# 在控制台录入一个四位的整数

# 计算每一位相加的和

# 显示结果

# 1234 --> 1+2+3+4 --> 10

num = int(input('请输入四位的整数：'))

result = num % 10

result += num // 10 % 10

result += num // 100 % 10

result += num // 1000

print('结果是：'+str(result))

#判断是否为闰年

year = int(input('请输入年份：'))

result = year % 4 == 0 and year % 100 != 0 or year % 400 == 0

print(result)

#身份运算符

a = 800

b = 1000

#id函数 获取变量储存对象的地址

print(id(a))

print(id(b))

# is运算本质是通过id函数进行判断的

print(a is b)

c = a

print(id(c))

print(c is a)

d = 1000

print(b is d)

e = 10

f = 10

print(e is f)

逻辑行

a = 1

b = a+1

c = a+2

物理行：a = 1;b = a+1;c = a+2#不建议

# if语句语法格式：

if a>b: #如果不满足向下执行

#满足条件a>b执行的代码

print(最大的数是a)#不报错原因在下方

elif a == b:#如果不满足向下执行

#满足条件a == b时执行的代码

print('a和b相等')

elif a<b:#如果不满足向下执行

#满足条件 a<b时执行的代码

print('最大的数是b')

#else中如果没有必须执行的代码 可省略

# else:

# pass#占位 填充语法空白

#选择语句执行的过程中 只会选一个符合条件的代码块执行，其余代码块都不会被解释器执行

#调试：

# 让程序中断 逐行执行

# 目的 审查程序运行时的变量以及变量取值

# 审查程序运行的流程

#步骤：

#1.加断点 (调试过程中遇到断点就停止程序)

#可能会出问题/你想详细了解的地方加断点

#2.运行调试Shift+F9

#3.程序会在断点处停止 按F8执行一行

#4.按Ctrl+F2停止调试

num1 = float(input('请输入第一个数:'))

op = input('请输入运算符:')

num2 = float(input('请输入第二个数:'))

if op == '+':

#执行+的操作

print(num1+num2)

else:

print('运算符有误')

# 打印最大的数字：

num1 = float(input('请输入第一个数字：'))

num2 = float(input('请输入第二个数字：'))

num3 = float(input('请输入第三个数字：'))

num4 = float(input('请输入第四个数字：'))

#假设第一个数是最大值

max\_value = num1

#将最大值与第二个数比较 如果第二个数比最大值大

if max\_value<num2:

#最大值等于第二个数

max\_value = num2

#依次与第三第四个数比较

if max\_value<num3:

max\_value = num3

if max\_value<num4:

max\_value = num4

#输出最大值

print(max\_value)

#在控制台输入一个月份 打印对应的天数

#1 3 5 7 8 10 12 --->31天

#4 6 9 11 ---> 30天

#2 ---> 28天

#其它 提示输入有误

month = int(input('请输入月份：'))

if month<1 or month>12:

print('输入有误')

elif month == 4 or month == 6 or month == 9 or month == 11:

print('30天')

elif month == 2:

print('28天')

else:

print('31天')

#在控制台获取一个整数

# 如果是偶数 为变量state赋值'偶数' 否则赋值 '奇数'

# num = int(input('请输入一个整数'))

# if num % 2 == 1:

# state = '奇数'

# else:

# state = '偶数'

# if num % 2:

# state = '奇数'

# else:

# state = '偶数'

#条件表达式

# state = '奇数' if num % 2 else '偶数'

# print(state)

#在控制台输入一个年份

#如果闰年 给变量day赋值29

#否则给day赋值28

year = int(input('请输入年份:'))

# if year % 4 == 0 and year % 100 != 0 or year % 400 == 0:

# day = 29

# else:

# day = 28

# if num % 2 ==1:

# if num % 2:

# if not year%4 and year%100 or not year%400:

# day = 29

# else:

# day = 28

day = 29 if year % 4 == 0 and year % 100 != 0 or year % 400 == 0 else 28

print(day)

#重复执行exerciese01.py代码 直到按e键退出

#死循环 循环条件永远满足

while True:

season = input('请输入季度：')

if season == '春':

print('1月2月3月')

elif season == '夏':

print('4月5月6月')

elif season == '秋':

print('7月8月9月')

elif season == '冬':

print('10月11月12月')

if input('输入e退出') == 'e':

break

#在控制台输出0 1 2 3 4 5

count = 0

while count<6:

print(count)

count += 1

#在控制台输出2 3 4 5 6 7

count = 2

while count < 8:

print(count)

count += 1

#在控制台输出0 2 4 6 8

count = 0

while count <= 8:

print(count)

count += 2

#在控制台输出 1~20之间的偶数

#循环找到1～20的数字

#判断 如果是偶数就输出

count = 1

while count<21:

if count % 2 == 0:

print(count)

count += 1

# 1.在控制台获取输入的月份 显示对应的季度 或提示月份错误

month = int(input('请输入月份：'))

if month<1 or month>12:

print('月份错误')

elif month<4:

print('春季')

elif month<7:

print('夏季')

elif month<10:

print('秋季')

else:

print('冬季')

