



對外經濟貿易大學

University of International Business and Economics

信息学院

School of Information Technology & Management

第一章 Python基础

任课教师：袁石

办公地点：求索楼1006

电子邮箱：ystone1025@uibe.edu.cn



- 1、Python语言基础
- 2、运算符
- 3、程序结构
- 4、选择语句
- 5、循环语句

流程控制语句





1、Python语言基础



注释规则

注释是指在程序代码中添加的标注性的文字

1 单行注释

注释内容

2 多行注释

```
...  
注释内容1  
注释内容2  
.....  
...
```

```
.....  
注释内容1  
注释内容2  
.....  
.....
```

- 三个单引号 (''.....') 或者三个双引号 ("\".....\"") 之间，并且不属于任何语句的内容

3 中文编码声明注释

```
#-*-coding:编码-*-
```

```
#!/usr/bin/python  
#-*-coding=utf-8-*-  
#vision2  
import os  
import re
```

- Python 2.x不支持直接写中文：中文编码声明规定中文编码格式
- Python 3.x支持直接写中文：编码规范，建议添加中文编码声明



代码缩进

- **代码缩进**是指在每一行代码左端空出一定长度的**空白**，从而可以更加清晰地从外观上看出程序的逻辑结构
- 缩进可以使用**空格**或者**Tab键**实现：
 - ✓ 空格：4个空格作为一个缩进量
 - ✓ Tab键：一个Tab键作为一个缩进量

```
from parameter import DOMAIN_ABS_PATH as abs_path

def readProtoUser():
    f = open(abs_path+"/protou_combine/protou.txt", "r")
    protou = dict()
    for line in f:
        area=line.split(":")[0]
        if area not in protou:
            protou[area]=set()
        for u in (line.split(":")[1]).split():
            protou[area].add(str(u))

    return protou

proto_users = readProtoUser()
```



编码规则

- ✓ 每个import语句只导入一个模块，尽量避免一次导入多个模块
- ✓ 不要在行尾添加分号 “;” ， 也不要加分号将两条命令放在同一行
- ✓ 建议每行不超过80个字符
- ✓ 使用必要的空行可以增加代码的可读性
- ✓ 运算符两侧、函数参数之间、逗号 “,” 两侧建议使用空格进行分隔
- ✓ 避免在循环中使用+和+=运算符累加字符串
- ✓ 适当使用异常处理结构提高程序容错性



保留字与标识符

保留字是Python语言中已经被赋予特定意义的一些单词，开发程序时，不可以作为变量、函数、类、模块和其他对象的名称来使用

and	as	assert	break	class	continue
def	del	elif	else	except	finally
for	from	False	global	if	import
in	is	lambda	nonlocal	not	None
or	pass	raise	return	try	True
while	with	yield			



保留字与标识符

标识符可以简单理解为一个名字，主要用来标识变量、函数、类、模块和其他对象的名称

标识符命名规则：

- 由字母、下划线 “_” 和数字组成，且第一个字符不能是数字
- 不能使用Python中的保留字
- 区分字母大小写
- Python中以下划线开头的标识符具有特殊意义，一般应避免使用类似的标识符
 - ✓ 以单下划线开头的标识符（如_width）表示不能直接访问的类属性，也不能通过 `from *** import ***` 导入
 - ✓ 以双下划线开头的标识符（如__add）表示类的私有成员
 - ✓ 以双下划线开头和结尾的标识符是Python专用的标识，如__init()__表示构造函数



1/2 基本数据类型



数字类型

1 整数

- **十进制整数**：常用的表现形式
 - ✓ 由0-9组成，“逢十进一”
 - ✓ 不能以0作为十进制数的开头
- **八进制整数**：
 - ✓ 由0-7组成，“逢八进一”
 - ✓ 以0o或0O开头
- **十六进制整数**：
 - ✓ 由0-9、A-F组成，“逢十六进一”
 - ✓ 以0x或0X开头
- **二进制整数**：
 - ✓ 只有0和1两个数字，“逢二进一”
 - ✓ 以0b或0B开头

2 浮点数

- 包含整数部分和小数部分，如1.14、-1.732
- 可以用科学计数法表示，如2.7e2、-3.14e5

3 复数

- 包含实部和虚部
- 使用j或者J表示虚部，如3.14+12.5j

```
height = 1.70      # 保存身高的变量，单位：米
print("您的身高：" + str(height))
weight = 48.5      # 保存体重的变量，单位：千克
print("您的体重：" + str(weight))
bmi=weight/(height*height)      # 用于计算BMI指数，公式为“体重/身高的平方”
print("您的BMI指数为：" + str(bmi))      # 输出BMI指数
# 判断身材是否合理
if bmi<18.5:
    print("您的体重过轻 ~@_@~")
if bmi>=18.5 and bmi<24.9:
    print("正常范围，注意保持 (-_-)")
if bmi>=24.9 and bmi<29.9:
    print("您的体重过重 ~@_@~")
if bmi>=29.9:
    print("肥胖 ~@_@~")
```

1/2 基本数据类型

字符串类型

字符串：连续的字符序列，可以是计算机所能表示的一切字符的集合

- 字符串可用**单引号**、**双引号**、**三引号**来表示
- 字符串开始和结尾使用的引号形式必须一致
- 表示复杂的字符串时，可以进行引号嵌套

#字符串引号嵌套

```
str1 = '践行“社会主义核心价值观”' #单引号嵌套双引号
str2 = "'四个自信': 道路自信、理论自信、制度自信、文化自信" #双引号嵌套单引号
```

#输出实例

```
print(str1)
print(str2)
```

践行“社会主义核心价值观”

