python基础语法

输入和输出

输入

print()遇到逗号“，”会输出空格：

```
print('hello,world')
print('the quick brown fox','jumps over','the lazy dog')
```

```
D:\soft\Python\Python39\python.exe "E:/OneDrive - cumt.edu.cn/program/python/test/test.py"
the quick brown fox jumps over the lazy dog

进程已结束，退出代码为 0
```

print()打印整数，计算结果：

```
print(300)
print(100+100)
print('100 + 200 =',100+200)
```

```
D:\soft\Python\Python39\python.exe "E:/OneDrive - cumt.edu.cn/program/python/test/test.py"
300
200
100 + 200 = 300

进程已结束，退出代码为 0
```

输出

input()函数：

```
x=input()
print(x)
y=input('请输入y：')
print(y)
```

```
D:\soft\Python\Python39\python.exe
2
2
请输入y：6
6

进程已结束，退出代码为 0
```

练习

```
#请利用print()输出1024 * 768 = xxx
print('1024 * 768 =',1024*768)
```

```
D:\soft\Python\Python39\python.exe "E:/OneD
1024 * 768 = 786432

进程已结束，退出代码为 0
```

数据类型和变量

数据类型：整数，浮点数，字符串，布尔值，空值（None），变量，常量。

字符串：

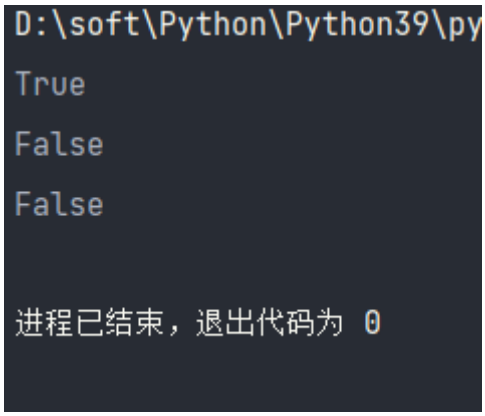
```
print("I'm ok")
print("I'm \"ok\"!")
print('I\'m "ok"!')
print('I\'m \nok!')
print('''abcdefg
abcdefg
abcdefg''')
```

```
D:\soft\Python\Python39\python.exe "E:/Or
I'm ok
I'm "ok"!
I'm "ok"!
I'm
ok!
abcdefg
abcdefg
abcdefg

进程已结束，退出代码为 0
```

布尔值：

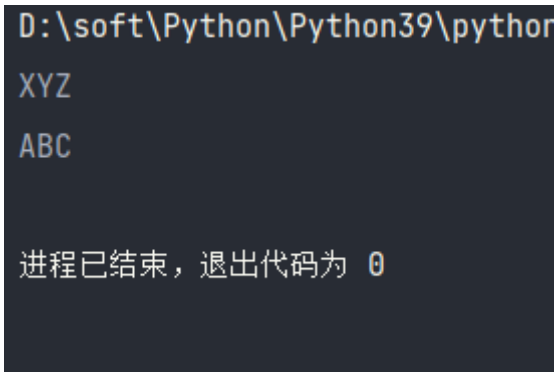
```
print(True or False)
print(True and False)
print(not True)
```



变量：

变量名必须是大小写英文、数字和_的组合，且不能用数字开头。

```
a = 'ABC'
b = a    #b指向的是与a指向相同的字符串，并非a
a = 'XYZ'
print(a)
print(b)
```

