Programming Tip:
Character constants are enclosed in single quotes. String constants are enclosed in double quotes.

In C language, a string is nothing but a null-terminated character array. This means that after the last character, a null character (`'\0'`) is stored to signify the end of the character array. For example, if we write,

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


Scanned with CamScanner

We are declaring a character array that has five usable characters namely, H, E, L, L, and O. Apart from these characters, a null character ('\0') is stored at the end of the string. So, the internal representation of the string becomes HELLO'\0'. To store a string of length 5, we need 5 + 1 locations (1 extra for the null character). The name of the character array (or the string) is a pointer to the beginning of the string. Figure 6.1 shows the difference between character storage and string storage.

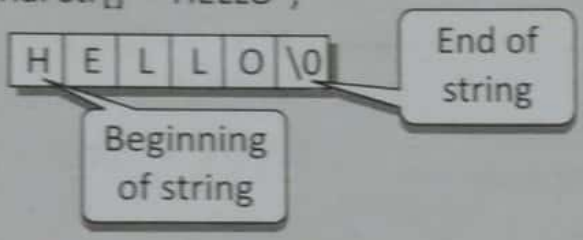
If we had declared str as,

```
char str[5] = "HELLO";
```

Then the null character will not be appended automatically to the character array. This is because, str can hold only 5 characters and the characters in HELLO have already filled the space allocated to it.

Scanned with CamScanner

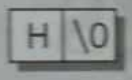
char str[] = "HELLO";



char ch = 'H';

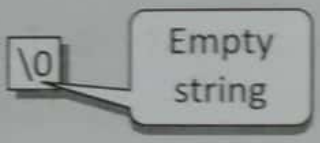
Here H is a character not a string. The character H requires only one memory location.

char str[] = "H";



Here H is a string not a character. The string H requires two memory locations. One to store the character H and another to store the null character.

char str[] = "";



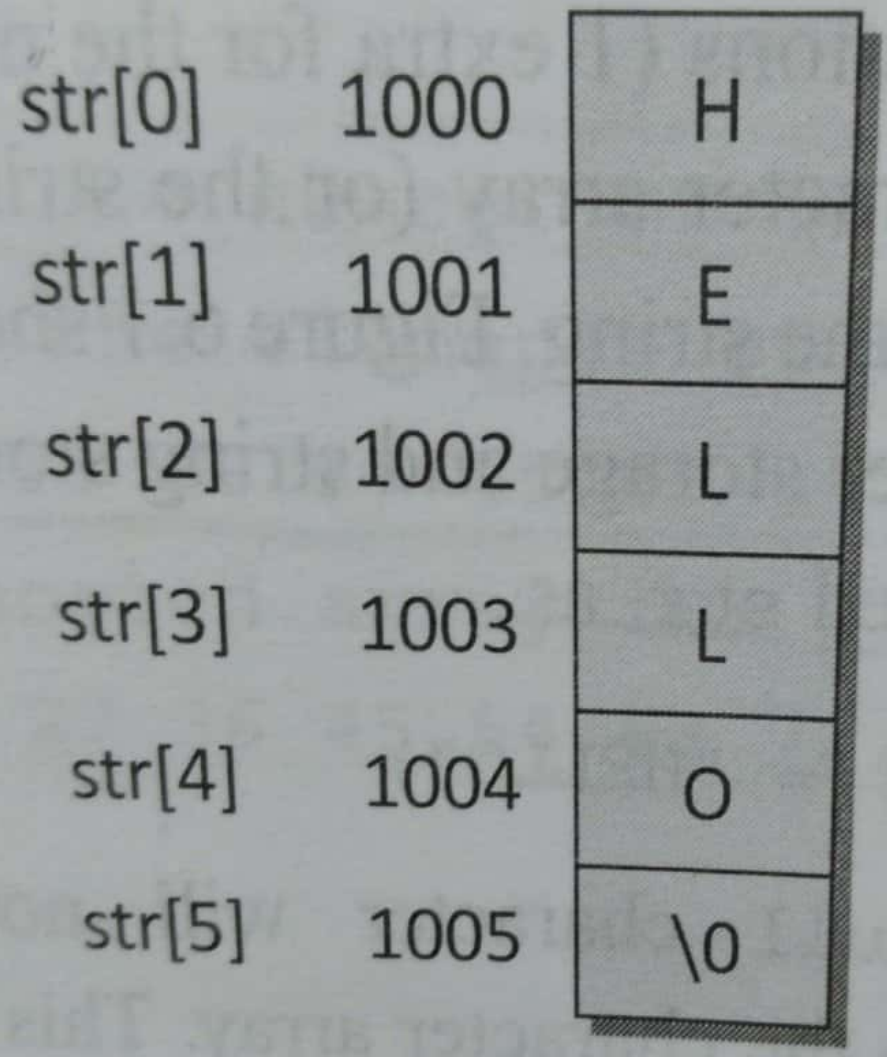
Although C permits empty string, it does not allow an empty character.

Scanned with CamScanner

Programming Tip:

When allocating memory space for a character array, reserve space to hold the null character also.

Scanned with CamScanner



Memory representation of a character array

Scanned with CamScanner

6.1.1 Reading Strings

If we declare a string by writing

```
char str[100];
```

Then `str` can be read from the user by using three ways:

1. using `scanf` function
2. using `gets()` function

Scanned with CamScanner

3. using `getchar()`, `getch()` or `getche()` function repeatedly

The string can be read using `scanf()` by writing

```
scanf("%s", str);
```


Scanned with CamScanner

Programming Tip:
Using & operand with a string variable in the scanf statement generates an error.

Scanned with CamScanner

An array name cannot be used as the left operand of an assignment operator. Therefore, the following statement is illegal in C.

```
char str2, str1[]="HI";
str2 = str1;
```


Scanned with CamScanner

6.1.2 Writing Strings

The string can be displayed on screen using three ways:

1. using `printf()` function
2. using `puts()` function
3. using `putchar()` function repeatedly

The string can be displayed using `printf()` by writing

```
printf("%s", str);
```


Scanned with CamScanner

. Write a program to find the length of a string.

```
#include <stdio.h>
#include <conio.h>
int main()
{
   char str[100], i = 0, length;
   clrscr();
   printf("\n Enter the string :");
   gets(str);
   while(str[i] != '\0')
      i++;
   length = i;
   printff("\n The length of the string is :
   %d", length);
   getch();
}
```

Output

```
Enter the string : HELLO
The length of the string is : 5
```


Scanned with CamScanner

if (strcmp (str1, str2) > 0, the str1 is greater than str2

if (strcmp (str1, str2) < 0, the str1 is less than str2

if (strcmp (str1, str2) = 0, the str1 is equal to str2.

Scanned with CamScanner