'四个自信': 道路自信、理论自信、制度自信、文化自信

#字符串实例

```
str1 = '我爱中国，我爱贸大' #单引号，不可跨行
str2 = "我爱中国，我爱贸大" #双引号，不可跨行
str3 = '''我爱中国，我爱贸大。
祝愿中国发展越来越好。''' #三引号，可以跨行
```

#输出实例

```
print('单引号字符串：')
print(str1)
print('=====')
print('双引号字符串：')
print(str2)
print('=====')
print('三引号字符串：')
print(str3)
```

单引号字符串：

我爱中国，我爱贸大

双引号字符串：

我爱中国，我爱贸大

三引号字符串：

我爱中国，我爱贸大。

祝愿中国发展越来越好。



字符串类型

转义字符：指使用“\”对一些特殊字符进行转义

转义字符	说 明
\	续行符
\n	换行符
\0	空
\t	水平制表符，用于横向跳到下一制表位
\"	双引号
\'	单引号
\\	一个反斜杠
\f	换页
\odd	八进制数，dd代表的字符，如\012代表换行
\xhh	十六进制数，hh代表的字符，如\x0a代表换行



布尔类型

布尔类型：表示真或假的值

- ✓ 真：True，数值表示为1
- ✓ 假：False，数值表示为0

下面几种情况得到的值为假，其他对象在if或者while语句中都表现为真：

- False或None
- 数值中的零，包括0、0.0、虚数0
- 空序列，包括字符串、空元组、空列表、空字典
- 自定义对象的实例，该对象的`__bool__`方法返回False或者`__len__`方法返回0



数据类型转换

Python是动态类型的语言（也称为弱类型语言），不需要像Java或C语言一样在使用变量前必须先声明变量的类型

int()	→	整型
float()	→	浮点型
str()	→	字符串类型
hex()	→	整数转换为十六进制字符串
oct()	→	整数转换为八进制字符串



输入

`variable = input("提示文字")`

- ✓ Variable表示保存输入结果的变量，input双引号内的文字是用于提示要输入的内容
- ✓ Python 3.x中，**输入的数字或字符串都将被作为字符串读取**，若想输入数字，需要将输入的字符串进行类型转换

输出

`print(输出内容)`

- ✓ 数字和字符串：直接输出
- ✓ 包含运算符的表达式：输出结算结果
- ✓ 默认情况下，print语句输出后会自动换行

```
#输入和输出测试
variable = input('请输入文字：')

print(variable)
print(type(variable)) #输出variable的数据类型

a = 2
b = 3
print(a) #直接输出数字
print(a+b) #包含运算符的表达式，输出计算结果
```

```
请输入文字：333
333
<class 'str'>
2
5
```





2、运算符



2/1 运算符的类型



2/1 运算符的类型

算术运算符

处理四则运算的符号

运算符	描述符	例子
+	加法 - 对操作符的两侧增加值	$a + b = 30$
-	减法 - 减去从左侧操作数右侧操作数	$a - b = -10$
*	乘法 - 相乘的运算符两侧的值	$a * b = 200$
/	除 - 由右侧操作数除以左侧操作数	$b / a = 2$
%	模 - 由右侧操作数和余返回除以左侧操作数	$b \% a = 0$
**	指数- 执行对操作指数（幂）的计算	$a^{**}b = 10 \text{ 的幂 } 20$
//	取整除 - 操作数的除法，其中结果是将小数点后的位数被除去的商。	$9//2 = 4$ 而 $9.0//2.0 = 4.0$



2/1 运算符的类型

赋值运算符

为变量等赋值

运算符	描述	示例
=	简单的赋值运算符，赋值从右侧操作数左侧操作数	$c = a + b$ 将指定的值 $a + b$ 到 c
+=	加法AND赋值操作符，它增加了右操作数左操作数和结果赋给左操作数	$c += a$ 相当于 $c = c + a$
-=	减AND赋值操作符，它减去右边的操作数从左边操作数，并将结果赋给左操作数	$c -= a$ 相当于 $c = c - a$
*=	乘法AND赋值操作符，它乘以右边的操作数与左操作数，并将结果赋给左操作数	$c *= a$ 相当于 $c = c * a$
/=	除法AND赋值操作符，它把左操作数与正确的操作数，并将结果赋给左操作数	$c /= a$ 相当于 $c = c / a$
%=	模量AND赋值操作符，它需要使用两个操作数的模量和分配结果左操作数	$c \% = a$ 相当于 $c = c \% a$
**=	指数AND赋值运算符，执行指数（功率）计算操作符和赋值给左操作数	$c ** = a$ 相当于 $c = c ** a$
//=	地板除，并分配一个值，执行地板除对操作和赋值给左操作数	$c //= a$ 相当于 $c = c // a$



2/1 运算符的类型

比较运算符

也称关系运算符，用于对变量或者表达式的结果进行大小、真假等比较

运算符	描述	示例
==	检查两个操作数的值是否相等，如果是则条件变为真。	(a == b) 不为 true.
!=	检查两个操作数的值是否相等，如果值不相等，则条件变为真。	(a != b) 为 true.
>	检查左操作数的值是否大于右操作数的值，如果是，则条件成立。	(a > b) 不为 true.
<	检查左操作数的值是否小于右操作数的值，如果是，则条件成立。	(a < b) 为 true.
>=	检查左操作数的值是否大于或等于右操作数的值，如果是，则条件成立。	(a >= b) 不为 true.
<=	检查左操作数的值是否小于或等于右操作数的值，如果是，则条件成立。	(a <= b) 为 true.



