



## 第2章 数据类型、运算符与表达式

### 任务1：数据类型及常量变量

主讲人 |

## 知识 目标



1. 掌握常量与变量的书写和定义
2. 掌握三种基本数据类型的应用

## 技能 目标



1. 会编写变量与常量的简单C程序
2. 会进行数制之间的转换
3. 会调试和运行程序

## 素质 目标



1. 学习过程中形成乐于探究的意识和敢于创新的精神。
2. 养成主动与他人交流合作的精神。

# 提纲

1. 基本数据类型

2. 常量与变量

3. 数据类型转换

## 1.基本数据类型



生活中智能储存柜的例子：  
根据物品的大小  
存放到相应大小  
的抽屉中。

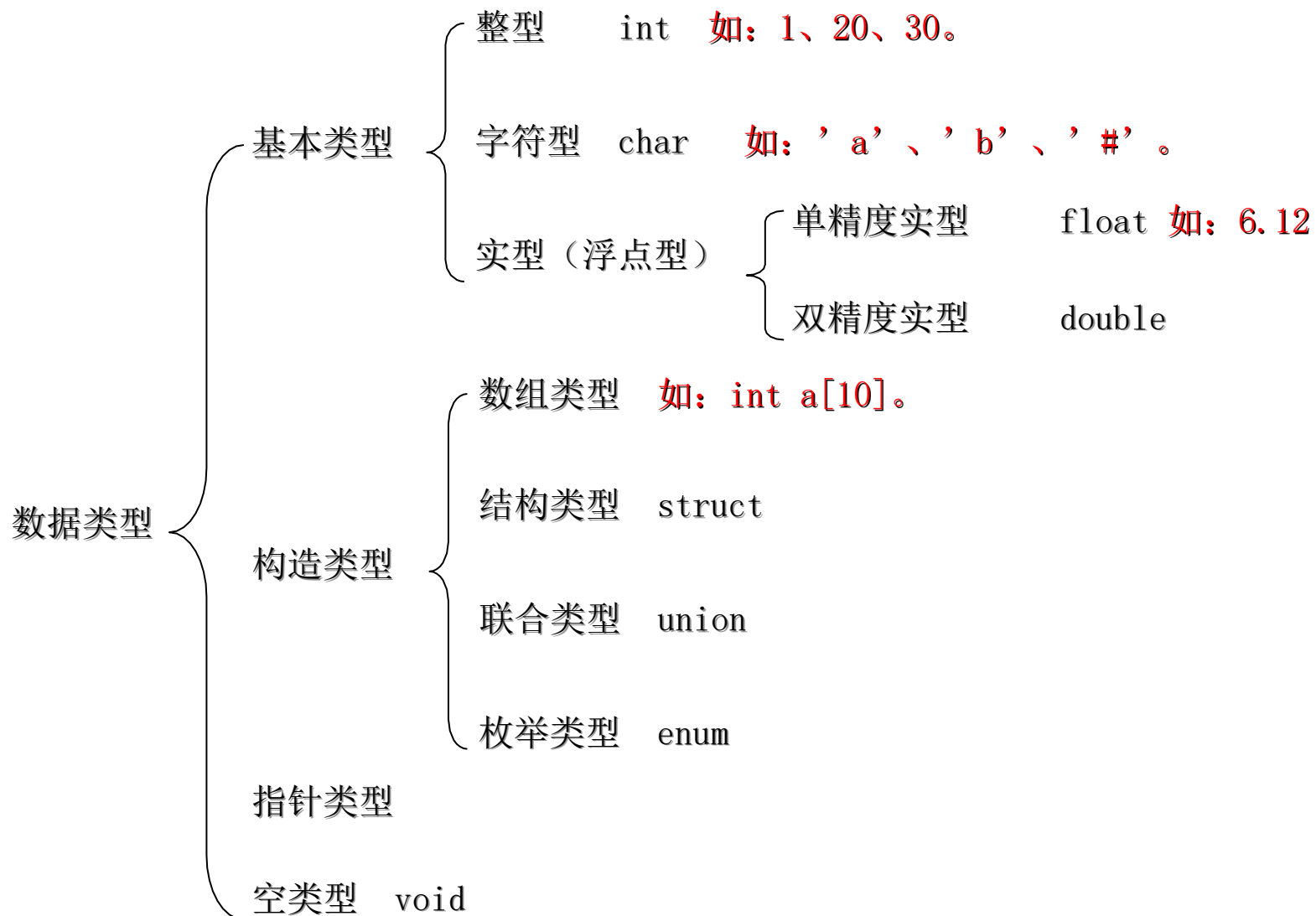


存放大物品

存放中等大小  
物品

存放小物品

# 1.基本数据类型



# 1.基本数据类型



数据类型决定:

