Python学习笔记

Day1（8月21日）

一．安装Python

Window下Python安装极为简单，只需前往官网下载相应安装包安装即可，此处提供两个网址

1. Python官网：

https://www.python.org/

2.国内镜像：https://pan.baidu.com/s/1kU5OCOB#list/path=%2Fpub%2Fpython%2Fpycharm&parentPath=%2F

注意在安装时一定要勾选Add Python 3.7 to PATH选项。

二．安装Sublime Text2

Sublime Text2是一款非常好用的文本编辑器，Word和Windows自带的记事本。Word保存的不是纯文本文件，而记事本会在文件开始的地方加上几个特殊字符（UTF-8 BOM），结果会导致程序运行出现错误。

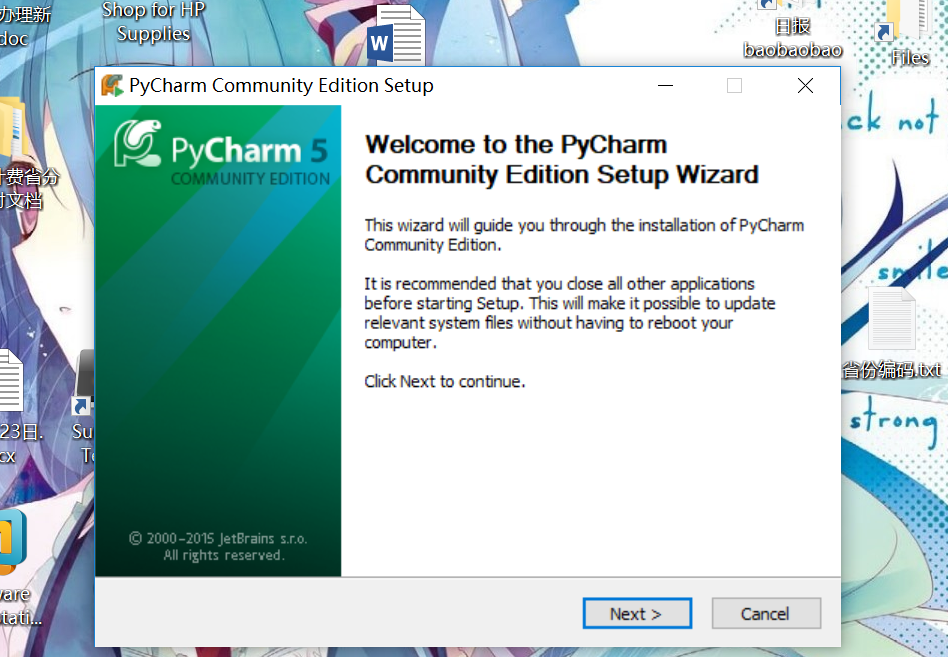
注意在使用编辑器编辑Python代码时不要有空格

三．安装Pycharm

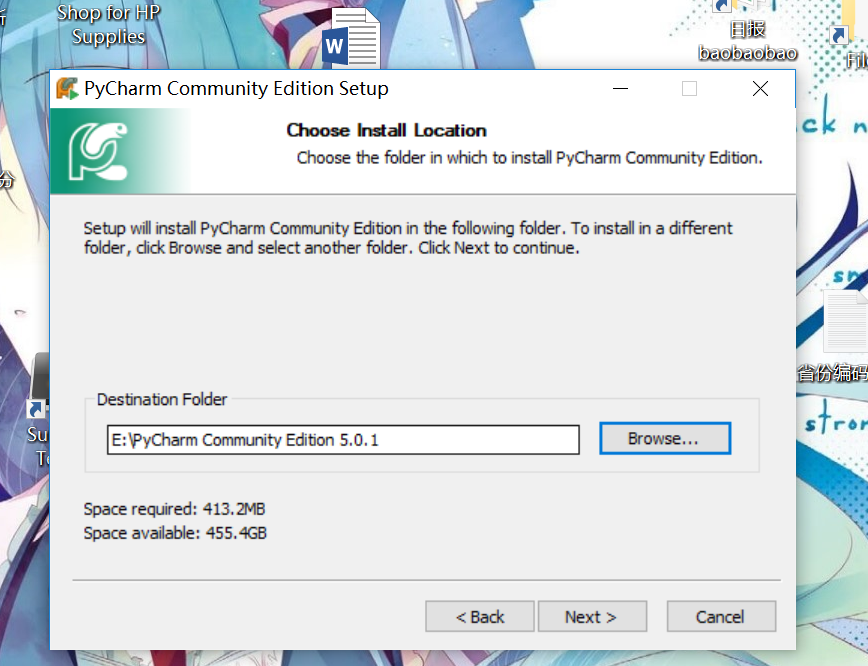
Pycharm是非常好用的Python IDE，由于其安装与配置步骤较为繁琐，接下来展示安装配置流程：

