在讲结合性之前，先提一下c语言的运算符及其优先级关系。

C语言运算符优先级

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 优先级 | 运算符 | 名称或含义 | 使用形式 | 结合方向 | 说明 |
| 1 | [] | 数组下标 | 数组名[常量表达式] | 左到右 | -- |
| () | 圆括号 | (表达式）/函数名(形参表) | -- |
| . | 成员选择（对象） | 对象.成员名 | -- |
| -> | 成员选择（指针） | 对象指针->成员名 | -- |
|  | | | | | |
| 2 | - | 负号运算符 | -表达式 | 右到左 | 单目运算符 |
| ~ | 按位取反运算符 | ~表达式 |
| ++ | 自增运算符 | ++变量名/变量名++ |
| -- | 自减运算符 | --变量名/变量名-- |
| \* | 取值运算符 | \*指针变量 |
| & | 取地址运算符 | &变量名 |
| ! | 逻辑非运算符 | !表达式 |
| (类型) | 强制类型转换 | (数据类型)表达式 | -- |
| sizeof | 长度运算符 | sizeof(表达式) | -- |
|  | | | | | |
| 3 | / | 除 | 表达式/表达式 | 左到右 | 双目运算符 |
| \* | 乘 | 表达式\*表达式 |
| % | 余数（取模） | 整型表达式%整型表达式 |
| 4 | + | 加 | 表达式+表达式 | 左到右 | 双目运算符 |
| - | 减 | 表达式-表达式 |
| 5 | << | 左移 | 变量<<表达式 | 左到右 | 双目运算符 |
| >> | 右移 | 变量>>表达式 |
|  | | | | | |
| 6 | > | 大于 | 表达式>表达式 | 左到右 | 双目运算符 |
| >= | 大于等于 | 表达式>=表达式 |
| < | 小于 | 表达式<表达式 |
| <= | 小于等于 | 表达式<=表达式 |
| 7 | == | 等于 | 表达式==表达式 | 左到右 | 双目运算符 |
| ！= | 不等于 | 表达式!= 表达式 |
|  | | | | | |
| 8 | & | 按位与 | 表达式&表达式 | 左到右 | 双目运算符 |
| 9 | ^ | 按位异或 | 表达式^表达式 | 左到右 | 双目运算符 |
| 10 | | | 按位或 | 表达式|表达式 | 左到右 | 双目运算符 |
| 11 | && | 逻辑与 | 表达式&&表达式 | 左到右 | 双目运算符 |
| 12 | || | 逻辑或 | 表达式||表达式 | 左到右 | 双目运算符 |
|  | | | | | |
| 13 | ?: | 条件运算符 | 表达式1?  表达式2: 表达式3 | 右到左 | 三目运算符 |
|  | | | | | |
| 14 | = | 赋值运算符 | 变量=表达式 | 右到左 | -- |
| /= | 除后赋值 | 变量/=表达式 | -- |
| \*= | 乘后赋值 | 变量\*=表达式 | -- |
| %= | 取模后赋值 | 变量%=表达式 | -- |
| += | 加后赋值 | 变量+=表达式 | -- |
| -= | 减后赋值 | 变量-=表达式 | -- |
| <<= | 左移后赋值 | 变量<<=表达式 | -- |
| >>= | 右移后赋值 | 变量>>=表达式 | -- |
| &= | 按位与后赋值 | 变量&=表达式 | -- |
| ^= | 按位异或后赋值 | 变量^=表达式 | -- |
| |= | 按位或后赋值 | 变量|=表达式 | -- |
|  | | | | | |
| 15 | ， | 逗号运算符 | 表达式,表达式,… | 左到右 | -- |

说明：

    运算符共分为15级，1级优先级最高，15级优先级最低。

    同一优先级的运算符，运算次序由结合方向所决定。(结合性：2 13 14 是从右至左 其他都是 从左至右)  
    简单记就是：！ > 算术运算符 > 关系运算符 > && > || > 赋值运算符

再详细一点，附带一个口诀。(网上找的)

括号成员第一;                //括号运算符[]() 成员运算符. ->

全体单目第二;              //所有的单目运算符比如++、 --、 +(正)、 -(负) 、指针运算\*、&

乘除余三,加减四;         //这个"余"是指取余运算即%

移位五，关系六;          //移位运算符：<< >> ，关系：> < >= <= 等

等于(与)不等排第七;      //即== 和!=

位与异或和位或;   "三分天下"八九十;     //这几个都是位运算: 位与(&)异或(^)位或(|)