2/1 运算符的类型

逻辑运算符

对真和假两种布尔值进行运算，运算后的结果仍是一个布尔值

运算符	描述	示例
and	逻辑与 运算符。如果两个操作数都是真的，那么则条件成立。	a and b
or	逻辑或 运算符。如果有两个操作数都是非零然后再条件变为真。	a or b
not	逻辑非 运算符。用于反转操作数的逻辑状态。如果一个条件为真，则逻辑非运算符将返回false。	not(a and b)



2/1 运算符的类型



位运算符

- 把数字看作二进制数来进行计算
- 需要先将执行运算的数据转换为二进制，再执行运算

a = 00001100 b=10001000

运算符	描述	示例
&	“按位与” 运算：两个操作数据的二进制表示，只有对应位都为1时，结果位为1，否则为0	a&b=00001000
	“按位或” 运算：两个操作数据的二进制表示，只有对应位都为0时，结果位为0，否则为1	a b=10001100
^	“按位异或” 运算：当两个操作数据的二进制表示相同（同为1或同为0）时，结果位0，否则为1	a^b=10000100
~	“按位取反” 或 “按位非” 运算：将操作数对应二进制中的1修改为0，0修改为1	~a=11110011
<<	“左移位” 运算：将一个二进制操作数向左移动指定的位数，左边（高位端）溢出的位将被丢弃，右边（低位端）的空位用0补充	a<<1=00011000
>>	“右移位” 运算：将一个二进制操作数向右移动指定的位数，右边（低位端）溢出的位将被丢弃，左边（高位端）的空位，若最高位是0，则填入0；若最高位是1，则填入1	a>>1=00000110



成员运算符 判断右操作数是否包含左操作数

```
a = '1'  
b = ['1', '2', '3', '4']
```

运算符	描述	示例
in	如果左操作数 在 右操作数中，结果为True，否则为False	a in b = True
not in	如果左操作数 不在 右操作数中，结果为True，否则为False	a not in b = False

标识运算符

运算符	描述
is	如果操作符两侧的变量指向相同的对象，结果为True，否则为False
is not	如果操作符两侧的变量操作符指向相同的对象，结果为False，否则为True



2/2 运算符优先级

运算符	描述
**	幂
~ + -	取反、正号、负号
* / % //	乘、除、取模、取整除
+ -	加法和减法
>> <<	按位左移、右移
&	按位与
^	按位异或
	按位或
<= < > >= == !=	比较运算符
= %= /= //= -= += *= **=	赋值运算符
is、is not	标识运算符
in、not in	成员运算符
not、or、and	逻辑运算符

