

[Logging in From a Linux System or Localhost](#)
4 days 38 min ago

[Auto response](#)
4 days 6 hours ago

[Re: final issue.](#)
4 days 22 hours ago

[Re: Rocky, thank you for this](#)
4 days 23 hours ago

Newsletter

Subscribe to
HowtoForge
Newsletter
and stay informed about
our latest HOWTOs and
projects.

enter email address:

Submit

(To unsubscribe from
our newsletter, visit this
[link](#).)

 [English](#) |  [Deutsch](#) | [Site Map/RSS Feeds](#) | [Advertise](#)

You are here: [Home](#) » [Learning C/C++ Step-By-Step](#) » [Learning C/C++ Step-By-Step - Page 08](#)

Learning C/C++ Step-By-Step - Page 08

Want to support HowtoForge? Become a [subscriber](#)!

Submitted by [ganesh35](#) ([Contact Author](#)) ([Forums](#)) on Wed, 2009-01-07 18:31. ::

0

[Tweet](#)

08. Step-by-Step C/C++ --- C Programming - Strings

Strings

- Introduction
- Characteristics of a strings
- Operations on Strings
 1. Definition of Strings
 2. Initialization of Strings
 3. Reading and printing of Strings
 4. Reading Embedded Blanks
 5. Length of a String
 6. Strings and Functions
 7. Array of Strings

Introduction

Arrays are used to examine strings, generally strings are of array type variables.

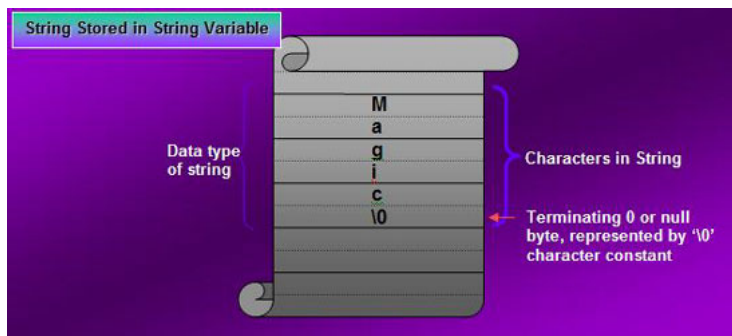
A string is a collection of characters including space where as word is a collection of characters excluding space. Every string variable must be terminating with '\0' null character and the index value is starts with 0.

Every string has the following characteristics:

1. It must be a collection of characters (i.e. characters, numbers, and special characters).
2. Every string must be ends with a NULL character (i.e. '\0')
3. A unique positive number called index identifies each character of a string.
4. Index value must be starts with 0.
5. Random access on characters in a string is possible.
6. A string must be declared with its fixed size like arrays.

For Example consider the following example:

```
char str = " magic";
```



A variety of string library functions are used to manipulate strings. An array of strings is an array of arrays of type char.

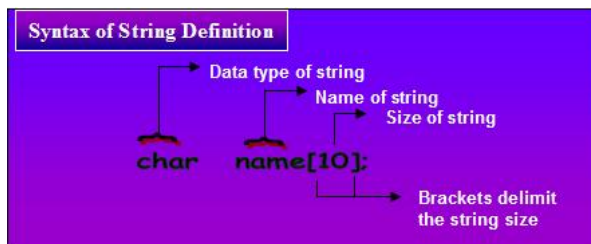
Operations on Strings

We can perform much better operations than using Library string functions. Strings can accept the following operations.

1. Definition of Strings
2. Initialization of Strings
3. Reading and printing of Strings
4. Reading Embedded Blanks
5. Length of a String
6. Strings and Functions
7. Array of Strings

1. Definition of a String

Every variable must be declared at the beginning of the program. Definition of string variable is as follows.



2. Initialization of Strings

Strings can be initialized in the following methods.

1. Direct Assignment

```
char name[10] = "Ashitha";
```

Assigns "Ashitha" to name rest of the place left blank.

2. Direct Assignment without Size

```
char name[] = "Ashitha";
```

Assigns "Ashitha" to name and fix it's width up to the size of Constant.

3. Design time Assignment

```
char name[10];
strcpy(name, "Ashitha");
```

Using Strings functions it is possible.
But C never support the assignment like :
name = "Ashitha";

4. Runtime Assignment

```
char name[10];
scanf("%s", name);
```

It accepts and assigns constant value to variable at runtime.