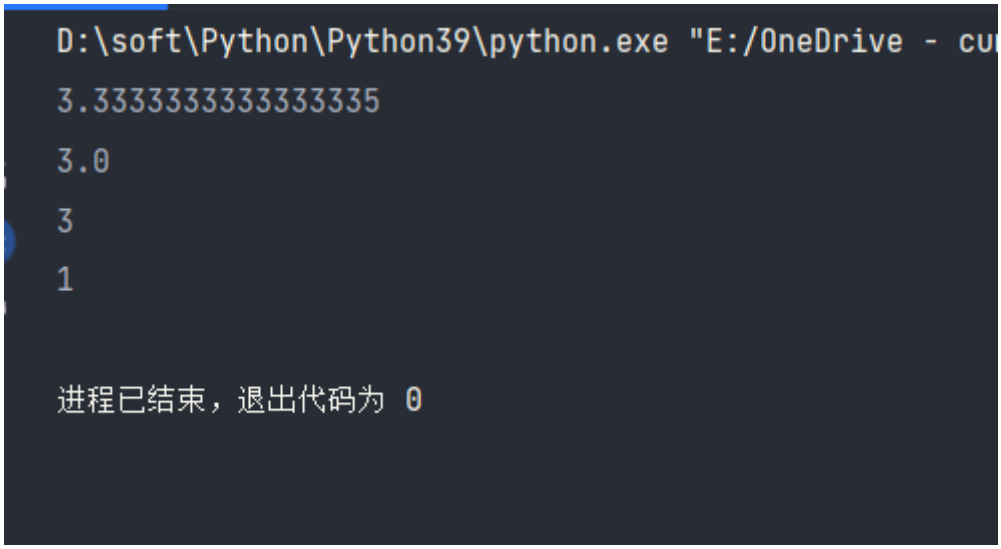


即：对变量赋值 `x = y` 是把变量 `x` 指向真正的对象，该对象是变量 `y` 所指向的。随后对变量 `y` 的赋值不影响变量 `x` 的指向。

常量：

在Python中，通常用全部大写的变量名表示常量。python没有机制保证常量不被改变，只是习惯。

```
print(10/3) #计算结果为浮点数
print(9/3)
print(10//3) #计算结果为整数
print(10%3) #取余
```



练习：

```
'''
请打印出以下变量的值：
# -*- coding: utf-8 -*-
n = 123
f = 456.789
s1 = 'Hello, world'
s2 = 'Hello, \'Adam\''
s3 = r'Hello, "Bart"'
s4 = r'''Hello,
Lisa!'''
'''
n = 123
f = 456.789
s1 = 'Hello, world'
s2 = 'Hello, \'Adam\''
s3 = r'Hello, "Bart"'
s4 = r'''Hello,Lisa!'''
print(n)
print(f)
print(s1)
print(s2)
print(s3)
print(s4)
```

```
D:\soft\Python\Python39\python.exe "E:/OneDrive - cumt.e
123
456.789
Hello, world
Hello, 'Adam'
Hello, "Bart"
Hello,Lisa!

进程已结束，退出代码为 0
```

字符串和编码

字符串

对于单个字符的编码，Python提供了`ord()`函数获取字符的整数表示，`chr()`函数把编码转换为对应的字符

```
print(ord('A'))
print(chr(66))
print(ord('abc'))#错误单个字符编码
```

```
D:\soft\Python\Python39\python.exe "E:/OneDrive - cumt.edu.cn/program/python/test/
Traceback (most recent call last):
  File "E:\OneDrive - cumt.edu.cn\program\python\test\test.py", line 3, in <module>
    print(ord('abc'))
TypeError: ord() expected a character, but string of length 3 found
65
B

进程已结束，退出代码为 1
```

统计字符个数函数：`len()`

格式化：

占位符	替换内容
%d	整数
%f	浮点数
%s	字符串
%x	十六进制整数

练习：

```
'''
小明的成绩从去年的72分提升到了今年的85分，请计算小明成绩提升的百分点，并用字符串格式化显示出'xx.x%'，只保留小数点后1位：
# -*- coding: utf-8 -*-
s1 = 72
s2 = 85
'''
s1 = 72
s2 = 85
r=(s2-s1)/s1
print('小明成绩提升了%f',r)
```

```
D:\soft\Python\Python39\python.exe "E:/OneDrive - cumt.edu.cn/program/python/test/test.py"
小明成绩提升了%f 0.18055555555555555

进程已结束，退出代码为 0
```

list和tuple

list

在Python中，用方括号（[] ）表示列表，并用逗号分隔其中的元素。

```
bicycles=['trek','cannondale','redline','specialized']
print(bicycles)
print(bicycles[0])#访问列表元素
print(bicycles[0].title())#首字母大写的元素
```

```
D:\soft\Python\Python39\python.exe "E:/OneDrive - cumt.edu.cn/program/python/test/test.py"
['trek', 'cannondale', 'redline', 'specialized']
trek
Trek

进程已结束，退出代码为 0
```

