



实用Python程序设计

郭 炜

微信公众号



微博: <http://weibo.com/guoweiofpku>

学会程序和算法，走遍天下都不怕!



北京大学
PEKING UNIVERSITY

信息科学技术学院

循环语句



北京大学
PEKING UNIVERSITY

信息科学技术学院 郭炜

for 循环语句



内蒙古阿斯哈图石林

循环语句

有时，需要重复多次执行一系列语句，因此需要
循环语句

for循环语句

```
for <variable> in <sequence>:  
    <statements 1>  
else:  
    <statements 2>
```

依次对 sequence 中的每个值,执行 <statements 1>
然后再执行 <statements 2>。通常不需要 else 部分

sequence 可以是range(..), 也可以是字符串、列表、元组、字典、集合

for循环语句

```
for i in range(5):    #[0,5)  
    print(i)
```

0

1

2

3

4

```
for i in range(5,9) : #[5,9)  
    print(i)
```

5

6

7

8

for循环语句

```
for i in range(0, 10, 3) :    #步长3  
    print(i)
```

0

3

6

9

```
for i in range(-10, -100, -30) :  
    print(i)
```

-10

-40

-70

for循环语句

```
for i in range(0):  
    print(i)
```

无输出

```
for i in range(2,2) :  
    print(i)
```

无输出


```
for i in range(3,12,3):  
    print(i,"",end = "")
```

输出结果是:

- ☐ A 3 6 9
- ☐ B 3 6 9 12
- ☐ C 3 4 5 6 7 8 9 10 11
- ☐ D 3 4 5 6 7 8 9 10 11 12

提交

```
for i in range(3,12,3):  
    print(i,"",end = "")
```

输出结果是:

- ☒ A 3 6 9
- ☐ B 3 6 9 12
- ☐ C 3 4 5 6 7 8 9 10 11
- ☐ D 3 4 5 6 7 8 9 10 11 12

提交

for循环遍历列表 – 写法1

```
a = ['Google', 'Baidu', 'IBM', 'Taobao', 'QQ']  
for i in range(len(a)):    # len,求列表长度（元素个数）  
    print(i, a[i])
```

```
0 Google  
1 Baidu  
2 IBM  
3 Taobao  
4 QQ
```

len也可以用来求字符串长度，元组、集合、字典元素个数

```
print(len("abc"))    #>>3
```

for循环遍历列表 – 写法2

```
a = ['Google', 'Baidu', 'IBM', 'Taobao', 'QQ']  
for i in a:  
    print(i)
```

Google

Baidu

IBM

Taobao

QQ

for循环遍历字符串

```
for letter in 'Taobao':  
    print (letter)
```

T
a
o
b
a
o

break 语句

```
sites = ["Baidu", "Google", "IBM", "Taobao"]
for site in sites:
    if site == "IBM":
        print("OK")
    print("site: " + site)
else:
    print("No break")
print("Done!")
```

site: Baidu
site: Google
OK
site: IBM
site: Taobao
No break
Done!

break 语句

```
sites = ["Baidu", "Google", "IBM", "Taobao"]    #list
for site in sites:    #对sites中的每个值 site
    if site == "IBM":
        print("OK")
        break    #跳出循环
    print("site: " + site)
else:
    print("No break")
print("Done!")
```

else子句在循环结束时会执行，但是如果break了，则不会执行

```
site: Baidu
site: Google
OK
Done!
```

continue语句

```
for letter in 'Taobao':  
    if letter == 'o':           # 字母为 o 时跳过输出  
        continue              # 直接跳到下次循环  
    print ('当前字母 :', letter)
```

当前字母 : T
当前字母 : a
当前字母 : b
当前字母 : a

连续输出26个字母

```
for i in range(26):  
    print(chr(ord("a") + i),end="")
```

abcdefghijklmnopqrstuvwxyz

字母的ASCII编码是连续的

字符的编码

ord(x) 求字符 x 的编码 （字符就是长度为1的字符串）

chr(x) 求编码为x的字符

可以 用8个连续的0或1（即1个字节）来表示一个字母、数字或标点符号，
比如用“00100000”表示空格，用“01100001”表示字母“a”，用
“01100010”表示字母“b”，用“01100011”表示字母“c”……。由8个0
或者1的组成的串，一共有 2^8 即256种不同的组合，这就足以表示10个阿拉伯
数字以及英语中用到的所有字母和标点符号了。此即为**ASCII**编码方案。