- 1. 数据占内存字节数 (1字节=8位)
- 2. 数据取值范围
- 3. 允许的操作。

类型	符号	关键字	所占字节数	数的表示范围
整型	有	(signed)int	4	-2147483648~2147483647
		(signed)short	2	-32768~32767
		(signed)long	4	-2147483648~2147483647
	无	<del>unsigned int</del>	4	<del>0~4294967295</del>
		<del>unsigned short</del>	2	<del>0~65535</del>
		<del>unsigned long</del>	4	<del>0~4294967295</del>
实型	有	float	4	3.4e-38~3.4e38
	有	double	8	1.7e-308~1.7e308
字符型	有	char	1	-128~127
	无	unsigned char	1	0~255



**标识符 ( identifier )** 是对变量名、函数名、标号和其他各种用户定义的对象命名。

**命名规则**：标识符由**字母**、**数字**或者**下划线**组成，且**第一个字符**必须是字母或下划线。

**注意**：

- (1) 标识符区分大小写；**如：a与A不是一个标识符。**
- (2) 标识符的有效长度取决于具体的C编译系统；
- (3) 标识符的书写一般采用具有一定实际含义的单词，这样可提高程序的可读性；
- (4) 标识符不能与C语言的**关键字**同名，也不能与自定义函数或C语言库函数同名；



关键字(keywords)是具有固定名字和特定含义的特殊标识符,也称**保留字**,不允许将它们另作别用。

### 32个关键字:

- (1) 数据类型定义: typedef
- (2) 数据类型: char, double, enum, float, int, long, short, struct, union, unsigned, void, signed, volatile, auto, extern, register, static, const
- (3) 运算符: sizeof
- (4) 语句: break, case, continue, default, do, else, for, goto, if, return, switch, while





## 直接常量

### 1、整型常量：

- (1) 十进制整数：如250、-12，每个数字位是0~9，属于合法的；如058、35D，含有非十进制数码，属于不合法的。
- (2) 八进制整数：**最高位为0**，如十进制的128，用八进制表示为0200，每个数字位是0~7，属于合法的；如256(无前缀0)、02A6(包含了非八进制数码A)、-0256(出现了负号)，属于不合法的。
- (3) 十六进制整数：**以0x或0X开头**，如十进制的128，用十六进制表示为0x80或0X80，每个数字位可以是0~9，A~F，其中A代表10，B代表11...，属于合法的；如3A（无前缀0x）、0x8H（包含了数码H），属于不合法的。



## 补充：进制之间的转换

## 1、十进制转二进制

方法为：十进制数除2取余法，即十进制数除2，余数为权位上的数，得到的商值继续除2，依此步骤继续向下运算直到商为0为止。

	2	150	余数	
	2	75	0	150/2商为75, 余0
	2	37	1	75/2商为37, 余1
	2	18	1	37/2商为18, 余1
	2	9	0	18/2商为9, 余0
	2	4	1	9/2商为4, 余1
	2	2	0	4/2商为2, 余0
	2	1	0	2/2商为1, 余0
		0	1	1/2商为0, 余1

