

Arduino - String Object

The second type of string used in Arduino programming is the String Object.

What is an Object?

An object is a construct that contains both data and functions. A String object can be created just like a variable and assigned a value or string. The String object contains functions (which are called "methods" in object oriented programming (OOP)) which operate on the string data contained in the String object.

The following sketch and explanation will make it clear what an object is and how the String object is used.

Example

```
void setup() {  
    String my_str = "This is my string."  
    Serial.begin(9600);  
  
    // (1) print the string  
    Serial.println(my_str);  
  
    // (2) change the string to upper-case  
    my_str.toUpperCase();  
    Serial.println(my_str);  
  
    // (3) overwrite the string  
    my_str = "My new string."  
    Serial.println(my_str);  
  
    // (4) replace a word in the string  
    my_str.replace("string", "Arduino sketch");  
    Serial.println(my_str);  
  
    // (5) get the length of the string  
    Serial.print("String length is: ");  
    Serial.println(my_str.length());  
}  
  
void loop() {  
  
}
```

Result

```
This is my string.  
THIS IS MY STRING.  
My new string.  
My new Arduino sketch.  
String length is: 22
```

A string object is created and assigned a value (or string) at the top of the sketch.

```
String my_str = "This is my string." ;
```

This creates a String object with the name **my_str** and gives it a value of "This is my string.".

This can be compared to creating a variable and assigning a value to it such as an integer –

```
int my_var = 102;
```

The sketch works in the following way.

Printing the String

The string can be printed to the Serial Monitor window just like a character array string.

Convert the String to Upper-case

The string object **my_str** that was created, has a number of functions or methods that can be operated on it. These methods are invoked by using the objects name followed by the dot operator (.) and then the name of the function to use.

```
my_str.toUpperCase();
```

The **toUpperCase()** function operates on the string contained in the **my_str** object which is of type String and converts the string data (or text) that the object contains to upper-case characters. A list of the functions that the String class contains can be found in the Arduino String reference. Technically, String is called a class and is used to create String objects.

Overwrite a String

The assignment operator is used to assign a new string to the **my_str** object that replaces the old string

```
my_str = "My new string." ;
```

The assignment operator cannot be used on character array strings, but works on String objects only.

Replacing a Word in the String

The **replace()** function is used to replace the first string passed to it by the second string passed to it. **replace()** is another function that is built into the String class and so is available to use on the String object

```
my_str.
```

Getting the Length of the String

Getting the length of the string is easily done by using `length()`. In the example sketch, the result returned by `length()` is passed directly to `Serial.println()` without using an intermediate variable.

When to Use a String Object

A String object is much easier to use than a string character array. The object has built-in functions that can perform a number of operations on strings.

The main disadvantage of using the String object is that it uses a lot of memory and can quickly use up the Arduino's RAM memory, which may cause Arduino to hang, crash or behave unexpectedly. If a sketch on an Arduino is small and limits the use of objects, then there should be no problems.

Character array strings are more difficult to use and you may need to write your own functions to operate on these types of strings. The advantage is that you have control on the size of the string arrays that you make, so you can keep the arrays small to save memory.

You need to make sure that you do not write beyond the end of the array bounds with string arrays. The String object does not have this problem and will take care of the string bounds for you, provided there is enough memory for it to operate on. The String object can try to write to memory that does not exist when it runs out of memory, but will never write over the end of the string that it is operating on.

Where Strings are Used

In this chapter we studied about the strings, how they behave in memory and their operations.

The practical uses of strings will be covered in the next part of this course when we study how to get user input from the Serial Monitor window and save the input in a string.