String()

[Data Types]

Description

Constructs an instance of the String class. There are multiple versions that construct Strings from different data types (i.e. format them as sequences of characters), including:

* a constant string of characters, in double quotes (i.e. a char array)
* a single constant character, in single quotes
* another instance of the String object
* a constant integer or long integer
* a constant integer or long integer, using a specified base
* an integer or long integer variable
* an integer or long integer variable, using a specified base
* a float or double, using a specified decimal places

Constructing a String from a number results in a string that contains the ASCII representation of that number. The default is base ten, so

String thisString = String(13);

gives you the String "13". You can use other bases, however. For example,

String thisString = String(13, HEX);

gives you the String "D", which is the hexadecimal representation of the decimal value 13. Or if you prefer binary,

String thisString = String(13, BIN);

gives you the String "1101", which is the binary representation of 13.

Syntax

String(val)  
String(val, base)  
String(val, decimalPlaces)

Parameters

val: a variable to format as a String. Allowed data types: string, char, byte, int, long, unsigned int, unsigned long, float, double.  
base: (optional) the base in which to format an integral value.  
decimalPlaces: **only if val is float or double**. The desired decimal places.

Returns

An instance of the String class.

Example Code

All of the following are valid declarations for Strings.

String stringOne = "Hello String"; // using a constant String

String stringOne = String('a'); // converting a constant char into a String

String stringTwo = String("This is a string"); // converting a constant string into a String object

String stringOne = String(stringTwo + " with more"); // concatenating two strings

String stringOne = String(13); // using a constant integer

String stringOne = String(analogRead(0), DEC); // using an int and a base

String stringOne = String(45, HEX); // using an int and a base (hexadecimal)

String stringOne = String(255, BIN); // using an int and a base (binary)

String stringOne = String(millis(), DEC); // using a long and a base

String stringOne = String(5.698, 3); // using a float and the decimal places

Functions

* LANGUAGE [charAt()](https://www.arduino.cc/reference/en/language/variables/data-types/string/functions/charat)
* LANGUAGE [compareTo()](https://www.arduino.cc/reference/en/language/variables/data-types/string/functions/compareto)
* LANGUAGE [concat()](https://www.arduino.cc/reference/en/language/variables/data-types/string/functions/concat)
* LANGUAGE [c\_str()](https://www.arduino.cc/reference/en/language/variables/data-types/string/functions/c_str)
* LANGUAGE [endsWith()](https://www.arduino.cc/reference/en/language/variables/data-types/string/functions/endswith)
* LANGUAGE [equals()](https://www.arduino.cc/reference/en/language/variables/data-types/string/functions/equals)
* LANGUAGE [equalsIgnoreCase()](https://www.arduino.cc/reference/en/language/variables/data-types/string/functions/equalsignorecase)
* LANGUAGE [getBytes()](https://www.arduino.cc/reference/en/language/variables/data-types/string/functions/getbytes)
* LANGUAGE [indexOf()](https://www.arduino.cc/reference/en/language/variables/data-types/string/functions/indexof)
* LANGUAGE [lastIndexOf()](https://www.arduino.cc/reference/en/language/variables/data-types/string/functions/lastindexof)
* LANGUAGE [length()](https://www.arduino.cc/reference/en/language/variables/data-types/string/functions/length)
* LANGUAGE [remove()](https://www.arduino.cc/reference/en/language/variables/data-types/string/functions/remove)
* LANGUAGE [replace()](https://www.arduino.cc/reference/en/language/variables/data-types/string/functions/replace)
* LANGUAGE [reserve()](https://www.arduino.cc/reference/en/language/variables/data-types/string/functions/reserve)
* LANGUAGE [setCharAt()](https://www.arduino.cc/reference/en/language/variables/data-types/string/functions/setcharat)
* LANGUAGE [startsWith()](https://www.arduino.cc/reference/en/language/variables/data-types/string/functions/startswith)
* LANGUAGE [substring()](https://www.arduino.cc/reference/en/language/variables/data-types/string/functions/substring)
* LANGUAGE [toCharArray()](https://www.arduino.cc/reference/en/language/variables/data-types/string/functions/tochararray)
* LANGUAGE [toDouble()](https://www.arduino.cc/reference/en/language/variables/data-types/string/functions/todouble)
* LANGUAGE [toInt()](https://www.arduino.cc/reference/en/language/variables/data-types/string/functions/toint)
* LANGUAGE [toFloat()](https://www.arduino.cc/reference/en/language/variables/data-types/string/functions/tofloat)
* LANGUAGE [toLowerCase()](https://www.arduino.cc/reference/en/language/variables/data-types/string/functions/tolowercase)
* LANGUAGE [toUpperCase()](https://www.arduino.cc/reference/en/language/variables/data-types/string/functions/touppercase)
* LANGUAGE [trim()](https://www.arduino.cc/reference/en/language/variables/data-types/string/functions/trim)

Operators

* LANGUAGE [[] (element access)](https://www.arduino.cc/reference/en/language/variables/data-types/string/operators/elementaccess)
* LANGUAGE [+ (concatenation)](https://www.arduino.cc/reference/en/language/variables/data-types/string/operators/concatenation)
* LANGUAGE [+= (append)](https://www.arduino.cc/reference/en/language/variables/data-types/string/operators/append)
* LANGUAGE [== (comparison)](https://www.arduino.cc/reference/en/language/variables/data-types/string/operators/comparison)
* LANGUAGE [> (greater than)](https://www.arduino.cc/reference/en/language/variables/data-types/string/operators/greaterthan)
* LANGUAGE [>= (greater than or equal to)](https://www.arduino.cc/reference/en/language/variables/data-types/string/operators/greaterthanorequalto)
* LANGUAGE [< (less than)](https://www.arduino.cc/reference/en/language/variables/data-types/string/operators/lessthan)
* LANGUAGE [<= (less than or equal to)](https://www.arduino.cc/reference/en/language/variables/data-types/string/operators/lessthanorequalto)
* LANGUAGE [!= (different from)](https://www.arduino.cc/reference/en/language/variables/data-types/string/operators/differentfrom)
* EXAMPLE [String Tutorials](https://www.arduino.cc/en/Tutorial/BuiltInExamples#strings)