

C语言基础Day3-运算符

一、运算符

- 算术运算符：用于处理四则运算
- 赋值运算符：用于将表达式的值赋给变量
- 比较运算符：用于表达式的比较，并返回一个真值或者假值
- 逻辑运算符：用于根据表达式的值返回真值或者假值
- 位运算符：用于处理数据的位运算
- sizeof运算符：用于求字节数长度

```
int a = 10;
int b = 20;

// 两个整数相除，如果要得到小数，必须有一个是double类型或者float类型（强制类型转换）
double f = a * 1.0 / b; // a * 1.0 将转换成double类型
printf("%lf\n", f);
```

使用取余运算符求出一个数字的个位、十位、百位：

```
int num = 1234;
int a = num % 10; // 个位
int b = num / 10 % 10; // 十位
int c = num / 100 % 10; // 百位
int d = num / 1000 % 10; // 千位

printf("%d %d %d %d\n", a, b, c, d);
```

二、自加或者自减运算符

- a++ ++在后，先使用 在a = a + 1
- ++a ++在前，先a = a+1,在使用

三、短路求值

- A || B 或者 B只要有一个是真，整个表达式就是真，那么如果A是真，B表达式就不执行
- A && B A和B必须同时为真，总的结果就是真，如果A是假的，那么整个表达式就是假的，B表达式就不执行

四、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 ^Q]	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 ^Q	-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		--

```
int a1 = 1;
int b1 = 2;
int e = (a1++, b1++, a1 + b1, 10, 200, a1 + b1);// 逗号运算符 最终的结果是最后
一个表达式的结果
printf("%d\n",e);
```

五、类型转换

数据有不同的类型，不同类型数据之间进行混合运算时必定会涉及到类型的转换问题。

转换的方法有两种：

- 自动转换（隐式转换）：遵循一定的规则，由编译系统自动完成
- 强制类型转换：把表达式的运算结果强制转换成所需的数据类型

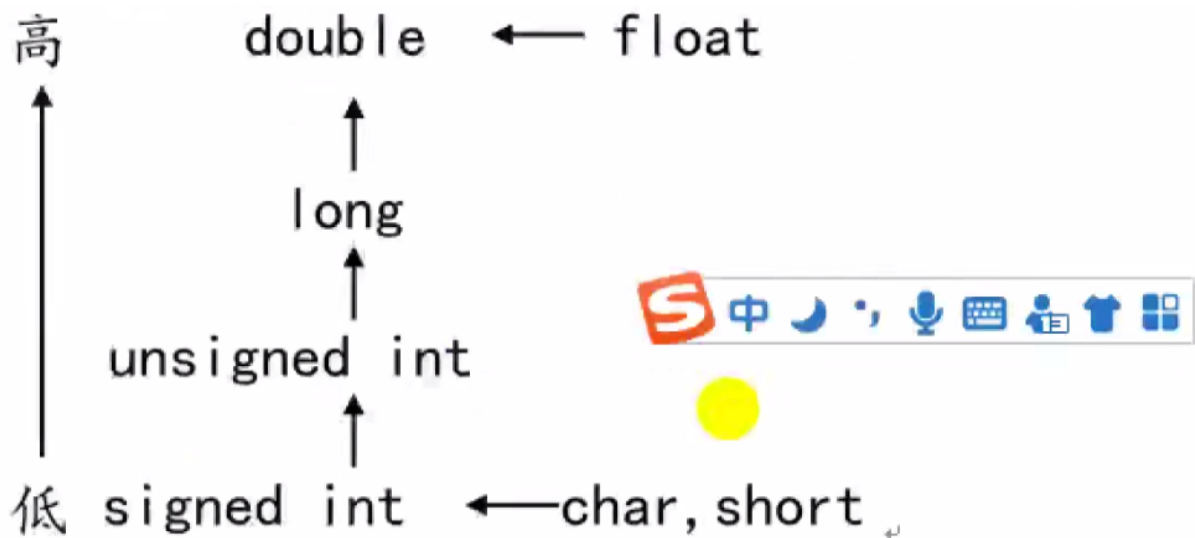
类型转换的原则：占用字节数少的类型，向占用内存字节数多的类型转换，以保证精度不降低。

```
int a = 3;
double b = a; // 隐式转换

// 类型的转换 所需要的类型和给的类型不匹配 需要转换
// 强制转换 (需要转的类型) 原来的数据类型
printf("%d\n", (int)3.1456);
```

强制类型转换指的是使用强制类型转换符，将一个变量或者表达式转换成所需要的类型，（类型说明符）表达式

强制转化的宗旨：数据不丢失，小的转大的



有符号的可以转无符号的，多一位