

# Python 语言程序设计

**陈 峦 副教授**

**13880209111, [chluan@uestc.edu.cn](mailto:chluan@uestc.edu.cn)**

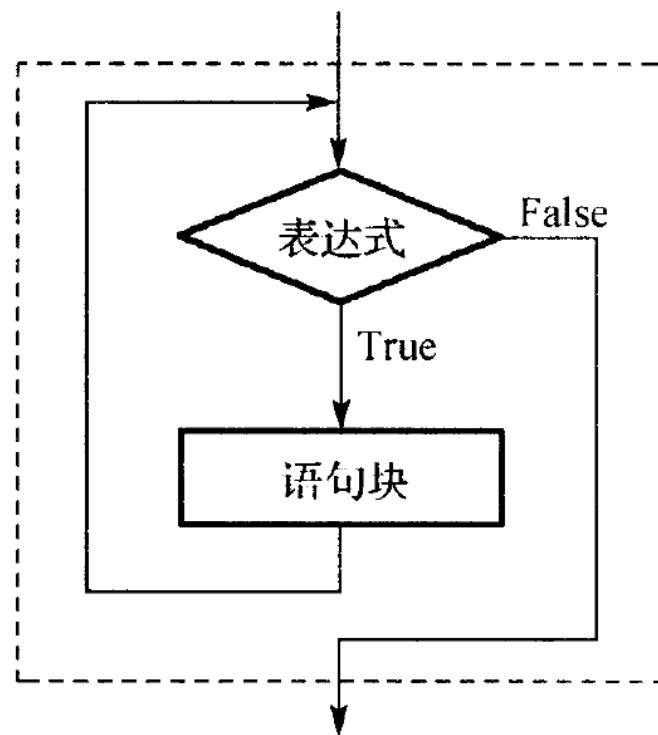
**研究院大楼316#**

# 第四章 循环结构

- 循环结构由循环体及循环条件两部分组成，被重复执行的语句称为循环体，决定是否继续重复的表达式称为循环条件。
- Python提供了while语句和for语句来实现循环结构。

## 4.1 while循环结构

- 当型循环：先判断循环条件，条件满足时执行循环体。
- 4.1.1 While语句
- 1. while语句的一般格式
- while 表达式：
- 语句块



- 例:
- `>>>n=0`
- `>>>while n<10:`
- `print(n,end=' ')`
- `n=n+3`
- `0 3 6 9`
- **注意：**在循环体内必须有修改循环条件表达式值的语句，使其值趋向False，让循环趋于结束，避免无限循环（死循环）。

- 例:

- **i=1**

- **while i<5:**

- **print(i,end=' ')**

- **i=i+1**

```
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
```

- 当程序死循环时，可以按**Ctrl+C**快捷键来中断循环。

- 如果循环体中只有一条语句，可以将该语句与 **while** 写在同一行中。
- 例：
- **>>>i=1**
- **>>>while i in range(5):print(i,end=' ');i+=1**
- **1 2 3 4**

- **2. 在while语句中使用else子句**
- 在Python中，可以在循环语句中使用else子句，else中的语句会在循环**正常执行完**的情况下执行（不管是否执行循环体）。
- 但当通过break语句跳出循环体而中断循环时，else部分将不会被执行。

- 例:
- `>>>i=1`
- `>>>while i in range(5):`
- `print(i,end=' ')`
- `i+=1`
- `else:`
- `print("Goodbye!")`
- `1 2 3 4 Goodbye!`



- 例:
- `count=int(input())`
- `while count<5:`
- `print(count,"is less than 5")`
- `count=count+1`
- `else:`
- `print(count,"is not less than 5")`

程序运行结果:

1

1 is less than 5

2 is less than 5

3 is less than 5

4 is less than 5

5 is not less than 5

- 例:
- `count=int(input())`
- `while count<5:`
- `print(count,"is less than 5")`
- `count=count+1`
- `if count==5:break`
- `else:`
- `print(count,"is not less than 5")`
- 执行**break**时将退出**while**语句,  
    **else**部分也不执行。

程序运行结果:

1

1 is less than 5

2 is less than 5

3 is less than 5

4 is less than 5

- **i=1**
- **while i<5:**
- **print(i)**
- **i=i+1**
- **if i==3:break**
- **print(i)**
- **else:**
- **print("end")**

1  
2  
2

退出本层循环

- **i=1**
- **while i<5:**
- **print(i)**
- **i=i+1**
- **if i==3:continue**
- **print(i)**
- **else:**
- **print("end")**

1  
2  
2  
3  
4  
4  
5  
end

退出本次循环

## 4.1.2 while循环的应用

- 例：计算 $1+2+3+\dots+100$ 的值。
- $s=0$
- $n=1$
- **while  $n \leq 100$ :**    #循环条件
- $s+=n$             #实现累加求和
- $n+=1$             #n增1
- **print("1+2+3+.....+9+100=",s)**

$1+2+3+\dots+9+100= 5050$

- 例：计算 $1+2+3+\dots+100$ 的值。
- 也可以利用列表的求和函数来实现：
- `>>>t=list(range(1,101))`
- `>>>s=sum(t)`
- `>>>s`
- 5050

例：求  $\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots$ ，直到最后一项的绝对值小于  $10^{-6}$  时停止计算。

其中  $x$  为弧度，但从键盘输入时以角度为单位。

- **from math import \***
- **i=1**
- **x1=int(input())**                      #输入一个角度
- **x=radians(x1)**                      #将角度化为弧度
- **s=x;a=x**
- **while fabs(a)>=1e-6:**              #精度不够时继续循环
- **i+=1**
- **a\*=-x\*x/(2\*i-2)/(2\*i-1)**      #求累加项
- **s+=a**
- **print("x={0},sinx={1}".format(x1,s))**

30

x=30, sinx=0.5000000000202799

- 例：输入一个整数，输出其位数。
- `n=int(input())`
- `k=0`
- `while n>0:`
- `k+=1`
- `n//=10`
- `print('k=',k)`

