

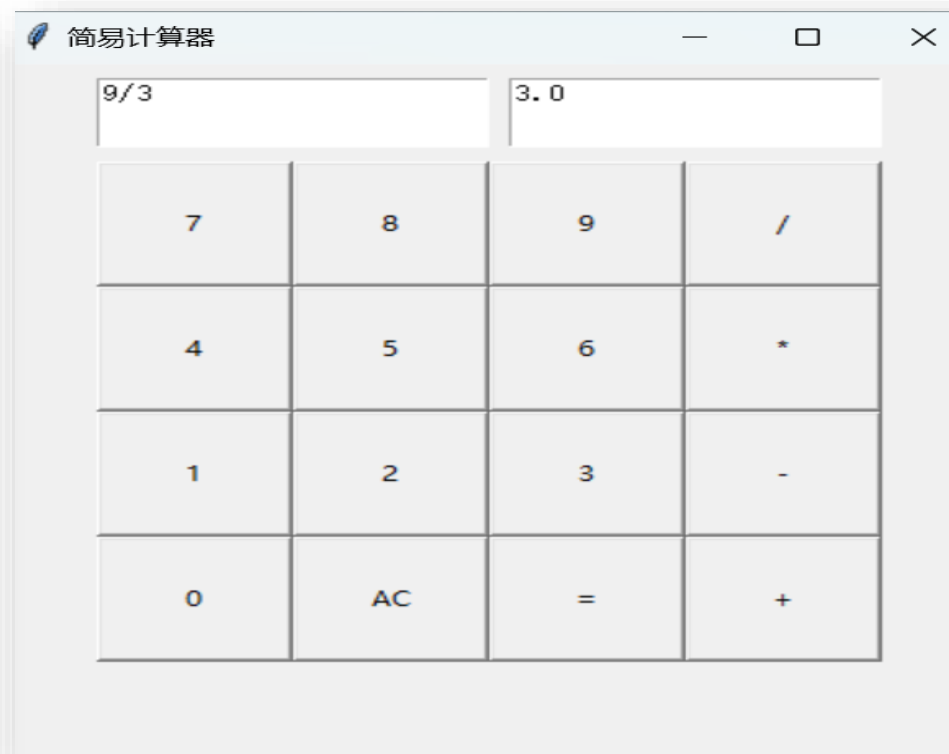


Python与金融数据挖掘(2)

文欣秀

wenxinxiu@ecust.edu.cn

计算器问题



程序设计方法



图灵奖获得者沃思

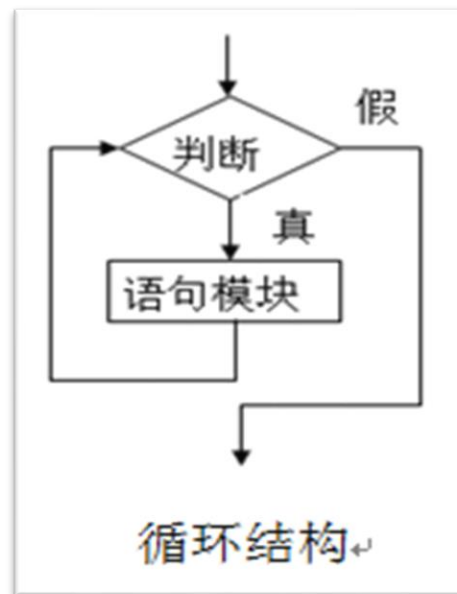
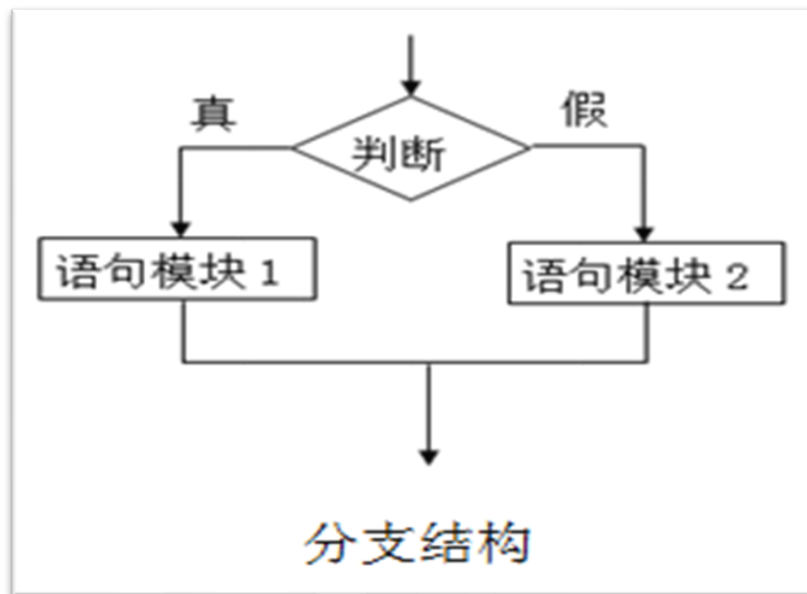
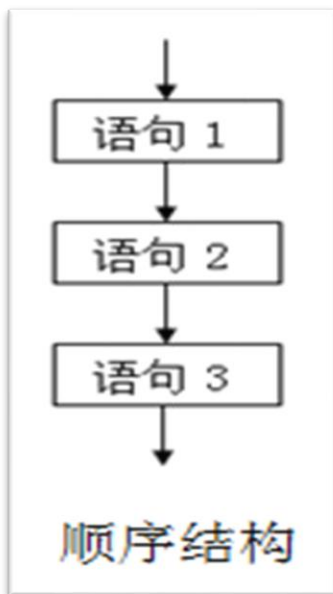
算法+数据结构=程序

