

Python与金融数据挖掘(2)

文欣秀

wenxinxiu@ecust.edu.cn

程序设计方法



图灵奖获得者沃思

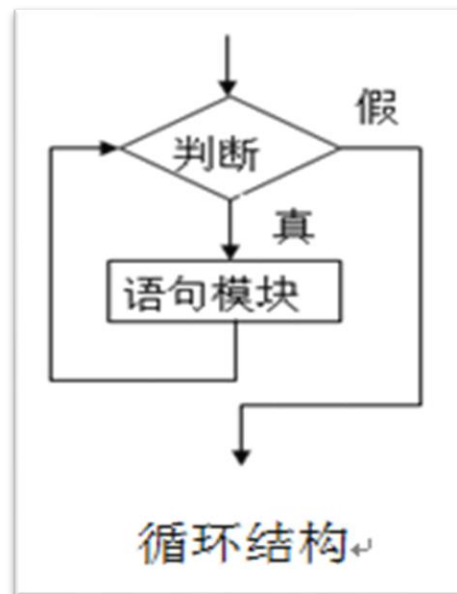
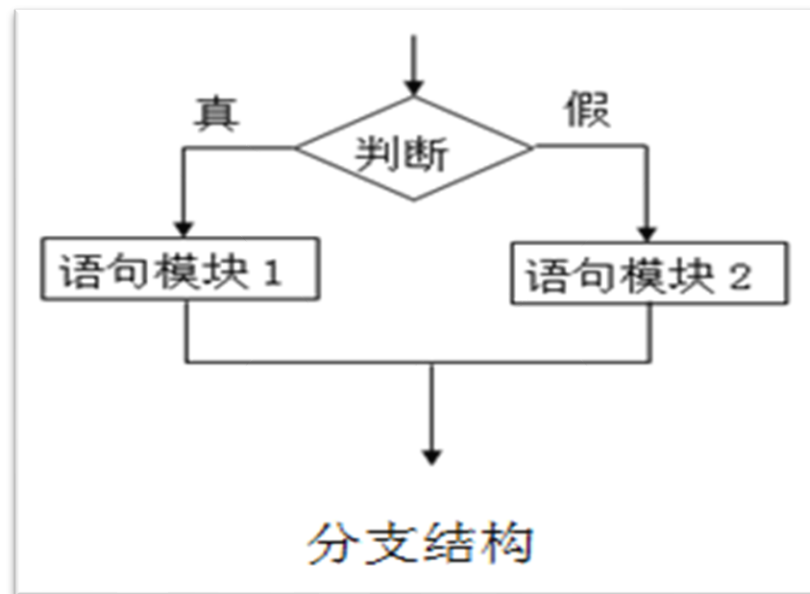
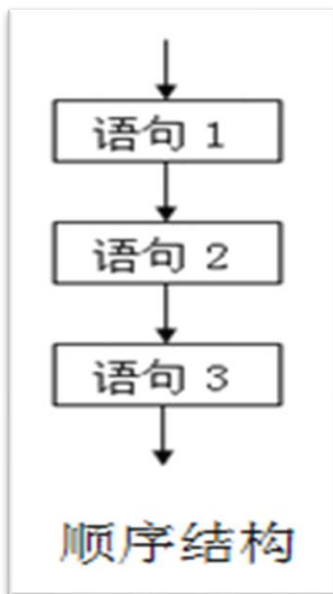
算法+数据结构=程序