# 2.在控制台获取年龄

# 如果小于0 打印输入错误

# 如果 小于2 打印是婴儿

# 如果 小于2~13 儿童

# 13~20 青年

# 20~65 成年人

# 65~130 老年人

# 超过130 不可能

age = int(input('请输入年龄：'))

if age<0:

print('输入错误')

elif age<2:

print('是婴儿')

elif age<13:

print('是儿童')

elif age<20:

print('青年')

elif age<65:

print('成年人')

elif age<=130:

print('老年人')

else:

print('妖怪')

# 3.根据身高和体重 参照BMI 返回身体状况

# BMI 体重(kg)/身高(m)\*\*2

# 中国参考标准

# BMI<18.5 体重过低

# 18.5<=BMI<24 正常

# 24<=BMI<28 超重

# 28<=BMI<30 I度肥胖

height = float(input('请输入身高(m):'))

weight = float(input('请输入体重(kg):'))

BMI = weight/height\*\*2

if BMI<18.5:

print('体重过低')

elif BMI<24:

print('正常')

elif BMI<28:

print('超重')

else:

print('肥胖')

#在控制台获取一个整数

#判断是否为素数

#素数：只能被1和自身整除的数

#1 (2 3 4 5 6 7 8) 9

#1 (2 3 4 5 6) 7

#判断11 从2~10之间的数字能不能整除

#if 11 % 2 == 0：

# print('11不是素数')

#if 11 % 3 == 0：

# print('11不是素数')

# if 11 % 4 == 0：

# print('11不是素数')

#if 11 % 5 == 0：

# print('11不是素数')

#...

number = int(input('请输入一个整数：'))

for i in range(2,number):#(2,11)

if number % i == 0:

print(str(number)+'不是素数')

#如果发现满足条件的数字 就不用再判断后面的数字

break

else:

print(str(number)+'是素数')

# print(number,'是素数')

#1.在控制台 获取一个字符串

#打印这个字符串的每一个字符的编码值

#字 --> 数 ord('a')

#数 --> 字 chr(97)

str1 = input('请输入文字:')

for ch in str1:

print(ord(ch))

#2.在在控制台 重复录入编码值 将编码值转为字符打印

#如果录入的是空字符串 则退出程序

while True:

str2 = input('请输入编码值')

if str2 == '':

break

code\_value = int(str2)

print(chr(code\_value))

# print(chr(int(str2)))

#for循环嵌套

#外层循环执行一次 内层循环执行一遍

for i in range(3):

for j in range(3):

print(i,j)

#累加1~100之间 能被5整除的数字

# sum\_number = 0

# for i in range(1,101):

# if i % 5 == 0:

# sum\_number += i

# print(sum\_number)

# sum\_number = 0

# for i in range(1, 101):

# if i % 5 != 0:

# #跳过本次循环 直接开始新的循环

# continue

# sum\_number += i

# print(sum\_number)

#打印1~20之间的数字 跳过所有偶数

for i in range(1,21):

if i % 2 == 0:

continue

print(i)

#猜数字

#导入随机模块

import random

#产生一个从1到100之间的随机数

random\_number = random.randint(1,100)

print(random\_number)

#用户输入一个数字 电脑随机生成一个数字

#判断用户输入的数字和电脑随机生成的数字是否相同

# 如果相同 提示猜对了

# 如果用户输入的大 提示猜大了

# 否则 提示猜小了

# input\_number = int(input('请输入1~100的数字:'))

# if input\_number == random\_number:

# print('猜对了')

# elif input\_number > random\_number:

# print('猜大了')

# else:

# print('猜小了')

#1.重复执行以上代码 直到猜对为止

#2.重复执行三次代码

# 如果用户猜对了 提示:“猜对了，总共猜了xx次”

# 如果用户三次没猜对 提示用户 “你输了正确的数字是xx”

# 1.重复执行以上代码 直到猜对为止

# while True:

# input\_number = int(input('请输入1~100的数字:'))

# if input\_number == random\_number:

# print('猜对了')

# break

# elif input\_number > random\_number:

# print('猜大了')

# else:

# print('猜小了')

#2.重复执行三次代码

# 如果用户猜对了 提示:“猜对了，总共猜了xx次”

# 如果用户三次没猜对 提示用户 “你输了正确的数字是xx”

#统计循环的次数

count = 0

while count<3:#0 1 2

count+=1

input\_number = int(input('请输入1~100的数字:'))

if input\_number == random\_number:

print('猜对了,总共猜了'+str(count)+'次')

break

elif input\_number > random\_number:

print('猜大了')

else:

print('猜小了')

else:#循环条件不满足时执行的代码

print('你输了,正确的数字是'+str(random\_number))

