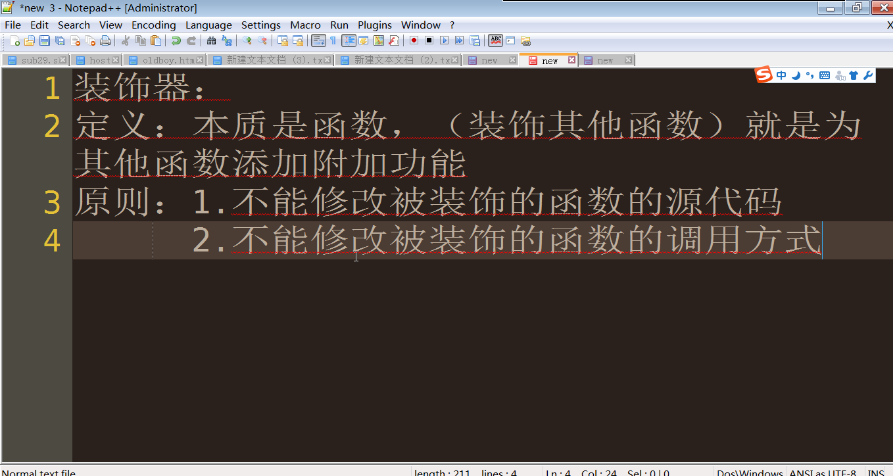
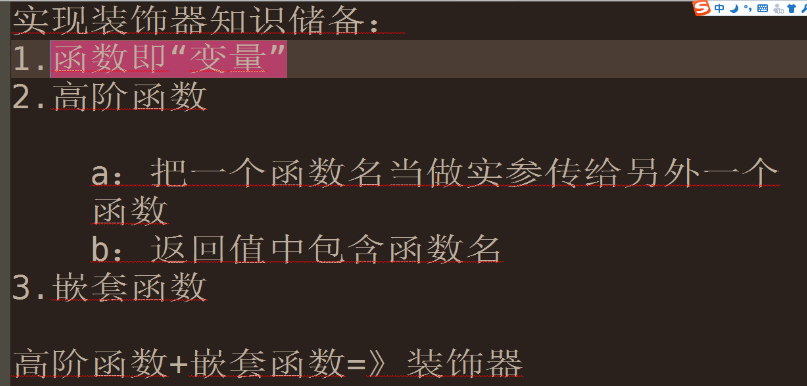
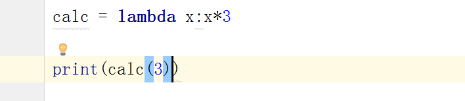
**1.装饰器(很有用)**