十进制转二进制

从最后一个余数读到第一个

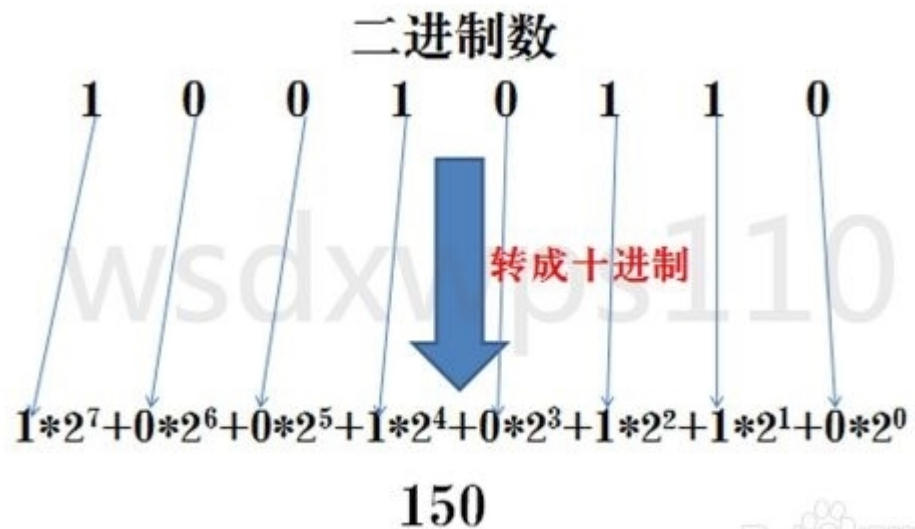
150的二进制数就是：10010110



补充：进制之间的转换

## 2、二进制转十进制

方法为：把二进制数按权展开、相加即得十进制数。

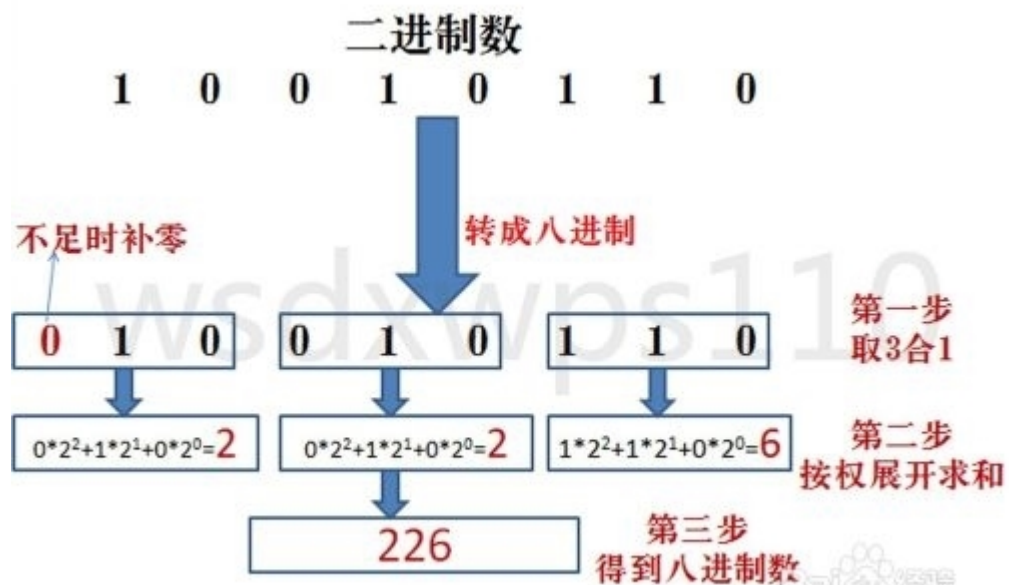




## 补充：进制之间的转换

## 3、二进制转八进制

方法为：3位二进制数按权展开相加得到1位八进制数。（注意事项，3位二进制转成八进制是从右到左开始转换，不足时补0）。

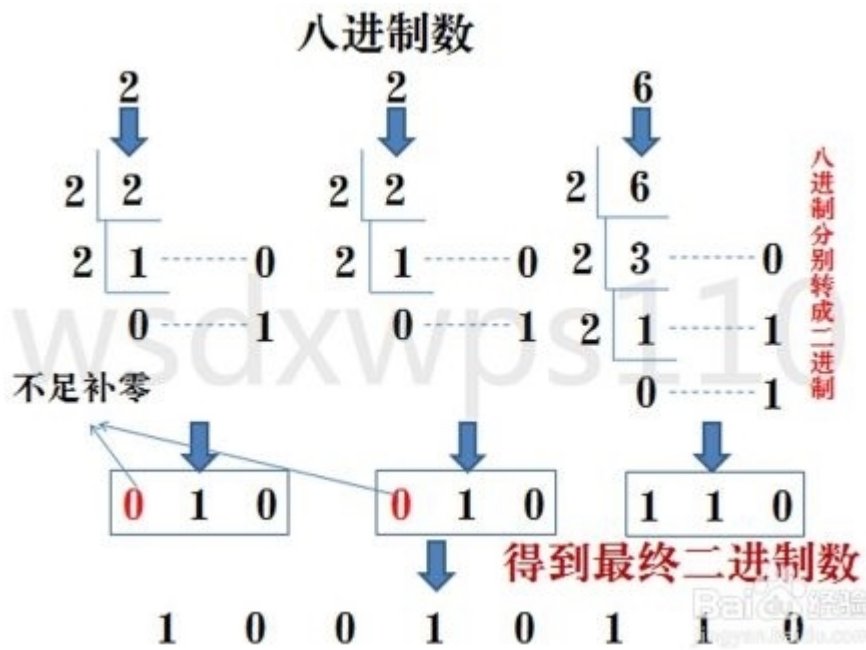




## 补充：进制之间的转换

## 4、八进制转二进制

方法为：八进制数通过除2取余法，得到二进制数，对每个八进制为3个二进制，不足时在最左边补零。

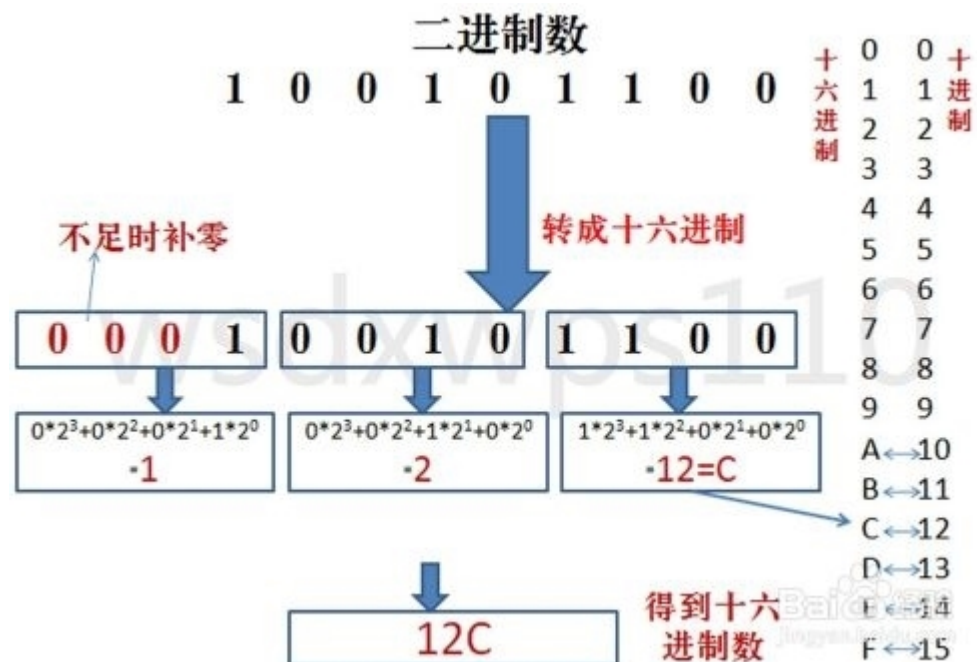




## 补充：进制之间的转换

### 5、二进制转十六进制

方法为：与二进制转八进制方法近似，八进制是取三合一，十六进制是取四合一。（注意事项，4位二进制转成十六进制是从右到左开始转换，不足时补0）。

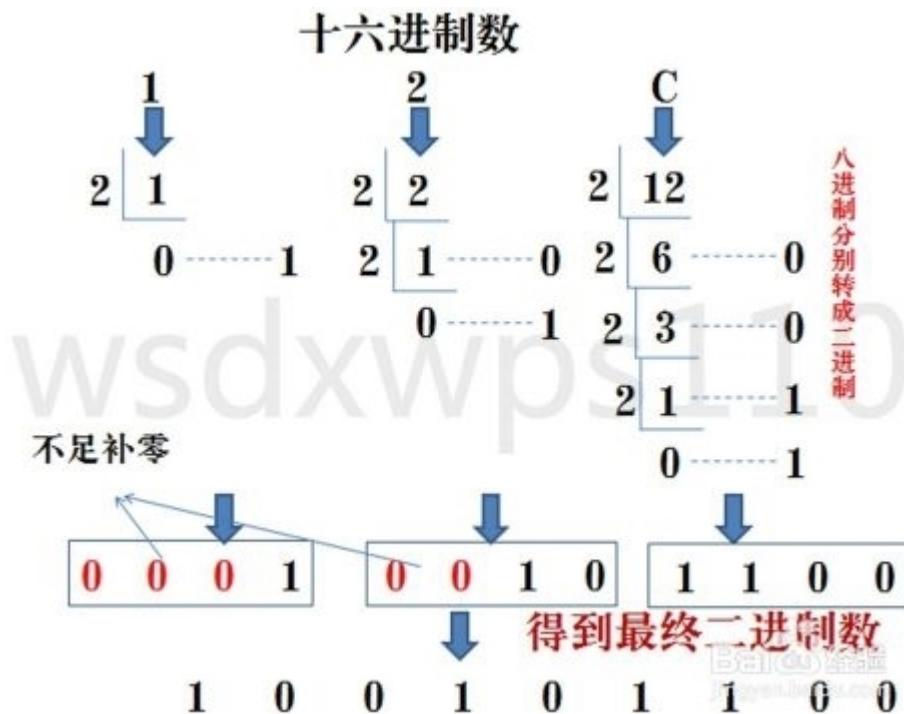




