

3.5 STRINGS IN C

A string is a sequence of characters. e.g. "ScholarHat". Many programming languages have a data type known as a string for storing string values.

In C, a string is stored as an array of characters. You have to use char data type and make an array of characters to create a string. The string is terminated with a null character '\0'.

A string is represented using double quotes (""), or single quotes ("). Whenever an array of characters e.g. c[10] gets created, the compiler implicitly appends '\0' at the end of the string.

A String in C programming is a sequence of characters terminated with a null character '\0'. The C String is stored as an array of characters. The difference between a character array and a C string is that the string in C is terminated with a unique character '\0'.

a) STRING DECLARATION

Declaring a string in C is as simple as declaring a one-dimensional array. Below is the basic syntax for declaring a string.

```
char string_name[size];
```

In the above syntax string_name is any name given to the string variable and size is used to define the length of the string, i.e the number of characters strings will store.

There is an extra terminating character which is the Null character ('\0') used to indicate the termination of a string that differs strings from normal character arrays.

b) STRING INITIALIZATION

A string in C can be initialized in different ways. We will explain this with the help of an example. Below are the examples to declare a string with the name str and initialize it with "ArtificialIntelligence"

We can initialize a C string in 4 different ways which are as follows:

1. **Assigning A String Literal Without Size**

String literals can be assigned without size. Here, the name of the string str acts as a pointer because it is an array.

```
char str[] = "ArtificialIntelligence ";
```

2. **Assigning a String Literal with a Predefined Size**

String literals can be assigned with a predefined size. But we should always account for one extra space which will be assigned to the null character.

If we want to store a string of size n then we should always declare a string with a size equal to or greater than n+1.

```
char str[50] = "ArtificialIntelligence ";
```

3. Assigning Character by Character with Size

We can also assign a string character by character. But we should remember to set the end character as '\0' which is a null character.

```
char str[14] = {'W','e','l','c','o','m','e','\0'};
```

4. Assigning Character by Character without Size

We can assign character by character without size with the NULL character at the end. The size of the string is determined by the compiler automatically.

```
char str[] = {'W','e','l','c','o','m','e','\0'};
```

C program to illustrate strings

```
#include <stdio.h>
#include <string.h>
int main()
{
    char str[] = "Hello";
    printf("%s\n", str);
    int length = 0;
    length = strlen(str);
    printf("Length of string str is %d", length);
    return 0;
}
```



Output

Hello

Length of string str is 5