12345

k= 5

## 4.2 for循环结构

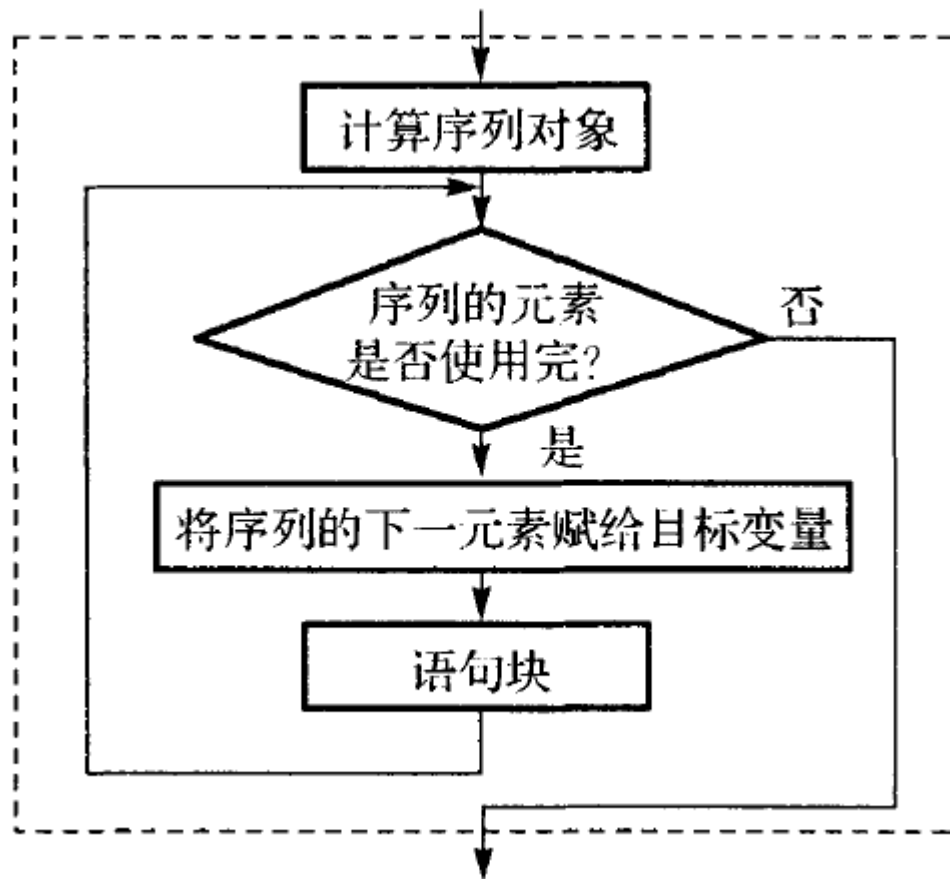
- 计数循环：已知重复执行次数的循环。
- while语句和for语句都可以实现计数循环。
- Python中的for循环也是一个通用的序列迭代器，可以遍历任何有序的序列对象的元素。
- for语句可用于字符串、列表、元组以及其他内置可迭代对象。



## 4.2.1 for语句

### 1. for语句的一般格式

**for** 目标变量 **in** 序列对象:  
语句块



- **for 目标变量 in 序列对象:**
- **语句块**
- **注意:**
- **(1) for语句是通过遍历任意序列的元素进行来建立循环的，针对序列的每一个元素执行一次循环体。**
- **列表、元组、字符串都是序列，可以利用它们来建立循环。**

- 遍历列表:

- 例:

- `s=['AA','BB','CC']`

- `for i in s:`

- `print('s=',i)`

- 程序运行结果:

- `s= AA`

- `s= BB`

- `s= CC`

- 遍历元组:

- 例:

- `s=('AA','BB','CC')`

- `for i in s:`

- `print('s=',i)`

- 程序运行结果:

- `s= AA`

- `s= BB`

- `s= CC`

- 遍历字符串:

- 例:

- `for i in "uestc":`

- `print(i,end='-')`

- 程序运行结果:

- `u-e-s-t-c-`

- 例:

- `ch='c'`

- `for ch in "abcde":`

- `print(ch,end='-')`

- 程序运行结果:

- `a-b-c-d-e-`

- 遍历集合：
- 例：
- `s={'AA','BB','CC'}`
- `for i in s:`
- `print('s=',i)`
- 运行结果可能每次都不相同（无序）。

`s= AA`

`s= CC`

`s= BB`

`s= CC`

`s= AA`

`s= CC`

`s= BB`

`s= AA`

`s= AA`

`s= BB`

`s= BB`

`s= AA`

`s= CC`

`s= BB`

`s= CC`

- (2) **for**循环的循环次数显然就是序列中的元素个数，即序列的长度。
- 可以利用序列长度来控制循环次数，这时关注的不是序列元素的值，而是元素的个数。
- 例：读入5个数，求其和。
- **s=0**
- **for i in [5,2,4,1,3]:**    #与列表元素的值无关
- **x=int(input())**
- **s+=x**
- **print("s=",s)**

- (3) 可以在for循环体中修改目标变量的值，但当程序执行流程再次回到循环开始时，就会自动被设成序列的下一个元素。
- 退出循环之后，该变量的值就是序列中最后的元素。

- 例:
- `s=[1,2,3,4,5]`
- `for i in s:`
- `print(i,end="")`
- `i=i+3`
- `print(",i=",i)`
- 运行结果:
- `12345,i= 8`



- (4) **for**语句也支持一个可选的**else**块，它的功能就像在**while**循环中一样，如果循环离开时没有遇到**break**语句，就会执行**else**块。
- 即序列所有元素都被访问过了之后，执行**else**块。
- 例：
- **for i in "abcd":**
- **print(i,end="")**
- **else:**
- **print(" Goodbye!")**
- 运行结果：
- **abcd Goodbye!**

- 例:
- `for i in "1234":`
- `print(i,end="")`
- `if i=='3':break`
- `print(i,end="")`
- `else:`
- `print(" Goodbye!")`
- 运行结果:
- **11223**
- 退出本层循环

- 例:
- `for i in "1234":`
- `print(i,end="")`
- `if i=='3':continue`
- `print(i,end="")`
- `else:`
- `print(" Goodbye!")`
- 运行结果:
- **1122344 Goodbye!**
- 退出本次循环

## 2. range对象在for循环中的应用

- `range()`函数返回的是可迭代对象。
- 在for循环中，Python将自动调用内置函数`iter()`获得迭代器，自动调用内置函数`next()`获取元素，还完成了检查`stopIteration`异常的工作。
- 如果需要遍历一个数字序列，可以使用`range`对象。

- 例:
- `for i in range(5):`
- `print(i,end=' ')`
- 运行结果:
- `0 1 2 3 4`
- 首先Python对关键字in后的对象调用iter()函数获得迭代器，然后调用next()函数获得迭代器的元素，直到抛出stopIteration异常。

- **i=2**
- **for i in range(5):**
- **print(i,end=' ')**
- **运行结果:**
- **0 1 2 3 4**

- **i=-2**
- **for i in range(5):**
- **print(i,end=' ')**
- **运行结果:**
- **0 1 2 3 4**

- **for i in range(5):**
- **print(i,end=' ')**
- **i+=1**
- **运行结果:**
- **0 1 2 3 4**

- **for i in range(5):**
- **print(i,end=' ')**
- **i-=1**
- **运行结果:**
- **0 1 2 3 4**

- **for i in range(2,7):**

- **print(i,end=' ')**

- **运行结果:**

- **2 3 4 5 6**

- **for i in range(7,2):**

- **print(i,end=' ')**

- **运行结果:**

- **(无)**

- **for i in range(2,7,2):**

- **print(i,end=' ')**

- **运行结果:**

- **2 4 6**

- **for i in range(7,2,-2):**

- **print(i,end=' ')**

- **运行结果:**

- **7 5 3**

- **for i in range(5,2,-1):**

- **print(i,end=' ')**

- **运行结果:**

- **5 4 3**

- **for i in range(2,5,1):**

- **print(i,end=' ')**

- **运行结果:**

- **2 3 4**

- **for i in range('a','f'):**

- **print(i,end=' ')**

- **运行结果:**

- **(语法错误)**

- **for i in range(1,1,1):**

- **print(i,end=' ')**

- **运行结果:**

- **(无)**

- **for i in range(False,True):**
  - **print(i,end=' ')**
  - **运行结果:**
  - **0**
- 

- **for i in range(True,-3,-1):**
- **print(i,end=' ')**
- **运行结果:**
- **1 0 -1 -2**



- **i=2**
- **while i in range(5):**
- **print(i,end=' ')**
- **运行结果:**
- **(死循环) 2 2 2 ...**

- **i=-2**
- **while i in range(5):**
- **print(i,end=' ')**
- **运行结果:**
- **(无)**

- **i=2**
- **while i in range(5):**
- **print(i,end=' ')**
- **i+=1**
- **运行结果: 2 3 4**