## 补充：进制之间的转换

## 6、十六进制转二进制

方法为：十六进制数通过除2取余法，得到二进制数，对每个十六进制为4个二进制，不足时在最左边补零。





## 直接常量

2、实型常量： 实型也叫浮点型，实型常量也叫实数或浮点数，在C语言中，实数只用十进制表示。

(1) 十进制数形式：（**必须有小数点**），如0.123、.123、123.0、0.0都属于合法的。

(2) 指数形式：（**e或E之前必须有数字，指数必须为整数**），如12.3e3、123E2、1.23e4属于合法的；而 e-5、1.2E-3.5属于不合法的。





## 直接常量

3、字符常量： 字符常量是用单引号括起来的一个字符。

- (1) 字符常量只能用单引号括起来，不能用双引号或其它括号。  
如'a'、'b'、'='、'+','?' 都属于合法的。
- (2) 字符常量只能是单个字符，不能是字符串。
- (3) 字符可以是字符集中任意字符。但数字被定义为字符型之后就不能参与数值运算。  
如'5'和 5 是不同的。



## 直接常量

4、转义字符： 转义字符以反斜线“\”开头，后跟一个或几个字符。

字符形式	功 能
\n	换行
\t	横向跳格（即跳到下一个输出区）
\b	退格
\r	回车
\f	走纸换页
\\	反斜杠字符’\’
\’	单撇号字符
\”	双撇号字符
\a	报警，相当于’\007’
\ddd	1~3位8进制数所代表的字符
\xhh	1~2位16进制数所代表的字符

## 语句编写

```
main()
{
    printf("I love music!\n");
    printf("the music is \"D:\\music\\love.mp3\".\n");
}
```

## 运行结果

```
I love music!
the music is "D:\music\love.mp3".
```



## 符号常量

一般形式: #define 标识符 常量



已知圆的半径为1.5, 求圆的面积。



## 注意说明:

- (1) 宏定义必须以#define开头, 标识符和常量之间不加等于号, 行末不加分号。
- (2) 宏定义#define应该放在函数外部, 这样可以控制到程序结束。
- (3) 符号常量的标识符用**大写字母 (也可以用小写字母)**, 变量标识符用小写字母, 以示区别。

## 语句编写

```
#include "stdio.h"
#define PI 3.14           //符号常量
main()
{
    float r,area;        //定义变量
    r=1.5;               //赋值
    area=PI*r*r;         //计算语句
    printf("area=%f\n",area); //输出结果
}
```

