语法错误 invalid character 中文字符

逻辑错误

函数和库

内置函数（标准函数） 直接使用

库函数（使用函数前导入库）[标准库函数（math,random,time ），第三方库函数(需安装)]

import turtle

turtle.fd(100)

from turtle import \*(后面不写库名)/

import turtle as t(后面引函数：t.)

自定义函数

def fff(a,b):

s=a=b

return s

fff(2,3)

缩进

if ：自动缩进四格

标识符：名字（系统标识符（关键字，关键词）（系统预设的）

用户自定义标识符（字母，数字，下划线）（不以数字开头））

所有的数据类型

（1）简单类型

数值型

1 int整数型

1.1float浮点型（指数形式表示：1e3即0.001 右边只能为整型）

complex 复数型 1+2j

字符串 str(string)

s=''/s=""/s=''' '''

替换 s.replace("大""小")

（2）组合类型

元组 tuple t=(1,2) 无法修改

字典 dict d={"江苏"：“南京”}

集合 set s1={1,2,3}

列表list 1s=["小"，90，10]（后三个可修改）

a=input("请输入：")

请输入：99

a

'99' #input()先输出（）内提示信息，以字符串形式

r=eval（input（））——去引号、适用于纯数、int()函数处理数值型

#即input（）存储的是字符串，通过eval（）转化成数值型

#计算圆面积

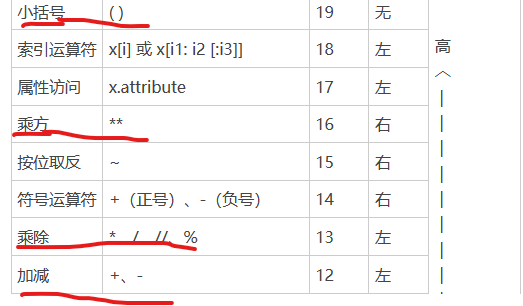
r=eval(input("请输入圆的半径:"))

s=3.14\*r\*r

print ("半径为:",r,"的圆的面积为：",s,sep="")

运算符：算数运算符（+-\*/ +=：复合 //整除 %取余 \*\*乘方）、关系、逻辑

优先级



=：赋值、引用（左边不能x+1类似，只能一个合法标识符）

同步赋值（解包赋值） x,y=y,x

连续赋值 x=y=z=1，右结合

复合赋值 +=

%:a,b异号

-4%3

-4/3=-2···2 -2为小于商且离商最近的整数 2为取余运算结果

divmod(5,3) （整除，取余）（以元组返回）

内置函数

Round(3.211,2)后一位为小数点后位数

Abs（）绝对值

Pow（2，3） 2的3次方

Max（）

Min（）

Sum([1,2,3])

布尔型（bool）true 1,false 0

And 与，or 或，not 非（0 假，非0 真）(优先级not>and>or 算数>关系>逻辑)

And 真取右 假取假值

Or 两真取左，一真取真(一假必为右)

十进制 十六进制 八进制 二进制

关系运算

If 5==3

1<2<3(左结合)

5>4>3 (等价5>4 and 4>3)

X=y=1(右结合)

#判断闰年（能被4整除且不被100整除 或 能被400整除）

y=eval(input(“请输入年份：”))

if y%4==0 and y%100!=0 or y%400==0:

print("是")

else:

print("否")

可加括号（优先）

Is/is not(左右是不是同一个)

类型转换（自动转换、强制转换）

强制 int(“1”)

Print(“{}{}{}”.format(a,b,c))

# {}为槽 {0}则输出第一个数 写序号要写就全写

{1:} :——格式引导符 :右边写数据类型 f 6位 （d为整）

{:.2f}——小数点后两位

{:5.2f}—— 5为宽度（空格也占宽度 ）（只能更大）

{:2d}——宽度为2

默认右对齐

{:=^20.4f}——小数点后4位，宽度20，居中对齐，空格由=填充

\n 换行

分支结构

单分支结构 if 条件表达式：

语句

（可以不写else）

双分支 if ：

语句1

else：

语句2

多分支 if ：

语句1

elif :

语句2

……

else：

语句n

#判断奇偶数

a=eval(input("请输入"))

if a%2==0: 【if not a%2 也可以】

print (x,"是偶数")

else:

print(x,"是奇数")

#求三角形面积

import math

a=eval(input("请输入第一条边："))

b=eval(input("请输入第二条边："))

c=eval(input("请输入第三条边："))

if a+b>c and a+c>b and b+c>a:

p=(a+b+c)/2

s=math.sqrt(p\*(p-a)\*(p-b)\*(p-c))

print (s)

else:

print("不能构成三角形")

#多分支结构判断数的类型

x=eval(input("输入数值："))

if x>0:

print(x,"为正")

elif x==0:

print(x,"为0")

else:

print(x,"为负")

当前循环 while

#连续输入求最大、最小、平均，-1停

x=eval(input("请输入"))

zuida=x

zuixiao=x

c=0

s=0

while x!=-1:

if x>zuida:

zuida=x

if x<zuixiao:

zuixiao=x

c+=1

s+=x

x=eval(input("请输入"))

print("最大值"，zuida,"最小值"zuixiao,"平均值"，s/c)

#求和

s=0

i=1

while i<=100:

if i%2:

s+=i

i+=1

print(s)

#迭代求法

手动开平方根

y=(y+x/y)/2

x=eval(input("请输入一个数："))

y=1

while abs(y\*y-x)>=1e-6:

y=(y+x/y)/2

print(y)