Algorithm+Data Structures=Programs

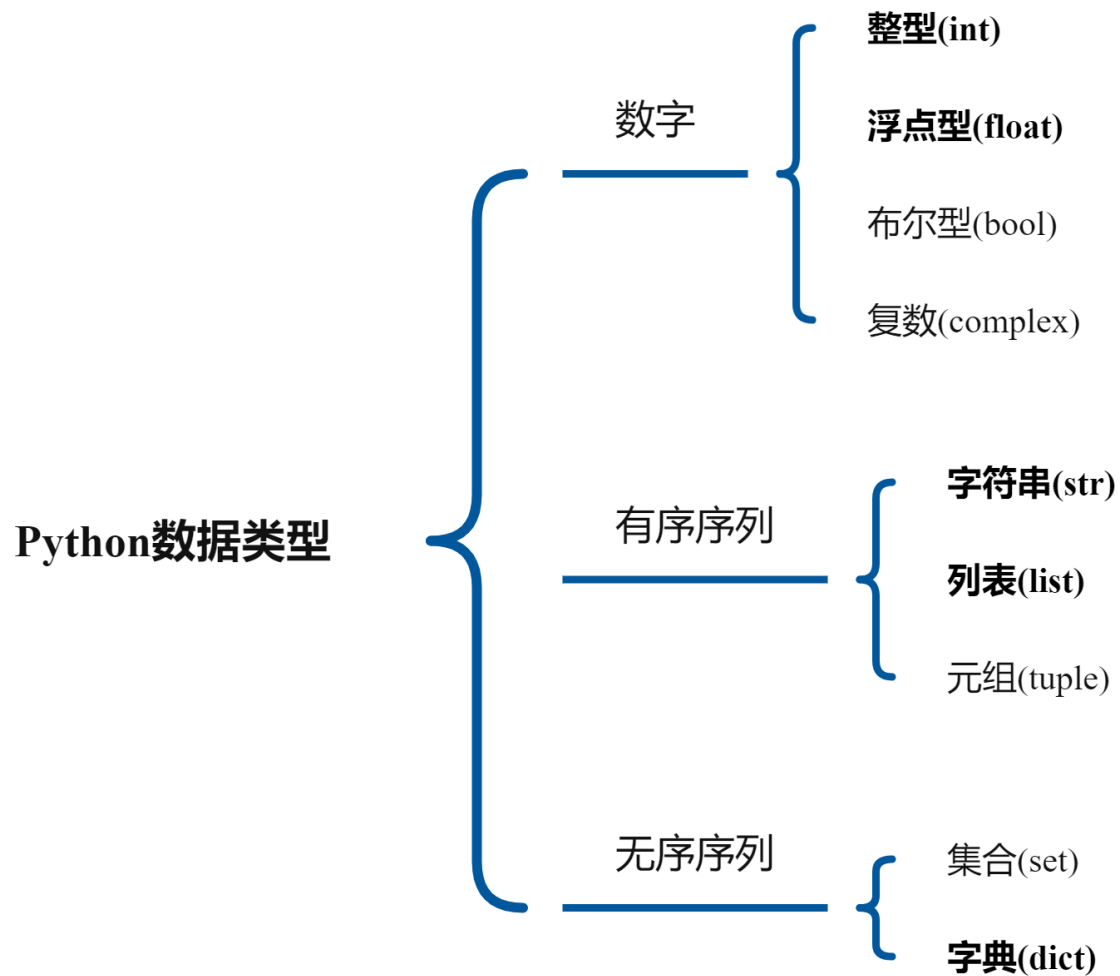
算法：对操作的描述，即要求计算机进行操作的步骤。

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



Python基本数据类型

int (整数) 12 0o12 0x12

float (浮点值) 3.2 3.5e-2 -1.5E3

bool (布尔值) True False

complex (复数) 3.0+2j 4.8-3.5J

注意： Python语言**没有char**类型

Python基本数据类型

type(object) : 返回object的数据类型

isinstance(object, class): 测试对象object是否为指定类型class的实例

```
>>> type(1)
<class 'int'>

>>> type('1')
<class 'str'>
```

```
>>> isinstance(1,int)
True

>>> isinstance(1,str)
False
```

变 量

变量： 指程序运行过程中值会改变的量，命名要遵循一定规则

例： `age=3`

注意： Python的变量实质是引用，使用`del` 命令可以删除一个变量

例： `del age`

变量引用

id(object): 返回object的地址

is运算符: 可以判断两个变量是否引用同一个对象

```
>>> age=3
>>> id(age)
>>> age=age+3
>>> id(age)
```

```
>>> year=6
>>> age is year
>>> year=5
>>> age is year
```

简单输入示例

input(): 接收从键盘输入的字符串

```
>>> myAge = int(input("请输入年龄: "))
```

int(): 将字符串转换为整数

```
>>> myScore = float(input("请输入成绩: "))
```

float(): 将字符串转换为单精度数

简单输出示例

print(): 输出程序运行结果

```
>>> course = "Python与金融数据挖掘"
```

```
>>> print(course *3)
```

```
>>> price = 12.56
```

```
>>> print(price *2)
```

计算器答案

