

C 库函数 - scanf()

 C 标准库 - <stdio.h>

描述

C 库函数 `int scanf(const char *format, ...)` 从标准输入 `stdin` 读取格式化输入。

声明

下面是 `scanf()` 函数的声明。

```
int scanf(const char *format, ...)
```

参数

● `format` – 这是 C 字符串，包含了以下各项中的一个或多个：空格字符、非空格字符和 *format 说明符*。

`format` 说明符形式为：

```
[%h*](width)[modifiers]type-]
```

具体讲解如下：

参数	描述
*	这是一个可选的星号，表示数据是从流 <code>stream</code> 中读取的，但是可以被忽略，即它不存储在对应的参数中。
width	这指定了在当前读取操作中读取的最大字符数。
modifiers	为对应的附加参数所指向的数据指定一个不同于整型（针对 <code>d</code> 、 <code>i</code> 和 <code>n</code> ）、无符号整型（针对 <code>o</code> 、 <code>u</code> 和 <code>x</code> ）或浮点型（针对 <code>e</code> 、 <code>f</code> 和 <code>g</code> ）的大小： <code>h</code> ：短整型（针对 <code>d</code> 、 <code>i</code> 和 <code>n</code> ），或无符号短整型（针对 <code>o</code> 、 <code>u</code> 和 <code>x</code> ）； <code>l</code> ：长整型（针对 <code>d</code> 、 <code>i</code> 和 <code>n</code> ），或无符号长整型（针对 <code>o</code> 、 <code>u</code> 和 <code>x</code> ），或双精度型（针对 <code>e</code> 、 <code>f</code> 和 <code>g</code> ） <code>L</code> ：长双精度型（针对 <code>e</code> 、 <code>f</code> 和 <code>g</code> ）
type	一个字符，指定了要被读取的数据类型以及数据读取方式。具体参见下一个表格。

scanf 类型说明符：

类型	合格的输入	参数的类型
%a, %A	读入一个浮点值(仅 C99 有效)。	float *
%c	单个字符：读取下一个字符。如果指定了一个不为 1 的宽度 <code>width</code> ，函数会读取 <code>width</code> 个字符，并通 过参数传递，把它们存储在数组中连续位置。在末尾不会追加空字符。	char *
%d	十进制整数：数字前面的 + 或 - 号是可选的。	int *
%e, %E, %f, %F, %g, %G	浮点数：包含了一个小数点、一个可选的前置符号 + 或 -、一个可选的后置字符 <code>e</code> 或 <code>E</code> ，以及一个十 进制数字。两个有效的实例 -732.103 和 7.12e4	float *
%i	读入十进制，八进制，十六进制整数。	int *
%o	八进制整数。	int *
%s	字符串。这将读取连续字符，直到遇到一个空格字符（空格字符可以是空白、换行和制表符），	char *
%u	无符号的十进制整数。	unsigned int *
%x, %X	十六进制整数。	int *
%p	读入一个指针。	
%[]	扫描字符集合。	
%%	读 % 符号。	

● **附加参数** – 根据不同的 `format` 字符串，函数可能需要一系列的附加参数。每个参数包含了一个要被插入的值，替换了 `format` 参数中指定的每个 % 标志。参数的个数应与 % 标志的个数相同。

返回值

如果成功，该函数返回成功匹配和赋值的个数。如果到达文件末尾或发生读错误，则返回 EOF。

实例

下面的实例演示了 `scanf()` 函数的用法。

```
实例

#include<stdio.h>
int main(void)
{
    int a,b,c;

    printf("请输入三个数字: ");
    scanf("%d%d%d",&a,&b,&c);
    printf("%d,%d,%d\n",a,b,c);
    return 0;
}
```

让我们编译并运行上面的程序，这将在交互模式下产生以下结果：

```
请输入三个数字: 1 2 3
1,2,3
```

解析说明：

- 1、`&a`、`&b`、`&c` 中的 `&` 是地址运算符，分别获得这三个变量的内存地址。
- 2、`%d%d%d` 是按十进制格式输入三个数值。输入时，在两个数据之间可以用一个或多个空格、tab 键、回车键分隔。

如果使用 `␣` 来分隔输入的 `%d`，相应的输入时也需要添加 `␣`：

```
实例

#include<stdio.h>
int main(void)
{
    int a,b,c;

    printf("请输入三个数字: ");
    scanf("%d, %d, %d",&a,&b,&c);
    printf("%d, %d, %d\n",a,b,c);
    return 0;
}
```

让我们编译并运行上面的程序，这将在交互模式下产生以下结果：

```
请输入三个数字: 1, 2, 3
1, 2, 3
```

注意：输入时 `␣` 前一定要紧跟在数字后面，数字与 `␣` 之间不能有空格。
在用 `%c` 输入时，空格和“转义字符”均作为有效字符。

```
实例

#include<stdio.h>
int main(void)
{
    char a,b,c;

    printf("请输入三个字符: ");
    scanf("%c%c%c",&a,&b,&c);
    printf("%c,%c,%c\n", a,b,c);
    return 0;
}
```

让我们编译并运行上面的程序，这将在交互模式下产生以下结果：

```
$ ./a.out
请输入三个字符: run
r,u,n
$ ./a.out
请输入三个字符: r u n
r, ,u
```

以下实例演示的是接收字符串：

```
实例

#include <stdio.h>

int main()
{
    char str1[20], str2[30];

    printf("请输入用户名: ");
    scanf("%s", str1);

    printf("请输入您的网站: ");
    scanf("%s", str2);

    printf("输入的用户名: %s\n", str1);
    printf("输入的网站: %s", str2);

    return(0);
}
```


让我们编译并运行上面的程序，这将在交互模式下产生以下结果：

```
请输入用户名: admin
请输入您的网站: www.runoob.com
输入的用户名: admin
输入的网站: www.runoob.com
```

 C 标准库 - <stdio.h>

2 篇笔记


✎ 写笔记

 如上代码，报错：返回值被忽略: "scanf"
解决方案：

1) 把 `scanf` 换为 `scanf_s`

111 2) 在前面加上 `#pragma warning(disable:4996)`

 learner 3年前 (2018-11-03)

 在较新版本 vs 中输入字符串时，需要给 `scanf_s` 函数制定一个参数，告诉编译器预留多大的空间来存储字符串
`sizeof()` 或直接给定一个数字大小，够用就可以。

16 例：

```
#include <stdio.h>

int main()
{
    char str1[20], str2[30];

    printf("请输入用户名: ");
    scanf("%s", str1, sizeof(str1));

    printf("请输入您的网站: ");
    scanf("%s", str2, sizeof(str2));

    printf("输入的用户名: %s\n", str1);
    printf("输入的网站: %s", str2);

    return(0);
}
```

 mhy888 10个月前 (02-22)

分类导航

HTML / CSS

JavaScript

服务器端

数据库

数据分析

移动端

XML 教程

ASP.NET

Web Service

开发工具

网站建设

C 标准库
C 标准库 - 参考手册
C 标准库 - <assert.h>
C 标准库 - <ctype.h>
C 标准库 - <errno.h>
C 标准库 - <float.h>
C 标准库 - <limits.h>
C 标准库 - <locale.h>
C 标准库 - <math.h>
C 标准库 - <setjmp.h>
C 标准库 - <signal.h>
C 标准库 - <stdarg.h>
C 标准库 - <stddef.h>
🔍 C 标准库 - <stdio.h>
C 标准库 - <stdlib.h>
C 标准库 - <string.h>
C 标准库 - <time.h>