Python为访问最后一个列表元素提供了一种特殊语法。通过将索引指定为-1 ,可让Python返回最后一个列表元素。

list常用函数

```
classmates = ['Michael', 'Bob', 'Tracy']
print(classmates)
classmates.append('Adam')#在末尾追加元素
print(classmates)
classmates.insert(1,'Jack')#在索引号为1处插入元素
print(classmates)
classmates.pop()#删除末尾元素
print(classmates)
classmates.pop(1)#删除指定位置元素
print(classmates)
del classmates[0]
print(classmates)#删除指定位置元素
```

```
#如果你要从列表中删除一个元素，且不再以任何方式使用它，就使用del 语句；如果你要在删除元素后还能继续使用它，就使用方法pop()
classmates[0]='Sarah'#替换元素
print(classmates)
classmates.remove('Sarah')#知元素不知位置的删除
print(classmates)
```

```
D:\soft\Python\Python39\python.exe "E:/OneDrive - cumt.edu.cn/program/pytho
['Michael', 'Bob', 'Tracy']
['Michael', 'Bob', 'Tracy', 'Adam']
['Michael', 'Jack', 'Bob', 'Tracy', 'Adam']
['Michael', 'Jack', 'Bob', 'Tracy']
['Michael', 'Bob', 'Tracy']
['Bob', 'Tracy']
['Sarah', 'Tracy']
['Tracy']

进程已结束，退出代码为 0
```

```
cars = ['bmw', 'audi', 'toyota', 'subaru']
cars.sort()#按字母顺序对元素进行排序
print(cars)
cars.sort(reverse=True)#字母反向排序
print(cars)
cars = ['bmw', 'audi', 'toyota', 'subaru']
print(sorted(cars))#对列表临时排序，不修改列表
print(cars)
print(len(cars))#统计表元素个数，从1开始！！！
```

```
D:\soft\Python\Python39\python.exe "E:/OneDrive - cu
['audi', 'bmw', 'subaru', 'toyota']
['toyota', 'subaru', 'bmw', 'audi']
['audi', 'bmw', 'subaru', 'toyota']
['bmw', 'audi', 'toyota', 'subaru']
4

进程已结束，退出代码为 0
```

数值列表

```python
#打印1到4
for value in range(1,5):
    print(value)
numbers=list(range(1,6))
print(numbers)
numbers=list(range(1,11,2))#指定步长
print(numbers)
```

```
D:\soft\Python\Python39\python.exe "E:/OneDrive - cumt.edu
1
2
3
4
[1, 2, 3, 4, 5]
[1, 3, 5, 7, 9]

进程已结束，退出代码为 0
```

统计计算：

```python
digits=[1,2,3,4,5,6,7,8]
print(min(digits))
print(max(digits))
print(sum(digits))
```

切片

处理列表的部分元素， Python称之为切片。

```python
players = ['charles', 'martina', 'michael', 'florence', 'eli']
print(players[0:3])
print(players[1:4])
print(players[:4])#从头开始提取
print(players[2:])#省略终止索引
print(players[-3:])#打印最后三个
for player in players[:3]:
    print(player.title())#遍历方法
```

```
D:\soft\Python\Python39\python.exe "E:/OneDrive - cumt.edu.cn/program/python/te
['charles', 'martina', 'michael']
['martina', 'michael', 'florence']
['charles', 'martina', 'michael', 'florence']
['michael', 'florence', 'eli']
['michael', 'florence', 'eli']
Charles
Martina
Michael

进程已结束，退出代码为 0
```

```
my_foods=['pizza','falafel','carrot cake']
friend_foods=my_foods[:]
print("My foods are:",my_foods)
print("My friend's foods are:",friend_foods)
```

```
D:\soft\Python\Python39\python.exe "E:/OneDrive - cumt.edu.cn/program/python/test
My foods are: ['pizza', 'falafel', 'carrot cake']
My friend's foods are: ['pizza', 'falafel', 'carrot cake']

进程已结束，退出代码为 0
```

注意：若`friend_foods = my_foods`两个变量相关了，而非复制的副本。

tuple

tuple和list非常类似，但是tuple一旦初始化就不能修改。

tuple定义`t=(1)`错误，`t=(1,)`正确

练习

```
#请用索引取出下面list的指定元素：
'''# 打印Apple:
print(?)
# 打印Python:
print(?)
# 打印Lisa:
print(?)'''
# -*- coding: utf-8 -*-
L = [
    ['Apple', 'Google', 'Microsoft'],
    ['Java', 'Python', 'Ruby', 'PHP'],
    ['Adam', 'Bart', 'Lisa']
]
print(L[0][0])
print(L[1][1])
print(L[2][2])
```

```
D:\soft\Python\Python39\python.exe "E:/OneDrive - cumt.edu.cn/program/pytho
Apple
Python
Lisa

进程已结束，退出代码为 0
```