逻辑或跟与;              //逻辑运算符:|| 和 &&

十二和十一;           //注意顺序:优先级(||) 底于 优先级(&&)

条件高于赋值,                //三目运算符优先级排到13 位只比赋值运算符和","高

逗号运算级最低!        //逗号运算符优先级最低

如果想详细了解运算符优先级的关系，我想上面的表格和口诀足以了。

有了以上的基础，我们回归本篇文章的主题－－－结合性。

观察上述运算符表格我们不难发现，在优先级相同的列表中，有的不止存在一种运算符，那当这两种运算符碰到一起了，该怎么计算呢？

又或者，两个相同的运算符在一起，该怎么进行运算呢？

要解决这些问题，就必须了解结合性了。

C 语言与其他高级语言相比, 一个显著的特点就是其运算符特别丰富, 共有34 种运算符。C 语言将这34 种运算符规定了不同的优先级别和结合性。优先级是用来标识运算符在表达式中的运算顺序的, 在求解表达式的值的时候, 总是先按运算符的优先次序由高到低进行操作, 可是, 当一个运算对象两侧的运算符优先级别相同时, 则按运算符的结合性来确定表达式的运算顺序。   
运算符的结合性指同一优先级的运算符在表达式中操作的组织方向, 即: 当一个运算对象两侧运算符的优先级别相同时, 运算对象与运算符的结合顺序, C 语言规定了各种运算符的结合方向( 结合性) 。

大多数运算符结合方向是“自左至右”, 即: 先左后右, 例如a- b+c, b 两侧有- 和+两种运算符的优先级相同, 按先左后右结合方向, b 先与减号结合, 执行a- b 的运算, 再执行加c 的运算。

除了自左至右的结合性外, C 语言有三类运算符参与运算的结合方向是从右至左。即: 单目运算符, 条件运算符, 以及赋值运算符。

关于结合性的概念在其他高级语言中是没有的, 这是C语言的特点之一,特别是从右至左结合性容易出错, 下面通过几个具体的运算符来剖析C 语言运算符的结合性。   
若a 是一个变量, 则++a 或a++和- - a 或a- - 分别称为前置加或后置加运算和前置减或后置减运算, 且++a 或a++等价于a=a+1, - - a 或a- - 等价于a=a- 1, 即都是使该变量的值增加1 或减少1。由此可知, 对一个变量实行前置或后置运算, 其运算结构是相同的, 但当它们与其他运算结合在一个表达式中时, 其运算值就不同了。

前置运算是变量的值先加1 或减1, 然后将改变后的变量值参与其他运算, 如x=5; y=8; c=++x\*y; 运算后, c 的值是48,x 的值是6,y 的值是8。

而后置运算是变量的值先参与有关运算, 然后将变量本身的值加1 减1, 即参加运算的是该变量变化前的值。如x=5; y=8; c=x++\*y;运算后, c 的值是40,x 的值是6, y 的值是8。

值得注意的是, 前置、后置运算只能用于变量, 不能用于常量和表达式, 且结合方向是从右至左。如当i=6 时, 求- i++的值和i 的值。由于“- ”(负号) “++”为同一个优先级, 故应理解为- (i++), 又因是后置加, 所以先有- i++的值为- 6, 然后i 增值1 为7, 即i=7。

讲到这里，大家应该都明白了吧。说白了，许多操作符的优先级都是相同的。这时，操作符的结合性就开始发挥作用了。在表达式中如果有几个优先级相同的操作符，结合性就起仲裁的作用，由它决定哪个操作符先执行。先执行，可以看做是加上括号。比如，右结合（从右到左），那就是，把右边的运算用括号先括起来，再与左边的进行计算，这样自然是先执行右边的了。  比如   int a,b=1,c=2;         a=b=c;  这里'='的结合性是从右到左。 故a=b=c; 可变为a=(b=c);  即a=2。

差不多就讲这些了。

总的来说，掌握上述优先级，结合性是有一定用处的，但也不是必须的。 毕竟，咱们有()这个大招不是，哪里不确定，哪里补()就是了。