**1.运算符是一个符号，告诉编译器执行特定的数学或者逻辑操作，scala提供一下类型的运算符：**

1.1算数运算符

1.2关系运算符

1.3逻辑运算符

1.4位运算符

1.5赋值运算符

**2.算数运算符**

声明变量x,y，一般是数值类型：

|  |
| --- |
| scala> val x=10  x: Int = 10  scala> val y=20  y: Int = 20 |

2.1 + 表示两个变量相加：

|  |
| --- |
| scala> x + y  res0: Int = 30 // res 为REPL的内置变量 |

2.2 – 表示两个变量相减：

|  |
| --- |
| scala> x - y  res1: Int = -10 |

2.3 \* 表示两个变量相乘

|  |
| --- |
| scala> x \* y  res2: Int = 200 |

2.4 / 表示前面的变量除以后面的变量

|  |
| --- |
| scala> y / x  res4: Int = 2 |

2.5 % 表示模运算，整数除法后的余数

|  |
| --- |
| scala> y % x  res6: Int = 0 |

**3.关系运算符**

3.1 == 表示两个变量的值是否相等，如果是的话条件为真，即true,否则为false

|  |
| --- |
| scala> x == y  res8: Boolean = false |

3.2 != 表示两个变量的值是否不相等，如果是的话条件为真

|  |
| --- |
| scala> x != y  res9: Boolean = true |

3.3 > 表示左边的变量是否大于右边变量，如果是的话条件为真

|  |
| --- |
| scala> x > y  res10: Boolean = false |

3.4 < 表示左边的变量是否小于右边变量，如果是的话条件为真

|  |
| --- |
| scala> x < y  res11: Boolean = true |

3.5 >= 表示左边的变量是否大于或等于右边变量，如果是的话条件为真

|  |
| --- |
| scala> x >= y  res12: Boolean = false |

3.6 <= 表示左边的变量是否小于或等于右边，如果是的话条件为真

|  |
| --- |
| scala> x <= y  res13: Boolean = true |

**4.逻辑运算符**

声明变量T = true,F = false ：

|  |
| --- |
| scala> val T = true  T: Boolean = true  scala> val F = false  F: Boolean = false |

4.1 && 表示与操作，如果两个变量都为真，则返回真

|  |
| --- |
| scala> T && F  res17: Boolean = false |

4.2 || 表示或操作，如果有一个变量为真，则返回真

|  |
| --- |
| scala> T || F  res18: Boolean = true |

4.3 ！表示非操作，即取反，如果条件为真，则取反后返回值为家

|  |
| --- |
| scala> ! (T || F)  res19: Boolean = false |

**5.位移运算符**

假设x=60,y=13,用二进制格式表示为：

A = 0011 1100

B = 0000 1101

----------------

A&B = 0000 1100 //与，12

A|B = 0011 1101 //或，61

A^B = 0011 0001 //异或，49

~A = 1100 0011 //非，-61

5.1 >> 表示向右移动指定位数，如 A >> 2表示 0011 1100 向右移动两位，则为1111 是15

5.2 << 表示向左移动指定位数，如A << 2表示 0011 1100 向左移动两位，则为1111 0000 是240

5.3 >>> 表示向右移动并补零操作，A >>> 2 即 0000 1111 是15

**6.赋值运算符**

假设声明变量如：

|  |
| --- |
| scala> var x = 10  x: Int = 10  scala> var y = 20  y: Int = 20 |

6.1 = 表示右边的值分配给左边的变量，如var x = 10

6.2 += 表示加法和赋值操作，即将左边变量+右边变量然后重新将值赋值给左边变量，相当于 x = x + y

|  |
| --- |
| scala> x += y  scala> x  res2: Int = 30 |

6.3 -= 表示减法和赋值操作，即将左边变量-右边变量的值重新赋值给左边变量，相当于x = x –y

|  |
| --- |
| scala> x -= y  scala> x  res4: Int = 10 |

6.4 \*= 表示乘法和赋值运算，即左边变量\*右边变量然后重新将结果赋值给左边变量，相当于 x = x \* y

|  |
| --- |
| scala> x \*= y  scala> x  res6: Int = 200 |

6.5 /= 表示除法和赋值运算，即左边变量/右边变量然后重新将结果赋值给左边变量，相当于 x = x / y

|  |
| --- |
| scala> x /= y  scala> x  res8: Int = 10 |

6.6 %= 表示取模和赋值运算，即左边变量%右边变量然后重新将结果赋值给左边变量，相当于 x = x % y

|  |
| --- |
| scala> x %= y  scala> x  res10: Int = 10  scala> 10 % 20  res14: Int = 10 |

6.7 <<= 左移位并赋值运算，相当于 x = x << y

|  |
| --- |
| scala> var z = 60  z: Int = 60  scala> z << 2  res17: Int = 240 |

6.8 >>= 右移并赋值运算，相当于 x = x >> y

|  |
| --- |
| scala> var z = 60  z: Int = 60  scala> z >>= 2  scala> z  res19: Int = 15 |

6.8 &= 位与和赋值运算，相当于 x = x & y

|  |
| --- |
| scala> var x = 60  x: Int = 60  scala> var y = 13  y: Int = 13  scala> x &= y  scala> x  res23: Int = 12 |

6.9 |= 位或和赋值运算，相当于 x = x | y

|  |
| --- |
| scala> var x = 60  x: Int = 60  scala> var y = 13  y: Int = 13  scala> x |= y  scala> x  res25: Int = 61 |

6.10 ^= 位异或和赋值运算，相当于 x = x ^ y

|  |
| --- |
| scala> var x = 60  x: Int = 60  scala> var y = 13  y: Int = 13  scala> x ^= y  scala> x  res27: Int = 49 |

**7.运算符优先级**

如果有两个或更多运算符，又没有给出括号的时候，首先执行的是优先级高的运算符，优先级由运算符的首字符决定：

|  |
| --- |
| \* / % |
| + - |
| : |
| < > |
| != |
| & |
| ^ |
| | |
| ~ |
| 最低优先级：= |

一般后置运算符优先级地狱中置运算符：

a 中置运算符 b 后置运算符 相当于：

(a 中置运算符 b )后置运算符,为避免编程时优先级错误，则应该使用括号()做分层处理。