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

Strings are used in string handling operations such as,
- Counting the length of a string.
- Comparing two strings.
- Copying one string to another.
- Converting lower case string to upper case.
- Converting upper case string to lower case.
- Joining two strings.
- Reversing string.

**Declaration**

**Syntax** : char string nm[size];

**Example** : char name [15];

**Declaring and Initializing String**

Declaring character arrays with a string constant

```c
char atomic[ ] = “hydrogen”;;
```

Declaring a string array with character constants

```c
char astring[ ] = {‘Z’, ‘i’, ‘p’, ‘!’,’\0’};
```

**Note** In C, you must always terminate a character array with the NULL character, ‘\0’. Therefore, the array size of your character array should be one plus the maximum length of the string you want to store. Example: In the declaration

```c
char atomic[ ] = “hydrogen”;
“atomic” is an array of nine elements, the last being ‘\0’.
```

**Memory Representation of String**

The memory presentation of the string in C is:

```c
char atomic[10] = “hydrogen”;
```

<table>
<thead>
<tr>
<th>Index Variable</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>h</td>
<td>y</td>
<td>d</td>
<td>r</td>
<td>o</td>
<td>g</td>
<td>e</td>
<td>n</td>
<td>\0</td>
</tr>
</tbody>
</table>

**String memory representation**
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.

**Input & Output of String**

- **Read Strings:**
  To read a string, we can use `scanf()` function with format specifier `%s`.

```
    char name[50];
    scanf("%s",name);
```

- **Write Strings:**
  To write a string, we can use `printf()` function with format specifier `%s`.

```
    char name[50];
    scanf("%s",name);
    printf("%s",name);
```

**Try Yourself:**
1. What is string in C?
2. What is meant by Null in character string.