1.下载相关安装包，可以在国内镜像网址中找到

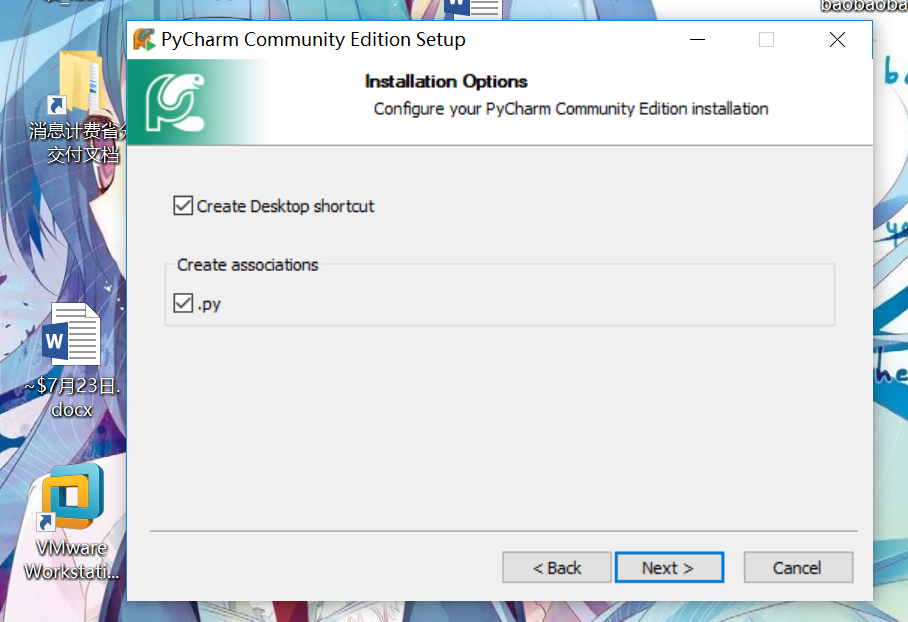
2.开始安装



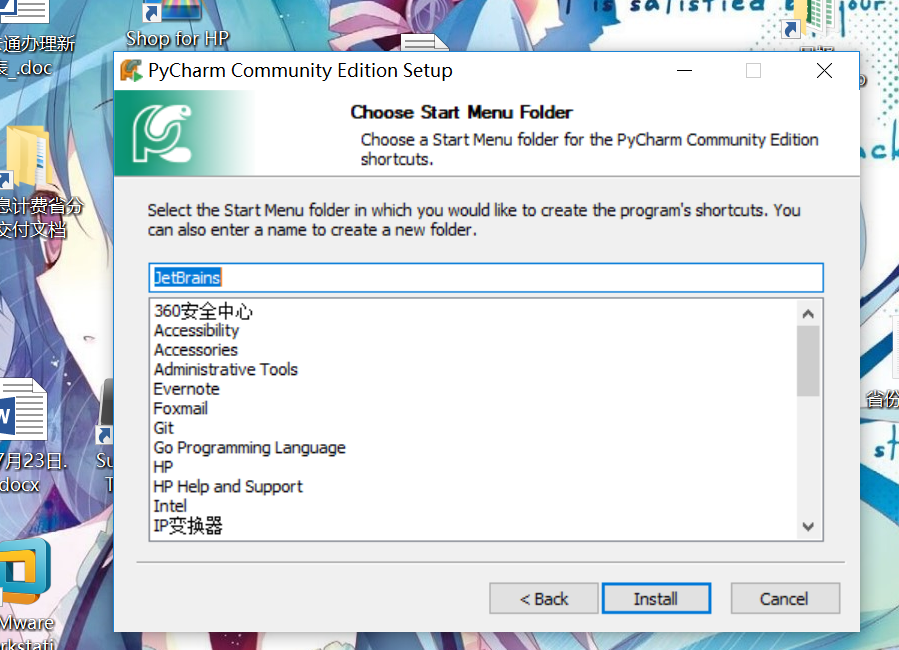
选择next 继续安装



选择好安装的磁盘后继续安装

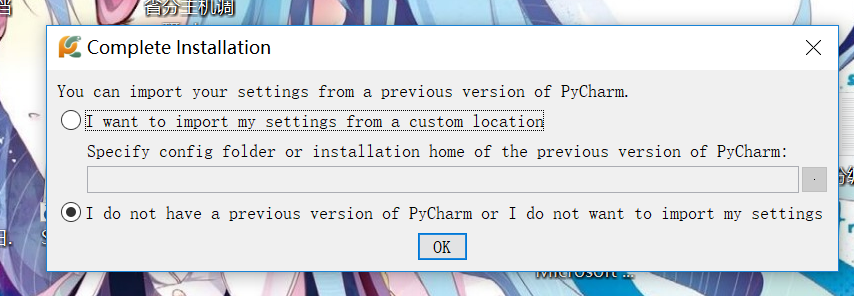


创建桌面快捷方式的同时，使用Pycharm打开所有扩展名为.py的文件

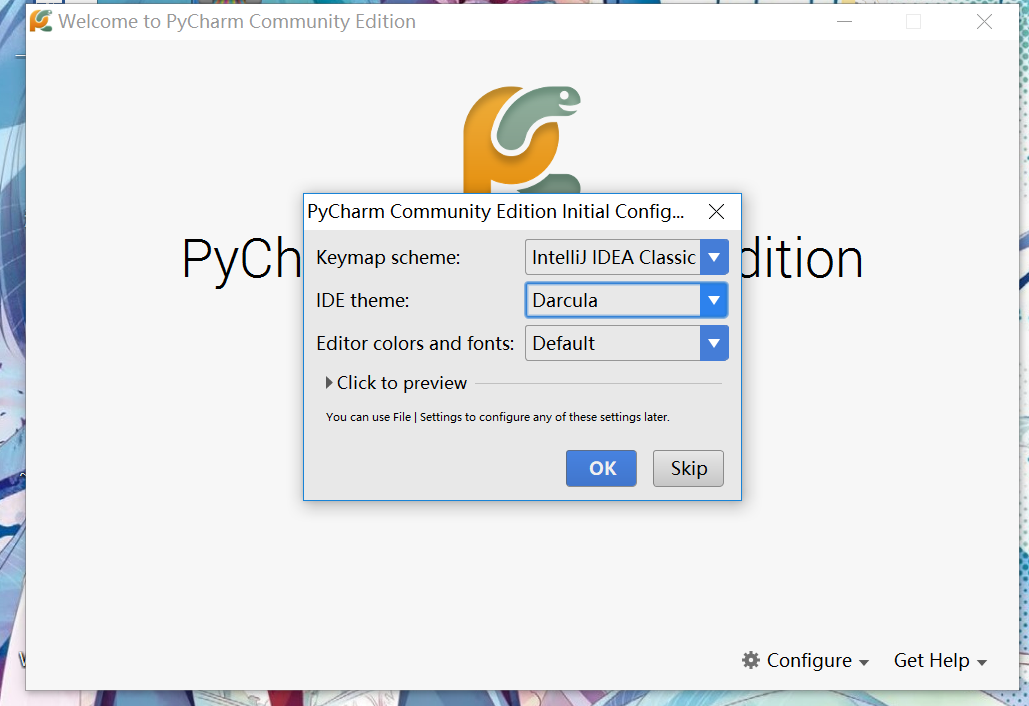


选择默认的选项继续即可安装完成

下面开始配置Pycharm



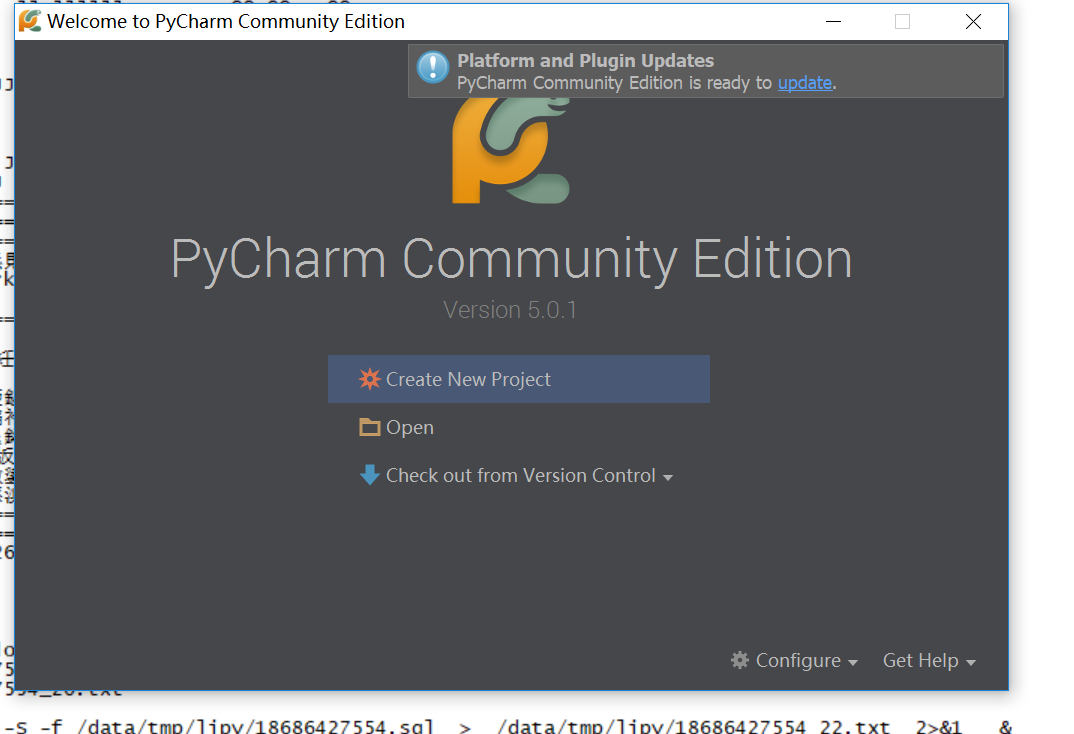
选择从未设置过并进入下一步



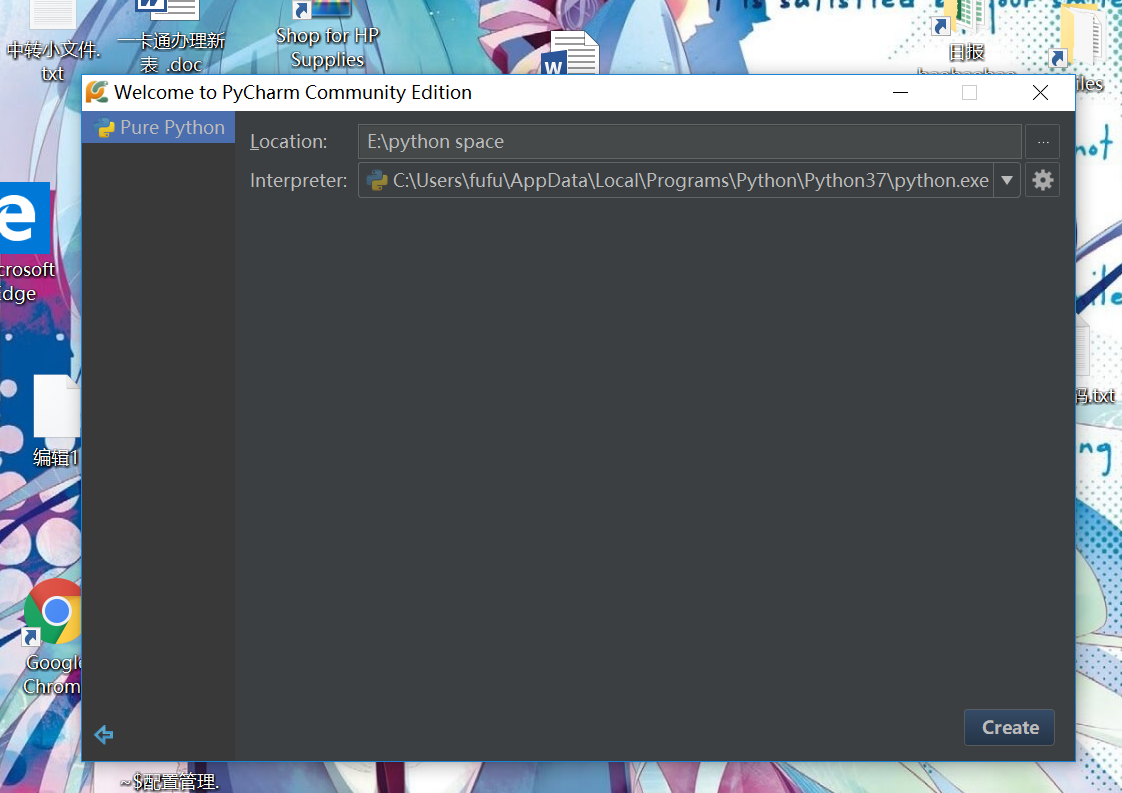
进入主题选择界面 我选择的是Darcula主题（可以在使用过程中随时更改）

完成后重启IDE

