

Python基础 运算符

算数运算符

```
# + 加法
print("2 加 5等于{}".format(2 + 5))

# - 减法
print("6减3等于{}".format(6 - 3))

# * 乘法
print(" 2 乘以 4 等于{}".format(2 * 4))

# / 除法
print(" 4 除以 2 等于{}".format(4 / 2))

# % 求余, 即返回除法的余数
print(" 4 除以 3的余数等于{}".format(4 % 3))

# // 取整除, 即返回商的整数部分
print(" 12 乘以 5的商 等于{}".format(12 // 5))

# ** 幂, 即返回x的y次方
print(" 2的4次方等于{}".format(2 ** 4))
```

```
分享/day03')
2 加 5等于7
6减3等于3
 2 乘以 4 等于8
 4 除以 2 等于2.0
 4 除以 3的余数等于1
12 乘以 5的商 等于2
2的4次方等于16
```

在Python中进行数学计算时, 与我们数学中的优先级是一致的。

- 先乘除后加减
- 同级运算从左到右
- 可以使用 () 调整优先级

算数优先级从高到低如下:

```
第一级: **
第二级: *, /, %, //
第三级: +, -
```

在Python中* 号可以和字符串进行运算

```
# * 可以和字符串进行运算
print("20个加号:{}".format"+" * 20))
```

20个加号:+++++

赋值运算符

```
# -*- coding: utf-8 -*-
"""
Created on Sat Dec 29 21:50:00 2018

@author: Tango
"""
```

= 简单的赋值运算符

```
x = 20
y = x
print("x:",x)
print("y:",y)
print("-" * 30)
# += 加赋值 x+=y 等价于 x= x +y
x += y
print("x:",x)
print("y:",y)
print("x+=y:",x)
```

```
print("-" * 30)
# -= 减赋值 x-=y 等价于 x=x -y
x = 20
y = 5
x-=y
print("x:",x)
print("y:",y)
print("x-=y:",x)
```

```
print("-" * 30)
# *= 乘赋值 x*=y 等价于 x = x * y
x = 20
y = 5
x*=y
print("x:",x)
print("y:",y)
print("x*=y:",x)
```

```
print("-" * 30)
# /= 出赋值, x/=y 等价于 x = x/y
x = 20
y = 5
x/=y
print("x:",x)
print("y:",y)
```

```
print("x/=y:",x) #注意Python运行除法的结果实浮点型
```

```
#其他运算也一样，大家试试 % =, **= /=
```

```
In [19]: runfile('F:/笔记相关/Python基础：
二次分享/day03')
x: 20
y: 20
-----
x: 40
y: 20
x+=y: 40
-----
x: 15
y: 5
x-=y: 15
-----
x: 100
y: 5
x*=y: 100
-----
x: 4.0
y: 5
x/=y: 4.0
. .
```

比较（关系）运算符

```
# -*- coding: utf-8 -*-
"""
```

```
Created on Sat Dec 29 22:01:57 2018
```

```
@author: Tango
```

```
比较运算符
"""
```

```
# > 大于
```

```
print("3 > 1 :", 3 > 1)
print("-" * 30)
```

```
# < 小于
```

```
print("1 < 11 :", 1 < 11)
print("-" * 30)
```

```
# == 等于 注意是两个等号
```

```
print("1 == 11 :", 1 == 11)
print("-" * 30)
```

```
# != 不等于
```

```
print("1 != 11 :", 1 != 11)
print("-" * 30)
```

```
# >= 大于等于
```

```
print("11 >= 11 :", 11 >= 11)
print("-" * 30)
```

```
# <= 小于等于
print("11 <= 11 :", 11 <= 11)
print("-" * 30)
```

```
3 > 1 : True
-----
1 < 11 : True
-----
1 == 11 : False
-----
1 != 11 : True
-----
11 >= 11 : True
-----
11 <= 11 : True
-----
```

逻辑运算符

```
# -*- coding: utf-8 -*-
"""
Created on Sat Dec 29 22:07:36 2018

@author: Tango
逻辑运算符
"""

# and 逻辑与 两边同时为真则为真
print("1 < 2 and 2 >0:", 1 < 2 and 2 > 0)
print("1 < 0 and 2 >0:", 1 < 0 and 2 > 0)
print("-" * 30)

# or 逻辑或 两边有一个为真则为真
print("1 < 2 or 2 >0:", 1 < 2 or 2 > 0)
print("1 < 0 or 2 >0:", 1 < 0 or 2 > 0)

print("-" * 30)
# not 逻辑非, 取反
print("not 1 < 2:{}".format(not 1 < 2))
```

```
二次分享/day03')
1 < 2 and 2 >0: True
1 < 0 and 2 >0: False
-----
1 < 2 or 2 >0: True
1 < 0 or 2 >0: True
-----
not 1 < 2:False
```