C - Strings

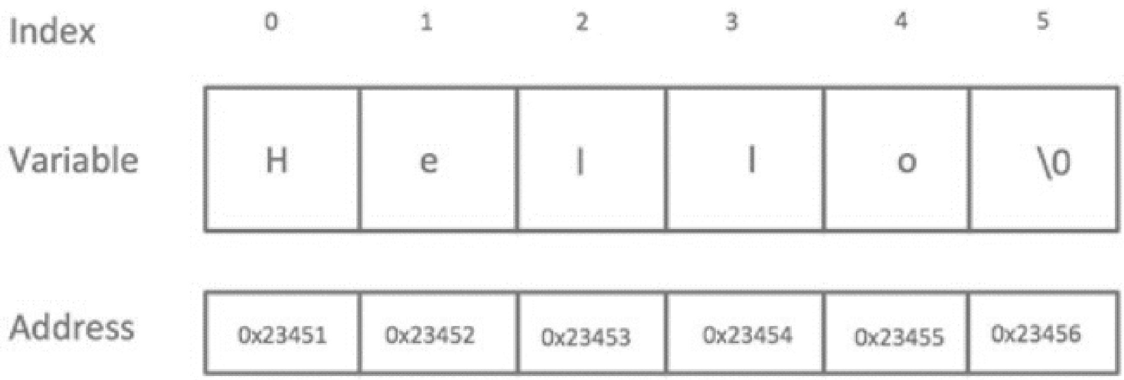
In C programming, a string is a sequence of characters terminated with a null character \0. For example:

```
char c[] = "c string";
```

The following declaration and initialization create a string consisting of the word "Hello". To hold the null character at the end of the array, the size of the character array containing the string is one more than the number of characters in the word "Hello."

```
char greeting[6] = {'H', 'e', 'l', 'l', 'o', '\0'};
```

Following is the memory presentation of the above defined string in C



Index	0	1	2	3	4	5
Variable	H	e	l	l	o	\0
Address	0x23451	0x23452	0x23453	0x23454	0x23455	0x23456

Actually, you do not place the *null* character at the end of a string constant. The C compiler automatically places the '\0' at the end of the string when it initializes the array.

How to initialize strings?

You can initialize strings in a number of ways.

```
char c[] = "abcd";

char c[50] = "abcd";

char c[] = {'a', 'b', 'c', 'd', '\0'};

char c[5] = {'a', 'b', 'c', 'd', '\0'};
```

Let us try to print the above-mentioned string –

```
#include <stdio.h>

int main () {

   char greeting[6] = {'H', 'e', 'l', 'l', 'o', '\0'};
   printf("Greeting message: %s\n", greeting );
   return 0;
}
```

When the above code is compiled and executed, it produces the following result −

Greeting message: Hello

Arrays and strings are second-class citizens in C; they do not support the assignment operator once it is declared. For example,

```
char c[100];
c = "C programming”; // Error! array type is not assignable.
```

Note: Use the strcpy() function to copy the string instead.

C supports a wide range of functions −

Sr.No.	Function & Purpose
1	**strcpy(s1, s2);** Copies string s2 into string s1.
2	**strcat(s1, s2);** Concatenates string s2 onto the end of string s1.
3	**strlen(s1);** Returns the length of string s1.
4	**strncmp(s1, s2);** Returns 0 if s1 and s2 are the same; less than 0 if s1<s2; greater than 0 if s1>s2.
5	**strchr(s1, ch);** Returns a pointer to the first occurrence of character ch in string s1.
6	**strstr(s1, s2);** Returns a pointer to the first occurrence of string s2 in string s1.

Read & write Strings in C using Printf() and Scanf() functions

```
#include <stdio.h>
#include <string.h>
int main()
{
    /* String Declaration*/
    char name[20];

    printf("Enter your Nick name:");

    /* I am reading the input string and storing it in nickname
     * Array name alone works as a base address of array so
     * we can use nickname instead of &nickname here
     */
    scanf("%s", name);

    /*Displaying String*/
    printf("%s",name);

    return 0;
}
```

Output:

```
Enter your Nick name:HASH
HASH
```

Read & Write Strings in C using gets() and puts() functions

```
#include <stdio.h>
#include <string.h>
int main()
{
    /* String Declaration*/
    char nickname[20];

    /* Console display using puts */
    puts("Enter your Nick name:");

    /*Input using gets*/
    gets(nickname);

    puts(nickname);

    return 0;
}
```

Output:

```
Enter your Nick name:HASH
HASH
```