连续输出0 - 9

```
for i in range(10):  
    print(chr(ord("0") + i),end="")
```

0123456789

'0'-'9'的ASCII编码是连续的

```
print(chr(ord("A") + 4))
```

结果是

☐ A 101

☐ B 69

☐ C E

☐ D e

提交

```
print(chr(ord("A") + 4))
```

结果是

☐ A 101

☐ B 69

☒ C E

☐ D e

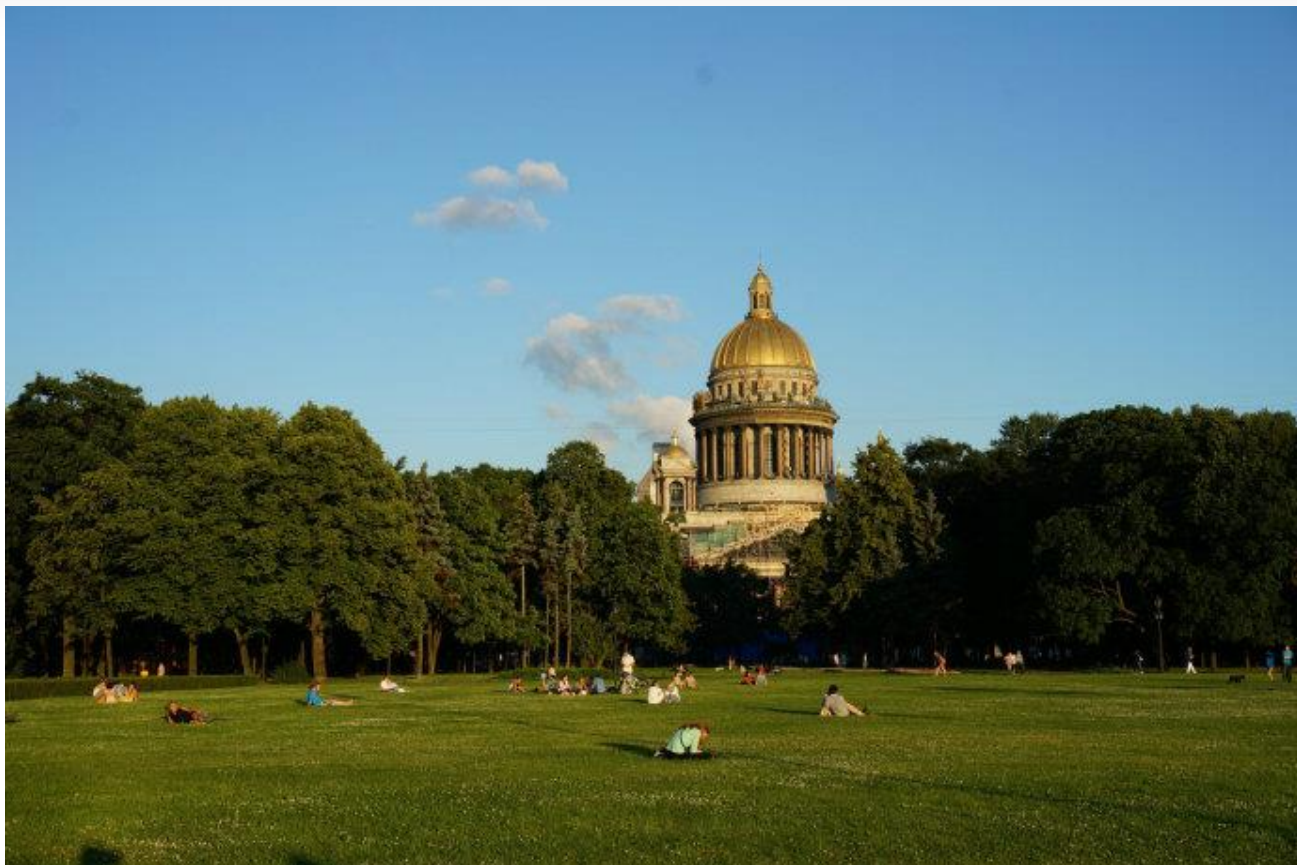
提交



北京大学
PEKING UNIVERSITY

信息科学技术学院 郭炜

for循环例题



俄罗斯圣彼得堡圣以撒教堂

例题1. 输入n个整数求和

输入

第一行是整数 n , $n \geq 1$

后面有 n 行, 每行一个整数

输出

输出后面那 n 个整数的和

样例输入

3

1

2

8

样例输出

11

例题1. 输入n个整数求和

```
n = int(input())
total = 0
for i in range(n):      #做n次
    total += int(input()) #每次读入一行
print(total)
```


