

## 一,是什么?

“ + ”加法运算符 “ - ”减法运算符 “ \* ” 乘法运算符; “ / ”除法运算符  
“ % ” 取余运算符.

## 二,有什么用?

数据运算

## 三,怎么用?

直接在两个数之间加符号就可以了.如:1+1, 2-1, 3\*1, 4/2, 5%2

## 四,什么时候用?

当你想进行数据运算时候.

## 五,有什么特点?

- 1.不同数据类型精度不一样.double>float>int. 取余运算符只能整数.+1:正数 -1:负数
- 2.若用高精度向低精度变量赋值.会丢失精度.如:int a=1.8;那么a值是1.(非四舍五入)
- 3.运算优先级是.从左到右 但: \* = / = % > + = - 如果要人为改变则加().
- 4.运算时两个数据精度不一样,会自动类型转换,且是低精度向高精度转换.
- 5.同精度运算只能得到同精度的结果.如果需要提高精度,需要强制类型转换.

```
1  /*算术运算*/
2  #include<stdio.h>
3  int main(){
4      int a=4;
5      int b=2;
6      printf("a+b=%d\n",a+b);
7      printf("a-b=%d\n",a-b);
8      printf("a*b=%d\n",a*b);
9      printf("a/b=%d\n",a/b);
10     printf("a%%b=%d\n",a%%b);
11 }
12 |
```

连续写两个百分号编译器就认为  
%是一个字符,而不是格式控制符

```
printf("%d\n",a);
*/
```

```
int a=(int)1.8;
printf("%d\n",a);
```

强制类型转换

前面加一个小括号里面写相应的数据类型

```
}
```

# 1.C语言的基本运算

1. 算术运算符	(+ - * / %)
2. 关系运算符	(> < == >= <= !=)
3. 逻辑运算符	(! &&   )
4. 位运算符	(<< >> ~   ^ &)
5. 赋值运算符	(= 及其扩展赋值运算符)
6. 条件运算符	(? :)
7. 逗号运算符	(,)
8. 指针运算符	(* 和 &)
9. 求字节数运算符	(sizeof)
10. 强制类型转换运算符	((类型))
11. 分量运算符	(. →)
12. 下标运算符	([ ])
13. 其他	(如函数调用运算符())

运算时两个数据精度不一样,会自动类型转换,且是低精度向高精度转换.

```
*/  
int a=1.2+3; //1.2+3.0=4.2---4;
```

运算过程是把低精度的整型3 转换成高精度的浮点数  
然后和浮点数1.2运算