#输入一串字符，统计数字字符、字母、空格和其他字符的数目

l=0

d=0

s=0

other=0

ss=input("请输入")

for I in range ss:

if c>='A' and c<='Z'or c>='a' and c<='z':

l=l+1

elif c>='0' and c<='9':

d=d+1

elif c==' ':

s=s+1

else:

other=other+1

c=input("请输入")

print("字母",l,"个","数字",d,"个","空格",s,"个","其他",other,"个")

randrange 不包含右边界

randint（）生成整数

#掷色子

import random

s=10

c=0

max=s

while s>0:

a=random.randint(1,6)

b=random.randint(1,6)

if a+b==7:

s=s+4

else:

s=s-1

if s>max:

max=s

c=c+1

print (s,end=" ")

print ("次数",c)

print("最大值",max)

while exp 未知次数

for 循环 已知次数

for +循环变量+in+迭代器(列表，元组，字符串可迭代——序列类型，数值不可迭代——非序列类型)：

语句

For i in range (10)：#未定义，默认从0开始，以n-1结束

#6次投100000次

import random

for x in range (5):

c=0 #放在循环之前

for i in range (100000):

a=random.randint(1,6)

b=random.randint(1,6)

if a+b==7:

c=c+1

print(c/100000,end=" ")

#输出十个星号

for i in range(10):

print("\*",end=" ") #使不换行

如果

for i in range(10):

print("\*") #默认输出一次换行一次

输出10行1颗星

#输出五行十个星号

for x in range(5):

for i in range(10):

print("\*",end=" ") #第一件事

print() #第二件事

穷举算法（水仙花）

for a in range (1,10):

for b in range (10):

for c in range (10):

s=100\*a+b\*10+c

if s==a\*\*3+b\*\*3+c\*\*3:

print(s)

正三角

For I in range(1,5)

空格

\*

换行

for i in range (1,5):

for x in range(4-i):

print(" ",end="")

for j in range(2\*i-1):

print("\*",end="")

print()

菱形

for i in range (1,5):

for x in range(4-i):

print(" ",end="")

for j in range(2\*i-1):

print("\*",end="")

print()

for i in range (1,4):

for x in range(i):

print(" ",end="")

for j in range(7-2\*i):

print("\*",end="")

print()

数字金字塔

for i in range (1,5):

for x in range(4-i):

print(" ",end="")

for j in range(2\*i-1):

print(i,end="")

print()

乘法口诀表

for i in range (1,10):

for j in range(1,i+1):

print("{}\*{}={}".format(i,j,i\*j),end=" ")

print()

for i in range (1,10):

for j in range(1,i+1):

print("{}\*{}={:<3d}".format(i,j,i\*j),end="")

print()

for i in range (1,11):

if i%3==0:

break

print(i,end=" ")

输出1，2

for i in range (1,11):

if i%3==0:

continue

print(i,end=" ")

输出1，2，4，5，7，8，10

#判断是否为素数

m=eval(input("请输入一个值："))

for i in range (2,m):

if m%i==0:

break

if i==m-1:

print("是")

else:

print("不是")

#判断是否为素数

m=eval(input("请输入一个值："))

for i in range (2,m):

if m%i==0:

break

else: #正常结束才能被执行

print("是")

#1000以内的完数按 6=1+2+3表达式形式输出

for i in range (1,1001):

s=0

for j in range (1,i):

if i%j==0:

s=s+j

if s==i:

print("{}=1 ".format(i),end="")

for j in range(2,i):

if i%j==0:

print("+{}".format(j),end="")

print()

#100-200以内的素数（5个一行）

for m in range(100,201):

c=0

for i in range(2,m):

if m%i==0:

break

if i==m-1:

print(m,end=" ")

c=c+1

if c%5==0:

print()（有问题）

#斐波那契

import math

n=eval(input("输入n"))

sum=1.0

if n<=2:

sum=0

s0=0

s1=0

s2=1

for k in range (4,n+1): #for k in range (4,n+1):

s=s0+s1+s2

print(s)

sum+=math.sqrt(s)

s0=s1

s1=s2

s2=s

print("该数列的前{}项的平方根之和为{:.6f}".format(n,sum))

或

n=eval(input("请输入n："))

x1=1

x2=1

print("{:>8}{:>8}".format(x1,x2),end="")

count=2

for i in range(3,n+1):

x3=x1+x2

print("{:>8}".format(x3),end="")

count+=1

if count%5==0:

print()

x1,x2=x2,x3

#最大公约数

a=eval(input("请输入a："))

b=eval(input("请输入b："))

c=eval(input("请输入c："))

for i in range(min(a,b,c),0,-1):

if a%i==0 and b%i==0 and c%i==0:

print(i,end=" ")

break #否则会一直往下取

字符串（序列类型，可迭代）

字符串的表示/访问

Len. #字符串长度

加法 连接

乘法 复制

字符串访问 （不可修改）

正向访问 0.1.2

反向访问 -1.-2.-3

索引s[0]

s[len(s)-2]

'学'

切片 s[start:stop:step]

左闭右开

s[-2:]

'学院'

回文【：：-1】

0.1+0.2==0.3

False

s="janfebmar"

a=eval(input("请输入月份："))

print(s[3\*a-3:3\*a])

字符串操作方法

字符串查找

S=”zzxxc”

s.capitalize(s)

字符串查找 find rfind index rindex count

s.find(a) 第一次出现

s.rfind(a) 从右边开始

s.index

s.find("学校")

-1

s.index("学校")

Traceback (most recent call last):

File "<pyshell#26>", line 1, in <module>

s.index("学校")