例题2. 从小到大输出n的因子

➤输入一个正整数n，从小到大输出它的所有因子

```
n = int(input())  
for x in range(1,n+1):  
    if n % x == 0:  
        print(x," ",end="")
```

$\begin{array}{r} 15 \checkmark \\ \hline 1 \ 3 \ 5 \ 15 \end{array}$

例题3. 从大到小输出n的因子

➤输入一个正整数n，从大到小输出它的所有因子

```
n = int(input())  
for x in range(n, 0, -1): #步长-1  
    if n % x == 0:  
        print(x, " ", end="")
```

15	✓
15	5 3 1



北京大学
PEKING UNIVERSITY

信息科学技术学院 郭炜

多重循环



美国拱门国家公园

多重循环

➤循环可以嵌套，形成多重循环：

```
for i in range(n):  
    .....  
    for j in range(m):  
        ..... #内重循环的执行次数一共是 $n \times m$ 次
```

多重循环例题1：多次求n个数的和

输入

第一行是整数 m ， $m \geq 1$ ，表示有 m 组数据
接下来就是 m 组数据

对于每组数据：

第一行是整数 n ， $n \geq 1$

接下来是 n 行，每行一个整数

输出

对每组数据，输出后面那 n 个整数的和

样例输入

2

3

1

2

3

2

10

20

样例输出

6

30

多重循环例题1：多次求n个数的和

```
m = int(input())  
for i in range(m): #m组数据，所以要处理m次  
    n = int(input())  
    total = 0  
    for i in range(n): #n个数，每个一行，所以要input n次  
        total += int(input())  
    print(total)
```

```
total = 0
for i in range(2):
    for j in range(1,3):
        total += i + j
print(total)
```

输出结果是:

- ☐ A 6
- ☐ B 8
- ☐ C 10
- ☐ D 12

提交

```
total = 0
for i in range(2):
    for j in range(1,3):
        total += i + j
print(total)
```

输出结果是:

- ☐ A 6
- ☒ B 8
- ☐ C 10
- ☐ D 12

提交

多重循环例题2：取两个数

➤例题： 给定正整数 n 和 m , 在1至 n 这 n 个数中，取出两个不同的数，使得其和是 m 的因子，问有多少种不同的取法。输出这些取法。

多重循环例题2：取两个数

➤例题： 给定正整数 n 和 m , 在1至 n 这 n 个数中，取出两个不同的数，使得其和是 m 的因子，问有多少种不同的取法。输出这些取法。

思路：穷举1- n 这 n 个数中取两个数的所有取法，对每一种取法，判断其和是不是 m 的因子

多重循环例题3

►例题： 给定正整数 n 和 m , 在1至 n 这 n 个数中，取出两个不同的数，使得其和是 m 的因子，问有多少种不同的取法。输出这些取法。

思路： 穷举1- n 这 n 个数中取两个数的所有取法，对每一种取法，判断其和是不是 m 的因子

第一个数取1，第二个数分别取2, 3, ..., n

第一个数取2，第二个数分别取3, 4, ..., n

....

第一个数取 $n-2$ ，第二个数分别取 $n-1$, n

第一个数取 $n-1$ ，第二个数取 n

多重循环例题3

例题： 给定正整数 n 和 m , 在1至 n 这 n 个数中, 取出两个不同的数, 使得其和是 m 的因子, 问有多少种不同的取法。输出这些取法。

```
total = 0  #取法总数
lst = input().split()
n,m = int(lst[0]),int(lst[1])
for i in range(1,n):  #取第一个数i, 共n-1种取法
    for j in range(i+1,n+1):#第二个数要比第一个数大, 以免取法重复
        if m % (i + j) == 0:
            print(i,j)
            total += 1
print(total)
```