Algorithm+Data Structures=Programs

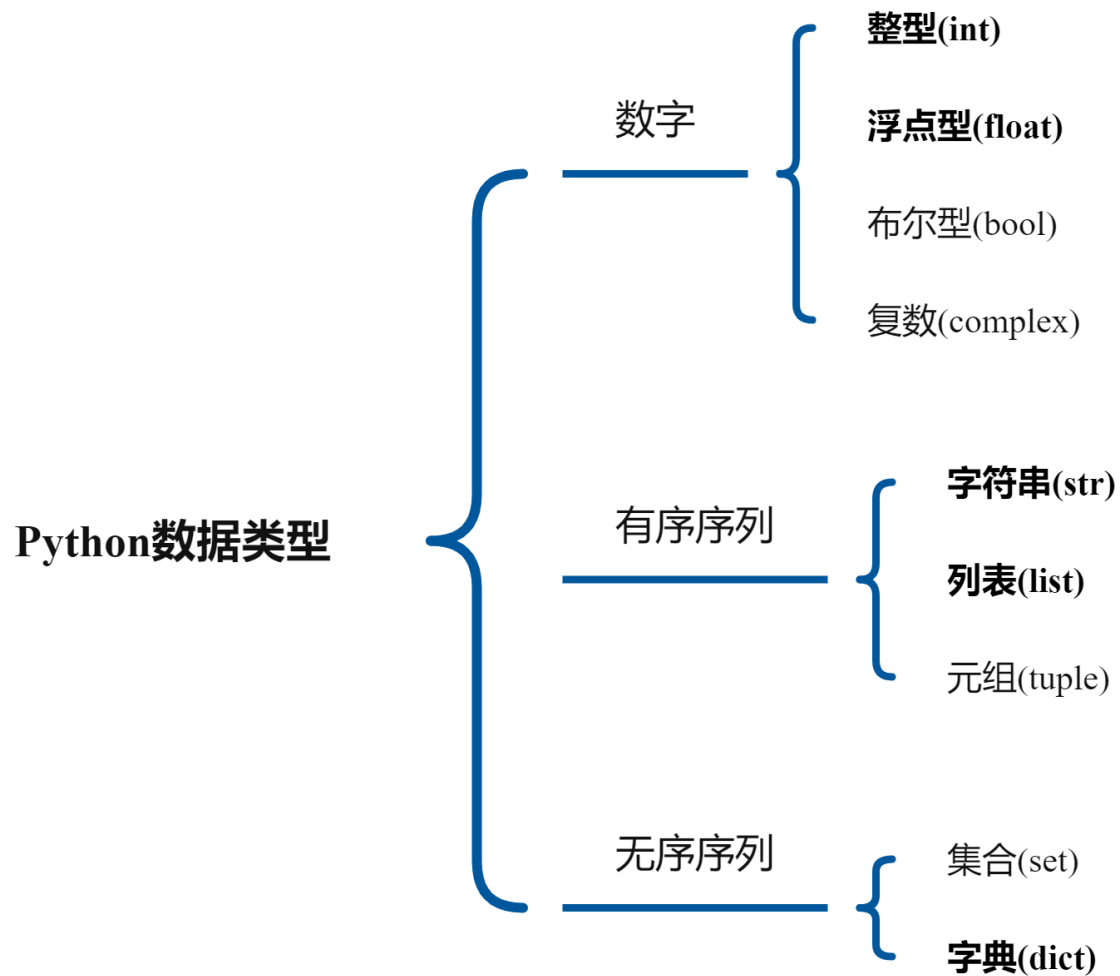
算法：对操作的描述，即要求计算机进行操作的步骤。

数据结构：对数据的描述，程序中用到数据的类型及数据的组织形式。

三种基本结构



Python数据类型



程序输入

input(): 接收从键盘输入的字符串

```
>>> myAge = int(input("请输入购买股数: "))
```

int(): 将字符串转换为整数

```
>>> myScore = float(input("请输入开盘价: "))
```

float(): 将字符串转换为单精度数

格式输出

```
>>> name, high="中国银行", 5.44
```

```
>>> print("名称:%s, 最高价:%.2f" % (name, high))
```

```
>>> print("名称:{},最高价:{:.2f}".format(name, high))
```

```
>>> print(f"名称:{name}, 最高价:{high:.2f}")
```

format使用方法

格式： <模板字符串>.format(<逗号分隔的参数>)

例如：

```
>>> print("{}今天最高价{}元".format("中国银行",5.44))
```

```
>>> print("{1}今天最高价{0}元".format(5.44,"中国银行"))
```

注意： 参数序号从0开始

format模板字符串

模板字符串格式：{<参数序号>:<格式控制标记>}

格式控制标记：<填充><对齐><宽度><,><.精度><类型>

整数	:	<填充>	<对齐>	<宽度>	<,>	<.精度>	<类型>
参数序号	引导符号	填充字符 默认使用空格填充	<: 左对齐 >: 右对齐 ^: 居中	输出宽度大于实际宽度默认补空格, 小于实际宽度按实际输出	数字千位分隔符	浮点数小数位或字符串最大输出长度	数字类型: b:二进制 d:十进制 ...