- **i=2**
- **while i in range(5):**
- **print(i,end=' ')**
- **i-=1**
- **运行结果: 2 1 0**

- **i=2**
- **while i in range(-5,5):**
- **print(i,end=' ')**
- **i+=2**
- **运行结果: 2 4**

- **i=2**
- **while i in range(1,7,2):**
- **print(i,end=' ')**
- **i+=1**
- **运行结果: (无)**

- **i=1**
- **while i in range(1,7,2):**
- **print(i,end=' ')**
- **i+=1**
- **运行结果: 1**

- **i=1**
- **while i in range(1,7,1):**
- **print(i,end=' ')**
- **i+=2**
- **运行结果: 1 3 5**

- **i=1**
- **while i in range(5,1,-1):**
- **print(i,end=' ')**
- **i+=1**
- **运行结果： （无）**

- **i=1**
- **while i in range(5,-5,-2):**
- **print(i,end=' ')**
- **i+=1**
- **运行结果： 1**

- **i=1**
- **while i in range(5,0,-1):**
- **print(i,end=' ')**
- **i+=1**
- **运行结果： 1 2 3 4 5**

- **i=5**
- **while i in range(5,-5,-1):**
- **print(i,end=' ')**
- **i-=2**
- **运行结果： 5 3 1 -1 -3**

## 4.2.2 for循环的应用

- 例：请写出下面程序段的运行结果。
- `for i in range(1,21):`
- `print(i,end='\t')`
- `if i%5==0:print()`      #控制一行输出5个数

1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20

- 例：输入20个数，输出其中的最大数与最小数。
- `x=int(input())`
- `max=min=x`
- `for i in range(1,20):`
- `x=int(input())`
- `if x>max:`
- `max=x`
- `elif x<min:`
- `min=x`
- `print("max={0},min={1}".format(max,min))`

- 例：输入一个整数 $m$ ，判断其是否为素数。
- 提示：素数是大于1，且除了1和它本身以外，不能被其他任何整数所整除的整数。

```
import math
m=int(input("请输入一个整数: "))
j=int(math.sqrt(m))
flag=True          #素数标志
for i in range(2,j+1):
    if m%i==0:flag=False    #修改素数标志
if flag and m>1:
    print(m,"是素数。")
else:
    print(m,"不是素数。")
```

- 例：输入一个整数 $m$ ，判断其是否为素数。
- 方法二：用while语句实现

```
import math
m=int(input("请输入一个整数："))
i,j=2,int(math.sqrt(m))
flag=1          #素数标志
while i<=j and flag==1:
    if m%i==0:
        flag=0    #不是素数时修改标志
    i+=1          #注意缩进对齐
if flag and m>1: #素数大于1
    print(m,"是素数。")
else:
    print(m,"不是素数。")
```

- **while**语句多用于循环次数不确定的情况，而对于循环次数确定的情况，使用**for**语句更方便。
- 例：读入一批整数，遇到**0**结束，统计其中正数和负数的个数。
- 提示：由于循环次数未知，故最好用**while**语句。



- `m=n=0`
- `x=int(input("请输入一个整数，结束输入请键入0: "))`
- `while x!=0:`
- `if x>0:`
- `m+=1`
- `else:`
- `n+=1`
- `x=int(input("继续输入: "))`
- `print("共读入了{}个非零整数，其中，有{}个正数，{}个负数。  
".format(m+n,m,n))`

请输入一个整数，结束输入请键入0: 23

继续输入: -5

继续输入: 6

继续输入: 0

共读入了3个非零整数，其中，有2个正数，1个负数。

- 例：请输出所有的水仙花数。（注：一个三位整数，其各位数字的立方和等于该数本身，则称其为水仙花数。例如 $153=1^3+5^3+3^3$ 。）
- 提示：由于循环次数已知，故最好用for语句。
- **for i in range(100,1000):**
- **a=i//100**
- **b=i%100//10 #或b=i/10%10**
- **c=i%10**
- **if i==a\*\*3+b\*\*3+c\*\*3:**
- **print(i)**

153  
370  
371  
407

## 4.3 循环控制语句

- 循环控制语句可以改变循环的执行路径。
- Python支持以下循环控制语句：**break**语句、**continue**语句和**pass**语句。

## 4.3.1 break语句

- **break**语句用在循环体内，迫使所在循环立即终止，即跳出所在循环体，继续执行循环结构后面的语句。
- 提前结束本层循环。

- 例:
- **i=10**
- **while i>0:**
- **print(i,end=' ')**
- **i=i-1**
- **if i==5:break**
- **print("Goodbye!")**
- 运行结果:
- **10 9 8 7 6 Goodbye!**

- 例：求两个整数a与b的最大公约数。
- `a,b=eval(input("请输入两个整数："))`
- `if a>b:a,b=b,a`                      #保证a为较小的数
- `for i in range(a,0,-1):`
- `if a%i==0 and b%i==0:`            #第一次能同时整除
- `print("最大公约数是",i)`
- `break`

- 例：求两个整数a与b的最大公约数。
- 方法二：辗转相除法
- （1）求a/b的余数r。
- （2）若r=0,则b为最大公约数,否则执行第（3）步。
- （3）将b的值放在a中，r的值放在b中。
- （4）转到第（1）步。

```
a,b=eval(input("请输入两个整数: "))  
if a>b:a,b=b,a      #保证a为较小的数  
r=a % b  
while r!=0:  
    a,b=b,r  
    r=a%b  
print("最大公约数是",b)
```

## 4.3.2 continue语句

- 在循环结构中执行**continue**语句时，并不会退出循环结构，而是立即结束本次循环，重新开始下一轮循环。
- 跳过循环体中在**continue**语句之后的所有语句，继续下一轮循环。
- 提前结束本次循环。



- 例:
- `i=5`
- `while i>0:`
- `i=i-1`
- `if i==3:continue`
- `print(i,end=' ')`
- `print("Goodbye!")`
- 运行结果:
- `4 2 1 0 Goodbye!`

- 例:
- `i=5`
- `while i>0:`
- `i=i-1`
- `if i==3:break`
- `print(i,end=' ')`
- `print("Goodbye!")`
- 运行结果:
- `4 Goodbye!`

- 例：求1~100之间的全部奇数之和。
- **x=y=0**
- **while True:**
- **x+=1**
- **if x%2==0:**
- **continue** #x为偶数，就直接进入下一次循环
- **elif x>100:break**     **#x>100，就直接退出循环**
- **else:y+=x**             **#实现累加**
- **print("y=",y)**

y= 2500

## 4.3.3 pass语句

- **pass**语句是一个空语句，它不做任何操作，代表一个空操作。
- **pass**语句用于在某些场合下语法上需要一个语句但实际却什么都不做的情况，就相当于一个占位符。

- 例:
- **`x,y=2,3`**
- **`if x>y:`**
- **`pass`**
- **`else:`**
- **`x,y=y,x`**
- **`print("max=",x)`**

- 例：软件延时程序
- `import time`
- `start=time.perf_counter()`
- `for i in range(10001):`
- `pass`
- `t=time.perf_counter()-start`
- `print("t={}s".format(t))`

`t=0.0017964225502196851s`

## 4.4 循环的嵌套

- 如果一个循环结构的循环体又包括一个循环结构，就称为循环的嵌套，或称为多重循环结构。
- 在多重循环结构中，处于内部的循环称为内循环，处于外部的循环称为外循环。
- 在多重循环中，**break**语句将停止执行所在层的循环（本层循环），返回执行外一层的循环体。