3. Reading and Printing Strings

C provides various types of string functions to read and print a string constant. Listed below.

Input Statements

```
getch
getche
getchar
gets
scanf
```

Output Statements

```
putch
putchar
puts
printf
```

/* Program to accept and display a string */

```
/* 51_strings.c */
#include <stdio.h>
int main()
{
    char str[20];
    scanf("%s", str);
    printf("%s" str);
    return 0;
}
```

/* Program to accept and display a string with a prompt */

```
/* 52_strings.c */
```

```
#include <stdio.h>
int main( )
{
    char str[20];
    printf("Enter a string :"); scanf("%s", str);
    printf("\nYou entered : %s", str);
    return 0;
}
```

4. Reading embedded blanks

scanf Accepts string, thus it will read strings consisting of a single word, but anything typed after a space is thrown away.

Eg. Enter String : Law is a bottomless pit.

You entered : Law

To read text containing blanks we use another function, gets().

```
/*read string with embedded blanks */

/* 53_gets.c */
const int MAX = 80;
int main()
{
    char str[MAX];
    printf("Enter a string :"); gets(str);
    printf("You entered :"); puts(str);
    return 0;
}
```

5. Length of String

Every string has its fixed length depending on its constant.

The following program demonstrates, How to find the length of the string

```
/* To find the length of a given string */

/* 53_length.c */
#include <stdio.h>
int main()
{
    int i=0;
    char str[50];
    printf("Enter a string "); gets(str);
    while(str[i] != '\0') i++;
    printf("Length is %d", i);
    return 0;
}
```

6. Strings and Functions

A function is a self-contained block of statements that perform a specific task. The best way to organize strings.

The following are the example of string organization using functions.

```
/* Program to find the length of a string */
```

```
/* 54_len.c */
#include <stdio.h>
int len_str(char s[]);
```

```
int main()
{
    int l; char str[50];
    printf("Enter a string "); gets(str);
    l = len_str(str);
    printf("\nLength of string : %d", l);
}
```

```
int len_str(char s[])
{
    int l=0;
    while(s[l] != '\0') l++;
    return l;
}
```

```
/* Program to accept and print a string */
```

```
/* 55_str.c */
#include <stdio.h>
void disp_str(char s[]);
```

```
int main()
{
    char str[50];
    printf("Enter a string "); gets(str);
    disp_str(str);
    return 0;
}
```

```
void disp_str(char s[])
{
    int i=0;
    while( s[i] != '\0' ) putchar(s[i++]);
}
```

7. Array of Strings

Arrays are used to examine strings, generally strings are of array type variables. So, we can access array of strings. The following examples illustrate, How Array of Strings organized.

/* Program to display an array of strings */

```
/* 56_display.c */
#include <stdio.h>
void main()
{
    char week[7][10] = { "Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday" };
    int i;
    for( i = 0; i<7; i++) puts(week[i]);
}
```

/* Program to accept and display an array of strings */

```
/* 57_strings.c */
#include <stdio.h>
void main()
{
    char names[7][10]; int i;
    for( i = 0; i<7; i++) gets(names[i]);
    for( i = 0; i<7; i++) puts(names[i]);
}
```

[previous](#)
[up](#)
[next](#)
[Learning C/C++ Step-By-Step - Page 07](#)
[Learning C/C++ Step-By-Step - Page 09](#)

Copyright © 2009 Ganesh Kumar Butcha
All Rights Reserved.

0

[Tweet](#)

[add comment](#) | [view as pdf](#) | [print: this](#) | [all page\(s\)](#)

Related Tutorials

- [Beginner's Guide To c++](#)
- [An Explanation of Pointers \(C++\)](#)



Please do not use the comment function to ask for help! If you need help, please use our [forum](#).
Comments will be published after administrator approval.

[Howtos](#) | [Mini-Howtos](#) | [Forums](#) | [News](#) | [Search](#) | [Contribute](#) | [Subscription](#)
[Site Map/RSS Feeds](#) | [Advertise](#) | [Contact](#) | [Disclaimer](#) | [Imprint](#)



Copyright © 2013 HowtoForge - Linux Howtos and Tutorials
All Rights Reserved.