C 语言字符串的输入输出

字符串的输出

在 C 语言中，输出字符串的函数有两个：

puts()：直接输出字符串，并且只能输出字符串。

printf()：通过格式控制符 %s 输出字符串。除了字符串，printf() 还能输出其他类型的数据。

这两个函数前面已经讲过了，这里不妨再演示一下，请看下面的代码：

```
#include <stdio.h>
int main(){
    int i;
    char str[] = "http://c.biancheng.net";
    printf("%s\n", str);  //通过变量输出
    printf("%s\n", "http://c.biancheng.net");  //直接输出
    puts(str);  //通过变量输出
    puts("http://c.biancheng.net");  //直接输出
    return 0;
}
```

运行结果：

```
http://c.biancheng.net
http://c.biancheng.net
http://c.biancheng.net
http://c.biancheng.net
```

在 printf() 函数中使用%s 输出字符串时，在变量列表中给出数组名即可，不能写为 printf("%s", str[]);。

字符串的输入

在 C 语言中，输入字符串的函数有两个：

scanf()：通过格式控制符 %s 输入字符串。除了字符串，scanf() 还能输入其他类型的数据。

gets()：直接输入字符串，并且只能输入字符串。

1) 使用 scanf() 读取字符串

请先看下面的例子：

```
#include <stdio.h>
int main(){
    char str1[30], str2[30];
    printf("Input str1: ");
    scanf("%s", str1);
    printf("Input str2: ");
    scanf("%s", str2);
    printf("str1: %s\nstr2: %s\n", str1, str2);
    return 0;
}
```

运行结果：

```
Input str1: c.biancheng.net↙
Input str2: Java Python C-Sharp↙
str1: c.biancheng.net
str2: Java
```

由于字符数组长度为 30，因此输入的字符串长度必须小于 30，以留出一个字节用于存放字符串结束标志`\0`。

对程序的说明：

① 我们本来希望将 "Java Python C-Sharp" 赋值给 str2，但是 scanf() 只读取到 "Java"，这是因为 scanf() 读取到空格时就认为字符串输入结束了，不会继续读取了。请看下面的例子：

```c
#include <stdio.h>
int main(){
    char str1[20], str2[20], str3[20];
    printf("Input string: ");
    scanf("%s", str1);
    scanf("%s", str2);
    scanf("%s", str3);
    printf("str1: %s\nstr2: %s\nstr3: %s\n", str1, str2, str3);
    return 0;
}
```

运行结果：

```
Input string: Java Python C-Sharp↙
str1: Java
str2: Python
str3: C-Sharp
```

第一个 scanf() 读取到 "Java" 后遇到空格，结束读取，将"Python C-Sharp" 留在缓冲区。第二个 scanf() 直接从缓冲区中读取，不会等待用户输入，读取到 "Python" 后遇到空格 ,结束读取 ,将 "C-Sharp" 留在缓冲区。第三个 scanf() 读取缓冲区中剩下的内容。

②scanf 的各个变量前面要加取地址符&，用以获得变量的地址，例如：

```c
int a, b;
scanf("%d %d", &a, &b);
```

但是在本节的示例中，将字符串读入字符数组却没有使用&，例如：

```c
char str1[20], str2[20], str3[20], str4[20];
scanf("%s %s %s %s",str1, str2, str3, str4);
```

这是因为 C 语言规定，数组名就代表了该数组的地址。整个数组是一块连续的内存单元，如有字符数组 char c[10]，在内存可表示为：

C[0]	C[1]	C[2]	C[3]	C[4]	C[5]	C[6]	C[7]	C[8]	C[9]

C 语言还规定，数组名所代表的地址为第 0 个元素的地址，例如 `char c[10];`，`c` 就代表 `c[0]`的地址。第 0 个元素的地址就是数组的起始地址，称为首地址。也就是说，数组名表示数组的首地址。

设数组 c 的首地址为 0X2000，也即 c[0]地址为 0X2000，则数组名 c 就代表这个地址。因为 c 已经表示地址，所以在 c 前面不能再加取地址符&，例如写作 `scanf("%s",&c);`是错误的。

有了首地址，有了字符串结束符'\0'，就可以在内存中完整定位一个字符串了。例如：

```
printf("%s", c);
```

printf 函数会根据数组名找到 c 的首地址，然后逐个输出数组中各个字符直到遇到 '\0' 为止。

int、float、char 类型的变量表示数据本身，数据就保存在变量中；而数组名表示的是数组的首地址，数组保存在其他内存单元，数组名保存的是这块内存的首地址。后面我们会讲解指针，大家将会有更加深刻的理解。

2) 使用 gets() 读取字符串

gets 是 get string 的缩写，意思是获取用户从键盘输入的字符串，语法格式为：

```
gets(arrayName);
```

arrayName 为字符数组。从键盘获得的字符串，将保存在 arrayName 中。请看下面的例子：

```
1. #include <stdio.h>
2. int main(){
3.     char str1[30], str2[30];
4.     printf("Input str1: ");
```

```
    gets(str1);
    printf("Input str2: ");
    gets(str2);
    printf("str1: %s\nstr2: %s\n", str1, str2);

    return 0;
}
```

运行结果：

```
Input str1: Java Python C-Sharp↙
Input str2: http://c.biancheng.net↙
str1: Java Python C-Sharp
str2: http://c.biancheng.net
```

可以发现，当输入的字符串中含有空格时，输出仍为全部字符串，这说明 gets() 函数不会把空格作为输入结束的标志，而只把回车换行作为输入结束的标志，这与 scanf() 函数是不同的。

总结：如果希望读取的字符串中不包含空格，那么使用 scanf() 函数；如果希望获取整行字符串，那么使用 gets() 函数，它能避免空格的截断。