1. vsc插件下载地址： <https://marketplace.visualstudio.com/vscode>
2. 换成中文

先ctrl+shift+p，输入configure Language 按回车，下载中文包

1. 注释

单行注释# Ctrl+/

多行注释’’’ ‘’’ Shift+Ctrl+a

1. python中常量建议都大写
2. 保证TAB是四个空格（设置里弄）
3. Elif用法

print("请输入你想的数字：")

a=input()

if a==1:

print("apple")

elif a==2:

print("orange")

elif a==3:

print("banana")

else:

print("go shopping")

 用elif代替else if这样的多重嵌套，代码更少，python中是没有switch的

输入1，结果是go shopping 是因为我们以为自己输入的是数字，实际系统任务是字符，需要加一行a=int(a)，将a转换为int类型

1. while用法（递归方便）
2. for用法（遍历/循环 序列或者集合、字典）

a=[['apple','banana','orangle'],(1,2,3)]

for x in a:

for y in x:

print(y,end='') #end=''可以使不换行，里面加\n为换行

else:

print("fruit is gone") #if中很少用else

for循环到2终止

b=[1,2,3]

for x in b:

if x==3:

break

print(x)

**For循环跳过2**

b=[1,2,3]

for x in b:

if x==2:

continue

print(x)

**xy循环break的误区**

a=[['apple','banana','orangle','grape'],(1,2,3)]

for x in a:

for y in x:

if y=='orangle':

break

print(y) #end=''可以使不换行，里面加\n为换行

else:

print("fruit is gone")

结果是：apple banana 1 2 3 fruit is gone

原因：这个break是跳出内部的循环，但外部的循环还在继续

**Python中如何模拟for (i=o;i<10;i++){ }的情况**

for c in range(0,10): #从0开始,10个数

print(c)

接上一步，如何间隔n个数

for c in range(0,10,2): #从0开始,10个数,间隔两个数

print(c)

间隔2个数倒序输出

for c in range(10,0,-2):

print(c,end=' | ')

**作业**

**a=[1,2,3,4,5,6,7,8]打印出1,3,5,7或者2,4,6,8**

#d=[1,2,3,4,5,6,7,8,9]

print("请输入数列：")

d=input()

d=d.split(',') #split分割输入的数据

x=len(d)

a=int(d[0])

print("输出结果：")

for b in range(a,x,2):

print(b,end='|')

最佳答案：

a=[1,2,3,4,5,6,7,8,9]

b=a[0:len(a):2]

print(b,end='|')

1. 包--->模块------>类----->函数、变量

文件夹代表包，文件代表模块

\_\_init\_\_.py 任务是一个包，否则认为是普通模块。Eg:seven文件夹下有\_\_init\_\_，不是写seven.\_\_init\_\_ ，而是直接写seven

引入import(1)

import test.t1 as m

print(m.a)

引入import(2)

from t.t3 import b

print(b)

