String and Character Array

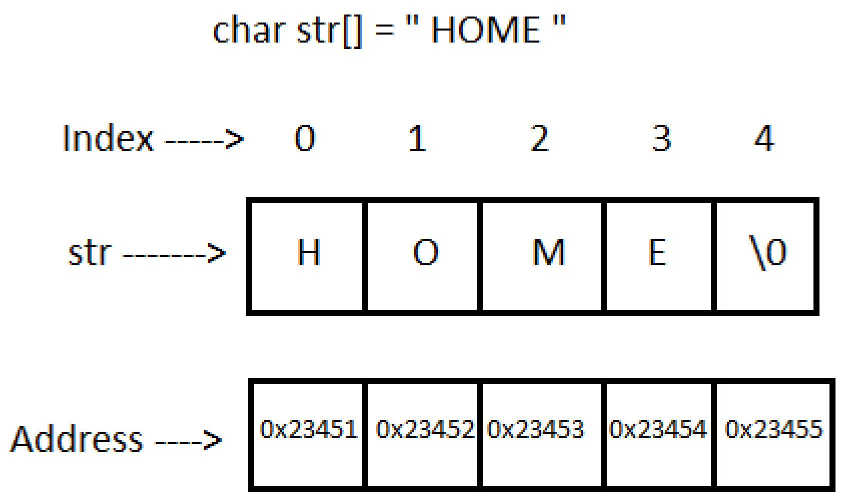
String is a sequence of characters that are treated as a single data item and terminated by a null character `'\0'`. Remember that the C language does not support strings as a data type. A **string** is actually a one-dimensional array of characters in C language. These are often used to create meaningful and readable programs.

Index

If you don't know what an array in C means, you can check the C Array tutorial to know about Array in the C language. Before proceeding further, check the following articles:

- C Function Calls
- C Variables
- C Datatypes
- C Syntax Rules

For example: The string "home" contains 5 characters including the `'\0'` character which is automatically added by the compiler at the end of the string.



Index

Declaring and Initializing a string variables:

```
// valid
char name[13] = "StudyTonight";
char name[10] = {'c','o','d','e','\0'};
```

```
// Illegal
char ch[3] = "hello";
char str[4];
str = "hello";
```

String Input and Output:

- **%s** format specifier to read a string input from the terminal.
- But scanf() function, terminates its input on the first white space it encounters.
- **edit set conversion code %[..]** that can be used to read a line containing a variety of characters, including white spaces.
- The `gets()` function can also be used to read character string with white spaces

```
char str[20];
printf("Enter a string");
scanf("%[^\n]", &str);
printf("%s", str);
```

Index

```
char text[20];
gets(text);
printf("%s", text);
```

String Handling Functions:

X

C language supports a large number of string handling functions that can be used to carry out many of the string manipulations. These functions are packaged in the **string.h** library. Hence, you must include **string.h** header file in your programs to use these functions.

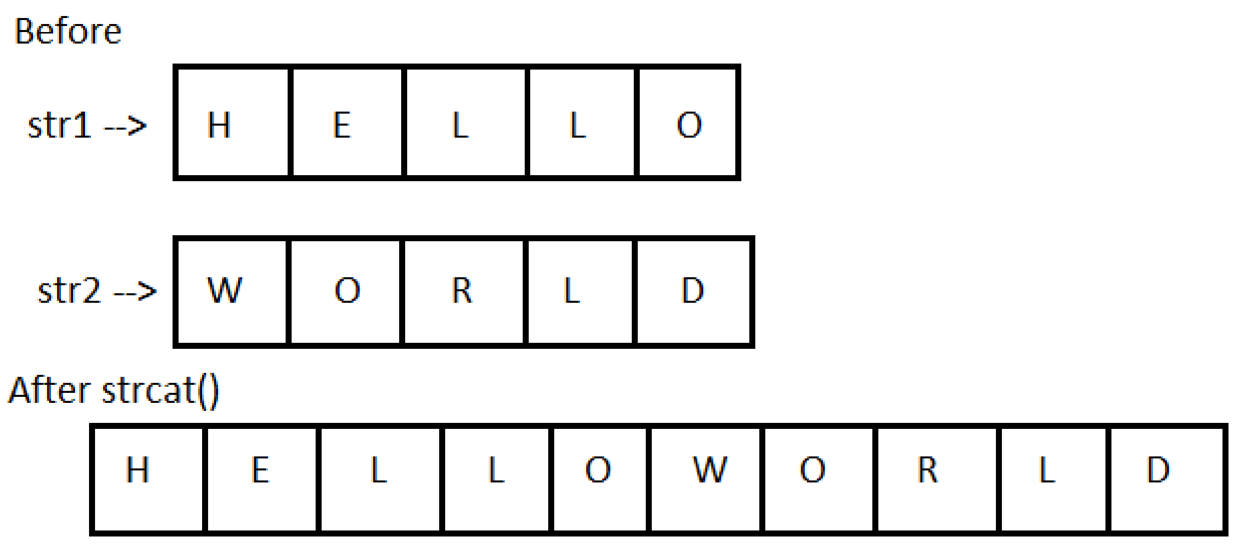
The following are the most commonly used string handling functions.