format应用案例一

```
>>> print("{0:*^11}".format("我喜欢投资"))
```

```
***我喜欢投资***
```

```
>>> print("{0:#<11}".format("我喜欢投资"))
```

```
我喜欢投资#####
```

```
>>> print("{0:@>11}".format("我喜欢投资"))
```

```
@ @ @ @ @ @我喜欢投资
```

format模板字符串

模板字符串格式：{<参数序号>:<格式控制标记>}

格式控制标记：<填充><对齐><宽度><,><.精度><类型>

整数	:	<填充>	<对齐>	<宽度>	<,>	<.精度>	<类型>
参数序号	引导符号	填充字符默认使用空格填充	<: 左对齐 >: 右对齐 ^: 居中	输出宽度大于实际宽度默认补空格, 小于实际宽度按实际输出	数字千位分隔符	浮点数小数位数或字符串最大输出长度	数字类型: b:二进制 d:十进制 ...

format应用案例二

```
>>> print('{:b}'.format(5))
```

101

```
>>> print("{:5,.2f}RMB".format(12345.6))
```

12,345.60RMB

```
>>> print("{0:.4s}{1}".format("上海证券交易所缩写是","SH"))
```

上海证券SH

在职党员交纳党费比例问题

党费缴纳标准： 每月工资收入(税后)在3000元以下(含3000元)者， 交纳月工资收入的0.5%； 3000元以上至5000元(含5000元)者， 交纳1%； 5000元以上至10000元(含10000元)者， 交纳1.5%；10000元以上者， 交纳2%。

编写程序， 输入当月工资， 计算并输出应交党费。

标准比较运算符

标准比较运算符: < <= > >= == !=

```
>>> 3.14 <= 3.14159
```

```
>>> 'A' <= 'B'
```

```
>>> 'a' == 'A'
```

```
>>> chr(ord('N')+32)
```

ord()函数: 就是用来返回单个字符的ASCII值

chr()函数: 返回一个整数对应的ASCII符号

逻辑运算符

逻辑运算符: and or not

```
>>> age=int(input("请输入年龄: "))
```

```
>>> age>=18 and age<=60
```

```
>>> age<18 or age>60
```