#打印数字0 1 2 3

#for+range(次数) 执行预定的次数

# for i in range(4):

# print(i)

#累加 0 1 2 3

# sum\_value = 0

# for i in range(4):

# sum\_value += i

# print(sum\_value)

#累加3 5 7 9

# sum\_value = 0

# for i in range(3,10,2):

# sum\_value += i

# print(sum\_value)

# #输出数字4 3 2 1 0

# for i in range(4,-1,-1):

# print(i)

#假设一张纸0.0001米

#折纸10次 求厚度

#珠穆朗玛 8848米 求一张纸对折多少次能达到8848米

thickness = 0.0001

# for i in range(10):

# thickness\*=2

# print(thickness)

count = 0#统计次数

while thickness<8848:

count += 1

thickness \*= 2

print(count)

#随机加法考试

import random

#随机生成两个数字

#在控制台获取用户输入的两个数字相加的结果

#3+2=？ 5

#8+5=？ 3 x

#如果用户输入正确得20分

# 共5道题

# 保存分数

score = 0

for i in range(5):

random\_number01 = random.randint(0, 100)

random\_number02 = random.randint(0, 100)

# 提示字符串 3+2=？

prompt = str(random\_number01) + '+' + str(random\_number02) + '=?'

# 用户输入前 显示提示字符串

input\_number = int(input(prompt))

if input\_number == (random\_number01 + random\_number02):

score += 20

#最后打印得分

print(score)

# 6.打印出1000以内所有的"水仙花数"，

# 所谓"水仙花数"是指一个三位数，其各位数字立方和等于该数本身。

# 例如：153是一个"水仙花数"，因为153=1的三次方＋5的三次方＋3的三次方。

#方法一

# for i in range(100,1000):

# ge = i % 10

# #123 --> 12 --> 2

# shi = i // 10 % 10

# bai = i // 100

# if (ge\*\*3+shi\*\*3+bai\*\*3) == i:

# print(str(i)+'是水仙花数')

#方法二

# for bai in range(1,10):

# for shi in range(0,10):

# for ge in range(0,10):

# #将获取到的各位数字组合成真正的数字

# i = ge + shi\*10 + bai\*100

# if (ge \*\* 3 + shi \*\* 3 + bai \*\* 3) == i:

# print(str(i) + '是水仙花数')

#方法三

# 1 2 3

# 百位 十位 个位

# 123 ———> '123' int('1')\*\*3 +'2' '3'

for i in range(100,1000):

#记录每一位数字的立方累加值 默认为0

sum\_number = 0

#将数字转换为字符串 遍历获取每一个字符

for ch in str(i):

#将每一位字符再转为数字 将立方值累加到sum\_number

sum\_number += int(ch)\*\*3

#当循环结束后sum\_number保存的就是各个位的累加和

#判断

if sum\_number == i:

print(str(i) + '是水仙花数')

#累加10~50之间的个位不是3 6 9的数字

#10+11+12+14+15+17+18+20...

#获取所有10~50的数字

#获取数字的个位 如果是 3 6 9 跳过

#如果不是 累加

#输出结果

sum\_number = 0

# for i in range(10,51):

# unit = i % 10

# #如果是3 6 9 跳过

# if unit == 3 or unit == 6 or unit == 9:

# continue

# sum\_number += i

n = 10

while n<51:

unit = n % 10

#如果是3 6 9 跳过

if unit == 3 or unit == 6 or unit == 9:

continue

sum\_number += n

n += 1

print(sum\_number)

#次数未知 while循环

#次数已知/有确切数据 for循环

#在控制台输入长和宽 求矩形的周长和面积

#打印格式如下

#矩形的长为xx 宽为xx 周长是xxx 面积是 xxx

unit1 = int(input('请输入矩形的长'))

unit2 = int(input('请输入矩形的宽'))

length = (unit1+unit2)\*2

area = unit1\*unit2

print('矩形的长为%d 宽为%d 周长是%d 面积是%d' % (unit1,unit2,length,area))

# 1.在控制台输入整数作为边长 打印如下

# 如果输入4

# \*\*\*\*

# \* \*

# \* \*

# \*\*\*\*

# 如果输入 5

# \*\*\*\*\*

# \* \*

# \* \*

# \* \*

# \*\*\*\*\*

number = int(input('请输入一个整数：'))

#上边

print('\*'\*number)

#中间

for item in range(number-2):

# print('\*'+' '\*(number-2)+'\*')

print('\*%s\*' % (' '\*(number-2)))

#下边

print('\*'\*number)