9 18✓

12

15

18

24

27

36

45

7

多重循环中的break

只会跳出当前那重循环，不会跳出多重循环

例题： 给定正整数 n 和 m , 在1至 n 这 n 个数中，取出两个不同的数 x, y ，使得 $x < y$ 且 $x+y$ 是 m 的因子。要求输出的数对里面， x 不重复，且 y 尽可能小。输出这些取法。

```
lst = input().split()
n,m = int(lst[0]),int(lst[1])
for i in range(1,n):    #取第一个数i，共n-1种取法
    for j in range(i+1,n+1):#第二个数要比第一个数大，以免取法重复
        if m % (i + j) == 0:
            print(i,j)
            break        #后面的j不用再取了，直接换下一个i
```

9	18	✓
1	2	
2	4	
3	6	
4	5	

多重循环中的continue

只会回到当前那重循环的开头



北京大学
PEKING UNIVERSITY

信息科学技术学院 郭炜

while 循环



荷兰阿姆斯特丹库肯霍夫公园

while循环语句

while 逻辑表达式 exp :

语句组1

else:

语句组2

.....

- 1) 判断 exp 是否为真，若为真，转2)，若为假，转3)
- 2) 执行 语句组1，回到 1)
- 3) 执行 语句组2
- 4) 继续往下执行

while循环语句

while 逻辑表达式 exp :
语句组1

.....

- 1) 判断 exp 是否为真，若为真，转2)，若为假，转3)
- 2) 执行 语句组1，回到 1)
- 3) 继续往下执行

while循环语句

```
count = 0
while count < 5:
    print (count, " 小于 5")
    count = count + 1
else:
    print (count, " 大于或等于 5")
```

0	小于 5
1	小于 5
2	小于 5
3	小于 5
4	小于 5
5	大于或等于 5

while循环语句

```
while True:
    .....
    if exp:
        break
    .....
```

不停执行，直到 exp为真时跳出循环

连续输出26个字母

```
i = 0
while i < 26:
    print(chr(ord("a") + i),end="")
    i+=1
```

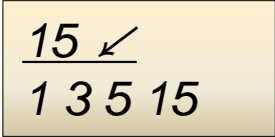
abcdefghijklmnopqrstuvwxyz

字母的ASCII编码是连续的

while循环语句示例1

➤例题：输入一个正整数n，从小到大输出它的所有因子

```
n = int(input())
x = 1
while x <= n:
    if n % x == 0:
        print(x, " ", end="")
    x += 1
```



15 ✓
1 3 5 15

while循环语句示例2

- 例题：提示用户输入密码，密码不正确则提示不正确，然后要求输入，密码正确则提示成功，然后结束。密码是pku

```
while (input("请输入密码:") != "pku") :  
    print("密码不正确！")  
print("密码输入成功!")
```

请输入密码:bba ✓
密码不正确！
请输入密码:std ✓
密码不正确！
请输入密码:pku ✓
密码输入成功!

while循环语句示例2

➤输入三个整数，求它们的最小公倍数

解法：枚举，一个个试

```
s = input().split()
x,y,z = int(s[0]),int(s[1]),int(s[2])
n = 1
while True:
    if n % x == 0 and n % y == 0 and n % z == 0:
        print(n)
        break
    n = n + 1
```

while循环语句示例2

➤输入三个整数，求它们的最小公倍数

解法：枚举，一个个试

```
s = input().split()
x,y,z = int(s[0]),int(s[1]),int(s[2])
n = 1
while not (n % x == 0 and n % y == 0 and n % z == 0):
    n += 1
print(n)
```


while循环语句示例2

➤输入三个整数，求它们的最小公倍数，改进解法

改进之处：即便是枚举，没必要试的，也不要试，这样速度才快

```
s = input().split()
x,y,z = int(s[0]),int(s[1]),int(s[2])
n = m = max(x,y,z) #从三者里面最大的开始试
while True:
    if n % x == 0 and n % y == 0 and n % z == 0:
        print(n)
        break
    n += m #没必要一个个试，而是每隔m个试一下 (还可进一步改进)
```



北京大学
PEKING UNIVERSITY

信息科学技术学院 郭炜

异常处理



锡林郭勒草原平顶山

用while语句和异常处理进行输入

在Openjudge做题时，有些题目，输入数据没有结束标志，也不告诉你有多少数据。
例如：

输入若干行，每行若干整数，求所有整数的最大值

输入样例：

15 23 9 3

87 6

3 4

输出样例：

87

如何判断输入结束，就是需要解决的问题

用while语句和异常处理进行输入

➤输入若干行，每行若干整数，求所有整数的最大值

```
s = input()
lst = s.split()
maxV = int(lst[0])
try: #异常处理
    while True:
        lst = s.split()
        for x in lst:
            maxV = max(maxV, int(x))
        s = input() #输入数据已经没有了还执行input, 会产生异常
except:
    pass #pass语句啥都不做
print(maxV)
```