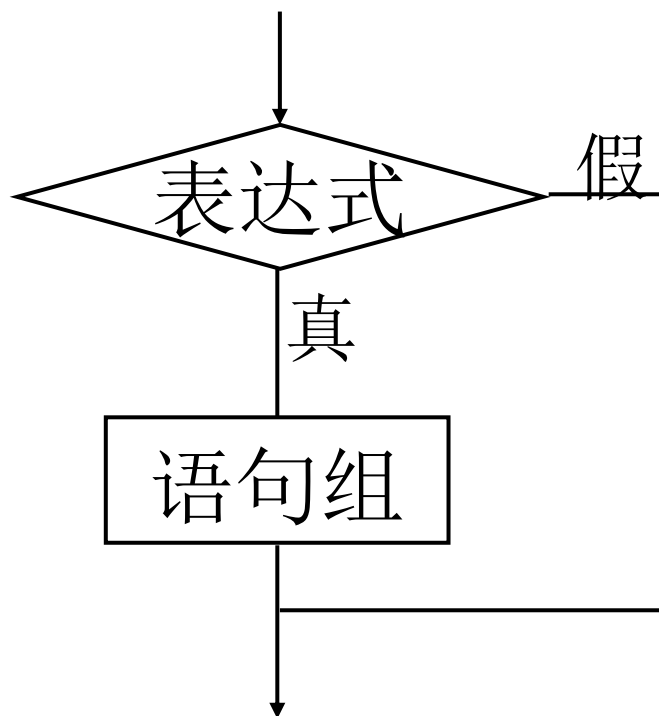
```
>>> not age<18
```

age>=18 and age<=60 等价于 18 <= age <=60

选择结构

- 单分支结构
- 双分支结构
- 多分支结构

单分支结构

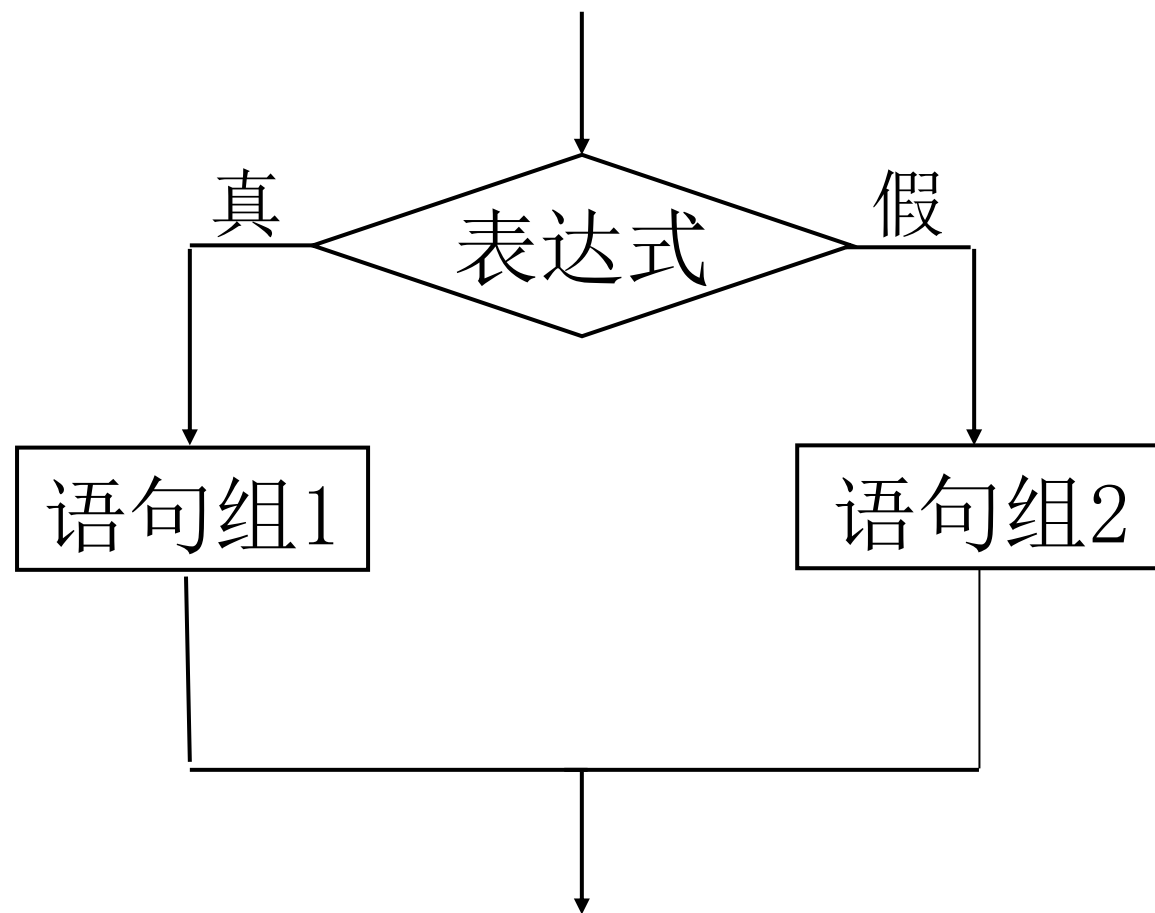


单分支条件语句

if 表达式:
语句组

```
import os
seconds=int(input("您希望多少秒之后关机:"))
os. popen("shutdown -s -t %d" % seconds)
flag=input("是否取消关机Y/N:")
if flag=='Y':
    os. popen("shutdown -a")
```