# 2.在控制台输入一个字符串 判断是否为回文

# 判断规则正向与反向相同

# 上海自来水来自海上

msg = input('请输入一个字符串：')

#判断是否为回文

#判断正向与反向是否相同

if msg == msg[::-1]:

print('是回文')

else:

print('不是回文')

内建函数

'''

str1 = '123456'

print(len(str1))#长度

print(max(str1))#最大值

print(min(str1))#最小值

#求和

# print(sum(str1))#报错 字符串里面的字符不能直接相加

list1 = []

for ch in str1:

list1.append(int(ch))

sum(list1)

字符串相关方法(函数)

语法：

字符串.函数()

#判断

# str1 = 'this is a test string'

# #判断字符串中是不是全都为空白

# print(str1.isspace())

# #判断字符串中是不是全都为数字类型

# num\_str = '123456'

# print(num\_str.isdigit())

# #判断字符串是否以xx开头

# print(str1.startswith('this'))

# #判断字符串是否以xx结尾

# print(str1.endswith('ing'))

#查找

# str1 = '01234'

#查找指定字符串 返回索引值

# print(str1.find('0'))

# #如果没找到返回-1

# print(str1.find('5'))

# #可以指定开始索引和结束索引 不包含结束索引

# print(str1.find('4',0,4))#0 1 2 3 -> -1

# str2 = 'this is a test string'

# # 012345678

# print(str2.count('t'))

# #可以指定范围 没找到返回0

# print(str2.count('e',0,9))#0

#修改

#替换 将旧的字符串替换成新的字符串

# str1 = 'this is a test string'

# print(str1.replace('is','are'))

# #替换 最多换1次

# print(str1.replace('is','are',1))

# #如果指定的旧的字符串不存在 就什么都不做

# print(str1.replace('there','that'))

# str1 = ' hello '

# print(str1.lstrip())用于截掉字符串左边的空格或指定字符

# print(str1.rstrip())删除 string 字符串末尾的指定字符

# print(str1.strip())用于移除字符串头尾指定的字符（默认为空格或换行符）或字符序列。

注意：该方法只能删除开头或是结尾的字符，不能删除中间部分的字符

# print(str1.lstrip(' h'))

# print(str1.rstrip('o '))

# str1 = 'He said'

# print(str1.lower())#转小写

# print(str1.upper())#转大写

# print(str1.swapcase())#小写转大写 大写转小写

列表的基础操作

# 1.创建列表

# #空列表

# list01 = []

# #list(可迭代对象) 使用可迭代对象快速创建列表

# # list01 = list()

# # print(list01)

# # #具有值的列表

# # list02 = ['shibw',18,True,[1,2,3]]

# # print(list02)

# list02 = list(range(4))

# print(list02)

# list02 = list("我是小妖怪")

# print(list02)

#增加

#需要借助列表的相关函数

#追加元素 append()

#向列表末尾添加一个元素

# list02 = list(range(4))#[0,1,2,3]

# list02.append(4)

# print(list02)

# # for i in range(5,8):

# # list02.append(i)

# # print(list02)

# #extend()使用可迭代对象扩展列表(向列表末尾添加一系列元素)

# list02.extend(range(5,8))

# print(list02)

# #insert(索引,元素) 插入 [0,1,2,3,4,5,6,7]

# list02.insert(0,0)

# print(list02)

# 列表的索引和切片

# list01 = [0,1,2,3,4,5]

# print(list01[0:5])#[0,1,2,3,4]

# print(list01[::-1])#[5,4,3,2,1,0]

# print(list01[-1])

# 修改

# list01[-1] = 5.0

# print(list01)

# print(list01[-2:])#倒数第二个 到最后一个

#获取列表的所有元素

# list01 = [0,1,2,3,4]

# for item in list01:

# print(item)

#删除

#根据元素删除

# list01 = [0,1,2,3,4]

# list01.remove(4)

# print(list01)

# #根据索引

# del list01[3]

# print(list01)

# list01 = ['三丰','翠山','无忌']

# #把list01中储存的地址赋值给list02

# list02 = list01

# list01[0] = '张三丰'

# print(list02[0])#张三丰

# list01 = [800,900,1000]

# list02 = list01

# list03 = list01

# list01[0] = '八百'

# print(list02[0])#?'八百'

# list03 = '九百'

# print(list02)#?

# list01 = [800,900,1000]

# list02 = list01[:]

# list01[0] = '八百'

# print(list02[0])#?

# list01 = [800,900,1000]

# list02 = list01

# list01[1:2] = ['a','b']

# print(list02)#?[800,'a','b',1000]

# list01 = [100,[200,300]]

# list02 = list01

# #list01[1] = [200,300]

# #[200,300][0] = 200

# list01[1][0] = 500

# print(list02)#[100,[500,300]]

# list01 = [100,[200,300]]

# list02 = list01.copy()#浅拷贝

# list01[1][0] = 500

# print(list02[1][0])#500

# list01 = [1000,2000,3000]

# list02 = [1000,2000,3000]

# print(list01 == list02)#比较的是值是否相等

# print(list01 is list02)#比较的是地址id(list01) == id(list02)

#准备拷贝工具

import copy

list01 = [100,[200,300]]