15 23 6 ✓
15 5 3 1 ✓
3 4 ✓
Ctrl+Z ✓

异常处理

```
try:  
    <语句组1>  
except:  
    <语句组2>
```

如果在<语句组1>执行过程中出现了异常(runtime error)，程序立即从语句组1中跳出，去执行<语句组2>，然后再继续往下执行。如果<语句组1>执行正常完，则程序继续往下执行，不会执行<语句组2>

常见的异常有：

- 1) 不合适的转换，例如 `int("abc")` `int("23.34")` `float("abc")`
- 2) 输入已经结束（已经没有输入数据了）后，还执行 `input()` 在openjudge做题常见
- 3) 除法除数为0
- 4) 整数和字符串相加
- 5) 列表下标越界（比如3个元素的列表用到了下标4）

异常处理

```
try:  
    n = int(input())  
    print("hello")  
    a = 100/n  
    print(a)  
except:  
    print("error")  
print("end")
```

5 ✓

hello
20.0
end

0 ✓

hello
error
end

abc ✓

error
end



北京大学
PEKING UNIVERSITY

信息科学技术学院 郭炜

循环综合例题



旧金山九曲花街

例题1. 求斐波那契数列第k项

斐波那契数列是指这样的数列：数列的第一个和第二个数都为1，接下来每个数都等于前面2个数之和。给出一个正整数k，要求斐波那契数列中第k个数是多少。

输入：

输入一行，包含一个正整数k。 ($1 \leq k \leq 46$)

输出：

输出一行，包含一个正整数，表示斐波那契数列中第k个数的大小

样例输入

19

样例输出

4181

例题1. 求斐波那契数列第k项

解法：迭代，不停地由已知推未知

```
k = int(input())
if k == 1 or k == 2:
    print(1)
else:
    a1 = a2 = 1
    for i in range(k-2):
        a1, a2 = a2, a1+a2
    print(a2)
```

例题2. 求阶乘的和

给定正整数 n ，求不大于 n 的正整数的阶乘的和
(即求 $1!+2!+3!+\dots+n!$)

输入

输入有一行，包含一个正整数 n ($1 < n < 12$)。

输出

输出有一行：阶乘的和。

样例输入

5

样例输出

153

例题2. 求阶乘的和

解法1:

```
n = int(input())
s = 0
for i in range(1,n+1):
    f = 1 #存放i阶乘
    for j in range(1,i+1):
        f *= j #此操作一共做1+2+3+...+n次
    s += f
print(s)
```

重复计算多。比如算 $3!$ 时算了一遍 $1*2*3$ ，算 $4!$ 时又算一遍 $1*2*3$
改进： $1*2*3$ 只要算一遍就应该记下来，下次算 $4!$ 直接用它

例题2. 求阶乘的和

更快的解法2:

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

f值变化过程: $1*2$, $1*2*3$, $1*2*3*4$

例题3. 输入正整数 n ($n \geq 2$), 求不大于 n 的全部质数

解法1:

```
n = int(input())
for i in range(2,n+1): #每次判断i是否是质数
    ok = True         #开始假设i是质数
    for k in range(2,i):
        if i % k == 0:
            ok = False
            break
    if ok:
        print(i)
```

例题3. 输入正整数 n ($n \geq 2$), 求不大于 n 的全部质数

解法1:

```
n = int(input())
for i in range(2,n+1): #每次判断i是否是质数
    ok = True          #开始假设i是质数
    for k in range(2,i):
        if i % k == 0:
            ok = False
            break
    if ok:
        print(i)
```

#此解法做了没必要的尝试，偶数，以及 k 大于 i 的平方根后就不必再试

例题3. 输入正整数 n ($n \geq 2$), 求不大于 n 的全部质数
改进的解法2:

```
n = int(input())
print(2)
for i in range(3,n+1,2): #步长2, 只判断奇数
    ok = True
    for k in range(3,i,2):
        if i % k == 0:
            ok = False
            break
    if k*k > i: #大于根号i的数就不用试了
        break
    if ok:
        print(i)
```