双分支结构

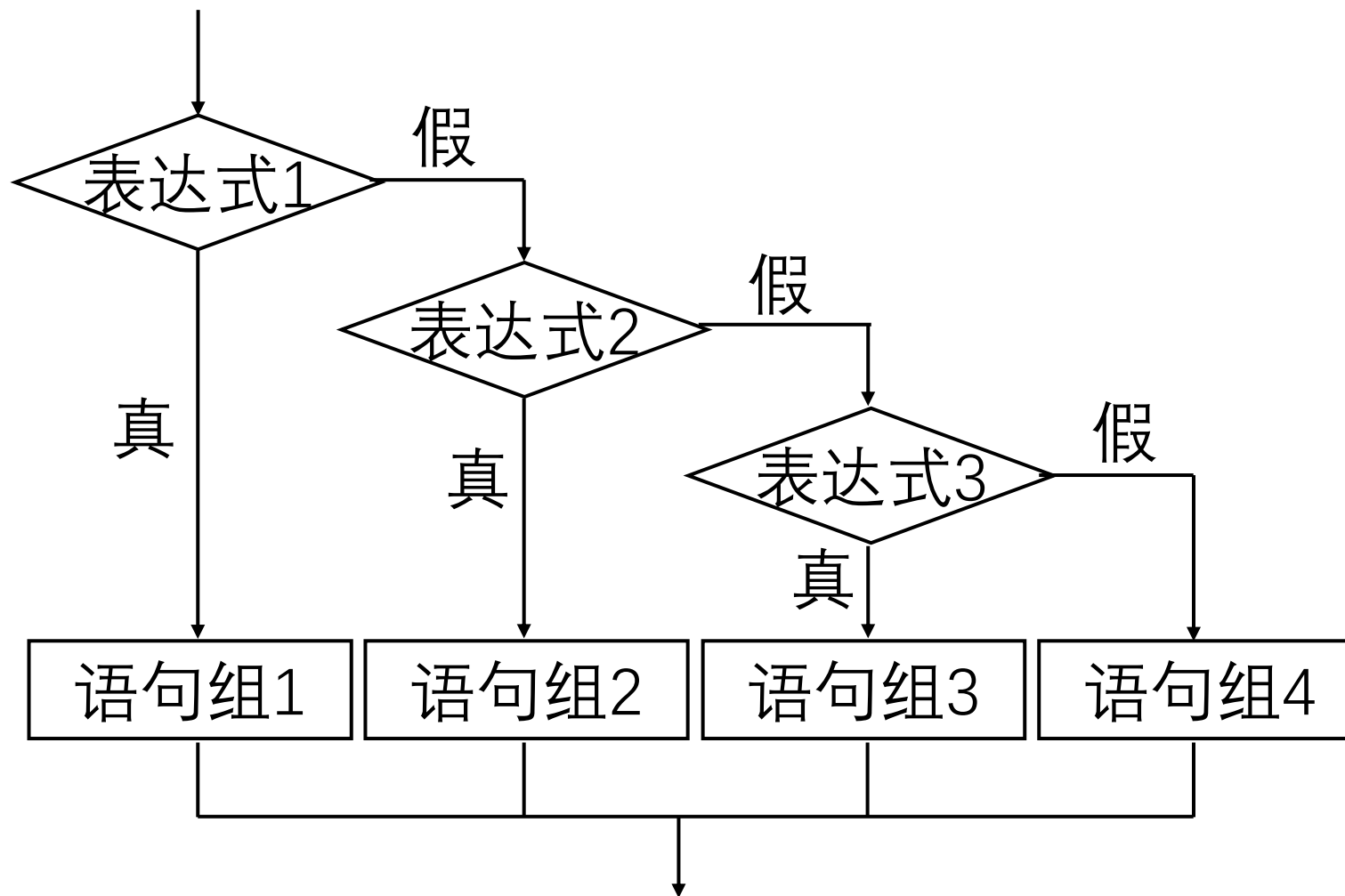


双分支条件语句

```
if 表达式:
    语句组一
else:
    语句组二
```

```
import datetime
date=datetime.datetime.now()
if date.day<=15:
    print("还有时间，加油!!!")
else:
    print("本月又过了一半!!!")
```

多分支结构



多分支条件语句

```
if 表达式1:
    语句组一
elif 表达式2:
    语句组二
...
...
else:
    语句组三
```

```
import qrcode          #第三方库，需要安装
goal=input("请输入目标：")
if goal=="中新经纬":
    img=qrcode.make("https://www.jwview.com/")
elif goal=="经济观察网":
    img=qrcode.make("https://www.eeo.com.cn/")
else:                  #中国经济网
    img=qrcode.make("http://www.ce.cn/")
img.save("caijing.png")
```

在职党员交纳党费比例问题

```
salary=float(input("请输入本月税后工资: "))
if salary<=3000:
    money=salary*0.005
elif salary<=5000:
    money=salary*0.01
elif salary<=10000:
    money=salary*0.015
else:
    money=salary*0.02
print(f"本月应缴纳党费{money:.2f}元")
```

猜测随机数问题

编写程序，某银行计划生成一个100到500之间的随机红包，然后给五次机会让在场客户猜测这个数字。程序给出猜测结果（太大、太小、成功），如果猜测成功则中断循环将红包赠予客户或机会用完结束循环。

