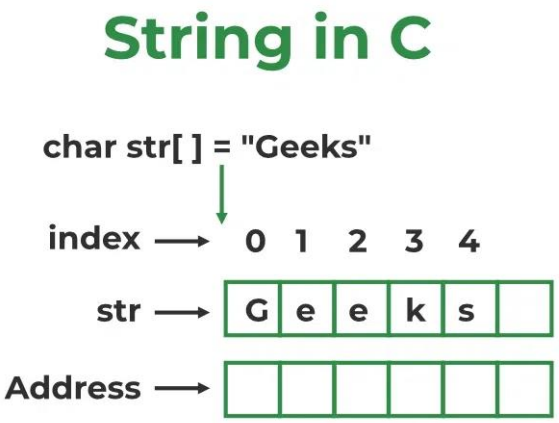
String in C

- 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 the string is terminated with a unique character '\0'.



C String Declaration Syntax

Declaring a string in C is as simple as declaring a one-dimensional array.

Syntax

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

- In the above syntax **str_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**.

Initilization of String

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[] = "Myselfkritika";
```

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] = "MyselfKritika";
```

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] = { 'M','y','s','e','l','f','K','r','i','t','i','k','a','\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[] = { 'M','y','s','e','l','f','K','r','i','t','i','k','a','\0'};
```

Notes:

- When a Sequence of characters enclosed in the double quotation marks is encountered by the compiler, a null character '\0' is appended at the end of the string by default.
- After declaration, if we want to assign some other text to the string, we have to assign it one by one or use built-in strcpy() function because the

direct assignment of string literal to character arrray is only possible in declration.

The C arrays are static in nature, i.e., they are allocated memory at the compile time.

Example:

```
// C Program to illustrate string
#include <stdio.h>
#include <string.h>

int main()
{
   // declare and initialize string
   char str[] = "Kritika";

   // print string
   printf("%s\n", str);

   int length = 0;
   length = strlen(str);

   // displaying the length of string
   printf("Length of string str is %d", length);

   return 0;
}
```

Output:

```
Kritika
Length of string str is 7
```

- We can see in the above program that strings can be printed using normal printf statements just like we print any other variable. Unlike arrays, we do not need to print a string, character by character.

- **Note:** The C language does not provide an inbuilt data type for strings but it has an access specifier "**%s**" which can be used to print and read strings directly.

How to Read a String Separated by Whitespaces in C?

We can use multiple methods to read a string separated by spaces in C. The two of the common ones are:

→ We can use the fgets() function to read a line of string and gets() to read characters from the standard input (stdin) and store them as a C string until a newline character or the End-of-file (EOF) is reached.

→ We can also scanset characters inside the scanf() function

Example:

```
// C program to illustrate fgets()
#include <stdio.h>
#define MAX 50
int main()
{
char str[MAX];

// MAX Size if 50 defined
fgets(str, MAX, stdin);

printf("String is: \n");

// Displaying Strings using Puts
puts(str);

return 0;
}
```

Input:

Wearedoingcode

Output:

```
String is:
Wearedoingcode
```

C String Length

The length of the string is the number of characters present in the string except for the NULL character. We can easily find the length of the string using the loop to count the characters from the start till the NULL character is found.

Array of Strings in C

In C programming String is a 1-D array of characters and is defined as an array of characters. But an array of strings in C is a two-dimensional array of character types. Each String is terminated with a null character (\0). It is an application of a 2d array.

➔ **Syntax:**

```
char variable_name[r] = {list of string};
```

Here,

- **var_name** is the name of the variable in C.
- **r** is the maximum number of string values that can be stored in a string array.
- **c** is the maximum number of character values that can be stored in each string array.
- to avoid high space consumption in our program we can use an Array of Pointers in C.