优先级从高到底



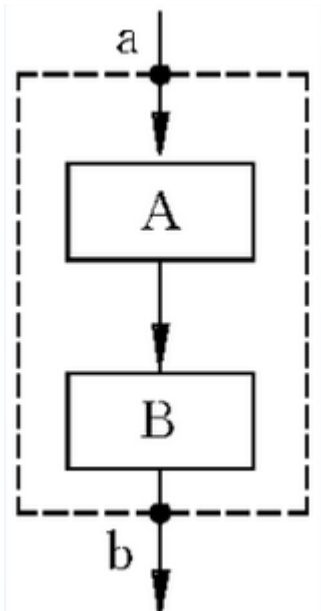


3、程序结构

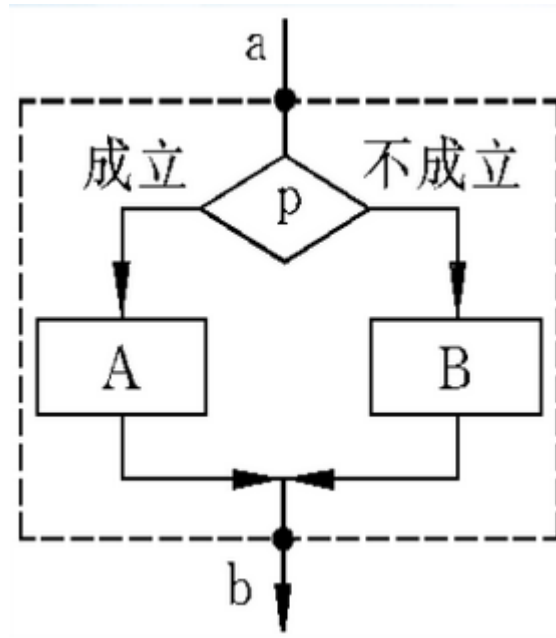


3/1 程序结构

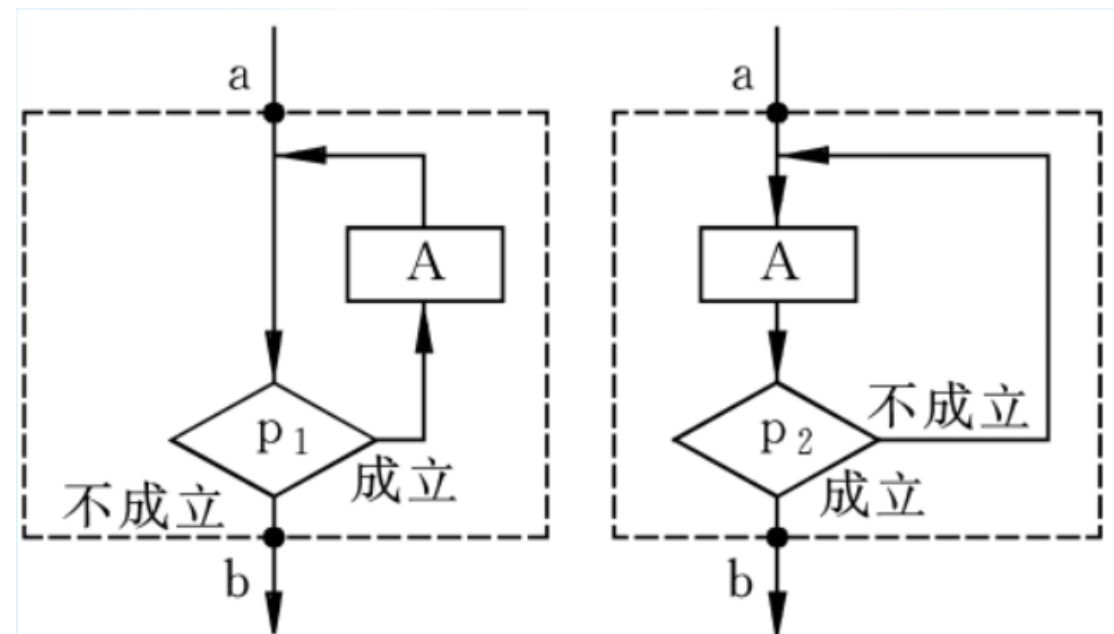
顺序结构



选择结构



循环结构





4、选择语句



4/1 选择与生活



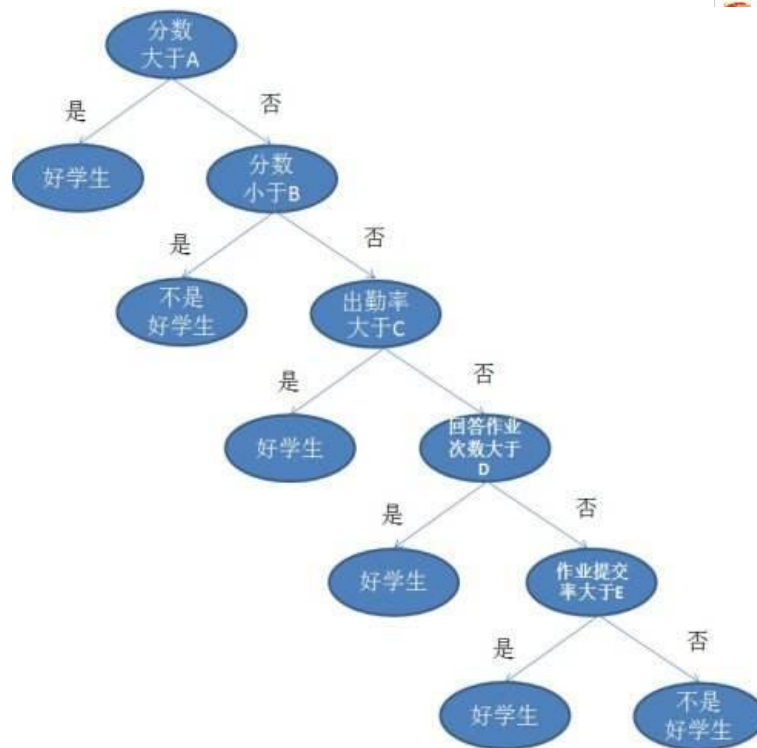
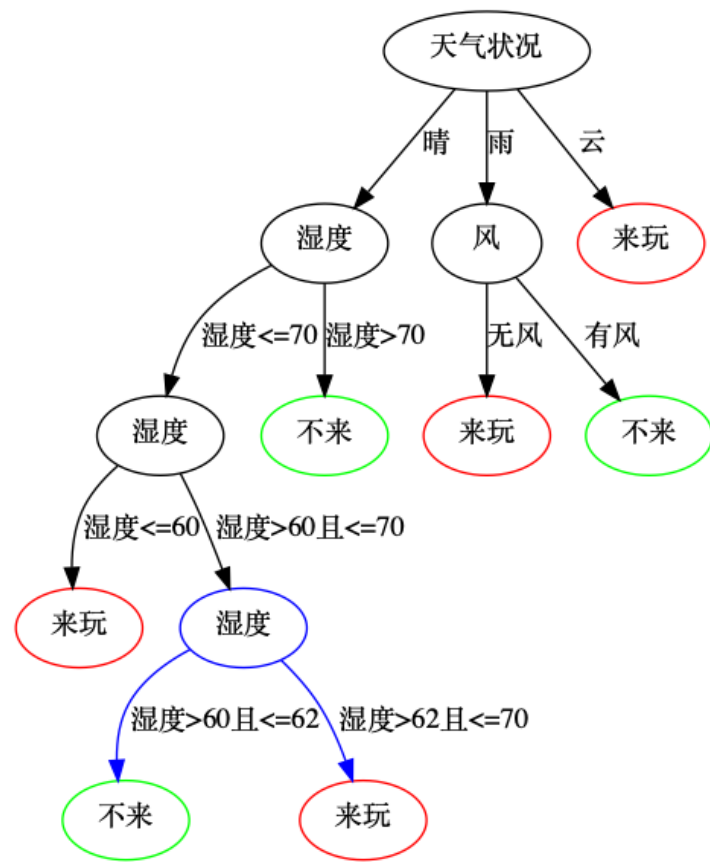
输入账号
和密码

正确

登录成功

错误

登录失败

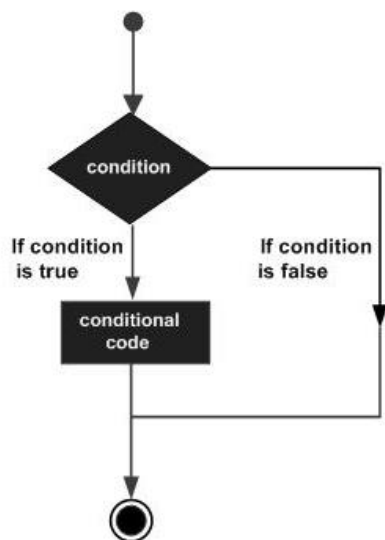


生活中我们一直在做选择：

- ✓ 今天出门穿运动装还是休闲服？
- ✓ 今天午餐吃辣子鸡丁还是西红柿炒鸡蛋？
- ✓ “Python与大数据分析” 是选A老师还是B老师的课？
- ✓ ...