ValueError: substring not found

#查找水果

s="苹果:12个;梨:20个;桔子:35个;香蕉:18个"

name=input("请输入水果名称：")

if name in s:

pos=s.find(name)

print(name+"个数是"+s[pos+3:pos+6])

else:

print("没有")【查梨，三次机会】

替换0.l

S=”abc abc”

s.replace(“a”,”x”,1) 换一次

s="苹果：12个；梨：20个；桔子：35个；香蕉：18个"

f=s.replace("桔子","橙子")

f

'苹果：12个；梨：20个；橙子：35个；香蕉：18个'

date="2023-4-14"

date.split()

['2023-4-14'] #以列表形式返回

date.split("-")

['2023', '4', '14']

ss="ababcabcdbacde"

ss.partition("abc")

('ab', 'abc', 'abcdbacde')

ss.rpartition("abc")

('ababc', 'abc', 'dbacde')

ss.rpartition("abcx")

1s=[“11”,”22”,”33”]

“”.join(1s)

输出‘112233’

('', '', 'ababcabcdbacde')

合并

“：“.Join（1s）

a=input("请输入一组以英文逗号为分隔的数字：")

max=int(a[0])

c=""

b=a.split(",")

d=c.join((b))

for i in range(d):

if int(i)>max:

max=int(i)

print(max)

#首字母缩写

S=””

ls=s.split()

For x in range(ls):  
 print(x[0].upper)

a,b,c="xyz" 同步赋值

列表的表示 创建 运算 操作方法 内置函数 命令

创建

ls=[]

或者 list()

输入

ls=eval(input(“请输入：”)) #输入时加方括号

访问：索引、切片

ls=[0,0,0]

ls[0]

ls[1:3:1]

列表的访问 遍历

For x in ls:

或者 for i in range(len(ls)):

Print(ls[i])

import random

ls=["很满意","满意","一般","不满意"]

r=[]

for i in range (100):

r.append(random.choice(ls))

也可列表生成式

import random

ls=["很满意","满意","一般","不满意"]

r=[random.choice(ls) for i in range(100)]

print(r)

列表的操作方法

Ls.append([5]) #追加

Ls

[1,2,3,[5]]

s=["a","b"]

s.append([1,2])

s

['a', 'b', [1, 2]]

s.extend([5,6])

s

['a', 'b', [1, 2], 5, 6]

s.insert(10,8)

s

['a', 'b', [1, 2], 5, 6, 8]

Ls.insert(index#插入位置,6) #插入指定位置前

删除列表

操作方法

Ls.pop(index#默 ，认最后一位,)

Ls=remove() #移走最先出现的一个数

ls=clear()

命令

Del ls1

Del ls[0]

Ls.count(1) #统计次数

Ls.index() #找不到报错

Ls.reverse() #反序

Ls.sort() #升序排列

Ls.sort(reverse=True) #降序

排序的内置函数(本身不变，需以另一个命名) len() sorted()

Ls.copy() (若直接ls1=ls2，改一个两个都要改)——浅拷贝

Ls2=sl1.copy() ——深拷贝

#求字符串中最长的单词

ls="In the face of ambiguity refuse the temptation to guess"

r=[]

for i in ls.split():

r.append(len(i))

print(ls.split()[r.index(max(r))])

元组(tuple) 不可修改 操作方法（count index） {元组里面的列表元素可以改（间接修改）}

T=（1，2，3）

无关闭分隔符

X=1，2，3

X

vf（1，2，3）

#组合

snacks=["松饼","提拉米苏","芝士蛋糕","三明治"]

d=["红茶","咖啡","橙汁"]

m=[]

for x in snacks:

for y in d:

m.append((x,y))

for x in m:

print(x)

列表生成式的一般表示形式

[循环相关的表达式 for 外循环变量 in 迭代器 for 内循环变量 in 迭代器+筛选条件]（从左到右）

import random

ls=["不满意","一般","满意","很满意"]

a=[]

for i in range (100):

a.append(random.choice(ls))

print(a)

b=[]

for x in ls:

b.append(a.count(x))

print(b)

print("{}是评价最多的一项".format(ls[b.index(max(b))]))

逻辑判断

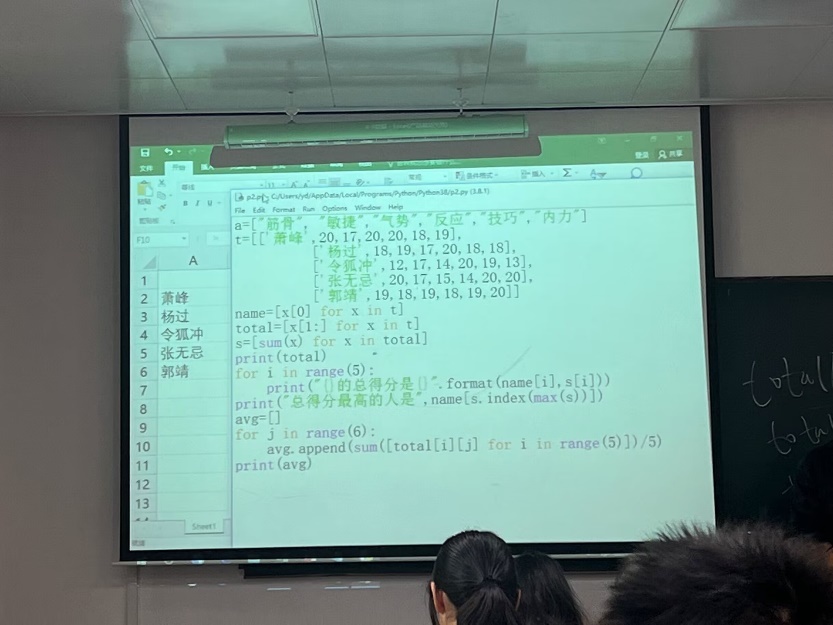
a=["甲","乙","丙","丁"]

for x in a:

if (x!="甲")+(x=="丙")+(x=="丁")+(x!="丁")==3:

print("做好事的人是：",x,sep="")

break



查找算法 顺序查找 折半查找

顺序查找

ls=[1,2,3,4,5,6]

x=eval(input("输入"))

for i in range(len(ls)):

if ls[i]==x:

print(ls[i])

break

折半查找

先排序 求中间值 比较大小

Low=[0] high=[len(ls)-1]

