String in c programming



Prepared By

Poly Bhoumik

Dept. of CSE, DIIT



Strings

- A string is nothing but the collection of the individual array elements or characters.
- String is enclosed within Double quotes.
- “programming" is a example of String.
- Each Character Occupy 1 byte of Memory.
- Size of “programming“ = 11 bytes
- String is always Terminated with NULL Character (‘\0′).
  char word[20] = ““p’ , ‘r’ , ‘o’ , ‘g’ , ‘r’ , ‘a’ , ‘m’ , ‘m’ , ‘I’ , ‘n’ , ‘g’ , ‘\0’”



NULL Character

- NULL Character is also known as string terminating character.
- It is represented by “\0”.
- NULL Character is having ASCII value 0
- NULL terminates a string, but isn’t part of it
- important for strlen() – length doesn’t include the NULL



Declaration of a string

- Since we cannot declare string using String Data Type, instead of which we use array of type “char” to create String.
- ***Syntax :***
- char String_Variable_name [SIZE] ;
- ***Examples :***
- char city[30];
- char name[20];
- char message[50];



Rules for declaring a string

- String / Character Array Variable name should be legal C Identifier.
- String Variable must have Size specified.
- *char city[];*
- Above Statement will cause compile time error.
- Do not use String as data type because String data type is included in later languages such as C++ / Java. C does not support String data type
- When you are using string for other purpose than accepting and printing data then you must include following header file in your code –

#include<string.h>

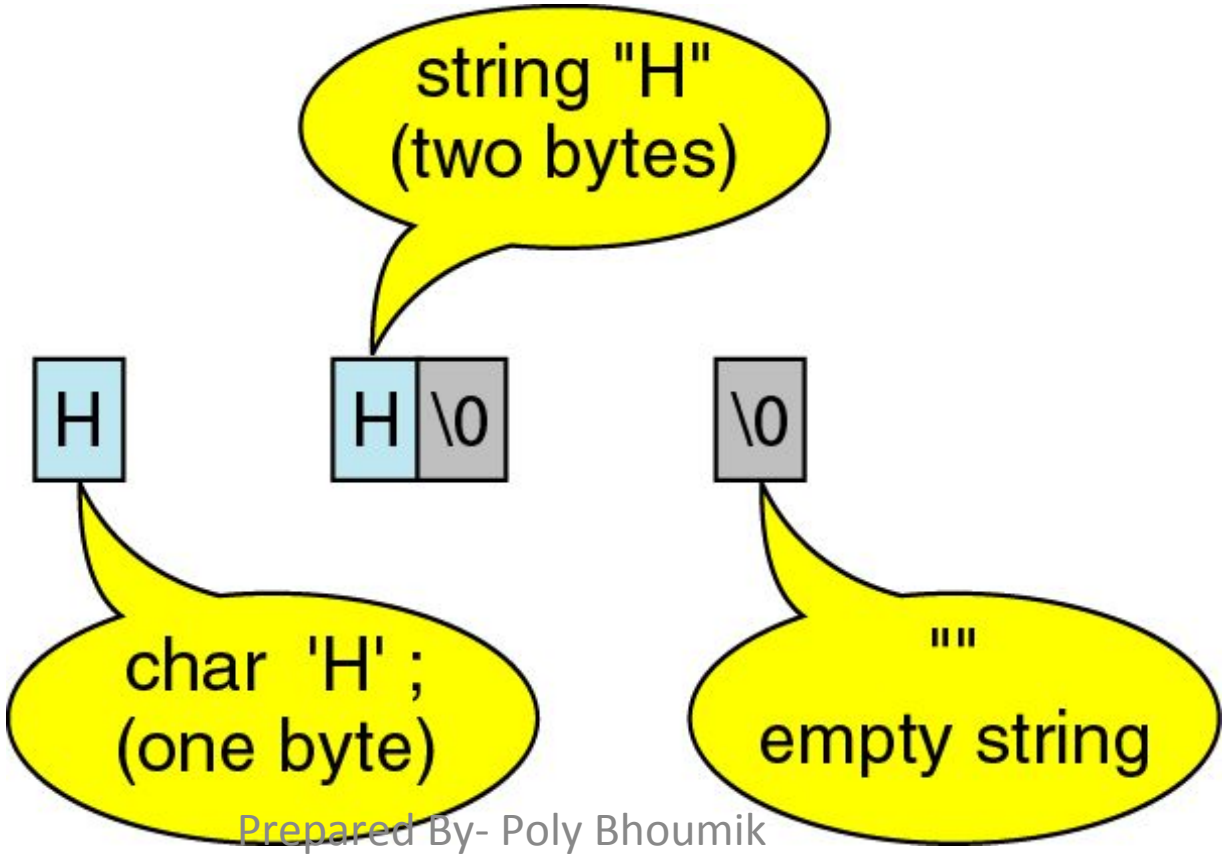


Initializing String (Character Array)