```
num1=float(input("请输入第一个数字: "))
```

```
num2=float(input("请输入第二个数字: "))
```

```
r_sum=num1+num2
```

```
r_difference=num1-num2
```

```
r_product=num1*num2
```

```
r_quotient=num1/num2
```

```
print("两数之和为: ", r_sum)
```

```
print("两数之差为: ", r_difference)
```

```
print("两数之积为: ", r_product)
```

```
print("两数之商为: ", r_quotient)
```

请输入第一个数字: 89

请输入第二个数字: 75

两数之和为: 164.0

两数之差为: 14.0

两数之积为: 6675.0

两数之商为: 1.18666666666666666668

格式输出示例

```
>>> myName = input("请输入姓名: ")
>>> myAge, myScore = 24, 35.5
>>> print ("我的成绩是%f" % myScore)
>>> print ("%s今年%d岁" % (myName, myAge))
```

%s 以字符串格式输出

%d 以整型格式输出

%f 以浮点型格式输出

计算器问题代码

```
num1=float(input("请输入第一个数字: "))
num2=float(input("请输入第二个数字: "))
r_sum=num1+num2
r_difference=num1-num2
r_product=num1*num2
r_quotient=num1/num2
print("两数之和为: %.2f"% r_sum)
print("两数之差为: %.2f"% r_difference)
print("两数之积为: %.2f"% r_product)
print("两数之商为: %.2f"% r_quotient)
```

```
请输入第一个数字: 98
请输入第二个数字: 75
两数之和为: 173.00
两数之差为: 23.00
两数之积为: 7350.00
两数之商为: 1.31
```

format格式输出

- 无需关注数据类型
- 参数顺序可以不同
- 对齐方式非常强大
- 填充方式十分灵活
- 官方推荐使用方式

format使用方法

格式： <模板字符串>.format(<逗号分隔的参数>)