4/2 选择语句



最简单的if

If 表达式:
语句块

if + elif + else

```
if impulse >= COUNT_STA and impulse > depressed:
    flag = 1
elif depressed >= COUNT_STA and impulse < depressed:
    flag = 2
elif impulse >= COUNT_STA and depressed >= COUNT_STA and impulse == depressed:
    flag = 3
else:
    flag = 0
```

if + else

```
if s_dict.has_key(t0):
    r0 = s_dict[t0]
else:
    r0 = [0,0,0]
```

if + 内嵌if

```
if r0[1] == 0:
    if r1[1] > ZERO_STA:
        impulse_flag = impulse_flag + 1
else:
    if abs(r1[1]-r0[1])/float(r0[1]) >= RATE_STA:
        impulse_flag = impulse_flag + 1
```

Python中没有switch语句



4/3 选择语句的例子

例1：中国南北朝时期（公元5世纪）的数学著作《孙子算经》的“物不知数”问题：
今有物不知其数，三三数之剩二，五五数之剩三、七七数之剩二，问几何？

枚举法：想到哪个数字则输入哪个数字

```
#选择语句：例1
print("今有物不知其数，三三数之剩二，五五数之剩三，七七数之剩二，问几何？\n")
number = int(input("请输入您认为符合条件的数：")) # 输入一个数
if number % 3 ==2 and number % 5 ==3 and number % 7 ==2:
    print(number, "符合条件：三三数之剩二，五五数之剩三，七七数之剩二")
else:
    print(number, "不符合条件")
```

今有物不知其数，三三数之剩二，五五数之剩三，七七数之剩二，问几何？

请输入您认为符合条件的数：5
5 不符合条件

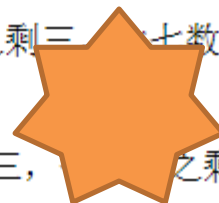
今有物不知其数，三三数之剩二，五五数之剩三，七七数之剩二，问几何？

请输入您认为符合条件的数：10
10 不符合条件



今有物不知其数，三三数之剩二，五五数之剩三，七七数之剩二，问几何？

请输入您认为符合条件的数：23
23 符合条件：三三数之剩二，五五数之剩三，七七数之剩二



4/3 选择语句的例子

例2：闰年判断问题：

输入一个年份，判定该年份是不是闰年？

四年一闰、百年不闰、四百年再闰

#选择语句：例2 闰年判定

```
year = int(input("请输入一个年份：")) # 获取用户输入的年份，并转换为整型
if year % 4 == 0:                        # 四年一闰
    if year % 100 == 0:
        if year % 400 == 0:            # 四百年再闰
            print(year, "年是闰年")
        else:                          # 百年不闰
            print(year, "年不是闰年")
    else:
        print(year, "年是闰年")
else:
    print(year, "年不是闰年")
```



让计算机自动枚举，以减少人为参与的工作量

2016-2020哪几年是闰年？？？

请输入一个年份：2016
2016 年是闰年

请输入一个年份：2017
2017 年不是闰年

请输入一个年份：2018
2018 年不是闰年

请输入一个年份：2019
2019 年不是闰年

请输入一个年份：2020
2020 年是闰年

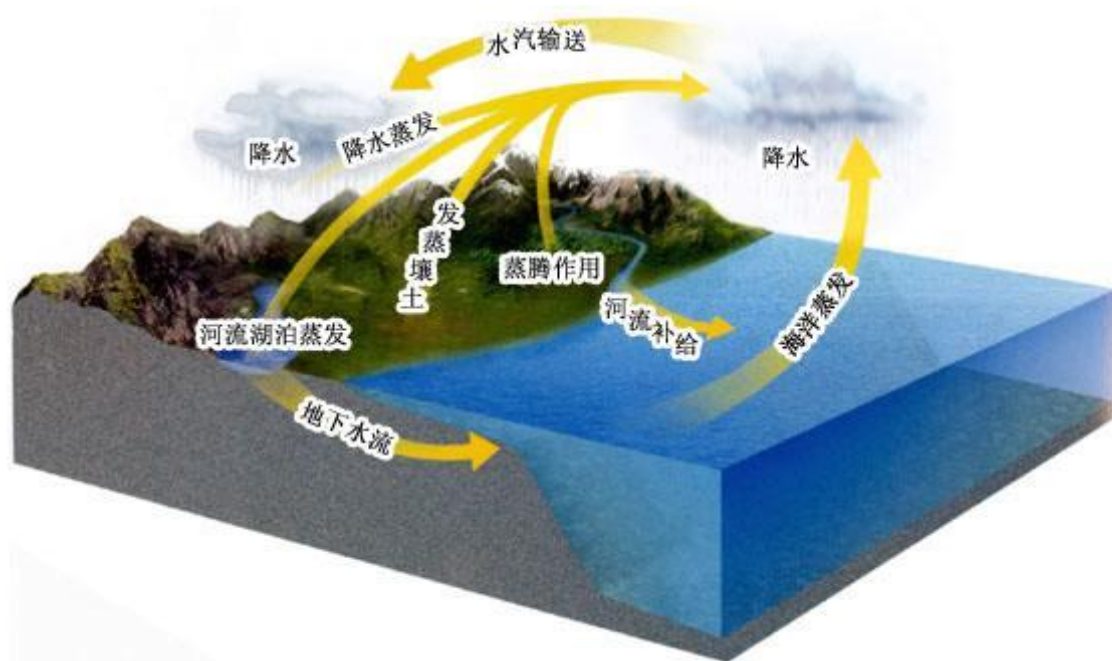




5、循环语句



垃圾回收



公共交通



- **循环：**反复做同一件事情
- **循环的类型：**
 - ✓ 计次循环：重复一定次数的循环，如操场跑步
 - ✓ 条件循环：一直重复，直到条件不满足时才结束



