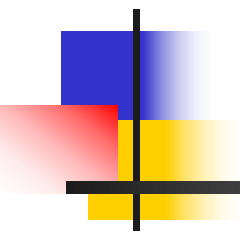


第2章 数据类型、 运算符和表达式





第一节 标识符

定义：用来标识符号常量名、变量名、函数名、数组名、文件名等对象的有效字符序列

命名规则：

- 1) 由字母（大小写）、数字、下划线
- 2) 第一个字符必须为字母或下划线

如： ab	name	file1	_sum
5x	Doc.	Bao ei	int





注意事项:

(1) 区分大小写。

如: name Name NAME

(2) C语言的关键字不可用作命名

(3) ”见名知意”，即用含有相应含义的英文单词作为标识符

第二节 C的数据类型

C
数据类型

基本类型

整型

短整型short

整型int

长整型long

实型

单精度型

float

双精度型double

字符类型char

枚举类型enum

数组

结构体struct

共用体union

构造类型

指针类型

空类型void

定义类型typedef

数据类型决定：

1. 数据占内存字节数
2. 数据取值范围
3. 其上可进行的操作

从另一个角度来看，C语言中的数据又可以分成：常量和变量



第三节 常量

整型常量（整数）

如： 123 -5 8

实型常量（实数）

如： 5.4 0.486

字符常量（字符）

如： ‘a’ ‘H’

转义字符

如： ‘\n’ ‘\012’

字符串常量

如： “China”



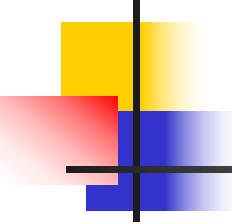
1. 整型常量（整数）

{ 正整数
负整数

{ 十进制整型常量 0~9 1234
八进制整型常量 0~7 023
十六进制整型常量 0~9, A~F 0x2A

2. 实型常量（实数、浮点数）

表示形式 { 小数形式 -12.3
指数形式 -1.2e-2



3. 字符常量

用一对单引号括起来的一个字符

如： ‘a’ ‘B’ ‘>’ ‘?’

注意：

- (1) 其中单引号只作为定界符使用，并不是字符常量的组成部分，即在输出字符常量时，一对单引号并不输出



注意：

- (2) 在C语言中，字符常量具有数值，在ASCII代码表中，‘a’的值为97，‘A’的值为65
- (3) 字符常量在计算机内部以整数常量的形式存放，因此，字符常量与整型常量等价

如： $a = 'D' + 8;$ 等价于 $a = 68 + 6;$
 $b = '?' + 'x';$ 等价于 $b = 63 + 88;$



4. 转义字符常量

在C语言中，允许另一种特殊形式的字符常量，即以一个“\”开头的字符序列。

转义字符都是以“\”开头 p29

如： ‘\n’ ‘\t’ ‘\b’ ‘\r’ ‘\f’
 ‘\\’ ‘\”’ ‘\”’
 ‘\101’ ‘\x4A’



例： 转义字符

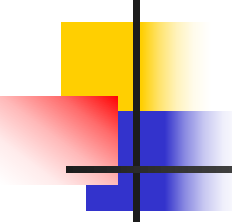
```
main( )  
{  
    printf("boy\t\girl\bk\n");  
    printf("I say:\' Goodby!\' \n");  
}
```

结果： boy girk



8列

I say:' Goodby!'



5. 字符串常量

由一对双引号括起来的字符序列

如: “Hello” “C Program”
“How are you!”