例如：

```
>>> print("{}今年{}岁".format("李斯",20))
```

```
>>> print("{1}今年{0}岁".format(20,"李斯"))
```

注意： 参数序号从**0**开始

计算器问题代码

```
num1=float(input("请输入第一个数字: "))
num2=float(input("请输入第二个数字: "))
r_sum=num1+num2
r_difference=num1-num2
r_product=num1*num2
r_quotient=num1/num2
print("两数之和为: {}".format(r_sum))
print("两数之差为: {}".format(r_difference))
print("两数之积为: {}".format(r_product))
print("两数之商为: {:.2f}".format(r_quotient))
```

```
请输入第一个数字: 89
请输入第二个数字: 75
两数之和为: 164.0
两数之差为: 14.0
两数之积为: 6675.0
两数之商为: 1.19
```

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("我爱ECUST"))
```

```
**我爱ECUST**
```

```
>>> print("{0:#<11}".format("我爱ECUST"))
```

```
我爱ECUST####
```

```
>>> print("{0:@>11}".format("我爱ECUST"))
```

```
@ @ @ @我爱ECUST
```

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}{1:.4s}".format("我校缩写是","华东理工大学"))
```

我校缩写是华东理工

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
```

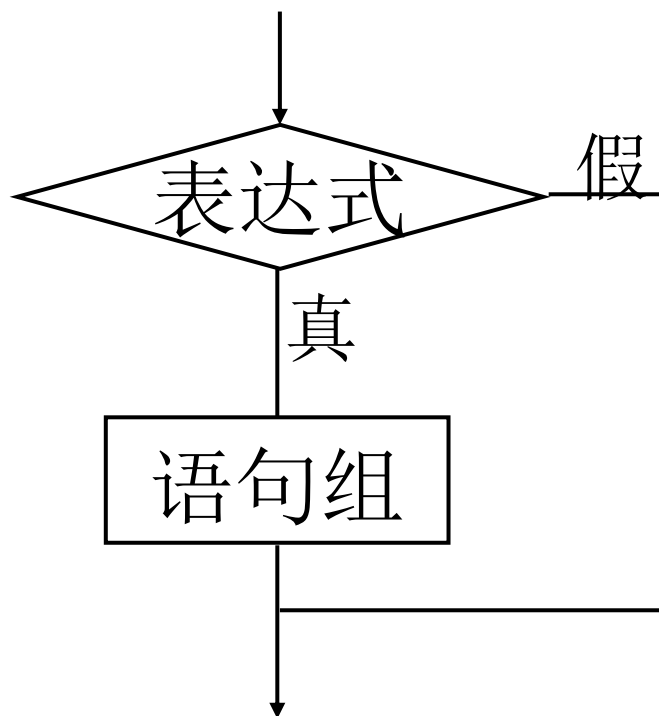
```
>>> not age<18
```

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

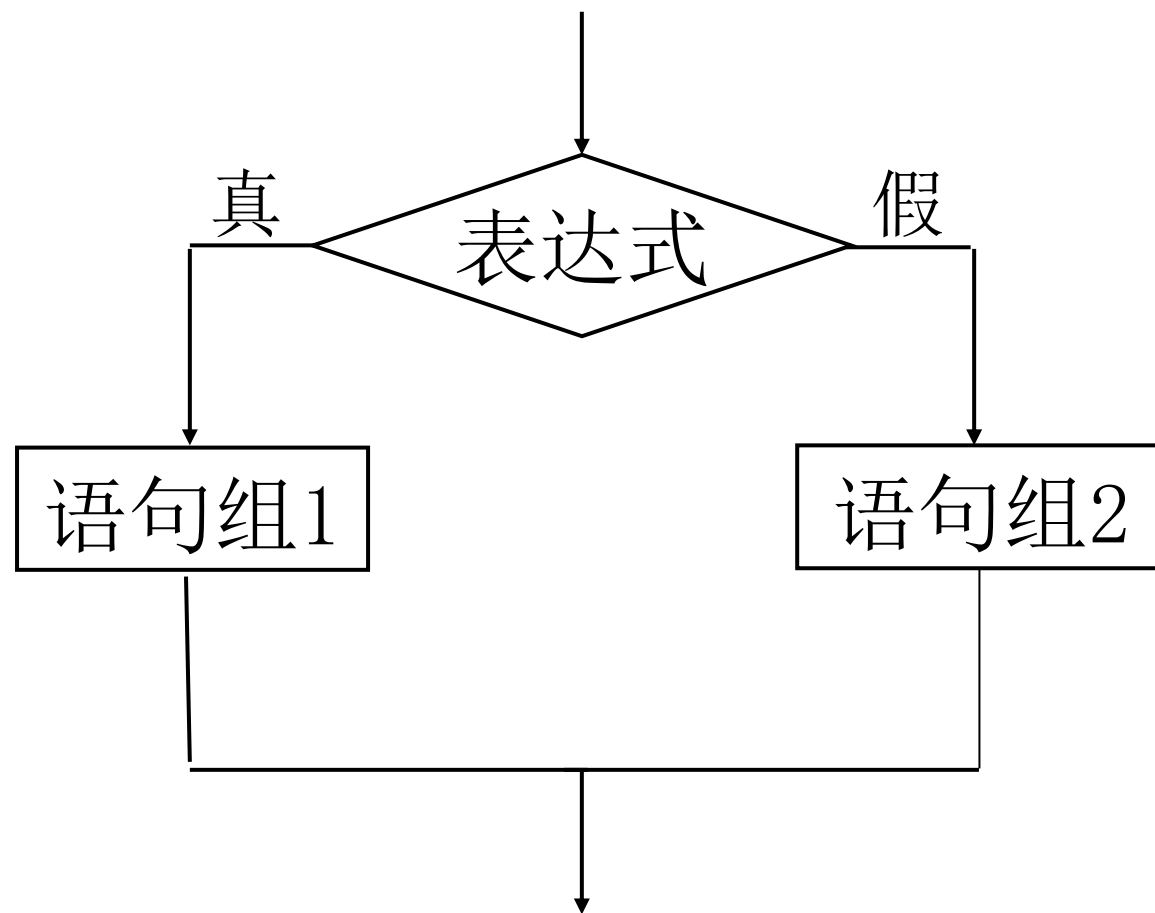


单分支条件语句

if 表达式:
语句组

例: `import random`
`num=random. randint(-5,5)`
`if num>0:`
 `print("%d is positive" % num)`

双分支结构



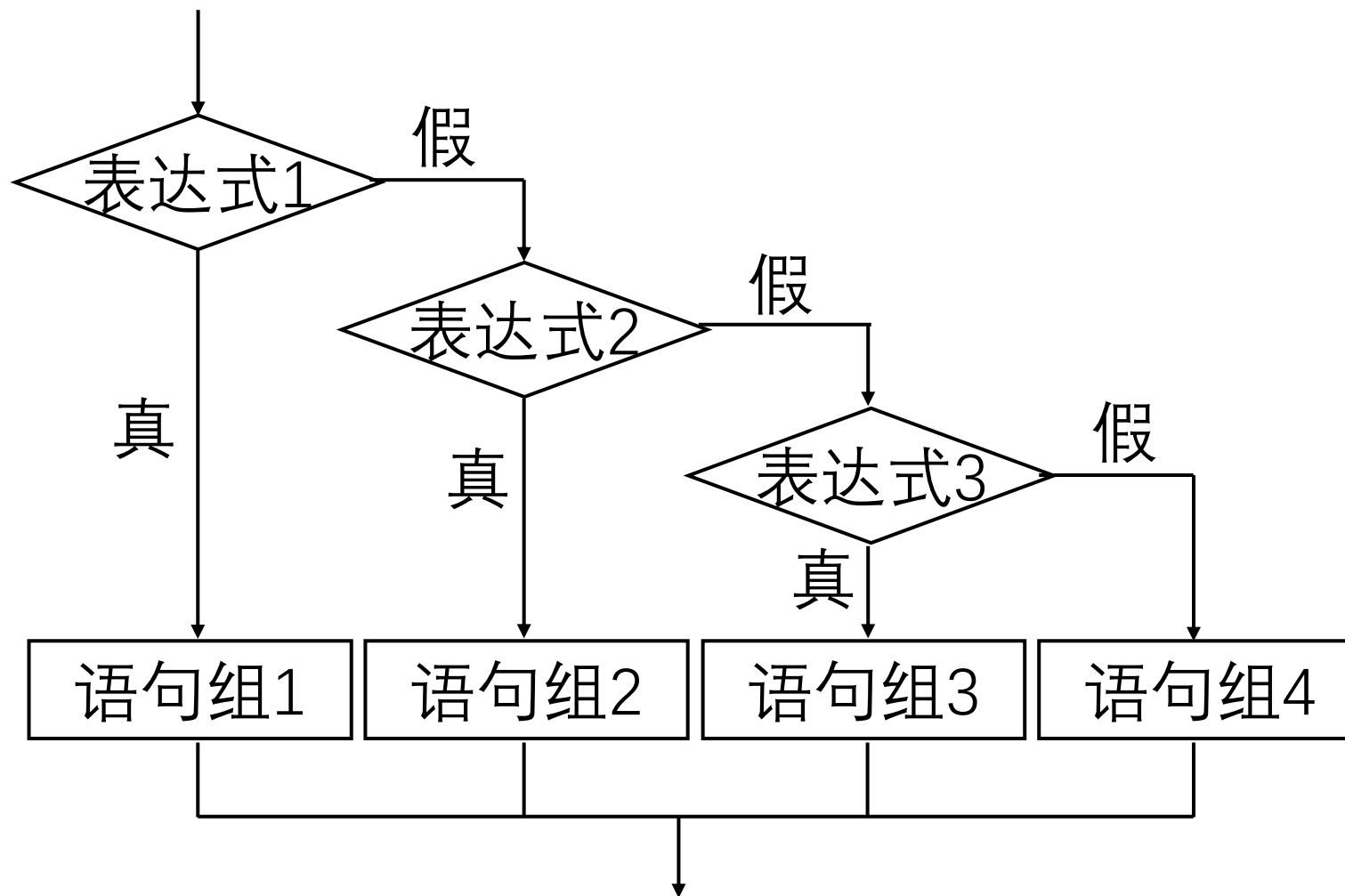
双分支条件语句

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

例：

