String

The string can be defined as the one-dimensional array of characters terminated by a null ('\0'). Each character in the array occupies one byte of memory, and the last character must always be 0. The termination character ('\0') is important in a string since it is the only way to identify where the string ends.

There are two ways to declare a string in c language.

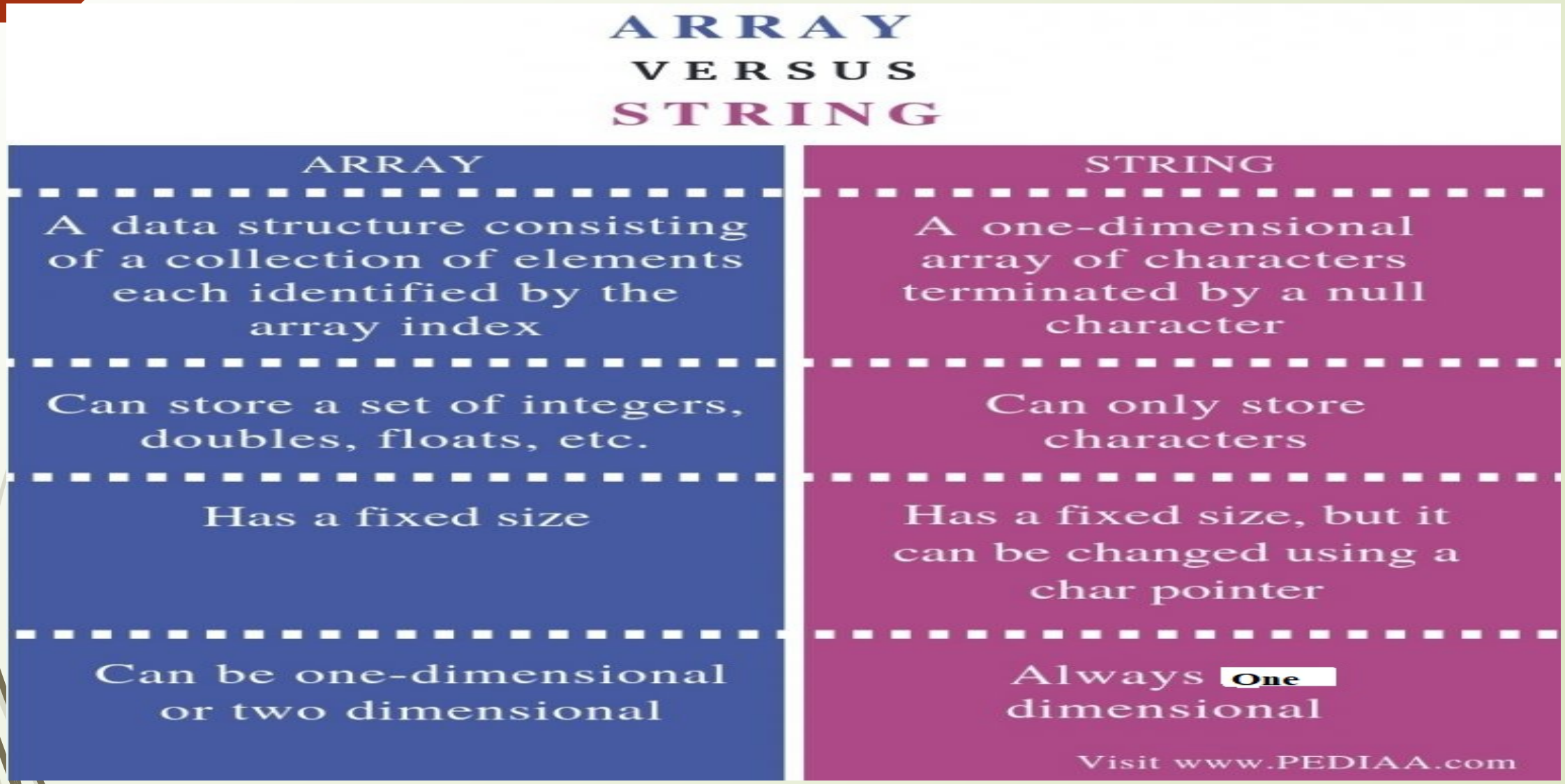
1. By char array

2. By string literal

Let's see the example of declaring **string by char array** in C language.

