Characters Strings Defining a String

- C has no special variable type for strings
    - ➔ This means there are no special operators in the language for processing strings
    - ➔ The standard library provides an extensive range of functions to handle strings/

- Strings in C are stored in an array of type char
    - ➔ Characters in a string are stored in adjacent memory cells, one character per cell.
- To declare a string in C, simply use the char type and the brackets to indicate the size.

char myString[20];

- This variable can accommodate a string that can contain up to 19 characters.
    - ➔ You must allow one element for the termination character (null character)
- When you specify the dimension of the array that you intend to use to store a string , it must be at least one greater than the number of characters in the string that you want to store .
    - ➔ The compiler automatically adds \0 to the end of every string constant.

Initializing a String

- You can initialize a string variable when you declare it

  char word[]= {'H','E','L','L','O','!'};

- To initialize a string, it is the same as any other array initialization.
    - ➔ In the absence of a particular array size, the C compiler automatically computes the number of elements in the array.
        - ★ Based upon the number of initializers
        - ★ This statement reserves space in memory for exactly seven characters
        - ★ Automatically adds the null terminator.
- You can specify the size of the string explicitly, just make sure you leave enough space for the terminating null character.

Char word[7] = {"Hello!"};

- If the size specified is too small, then the compiler can't fit a terminating null character at the end of the array , and it doesn't put one there (and it doesn't complain about it either)

Char word[6] = {"Hello!"};

- So … do not specify the size, let the compiler figure out, you can be sure it will be correct
- You can initialize just part of an array of elements of type char with a string.

Char str[40] = "to be";

- The compiler will initialize the first five elements, str[0] to str[4], with the characters of the string constant
    - ➔ Str[5] will contain the null character, '\0'
    - ➔ Space is allocated for all 40 elements if the array

Assigning a value to a string after initializing

- Since you can not assign arrays in C, you can not assign strings either
- The following is an error:

```
char s[100]; // declare
s = “hello”; // initialize - DOESN’T WORK (‘value required’ error)
```

- You are performing an assignment operation, and you cannot assign one array of characters to another array of characters like this .
    - ➔ You have to use strncpy() to assign a value to a char array after it has been declared or initialized.
- The below is perfectly valid

```
S[0] = ‘h’;
S[1] = ‘e’;
S[2] = ‘l’;
S[3] = ‘l’;
S[4] = ‘o’;
S[5] = ‘\0’;
```

Displaying a string

- When you want to refer to a string stored in an array, you just use the array name by itself
- To display a string as output using the printf function, you do the following

```
printf(“\nThe message is: %s”, message);
```

- The %s format specifier is for outputting a null-terminated string
- The printf() function assumes when it encounters the %s format characters that the corresponding argument is a character string that is terminated by a null character.

Inputting a string

- To input a string via the keyboard , use the scanf function

```
char input[10];
printf(“Please input your name: ”);
scanf(“%s” , input);
```

- The %s format specifier is for inputting string
- No need to use the & (address of operator) on a string

Testing if two strings are equal

- You CANNOT directly test two strings to see if they are equal with a statement such as if (string1 == string2)
- The equality operator can only be applied to simple variable types, such as floats, ints, or chars .
    - ➔ Does not work on structures or arrays
- To determine if two strings are equal, you must explicitly compare the two character strings character by character.
    - ➔ We will discuss as easier way with the strcmp Function

Reminder:::

- The string constant “x” is not the same as the character constant ‘x’
- ‘x’ is a basic type (char)
- “x” is a derived type, an array of char
- “x” really consists of two characters, ‘x’ and ‘\0’, the null character

```
#include <stdio.h>

int main(void)
{
    char str1[] = "To be or not to be";
    char str2[] = "that is the question";
    unsigned int count = 0;     //Stores the string length

    while (str1[count] != '\0') //increments count till we reach the
        ++count;                // terminating character.

        printf("The length of the string \"%s\" is %d characters.\n",
str1, count);

        count = 0; //Reset count for next string
        while (str2[count] != '\0')
            ++count;
            printf("The length of the string \"%s\" is %d
characters.\n", str2, count);
            return 0;
}
```