Mid=(low+high)//2

若x<mid high=mid-1

当low<high 继续执行

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

x=eval(input("输入"))

low=0

high=len(ls)-1

while low<=high:

mid=(low+high)//2

if x==ls[mid]:

print(x)

break

elif x<ls[mid]:

high=mid-1

else:

low=mid+1

if low>high:#或者写else:

print("找不到")

点数概率

import random

ls=[0]\*13

for i in range(10000):

f1=random.randint(1,6)

f2=random.randint(1,6)

ls[f1+f2]+=1

for i in range(2,13):

rate=ls[i]/10000

print("{}点数出现次数{},出现概率为{:.2%}".format(i,ls[i],rate))

字典和集合

字典 {} d={} d=dict() 创建空字典

d={条目1，条目2，条目3}

d={key1(#不可修改：字符串、数值、元组):value1(#只能为一个值),key2:value2}

多个相同的键 取最后

字典访问 按键值对字典进行访问 #d[键名]

遍历只能访问键

字典的操作 添加/修改/删除条目

修改d[“xx”]=”xc”

添加 d[“xx”]=”xc”

删除 del d[“xx”] #按键删除

d.clear() / #清空

d.pop(“键值”) 返回值为键值对应值

d.pop(1,”该键值不存在”) #存在输出1对应值，不存在输出”该键值不存在”

d.popitem()

字典的运算 字典的查找 成员运算符（in not in） 、 get方法 （pwd.get(键值) 返回value pwd.get(键值，“该键不存在”)）

键值 in d

#模拟输入密码

pwd={"aa":111,"bb":222,"cc":333}

name=input("请输入用户名：")

if name in pwd:

mima=eval(input("请输入密码："))

if mima==pwd[name]:

print("登陆成功")

else:

print("密码不正确，请重新输入")

else:

print("该用户不存在，请注册")

test={}

test.fromkeys((1,2,3),8)

{1: 8, 2: 8, 3: 8}

#计数

import random

ls=["综合","理工","师范","农林","民族","军事"]

ls1=[random.choice(ls) for i in range(39)]

d={}

for x in ls1:

d[x]=d.get(x,0)+1

for x in d:

print(x,d[x],sep=":")

#字典的整体操作：遍历

For k in pwd.keys():

Print(k)

For I in pwd.values():

Print(i)

For x in pwd.items():  
 print(x) #以元组形式返回

For k,v in pwd.items():  
 print(k,v)

按名字排序

d1={"cc":19,"aa":20,"bb":17}

ls=sorted(d1) #输出：['aa', 'bb', 'cc']

for x in ls:

print(x,d1[x])

按年龄排序

d1={"cc":19,"aa":20,"bb":17}

d2=[(v,k) for k,v in d1.items()]

d2.sort()

for x in d2:

print(x[1],x[0],sep=":")

统计性别与年龄：  
d={"李明":("男",19),"杨柳":("女",18)}

c={}

name=[]

for k,v in d.items():

c[v[0]]=c.get(v[0],0)+1

if v[1]>18:

name.append(k)

print(c)

print(name)

字典的合并

1 先转换成列表

d1={1:2,3:4}

d2={5:6}

ls1=list(d1.items())

ls2=list(d2.items())

dict(ls1+ls2)

2 遍历

for k,v in d2.items:

d1[k]=v

3.

d1.update(d2)

4.

dict(d1,\*\*d2) 键必须为字符串

#统计各科平均值

ls={………}

t={}

for d in ls:

for k,v in d.items():

if k!="性别":

t[k]=t.get(k,0)+v

print(t)

for k,v in t.items():

print("{}平均为{}".format(k,v/len(ls)))

集合{1，2，3，4} s=set() 创建空集合

集合内不能有重复元素 可以直接去除重复

集合不能访问（先转换成list），但可迭代

For x in s:

集合的操作方法

s.add() 只能一个对象 两个打包成一个元组（不可修改） （列表、字典可修改）

加重复不报错

S.remove(5) 删除不存在报错

S.discard(5) 删除不存在不报错

集合的数学运算 交 并 差 对称差

交 s1 & s2 或者 s1.intersection(s2)

并 s1 | s2 或者 s1.union(s2)

差 s1-s2 s1.difference(s2)

对称差 s1 ^ s2 s1.symmetric\_differences(s2)

优先性 交>并

可通过ls2=sorted(s) 对s排序 输出列表

函数

函数类型 内置函数 库函数 （内置函数 标准库函数）

Sum(range(1,11))

