

C LIBRARY FUNCTION - SSCANF

http://www.tutorialspoint.com/c_standard_library/c_function_sscanf.htm

Copyright © tutorialspoint.com

Description

The C library function **int sscanf***constchar * str, constchar * format, ...* reads formatted input from a string.

Declaration

Following is the declaration for sscanf function.

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

Parameters

- **str** -- This is the C string that the function processes as its source to retrieve the data.
- **format** -- This is the C string that contains one or more of the following items: *Whitespace character, Non-whitespace character and Format specifiers*

A format specifier follows this prototype: [=%[*][width][modifiers]type=]

argument	Description
*	This is an optional starting asterisk, which indicates that the data is to be read from the stream but ignored, i.e. it is not stored in the corresponding argument.
width	This specifies the maximum number of characters to be read in the current reading operation.
modifiers	Specifies a size different from int <i>inthecaseofd</i> , <i>iandn</i> , unsigned int <i>inthecaseofo</i> , <i>uandx</i> or float <i>inthecaseofe</i> , <i>fandg</i> for the data pointed by the corresponding additional argument: h : short int <i>ford</i> , <i>iandn</i> , or unsigned short int <i>foro</i> , <i>uandx</i> l : long int <i>ford</i> , <i>iandn</i> , or unsigned long int <i>foro</i> , <i>uandx</i> , or double <i>fore</i> , <i>fandg</i> L : long double <i>fore</i> , <i>fandg</i>
type	A character specifying the type of data to be read and how it is expected to be read. See next table.

scanf type specifiers

type	Qualifying Input	Type of argument
c	Single character: Reads the next character. If a width different from 1 is specified, the function reads width characters and stores them in the successive locations of the array passed as argument. No null character is appended at the end.	char *
d	Decimal integer: Number optionally preceded with a + or - sign	int *
e, E, f, g, G	Floating point: Decimal number containing a decimal point, optionally preceded by a + or - sign and optionally followed by the e or E character and a decimal number. Two examples of valid entries are -732.103 and 7.12e4	float *
o	Octal Integer:	int *
s	String of characters. This will read subsequent characters until a whitespace is found <i>whitespacecharactersareconsideredtobebank, newlineandtab</i>	char *

u	Unsigned decimal integer.	unsigned int *
x, X	Hexadecimal Integer	int *

- **other arguments** -- This function expects a sequence of pointers as additional arguments, each one pointing to an object of the type specified by their corresponding %-tag within the format string, in the same order.

For each format specifier in the format string that retrieves data, an additional argument should be specified. If you want to store the result of a sscanf operation on a regular variable you should precede its identifier with the reference operator, i.e. an ampersand sign **&**, like:
 int n; sscanf str, " ;

Return Value

On success, the function returns the number of variables filled. In the case of an input failure before any data could be successfully read, EOF is returned.

Example

The following example shows the usage of sscanf function.

```
#include <stdio.h>
#include <stdlib.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);
}
```

Let us compile and run the above program that will produce the following result:

```
March 25 1989 = Saturday
Loading [MathJax]/jax/output/HTML-CSS/jax.js
```