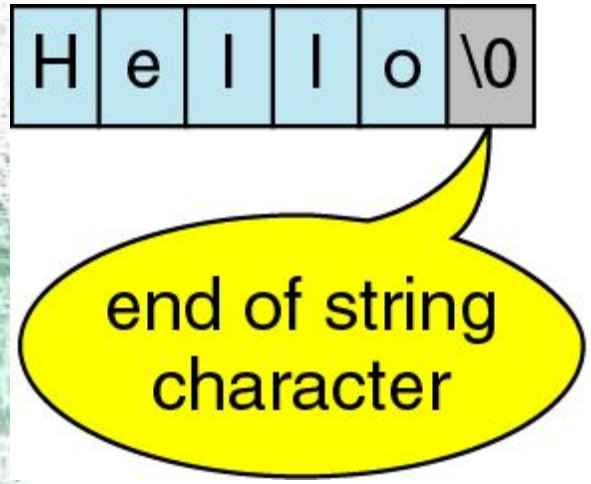
- Process of Assigning some legal default data to String is Called **Initialization of String**.
- There are different ways of initializing String in C Programming –
- Initializing Unsized Array of Character
- Initializing String Directly
- Initializing String Using Character Pointer

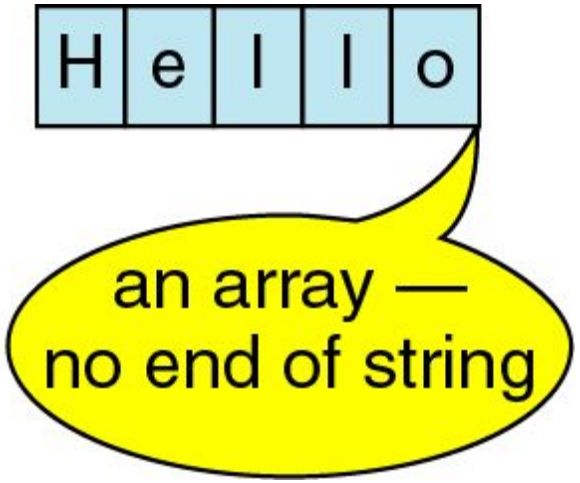
Character Strings

- A string containing a single character takes up 2 bytes of storage.