自定义函数 函数首部

函数名

def 函数名（【形参列表】）：

函数体

def f(n1,n2):

s=0

for i in range(n1,n2+1):

s=s+i

return s

def p1():

print("\*\*\*\*")

def p2():

print("python")

p1()

p2()

p1() （#函数调用语句）

def he(x,y):

s=x+y

return s

print(he(3,5))

def p(n):

for i in range(2,n):

if n%i==0:

return False

else:

return True

for i in range(2,101):

if p(i)==True:

print("{:^4}".format(i),end="")

for i in range(2,101):

if p(i)==True and p(i+2)==True:

print("{},{}".format(i,i+2))

for i in range(4,21,2):

for j in range(2,i):

if p(j)==True and p(i-j)==True:

print("{}{}{}".format(i,j,i-j))

break

互换

Def c(x,y):  
 x,y=y,x

Return x /return x,y #以元组形式打包返回

函数的参数

形参、实参（向形参值传递）

#按位置传递（从左向右） F(1,2)

按名称传递 f(a=1,b=2)

默认值参数

Def f(a=b=1): def f(b,a=1) (默认放表尾)

可变长参数 def f(a,\*b) f(1,2,3) 1传递给a 再2，3以元组打包传递给b

（a,\*\*b）

f(1,x=1)

1

{‘x’:1}

Lambda 表达式

ff=Lambda x,y:x+y

ff(1,2)

d={"cc":20,"aa":17,"bb":21}

sorted(d.items(),key=lambda x:x[1],reverse=True)

for i in filter(lambda x:x>0,[-1,-2,1,2,3,4]): #先写条件 再写序列

print(i,end=" ")

对字典sorted

d={"cc":20,"aa":17,"bb":21}

sorted(d.items())

[('aa', 17), ('bb', 21), ('cc', 20)]

d={"cc":20,"aa":17,"bb":21}

sorted(d)

['aa', 'bb', 'cc']

ls1=[1,2,3]

ls2=[4,5,6]

list(map(lambda x,y:x+y,ls1,ls2))

[5, 7, 9]

list(zip(ls1,ls2))

[(1, 4), (2, 5), (3, 6)]

#变量的作用域

x=99

def f():

global x #用的是全局变量

x=98

f()

print(x)

输出 98

x=99

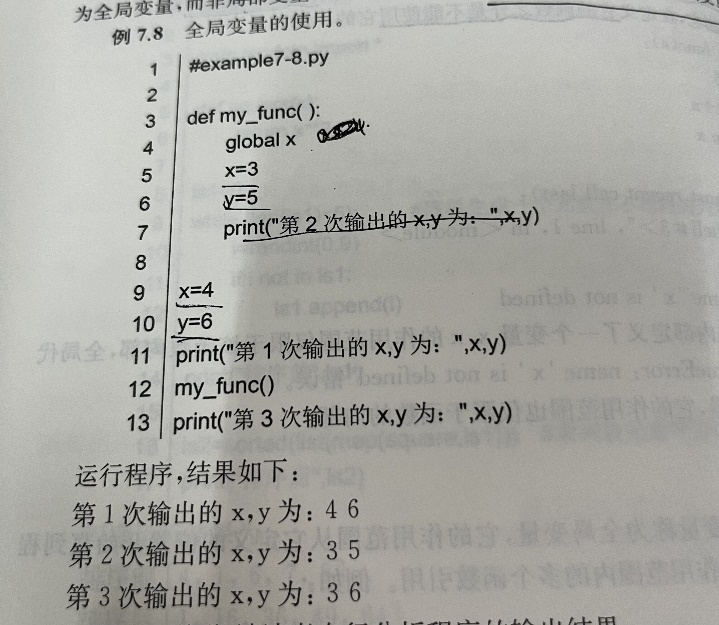
def f():

x=98 #创建新的局部变量

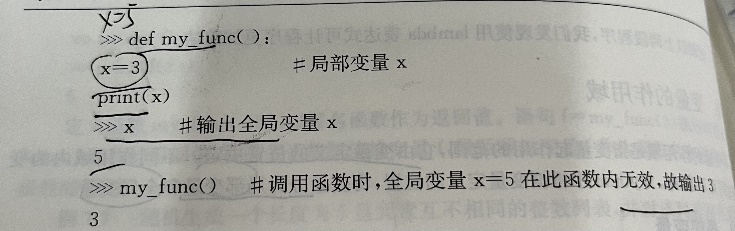
f()

print(x) #输出的始终全局变量

输出 99



重名时局部范围（在调用函数时）全局变量不起作用



#函数的递归调用

#5！

def p(n):

if n==1:

r=1

else:

r=p(n-1)\*n

return r

print(p(5))

120

异常处理机制

Try子句进行异常的捕获和处理

Try:

语句

Except [异常名称]：

捕获异常时的处理

Else：

未发生异常时的处理

try:

a=int(input("请输入a："))

b=int(input("请输入b："))

c=a/b

except ZeroDivisionError:

print("除数不能为0")

else:

print("c=",c,sep="")

也可用 if/else 语句

文件

fp=open("f1.txt","w") #打开文件的位置 打开文件的方式“w”（写）“r”（读）“a”（追加）

fp.write("yzdx") #只能写一个str #文件不存在时 写可以新建（有则删除原文件） 读则报错

fp.writelines([1,2]) #写入列表对象

4 #返回长度

fp.close()

import os

os.getcwd()

'C:\\Users\\辛耶·诺赞[\\AppData\\Local\\Programs\\Python\\Python310](file:///\\\\AppData\\\\Local\\\\Programs\\\\Python\\\\Python310)' (二个\改一 才能复制到地址栏 )

Fp=open(r“C:\Users\辛耶·诺赞[\AppData\Local\Programs\Python\Python310](file:///\\\\AppData\\\\Local\\\\Programs\\\\Python\\\\Python310)0\f1.txt”,”w”，encoding=”utf-8”)