- 例： 写出下面程序段的运行结果。

```
for i in range(1,6):  
    for j in range(1,6):  
        print(j,end=' ')  
    print()
```

```
for i in range(1,6):  
    for j in range(1,6):  
        print(i,end=' ')  
    print()
```

- 例：写出下面程序段的运行结果。

```
for i in range(1,6):  
    for j in range(1,i):  
        print(j,end=' ')  
    print()
```

```
for i in range(1,6):  
    for j in range(1,i):  
        print(i+j,end=' ')  
    print()
```



- 例：写出下面程序段的运行结果。

```
for i in range(1,6):  
    for j in range(i,6):  
        print(j,end=' ')  
    print()
```

```
for i in range(1,6):  
    for j in range(1,6-i):  
        print(i,end=' ')  
    print()
```

- 例： 写出下面程序段的运行结果。

```
for i in range(1,6):  
    for j in range(1,i+6):  
        print(j,end=' ')  
    print()
```

```
for i in range(1,6):  
    for j in range(i,i+6):  
        print(i,end=' ')  
    print()
```

- 例：写出下面程序段的运行结果。

```
for i in range(1,6):  
    for j in range(1,2*i):  
        print(j,end=' ')  
    print()
```

```
for i in range(1,6):  
    for j in range(i,2*i):  
        print(i,end=' ')  
    print()
```

- 例：写出下面程序段的运行结果。

```
for i in range(1,6):  
    for j in range(1,6-i):  
        print(' ',end='')  
    for j in range(1,2*i):  
        print(i,end='')  
    print()
```

```
for i in range(1,6):  
    for j in range(1,6-i):  
        print(' ',end='')  
    for j in range(i,2*i):  
        print(i,end='')  
    print()
```

- 例：写出下面程序段的运行结果。

```
for i in range(1,6):  
    for j in range(1,6-i):  
        print(' ',end='')  
    for j in range(1,2*i):  
        print(j,end='')  
    print()
```

```
for i in range(1,6):  
    for j in range(1,6-i):  
        print(' ',end='')  
    for j in range(i,i+6):  
        print(i,end='')  
    print()
```

- 例：写出下面程序段的运行结果。

```
for i in range(1,6):
    for j in range(1,6-i):
        print(' ',end='')
    for j in range(1,2*i):
        print(i,end='')
    print()
for i in range(1,5):
    for j in range(1,i+1):
        print(' ',end='')
    for j in range(1,10-2*i):
        print(5+i,end='')
    print()
```

```
n=10
for i in range(1,n+1):
    for j in range(1,n-i+1):
        print(' ',end='')
    for j in range(1,2*i):
        print('*',end='')
    print()
for i in range(1,n):
    for j in range(1,i+1):
        print(' ',end='')
    for j in range(1,2*n-2*i):
        print('*',end='')
    print()
```

- 例：请打印“九九乘法表”。
- **for i in range(1,10):**
- **for j in range(1,i+1):**
- **print("{}x{}={:<3}".format(j,i,i\*j),end="")**
- **#或print(j,"x",i,"=",format(i\*j,"<3"),sep=" ",end="")**
- **print()**

```
1x1=1
1x2=2  2x2=4
1x3=3  2x3=6  3x3=9
1x4=4  2x4=8  3x4=12  4x4=16
1x5=5  2x5=10  3x5=15  4x5=20  5x5=25
1x6=6  2x6=12  3x6=18  4x6=24  5x6=30  6x6=36
1x7=7  2x7=14  3x7=21  4x7=28  5x7=35  6x7=42  7x7=49
1x8=8  2x8=16  3x8=24  4x8=32  5x8=40  6x8=48  7x8=56  8x8=64
1x9=9  2x9=18  3x9=27  4x9=36  5x9=45  6x9=54  7x9=63  8x9=72  9x9=81
```

● 例：求 $s=1!+2!+3!+\dots+n!$

方法一：

```
s=0
n=int(input("n="))
for i in range(1,n+1):
    t=1
    for j in range(1,i+1):
        t*=j
    s+=t
print("s=",s)
```

方法二：

```
s=0
t=1
n=int(input("n="))
for i in range(1,n+1):
    t*=i
    s+=t
print("s=",s)
```



例：输入 n，求下列表达式的值。

$$1 + \frac{1}{1+2} + \frac{1}{1+2+3} + \cdots + \frac{1}{1+2+3+\cdots+n}$$

方法一：

```
s=0
```

```
n=int(input("n="))
```

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

```
    t=0
```

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

```
        t+=j
```

```
    x=1/t
```

```
    s+=x
```

```
print("s=",s)
```

方法二：

```
s=0
```

```
t=0
```

```
n=int(input("n="))
```

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

```
    t+=i
```

```
    x=1/t
```

```
    s+=x
```

```
print("s=",s)
```

- 例：输出三位整数中的全部素数。

```
import math
n=0
for m in range(101,1000,2): #大于2的素数全为奇数
    i,j=2,int(math.sqrt(m))
    while i<=j:
        if m%i==0:
            break
        else:
            i=i+1
    else:
        print(m,end=" ")
        n+=1      #n统计素数个数
        if n%10==0:print()  #每行输出10个素数
```

- 例：请输出所有的水仙花数。（注：一个三位整数，其各位数字的立方和等于该数本身，则称其为水仙花数。例如 $153=1^3+5^3+3^3$ 。）
- 方法一：算法的时间复杂度 $T(n)=O(n)$ 。
- `for i in range(100,1000):`
- `a=i//100`
- `b=i%100//10 #或b=i/10%10`
- `c=i%10`
- `if i==a**3+b**3+c**3:`
- `print(i)`

153  
370  
371  
407

- 例：请输出所有的水仙花数。
- 方法二：算法的时间复杂度 $T(n)=O(n^3)$ 。
- **for i in range(1,10):**
- **for j in range(0,10):**
- **for k in range(0,10):**
- **n=i\*100+j\*10+k**
- **if n==i\*\*3+j\*\*3+k\*\*3:**
- **print(n)**

153  
370  
371  
407

- 例：百鸡问题。鸡翁一，值钱五；鸡母一，值钱三；鸡雏三，值钱一。百钱买百鸡，问鸡翁、母、雏各多少？

- 方法一：算法的时间复杂度 $T(n)=O(n^3)$ 。

- `for i in range(0,101):` #实际上 $i < 15$

- `for j in range(0,101):` #实际上 $i < 26$

- `for k in range(0,101):` #实际上 $i < 87$

- `if (i+j+k==100) and (15*i+9*j+k==300):`

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

`i=0, j=25, k=75`  
`i=4, j=18, k=78`  
`i=8, j=11, k=81`  
`i=12, j=4, k=84`

- 实际上，当*i*和*j*的值确定后，*k*的值就唯一了！
- $k=100-i-j$
- 方法二：算法的时间复杂度 $T(n)=O(n^2)$ 。
- `for i in range(0,101):` #实际上*i*<15
- `for j in range(0,101):` #实际上*i*<26
- $k=100-i-j$
- `if (k>=0) and (15*i+9*j+k==300):`
- `print("i={},j={},k={}".format(i,j,k))`

- $i+j+k=100$
- $15i+9j+k=300$
- 由上述两个方程可以解出：  $j=f(i)$ ;  $k=g(i)$
- 方法三： 算法的时间复杂度  $T(n)=O(n)$ 。
- 伪代码：
- `for i in range(0,101):`    #实际上  $i < 15$
- `j=f(i)`
- `k=g(i)`
- `if (0<=i<=100) and 0<=k<=100):`
- `print("i={},j={},k={}".format(i,j,k))`