char ch[10]={‘s, 'a', ‘b', ‘b', ‘i', ‘r', ‘h', ‘o', ‘s', ‘s', '\0'};

Array Vs String



Example

```
#include<stdio.h>
#include <string.h>
int main(){
 char ch[11]={‘S', 'a', ‘k', ‘i', ‘l','\0'};
 char ch2[11]=“Sakil";
 printf("Char Array Value is: %s\n", ch);
 printf("String Literal Value is: %s\n", ch2);
 return 0;
}
```

Reading a line of text

C language programming : Beginners

gets(), puts()

- **gets():** Reads a string from keyboard.

  syntax:

  ```
  gets(string_variable);
  ```

- **puts():** Writes a string to output screen.

  syntax:

  ```
  puts(string_variable);
  ```

Example

```
#include<stdio.h>
#include <string.h>
int main(){
char name[50];
printf("Enter your name: ");
gets(name); //reads string from user
printf("Your name is: ");
puts(name);  //displays string
return 0;
}
```

String pointer

- String pointers are declared as a pointer to a char.
- When there is a value assigned to the string pointer the NULL is put at the end automatically.

Take a look at this example:

```
#include <stdio.h>
#include <string.h>

void main()
    {
        char *ptr_mystring;

        ptr_mystring = "HELLO";

        printf("%s\n", ptr_mystring);

    }
```

String pointer

It is not possible to read, with scanf(), a string with a string pointer. You have to use a character array and a pointer. See this example:

```c
#include <stdio.h>
#include <string.h>

void main()
    {
            char my_array[10];
            char *ptr_section2;

            printf("Type hello and press enter\n");
            scanf("%s", my_array);
            ptr_section2 = my_array;
            printf("%s\n", ptr_section2);
    }
```

String Function

C String Functions

← Prev Next →

There are many important string functions defined in "string.h" library.

No.	Function	Description
1)	strlen(string_name)	returns the length of string name.
2)	strcpy(destination, source)	copies the contents of source string to destination string.
3)	strcat(first_string, second_string)	concats or joins first string with second string. The result of the string is stored in first string.
4)	strcmp(first_string, second_string)	compares the first string with second string. If both strings are same, it returns 0.
5)	strrev(string)	returns reverse string.
6)	strlwr(string)	returns string characters in lowercase.
7)	strupr(string)	returns string characters in uppercase.

Example of String using Function To Compare String

```
#include<stdio.h>
#include <string.h>
int main(){
char str1[20],str2[20];
printf("Enter 1st string: ");
gets(str1);//reads string from console
printf("Enter 2nd string: ");
gets(str2);
if(strcmp(str1,str2)==0)
printf("Strings are equal");
else
printf("Strings are not equal");
```

Example of String using Function To Find String Length

```
#include<stdio.h>
#include <string.h>
int main(){
char ch[20]={'j', 'a', 'v', 'a', 't', 'p', 'o', 'i', 'n', 't', '\0'};
printf("Length of string is: %d",strlen(ch));
return 0;
}
```

Example of String using Function To Reverse String

```
#include<stdio.h>
#include <string.h>
int main(){
  char str[20];
  printf("Enter string: ");
  gets(str);//reads string from console
  printf("String is: %s",str);
  printf("\nReverse String is: %s",strrev(str));
 return 0;
}
```

Math Function

No.	Function	Description
1)	ceil(number)	rounds up the given number. It returns the integer value which is greater than or equal to given number.
2)	floor(number)	rounds down the given number. It returns the integer value which is less than or equal to given number.
3)	sqrt(number)	returns the square root of given number.
4)	pow(base, exponent)	returns the power of given number.
5)	abs(number)	returns the absolute value of given number.

