

# 位运算

---

参加运算的两个数据，按二进制位进行“与”运算。

运算规则：0&0=0; 0&1=0; 1&0=0; 1&1=1;

即：两位同时为“1”，结果才为“1”，否则为0

## 按位或运算符（|）

参加运算的两个对象，按二进制位进行“或”运算。

运算规则：0|0=0; 0|1=1; 1|0=1; 1|1=1;

即：参加运算的两个对象只要有一个为1，其值为1。

例如：3|5 即 00000011 | 0000 0101 = 00000111 因此，3|5的值得7。

## 异或运算符（^）

参加运算的两个数据，按二进制位进行“异或”运算。

运算规则：0^0=0; 0^1=1; 1^0=1; 1^1=0;

即：参加运算的两个对象，如果两个相应位为“异”（值不同），则该位结果为1，否则为0。

## 取反运算符（~）

参加运算的一个数据，按二进制位进行“取反”运算。

运算规则：~1=0; ~0=1;

即：对一个二进制数按位取反，即将0变1，1变0

## 左移运算符（<<）

将一个运算对象的各二进制位全部左移若干位（左边的二进制位丢弃，右边补0）。

例：a = a<< 2将a的二进制位左移2位，右补0，

左移1位后a = a \* 2;

若左移时舍弃的高位不包含1，则每左移一位，相当于该数乘以2。

## 右移运算符（>>）

将一个数的各二进制位全部右移若干位，正数左补0，负数左补1，右边丢弃。

操作数每右移一位，相当于该数除以2。

例如：a = a>> 2 将a的二进制位右移2位，

左补0 or 补1得看被移数是正还是负。

## 复合赋值运算符

**\*位运算\***符与赋值运算符结合，组成新的复合赋值运算符，它们是：

**&=** 例：a &=b 相当于a=a&b

**|=** 例：a |=b 相当于a=a|b

**>>=** 例：a >>=b 相当于a=a>>b

**<<=** 例：a <<=b 相当于a=a<<b

**^=** 例：a ^= b 相当于a=a^b

运算规则：和前面讲的复合赋值运算符的运算规则相似。