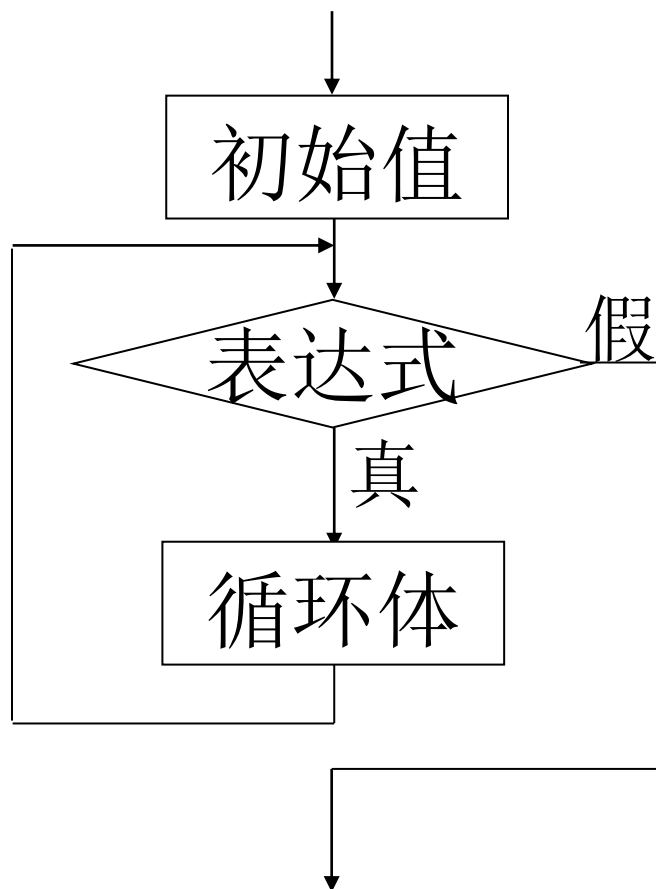
```
请输入您的猜测：300  
您猜的太小  
请输入您的猜测：400  
您猜的太大  
请输入您的猜测：350  
您猜的太大  
请输入您的猜测：320  
您猜的太小  
请输入您的猜测：330  
您猜的太小  
红包金额为348
```

循环结构

➤ **WHILE**循环

➤ **FOR**循环

WHILE循环语句



while语句流程图

WHILE循环语句

while 条件表达式:
语句组

例： time=8
while time<12:
 print("Doing homework")
 time=time+1

循环中断语句

➤ **break**

中断循环的执行，跳出循环体

➤ **continue**

中断本次循环，进入下一次循环判断

猜测随机红包问题

```
import random
money=random.randint(100,500)
count=1
while count<=5:
    guess=int(input("请输入您的猜测: "))
    if money>guess:
        print("您猜的太小")
    elif money<guess:
        print("您猜的太大")
    else:
        print("您猜的正好")
        break
    count=count+1
print("红包金额为{}".format(money))
```

range() 函数

range()函数：生成整数序列，由三个参数决定序列的大小和范围：起始值、终值和步长。是半开区间，不包括序列的终值。

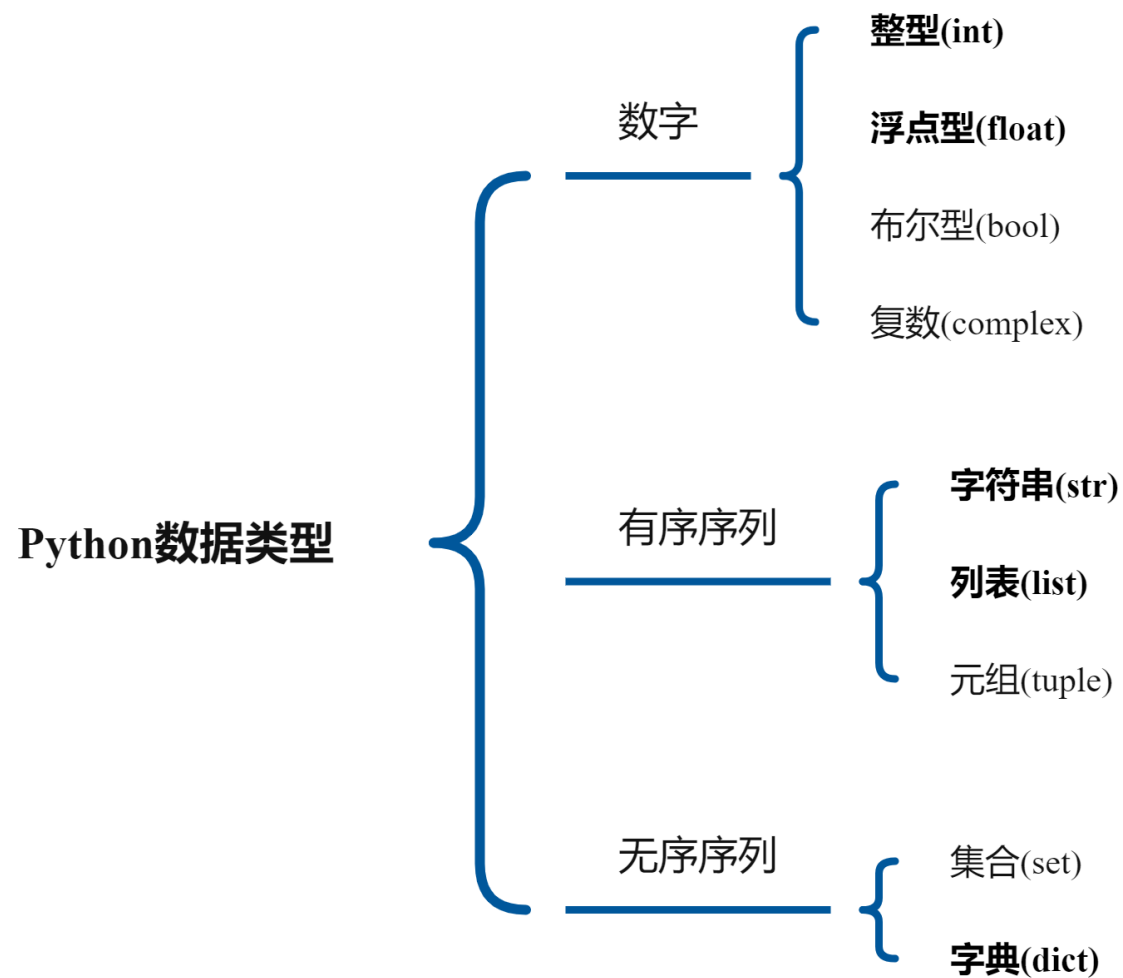
```
for i in range(4,10,2):  
    print(i, end=" ")
```

```
for i in range(5):  
    print(i, end=",")
```