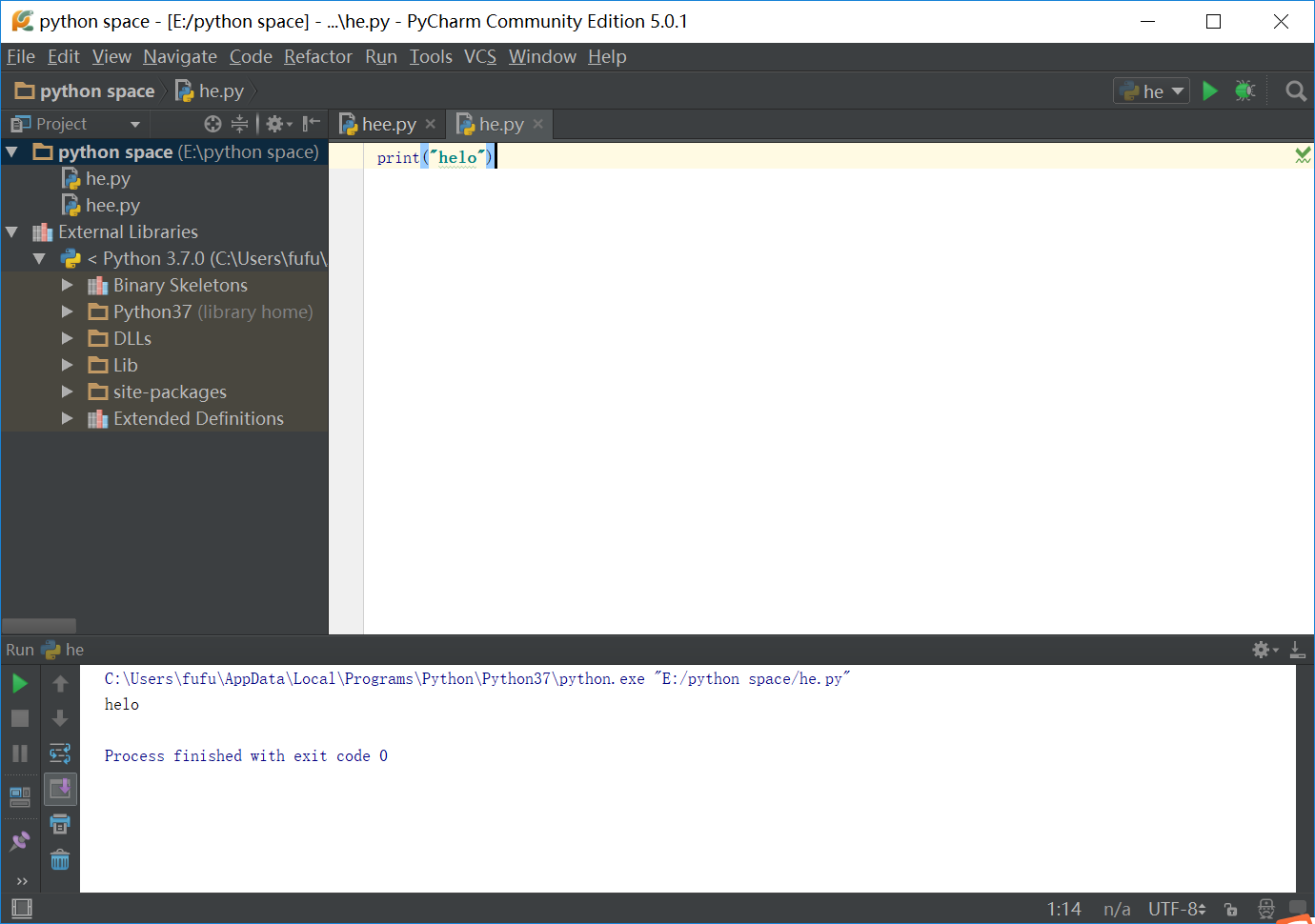
重新打开后，如图所示



我们选择创建新项目



自定义工作空间



确定后新建并开始使用Pycharm

Day2（8月22日）

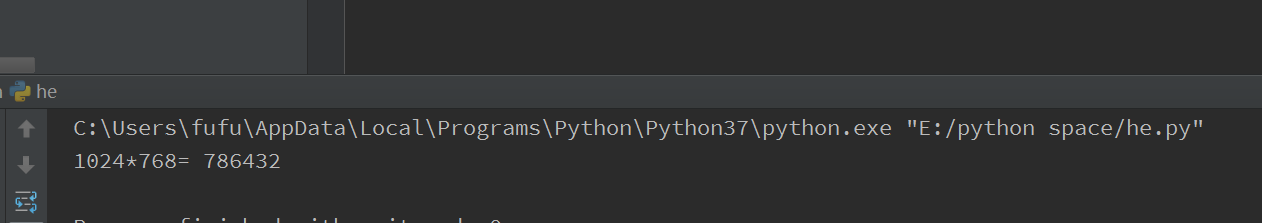
一．输入与输出练习

请利用print()输出1024 \* 768 = xxx：

源代码如下

print("1024\*768=",1024\*768)

运行结果如下



二．数据类型和变量练习

请打印出以下变量的值：

# -\*- coding: utf-8 -\*-

n = 123

f = 456.789

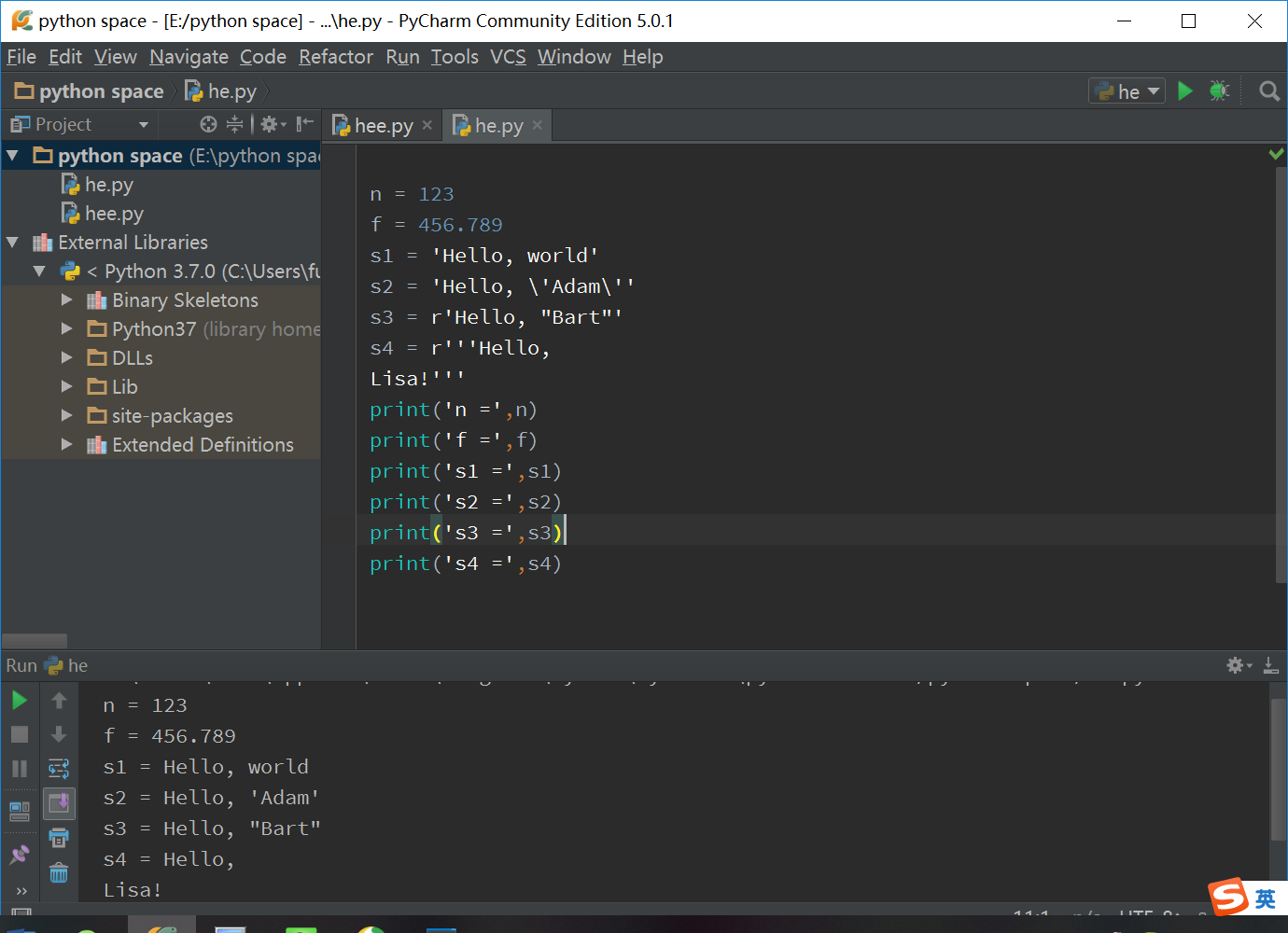
s1 = 'Hello, world'

s2 = 'Hello, \'Adam\''

s3 = r'Hello, "Bart"'

s4 = r'''Hello,

Lisa!'''