换行，某位加反斜杠，一行默认是80个字符

from t.t3 import b,c,\

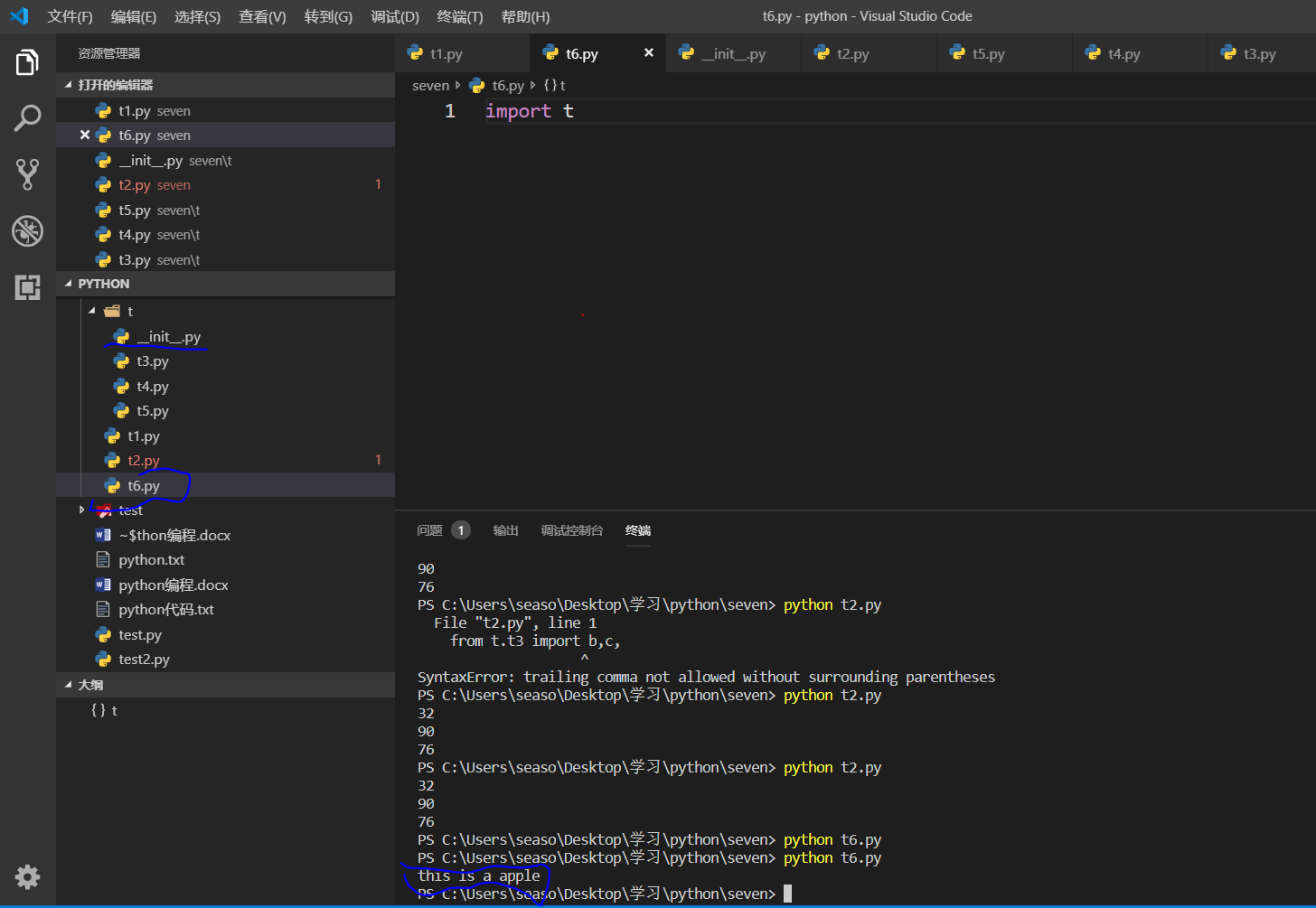
d

或者用（）

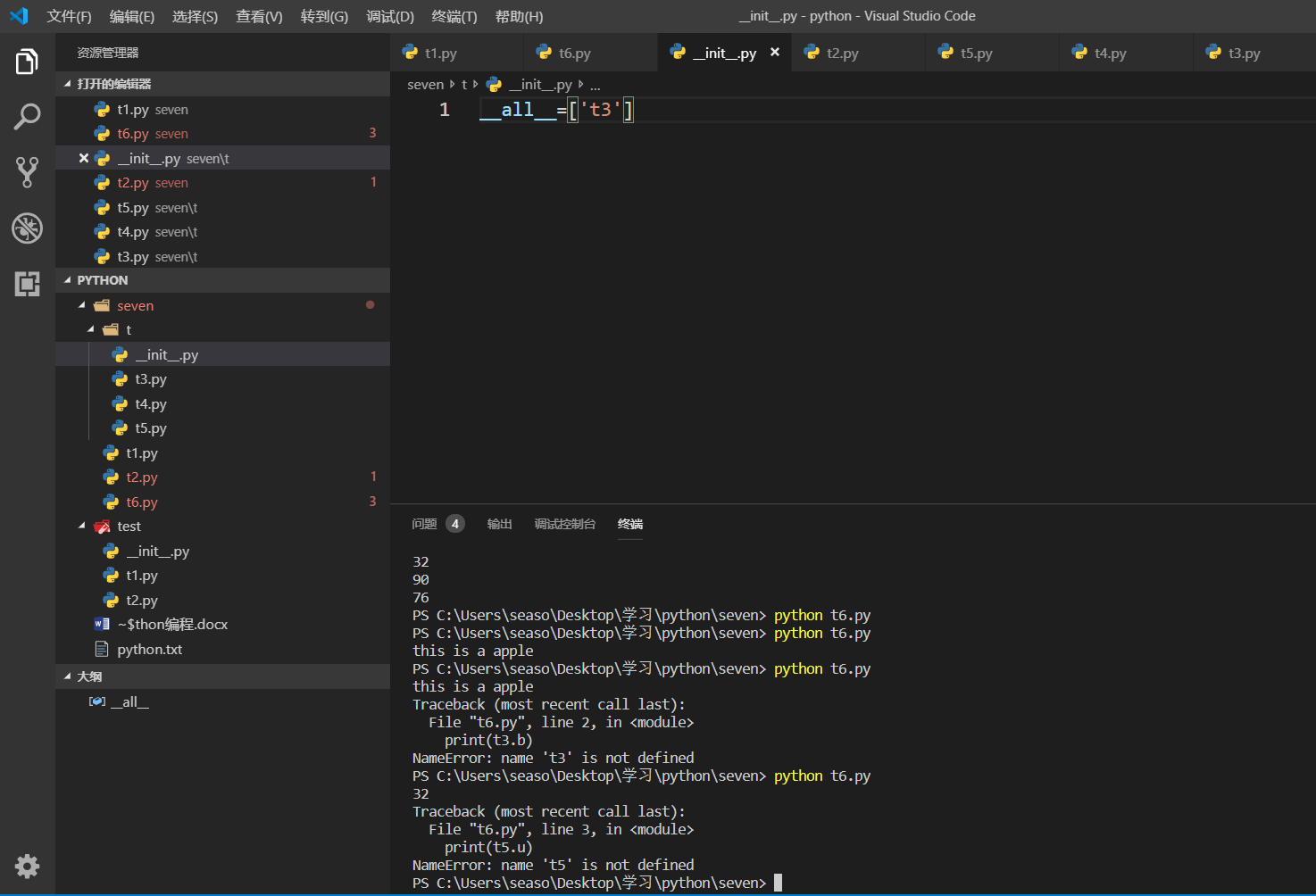
from t.t3 import (b,c,

d)

\_\_init\_\_.py的用法



不管引用t还是t下面的文件，都是输出\_\_init\_\_下输出的文件



只引用了t3里的，所以正常输出，t5的报错了

包和模块不是会被重复导入的?

eg:\_\_init\_\_.py里用东西引用，t12和t13不能都引用，会有性能问题，所以避免这个情况

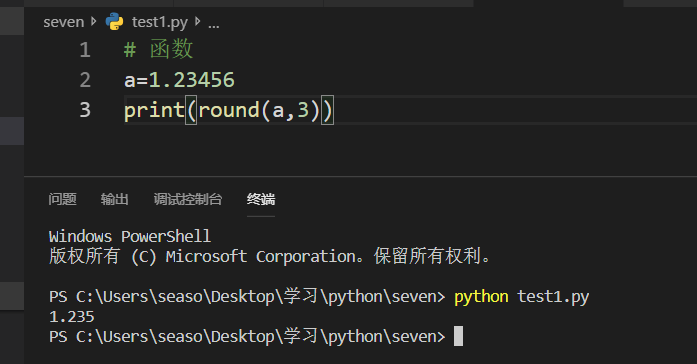
