

---

# 嵌入式开发

## 运算符

---

创客学院 小美老师

- ▶ 1 / 掌握赋值运算符
- 2 / 掌握特殊运算符
- 3 / 熟悉运算符的优先级
- 4 / 总结与思考

# 赋值运算符

赋值运算符为 “=”，其运算的一般形式如下：

<左值表达式> = <右值表达式>

赋值复合运算符其运算的一般形式如下：

<变量> <操作符> = <表达式>

# 赋值复合运算符

C语言的赋值复合运算符如下：

运算符	功能说明	示例
<code>+=</code>	加赋值复合运算符	<code>a+=b</code> 等价于 <code>a=a+b</code>
<code>-=</code>	减赋值复合运算符	<code>a-=b</code> 等价于 <code>a=a-b</code>
<code>*=</code>	乘法赋值复合运算符	<code>a*=b</code> 等价于 <code>a=a*b</code>
<code>/=</code>	除法赋值复合运算符	<code>a/=b</code> 等价于 <code>a=a/b</code>
<code>%=</code>	求余赋值复合运算符	<code>a%=b</code> 等价于 <code>a=a%b</code>
<code>&amp;=</code>	位与赋值复合运算符	<code>a&amp;=b</code> 等价于 <code>a=a&amp;b</code>
<code> =</code>	位或赋值复合运算符	<code>a =b</code> 等价于 <code>a=a b</code>
<code>^=</code>	位异或赋值复合运算符	<code>a^=b</code> 等价于 <code>a=a^b</code>
<code>&gt;&gt;=</code>	位右移赋值复合运算符	<code>a&gt;&gt;=b</code> 等价于 <code>a=a&gt;&gt;b</code>
<code>&lt;&lt;=</code>	位左移赋值复合运算符	<code>a&lt;&lt;=b</code> 等价于 <code>a=a&lt;&lt;b</code>

# 案例

```
#include <stdio.h>
int main(int argc, char **argv)
{
    int count, sum;
    count = 0; sum = 0;
    while (count++ < 20) {    sum +=count; }
    printf("sum = %d\n", sum);
    return 0;
}
```

# C语言的特殊运算符

条件运算符"?:"

是三元运算符, 其运算的一般形式是:

✓ <表达式1> ? <表达式2> : <表达式3>

例如:

```
int x=82, y=101;
```

```
x >= y ? x+18 : y-100 //运算结果为1
```

```
x < (y-11) ? x-22 : y-1 //运算结果为60
```

以 "降" 为例  
"就是"  
它最好了。

# 案例

```
int main()
{   int x, y=25;
    x=70;
    y= x++ > 70 ? 100 : 0 ;
    printf( "x=%d y=%d" , x, y);
    return 0;
}
```

运行结果： x=71 y=0 ✓

# C语言的特殊运算符

## 逗号运算符

```
float x=10.5, y=1.8, z=0;
```

```
z = (x+=5, y = x+0.2) ;
```

```
z = (x=y=5, x +=1) ;
```

```
z = (x=5, y=6, x+y) ;
```

```
z = (z=8, x=5, y=3) ;
```



# C语言的特殊运算符

## ✓ sizeof运算符

运算的一般形式: sizeof(<类型或变量名>)

**注意：它只针对数据类型，而不针对变量！**

✓ 例如：

sizeof(double)

sizeof(long)

.....

# C运算符的优先级

优先级	运算符及其含义	结合规律
1	[] () . -> 后缀++ 后缀--	从左向右
2	前缀++ 前缀-- sizeof & * + - ~ !	从右向左
3	强制类型转换	从右向左
4	* / % (算术乘除)	从左向右
5	+ - (算术加减)	从左向右
6	<< >> (位移位)	从左向右
7	< <= > >=	从左向右
8	== !=	从左向右
9	& (位逻辑与)	从左向右
10	^ (位逻辑异或)	从左向右
11	(位逻辑或)	从左向右
12	&&	从左向右
13		从左向右
14	? :	从右向左
15	= *= /= %= += -= <<= >>= &= ^=  =	从右向左
16	,	从左向右

# C运算符的优先级

例如:

```
int x=1, y=0, z=0;
```

```
((x++) && (y++)) || (++z) //结果为 1 (2 1 1)
```

```
((!(x+1>0)) && (y++)) || (++z) //结果为 1 (1 0 1)
```

```
x += (y==z), y=(x+2), z=(x+y+x)>0 //x为2, y为4, z为1
```

✓ 事实证明，我右法对的。

# 总结与思考

✓ 人觉类型是9框架，常数是实体，变量是箱子。 0 ① U

主要介绍了C语言中的运算符，包括赋值运算符、特殊运算符以及运算符的优先级。

思考：

✓

$(fd = open()) == 1$

- C语言中有没有三目运算符？

系统调用，api等时候

- “=” 和 “==” 的优先级哪个高？