三．输入与输出练习

源代码

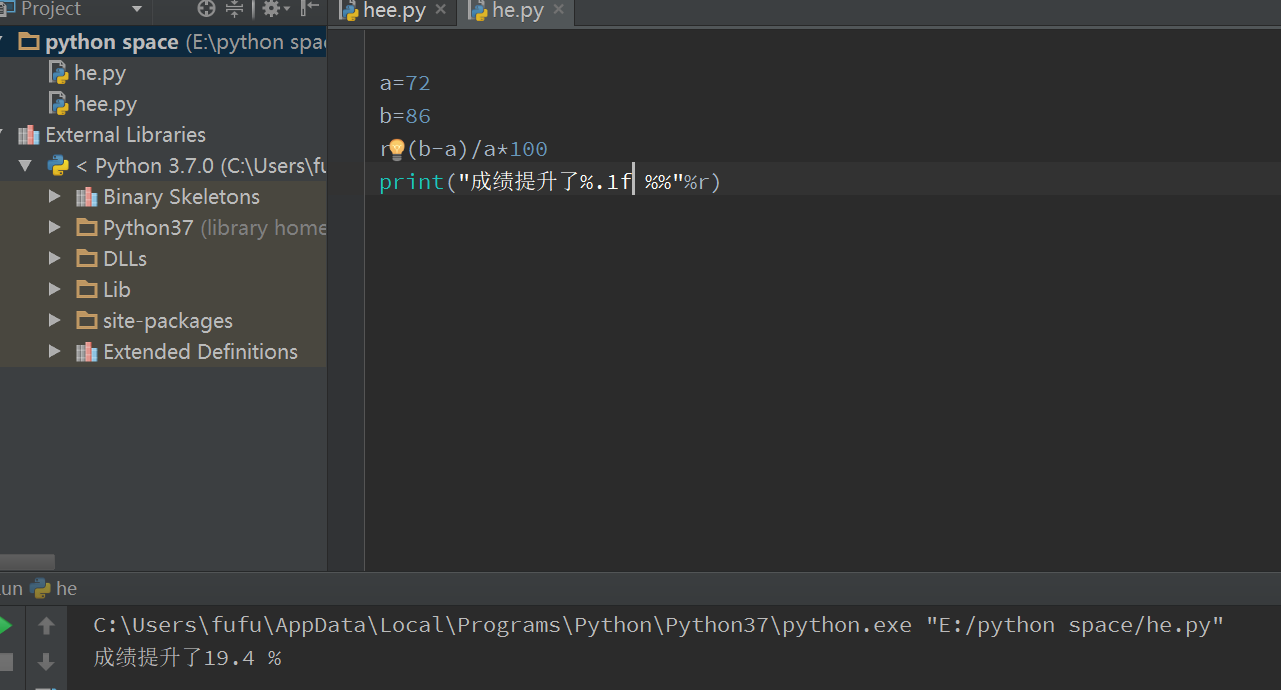
a=72

b=86

r=(b-a)/a\*100

print("成绩提升了%.1f %%"%r)

运行结果展示



四．Lsit与tuple练习

请用索引取出下面list的指定元素：

# -\*- coding: utf-8 -\*-

L = [

['Apple', 'Google', 'Microsoft'],

['Java', 'Python', 'Ruby', 'PHP'],

['Adam', 'Bart', 'Lisa']

]

# 打印Apple:

print(?)

# 打印Python:

print(?)

# 打印Lisa:

print(?)

源代码

L = [

['Apple', 'Google', 'Microsoft'],

['Java', 'Python', 'Ruby', 'PHP'],

['Adam', 'Bart', 'Lisa']

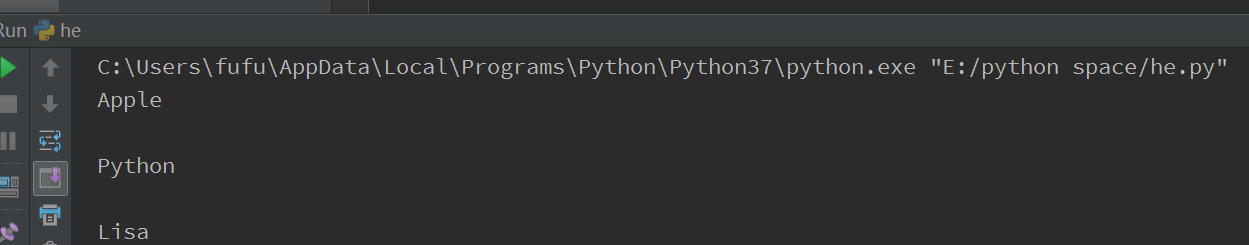
]

print(L[0][0],"\n")

print(L[1][1],"\n")

print(L[2][2])

