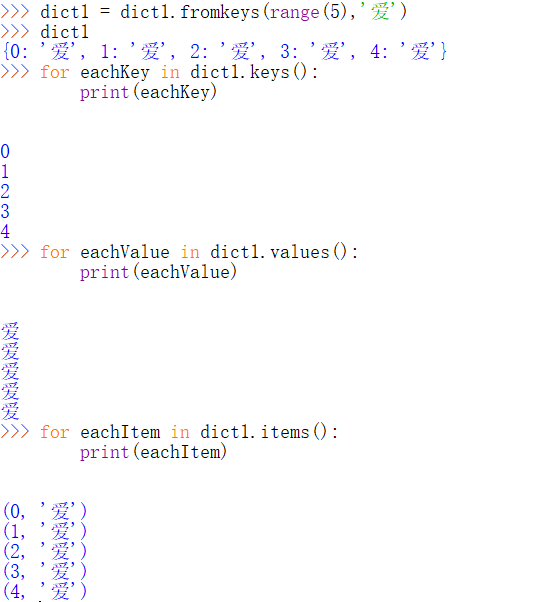


EachKey

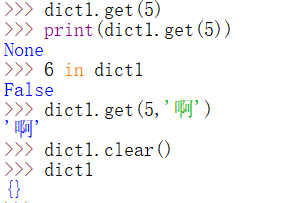
EachValue

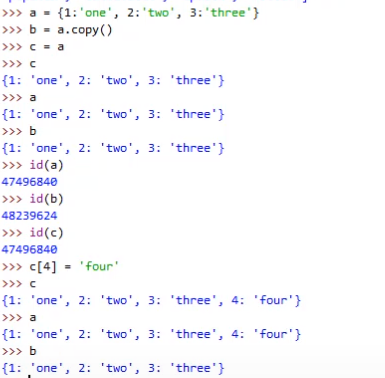
EachItem

：



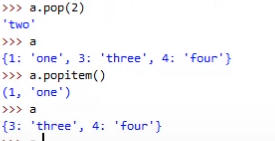
Get （） clear（）



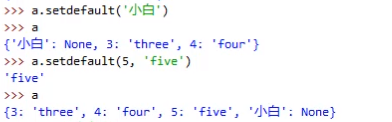


Pop（） 给出指定位置上的值 并删除

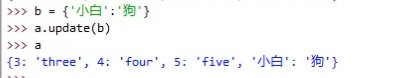
Popitem（） 弹出首位键值



Setdefault（）

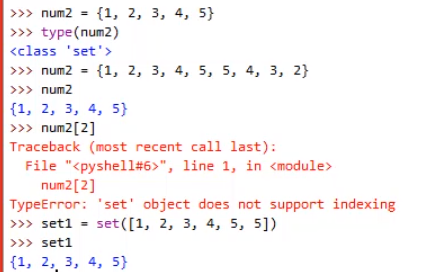


Update（）

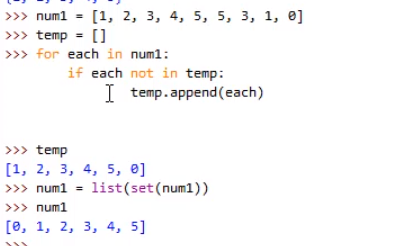


1. 集合

元素唯一 无序 不支持索引

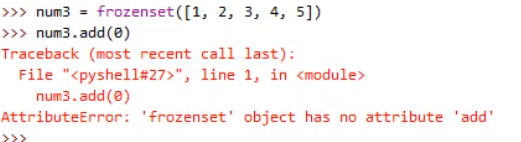
set（） 创建

去掉重复元素



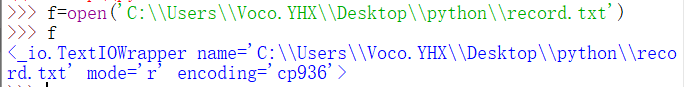
1. 不可变集合

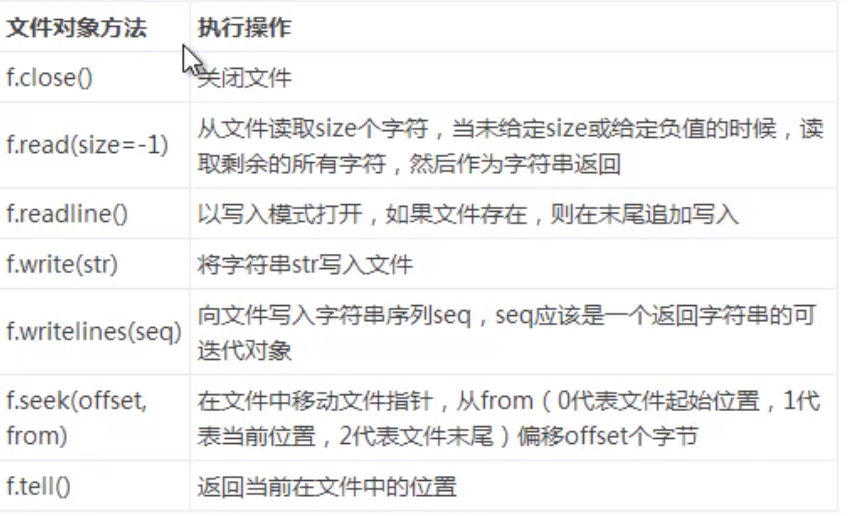
不能随意修改和增加删除等



1. 文件







1. 文件分别保存