- 可执行代码:
- `for i in range(0,101):` #实际上 $i < 15$
- `j=100-7*i` # $4j=100-7i$
- `k=300+3*i` # $4k=300+3i$
- `if (0<=j<=400) and (0<=k<=400) and (j%4==0) and (k%4==0):`
- `print("i={},j={},k={}".format(i,j//4,k//4))`



- 结论：
- 有 $n$ 个未知数，只有 $m$ 个方程，则求解时至少需要 $(n-m)$ 重循环。
- 例：一张100元的钞票，需要换成10元、5元、2元和1元的零钞40张，问共有几种换法？

- 分析:  $a+b+c+d=40$ ;  $10a+5b+2c+d=100$
- $0 \leq a, b, c, d \leq 40$
- 4个未知数, 2个方程, 求解时至少需要2重循环。
- $n=0$
- `for a in range(0,41):` #实际上 $a < 7$
- `for b in range(0,41):` #实际上 $b < 16$
- $c=60-9*a-4*b$  # $c=60-9a-4b$
- $d=8*a+3*b-20$  # $d=8a+3b-20$
- `if (0<=c<=40) and (0<=d<=40):`
- `print("a={},b={},c={},d={}".format(a,b,c,d))`
- $n=n+1$
- `print("n=",n)`

- 例：一张100元的钞票，需要换成10元、5元、2元和1元的零钞40张，要求每种零钞至少有一张，问共有几种换法？

- `n=0`

- `for a in range(1,41):` #实际上 $a < 7$

- `for b in range(1,41):` #实际上 $b < 16$

- `c=60-9*a-4*b` # $c=60-9a-4b$

- `d=8*a+3*b-20` # $d=8a+3b-20$

- `if (1<=c<=40) and (1<=d<=40):`

- `print("a={},b={},c={},d={}".format(a,b,c,d))`

- `n=n+1`

- `print("n=",n)`

`n= 34`

- 例：将1元钱换成1分、2分、5分的硬币，共有多少种方法？求解时至少需要几重循环？
- 例：将1元钱换成1分、2分、5分的硬币40枚，共有多少种方法？求解时至少需要几重循环？

```
count=0
for x in range(101):
    for y in range(51):
        for z in range(21):
            if x+2*y+5*z==100:
                print("x={0},y={1},z={2}".format(x,y,z))
                count+=1
print("count={0}".format(count))
```

---

```
count=0
for y in range(51):
    for z in range(21):
        x=100-2*y-5*z
        if x>=0:
            print("x={0},y={1},z={2}".format(x,y,z))
            count+=1
print("count={0}".format(count))
```

## 4.5 循环结构程序举例

例：已知  $y = 1 + \frac{1}{3} + \frac{1}{5} + \cdots + \frac{1}{2n-1}$ ，求：

(1)  $y < 3$  时的最大  $n$  值。(2) 与 (1) 的  $n$  值对应的  $y$  值。

$y=0$
$n=1$
当 $y < 3$ 时
$t=1/(2n-1)$
$y=y+t$
$n=n+1$
输出 $y-t, n-2$

- 当退出循环时， $y$  的值已超过 3，因此要减去最后一项， $n$  的值相应也要减去 1。
- 又由于最后一项累加到  $y$  后， $n$  又增加了 1，故  $n$  还要减去 1，即累加的项数是  $n-2$ 。