安装pycharm断点跑一次decorator1-4+高潮版

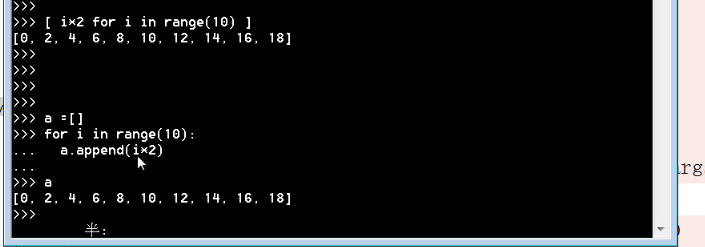


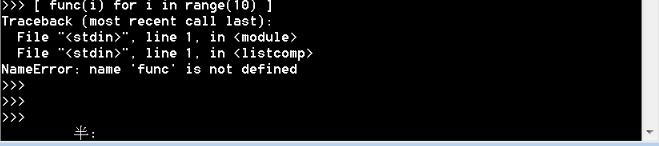


**2匿名函数**



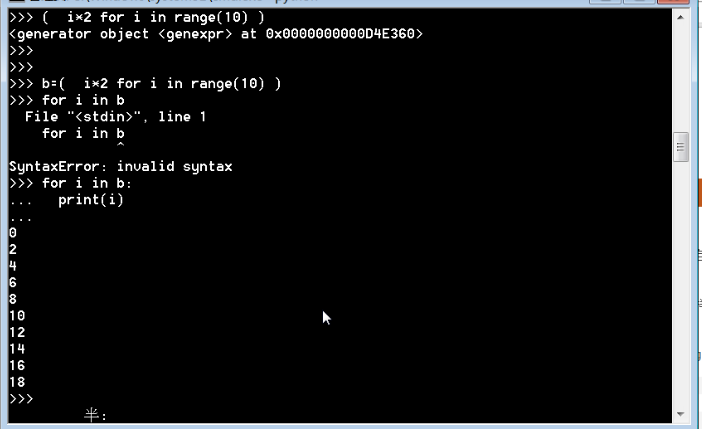
**3.列表生成式 ，全部生成供调用。**





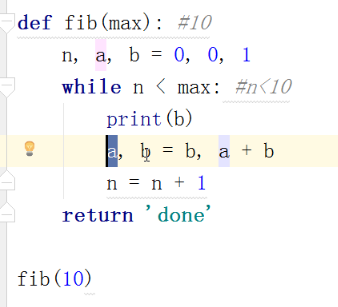
**列表生成器generator**，调用时候生成。不支持“切片”

只记住当前位置，只能 .\_ \_next\_ \_()或者next()取下一个值。

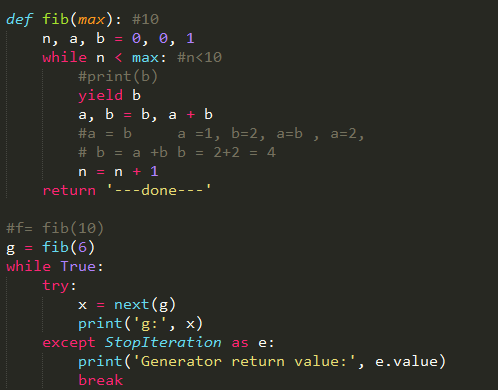


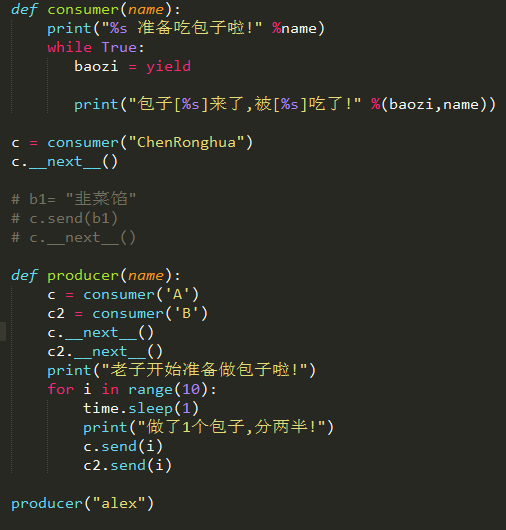
generator应用：斐波那契

赋值语句： a,b = b, a+b



**Yield将函数变成生成**器，return没作用。。yield位置很关键，保存函数的中断状态。





**迭代器Iterator，迭代对象Iterable(list，dict，str，set，tuple等)**

生成器都是迭代器对象，

list，dict，str是迭代对象但不是迭代器，可以使用iter()变成迭代器

json序列化和反序列化,,只能转化简单的数据类型..优点：跨语言

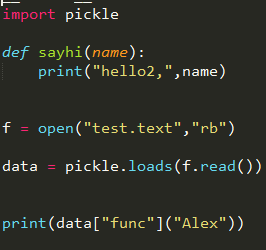
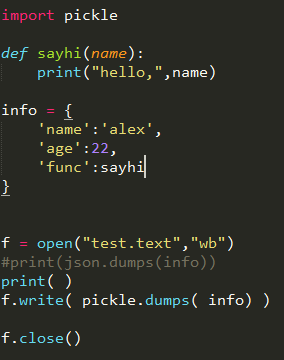
json.dump()

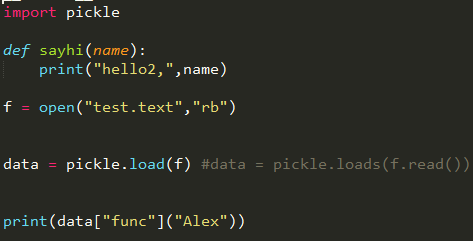
json.load()

pickle序列化和反序列化。。python使用

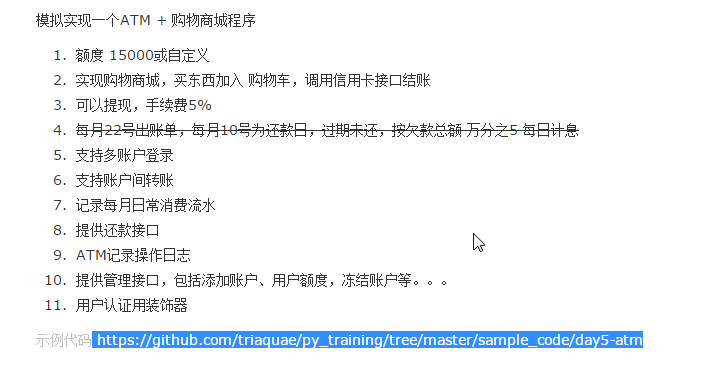
pickle.dump()

pickle.load()