5/2 循环语句

while

```
while expression:  
    statement(s)
```

```
count = 0  
while (count < 9):  
    print 'The count is:', count  
    count = count + 1
```

for

```
for iterating_var in sequence:  
    statements(s)
```

```
for letter in 'Python': # First Example  
    print 'Current Letter :', letter  
  
fruits = ['banana', 'apple', 'mango']  
for fruit in fruits:    # Second Example  
    print 'Current fruit :', fruit
```

循环嵌套

While嵌套

```
while 条件表达式1:  
    while 条件表达式2:  
        循环体2  
    循环体1
```

For嵌套

```
for 迭代变量1 in 对象1:  
    for 迭代变量2 in 对象2:  
        循环体2  
    循环体1
```

混合嵌套

```
while 条件表达式1:  
    for 迭代变量2 in 对象2 :  
        循环体2  
    循环体1  
  
for 迭代变量1 in 对象1:  
    while 条件表达式2 :  
        循环体2  
    循环体1
```



5/3 循环语句的例子

例1：中国南北朝时期（公元5世纪）的数学著作《孙子算经》的“物不知数”问题：
今有物不知其数，三三数之剩二，五五数之剩三、七七数之剩二，问几何？

让计算机代替人工进行枚举

#循环语句：例1 物不知数问题

```
print("今有物不知其数，三三数之剩二，五五数之剩三，七七数之剩二，问几何？\n")
none = True # 作为循环条件的变量
number = 0 # 计数的变量
while none:
    number += 1 # 计数加1
    if number%3 ==2 and number%5 ==3 and number%7 ==2: # 判断是否符合条件
        print("答曰：这个数是", number) # 输出符合条件的数
        none = False # 将循环条件的变量赋值为否
```

今有物不知其数，三三数之剩二，五五数之剩三，七七数之剩二，问几何？

答曰：这个数是 23



5/3 循环语句的例子

例2：打印九九乘法表



九九乘法口诀表

1×1=1								
1×2=2	2×2=4							
1×3=3	2×3=6	3×3=9						
1×4=4	2×4=8	3×4=12	4×4=16					
1×5=5	2×5=10	3×5=15	4×5=20	5×5=25				
1×6=6	2×6=12	3×6=18	4×6=24	5×6=30	6×6=36			
1×7=7	2×7=14	3×7=21	4×7=28	5×7=35	6×7=42	7×7=49		
1×8=8	2×8=16	3×8=24	4×8=32	5×8=40	6×8=48	7×8=56	8×8=64	
1×9=9	2×9=18	3×9=27	4×9=36	5×9=45	6×9=54	7×9=63	8×9=72	9×9=81

#循环语句：例2 九九乘法表打印

```
for i in range(1,10): # 输出9行
    for j in range(1, i + 1): # 输出与行数相等的列
        print(str(j) + "×" + str(i) + "=" + str(i * j) + "\t", end = '')
    print('') # 换行
```

```
1×1=1
1×2=2  2×2=4
1×3=3  2×3=6  3×3=9
1×4=4  2×4=8  3×4=12  4×4=16
1×5=5  2×5=10  3×5=15  4×5=20  5×5=25
1×6=6  2×6=12  3×6=18  4×6=24  5×6=30  6×6=36
1×7=7  2×7=14  3×7=21  4×7=28  5×7=35  6×7=42  7×7=49
1×8=8  2×8=16  3×8=24  4×8=32  5×8=40  6×8=48  7×8=56  8×8=64
1×9=9  2×9=18  3×9=27  4×9=36  5×9=45  6×9=54  7×9=63  8×9=72  9×9=81
```