例：已知  $y = 1 + \frac{1}{3} + \frac{1}{5} + \cdots + \frac{1}{2n-1}$ ，求：

(1)  $y < 3$  时的最大  $n$  值。(2) 与 (1) 的  $n$  值对应的  $y$  值。

- **$n=1$**

- **$y=0$**

- **while  $y < 3$ :**

- **$t=1.0/(2*n-1)$     #求累加项**

- **$y+=t$     #累加**

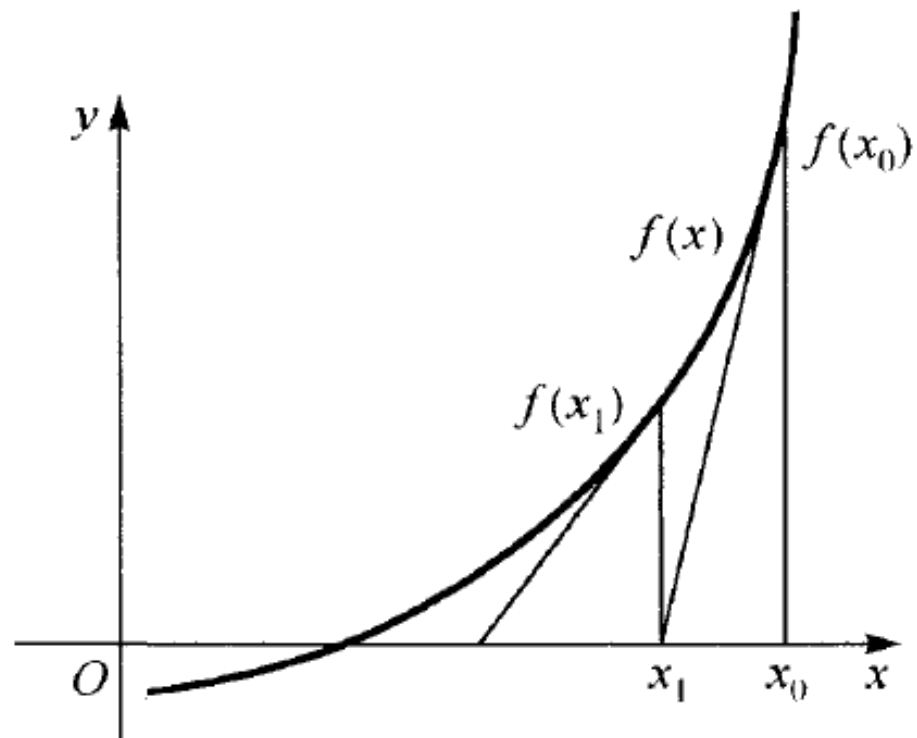
- **$n+=1$**

- **`print("y={0},n={1}".format(y-t,n-2))`**

- **#退出循环时的 $y$ 值和 $n$ 值与待求 $y$ 和 $n$ 不同**

$y=2.994437501289942, n=56$

例：用牛顿迭代法求方程  $f(x)=2x^3-4x^2+3x-7=0$  在  $x=2.5$  附近的实根，



直到满足  $|x_n - x_{n-1}| \leq 10^{-6}$  为止。

牛顿迭代公式为：

$$x_n = x_{n-1} - \frac{f(x_{n-1})}{f'(x_{n-1})}$$

$$(n = 1, 2, 3, \dots)$$

其中， $f'(x)$ 为  $f(x)$ 的一阶导数。

牛顿迭代法（切线法）：以切线与x轴的交点作为曲线与x轴交点的近似值，以逐步逼近解。

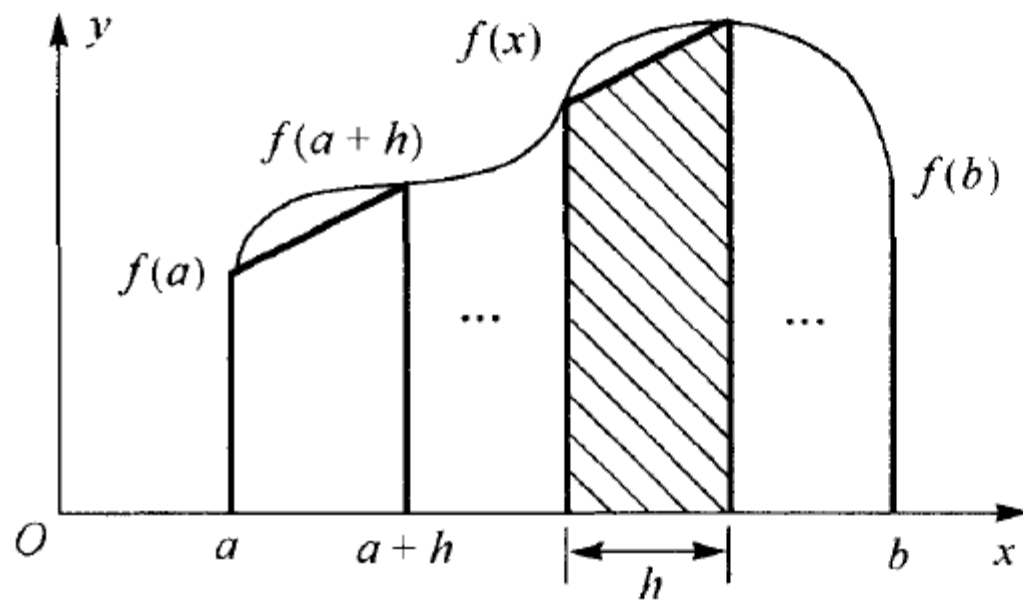
$$f(x)=((2x-4)x+3)x-7, \quad f'(x)=6x^2-8x+3=(6x-8)x+3$$



- **import math**
- **d,x2=1,2.5**
- **while math.fabs(d)>1.0e-6:**
- **x1=x2**
- **d=((((2.0\*x1-4.0)\*x1+3.0)\*x1-7.0)/((6.0\*x1-8.0)\*x1+3.0))**
- **x2=x1-d**
- **print("x=",x2)**

**x= 2.085481303712927**

例：设  $f(x) = \frac{1}{1+x}$ ，求  $f(x)$  在  $[a, b]$  上的定积分  $\int_a^b f(x) dx$ 。



定积分的几何意义就是求曲线  $y=f(x)$  与直线  $x=a$ ， $x=b$ ， $y=0$  所围成的曲边梯形的面积。

第一个小梯形的面积为：
$$s_1 = \frac{f(a) + f(a+h)}{2} \cdot h$$

第二个小梯形的面积为：
$$s_2 = \frac{f(a+h) + f(a+2h)}{2} \cdot h$$

.....

第  $n$  个小梯形的面积为：
$$s_n = \frac{f[a + (n-1)h] + f(a + n \cdot h)}{2} \cdot h$$

- `a,b,n=eval(input("a,b,n="))`

- `s=0`

- `h=(b-a)/n`

- `x=a`

- `f0=1/(1+x)`

- `for i in range(1,n+1):`

- `x=x+h`                    `#求x`

- `f1=1/(1+x)`            `#求新的函数值`

- `s=s+(f0+f1)*h/2` `#求小梯形的面积并累加`

- `f0=f1`                    `#更新函数值`

- `print("s=",s)`

```
a, b, n=0, 2, 1000
```

```
s= 1.0986125849642736
```

```
>>> import math
```

```
>>> math.log(3)        #ln(3)
```

```
1.0986122886681098
```

$$\int_0^2 \frac{1}{1+x} dX = [\ln(1+x)]_0^2 = \ln 3 - \ln 1 = \ln 3$$

- 例：编写一个“猜数字游戏”的程序。在  $\text{min} \sim \text{max}$  之间随机产生一个数，然后请用户循环猜这个数字。对于每个答案只回答“大了”或“小了”，直到猜对为止。输出用户的猜测次数。

```
猜数字游戏
请输入下限： 1
请输入上限： 10
现在开始：
第1次猜数： 5
恭喜你，猜对了！
本轮的猜测次数是： 1
```

```
猜数字游戏
请输入下限： 1
请输入上限： 10
现在开始：
第1次猜数： 5
小了！
第2次猜数： 7
小了！
第3次猜数： 9
恭喜你，猜对了！
本轮的猜测次数是： 3
```

- `print("猜数字游戏")`
- `min=eval(input("请输入下限: "))`
- `max=eval(input("请输入上限: "))`
- `import random`
- `t=random.randint(min,max)`
- `count=0`
- `print("现在开始: ")`
- `while True:`
  - `count=count+1`
  - `print("第{}次猜数: ".format(count),end=")`
  - `g=eval(input())`
  - `if g>t:`
    - `print("大了!")`
  - `elif g<t:`
    - `print("小了!")`
  - `else:`
    - `print("恭喜你, 猜对了!")`
    - `break`
- `print("本轮的猜测次数是: ",count)`

# 习题课

- 一、选择题
- 1. 关于**while**循环和**for**循环的区别，下列叙述中正确的是（    ）。
- A. **while**语句的循环体至少无条件执行一次，**for**语句的循环体有可能一次都不执行
- B. **while**语句只能用于循环次数未知的循环，**for**语句只能用于循环次数已知的循环
- C. 在很多情况下，**while**语句和**for**语句可以等价使用
- D. **while**语句只能用于可迭代变量，**for**语句可以用任意表达式表示条件

- 一、选择题

- 1. 关于**while**循环和**for**循环的区别，下列叙述中正确的是（    ）。C
- A. **while**语句的循环体至少无条件执行一次，**for**语句的循环体有可能一次都不执行
- B. **while**语句只能用于循环次数未知的循环，**for**语句只能用于循环次数已知的循环
- C. 在很多情况下，**while**语句和**for**语句可以等价使用
- D. **while**语句只能用于可迭代变量，**for**语句可以用任意表达式表示条件

- 2. 设有程序段:

- **k=10**

- **while k:**

- **k=k-1**

- **print(k)**

- 则下面描述中正确的是（    ）。

- **A. while循环执行10次**

- **B. 循环是无限循环**

- **C. 循环体语句一次也不执行**

- **D. 循环体语句执行一次**



- 2. 设有程序段:

- **k=10**

- **while k:**

- **k=k-1**

- **print(k)**

- 则下面描述中正确的是（    ）。 **A**

- **A. while循环执行10次**

- **B. 循环是无限循环**

- **C. 循环体语句一次也不执行**

- **D. 循环体语句执行一次**

● 3. 以下while语句中的表达式“not E”等价于  
(    )。

● while not E:

●     pass

● A. E==0

B. E!=1

● C. E!=0

D. E==1

● 3. 以下while语句中的表达式“not E”等价于

(    )。 A

● while not E:

●     pass

● A. E==0

B. E!=1

● C. E!=0

D. E==1

- 4. 有以下程序段：
- `n=0`
- `p=0`
- `while not (p==100 or n>3):`
- `p=int(input())`
- `n+=1`
- `while`循环结束的条件是（    ）。
- A. P的值不等于100并且n的值小于等于3
- B. P的值等于100并且n的值大于3
- C. P的值不等于100或者n的值小于等于3
- D. P的值等于100或者n的值大于3

- 4. 有以下程序段:
- `n=0`
- `p=0`
- `while not (p==100 or n>3):`
- `p=int(input())`
- `n+=1`
- `while`循环结束的条件是 (    ) 。 D
- A. P的值不等于100并且n的值小于等于3
- B. P的值等于100并且n的值大于3
- C. P的值不等于100或者n的值小于等于3
- D. P的值等于100或者n的值大于3