if语句

练习

```
'''小明身高1.75，体重80.5kg。请根据BMI公式（体重除以身高的平方）帮小明计算他的BMI指数，并根据BMI指数：

低于18.5：过轻
18.5-25：正常
25-28：过重
28-32：肥胖
高于32：严重肥胖
用if-elif判断并打印结果：'''
BMI=int(80.5/(1.75*1.75))
if (BMI<18.5):
    print('过轻')
elif(BMI>=18.5 and BMI<25):
    print('正常')
elif(BMI>=25 and BMI<28):
    print('过重')
elif(BMI>=28 and BMI<32):
    print('肥胖')
else:
    print('严重肥胖')
```

while循环

要特别注意，不要滥用 `break` 和 `continue` 语句。`break` 和 `continue` 会造成代码执行逻辑分叉过多，容易出错。

dict

字典是一系列键值对。每个键 都与一个值相关联，你可使用键来访问相关联的值。与键相关联的值可以是数、字符串、列表乃至字典。

```
alien_0={}
alien_0['color']='green'
alien_0['points']=5
print(alien_0)
```

```
D:\soft\Python\Python39\python.exe "E:/OneDrive - cumt.edu.cn/program/python/test/test.py"
{'color': 'green', 'points': 5}

进程已结束，退出代码为 0
```

常用函数

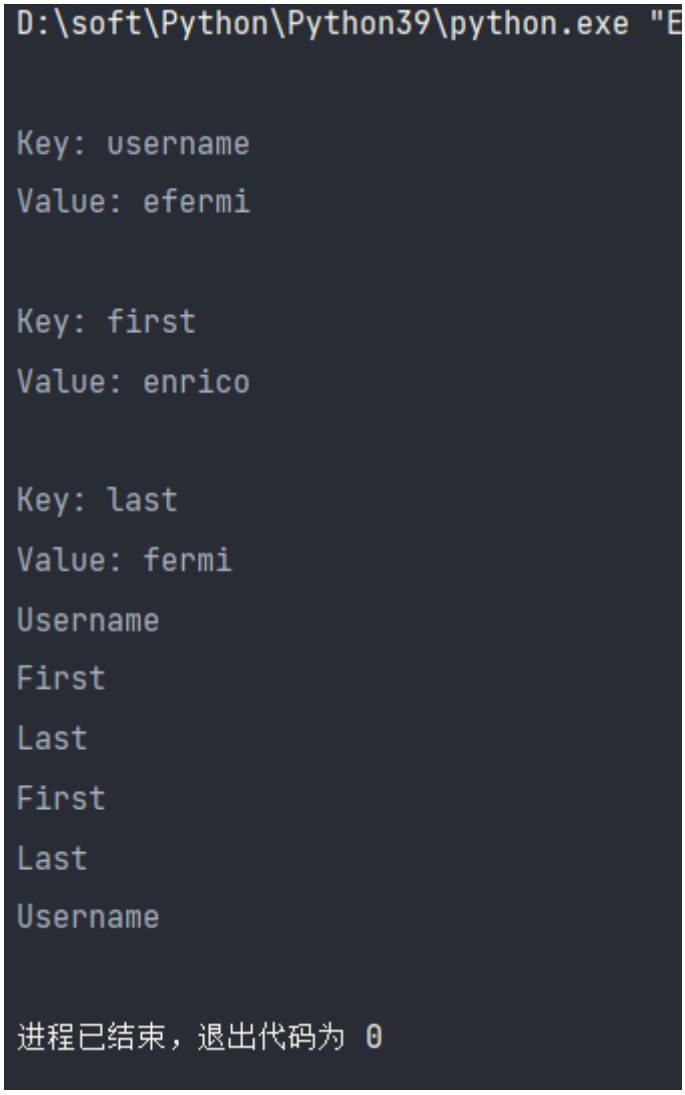
```
alien_0={}
alien_0['color']='green'
alien_0['points']=5
alien_0['speed']='slow'
del alien_0['points']
print(alien_0)
print_value=alien_0.get('points','No point value assigned')#在指定的键不存在时返回一个默认值，从而避免这样的错误。
print(print_value)
```

```
D:\soft\Python\Python39\python.exe "E:/OneDrive - cumt.edu.cn/program/python/test/test.py"
{'color': 'green', 'speed': 'slow'}
No point value assigned

进程已结束，退出代码为 0
```