实际运行结果



五．条件判断练习

小明身高1.75，体重80.5kg。请根据BMI公式（体重除以身高的平方）帮小明计算他的BMI指数，并根据BMI指数：

低于18.5：过轻

18.5-25：正常

25-28：过重

28-32：肥胖

高于32：严重肥胖

用if-elif判断并打印结果：

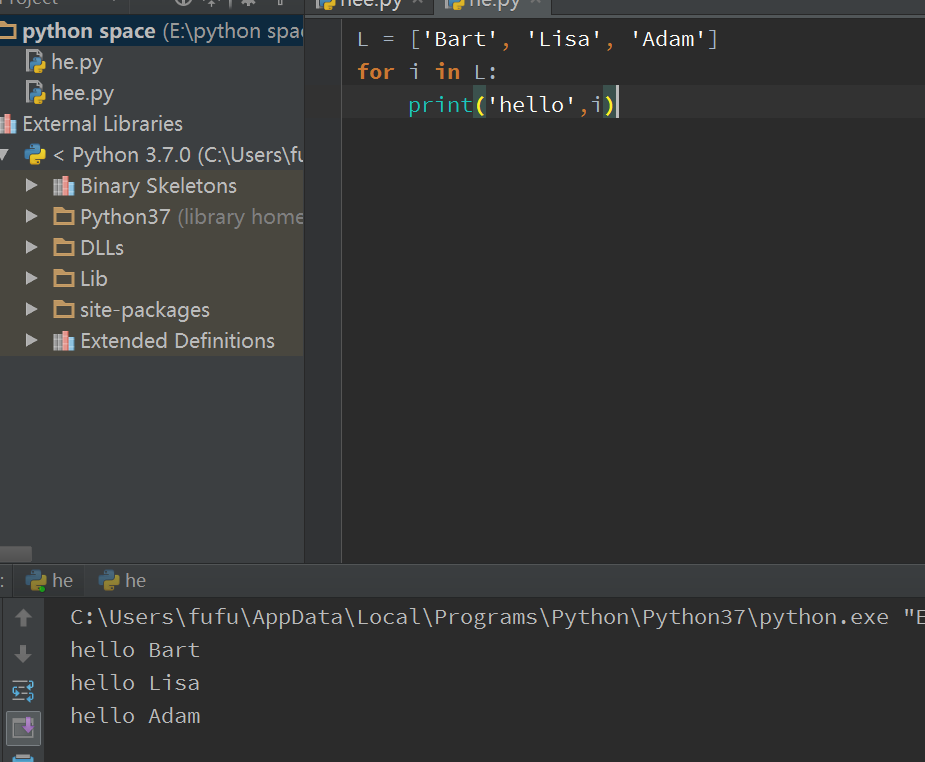
实际运行结果



六．循环练习

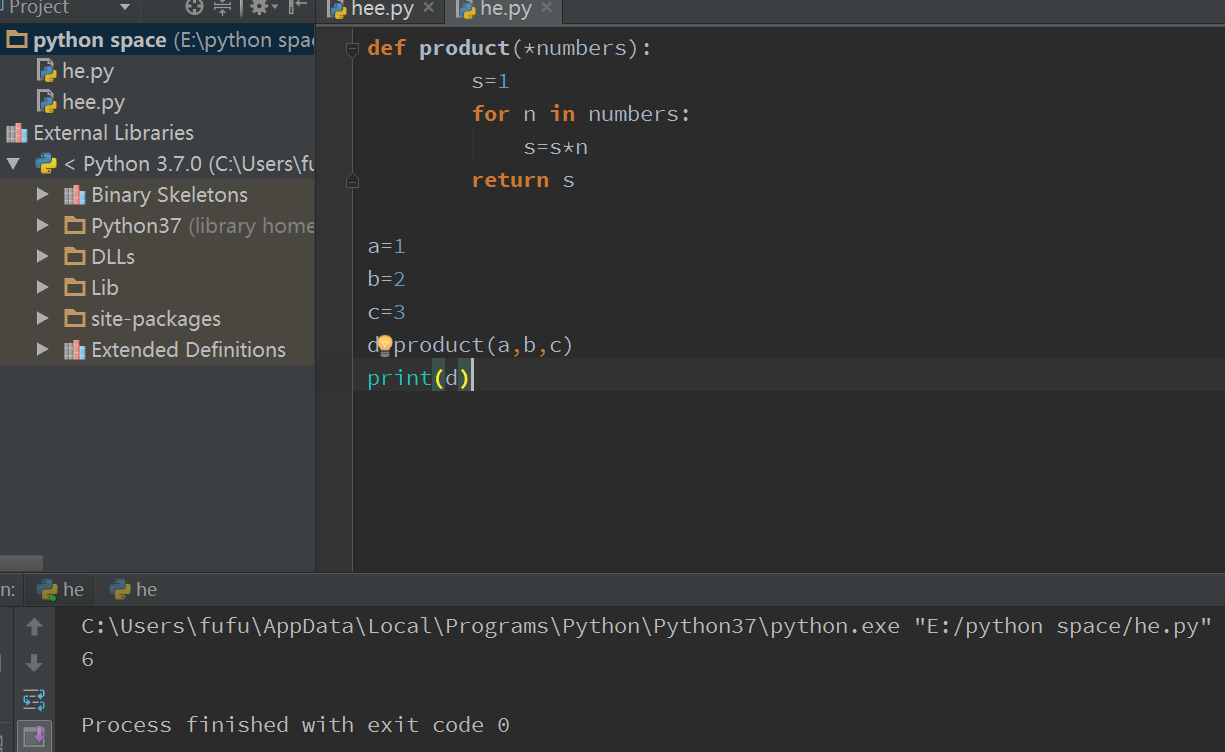
请利用循环依次对list中的每个名字打印出Hello, xxx!：

实际运行结果：



七．循环练习

以下函数允许计算两个数的乘积，请稍加改造，变成可接收一个或多个数并计算乘积：



八．递归练习

写一个汉诺塔

def move(n,a,b,c):

if n == 1:

print(a,"上的盘子移动至",c)

elif n == 2:

print(a,"上的盘子移动至",b)

print(a,"上的盘子移动至",c)

print(b,"上的盘子移动至",c)

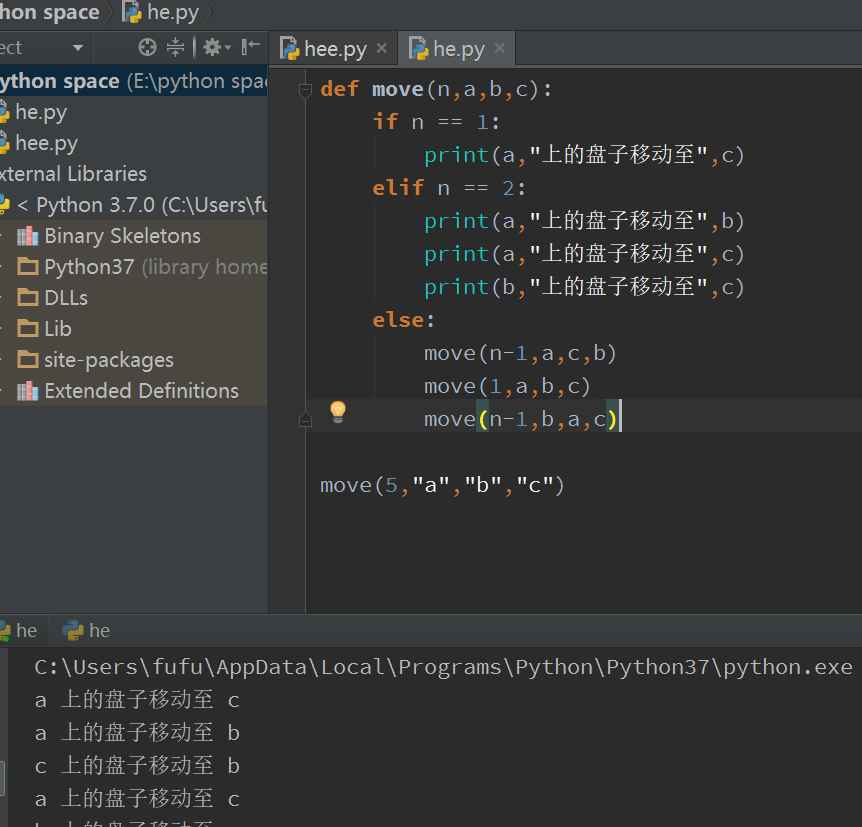
else:

move(n-1,a,c,b)

move(1,a,b,c)

move(n-1,b,a,c)

move(5,"a","b","c")



九．切片练习

利用切片操作，实现一个trim()函数，去除字符串首尾的空格，注意不要调用str的strip()方法：

原代码

def trim(s):

if s[:1] != " " and s[-1:] != " ":

return s

elif s[:1] == " ":

s = s[1:]

return trim(s)

elif s[-1:] == " ":

s = s[:-1]

return trim(s)

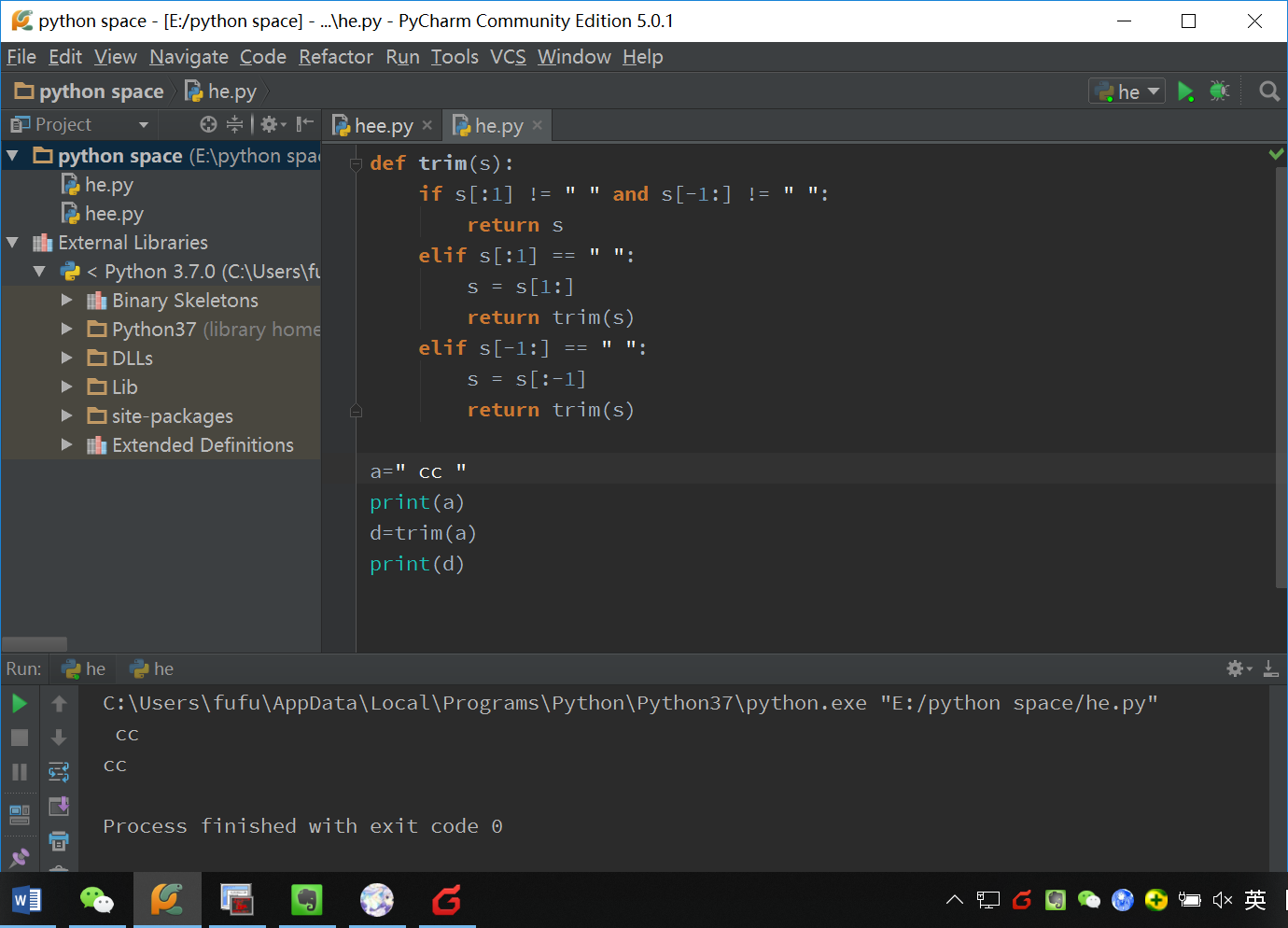
a=" cc "

print(a)

d=trim(a)

print(d)

运行结果



十．迭代练习

请使用迭代查找一个list中最小和最大值，并返回一个tuple：

代码为

def cc(L):

for a in L:

for b in L:

if a == min(L) and b == max(L):