● 5. 以下for语句中，不能完成1~10的累加功能的是（ ）。

● A. `for i in range(10,0):sum+=i;i=i-1`

● B. `for i in range(1,11):sum+=i`

● C. `for i in range(10,0,-1):sum+=i`

● D. `for i in (10,9,8,7,6,5,4,3,2,1):sum+=i`

● 5. 以下for语句中，不能完成1~10的累加功能的是（    ）。 A

● A. `for i in range(10,0):sum+=i;i=i-1`

● B. `for i in range(1,11):sum+=i`

● C. `for i in range(10,0,-1):sum+=i`

● D. `for i in (10,9,8,7,6,5,4,3,2,1):sum+=i`

● 6. 对下列语句不符合语法要求的表达式是  
(    )。

● `for var in (    ):`

● `print(var)`

● A. `(True,2.34,"abc")`

● B. `Hello`

● C. `[True,2.34,"abc"]`

● D. `{"abc",12.3,True}`



● 6. 对下列语句不符合语法要求的表达式是  
(    )。 B

● for var in (    ):

●     print(var)

● A. (True,2.34,"abc")

● B. Hello

● C. [True,2.34,"abc"]

● D. {"abc",12.3,True}

● 7. 下面Python循环体执行的次数与其他不同的是  
(    )。

A.

```
i=0
```

```
while i<=10:
```

```
    print(i)
```

```
    i+=1
```

B.

```
i=10
```

```
while i>0:
```

```
    print(i)
```

```
    i-=1
```

C.

```
for i in range(10):
```

```
    print(i)
```

D.

```
for i in range(10,0,-1):
```

```
    print(i)
```

● 7. 下面Python循环体执行的次数与其他不同的是  
(    )。 A

A.

```
i=0
```

```
while i<=10:
```

```
    print(i)
```

```
    i+=1
```

B.

```
i=10
```

```
while i>0:
```

```
    print(i)
```

```
    i-=1
```

C.

```
for i in range(10):
```

```
    print(i)
```

D.

```
for i in range(10,0,-1):
```

```
    print(i)
```

● 8. 下列for循环执行后，输出结果的最后一行是  
(    )。

● **for i in range(1,3):**

●     **for j in range(2,5):**

●         **print(i\*j)**

● **A. 2      B. 6      C. 8      D. 15**

● 8. 下列for循环执行后，输出结果的最后一行是  
(    )。 C

● for i in range(1,3):

●     for j in range(2,5):

●         print(i\*j)

● A. 2        B. 6        C. 8        D. 15

● 9. 关于下列for循环，叙述正确的是（    ）。

● `for t in range(1,11):`

●     `x=int(input())`

●     `if x<0:continue`

●     `print(x)`

● A. 当 $x<0$ 时整个循环结束

● B.  $x\geq 0$ 时什么也不输出

● C. `print()`函数永远也不执行

● D. 最多允许输出10个非负整数

● 9. 关于下列for循环，叙述正确的是（    ）。 D

● `for t in range(1,11):`

●     `x=int(input())`

●     `if x<0:continue`

●     `print(x)`

● A. 当 $x<0$ 时整个循环结束

● B.  $x\geq 0$ 时什么也不输出

● C. `print()`函数永远也不执行

● D. 最多允许输出10个非负整数

- 10. 下列说法中正确的是（    ）。
- A. **break**用在**for**语句中，而**continue**用在**while**语句中
- B. **break**用在**while**语句中，而**continue**用在**for**语句中
- C. **continue**能结束循环，而**break**只能结束本次循环
- D. **break**能结束循环，而**continue**只能结束本次循环



- 10. 下列说法中正确的是（    ）。D
- A. **break**用在**for**语句中，而**continue**用在**while**语句中
- B. **break**用在**while**语句中，而**continue**用在**for**语句中
- C. **continue**能结束循环，而**break**只能结束本次循环
- D. **break**能结束循环，而**continue**只能结束本次循环

## ● 二、填空题

● 1. 当循环结构的循环体由多个语句构成时，必须用（ ）的方式组成一个语句块。

● 2. 执行下列程序后的输出结果是（ ），其中 **while** 循环执行了（ ）次。

● **i=-1**

● **while i<0:**

● **i\*=i**

● **print(i)**

- 二、填空题

- 1. 当循环结构的循环体由多个语句构成时，必须用（ ）的方式组成一个语句块。缩进对齐

- 2. 执行下列程序后的输出结果是（ ），其中while循环执行了（ ）次。1， 1

- `i=-1`

- `while i<0:`

- `i*=i`

- `print(i)`

● 3. 以下while循环的循环次数是（ ）。

● `i=0`

● `while i<10:`

● `if i<1:continue`

● `if i==5:break`

● `i+=1`

● 3. 以下while循环的循环次数是（ ）。无限次

● `i=0`

● `while i<10:`

● `if i<1:continue`

● `if i==5:break`

● `i+=1`

● 4. 执行下列程序后，k的值是（    ）。

● **k=1**

● **n=263**

● **while n:**

●     **k\*=n%10**

●     **n//=10**

● 4. 执行下列程序后，k的值是（    ）。 36

●  $k=1$

●  $n=263$

● while n:

●      $k*=n\%10$

●      $n//=10$

- 5. 执行循环语句**for i in range(1,5,2):print(i)**，循环体执行的次数是（    ）。
- 6. 循环语句**for i in range(-3,21,4):print(i)**的循环次数为（    ）。
- 7. 要使语句**for i in range(26,-4,-2):print(i)**循环执行**15**次，则循环变量**i**的初值**k**应当为（    ）。



- 5. 执行循环语句**for i in range(1,5,2):print(i)**，循环体执行的次数是（    ）。 **2**
- 6. 循环语句**for i in range(-3,21,4):print(i)**的循环次数为（    ）。 **6**
- 7. 要使语句**for i in range(26,-4,-2):print(i)**循环执行**15**次，则循环变量**i**的初值**k**应当为（    ）。 **25**  
或**26**

- 8. 执行循环语句for i in range(1,5):pass后，变量i的值是（ ）。
- 9. 一个循环结构的循环体又包括一个循环结构，称为（ ）或（ ）结构。

- 8. 执行循环语句for i in range(1,5):pass后，变量i的值是（ ）。4
- 9. 一个循环结构的循环体又包括一个循环结构，称为（ ）或（ ）结构。循环嵌套，多重循环

● **10. 下列程序的输出结果是（    ）。**

● **s=10**

● **for i in range(1,6):**

●     **while True:**

●         **if i%2==1:**

●             **break**

●         **else:**

●             **s-=1**

●             **break**

● **print(s)**

● **10. 下列程序的输出结果是（    ）。 8**

● **s=10**

● **for i in range(1,6):**

●     **while True:**

●         **if i%2==1:**

●             **break**

●         **else:**

●             **s-=1**

●             **break**

● **print(s)**

- 三、问答题

- 1. 下列程序的输出结果是什么？如果将语句“**print(s)**”与语句“**pass**”缩进对齐，则输出结果是什么？通过比较两次输出结果，可以得到什么结论？

