

【Python】运算符

Python运算符优先级

以下表格列出了从最高到最低优先级的所有运算符：

运算符	描述
**	指数 (最高优先级)
~ + -	按位翻转，一元加号和减号 (最后两个的方法名为 +@ 和 -@)
* / % //	乘，除，取模和取整除
+ -	加法减法
>> <<	右移，左移运算符
&	位 'AND'
^	位运算符
<= < > >=	比较运算符
<> == !=	等于运算符
= %= /= //= -= += *= **=	赋值运算符
is is not	身份运算符
in not in	成员运算符
not and or	逻辑运算符

CSDN @fftx_00

一、 +

```
1  ##+运算符除了用于算术加法以外
2  #还可以用于列表、元组、字符串的连接
3
4  print(3 + 5)
5  print(3.4 + 4.5)
6  print((3+4j) + (5+6j))
7  print('abc' + 'def')
8  print([1,2] + [3,4])
9  print((1,2) + (3,))
```

二、 -

```
1  #-运算符除了用于整数、实数、复数之间的算术减法和相反数之外(注意浮点误差)
2  #还可以计算集合的差集。
3
4  print(7.9 - 4.5)          # 注意，结果有误差
5  print(5 - 3)
6  num = 3
7  print(-num)
8  print(--num)              # 注意，这里的--是两个负号，负负得正
9  print(-(-num))           # 与上一行代码含义相同
10 print({1,2,3} - {3,4,5})  # 计算差集
11 print({3,4,5} - {1,2,3})
```

三、 *

```

1  #*运算符除了表示整数、实数、复数之间的算术乘法
2  #还可用于列表、元组、字符串这几个类型的对象与整数的乘法，表示序列元素的重复，
3  #以生成新的列表、元组或字符串
4
5  print(33333 * 55555)
6  print((3+4j) * (5+6j))
7  print('重要的事情说三遍!' * 3)
8  print([0] * 5)
9  print((0,) * 3)

```

四、/, //

```

1  #运算符/和//在Python中分别表示真除法和求整商
2  #注意整除运算符//“向下取整”
3  #例如，-17 / 4的结果是-4.25，在数轴上小于-4.25的最大整数是-5，所以-17 // 4的结果是-5
4
5  print(17 / 4)
6  print(17 // 4)
7  print((-17) / 4)
8  print((-17) // 4)

```

五、%

```

1  # %运算符可以用于求余数运算，还可以用于字符串格式化
2  # 在计算余数时，结果与%右侧的运算数符号一致
3
4  print(365 % 7)
5  print(365 % 2)
6  print('%c,%c,%c' % (65, 97, 48))  # 把65、97、48格式化为字符
7  print(365 % -2)                    # -1

```

六、**

```

1  # **运算符表示幂运算。
2  # 该运算符具有右结合性
3  # 如果有两个连续的**运算符，那么先计算右边的再计算左边的
4
5  print(2 ** 4)
6  print(3 ** 3 ** 3)
7  print(3 ** (3**3))          # 与上一行代码含义相同
8  print((3**3) ** 3)          # 使用圆括号修改计算顺序
9  print(9 ** 0.5)              # 计算9的平方根
10 print((-1) ** 0.5)           # 对复数计算平方根得到复数

```

七、关系运算符 > < == >= <=

```

1  # 关系运算符
2
3  # 整数
4  print(3+2 < 7+8)              # 关系运算符优先级低于算术运算符
5  print(3 < 5 > 2)              # 等价于3<5 and 5>2
6  print(3 == 3 < 5)            # 等价于3==3 and 3<5
7
8  # 字符串
9  print('12345' > '23456')      # 第一个字符'1'<'2'，直接得出结论
10 print('abcd' > 'Abcd')        # 第一个字符'a'>'A'，直接得出结论
11
12 # 列表
13 print([85, 92, 73, 84] < [91, 82, 73]) # 第一个数字85<91，直接得出结论
14 print([180, 90, 101] > [180, 90, 99])  # 前两个数字相等，第三个数字101>99
15 print([1, 2, 3, 4] > [1, 2, 3])        # 前三个元素相等,且第一个列表有多余的元素
16
17 # 集合(比较包容性,是否超集or子集or完全相同)

```

```
18 | print({1, 2, 3, 4} > {3, 4, 5})           # 第一个集合不是第二个集合的超集19 |
    print({1, 2, 3, 4} <= {3, 4, 5})        # 第一个集合不是第二个集合的子集20 |
```

八、成员测试运算符 in, not in

```
1 | # 成员测试运算符in, not in
2 |
3 | print(60 in [70, 60, 50, 80])           # T
4 | print('abc' in 'a1b2c3d4fg')          # F
5 | print([3] in [[3], [4], [5]])         # T
6 | print('3' in map(str, range(5)))       # T
7 | print(5 in range(5))                  # F
8 | print(5 not in range(5))              # T
```

九、集合运算符 | & - ^

```
1 | # 集合运算符
2 |
3 | A = {35, 45, 55, 65, 75}
4 | B = {65, 75, 85, 95}
5 |
6 |
7 | print(A | B)                          # 并集
8 | print(A & B)                          # 交集
9 | print(A - B)                          # 差集
10 | print(B - A)                         # 差集
11 | print(A ^ B)                         # 并集-交集
```

十、逻辑运算符 and or not

运算符	逻辑表达式	描述
and	x and y	布尔"与" - 如果 x 为 False，x and y 返回 False，否则它返回 y 的计算值。
or	x or y	布尔"或" - 如果 x 是非 0，它返回 x 的计算值，否则它返回 y 的计算值。
not	not x	布尔"非" - 如果 x 为 True，返回 False 。如果 x 为 False，它返回 True。

上面这么说并不准确：

x and y: 如果x等价于False，则返回x；否则返回y
x or y: 如果x等价于True，则返回x；否则返回y
not x: 如果x等价于True，则返回False；
 如果x等价于False，返回True

```
>>> -1 and 1
1
>>> 0 and 1
0
>>> 6 and -1
-1
>>> {} and 1
{}
>>> [1] or 0
[1]
>>> False or [1]
[1]
>>> {} or 0
0
>>> not {}
True
>>> 3-3 or 5-2
3
```

十一、身份运算符 is

运算符	描述	实例
is	is 是判断两个标识符是不是引用自一个对象	x is y , 类似 id(x) == id(y) , 如果引用的是同一个对象则返回 True, 否则返回 False
is not	is not 是判断两个标识符是不是引用自不同对象	x is not y , 类似 id(a) != id(b) 。如果引用的不是同一个对象则返回结果 True, 否则返回 False。 <div>CSDN @fftx_00</div>

```
1 # 身份运算符
2 a = 20
3 b = 20
4 c = 10
5
6 print(a is b)           # T
7 print(a is c)           # F
8 print(b is not c)       # T
```