C 库函数 - sprintf()

C 标准库 - <stdio.h>

描述

C 库函数 **int sprintf(char *str, const char *format, ...)** 发送格式化输出到 **str** 所指向的字符串。

声明

下面是 sprintf() 函数的声明。

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

参数

str -- 这是指向一个字符数组的指针，该数组存储了 C 字符串。

format -- 这是字符串，包含了要被写入到字符串 str 的文本。它可以包含嵌入的 format 标签，format 标签可被随后的附加参数中指定的值替换，并按需求进行格式化。format 标签属性是 **%[flags][width][.precision][length]specifier**，具体讲解如下：

specifier（说明符）	输出
c	字符
d 或 i	有符号十进制整数
e	使用 e 字符的科学科学记数法（尾数和指数）
E	使用 E 字符的科学科学记数法（尾数和指数）
f	十进制浮点数
g	自动选择 %e 或 %f 中合适的表示法
G	自动选择 %E 或 %f 中合适的表示法
o	有符号八进制
s	字符的字符串
u	无符号十进制整数
x	无符号十六进制整数
X	无符号十六进制整数（大写字母）
p	指针地址
n	无输出
%	字符

flags（标识）	描述
-	在给定的字段宽度内左对齐，默认是右对齐（参见 width 子说明符）。
+	强制在结果之前显示加号或减号（+ 或 -），即正数前面会显示 + 号。默认情况下，只有负数前面会显示一个 - 号。
(space)	如果没有写入任何符号，则在该值前面插入一个空格。
#	与 o、x 或 X 说明符一起使用时，非零值前面会分别显示 0、0x 或 0X。 与 e、E 和 f 一起使用时，会强制输出包含一个小数点，即使后边没有数字时也会显示小数点。 默认情况下，如果后边没有数字时候，不会显示显示小数点。 与 g 或 G 一起使用时，结果与使用 e 或 E 时相同，但是尾部的零不会被移除。
0	在指定填充 padding 的数字左边放置零（0），而不是空格（参见 width 子说明符）。

width（宽度）	描述
(number)	要输出的字符的最小数目。如果输出的值短于该数，结果会用空格填充。如果输出的值长于该数，结果不会被截断。
*	宽度在 format 字符串中未指定，但是会作为附加整数值参数放置于要被格式化的参数之前。

.precision（精度）	描述
.number	对于整数说明符（d、i、o、u、x、X）：precision 指定了要写入的数字的最小位数。如果写入的值短于该数，结果会用前导零来填充。如果写入的值长于该数，结果不会被截断。精度为 0 意味着不写入任何字符。 对于 e、E 和 f 说明符：要在小数点后输出的小数位数。 对于 g 和 G 说明符：要输出的最大有效位数。 对于 s: 要输出的最大字符数。默认情况下，所有字符都会被输出，直到遇到末尾的空字符。 对于 c 类型：没有任何影响。 当未指定任何精度时，默认为 1。如果指定时不带有一个显式值，则假定为 0。
.*	精度在 format 字符串中未指定，但是会作为附加整数值参数放置于要被格式化的参数之前。

length（长度）	描述
h	参数被解释为短整型或无符号短整型（仅适用于整数说明符：i、d、o、u、x 和 X）。
l	参数被解释为长整型或无符号长整型，适用于整数说明符（i、d、o、u、x 和 X）及说明符 c（表示一个宽字符）和 s（表示宽字符字符串）。
L	参数被解释为长双精度型（仅适用于浮点数说明符：e、E、f、g 和 G）。

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 % 标签。参数的个数应与 % 标签的个数相同。

返回值

如果成功，则返回写入的字符总数，不包括字符串追加在字符串末尾的空字符。如果失败，则返回一个负数。

实例

下面的实例演示了 sprintf() 函数的用法。

```
#include <stdio.h>
#include <math.h>

int main()
{
   char str[80];

   sprintf(str, "Pi 的值 = %f", M_PI);
   puts(str);

   return(0);
}
```

让我们编译并运行上面的程序，这将产生以下结果：

```
Pi 的值 = 3.141593
```

C 标准库 - <stdio.h>