注意:

- (1) 字符串常量与字符常量的区别 如: “a” ‘a’
- (2) 占用内存空间大小不同, 字符串添加 ‘\0’ 作为结束标志
`printf("%d", sizeof("a"));`



第四节 变量

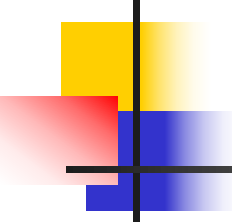
1. 变量和变量的地址

变量——在程序运行过程中，其值可以发生变化的量

变量的地址——定义变量后，变量在内存中存放的位置

如： 变量y的值为6，而y的地址为1800H
则说明1800H存储单元中存放的值为6

y 



2. 变量声明语句

一般形式:

数据类型 变量名;

如: `int age, total;`

注意:

- (1) 数据类型与变量名间至少要用一个空格隔开;多个变量名之间要用“,”隔开



注意：

- (2) 变量必须先定义后使用，通常放在函数体开头部分声明
- (3) 变量一经说明，每个变量就有一个特定的类型，在编译时就对其分配相应的存储空间



3. 整型变量——用于存放整数的变量

基本类型符	int	2/4字节
短整型	short int	2字节
长整型	long int	4字节
无符号型	<div><div>unsigned int</div><div>unsigned short</div><div>unsigned long</div></div>	同相应的有符号数

方法: `printf("%d",sizeof(int));`



4. 实型变量——用于存放实数的变量

{	单精度	float	4字节
	双精度	double	8字节
	长双精度	long double	8字节

如: `float a, b, c; /*定义a, b, c为单精度实型变量*/`
`double x, y; /*定义x, y为双精度实型变量*/`
`long double z; /*定义z为长双精度实型变量*/`

测量两个实数是否相等 $\text{abs}(a-b) < 1e-6$



5. 字符型变量——用于存放字符数据的变量

类型符 **char** 0~255 1个字节

```
如： char ch1, ch2; /*定义ch1, ch2为字符变量*/  
      ch1='a'; /*把字符常量'a'赋给字符变量ch1 */  
      ch2='b'; /*把字符常量'b'赋给字符变量ch2 */
```



注意：

(1) 在0~127内，字符型变量与整型变量可以通用

如： `char c1=97;`

 `int c1='a';`



注意：

(2) 一个字符既可以以字符形式输出，也可以以整数形式输出

例：字符以整数形式输出

```
main( )  
{ char c;  
  int i;  
  c=97;  
  i=' a' ;  
  printf("%c  
%c\n", c, i);  
  printf("%d %d\n", c, i);  
}
```



注意：

(3) 字符数据可以参与算术运算

例①：大小写字母转换

```
main( )  
{ char c1, c2;  
  printf("Please input a letter:");  
  scanf("%c", &c1);  
  c2=c1+'a'-'A';  
  printf("%c\t%c\n", c1, c2);  
}
```



例②:

```
main( )  
{  
    char c;  
    c='b'+'5'-'3';  
    printf("%c\n", c);  
}
```



第五节 变量赋初值

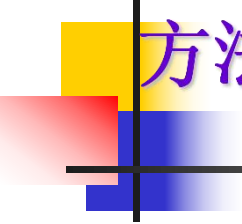
C语言规定，允许在定义变量的同时给变量初始化（即赋初值）

方法1:

```
int a=3;  
float f=3.5;  
char c='e';
```



```
int a;  
a=3;  
-----  
float f;  
f=3.5  
-----  
char c;  
c='e';
```