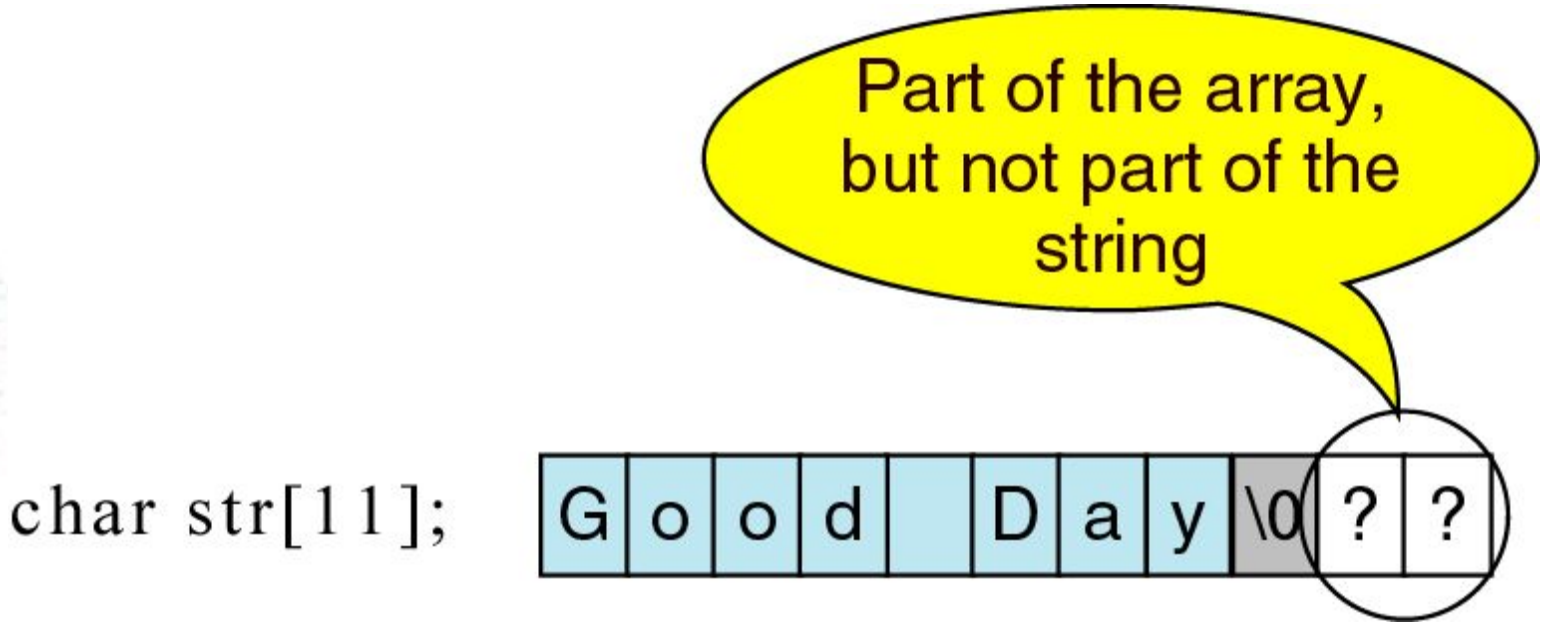


Character Strings





Character Strings





Character vs. String

A string constant is a sequence of characters enclosed in double quotes.

For example, the character string:

```
char s1[2]="a";  //Takes two bytes of storage.
```

s1:

a	\0

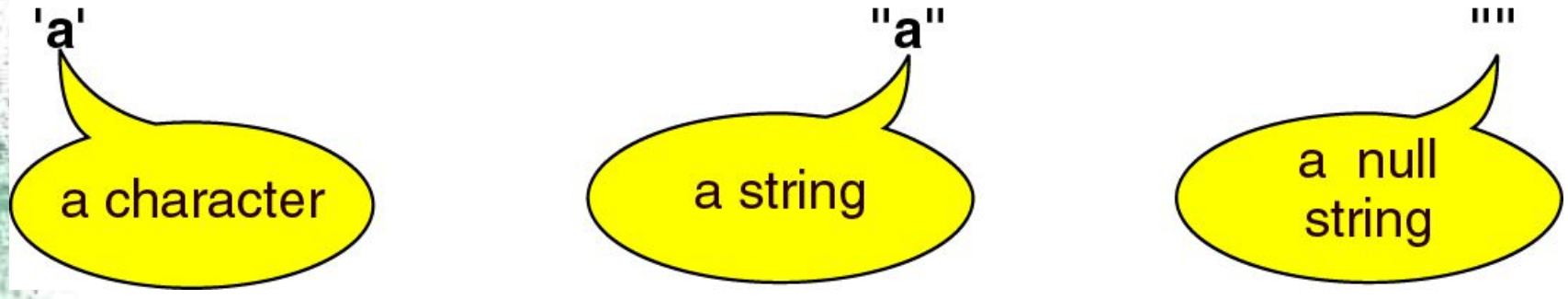
On the other hand, the character, in single quotes:

```
char s2= `a`; //Takes only one byte of storage.
```

s2:

a

Character vs. String





Reading and displaying Strings

- It can be done manually.

```c
#include <stdio.h>
int main(){
      char name[30],ch;
      int i=0;
      printf("Enter name: ");
      while(ch!='\n')      // terminates if user hit enter
      {
            ch=getchar();
            name[i]=ch;
            i++;
      }
      name[i]='\0';          // inserting null character at end
      printf("Name: %s",name);
      return 0;
}
```

Using printf() and scanf()

```c
#include <stdio.h>
int main(){
    char name[20];
    printf("Enter name: ");
    scanf("%s",name);
    printf("Your name is %s.",name);
    return 0;
}
```

Using gets() and puts()

```c
int main(){
    char name[30];
    printf("Enter name: ");
    gets(name);      //Function to read string from user.
    printf("Name: ");
    puts(name);     //Function to display string.
    return 0;
}
```



Passing String to function

```c
#include <stdio.h>
void Display(char ch[]);
int main(){
    char c[50];
    printf("Enter string: ");
    gets(c);
    Display(c);      // Passing string c to function.
    return 0;
}
void Display(char ch[]){
    printf("String Output: ");
    puts(ch);
}
```