避免循环导入

Eg：p1.py引用p2.py的东西，p2.py引用p1.py的东西

P1-->p2-->p3-->p4-->p1

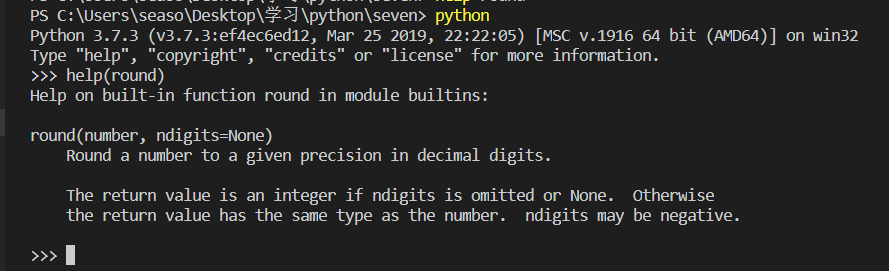
这样的循环引用时不允许的

1. 函数
2. round函数



保留输入数的几位小数点

1. 查看函数的用法



1. 自定义函数结构

def funcname(parameter\_list):

pass

1、参数列表可以没有

2、return value None

（4）设置递归层数

#设置系统的最大递归数，但超过系统本身的这个是无效的

import sys

sys.setrecursionlimit(800)