深拷贝 划清界限 拷贝前的对象和拷贝后的对象互不影响

#注意 深拷贝可能会占用大量内存

list02 = copy.deepcopy(list01)

list01[1][0] = 500

print(list02)

list --> str

'''

list01 = ['a','b','c']

str01 = '+'.join(list01)

print(str01)

print(type(str01))

# str\_result = ''

# for i in range(10):

# # str\_result += str(i)

# # '' + '0' 产生新的字符串 '0'

# # '0' + '1' 产生新的字符串 '01'

# # '01' + '2' 产生新的字符串 '012'

# str\_result = str\_result + str(i)

#思想：通过可变对象list保存所有要拼接的字符串 最后在将列表里的内容转为字符串

list\_result = []

for i in range(10):

list\_result.append(str(i))

str\_result = ''.join(list\_result)

print(str\_result)

str --> list

'''

# str01 = '2019-08-06'

str01 = '2019/08/06'

list\_result = str01.split('/')

print(list\_result)

#将一句英文的单词反转

#How are you --> you are How

#将字符串按照空格拆分成列表

#将列表反转

#将列表中的元素拼接为字符串

message = 'How are you'

temp\_list = message.split(' ')

message = " ".join(temp\_list[::-1])

print(message)

#在控制台输入一个字符串

#打印 第一个字符是xx

#打印 倒数第二个字符是xx

#打印 前2个字符是xx

#倒序打印所有字符串

#打印所有正向索引是奇数的字符

# 0 1 2 3 4 5

#如果字符串的长度是奇数 则打印中间的字符

# 如 12345 ———> 3

msg = input('请输入一个字符串')

print('第一个字符是%s' % msg[0])

print('倒数第二个字符是%s' % msg[-2])

print('前2个字符是%s' % msg[:2])

print(msg[::-1])

#打印所有正向索引是奇数的字符

print(msg[1::2])

#如果字符串的长度是奇数 则打印中间的字符

count = 0

for i in msg:

count+=1

print(count)

# if len(msg) % 2 != 0: 休息：10:00~10:15

# #打印中间字符

# #12345 5 [2] 1234567 7 [3]

# print(msg[len(msg)//2])

length = len(msg)

if length % 2:

#len(msg) % 2 == 1 是奇数时运行下面代码

print(msg[length // 2])

'''列表的索引赋值'''

list01 = ['a','b','c']

#用过切片定位前两个元素 将结果修改为[0,1,2]

#['a','b'] = [0,1,2]

list01[0:2] = [0,1,2]

#查看列表元素 画内存图

print(list01)#[0, 1, 2, 'c']

#通过切片获取到三个元素 修改为0个元素（删除）

list01[:3] = []

print(list01)

#定义列表 存储八大行星

#水 金 地 火 木 土 天王 海王

#打印 距离太阳最近的行星是xx

#打印 距离太阳最远的行星是xx

#打印 太阳到地球之间的行星

#打印 八大行星(一行一个)

#倒序打印 八大行星(从又向左 一行一个)

list\_planets = ['水星','金星','地球','火星','木星','土星','天王星','海王星']

# print('距离太阳最近的行星是%s' % list\_planets[0] )

# print('距离太阳最远的行星是%s' % list\_planets[-1] )

# print(list\_planets[0:2])

# for item in list\_planets:

# print(item)

#倒序获取所有行星

#如果采用切片的方式 会重新创建新列表 不建议

# for item in list\_planets[::-1]:

# print(item)

#['水星','金星','地球','火星','木星','土星','天王星','海王星']

# 0 1 2 3 4 5 6

# 7

# 7~-1

#建议通过索引 在原列表直接查找对应位置的值

for item in range(len(list\_planets)-1,-1,-1):

print(list\_planets[item])

#在控制台获取学生姓名 (循环输入 一个一个录入)

#将输入结果保存到列表

#如果输入的是空字符串 则停止录入

#打印所有学生姓名

name\_list = [] #空列表 保存数据

#循环添加数据到空列表

while True:

name = input('请输入姓名：')

if name == '':

break

name\_list.append(name)

#打印列表的每一个元素

for name in name\_list:

print(name)

#在控制台获取所有学生的成绩(循环 一个一个录入)

#如果录入空字符串 停止

#输出最高分 最低分 和平均分

scores\_list = []

while True:

str\_score = input('请输入成绩：')

if str\_score == '':

break

score = int(str\_score)

scores\_list.append(score)

print(max(scores\_list))

print(min(scores\_list))

print(sum(scores\_list)/len(scores\_list))

list01 = [3,45,8,12,36,7,3]