猜测随机红包问题

```
import random
money=random.randint(100,500)
for i in range(1,6):
    guess=int(input("请输入您的猜测: "))
    if money>guess:
        print("您猜的太小")
    elif money<guess:
        print("您猜的太大")
    else:
        print("您猜的正好")
        break
print("红包金额为{}".format(money))
```

Python数据类型



单词/词语替换问题

编写程序实现替换功能：从网上拷贝一篇财经短文存入字符串中，实现该短文中某单词/词语的全部替换。

请输入原文：财经新闻属于新闻的一个细分类目，侧重点是采集、报道、发布财经领域的新闻。财经新闻有广义和狭义之分。广义的财经新闻或称泛经济新闻，覆盖全部社会经济生活和与经济有关的领域。狭义的财经新闻，则重点关注资本市场，并用金融资本市场的视角看中国经济主义生活。

被替代字：财经新闻

替代为字：FN

FN属于新闻的一个细分类目，侧重点是采集、报道、发布财经领域的新闻。FN有广义和狭义之分。广义的FN或称泛经济新闻，覆盖全部社会经济生活和与经济有关的领域。狭义的FN，则重点关注资本市场，并用金融资本市场的视角看中国经济主义生活。

字符串

- ◆ 包含在单引号、双引号、三引号之间的字符集合
- ◆ 索引运算符[**i**]得到下标为*i*的字符
- ◆ 第一个字符索引为 **0**，最后一个字符索引为**-1**
- ◆ 切片运算符[**i : j**]得到从下标*i*到下标**j-1**的子串
- ◆ 切片运算符[**i : j : k**]中，**k**为步长(可为负数)

字符串切片示例

```
>>> fstr="Financial News"
```

```
>>> fstr[0]
```

```
>>> fstr[-4]
```

```
>>> fstr[3:5]
```

```
>>> fstr[::-1]
```

字符串运算

- ◆ 加号(+)用于字符串连接运算
- ◆ 星号(*)用于字符串复制
- ◆ **in**用于判断某个子串在字符串中
- ◆ **not in**用于判断某个子串不在字符串中

字符串运算示例

```
>>> fstr="Financial"
```

```
>>> nstr="News!"
```

```
>>> result=fstr+' '+nstr
```

```
>>> nstr * 3
```

```
>>> nstr in result
```