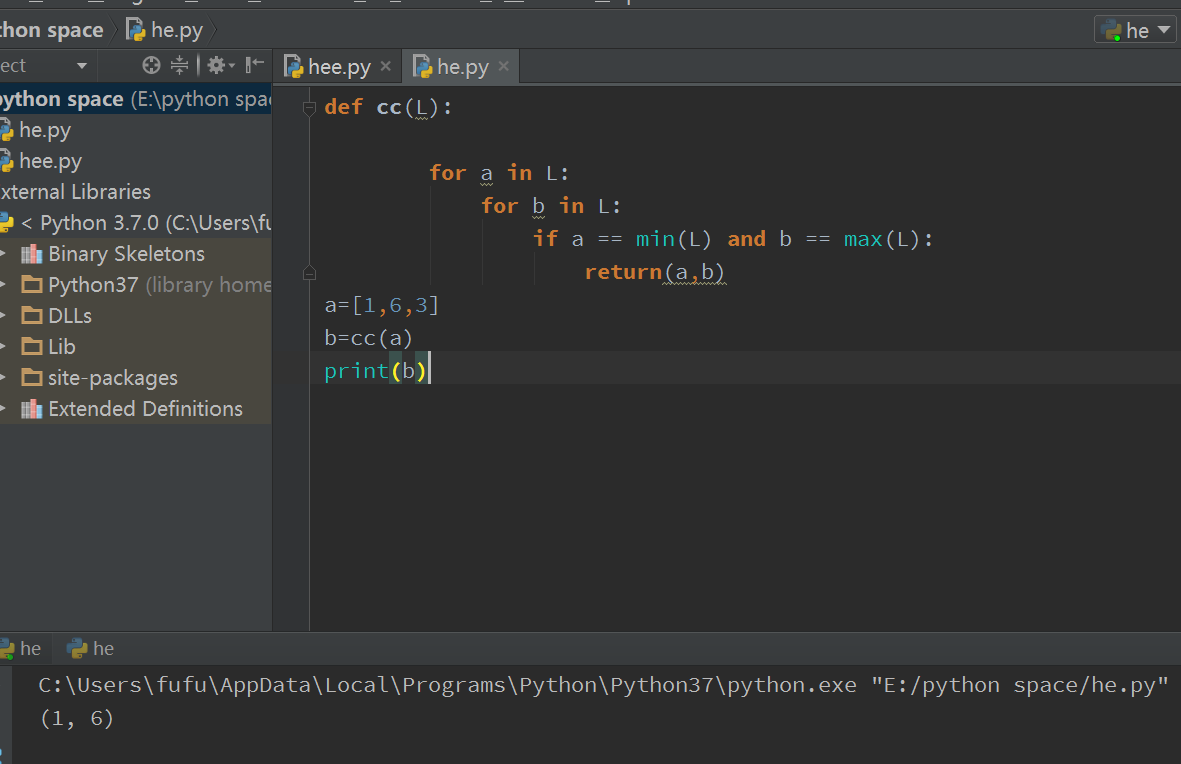
return(a,b)

a=[1,6,3]

b=cc(a)

print(b)

运行结果图为



十一．列表生成式练习

如果list中既包含字符串，又包含整数，由于非字符串类型没有lower()方法，所以列表生成式会报错：请修改列表生成式，通过添加if语句保证列表生成式能正确地执行：

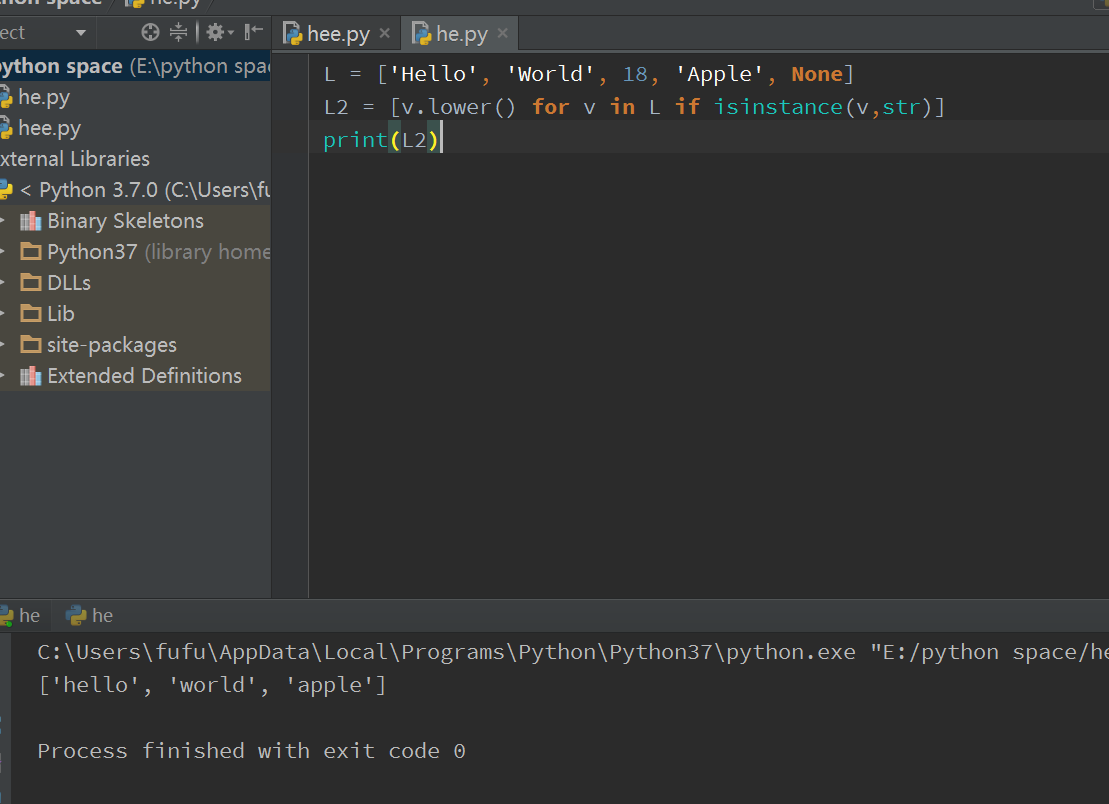
源代码：

L = ['Hello', 'World', 18, 'Apple', None]

L2 = [v.lower() for v in L if isinstance(v,str)]

print(L2)

运行结果图为



十一．生成器练习

使用生成器复现杨辉三角形

源代码

def d (n):

L = [1]

i=1

while i<n:

yield(L)

c= len(L)-1

k = [L[j]+L[j+1] for j in range(c)]

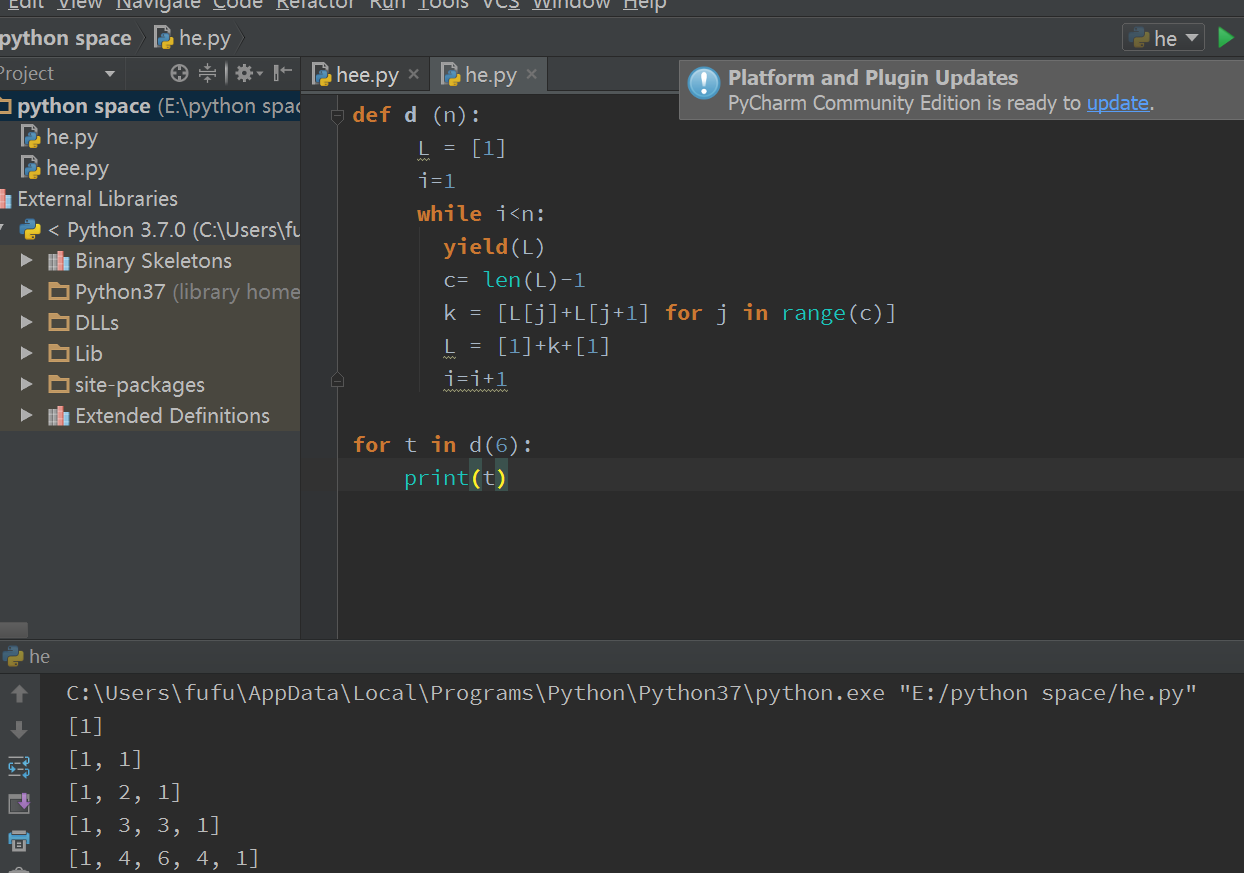
L = [1]+k+[1]

i=i+1

for t in d(6):

print(t)