1. 没有return会返回none

def add(a,b):

result=a+b

return result

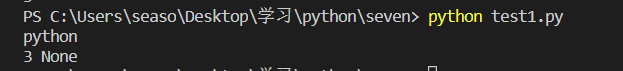
def print\_code(code):

print(code)

a=add(1,2)

b=print\_code('python')

print(a,b)



1. 函数中return为终止，return之后不会执行
2. 序列解包

def damage(skill1,skill2):

damage1=skill1\*3

damage2=skill2\*2+10

return(damage1,damage2)

skill1\_damage,skill2\_damage=damage(3,4) #序列解包

print(skill1\_damage,skill2\_damage)

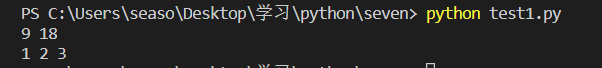
#a,b,c=1,2,3 可以这样写a=1;b=2;c=3

#序列解包

d=1,2,3

a,b,c=d

print(a,b,c)



接收数要相等

1. 必须参数和关键参数（可以自己设置值的位置，不一定按顺序）。

def divison(x,y): #必须参数

result=x/y

return result

c=divison(y=3,x=2) #这种就是关键参数的表示，不一定按顺序

print(c)



（9）默认参数

def print\_student\_files(name,gender='男',age=18,college='上海'):

print('我叫'+name)

print('我今年'+str(age)+'岁')

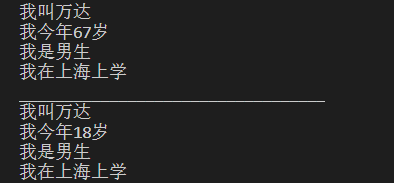
print('我是'+gender+'生')

print('我在'+college+'上学')

print\_student\_files('万达','男',67,'上海')

print('\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_')

print\_student\_files('万达')



默认参数和必须参数不能混在一起

def print\_student\_files(name,gender='男',age=18,college='上海'):

print('我叫'+name)

print('我今年'+str(age)+'岁')

print('我是'+gender+'生')

print('我在'+college+'上学')

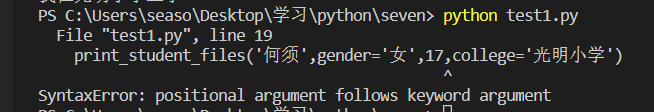
print\_student\_files('万达','男',67,'上海')

print('\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_')

print\_student\_files('万达')

print('---------------------------------')

print\_student\_files('何须',gender='女',17,college='光明小学') #默认参数和必须参数不能混在一起



1. 类（面向对象）

# 类=面向对象。作用：封装

class Studnet():

name='李四'

age=0

def print\_file(self):

print('姓名：'+ self.name)

print('年龄:'+ str(self.age))

# return self.name,self(str(age))

student=Studnet()

student.print\_file()

加self，原因自己查一下

1. \_\_init\_\_可以被调用

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

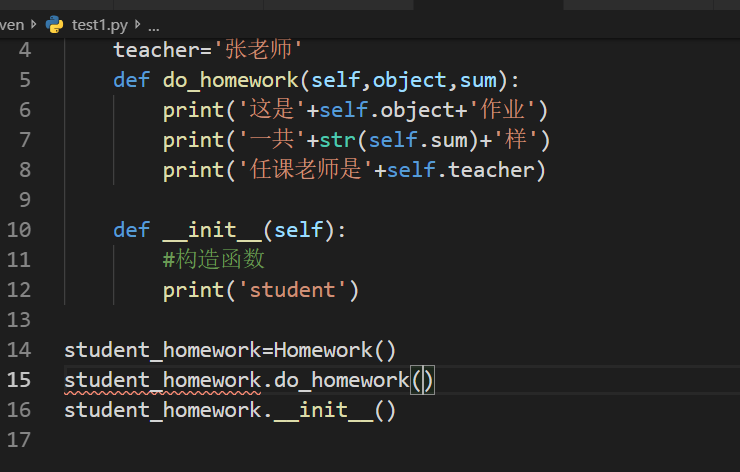
#构造函数

print('student')

student\_homework=Homework()

student\_homework.\_\_init\_\_()

1. 定义是object写了，下面输出也要



1. 类变量和实例变量（实例和对象相关联）
2. 排序

a=[1,6,8,11,9,1,8,6,8,7,8]  
*#打印次数最多的数*print(max(set(a), key=a.count))  
*#a从小到大排序*print(sorted(a))  
*#从大到小排序*print(sorted(a,reverse=**True**))  
*#去重从小到大排序*print(set(a))