# 0 1 2 3 4 5 6

#把list01中大于10的数字存入另一个列表

# result\_list = []

# for item in list01:

# if item > 10:

# result\_list.append(item)

# print(result\_list)

#获取list01中最大的数 不使用max()

#假设最大值为列表的第一个数

max\_value = list01[0]

# #循环取出列表剩余的全部数据和最大值比较

# for item in list01[1:]:

# #如果当前项比最大值大 将最大值赋值为当前项

# if item > max\_value:

# max\_value = item

# for i in range(1,len(list01)):

# if list01[i] > max\_value:

# max\_value = list01[i]

# #循环结束后 打印最大值

# print(max\_value)

#删除列表list01中所有的奇数

# for item in list01:

# if item % 2:

# list01.remove(item)

# print(list01)

#倒序删除

for i in range(len(list01)-1,-1,-1):

if list01[i] % 2:

print(list01[i])

# list01.remove(list01[i])

del list01[i]

print(list01)

#斐波那契数列

特点 ：每一个值都是前两个数相加的结果

fibs = [0,1]

#[0,1,1]

#[0,1,1,2]

# #[0,1,1,2,3,5,8,11...]

# first = fibs[0]

# second = fibs[1]

# for i in range(13):

# result = first + second

# fibs.append(result)

# #当结果添加到列表后 将前两个数替换

# first = second

# second = result

# print(fibs)

for i in range(13):

fibs.append(fibs[-1]+fibs[-2])

print(fibs)

list\_result = []

while True:

str\_input = input('请输入：')

if str\_input == '':

break

list\_result.append(str\_input)

str\_result = '\n'.join(list\_result)

print(str\_result)

list01 = ['a','b','c']

list01[0] = ['A','B']

print(list01)#[['A','B'],'b','c']

#将右边列表的值赋值给list01的第一个位置

list01[1:2] = ['哪吒']

#序列赋值

# name,age = ['shibw',18]

# print(name,age)

print(list01)#[['A','B'],'哪吒','c']

list02 = list01[::-1]

print(list02)#['c','哪吒',['A','B']]

print(list01[0] is list02[-1])

list01[0] = '李靖'

print(list02[-1])

列表推导式

快速的将可迭代对象变成列表

# list01 = [10,1,15,5,6,7]

# #将list01中的每一个数加一放到list02中

# list02 = []

# for item in list01:

# list02.append(item+1)

# print(list02)

# #从可迭代对象list01中获取元素

# #将元素带入到前面的表达式item+1

# #将结果保存到列表 继续从list01获取下一个元素 直到没有元素为止

# list03 = [item+1 for item in list01]

# print(list03)

# list04 = []

# for item in list01:

# if item >= 10:

# list04.append(item+1)

# print(list04)

# #从可迭代对象list01中获取元素

# #先做判断 如果结果为False 取下一个元素继续判断

# #如果结果为True 将元素带入到前面的表达式

# # 将结果保存到列表 重复执行

# list05 = [item+1 for item in list01 if item >= 10]

# print(list05)

#列表推导式嵌套

list01 = [1,2,3]

list02 = [4,5,6]

# list03 = []

# # 将两个列表中所有的值分别相加 将结果保存到list03

# for item1 in list01:

# for item2 in list02:

# list03.append(item1+item2)

list03 = [item1+item2 for item1 in list01 for item2 in list02]

print(list03)

元组的基本操作

#创建元组

# tuple01 = ()#空元组

# tuple01 = tuple()

# print(tuple01)

# tuple02 = (0,1,'nihao',True)

# print(tuple02)

# #预留空间的存储机制(列表) --> 按需分配的存储机制(元组)

# list01 = ['a','b']

# tuple02 = tuple(list01)

# print(tuple02)

# list02 = list(tuple02)

# print(list02)

# tuple02 = ('a')#不是元组

# tuple02 = ('a',)#如果元组中只有一个元素 必须加逗号

# print(type(tuple02))

# name,age = 'shibw',18#('shibw',18)

#获取元素

# tuple01 = ('a','b','c','d')

# print(tuple01[2])

# # tuple01[2] = 'C' #元组不可变 不能赋值

# print(tuple01[1:3])

# #所有元素

# for item in tuple01:

# print(item)

# for i in range(len(tuple01)-1,-1,-1):

# print(tuple01[i])

# del tuple01[3]#元组不可变 不能删除元素

tuple01 = ([1,2],3,4)

print(tuple01[0][1])

tuple01[0][1] = 2.0元组无法改变 元组中的列表可改变

print(tuple01)

