

C 输入 & 输出

当我们提到**输入**时，这意味着要向程序填充一些数据。输入可以是以文件的形式或从命令行中进行。C 语言提供了一系列内置的函数来读取给定的输入，并根据需要填充到程序中。

当我们提到**输出**时，这意味着要在屏幕上、打印机上或任意文件中显示一些数据。C 语言提供了一系列内置的函数来输出数据到计算机屏幕上和保存数据到文本文件或二进制文件中。

标准文件

C 语言把所有的设备都当作文件。所以设备（比如显示器）被处理的方式与文件相同。以下三个文件会在程序执行时自动打开，以便访问键盘和屏幕。

标准文件	文件指针	设备
标准输入	stdin	键盘
标准输出	stdout	屏幕
标准错误	stderr	您的屏幕

文件指针是访问文件的方式，本节将讲解如何从屏幕读取值以及如何把结果输出到屏幕上。

C 语言中的 I/O (输入/输出) 通常使用 printf() 和 scanf() 两个函数。

scanf() 函数用于从标准输入（键盘）读取并格式化，printf() 函数发送格式化输出到标准输出（屏幕）。

实例

```
#include <stdio.h> // 执行 printf() 函数需要该库
int main()
{
    printf("菜鸟教程"); //显示引号中的内容
    return 0;
}
```

编译以上程序，输出结果为：

菜鸟教程

实例解析：

- 所有的 C 语言程序都需要包含 **main()** 函数。代码从 **main()** 函数开始执行。
- **printf()** 用于格式化输出到屏幕。**printf()** 函数在 **"stdio.h"** 头文件中声明。
- **stdio.h** 是一个头文件 (标准输入输出头文件) and **#include** 是一个预处理命令，用来引入头文件。当编译器遇到 **printf()** 函数时，如果没有找到 **stdio.h** 头文件，会发生编译错误。
- **return 0;** 语句用于表示退出程序。

%d 格式化输出整数

```
#include <stdio.h>
int main()
{
    int testInteger = 5;
    printf("Number = %d", testInteger);
    return 0;
}
```

编译以上程序，输出结果为：

```
Number = 5
```

在 printf() 函数的引号中使用 "%d" (整型) 来匹配整型变量 testInteger 并输出到屏幕。

%f 格式化输出浮点型数据

```
#include <stdio.h>
int main()
{
    float f;
    printf("Enter a number: ");
    // %f 匹配浮点型数据
    scanf("%f",&f);
    printf("Value = %f", f);
    return 0;
}
```

getchar() & putchar() 函数

int getchar(void) 函数从屏幕读取下一个可用的字符，并把它返回为一个整数。这个函数在同一个时间内只会读取一个单一的字符。您可以在循环内使用这个方法，以便从屏幕上读取多个字符。

int putchar(int c) 函数把字符输出到屏幕上，并返回相同的字符。这个函数在同一个时间内只会输出一个单一的字符。您可以在循环内使用这个方法，以便在屏幕上输出多个字符。

请看下面的实例：

实例

```
#include <stdio.h>
int main( )
{
    int c;
    printf( "Enter a value :");
    c = getchar( );
    printf( "\nYou entered: ");
    putchar( c );
    printf( "\n");
    return 0;
}
```

当上面的代码被编译和执行时，它会等待您输入一些文本，当您输入一个文本并按下回车键时，程序会继续并只会读取一个单一的字符，显示如下：

```
$/a.out
Enter a value :runoob

You entered: r
```

gets() & puts() 函数

char *gets(char *s) 函数从 **stdin** 读取一行到 **s** 所指向的缓冲区，直到一个终止符或 EOF。

int puts(const char *s) 函数把字符串 **s** 和一个尾随的换行符写入到 **stdout**。

实例

```
#include <stdio.h>
int main( )
{
    char str[100];
    printf( "Enter a value :");
    gets( str );
    printf( "\nYou entered: ");
    puts( str );
    return 0;
}
```

当上面的代码被编译和执行时，它会等待您输入一些文本，当您输入一个文本并按下回车键时，程序会继续并读取一整行直到该行结束，显示如下：

```
$/a.out
Enter a value :runoob

You entered: runoob
```

scanf() 和 printf() 函数

int scanf(const char *format, ...) 函数从标准输入流 **stdin** 读取输入，并根据提供的 **format** 来浏览输入。

int printf(const char *format, ...) 函数把输出写入到标准输出流 **stdout**，并根据提供的格式产生输出。

format 可以是一个简单的常量字符串，但是您可以分别指定 **%s**、**%d**、**%c**、**%f** 等来输出或读取字符串、整数、字符或浮点数。还有许多其他可用的格式选项，可以根据需要使用。如需了解完整的细节，可以查看这些函数的参考手册。现在让我们通过下面这个简单的实例来加深理解：

实例

```
#include <stdio.h>
int main( ) {
    char str[100];
```

```
int i;
printf( "Enter a value :");
scanf("%s %d", str, &i);
printf( "\nYou entered: %s %d ", str, i);
printf("\n");
return 0;
}
```

当上面的代码被编译和执行时，它会等待您输入一些文本，当您输入一个文本并按下回车键时，程序会继续并读取输入，显示如下：

```
$/a.out
Enter a value :runoob 123

You entered: runoob 123
```

在这里，应当指出的是，scanf() 期待输入的格式与您给出的 %s 和 %d 相同，这意味着您必须提供有效的输入，比如 "string integer"，如果您提供的是 "string string" 或 "integer integer"，它会被认为是错误的输入。另外，在读取字符串时，只要遇到一个空格，scanf() 就会停止读取，所以 "this is test" 对 scanf() 来说是三个字符串。

[← C typedef](#)[C 文件读写 →](#)**8 篇笔记**** 写笔记**