```
import random
num=random. randint(5,10)
if num % 2==0:
    print("%d is even" % num)
else:
    print("%d is odd" % num)
```

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

例: import random
grade=random. randint(1,5)
if grade>=4:
 print("%d is good" % grade)
elif grade >=3:
 print("%d is ok" % grade)
else:
 print("%d is failed" % grade)

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("本月应缴纳党费{:.2f}元".format(money))
```


猜测随机数问题

编写程序，生成一个1到200之间的随机红包，然后给五次机会让家人猜测这个数字。程序给出猜测结果（太大、太小、成功），直到猜测成功中断循环或机会用完结束循环。

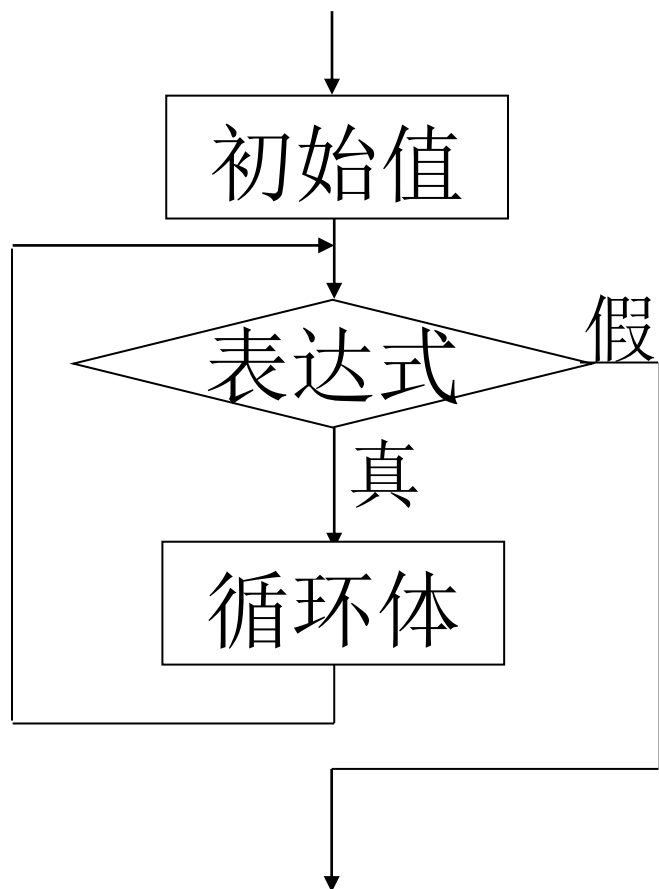
```
请输入您的猜测：80  
您猜的太小  
请输入您的猜测：120  
您猜的太大  
请输入您的猜测：100  
您猜的太小  
请输入您的猜测：110  
您猜的太小  
请输入您的猜测：118  
您猜的太大  
红包金额为115
```

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(1,200)
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=",")
```

思考

编写程序，解决以下问题：四人中有一个人做了好事，有下面一段对话，已知有三人说了真话，判断是谁做的好事：

A: 不是我;

B: 是C;

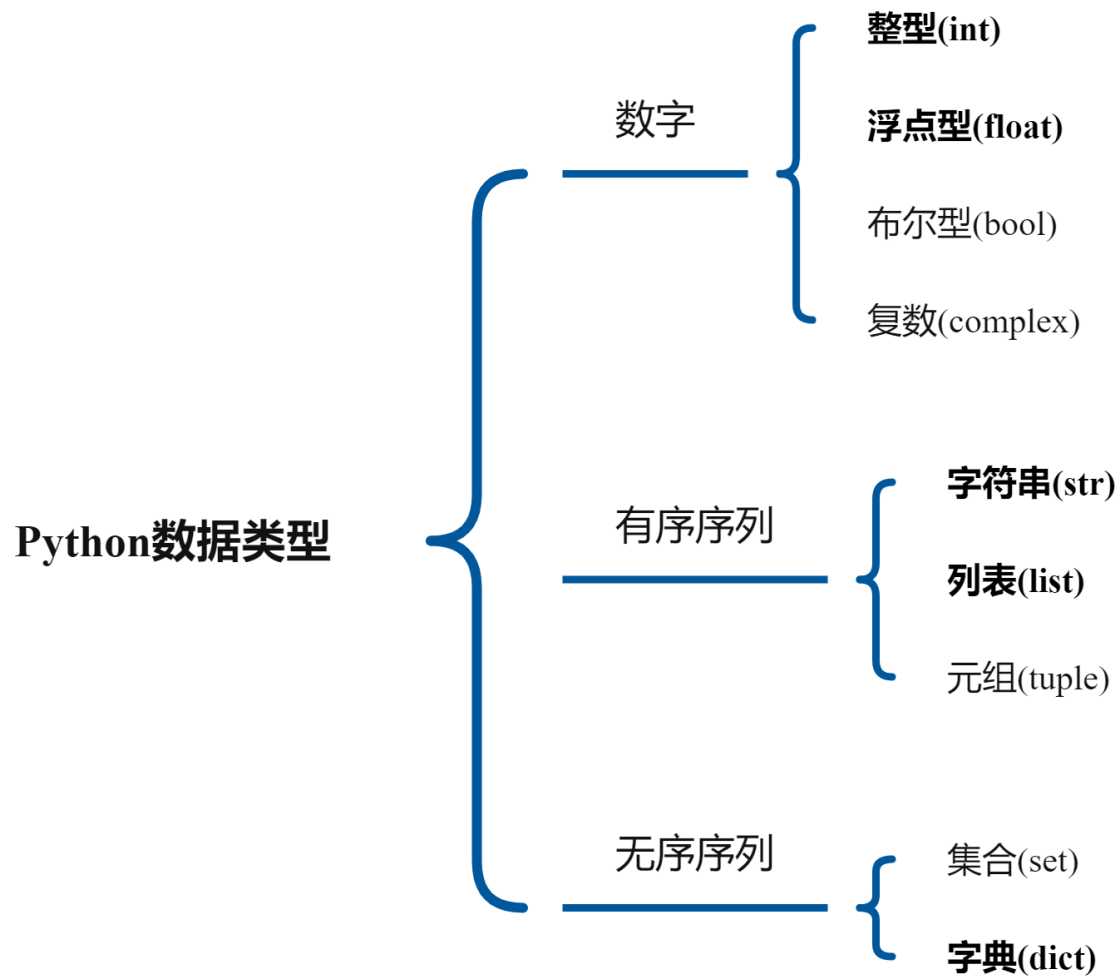
C: 是D;

D: 他胡说。

好人好事问题