字典的基础操作

#创建字典

# dict01 = {}#空字典

# dict01 = dict()

# print(dict01)

# #"shibw":20

# # 键 ：值 键值对

# dict02 = {"shibw":1311794174}

# #字典中的数据以键值对的形式保存

# #多个键值对用逗号分隔

# dict02 = {'shibw':20,'wwc':30}

#通过函数创建字典

# dict02 = dict([['name','shibw']])

# print(dict02)#{'name': 'shibw'}

# dict02 = dict([['name','shibw'],('age',20)])

# print(dict02)#{'name': 'shibw', 'age': 20}

# #添加元素

# dict01 = {'name':'shibw','age':20}

# #如果字典中没有对应的键存在

# #可以直接对不存在的键赋值 向字典中添加该键值对

# dict01['address'] = 'beijing'

# print(dict01)

# #修改

# #如果对已存在的键赋值 相当于修改原本键的值

# dict01['address'] = 'qiaowan'

# print(dict01)

# #删除

# del dict01['address']

# print(dict01)

#查找字典中的元素

# dict01 = {'name':'shibw','age':20}

# print(dict01['name'])

# print(dict01['address'])#KeyError

# if 'name' in dict01:

# print(dict01['name'])

# if 'address' in dict01:

# print(dict01['address'])

#字典的常用方法

#使用get 获取字典指定键的值

# print(dict01.get('name'))

#如果键不存在 可以指定返回默认值

# print(dict01.get('address','beijing'))

#同get 但是键不存在时 会添加键值对

# print(dict01.setdefault('address','beijing'))

# print(dict01)

# dict01.popitem()

# print(dict01)

# dict01 = {'name':'shibw','age':20}

# dict02 = {'name':'laowang','address':'北京'}

# #使用02更新字典01 如果02中的键在01中不存在 添加

# #如果02的键在01中存在 使用02的值

# dict01.update(dict02)

# print(dict01)

#获取字典所有元素

# for key in dict01:

# print(key)#获取的是字典中所有的键

# print(dict01[key])

#获取键值对(键，值)

# for item in dict01.items():

# print(item)

# for key,value in dict01.items():

# print(key)

# print(value)

# 获取字典的所有的值

# for value in dict01.values():

# print(value)

# dict01 = {1:"1.0",2:"2.0"}

# print(dict01)

# dict01 = {(1,2,3):3,(1,2,3,4):4}

# print(dict01)

# #可变类型无法通过哈希算法运算

# #所以字典的键都是不可变类型

# dict01 = {[1,2,3]:3}#TypeError

# print(dict01)

字典推导式

#需求 range(10) 作为key 值为key的平方

dict01 = {}

for i in range(10):

dict01[i] = i\*\*2

print(dict01)

#字典推导式

dict01 = {i:i\*\*2 for i in range(10)}

print(dict01)

#需求 range(10)大于5的 作为key 值为key的平方

dict02 = {}

for i in range(10):

if i > 5:

dict02[i] = i\*\*2

print(dict02)

dict02 = {i:i\*\*2 for i in range(10) if i > 5}

print(dict02)

#生成1～10的数字 将数字的平方放入列表list01

# list01 = []

# for i in range(1,11):

# list01.append(i\*\*2)

list01 = [i\*\*2 for i in range(1,11)]

print(list01)

#将list01中所有的奇数放入list02

# list02 = []

# for item in list01:

# if item % 2:

# list02.append(item)

list02 = [item for item in list01 if item % 2]

print(list02)

#将list01中所有的偶数加10放入list03

list03 = []

for item in list01:

if item % 2 == 0:

list03.append(item+10)

list03 = [item+10 for item in list01 if item % 2 == 0]

print(list03)

#获取用户输入的年份和月份 打印这个月份有多少天

#使用容器的思想去处理代码

year = int(input('请输入年份：'))

month = int(input('请输入月份：'))

# if month<1 or month>12:

# print('月份输入错误')

# elif month == 2:

# #平年28 闰年29

# if year%4==0 and year%100!=0 or year % 400 == 0:

# print('29天')

# else:

# print('28天')

# elif month == 4 or month == 6 or month == 9 or month == 11:

# print('30天')

# else:

# print('31天')

#1.判断条件 多个条件时 考虑用容器保存所有可能的结果 然后再判断

# if month<1 or month>12:

# print('月份输入错误')

# elif month == 2:

# #平年28 闰年29

# if year%4==0 and year%100!=0 or year % 400 == 0:

# print('29天')

# else:

# print('28天')

# elif month in (4,6,9,11):

# print('30天')

# else:

# print('31天')

#2.当结果值确定时 考虑先将结果保存到容器中 然后根据条件去容器中获取对应结果

# if year % 4 == 0 and year % 100 != 0 or year % 400 == 0:

# month02 = 29

# else:

# month02 = 28

if month<1 or month>12:

print('月份输入错误')

else:

month02 = 29 if year % 4 == 0 and year % 100 != 0 or year % 400 == 0 else 28

days\_of\_month = (31,month02,31,30,31,30,31,31,30,31,30,31)

print(days\_of\_month[month-1])

