

## Dart Programming - String

The String data type represents a sequence of characters. A Dart string is a sequence of UTF 16 code units.

String values in Dart can be represented using either single or double or triple quotes. Single line strings are represented using single or double quotes. Triple quotes are used to represent multi-line strings.

The syntax of representing string values in Dart is as given below –

### Syntax

```
String variable_name = 'value'

OR

String variable_name = 'value'

OR

String variable_name = '''line1
line2'''

OR

String variable_name= '''line1
line2'''
```

The following example illustrates the use of String data type in Dart.

[Live Demo](#)

```
void main() {
  String str1 = 'this is a single line string';
  String str2 = "this is a single line string";
  String str3 = '''this is a multiline line string''';
  String str4 = """this is a multiline line string""";

  print(str1);
  print(str2);
  print(str3);
  print(str4);
}
```

It will produce the following **Output** –

```
this is a single line string
this is a single line string
this is a multiline line string
this is a multiline line string
```

Strings are immutable. However, strings can be subjected to various operations and the resultant string can be a stored as a new value.

## String Interpolation

The process of creating a new string by appending a value to a static string is termed as **concatenation** or **interpolation**. In other words, it is the process of adding a string to another string.

The operator plus (+) is a commonly used mechanism to concatenate / interpolate strings.

### Example 1

Live Demo

```
void main() {  
    String str1 = "hello";  
    String str2 = "world";  
    String res = str1+str2;  
  
    print("The concatenated string : ${res}");  
}
```

It will produce the following **output** –

```
The concatenated string : Helloworld
```

### Example 2

You can use "\${}" can be used to interpolate the value of a Dart expression within strings. The following example illustrates the same.

Live Demo

```
void main() {  
    int n=1+1;  
  
    String str1 = "The sum of 1 and 1 is ${n}";  
    print(str1);  
  
    String str2 = "The sum of 2 and 2 is ${2+2}";  
    print(str2);  
}
```

It will produce the following **output** –

```
The sum of 1 and 1 is 2  
The sum of 2 and 2 is 4
```

## String Properties

The properties listed in the following table are all read-only.

Sr.No	Property & Description
1	codeUnits Returns an unmodifiable list of the UTF-16 code units of this string.
2	isEmpty Returns true if this string is empty.
3	Length Returns the length of the string including space, tab and newline characters.

### Methods to Manipulate Strings

The String class in the **dart: core library** also provides methods to manipulate strings. Some of these methods are given below –

Sr.No	Methods & Description
1	toLowerCase() Converts all characters in this string to lower case.
2	toUpperCase() Converts all characters in this string to upper case.
3	trim() Returns the string without any leading and trailing whitespace.
4	compareTo() Compares this object to another.
5	replaceAll() Replaces all substrings that match the specified pattern with a given value.
6	split() Splits the string at matches of the specified delimiter and returns a list of substrings.
7	substring() Returns the substring of this string that extends from startIndex, inclusive, to endIndex, exclusive.
8	toString() Returns a string representation of this object.
9	codeUnitAt() Returns the 16-bit UTF-16 code unit at the given index.