string

[Data Types]

Description

Text strings can be represented in two ways. you can use the String data type, which is part of the core as of version 0019, or you can make a string out of an array of type char and null-terminate it. This page described the latter method. For more details on the String object, which gives you more functionality at the cost of more memory, see the [String object](https://www.arduino.cc/reference/en/language/variables/data-types/stringobject) page.

Syntax

All of the following are valid declarations for strings.

char Str1[15];  
char Str2[8] = {'a', 'r', 'd', 'u', 'i', 'n', 'o'};  
char Str3[8] = {'a', 'r', 'd', 'u', 'i', 'n', 'o', '\0'};  
char Str4[] = "arduino";  
char Str5[8] = "arduino";  
char Str6[15] = "arduino";

**Possibilities for declaring strings**

* Declare an array of chars without initializing it as in Str1
* Declare an array of chars (with one extra char) and the compiler will add the required null character, as in Str2
* Explicitly add the null character, Str3
* Initialize with a string constant in quotation marks; the compiler will size the array to fit the string constant and a terminating null character, Str4
* Initialize the array with an explicit size and string constant, Str5
* Initialize the array, leaving extra space for a larger string, Str6

**Null termination**

Generally, strings are terminated with a null character (ASCII code 0). This allows functions (like [Serial.print()](https://www.arduino.cc/reference/en/language/functions/communication/serial/print)) to tell where the end of a string is. Otherwise, they would continue reading subsequent bytes of memory that aren’t actually part of the string.

This means that your string needs to have space for one more character than the text you want it to contain. That is why Str2 and Str5 need to be eight characters, even though "arduino" is only seven - the last position is automatically filled with a null character. Str4 will be automatically sized to eight characters, one for the extra null. In Str3, we’ve explicitly included the null character (written '\0') ourselves.

Note that it’s possible to have a string without a final null character (e.g. if you had specified the length of Str2 as seven instead of eight). This will break most functions that use strings, so you shouldn’t do it intentionally. If you notice something behaving strangely (operating on characters not in the string), however, this could be the problem.

**Single quotes or double quotes?**

Strings are always defined inside double quotes ("Abc") and characters are always defined inside single quotes('A').

**Wrapping long strings**

You can wrap long strings like this:

char myString[] = "This is the first line"

" this is the second line"

" etcetera";

**Arrays of strings**

It is often convenient, when working with large amounts of text, such as a project with an LCD display, to setup an array of strings. Because strings themselves are arrays, this is actually an example of a two-dimensional array.

In the code below, the asterisk after the datatype [char](https://www.arduino.cc/reference/en/language/variables/data-types/char) “char\*” indicates that this is an array of “pointers”. All array names are actually pointers, so this is required to make an array of arrays. Pointers are one of the more esoteric parts of C++ for beginners to understand, but it isn’t necessary to understand pointers in detail to use them effectively here.

Example Code

char \*myStrings[] = {"This is string 1", "This is string 2", "This is string 3",

"This is string 4", "This is string 5", "This is string 6"

};

void setup() {

Serial.begin(9600);

}

void loop() {

for (int i = 0; i < 6; i++) {

Serial.println(myStrings[i]);

delay(500);

}

}