

C 语言字符数组和字符串

用来存放字符的数组称为字符数组，例如：

```
1. char a[10]; //一维字符数组
2. char b[5][10]; //二维字符数组
3. char c[20]={'c', ' ', 'p', 'r', 'o', 'g', 'r', 'a', 'm'}; // 给部分数组元素赋值
4. char d[]={'c', ' ', 'p', 'r', 'o', 'g', 'r', 'a', 'm' }; //对全体元素赋值时可以省去长度
```

字符数组实际上是一系列字符的集合，也就是字符串 (String)。在 C 语言中，没有专门的字符串变量，没有 string 类型，通常就用一个字符数组来存放一个字符串。

C 语言规定，可以将字符串直接赋值给字符数组，例如：

```
1. char str[30] = {"c.biancheng.net"};
2. char str[30] = "c.biancheng.net"; //这种形式更加简洁，实际开发中常用
```

数组第 0 个元素为 'c'，第 1 个元素为 ' '，第 2 个元素为 'b'，后面的元素以此类推。也可以不指定数组长度，例如：

```
1. char str[] = {"c.biancheng.net"};
2. char str[] = "c.biancheng.net"; //这种形式更加简洁，实际开发中常用
```

在 C 语言中，字符串总是以 '\0' 作为串的结束符。上面的两个字符串，编译器已经在末尾自动添加了 '\0'。

'\0' 是 ASCII 码表中的第 0 个字符，用 NUL 表示，称为空字符。该字符既不能显示，也不是控制字符，输出该字符不会有任何效果，它在 C 语言中仅作为字符串的结束标志。

puts 和 printf 在输出字符串时会逐个扫描字符，直到遇见 '\0' 才结束输出。请看下面的例子：

```
1. #include <stdio.h>
2. int main() {
3.     int i;
4.     char str1[30] = "http://c.biancheng.net";
5.     char str2[] = "C Language";
6.     char str3[30] = "You are a good\0 boy!";
7.     printf("str1: %s\n", str1);
8.     printf("str2: %s\n", str2);
```

```
9.     printf("str3: %s\n", str3);
10.    return 0;
11. }
```

运行结果：

str1: http://c.biancheng.net

str2: C Language

str3: You are a good

str1 和 str2 很好理解，编译器会在字符串最后自动添加 '\0'，并且数组足够大，所以会输出整个字符串。对于 str3，由于字符串中间存在 '\0'，printf() 扫描到这里就认为字符串结束了，所以不会输出后面的内容。

需要注意的是，用字符串给字符数组赋值时由于要添加结束符 '\0'，数组的长度要比字符串的长度（字符串长度不包括 '\0'）大 1。例如：

```
char str[] = "C program";
```

该数组在内存中的实际存放情况为：

C		p	r	o	g	r	a	m	\0
---	--	---	---	---	---	---	---	---	----

字符串长度为 9，数组长度为 10。