#3.获取用户输入的年份和月份、和日期

#计算这是一年的第几天

#2019年3月10日 31+28+10

year = int(input('请输入年份：'))

month = int(input('请输入月份：'))

day = int(input('请输入日期:'))

if month<1 or month>12:

print('月份输入错误')

else:

month02 = 29 if year % 4 == 0 and year % 100 != 0 or year % 400 == 0 else 28

days\_of\_month = (31,month02,31,30,31,30,31,31,30,31,30,31)

print(days\_of\_month[month-1])

#先累加前几个月的天数

#再累加当月的天数

#方法一

# total\_day = 0

# for i in range(month-1):

# total\_day += days\_of\_month[i]

#方法二

#使用sum函数 获取days\_of\_month中索引值从0到月份-1的所有值的和

total\_day = sum(days\_of\_month[:month-1])

total\_day += day

print(total\_day)

#在控制台输入商品信息(名称和单价)

#如果名称为空 停止输入

#将商品的名称与单价打印在控制台

#如果录入了 “游戏机” 单独打印 游戏机的价格是xxx

dict\_info = {}

while True:

name = input('请输入商品名称：')

if name == '':

break

price = float(input('请输入商品价格：'))

dict\_info[name] = price

for key,value in dict\_info.items():

print('%s的单价为%.1f' % (key,value))

if '游戏机' in dict\_info:

print('游戏机的价格是%.1f' % dict\_info['游戏机'])

#在控制台录入人的信息（姓名、年龄、性别、体重）

#如果名称为空字符串 停止

#将所有人的信息打印

#数据结构

# {'name':'shibw','age':20,'gender':'男','weight':80},

# {....}

list\_info = []

while True:

name = input('请输入姓名：')

if name == '':

break

age = int(input('请输入年龄:'))

gender = input('请输入性别：')

weight = float(input('请输入体重：'))

#创建保存用户信息的字典

dict\_info = {'name':name,'age':age,'gender':gender,'weight':weight}

#将字典添加到列表

list\_info.append(dict\_info)

print(list\_info)

#获取列表中的每条数据

for dict\_item in list\_info:

#列表的每条数据都是字典 通过字典的键查找

print('%s的年龄是%d,性别是%s,体重是%.1f' % (dict\_item['name'],dict\_item['age'],dict\_item['gender'],dict\_item['weight']))

#在控制台录入人的信息（姓名、年龄、性别、体重）

#如果名称为空字符串 停止

#将所有人的信息打印

#数据结构

# {

# 'shibw':[18,'男',80.0],

# 'mengmeng':[...],

# }

#定义空字典保存数据

dict\_info = {}

#循环获取用户输入的姓名、年龄、性别、体重

while True:

name = input('请输入姓名：')

# 如果姓名为空 退出循环

if name == '':

break

age = int(input('请输入年龄:'))

gender = input('请输入性别：')

weight = float(input('请输入体重：'))

#将姓名作为键 [年龄、性别、体重]作为值添加到字典中

dict\_info[name] = [age,gender,weight]

#从字典中获取每一个键值对

#打印

for key,value in dict\_info.items():

print('%s的年龄是%d,性别是%s,体重是%.1f' % (key,value[0],value[1],value[2]))

存储多个数据时，使用什么样的额数据结构？

根据需求 结合本身的优缺点 综合考虑

列表：

优点：添加/获取元素时比较灵活 可使用索引/切片操作

缺点：数据量大时 查找慢 通过索引获取值时，如果信息多，代码可读性差

字典：

优点：根据key获取value 速度块

如果获取数据较多时 通过键获取值代码可读性较好

缺：不能使用索引和切片操作元素 不灵活占用内存较大

#['张三丰','翠山','无忌'] -->{'无忌'：2,'翠山'：2,'张三丰':3}

#list ——> dict key是列表的元素 value是key的长度

list\_name = ['张三丰','翠山','无忌']

# dict\_name = {}

# for item in list\_name:

# dict\_name[item] = len(item)

dict\_name = {item:len(item) for item in list\_name}

print(dict\_name)

#姓名列表['无忌','赵敏','芷若']

#房间列表[101,102,103]

#把两个列表合并为一个字典 姓名作为key 房间号作为value

name\_list = ['无忌','赵敏','芷若']

room\_list = [101,102,103]

# dict\_name\_room = {}

# for i in range(len(name\_list)):

# dict\_name\_room[name\_list[i]] = room\_list[i]

dict\_name\_room = {name\_list[i]:room\_list[i] for i in range(len(name\_list))}

print(dict\_name\_room)