方法2：被定义变量的一部分赋初值

如： `int a, b, c=5;`

`/*定义a, b, c为整型变量，其中c的值为5*/`

方法3：先定义后赋值

如： `int a, b, c;`

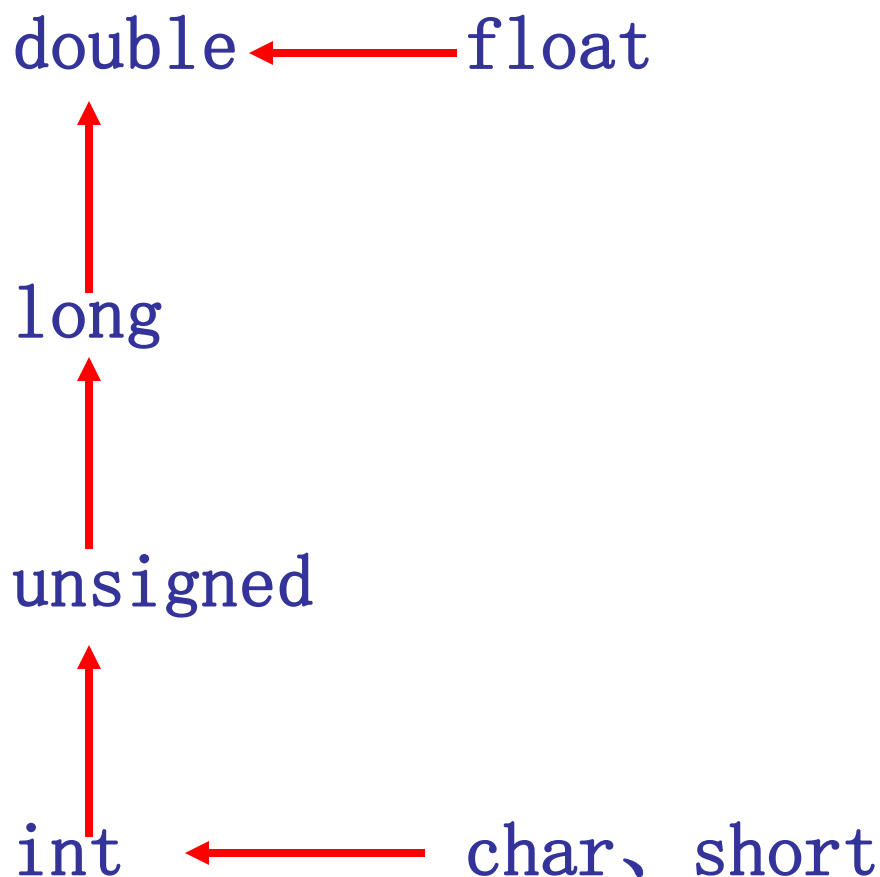
`a=b=c=5;`

但不允许： `int a=b=c=5;` ❌

第六节 各类数值型数据间的混合运算

1. 隐式转换:

转换规则:






例: $10 + 'a' + i * f - d / e$

设: i 为整型 `int`

f 为单精度实型 `float`

d 为双精度实型 `double`

e 为长整型 `long`



2.强制类型转换：（显式转换）

强制类型转换运算符——（ ）
（类型名） 表达式

```
如： int n; /*定义n为整型变量*/  
  
      double d;  
  
      d=(double)n; /*注意n转换后仍为int型*/
```

```
如： (int)x+y; /*将x转换为int型, 再与y相加*/  
  
      (int)(x+y); /*将x+y的值转换为int型*/
```



第七节 算术运算符和算术表达式

一、运算符简介 算术、关系、逻辑、赋值、条件等

二、算术运算符和表达式

1. 基本运算符：+ - * / %

注意：

(1) *不能省，也不能写成×或•

(2) %参与运算量必须为整数，既它只适用于
int和char型

如： 6%4



注意：

(3) /用法与一般数学中用法规则不完全相同

①参与运算的均为整数时，结果也为整数

如： $5/2$ $5/3$ $-7/4$ $8/-5$ $1/2$

②如果运算中有一个是实数，则结果为double型

如： $5.0/2$ $1/2.0$ $-7.0/4$



2. 优先级和结合性

优先级：按运算符的优先级别高低次序执行

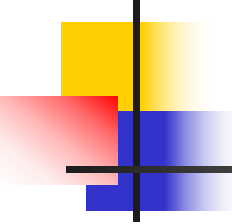
如： $a-b*c+d$

结合性："自左向右"

如： $a+b+c-d$

但也有一些运算符结合方向为"自右向左"

如： $a=b=3$



3. 自增、自减运算符（++、--）

++a, --a: 在使用之前, 先使变量a的值加 (减) 1

a++, a-- : 在使用之后, 再使变量a的值加 (减) 1

注意:

(1) ++a, a++的区别

++a, a++ 都相当于 **a=a+1**

区别在于:

如: a=5; b=++a;
a=5; b=a++;

如: a=5; b=--a;
a=5; b=a--;



注意:

