



链滴

Java 运算符

作者: [shuaibing90](#)

原文链接: <https://ld246.com/article/1582517207247>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)



Java运算符

算数运算符

算数运算符

+	加法运算, 字符串连接运算
-	减法运算
*	乘法运算
/	除法运算
%	取模运算, 两个数字相除取余数
++、--	自增自减运算

Java中, 整数使用以上运算符, 无论怎么计算, 也不会得到小数

```
public static void main(String[] args) {  
    int i = 1234;  
    System.out.println(i/1000*1000); //计算结果是1000  
}
```

- ++运算, 变量自己增长1, --运算, 变量自己减少1, 用法与++一致

独立运算:

变量在独立运算时, 前++和后++没有区别

变量前++: 例如 ++i

变量后 ++ : 例如 i++

混合运算:

和其他变量放在一起, 前 ++ 和后 ++ 就产生了不同

变量前 ++ : 变量a自己加1, 将加1后的结果赋值给b, 也就是说a先计算。a和b的结果都是2

变量后 ++ : 变量a先把自己的值1, 赋值给变量b, 此时变量b的值就是1, 变量a自己再加1。a的结果是2, b

的结果是1

变量前++

```
public static void main(String[] args) {  
    int a = 1;  
    int b = ++a;  
    System.out.println(a);//计算结果是2  
    System.out.println(b);//计算结果是2  
}
```

变量后++

```
public static void main(String[] args) {  
    int a = 1;  
    int b = a++;  
    System.out.println(a);//计算结果是2  
    System.out.println(b);//计算结果是1  
}
```

• +在字符串中的操作:

- 符号在遇到字符串的时候, 表示连接、拼接的含义

"a"+"b" 的结果是 "ab" , 连接含义

```
public static void main(String[] args){  
    System.out.println("5+5="+5+5);//输出5+5=55  
}
```

赋值运算符

赋值运算符包括:

=	等于号
+=	加等于
-=	减等于
*=	乘等于
/=	除等于
%=	取模等

• 赋值运算符, 就是将符号右边的值, 赋给左边的变量

```
public static void main(String[] args){
    int i = 5;
    i+=5;//计算方式 i=i+5 变量i先加5, 再赋值变量i
    System.out.println(i); //输出结果是10
}
```

比较运算符

比较运算符

==	比较符号两边数据是否相等，相等结果是true
<	比较符号左边的数据是否小于右边的数据，如 小于结果是true
>	比较符号左边的数据是否大于右边的数据，如 大于结果是true
<=	比较符号左边的数据是否小于或者等于右边 数据，如果小于结果是true
>=	比较符号左边的数据是否大于或者等于右边 数据，如果小于结果是true
!=	不等于符号，如果符号两边的数据不相等， 果是true

- 比较运算符，是两个数据之间进行比较的运算，运算结果都是布尔值 **true** 或者 **false**

```
public static void main(String[] args) {
    System.out.println(1==1);//true
    System.out.println(1<2);//true
    System.out.println(3>4);//false
    System.out.println(3<=4);//true
    System.out.println(3>=4);//false
    System.out.println(3!=4);//true
}
```

逻辑运算符

逻辑运算符包括：

&&短路与 1.两边都是true，结果是true 2. 一
是false，结果是false短路特点：符号左边是false，右边不再运算

!取反 1. ! true 结果是false 2. ! false结果是true

- 逻辑运算符，是用来连接两个布尔类型结果的运算符，运算结果都是布尔值 **true** 或者 **false**

```
public static void main(String[] args) {
    System.out.println(true && true);//true
    System.out.println(true && false);//false
    System.out.println(false && true);//false, 右边不计算

    System.out.println(false || false);//false
}
```

```
System.out.println(false || true);//true
System.out.println(true || false);//true, 右边不计算

System.out.println(!false);//true
}
```

三元运算符

- 三元运算符格式

数据类型 变量名 = 布尔类型表达式? 结果1: 结果2

- 三元表达式计算方式

布尔类型表达式结果是 true, 三元运算符整体结果为结果1, 赋值给变量。

布尔类型表达式结果是 false, 三元运算符整体结果为结果2, 赋值给变量。

```
public static void main(String[] args) {
    int i = (1==2 ? 100 : 200);
    System.out.println(i);//200
    int j = (3<=4 ? 500 : 600);
    System.out.println(j);//500
}
```