Example of math functions found in math.h header file

```
#include<stdio.h>
#include <math.h>
int main(){
printf("\n%f",ceil(3.6));
printf("\n%f",ceil(3.3));
printf("\n%f",floor(3.6));
printf("\n%f",floor(3.2));
printf("\n%f",sqrt(16));
printf("\n%f",sqrt(7));
printf("\n%f",pow(2,4));
printf("\n%f",pow(3,3));
printf("\n%d",abs(-12));
 return 0;
}
```

Lab Task:
Write a C program to Copy String Manually

```c
#include <stdio.h>
int main()
{
    char s1[100], s2[100], i;
    printf("Enter string s1: ");
    scanf("%s",s1);
    for(i=0; s1[i]!='\0'; ++i)
    {
        s2[i]=s1[i];
    }
    s2[i]='\0';
    printf("String s2: %s",s2);
    r
}
```

```
Enter String s1: programiz
String s2: programiz
```

You can use the strcpy( ) function to copy the content of one string to another but, this program copies the content of one string to another manually without using strcpy( ) function.

Write a C program to Find Number of Vowels, Consonants, Digits and White Space Character

```c
#include<stdio.h>

int main(){
    char line[150];
    int i,v,c,ch,d,s,o;
    o=v=c=ch=d=s=0;
    printf("Enter a line of string:\n");
    gets(line);
    for(i=0;line[i]!='\0';++i)
    {
        if(line[i]=='a' || line[i]=='e' || line[i]=='i' || line[i]=='o' || line[i]=='u' ||
line[i]=='A' || line[i]=='E' || line[i]=='I' || line[i]=='O' || line[i]=='U')
            ++v;
        else if((line[i]>='a'&& line[i]<='z') || (line[i]>='A'&& line[i]<='Z'))
            ++c;
        else if(line[i]>='0'&&c<='9')
            ++d;
        else if (line[i]==' ')
            ++s;
    }
    printf("Vowels: %d",v);
    printf("\nConsonants: %d",c);
    printf("\nDigits: %d",d);
    printf("\nWhite spaces: %d",s);
    return 0;
}
```

```
Enter a line of string:
This program is easy 2 un
Vowels: 9
Consonants: 18
Digits: 1
White spaces: 5
```

Write a C program to Reverse String

```c
#include<stdio.h>
#include<string.h>
void Reverse(char str[]);
int main(){
      char str[100];
      printf("Enter a string to reverse: ");
      gets(str);
      Reverse(str);
      printf("Reversed string: ");
      puts(str);
 return 0;
}
void Reverse(char str[]){
      int i,j;
      char temp[100];
      for(i=strlen(str)-1,j=0; i+1!=0; --i,++j)
      {
            temp[j]=str[i];
      }
      temp[j]='\0';
      strcpy(str,temp);
}
```

To solve this problem, two standard library functions strlen() and strcpy() are used to calculate length and to copy string respectively.

```
Enter a string to reverse: zimargorp
 Reversed string: programiz
```

Write a C program To Find Prime or Not Prime No

```c
#include <stdio.h>

main() {
  int n, i, c = 0;
  printf("Enter any number n:");
  scanf("%d", &n);

  //logic
  for (i = 1; i <= n; i++) {
      if (n % i == 0) {
         c++;
      }
  }

  if (c == 2) {
        printf("n is a Prime number");
  }
  else {
         printf("n is not a Prime number");
  }
  return 0;
}
```

Program Output:

```
Enter any number n: 7
```

Write a C program to sort a string in alphabetic order

```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

void sort_string(char *s)
{
    int c, d = 0, length;
    char *pointer, *result, ch;
    length = strlen(s);
    result = (char*)malloc(length+1);
    pointer = s;
    for ( ch = 'a' ; ch <= 'z' ; ch++ )
    {
        for ( c = 0 ; c < length ; c++ )
        {
            if ( *pointer == ch )
            {
                *(result+d) = *pointer;
                d++;
            }
            pointer++;
        }
        pointer = s;
    }
    *(result+d) = '\0';
    strcpy(s, result);
    free(result);
}

void main()
{
    char string[100];
    printf("Enter some text\n");
    gets(string);
    sort_string(string);
    printf("%s\n", string);
}
```

C program to sort a string in alphabetic order: For example if user will enter a string "programming" then output will be "aggimmnoprr" or output string will contain characters in alphabetical order.

```
Enter some text
game
aegm
```