Os.chadir('C:\\Users\\辛耶·诺赞[\\AppData\\Local\\Programs\\Python\\Python310](file:///\\\\AppData\\\\Local\\\\Programs\\\\Python\\\\Python310)' )

try:

fp=open("f2.txt","r",encoding="utf-8")

except FileNotFoundError:

print("不存在")

fp.read() #默认全读 也可指定大小

fp.closed

输出为True 则为关闭

Fp.read()

“rb”:二进制来读

13

fp=open("新建 文本文档.txt","r")

fp.read() #读后指针移到最后

'aa\nbbb\nccc'

fp.seek(0) #指针移到最前

0

fp.read(2) #读两个字节

'aa'

fp.seek(0)

0

fp.readline()

'aa\n'

fp.readline()

'bbb\n'

fp.readline()

'ccc'

fp.seek(0)

0

fp.readlines()

['aa\n', 'bbb\n', 'ccc']

fp=open("f2077.txt","w")

fp.writelines(["1","2","3"])

fp.close()

fp=open("f2077.txt","a")

fp.write("nxy")

3

fp.close()

复制

fp=open("f2077.txt","r")

x=fp.read() #也可先readlines 再writelines

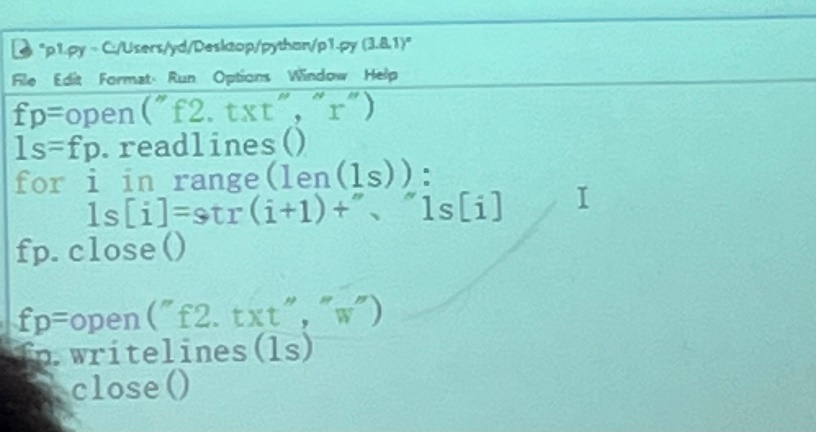
fp.close()

fp=open("f2088.txt","w")

fp.write(x)

6

fp.close()



with open("f2.txt","r") as fp1,open("lk.txt","w") as fp2:

ls=fp1.readlines()

fp2.writelines()

表格类型 .cxv 可以读取

创建常规读取对象

创建字典读取对象

fp=open("f5.csv","r",encoding="utf-8")

import csv

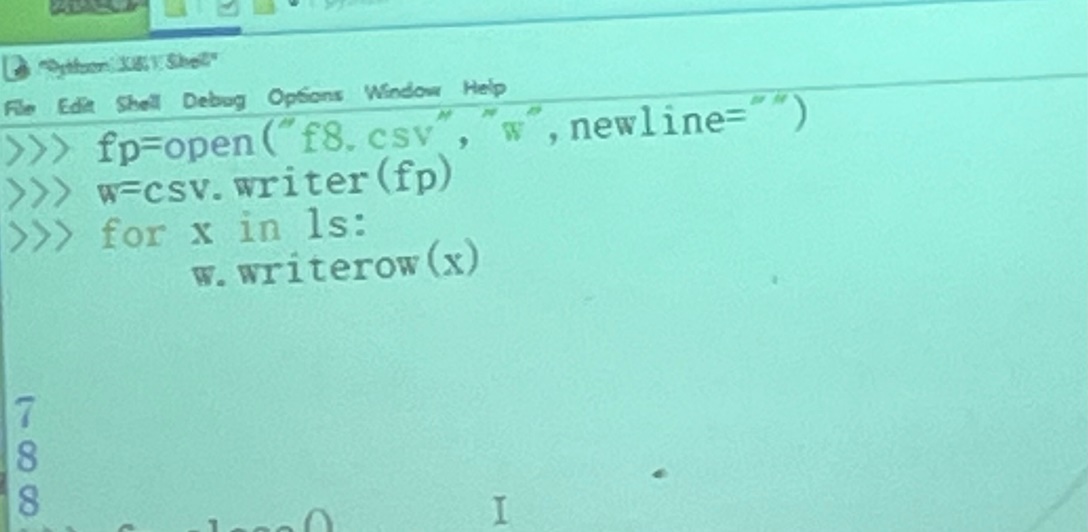
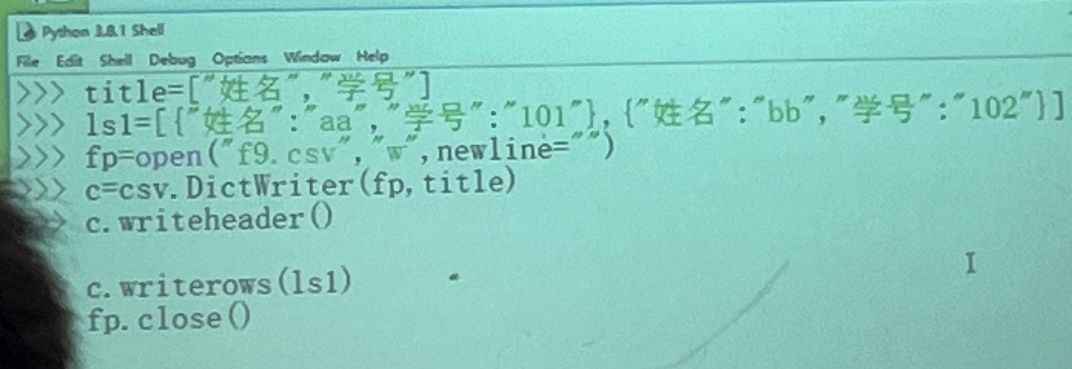
for x in d:

print(x)