字典遍历

```
user_0={
    'username':'efermi',
    'first':'enrico',
    'last':'fermi',
}
for k,v in user_0.items():
    print(f"\nKey: {k}")
    print(f"Value: {v}")
for v in user_0.keys():
    print(v.title())#打印值
for v in sorted(user_0.keys()):
    print(v.title())#排好序
```



注：python的print字符串前面加f表示格式化字符串，加f后可以在字符串里面使用用花括号括起来的变量和表达式，如果字符串里面没有表达式，那么前面加不加f输出应该都一样。

set

set和dict类似，也是一组key的集合，但不存储value。由于key不能重复，所以，在set中，没有重复的key。

```
s1=set([1,2,3])
print(s1)
s2=set([1,1,2,2,3,3])
print(s2)
s1.add(4)#追加
print(s1)
s1.remove(4)#移出
print(s1)
s3=set([1,2,3])
s4=set([3,4,5])
print(s3 & s4)#取并
print(s3|s4)#取交
```

```
D:\soft\Python\Python39\python.exe "E:/OneDrive - cumt.edu.cn/program/python/test/t
{1, 2, 3}
{1, 2, 3}
{1, 2, 3, 4}
{1, 2, 3}
{3}
{1, 2, 3, 4, 5}

进程已结束，退出代码为 0
```

调用简单函数

```
print(abs(100))#绝对值函数
print(abs(-100))
print(max(1,2,4,5))#求最大值函数
print(min(-1,3,5))#求最小值函数
a=abs
print(a(-1))#a为函数别名
```

```
D:\soft\Python\Python39\python.exe "E:/OneDrive - cumt.edu.cn/p
100
100
5
-1
1

进程已结束，退出代码为 0
```

递归函数

斐波那契:

```
def fact(n):
    if n==1:
        return 1
    return n*fact(n-1)
print(fact(5))
```

```
D:\soft\Python\Python39\python.exe "E:/OneDrive - cumt.edu.cn/program/python/te
120

进程已结束，退出代码为 0
```

汉诺塔:

```python
def move(n,a,b,c):
    if n==1:
        print(a,'-->',c)
    else:
        move(n-1,a,c,b)
        print(a,'-->',c)
        move(n-1,b,a,c)
move(3,'A','B','C')
```

```
D:\soft\Python\Python39\py
A --> C
A --> B
C --> B
A --> C
B --> A
B --> C
A --> C

进程已结束，退出代码为 0
```

高级特性

列表生成式：

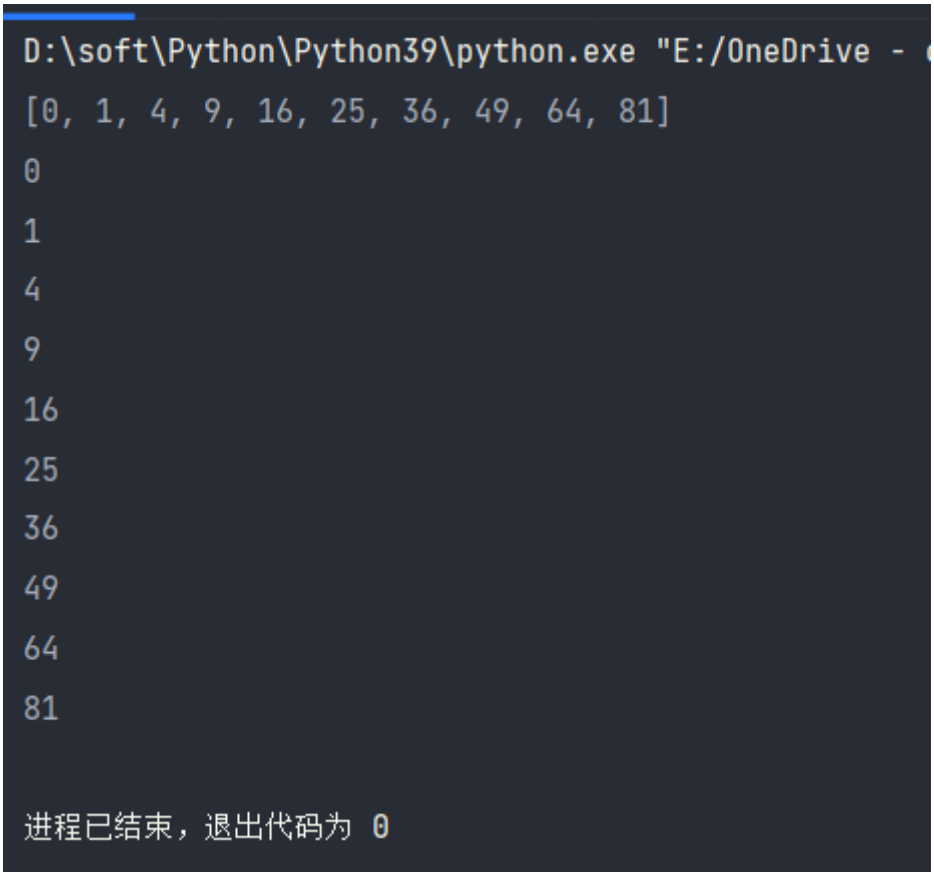
```python
L1=list(range(1,11))#生成包含1到10的列表
L2=[x*x for x in range(1,11)]#创建特殊规律列表
L3=[m+n for m in'ABC' for n in 'XYZ']
print(L1,L2,L3)
```

```
D:\soft\Python\Python39\python.exe "E:/OneDrive - cumt.edu.cn/program/python/test/test.py"
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10] [1, 4, 9, 16, 25, 36, 49, 64, 81, 100] ['AX', 'AY', 'AZ', 'BX', 'BY', 'BZ', 'CX', 'CY', 'CZ']

进程已结束，退出代码为 0
```