常用字符串方法

s. upper(): 将字符串都转换成大写字母

s. lower(): 将字符串都转换成小写字母

s. split(): 实现字符串的分割

s. replace(): 用第二个子串替代第一个子串

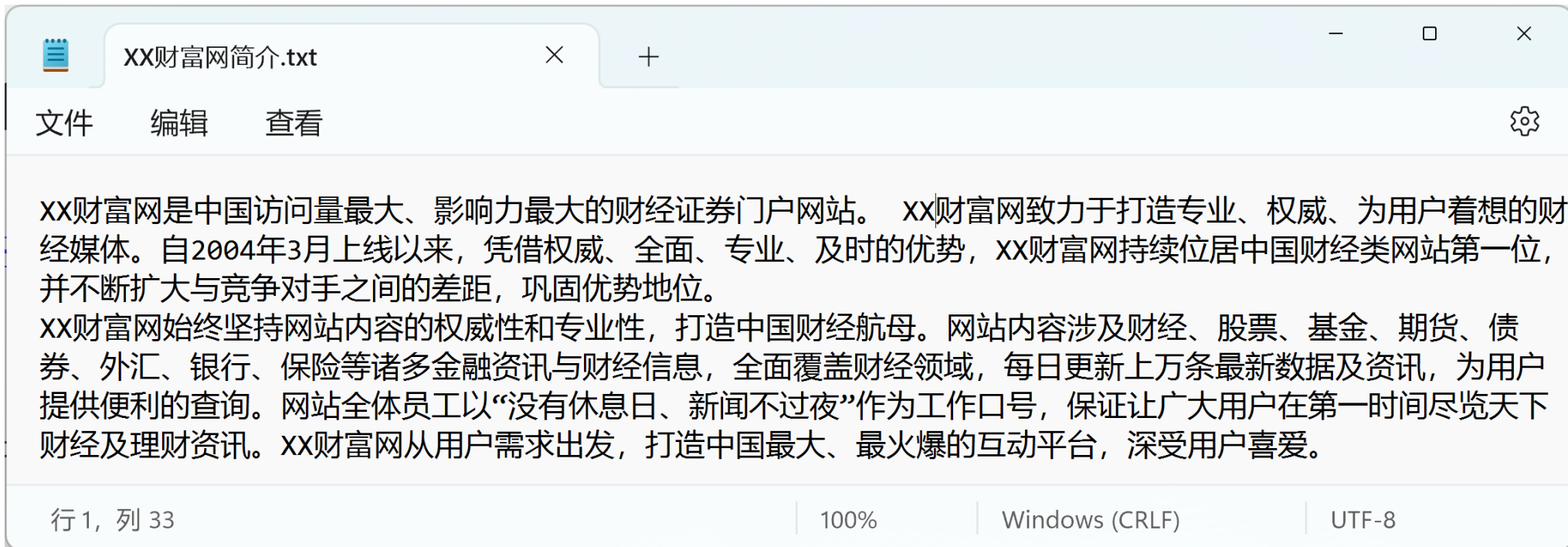
s. strip(): 消除字符串两端的空格及符号

单词/词语替换问题答案

```
paper=input("请输入原文： ")  
before=input("被替代字词： ")  
after=input("替代为字词： ")  
result=paper. replace(before, after)  
print(result)
```

拓展问题

编写程序，实现文件“xx财富网简介.txt”中的XX全部替换。



文件操作

文本文件：基于字符编码。在Python3中，文本文件以字符的Unicode码值进行存储和编码。

二进制文件：基于值编码的文件，存储的是二进制数据。

打开文件



读写文件



关闭文件

文件操作

- ◆ 文件打开方式一(打开指定位置的文件):

```
handle = open("D:\\exam\\test.txt", 'r')
```

- ◆ 文件打开方式二(打开当前目录下文件):

```
handle = open("test.txt", 'r')
```

- ◆ 文件关闭:

```
handle.close()
```

文本文件打开方式

模式	说明
" r "	打开文件以 读取 文件
" w "	打开文件以 写入 文件
" a "	打开文件以 添加 数据
" r+ "	打开文件以 读取和写入 数据
" w+ "	打开文件以 写入和读取 数据
" a+ "	打开文件以 添加和读取 数据

文件读操作一

handle.read(n) : 从文件读取n个字符到字符串

例: test=handle.read(10)

handle.read(): 读取整个文件到字符串中

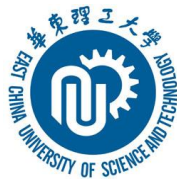
例: test=handle.read()

文章单词替换问题

```
fobj=open("XX财富网简介.txt","r",encoding="utf-8")
paper=fobj.read()
fobj.close()
before=input("被替代字: ")
after=input("替代为字: ")
result=paper.replace(before,after)
print(result)
```

文章单词替换问题

```
with open("XX财富网简介.txt","r",encoding="utf-8") as fobj:  
    paper=fobj.read()  
    before=input("被替代字: ")  
    after=input("替代为字: ")  
    result=paper.replace(before,after)  
    print(result)
```



谢 谢