## 运行结果

```
area=7.065
```

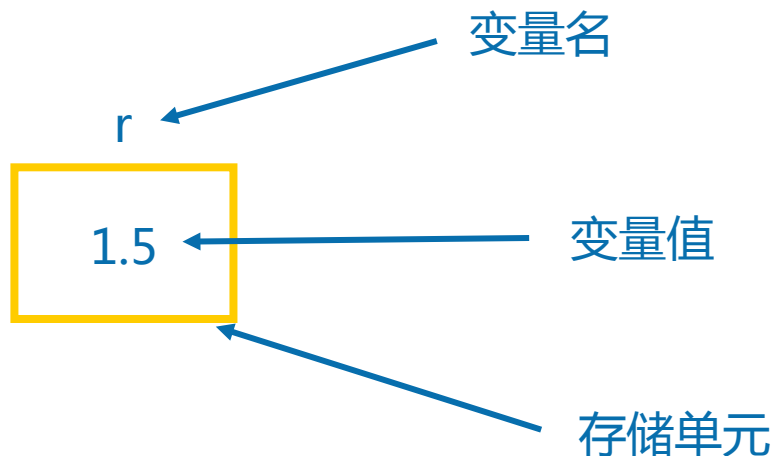


## 变量描述

变量是其值可以变化的量，在使用之前必须先定义，变量在内存中占据一定的存储单元。

例题：

定义一个变量r，那么内存中就应该开辟一个空间，可以存储数据。





## 变量分类

### 1、整型变量

#### (1) 整型变量的分类

基本型：类型说明符为int，在内存中占4个字节。

短整型：类型说明符为short int或者short，在内存中占2个字节。

长整型：类型说明符为long int或者long，在内存中占4个字节。

无符号型：类型说明符为unsigned。

注：无符号型也可以和上述三种类型匹配使用。



## 变量分类

### 1、整型变量

#### (2) 整型变量的定义

整型变量的定义一般形式为：

类型说明符 变量1[=值1]，变量2[=值2]，...；

例题：

```
int a,b;    //a,b为整型变量  
long x,y;   //x,y 为长整型变量
```

注意说明：

- 类型说明符与变量名之间至少要有一个空格间隔。
- 在一个类型说明符后，可定义多个相同类型的变量，但变量间要用逗号间隔。
- 最后一个变量名后必须用分号“；”结束。
- []内的为可选项，即为变量的初始化。

#### 语句编写

```
main()  
{  
    int x,y,z;  
    x=2;  
    y=3;  
    z=x+y;  
    printf("x+y=%d\n",z);  
}
```

#### 运行结果

```
x+y=5
```



## 变量分类

### 2、实型变量

#### (1) 实型变量的分类

单精度：类型说明符为float，在内存中占4个字节。

双精度：类型说明符为double，在内存中占8个字节。

#### (2) 实型变量的定义

实型变量的定义一般形式为：

类型说明符 变量1[=值1]，变量2[=值2]，...；

例题：

```
float a, b, c;    //a, b, c为单精度实型变量  
double x, y, z;  //x, y, z为双精度实型变量
```



已知圆的半径，求圆的面积。



注意说明：

- (1) 实型变量由有限的存储单元组成，能提供的有效数字有限，这样就存在舍入误差。
- (2) 一个单精度实型变量只能保证**7位有效数字（指整数部分与小数部分共7位，不包括小数点）**，后面的数字是无意义的，并不准确地表示该数。应当避免将一个很大的数和一个很小的数直接相加或相减，否则就会“丢失”小的数。
- (3) %f格式输出6位小数，但不能保证这6位小数全部有效，需要看整数和小数的位数之和。

#### 语句编写

```
main()
{
    float r=1.5,pi,area;
    pi=3.14;
    area=pi*r*r;
    printf("area=%f\n",area);
}
```

#### 运行结果

```
area=7.065000
```





## 变量分类

### 3、字符变量

#### (1) 字符变量的定义

字符变量的定义一般形式为：

类型说明符 变量1[=值1]，变量2[=值2]，...；

例题：

```
char a, b; //a, b为字符变量
```

注意说明：

- 字符变量用来存放字符常量，只能放一个字符。
- C语言允许对整型变量赋以字符值，也允许对字符变量赋以整型值。
- 允许把字符变量按整型量输出，也允许把整型量按字符量输出。