生成器：在Python中，这种一边循环一边计算的机制，称为生成器：generator。

```python
L=[x*x for x in range(10)]#创建列表
G=(x*x for x in range(10))#创建生成器
print(L)
for n in G:
    print(n)
```



迭代器：

凡是可作用于`for`循环的对象都是`Iterable`类型；

凡是可作用于`next()`函数的对象都是`Iterator`类型，它们表示一个惰性计算的序列；

集合数据类型如`list`、`dict`、`str`等是`Iterable`但不是`Iterator`，不过可以通过`iter()`函数获得一个`Iterator`对象。

高阶函数

变量可以指向函数，函数名也是变量

一个函数可以接收另一个函数作为参数，这种函数称之为高阶函数。

```
def add(x,y,f):
    return f(x)+f(y)
```

map/reduce/filter

`map()`函数接收两个参数，一个是函数，一个是`Iterable`，`map`将传入的函数依次作用到序列的每个元素，并把结果作为新的`Iterator`返回。

```
def f(x):
    return x * x
r=map(f,[1,2,3])
print(list(r))
```

```
D:\soft\Python\Python39\python.exe "E:/OneDrive - cumt.edu.cn/program/python/test/test.py"
[1, 4, 9]

进程已结束，退出代码为 0
```

`reduce`把一个函数作用在一个序列`[x1, x2, x3, ...]`上，这个函数必须接收两个参数，`reduce`把结果继续和序列的下一个元素做累积计算:

```
reduce(f,[x1,x2,x3,x4])=f(f(f(x1,x2),x3),x4)
```

```
#序列求和
from functools import reduce
def add(x,y):
    return x+y
print(reduce(add,[1,3,5,7,9]))
```

```
D:\soft\Python\Python39\python.exe "E:/OneDrive - cumt.edu.cn/program/python/test/test.py"
25

进程已结束，退出代码为 0
```

`filter()`把传入的函数依次作用于每个元素，然后根据返回值是`True`还是`False`决定保留还是丢弃该元素。

```
def odd(n):
    return n%2==1
L=list(filter(add,[1,2,3,4,5,6,10,15]))
print(L)
```

```
D:\soft\Python\Python39\python.exe "E:/OneDrive - cumt.edu.
[1, 3, 5, 15]

进程已结束，退出代码为 0
```

`sorted()`函数就可以对list进行排序,`sorted()`函数也是一个高阶函数，它还可以接收一个`key`函数来实现自定义的排序。要进行反向排序，不必改动key函数，可以传入第三个参数`reverse=True`

关键字`lambda`表示匿名函数，冒号前面的`x`表示函数参数。匿名函数有个限制，就是只能有一个表达式，不用写`return`，返回值就是该表达式的结果。用匿名函数有个好处，因为函数没有名字，不必担心函数名冲突。此外，匿名函数也是一个函数对象，也可以把匿名函数赋值给一个变量，再利用变量来调用该函数。

导入模块：import