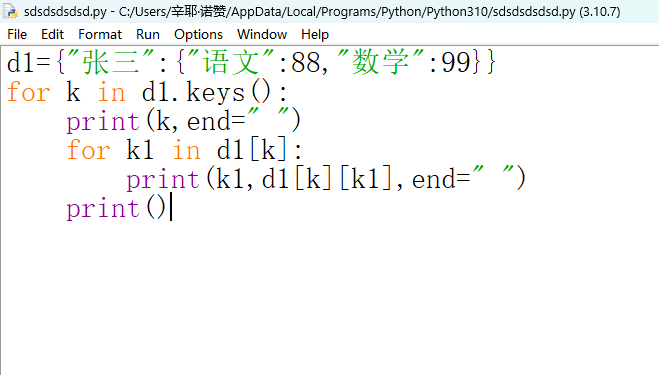
写入

#创建常规写入对象

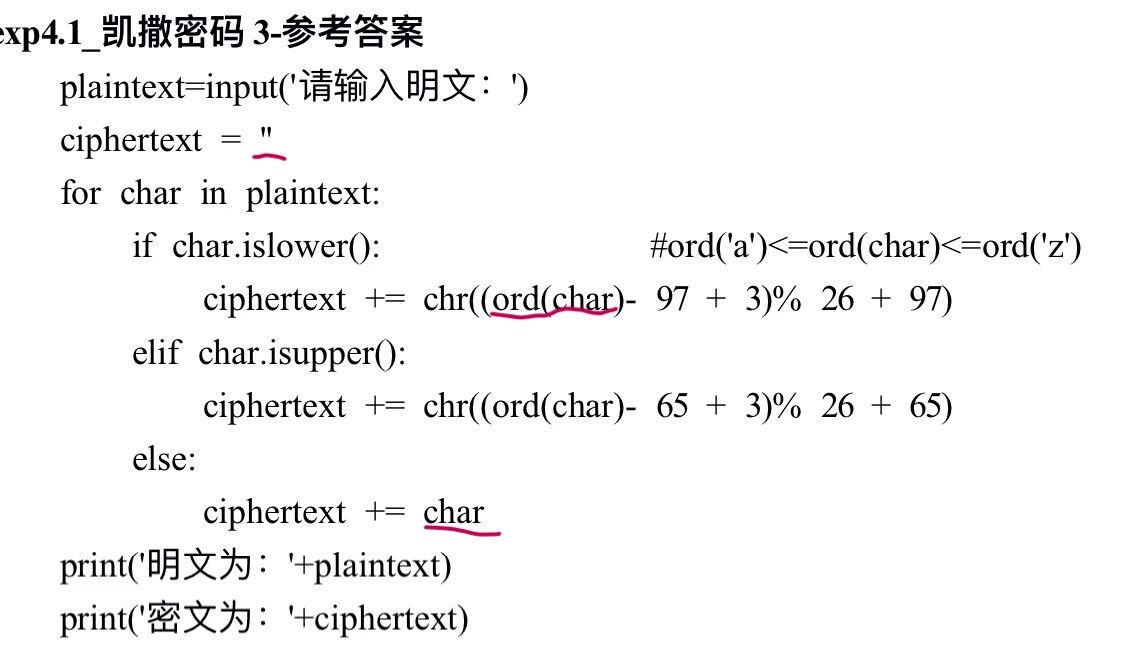
##创建字典写入对象

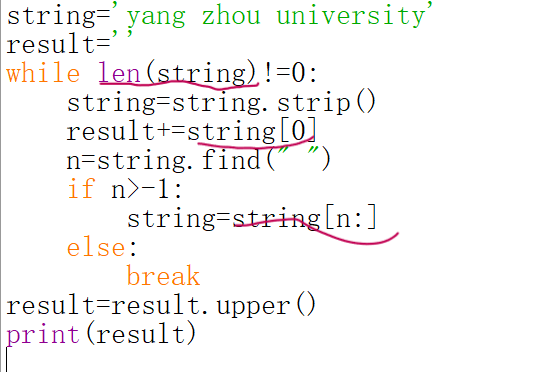
  


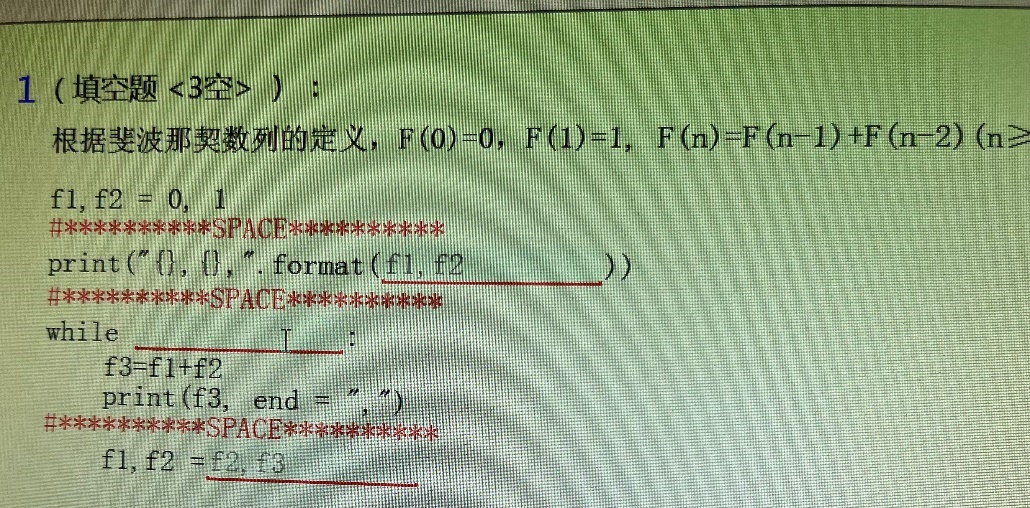
题目穷举



都写 keys()\items()\values()（除了这道的第二空）

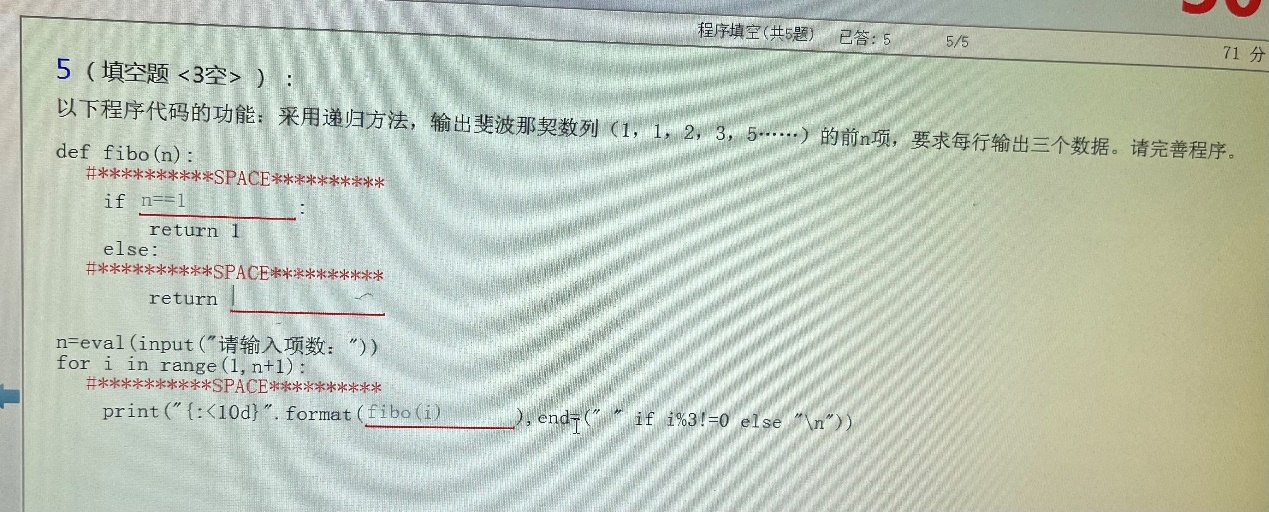






（题干：输出不大于100的序列元素）

F1<=100



Fibo(n-1)+fibo(n-2)

def func(n,jz):

if n<jz:

return (str(n) if n<10 else chr(ord("a")+n-10))

else:

t=n%jz

n=n//jz

return func(n,jz)+(str(t)) if t>10 else chr(ord("a")+t-10)

python turtle库位置控制函数是 goto()

else 可用于选择与循环等等

a=(1,2,3)

b=a[::-1]

b

(3, 2, 1)

将实参赋值（复制）给形参

"abce" in "abcdefg"

False

"abc" in "abcdefg"

True

最后一个字符的位置反向索引值 -1

lst=[1,2,1,7,5,1]

lst.remove(1)

print(lst)

[2, 1, 7, 5, 1] #移走最先出现的一个数

print(len({0}))

1

def fun(n):

return n,n+1,n+2,n

lst=fun(3)

print(lst)

(3, 4, 5, 3)

python的【os】模块提供了许多文件处理方法

def func(s):

if s=="":

return s

else:

