

C 库函数 - sscanf()

 [C 标准库 - <stdio.h>](#)

描述

C 库函数 `int sscanf(const char *str, const char *format, ...)` 从字符串读取格式化输入。

声明

下面是 `sscanf()` 函数的声明。

```
int sscanf(const char *str, const char *format, ...)
```

参数

- str** -- 这是 C 字符串，是函数检索数据的源。
- 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 `[=%[*][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 stream 中读取的，但是可以被忽视，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 d、i 和 n）、无符号整型（针对 o、u 和 x）或浮点型（针对 e、f 和 g）的大小：h：短整型（针对 d、i 和 n），或无符号短整型（针对 o、u 和 x）l：长整型（针对 d、i 和 n），或无符号长整型（针对 o、u 和 x），或双精度型（针对 e、f 和 g）L：长双精度型（针对 e、f 和 g）
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。

sscanf 类型说明符：

类型	合格的输入	参数的类型
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 width，函数会读取 width 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,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 e 或 E，以及一个十进制数字。两个有效的实例 -732.103 和 7.12e4	float *
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 *

附加参数 -- 这个函数接受一系列的指针作为附加参数，每一个指针都指向一个对象，对象类型由 format 字符串中相应的 % 标签指定，参数与 % 标签的顺序相同。
针对检索数据的 format 字符串中的每个 format 说明符，应指定一个附加参数。如果您想要把 `sscanf` 操作的结果存储在一个普通的变量中，您应该在标识符前放置引用运算符（&），例如：

```
int n;  
sscanf (str, "%d",&n);
```

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 `sscanf()` 函数的用法。

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

int main()
{
    int day, year;
    char weekday[20], month[20], dtm[100];

    strcpy( dtm, "Saturday March 25 1989" );
    sscanf( dtm, "%s %s %d %d", weekday, month, &day, &year );

    printf("%s %d, %d = %s\n", month, day, year, weekday );

    return(0);
}
```

让我们编译并运行上面的程序，这将产生以下结果：

```
March 25, 1989 = Saturday
```

 [C 标准库 - <stdio.h>](#)