Method	Description
`strcat()`	It is used to concatenate(combine) two strings
`strlen()`	It is used to show the length of a string
`strrev()`	It is used to show the reverse of a string
`strcpy()`	Copies one string into another
`strcmp()`	It is used to compare two string

`strcat()` function in C:

Index

X



Syntax:

```
strcat("hello", "world");
```

`strcat()` will add the string **"world"** to **"hello"** i.e ouput = helloworld.

`strlen()` and `strcmp()` function:

Index

`strlen()` will return the length of the string passed to it and `strcmp()` will return the ASCII difference between first unmatching character of two strings.

```
  int j = strlen("studytonight");
  int i=strcmp("study ", "tonight");
printf("%d %d",j,i);
```

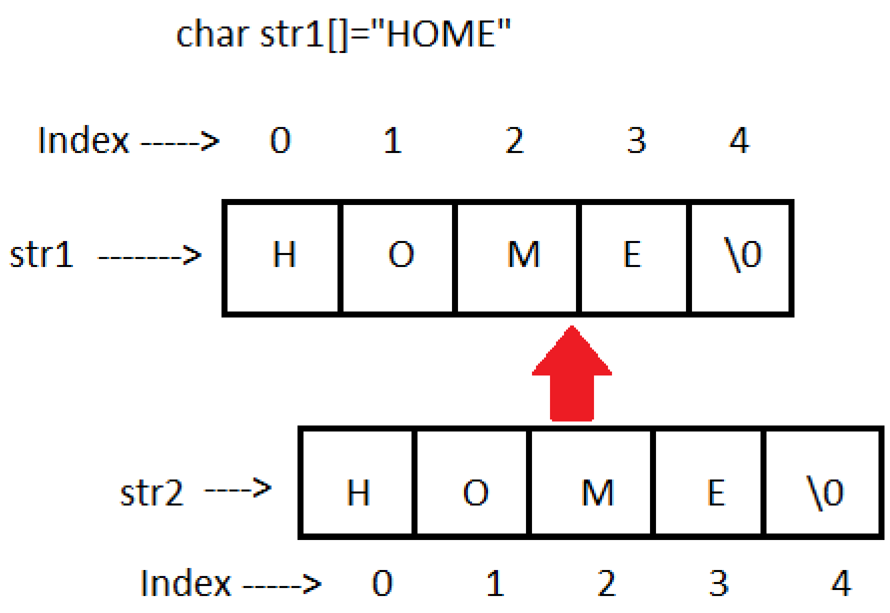
X

Output:

```
12 -1
```

`strcpy()` function:

It copies the second string argument to the first string argument.



Example of `strcpy()` function:

Index

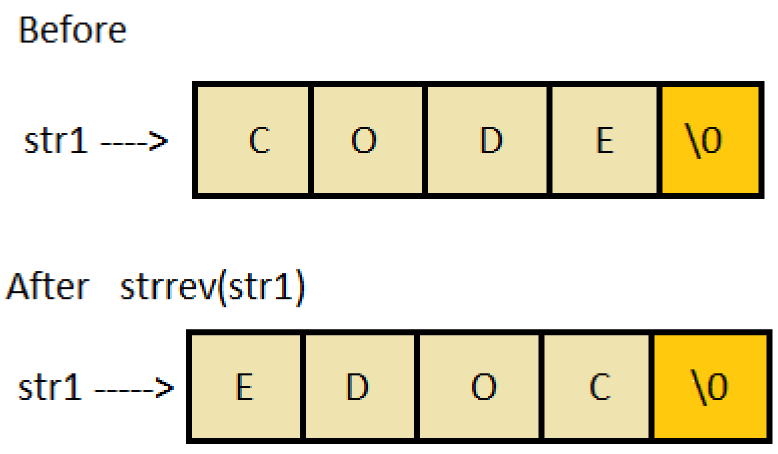
```
#include<stdio.h>
#include<string.h>

int main()
{
    char s1[50], s2[50];

    strcpy(s1, "StudyTonight");
    strcpy(s2, s1);

    printf("%
    return(0);
}
```

OUTPUT:

```
StudyTonight
```

`strrev()` function:

It is used to reverse the given string expression.



Index

Code snippet for `strrev()`:

```
#include <stdio.h>

int main()
{
    char s1[50];

    printf("Enter your string: ");
    gets(s1);
```

X

```
        printf("\nYour reverse string is: %s",strrev(s1));
        return(0);
   }
```

OUTPUT:

```
Enter your string: studytonight

Your reverse string is: thginotyduts
```

Related Tutorials:

- C Array
- C Functions
- C Pointers
- C Structures

C Programs