return func(s[1:])+s[0]

print(func("good"))

输出 doog

Ls=[1,2,3]

ls[-1]=3

3 in [1,2,3]

True

[3] in [1,2,3]

False

Extend括号内部加外面

Append 括号内部加里面

x,y='AB'

x

'A'

y

'B'

try:

n=0

n=input("请输入：")

print(n\*n)

except:

print("程序执行错误")

输入5

输出：程序执行错误

aSet={1,2,3,4,5}

bSet={3,4,5,6,7}

cSet={3}

print(cSet < aSet & bSet)

输出第一行：True

def fun(n):

return {n,n+1,n+2,n}

lst=fun(3)

print(lst)

输出{3, 4, 5}

def func(s):

if s=="":

return s

if s[0].lower() in "aeiou":

return func(s[1:])

else:

return s[0]+func(s[1:])

print(func("Bautiful"))

输出：Btfl

print(tuple(range(3)),list(range(0)))

(0, 1, 2) []

print(len(dict([])))

0

即使函数不需要接收任何参数,也必须保留一对空的圆括号

[0]\*3

[0, 0, 0]

[[0]]\*3

[[0], [0], [0]]

s=[[0]]\*3

print(s)

s[1][0]=3

print(s)

[[0], [0], [0]]

[[3], [3], [3]]

i=0

while i<10:

i=i+1

if i<1:

continue

if i==5:

break

执行5次

def demo(\*a,c=4):

return sum(a)+c

print(demo(1,2,3))

输出10