#### 语句编写

```
main()
{
    int m;
    char c;
    m='A';    //字符赋值给整型变量
    c=66;     //整数赋值给字符变量
    printf("%c,%d\n",m,m);
    printf("%c,%d\n",c,c);
}
```

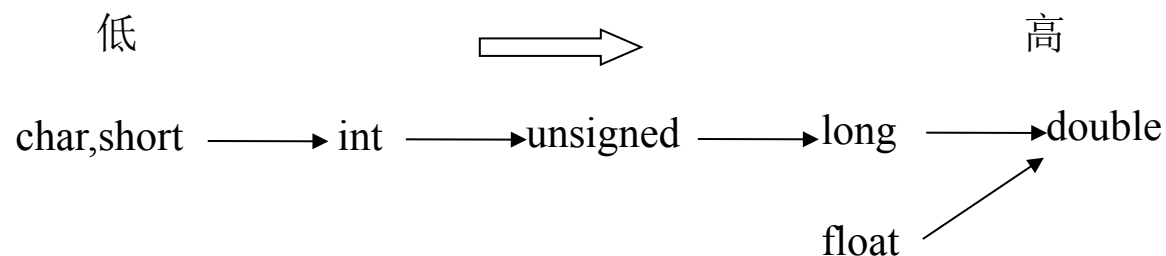
#### 运行结果

```
A, 65
B, 66
```



## 转换规则

- (1) 若参与运算量的类型不同，则先转换成同一类型，然后进行运算。
- (2) 转换按数据长度增加的方向进行，以保证精度不降低。如int型和long型运算时，先把int量转成long型后再进行运算。
- (3) 所有的浮点运算都是以双精度进行的，即使仅含float单精度量运算的表达式，也要先转换成double型，再作运算。
- (4) char型和short型参与运算时，必须先转换成int型。
- (5) 在赋值运算中，两边的数据类型不同时，赋值号右边量的类型将转换为左边量的类型。





假设已指定*i*为整型变量，*f*为float型变量，*d*为double型变量，*e*为long型变量，表达式为： $10+'a'+i*f-d/e$ ，则表达式的运行次序？

运行次序：

第一步运行： $i*f$

第二步运行： $d/e$

第三步运行： $10+'a'$

第四步运行： $10+'a'+i*f$

第五步运行： $10+'a'+i*f-d/e$

#### 语句编写

```
main()
{ int i=2;
  float f=1.5;
  double d=4.0;
  long e=2;
  printf("i*f=%f,i转化为float类型参与运算",i*f);
  printf("d/e=%lf,e转化为double类型参与运算",d/e);
  printf("10+'a'=%d,'a'转化为int类型参与运算",10+'a');
  printf("10+'a'+i*f=%f,int转化为float类型",10+'a'+i*f);
  printf("10+'a'+i*f-d/e=%lf,float转化为double类型",10+'a'+i*f-d/e);
}
```

#### 运行结果

```
i*f=3.000000,i转化为float类型参与运算
d/e=2.000000,e转化为double类型参与运算
10+'a'=107,'a'转化为int类型参与运算
10+'a'+i*f=110.000000,int转化为float类型
10+'a'+i*f-d/e=108.000000,float转化为double类型
```



## 一般形式

(类型说明符) (表达式)

## 例题:

(double)a 将变量a强制转换为double类型  
(int) (x+y) 将x+y的值强制转换为int类型  
(float) (5%3) 将5%3的值强制转换为float类型  
(float)x/y 将x强制转换成float类型后, 再参与运算

## 注意说明:

- 表达式应该用括号括起来。
- 进行强制类型转换时, 得到的是一个所需类型的中间变量, 原来变量的类型并未发生改变。

## 语句编写

```
main()
{
    float f;
    f=9.8;
    printf("(int)f=%d,f=%f\n", (int)f, f);
}
```

## 运行结果

```
(int)f=9,f=9.8
```

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：  
<https://d.book118.com/858103006141006076>