实际运行结果



Day3（8月23日）

一．Map Reduce练习

1. 利用map()函数，把用户输入的不规范的英文名字，变为首字母大写，其他小写的规范名字。输入：['adam', 'LISA', 'barT']，输出：['Adam', 'Lisa', 'Bart']：

源代码为：

def d(s):

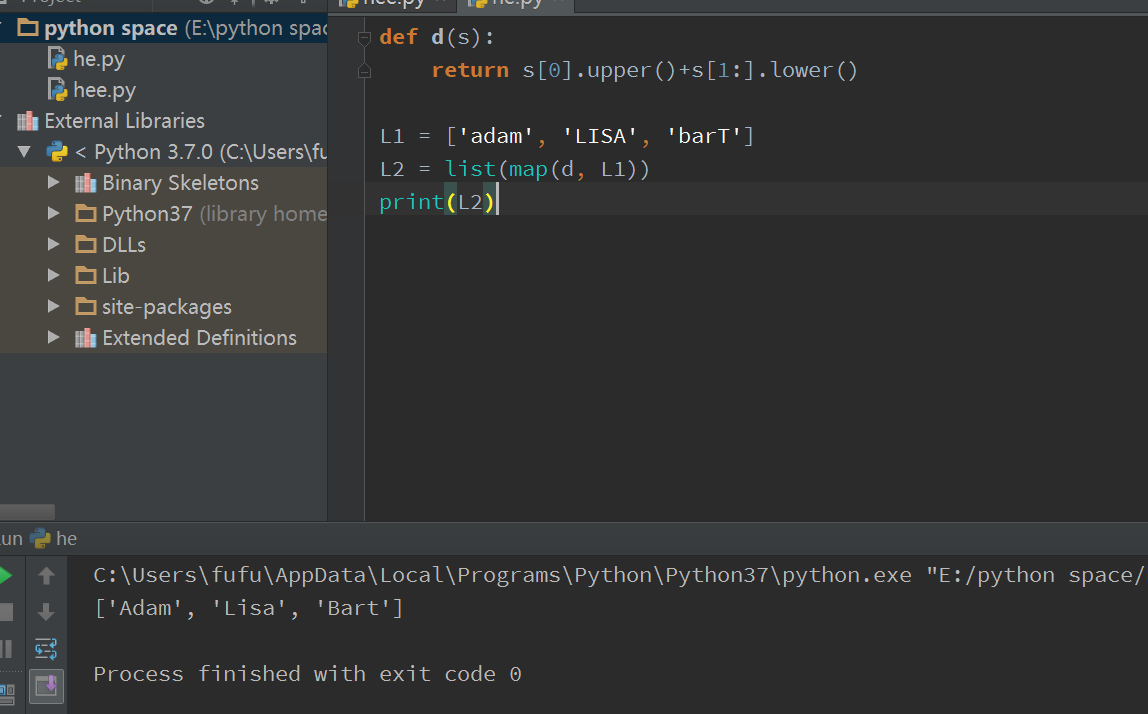
return s[0].upper()+s[1:].lower()

L1 = ['adam', 'LISA', 'barT']

L2 = list(map(d, L1))

print(L2)

实际运行结果为：



2. Python提供的sum()函数可以接受一个list并求和，请编写一个prod()函数，可以接受一个list并利用reduce()求积

源代码为：

from functools import reduce

def d(x,y):

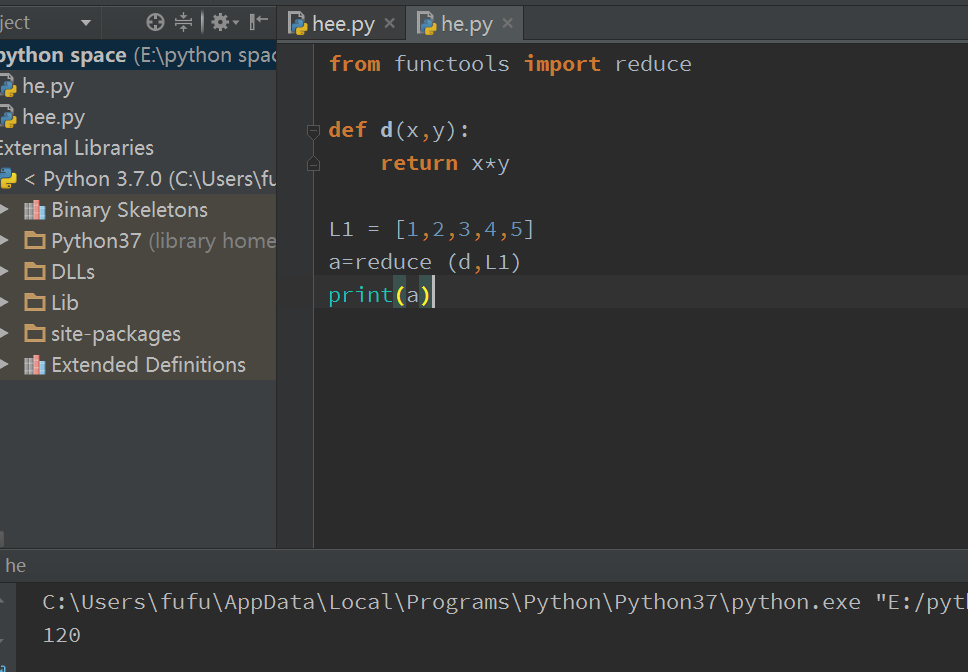
return x\*y

L1 = [1,2,3,4,5]

a=reduce (d,L1)

print(a)

实际运行结果为



二．Filter练习

回数是指从左向右读和从右向左读都是一样的数，例如12321，909。请利用filter()筛选出回数：

源代码为：

def c(n):

s=str(n)

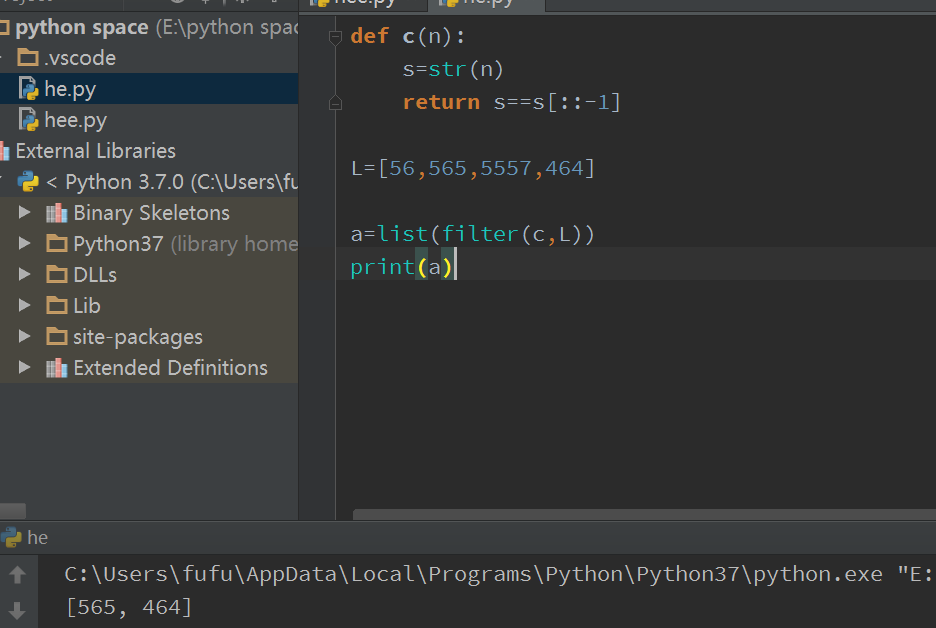
return s==s[::-1]

L=[56,565,5557,464]

a=list(filter(c,L))

print(a)

实际运行结果为



三．Sorted练习

假设我们用一组tuple表示学生名字和成绩：

L = [('Bob', 75), ('Adam', 92), ('Bart', 66), ('Lisa', 88)]

请用sorted()对上述列表分别按名字排序：再按成绩从高到低排序：

源代码为：

L = [('Bob', 75), ('Adam', 92), ('Bart', 66), ('Lisa', 88)]

print(L)

def byname(T):

return T[0]

def byscore(t):

return t[1]