**任务：**



**4 模块**

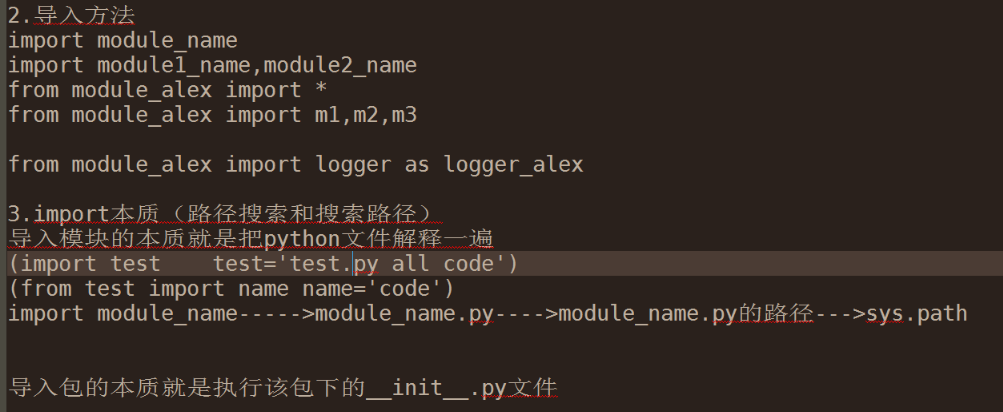
From import和import的区别

From 模块import方法/变量 （as 方法/变量新名）。（在使用方法/变量时，直接使用）

Import 模块。（使用模块方法/变量时，需要：模块.方法/变量）

导入包：执行包目录下的\_\_init\_\_.py。在\_\_init\_\_.py中导入当前目录的其他模块，该包再被

导入时，会加载导入的模块



导入优化：使用from import在py文件头导入方法，避免多次寻找模块的耗时

**5.时间**：

1. 时间戳 time.time()

元组转化为时间戳：mktime()

1. Struct\_time(tuple)

时间戳转化为元组的形式：   
time.gmtime() UTC时间

time.localtime() 本地时间  
格式化字符串转化为元组的形式：

time.strptime(字符串,格式(与字符串格式一一对应))

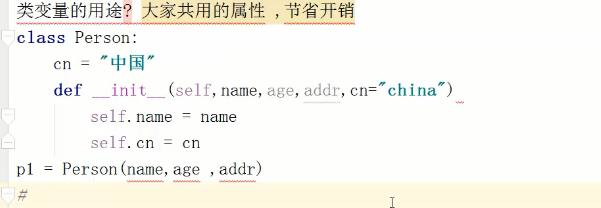
1. 格式化的字符串(格式自由设定)  
   （固定格式的字符串:%a %b %d %H:%M:%S %Y） time.asctime()元组转化为字符串 time.ctime()时间戳转化为字符串  
   元组转化为格式化字符串的格式（自定义格式）

time.strftime(格式,元组) eg.time.strftime(“%Y-%m-%d %H:%M:%S”,time.localtime())

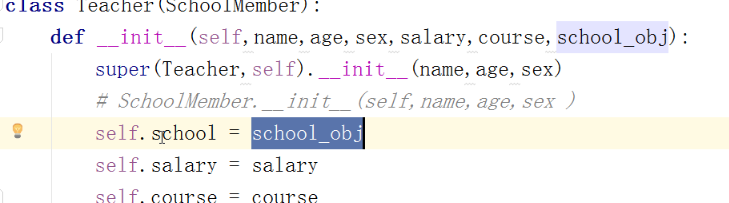
**datetime**

datetime.datetime.now():格式化的字符串

类变量



继承或者组合的方式



组合的方式将teacher和school关联

**MySQL**



**修改id为自增状态**