```
for p in ['A','B','C','D']:
    if (p!='A')+(p=='C')+(p=='D')+(p!='D')==3:
        print(p)
```

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

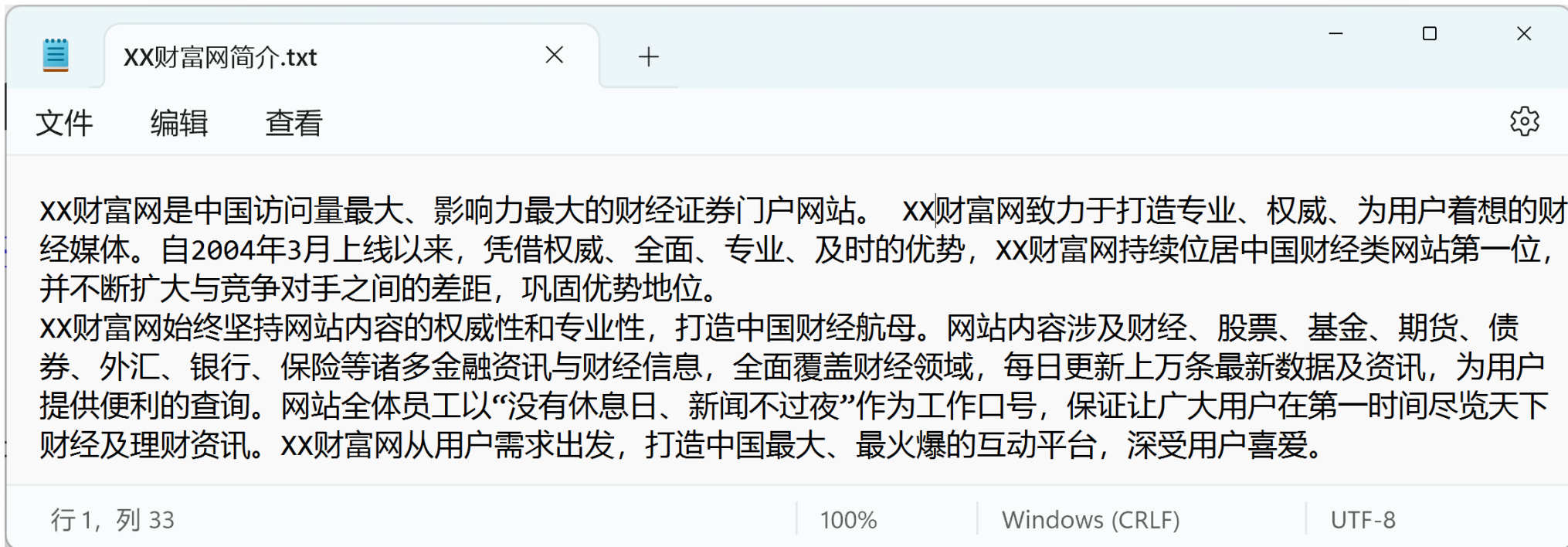
s. strip(): 消除字符串两端的空格及符号

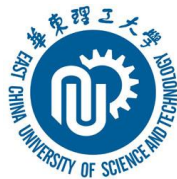
单词/词语替换问题答案

```
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。





谢 谢