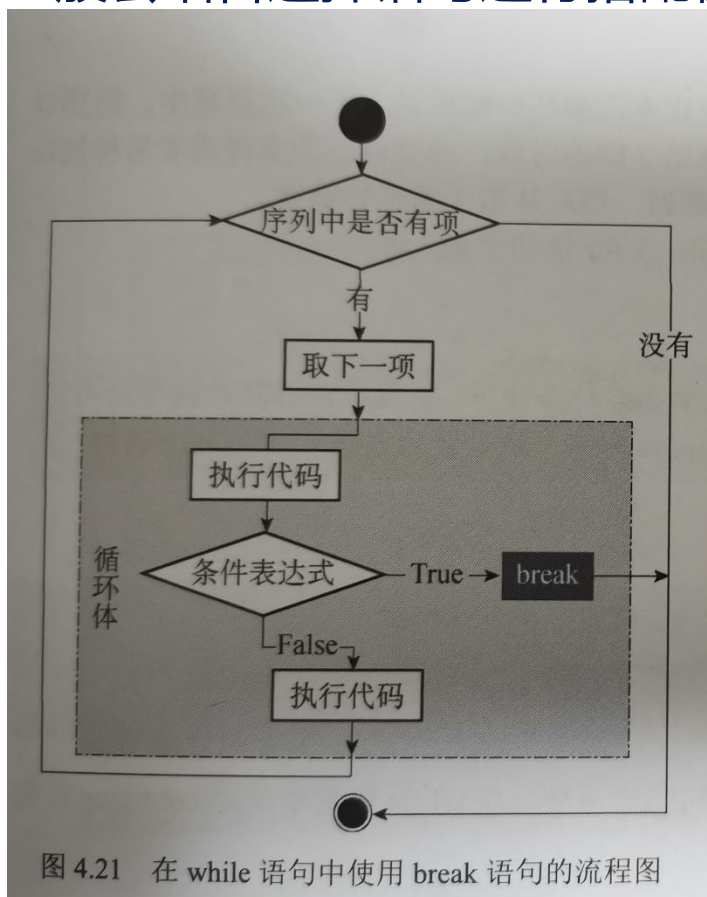
5/4 循环控制语句

break

- break语句可以终止当前的循环，包括while和for在内的所有控制语句
- break语句一般会结合选择语句进行搭配使用

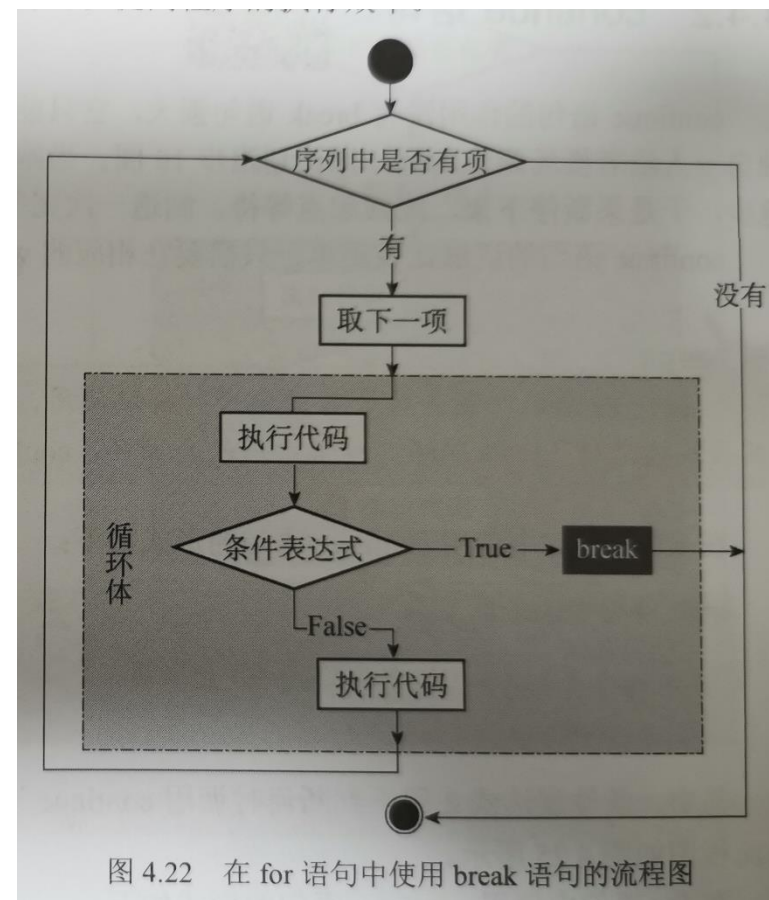
while+break

while 条件表达式1:
 执行代码
if 条件表达式2:
 break



for+break

for 迭代变量 in 对象:
 if 条件表达式:
 break



5/4 循环控制语句

例：中国南北朝时期（公元5世纪）的数学著作《孙子算经》的“物不知数”问题：
今有物不知其数，三三数之剩二，五五数之剩三、七七数之剩二，问几何？

让计算机代替人工进行枚举，并使用break语句控制循环

#循环语句+break: 例1 物不知数问题

```
print("今有物不知其数，三三数之剩二，五五数之剩三，七七数之剩二，问几何？\n")
none = True # 作为循环条件的变量
number = 0 # 计数的变量
while none:
    number += 1 # 计数加1
    if number%3 ==2 and number%5 ==3 and number%7 ==2: # 判断是否符合条件
        print("答曰：这个数是",number) # 输出符合条件的数
        break # 利用break语句终止循环
```