- **s=10**

- **for i in range(1,6):**

- **pass**

- **print(s)**

- **s=10**

- **for i in range(1,6):**

- **pass**

- **print(s)**

- 2. 用while语句改写下列程序。

- `s=0`

- `for i in range(2,101,2):`

- `s+=i`

- `print(s)`

# 自测题

- 一、选择题
- 1. **for**或者**while**与**else**搭配使用时，什么时候会执行**else**对应的语句块？（ ）
- A.总会执行
- B.永不执行
- C.仅循环正常结束时（**for**遍历完成/**while**条件不满足）
- D.仅循环非正常结束时（以**break**结束）



# 自测题

- 一、选择题
- 1. **for**或者**while**与**else**搭配使用时，什么时候会执行**else**对应的语句块？（ ） **C**
- **A.**总会执行
- **B.**永不执行
- **C.**仅循环正常结束时（**for**遍历完成/**while**条件不满足）
- **D.**仅循环非正常结束时（以**break**结束）

● 2. 关于**break**的作用，下列说法中正确的是（ ）

● A.按照缩进跳出一层语句块

● B.按照缩进跳出除函数缩进外的所有语句块

● C.跳出一层**for/while**循环

● D.跳出所有**for/while**循环

● 3. 下面程序段的运行结果是（ ）

● **for i in range(3):**

● **print(i,end=' ')**

● **A.0 1 2      B.1 2 3      C.0 1 2 3      D.3**

● 2. 关于**break**的作用，下列说法中正确的是（ ） **C**

● **A.**按照缩进跳出一层语句块

● **B.**按照缩进跳出除函数缩进外的所有语句块

● **C.**跳出一层**for/while**循环

● **D.**跳出所有**for/while**循环

● 3. 下面程序段的运行结果是（ ） **A**

● **for i in range(3):**

● **print(i,end=' ')**

● **A.0 1 2      B.1 2 3      C.0 1 2 3      D.3**

● 4. 下面程序段的运行结果是 ( )

● `for i in (1,3,5):`

● `print(i,end=' ')`

● A.1 2 3            B.1 4 7 10 13 16

● C.1 3 5            D.1 6 11 16

● 5. 下面程序段的运行结果是 ( ) (□表示空格)

● `for i in ['12','3','45']:`

● `print(i,end=' ')`

● A.1□2□3□4□5            B.1□2□3

● C.12345            D.12□3□45

● 4. 下面程序段的运行结果是 ( ) C

● `for i in (1,3,5):`

● `print(i,end=' ')`

● A.1 2 3            B.1 4 7 10 13 16

● C.1 3 5            D.1 6 11 16

● 5. 下面程序段的运行结果是 ( ) (□表示空格) D

● `for i in ['12','3','45']:`

● `print(i,end=' ')`

● A.1□2□3□4□5            B.1□2□3

● C.12345            D.12□3□45

● 6. 下面程序段的运行结果是 ( )

● `for i in range(1,3,-1):`

● `print(i,end=' ')`

● A.1 2 3    B.1 0 -1    C.1 3 5    D.无输出

● 7. 下面程序段的运行结果是 ( )

● `for i in range(1,5,1):`

● `print(i,end=' ');i=i-2`

● A.1 2 3 4    B.1 2 3 4 5

● C.死循环    D.无输出

● 6. 下面程序段的运行结果是 ( ) D

● `for i in range(1,3,-1):`

● `print(i,end=' ')`

● A.1 2 3    B.1 0 -1    C.1 3 5    D.无输出

● 7. 下面程序段的运行结果是 ( ) A

● `for i in range(1,5,1):`

● `print(i,end=' ');i=i-2`

● A.1 2 3 4    B.1 2 3 4 5

● C.死循环    D.无输出

● 8. 下面程序段不是死循环的是 ( )

- |  |                                       |
|--|---------------------------------------|
| A.   | B.                                    |
| <pre>i=1;s=2</pre>                         | <pre>i=1;</pre>                       |
| <pre>while i&lt;s:</pre>                   | <pre>while i&lt;3:</pre>              |
| <pre>    s+=i;i=i+1;print(i,end=' ')</pre> | <pre>    print(i,end=' ')</pre>       |
| C.   | D.                                    |
| <pre>i=1;</pre>                            | <pre>for i in range(1,5,1):</pre>     |
| <pre>while i!=6:</pre>                     | <pre>    i=i-1;print(i,end=' ')</pre> |
| <pre>    i+=2;print(i,end=' ')</pre>       |                                       |



● 8. 下面程序段不是死循环的是 ( ) **D**

A.

```
i=1;s=2
```

```
while i<s:
```

```
    s+=i;i=i+1;print(i,end=' ')
```

B.

```
i=1;
```

```
while i<3:
```

```
    print(i,end=' ')
```

C.

```
i=1;
```

```
while i!=6:
```

```
    i+=2;print(i,end=' ')
```

D.

```
for i in range(1,5,1):
```

```
    i=i-1;print(i,end=' ')
```

● 9. 下面程序段不是死循环的是 ( )

A.

```
i=1;
```

```
while i<3:
```

```
    print(i,end=' ')
```

```
i+=1
```

C.

```
i=5;
```

```
while i:
```

```
    i-=1;print(i,end=' ')
```

B.

```
i='c'
```

```
while i in "uestc":
```

```
    print(i,end=' ')
```

D.

```
i=1;
```

```
while '0':
```

```
    print(i,end=' ');i+=1
```

```
    if i>5:continue
```

```
    if i>8:break
```

● 9. 下面程序段不是死循环的是 ( ) **C**

A.

```
i=1;
```

```
while i<3:
```

```
    print(i,end=' ')
```

```
i+=1
```

C.

```
i=5;
```

```
while i:
```

```
    i-=1;print(i,end=' ')
```

B.

```
i='c'
```

```
while i in "uestc":
```

```
    print(i,end=' ')
```

D.

```
i=1;
```

```
while '0':
```

```
    print(i,end=' ');i+=1
```

```
    if i>5:continue
```

```
    if i>8:break
```

● 10. 下面程序段中与其他三个输出不同的是 ( )

A.

```
for i in "123":  
    print(i,end=' ')
```

B.

```
for i in ['123']:  
    print(i,end=' ')
```

C.

```
for i in ('1','2','3'):  
    print(i,end=' ')
```

D.

```
for i in ('123'):  
    print(i,end=' ')
```

● 10. 下面程序段中与其他三个输出不同的是 ( ) **B**

A.

```
for i in "123":  
    print(i,end=' ')
```

B.

```
for i in ['123']:  
    print(i,end=' ')
```

C.

```
for i in ('1','2','3'):  
    print(i,end=' ')
```

D.

```
for i in ('123'):  
    print(i,end=' ')
```

- 二、填空题

- 1. 下面程序段的输出结果是（ ）。

- `for i in "uestc":print(i,end=' ')`

- 2. 下面程序段的输出结果是（ ）。

- `i=1;`

- `while i<5:`

- `i+=1`

- `if i<3:continue`

- `print(i,end=' ')`

- 二、填空题

- 1. 下面程序段的输出结果是 (    ) 。 **u e s t c**

- **for i in "uestc":print(i,end=' ')**

- 2. 下面程序段的输出结果是 (    ) 。 **3 4 5**

- **i=1;**

- **while i<5:**

- **i+=1**

- **if i<3:continue**

- **print(i,end=' ')**

### ● 三、问答题

- 1. 读入两个正整数，求其最小公倍数。
- 2. 一个数如果恰好等于它的因子之和，这个数就称为“完数”，例如6的因子为1、2、3，而 $6=1+2+3$ ，因此6是“完数”。编程找出1000以内的所有完数。



