

4.运算符与表达式

运算符：

用来对数据进行运算的符号，就可以称为运算符

运算符种类	作用	包含运算符
算术运算符	用于各类数值运算	加(+)、减(-)、乘(*)、除(/)、求余(%）、自增(++）、自减(--)
关系运算符	用于比较运算	大于(>)、小于(<)、等于(==)、大于等于(>=)、小于等于(<=)、不等于(!=)
逻辑运算符	用于逻辑运算	与(&&)、或()、非(!)
位操作运算符	按二进制位进行运算	位与(&)、位或()、位非(~)、位异或(^)、左移(<<)、右移(>>)
赋值运算符	用于赋值运算	简单赋值(=)、复合运算赋值(+=、-=、*=、/=, ...)
条件运算符	用于条件求值	(?:)
逗号运算符	用于组合表达式	(,)
指针运算符	用于解引用和取地址	解引用(*)、取地址(&)
求字节数运算符	用于计算数据类型大小	(sizeof)
其它运算符	其它	括号(), 下标[], 成员(->, .)等

分类：

按运算符在表达式中与运算分量的关系（连接运算分量的个数），运算符可分为：

- 单目运算符，即一元运算符，只需要一个运算分量，
 - 如：-5，!a；
- 双目运算符，即二元运算符，需要两个运算分量，
 - 如：a+b，x||y；
- 三目运算符，即三元运算符，需要三个运算分量，如：a>b?a:b。
- 条件运算符（?:）是C语言中唯一的三目运算符

算数运算符优先级：

- 在复杂的算术表达式中：“()”的优先级最高，“*、/、%”运算符的优先级高于“+、-”运算符。
- 算术运算符中的结合性均为“左结合”。
- 左结合（Left Associativity）：左结合指的是相同优先级的多个运算符在表达式中从左到右依次结合。也就是说，先出现的运算符会先结合。

Tip: "/"运算：

注意：对于“/”运算符，C语言中规定：

- 当它的两个运算分量均为整数时，计算结果也必须为整数。
- 如果两个运算分量中有一个数是浮点型时，则结果也应该为浮点型数据。
例如：99.0/5的结果为19.8。

关系运算符优先级：

- “>”、“>=”、“<”和“<=”的优先级相同；
- “==”和“!=”的优先级相同；
- 前四种的优先级高于后两种。
- 例如：

● $a == b < c$

等价于

$a == (b < c)$

● $a > b > c$

等价于

$(a > b) > c$

5>4>3?

- 关系运算符中的结合性均为“左结合”。

逻辑运算符：

- 逻辑运算符主要用于逻辑运算；
- 包含了“&&”（逻辑与）、“||”（逻辑或）、“!”（逻辑非）。

a	b	a&&b	a b	!a
真	真	真	真	假
真	假	假	真	假
假	真	假	真	真
假	假	假	假	真

位运算：

- 左移位 (<<)

- 左移位表示将一个二进制数向左移动指定的位数，即在右侧补0。

- 右移位 (>>)

- 右移位表示将一个二进制数向右移动指定的位数，即在左侧补0或1(取决于最高位的值)。

- 在使用“<<”和“>>”进行移位操作时，需要注意最高位，即正数向左移位会在右侧补0，而负数向左移位会在右侧补1，反之，正数向右移位会在左侧补0，而负数向右移位会在左侧补1

- 尽可能只对无符号类型使用移位操作

条件运算符：

- 条件运算符中由“?”和“:”组成，是C语言中唯一的一个三目运算符

`<COND> ? <T> : <F>`

- 条件表达式的执行过程是：

- 先计算表达式<COND>的值；

- 若该值不为0，则计算表达式<T>的值，并将其作为整个条件表达式的值；

- 否则，就计算表达式<F>的值，并将该值作为整个条件表达式的值。

赋值运算符：

- `a /= b+c` 等价于 `a=a/(b+c)`

- `a %= b+c` 等价于 `a=a%(b+c)`

- 复合运算符在书写时，两个运算符之间不能有空格，否则就是错误的。

- 复合赋值运算符右边的表达式应作为一个整体对待。

未定义行为：同时读取和修改

- 避免在同一表达式中对同一变量同时进行修改和读取操作
- `printf("%d, %d", i, i++)`；是未定义行为
 - C 标准未规定函数参数的求值顺序（从左到右或从右到左）
 - 读取*i*和*i++*没有明确的顺序关系
- 同样的，`i=i++`也是未定义行为