今有物不知其数，三三数之剩二，五五数之剩三，七七数之剩二，问几何？

答曰：这个数是 23



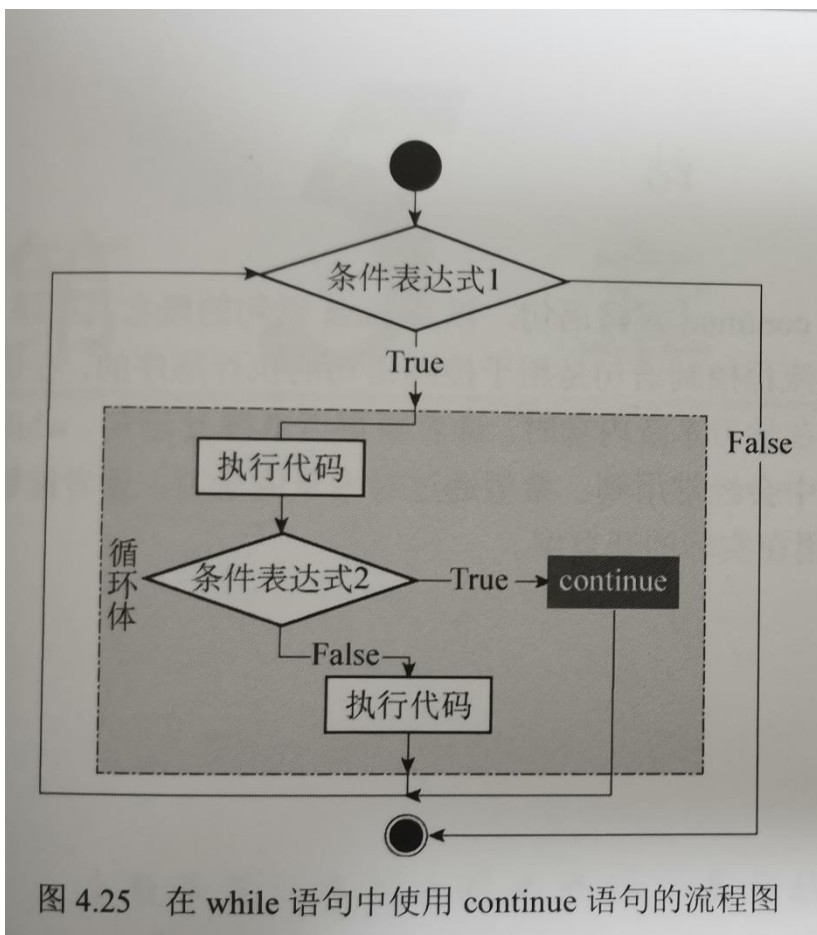
5/4 循环控制语句

continue

- continue语句只能中止本次循环而提前进入下一次循环
- continue语句一般会结合选择语句进行搭配使用

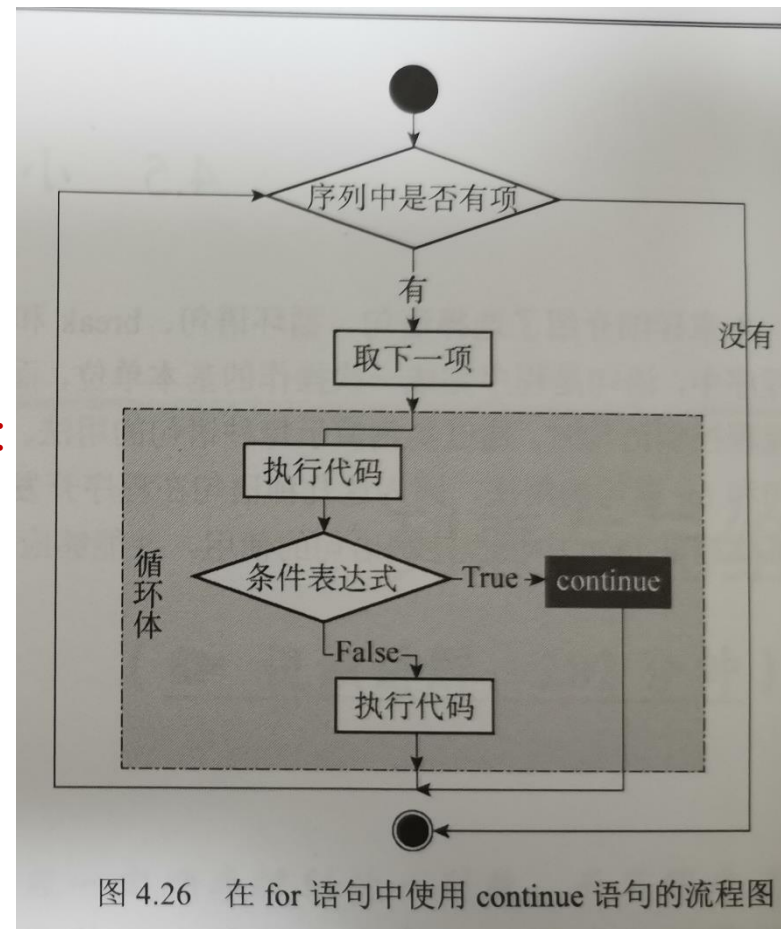
while+continue

while 条件表达式1:
 执行代码
 if 条件表达式2:
 continue



for+continue

for 迭代变量 in 对象:
 if 条件表达式:
 continue



5/4 循环控制语句

例：计算100以内所有偶数的和

#循环语句+continue: 例1 计算100以内偶数的和

```
total = 0      # 用于保存累加和的变量
for number in range(1, 100):
    if number%2 == 1:    # 判断是否符合条件
        continue        # 继续下一次循环
    total += number      # 累加偶数的和
print("1到100之间（不包括100）的偶数和为：", total)    # 输出累加结果
```

1到100之间（不包括100）的偶数和为： 2450



5/5 空语句：pass

pass语句，表示空语句。它不做任何事情，一般起到占位作用

例：利用for循环输出1-10的偶数，若不是偶数，用pass语句占位

```
#pass实例：输入1-10的偶数  
  
for i in range(1,11): #range不包括结尾数字  
    if i % 2 == 0: #判断是否为偶数  
        print(i)  
    else:  
        pass #若不是偶数，则不做任何处理
```

2
4
6
8
10



- Python语言基础：注释、缩进、数据类型
- 各类运算符及其优先级
- 选择语句：if
- 循环语句：for



- 1、输入一个数字，输出该数字的阶乘
 - 阶乘计算公式： $n! = n * (n-1) * (n-2) \dots * 2 * 1$
- 2、输入一个年龄，根据下表输出该年龄对应的阶段

年龄段（包含起始年龄，不含终止年龄）	名称
0-6	学龄前儿童
6-12	小学生
12-15	初中生
15-18	高中生
18-22	大学生
22以上	社会人