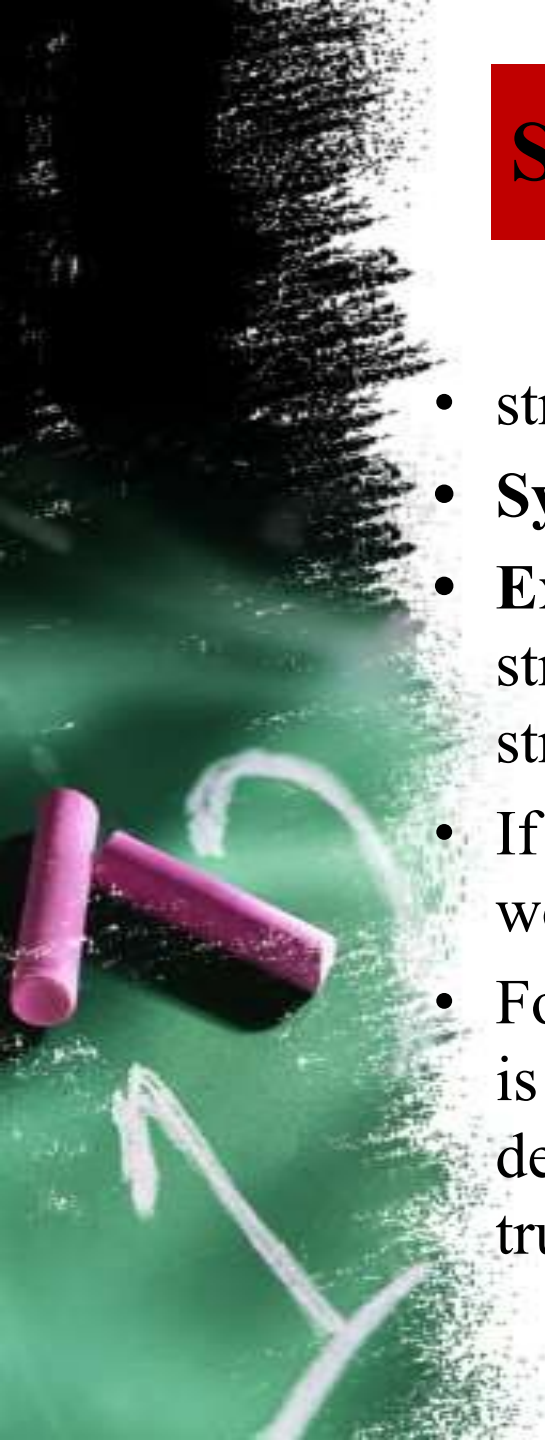


String handling functions

- Strings need to be manipulated by programmer.
- It can be done manually but is time consuming.

Functions of string.h

Function	Purpose	Example	Output
Strcpy();	Makes a copy of a string	strcpy(s1, “Hi”);	Copies “Hi” to ‘s1’ variable
Strcat();	Appends a string to the end of another string	strcat(“Work”, “Hard”);	Prints “WorkHard”
Strcmp();	Compare two strings alphabetically	strcmp(“hi”, “bye”);	Returns -1.
Strlen();	Returns the number of characters in a string	strlen(“Hi”);	Returns 2.
Strrev();	reverses a given string	Strrev(“Hello”);	olleH
Strlwr();	Converts string to lowercase	Strlwr(“HELLO”);	hello
Strupr();	Converts string to uppercase	Strupr(“hello”);	HELLO



String Copy (strcpy)

- strcpy( ) function copies contents of one string into another string.
- **Syntax** : strcpy (destination_string , source_string);
- **Example:**
  strcpy (str1, str2) – It copies contents of str2 into str1.
  strcpy (str2, str1) – It copies contents of str1 into str2.
- If destination string length is less than source string, entire source string value won't be copied into destination string.
- For example, consider destination string length is 20 and source string length is 30. Then, only 20 characters from source string will be copied into destination string and remaining 10 characters won't be copied and will be truncated.



String Concat (strcat)

- strncat( ) function in C language concatenates (appends) portion of one string at the end of another string.
- **Syntax** *:* strncat (destination_string , source_string, size);
- **Example** *:*
  strncat (str2, str1, 3); – First 3 characters of str1 is concatenated at the end of str2.
  strncat (str1, str2, 3); – First 3 characters of str2 is concatenated at the end of str1.
- As you know, each string in C is ended up with null character (‘\0’).
- In strncat( ) operation, null character of destination string is overwritten by source string’s first character and null character is added at the end of new destination string which is created after strncat( ) operation.



String Compare (strcmp)

- strcmp( ) function in C compares two given strings and returns zero if they are same.

✔ If length of string1 < string2, it returns < 0 value that is -1.

✔ If length of string1 > string2, it returns > 0 value that is 1

✔ If length of string1 = string2 it returns 0.

- **Syntax :** strcmp (str1 , str2);
- strcmp( ) function is case sensitive. i.e, “A” and “a” are treated as different characters.



String Length (strlen)

- strlen( ) function in C gives the length of the given string.
- **Syntax** : strlen(str);
- strlen( ) function counts the number of characters in a given string and returns the integer value.
- It stops counting the character when null character is found. Because, null character indicates the end of the string in C.



Classwork

- WAP to read name of 5 persons using array of strings and display them
- WAP to sort name of 5 persons in alphabetical order



FAQ'S

- How to declare character array?
- Define string?
- What are various functions of string?
- How to compare to strings?



Bibliography

- https://www.programiz.com/c-programming/c-string-examples
- https://www.tutorialspoint.com/cprogramming/c_strings.htm
- http://www.cprogramming.com/tutorial/c/lesson9.html
- http://fresh2refresh.com/c-programming/c-strings/



THANK YOU