9.2 字符与字符串

C语言中常常出现字符和字符串，而一串字符或者字符串其实就是数组。数组的每一个元素都是char类型，叫做字符数组，或者字符串。

9.2.1 字符数组

1. 字符数组的定义

（1）一维字符数组：用于存储和处理1个字符串，其定义格式与一维数值数组一样。

例：char  a[20];

（2）二维字符数组：用于同时存储和处理多个字符串，其定义格式与二维数值数组一样。

例： char b[10][20];

其中b[i]表示第i个字符串

2. 字符数组的初始化

char arr[ 5 ]={'h','e','l','l','o'};

char arr1[ ]="hello";

字符数组的赋值只能对其元素一一赋值

二维数组初始化时可以省略行长度，不能省略列长度。

3. 字符数组的输入与输出

（1）字符数组的输入

除了可以通过初始化使字符数组各元素得到初值外，也可以使用

getchar()或scanf()函数输入字符。

例1：char a[5];

//请输入5个字符串，每个长度不超过100

for(i=0;i<5;i++)

{

scanf("%c ",&a[i]);

}

例2：char a;

a=getchar();

（2）字符数组的输出

字符数组的输出，可以用putchar()或printf()输出。

例1：for(i=0;i<5;i++)

{

printf("%c",a[i]);

}

例2：putchar(a);

注意：

逐个字符输入、输出时，要指出元素的下标，而且使用“%c”格式符。

9.2.2 字符串

字符数组和字符串的区别是：字符串的末尾有一个空字符 ‘ **\0’**

1. 字符串及其结束标志

字符串，是指若干有效字符的序列。

C语言中的字符串，可以包括字母、数字、专用字符、转义字符等。

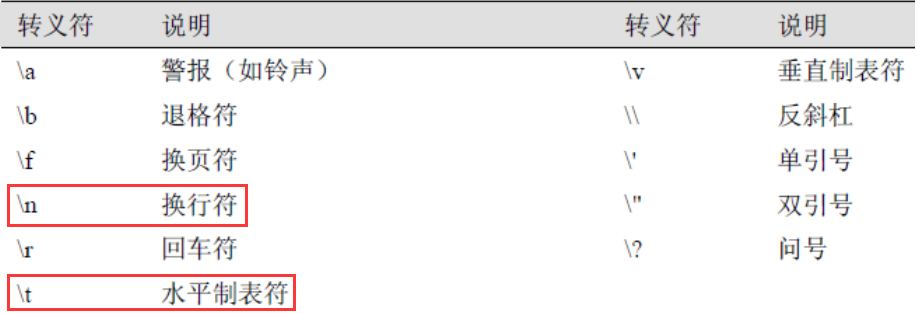
C语言规定：以‘\0’作为字符串结束标志。

2. 字符型字面值：

字符型字面值是用单引号括起来的一个或几个特殊字符。

（1） 执行字符集： ‘a’、 ‘A’、 ‘0’、 ‘9’、 ‘$’等

（2） 转义字符：



（3）. ASCII码，a的整数编码为97，A的整数编码为65，0的整数编码为48。

任何一个小写字符比其大写字符多32，任何一个字符字母后一个总比前一个大1.

3. 字符串字面值：（1）字符串字面值须以字符‘\0’结束

（2）数字转义字符十六进制表示的一般形式为‘\xdd’，即用\x（小写）和两位十六进制的数字表示一个字符。

（3）数字转义字符\abc是8进制表示“a”

**（八进制最多取3位，十六进制最多取2位）**

例：

char c=88 一个字节

char c=‘88’ char c=“\88”1个字节(单引号表示字符)

char c=“88” char c=“\88\0” 2个字节（双引号表示字符串，最后加\0）

4. 字符串的输入

除了可以通过初始化使字符数组各元素得到初值外，也可以使用scanf()函数输入字符串，但是字符串有特定的gets（）函数。

（1）调用方式：gets(字符数组)

（2）函数功能：读取1个字符串（可以包含空格），并将其存储到字符数组中去。

（3）注意

① gets()读取的字符串，其长度没有限制。

② 该函数输入的字符串中允许包含空格，而scanf()函数不允许，scanf遇空格结束。

5. 字符串的输出

除了我们熟知的printf函数外，输出字符串还可以使用puts（）函数。

（1）调用方式：puts(字符数组)

（2）函数功能：字符数组中所存放的字符串，用‘\n’取代字符串的结束标志‘\0’。所以用puts()函数输出字符串时，不要求另加换行符。

（ 3）使用说明

1. 字符串中允许包含转义字符，输出时产生一个控制操作。

② 该函数一次只能输出一个字符串，而printf()函数也能用来输出字符串，且一次能输出多个。

6. 字符串所占字节数sizeof：有多少个字符就占多少个字节，包括转义字符（ **\0**）

字符串长度strlen：从字符串第一个字符到第一个‘ **\0**’的位置之前有几个字符（不包括\0）

7. %s的打印规则是从打印变量的地址开始，直到遇见的第一个‘ \0’ 结束