(2) ++, --只能用于变量, 不能用于常量或表达式

如: $2++$ $(2*b-1)--$ ✗

(3) 与其它运算符混合使用时, 应注意结合方向
“自右向左”

如: $-a++ \longleftrightarrow -(a++)$
 $\longleftrightarrow (-a)++$



例1:

```
int i, j, k;  
i=3;  
j=i++;  
k=++i;
```

最后i= 5 j= 3 k= 5, 请填空



例2:

```
main( )
{
    int a, b, c;
    a=3;b=5;
    c=(--a)+(--b) ;
    printf("a=%d, b=%d, c=%d\n", a, b, c) ;
    a=3;b=5;
    c=(a--)+(b--);
    printf("a=%d, b=%d, c=%d\n", a, b, c) ;
}
```



4. 赋值运算符和赋值表达式

(1) 简单的赋值运算符：=

如： `a=5` /*把5赋给变量a*/

注意：

① “=”左边必须是变量，右边可以是变量、常量、函数调用、表达式等

如： `a=5.66` `y=23.5+a*b-c/d`
`b=c` `z=sqrt(3)+2*c`
`i=i+1`



注意：

- ② “=” 与数学中的等号外观相同，但含义、作用不同， $i=i+1$ 在数学中不成立，但在C语言中有意义。
- ③如果 “=” 两侧类型不同，在赋值过程中要进行类型转换

转换规则： 不论赋值号两侧级别高低如何，
一律将赋值号右边的类型转换
成左边的类型，然后再赋值



具体规定:

a. 整型变量=实数

如: `int a=5.99` 则i的值为5.

b. 单、双精度变量=整数

`float x=2;`

c. 整型变量=字符数据

d. 字符变量=整型数据 (低8位) `char ch=321;`

e. 无符号变量=整型数据

`unsigned short a=-1;`
`printf("%u", a);`



(2) 复合的赋值运算符

如: $x+=4$ /*先使 $x+4$, 再赋给 x */

$\longleftrightarrow x=x+4$

如: $b-=5$

$a*=b+2$

$m/=y-15$

$y\%=2$

令 $b=8$ $a=2$ $m=30$ $y=17$

则 $b=\underline{3}$ $a=\underline{10}$ $m=\underline{15}$ $y=\underline{1}$.



(3) 赋值表达式

一般形式：

变量=表达式

如： $a=5$

$b=c*5+12$

表达式的值就是左边变量的值



注意:

- ① “=”右边的表达式又可以是一个赋值表达式
即出现多个 “=” 的情况

如: $a=b=3$ /*把3赋给变量b

最后变量a的值为3*/

如: $a=b=c=d=3$ 表达式值为 3.

$x=(y=6)+3$ 表达式值为 9.

$z=(x=16)*(y=4)$ 表达式值为 64.



注意：

② 赋值表达式还可以包含复合运算符

如： $x+=y*=3$

令 $x=5; y=2$

求表达式最后值为 11.

x 的值为 11.

y 的值为 6.



5. 逗号运算符和逗号表达式: “ , ”

一般形式:

表达式1, 表达式2, 表达式3, ..., 表达式n

如: $2+3, 3*5$

则表达式最后值为 15.

注意:

(1) 优先级最低

如: $x=2+3, x-4$

则表达式最后值为 1.



注意:

(2) 逗号表达式可以嵌套，即表达式_n本身也可以为一逗号表达式

如： $(x=2*5, x-3), x*4$

则表达式最后值为 40 .

(3) 并非所有出现逗号的地方，就可以作为逗号表达式

如： `printf(“%d, %d, %d”, a, b, c);`

`printf(“%d, %d, %d”, (a, b, c), x, y);`



习题

例3:

```
main()
{
    int a=2;
    a%=5-1;
    printf("%d\n", a);
    a+=a*=a-=a*=5;
    printf("%d\n", a);
}
```



习题

例: `main()`

```
{  
    int  a=5, b=2;  
    float  x=4. 5, y=3. 0, u;  
    u=a/3+b*x/y+1/2;  
    printf("%f\n", u);  
}
```



作业

- P41 二、P42 三