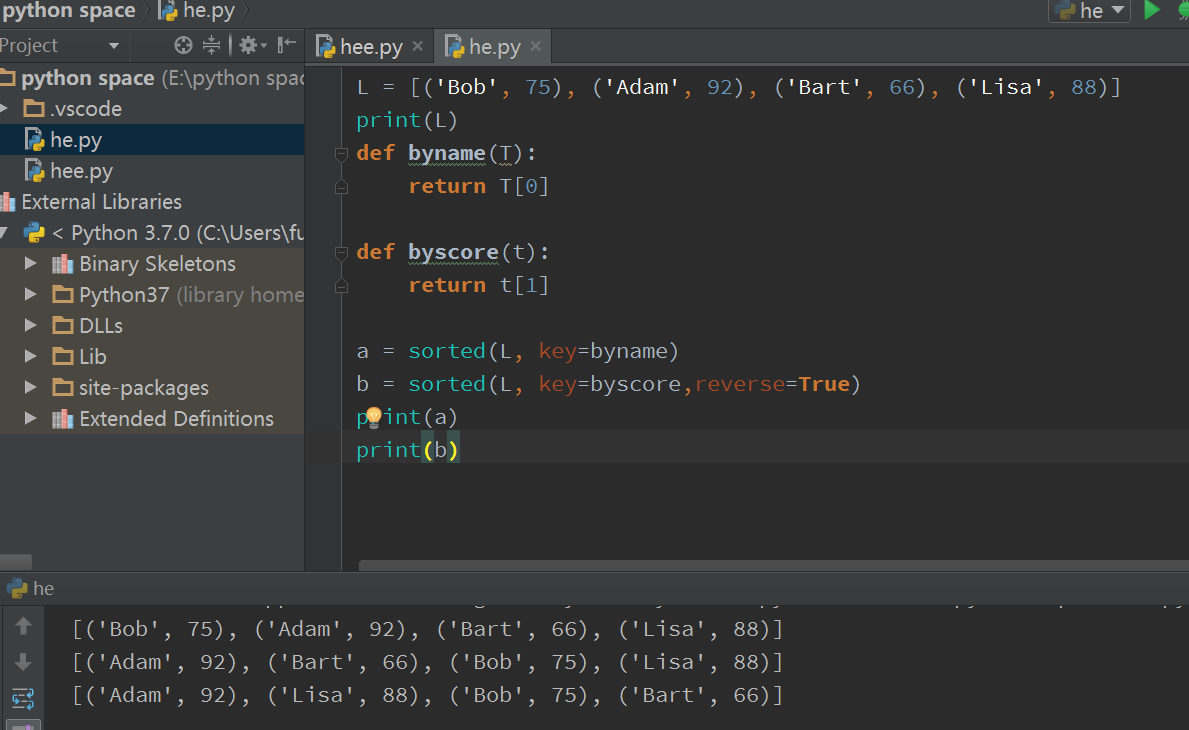
a = sorted(L, key=byname)

b = sorted(L, key=byscore,reverse=True)

print(a)

print(b)

实际运行结果为：



四．装饰器练习

请设计一个decorator，它可作用于任何函数上，并打印该函数的执行时间：

源代码：

import time, functools

def log(func):

@functools.wraps(func)

def wrapper(\*args, \*\*kw):

t1 = time.time()

result = func(\*args, \*\*kw)

t2 = time.time()

print('%s executed in %s ms' % (func.\_\_name\_\_, t2 - t1))

return result

return wrapper

@log

def s():

s=1

for i in list(range(50)):

s=s+i

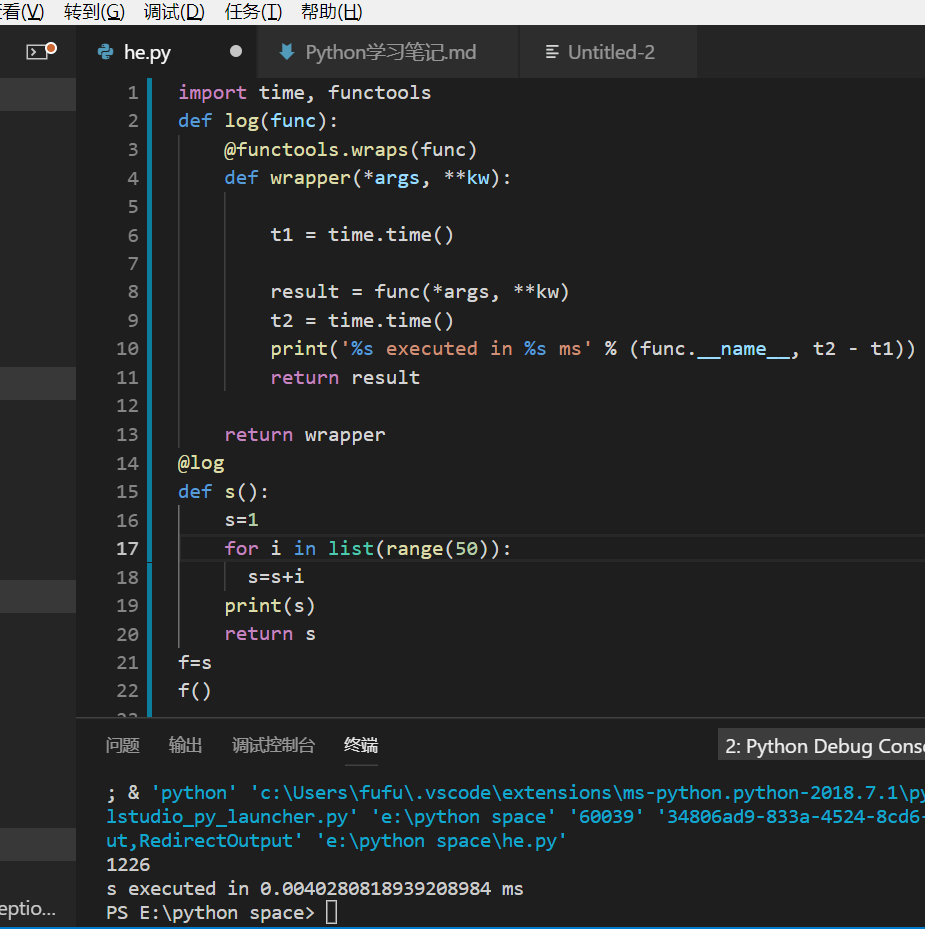
print(s)

return s

f=s

f()

实际运行结果为



五．类的访问限制练习

请把下面的Student对象的gender字段对外隐藏起来，用get\_gender()和set\_gender()代替，并检查参数有效性：

源代码为：

class Student(object):

def \_\_init\_\_(self, name, gender):

self.name = name

self.\_\_gender = gender

def get\_gender(self):

return self.\_\_gender

def set\_gender(self, gender):

if gender == 'female' or gender == 'male':

self.\_\_gender = gender

else:

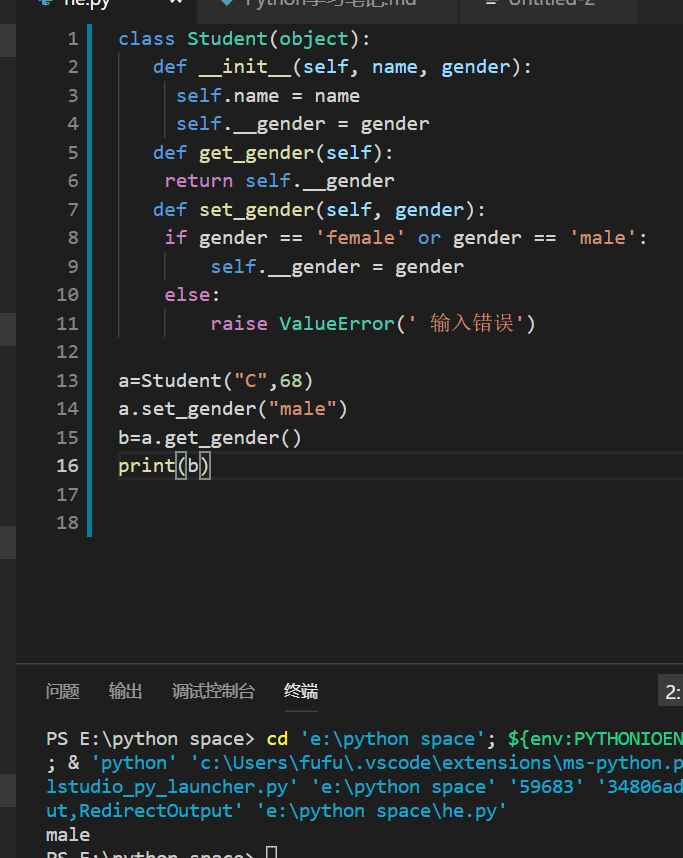
raise ValueError(' 输入错误')

a=Student("C",68)

a.set\_gender("male")

b=a.get\_gender()

print(b)



六．类的访问限制练习

为了统计学生人数，可以给Student类增加一个类属性，每创建一个实例，该属性自动增加：

源代码为：

class Student(object):

count=0

def \_\_init\_\_(self,name):

self.name=name

print(self.name)

Student.count=Student.count+1

a=Student("a")

b=Student("b")

print(Student.count)

