1. **Python的运算符和表达式**

**第一部分**

**一、运算符**

运算符用于执行程序代码运算，会针对一个以上操作数项目来进行运算。例如：2+3，其操作数是2和3，而运算符则是“+”。

Python运算符包括算术运算符、关系运算符、逻辑运算符、赋值运算符。

**1.1 算术运算符**

复习数学的+,-,×,÷,取商(取整),取余(取模),

|  |  |
| --- | --- |
| 表达式 | 说明 |
| x+y | 加法运算 |
| x-y | 减法运算 |
| x\*y | 乘法运算 |
| X/y | 除法运算 |
| X%y | 求模运算 |
| X\*\*y | X的y次方 |
| x//y | 两数相除向下取整 |

x = 5

y = 3

a = 4

b = 2

print(x + y) #结果为 7

print(x - y) #结果为2

print(x \* y) #结果为15

print(x / y) #结果为1.6666666666666667 不同的机器浮点数的结果可能不同

print(x // y) #向下去整结果为1

print(x % y) #两数相除取余结果为2

print(x\*\*y) #5的3次幂结果为125

print(a / b) #结果为浮点数2.0

print(a % b)#取余结果为0

print(a // b)#取整结果为2

**1.2 关系运算符**

布尔类型，True或者False

关系运算符是对两个对象进行比较，返回值为布尔类型，True或者False

|  |  |  |
| --- | --- | --- |
| 运算符 | 表达式 | 说明 |
| == | a==b | 等于，比较对象是否相等 |
| != | a !=b | 不等于，比较两个对象是否不相等  <>2.7版本可用，3.6版本不再支持。 |
| > | a > b | 大于，比较a是否大于b |
| < | a < b | 小于，比较a是否小于b |
| >= | a>=b | 大于等于，比较a是否大于或者等于b |
| <= | a<=b | 小于等于，比较a是否小于或者等于b |

a = 4

b = 2

c = 2

print(a == b) #False

print(a != b) #True

print(a <> b) #True

print(a > b) #True

print(a < b) #False

print(a >= b) #True

print(c <= b) #True

**1.3 逻辑运算符**

***生活案例：明*天是周末 and 天气为晴 出去游玩**

**数学考试得分85分以上or 英语考试得分90以上，奖励一台笔记本**

**议会做出添加投入失业补助金的决议，遭总统否则**

|  |  |  |
| --- | --- | --- |
| 运算符 | 表达式 | 说明 |
| and | a and b | 逻辑与，当a为True时才计算b |
| or | a or b | 逻辑或，当a为False时才计算b |
| not | not a | 逻辑非 |

a = 4

b = 2

c = 0

print(a>b and b>c) #a>b为True继续计算b>c，b>c也为True则结果为

True

print(a>b and b<c)#a>b为True继续计算c<b，b>c结果为False则结果为

False

print(a>b or c<b) #a>b为True则不继续计算c<b，结果为True

print(not c<b) #c<b为True not True结果为False

print(not a<b) #a<b为False not Flase结果为True

**第二部分**

**1.4 赋值运算符**

|  |  |  |
| --- | --- | --- |
| 运算符 | 表达式 | 说明 |
| = | c=a + b | 简单赋值运算符， 将 a + b 的运算结果赋值为 c（重点：两个变量值的交换） |
| += | c +=a | 加法赋值运算符 c += a 等效于 c = c + a |
| -= | c -=a | 减法赋值运算符 c -= a 等效于 c = c - a |
| \*= | c \*=a | 乘法赋值运算符 c \*= a 等效于 c = c \* a |
| /= | c /=a | 除法赋值运算符 c /= a 等效于 c = c / a |
| %= | c %=a | 取模赋值运算符 c %= a 等效于 c = c % a |
| //= | c //=a | 取整除赋值运算符 c //= a 等效于 c = c // a |
| \*\*= | c \*\*=a | 幂赋值运算符 c \*\*= a 等效于 c = c \*\* a |

使用赋值号实现两个数值的交换：

a=23

b=56

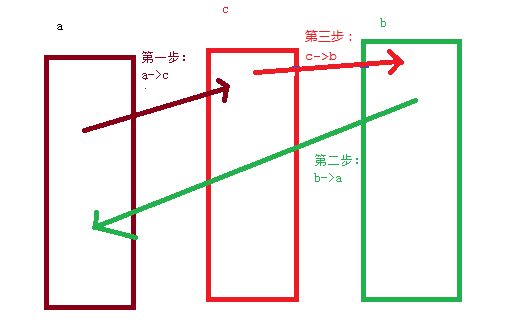
a和b的数值交换，采用第三方变量实现

c=a

a=b

b=c

交互过程如下图，



**为什么不能写成a=b，b=a？这是因为a=b,是把b的值赋给a,a就等于b的值，然后把a的值赋给b,b的值还是b的值，即a和b的值都等于b的值。**

**1.5 运算符的优先级**

Python的算术表达式具有结合性和优先性。结合性是指表达式按照从左到右、先乘除后加减的原则进行计算。

2+3\*\*2

（2+3）\*\*2

2+3\*\*2+5\*4-7\*2

（2+3）\*\*2+5\*（4-7）\*2

Python运算符优先级(由高到低)

\*\* 指数 (最高优先级)

~ + - 按位翻转, 一元加号和减号 (正负号)

\* / % // 乘，除，取模和取整除

+ - 加法减法

<= < > >= 比较运算符

<> == != 等于运算符

= %= /= //= -= += \*= \*\*= 赋值运算符

以上内容了解即可。

另外，小括号可以改变优先级，有()的情况优先计算()中的表达式。

1. **表达式**

表达式是将不同类型的数据(常量、变量、函数)用运算符按照一定得规则连接起来的式子。

2+3

2\*\*3

2>3