

# String in Dart

## String In Dart

String helps you to store text based data. In String, you can represent your name, address, or complete book. It holds a series or sequence of characters – letters, numbers, and special characters. You can use single or double, or triple quotes to represent String.

### Example: String In Dart

Single line String is written in single or double quotes, whereas multi-line strings are written in triple quotes. Here is an example of it:

```
void main() {
  String text1 = 'This is an example of a single-line string.';
  String text2 = "This is an example of a single line string using double quotes.";
  String text3 = """This is a multiline line
string using the triple-quotes.
This is tutorial on dart strings.
""";
  print(text1);
  print(text2);
  print(text3);
}
```

## String Concatenation

You can combine one String with another string. This is called concatenation. In Dart, you can use the `+` operator or use interpolation to concatenate the String. Interpolation makes it easy to read and understand the code.

```
void main() {
  String firstName = "John";
  String lastName = "Doe";
  print("Using +, Full Name is "+firstName + " " + lastName+".");
  print("Using interpolation, full name is ${firstName} ${lastName}.");
}
```

## Properties Of String

- `codeUnits`: Returns an unmodifiable list of the UTF-16 code units of this string.
- `isEmpty`: Returns true if this string is empty.

- `isNotEmpty`: Returns false if this string is empty.
- `length`: Returns the length of the string including space, tab, and newline characters.

## Methods Of String

- `toLowerCase()`: Converts all characters in this string to lowercase.
- `toUpperCase()`: Converts all characters in this string to uppercase.
- `trim()`: Returns the string without any leading and trailing whitespace.
- `compareTo()`: Compares this object to another.
- `replaceAll()`: Replaces all substrings that match the specified pattern with a given value.
- `split()`: Splits the string at matches of the specified delimiter and returns a list of substrings.
- `toString()`: Returns a string representation of this object.
- `substring()`: Returns the text from any position you want.
- `codeUnitAt()`: Returns the 16-bit UTF-16 code unit at the given index.

## String Methods Example In Dart

Here you will see various string methods that can help your work a lot better and faster.

### Converting String To Uppercase and Lowercase

You can convert your text to lower case using `.toLowerCase()` and convert to uppercase using `.toUpperCase()` method.

```
//Example of toUpperCase() and toLowerCase()
void main() {
  String address1 = "Florida"; // Here F is capital
  String address2 = "TexAs"; // Here T and A are capital
  print("Address 1 in uppercase: ${address1.toUpperCase()}");
  print("Address 1 in lowercase: ${address1.toLowerCase()}");
  print("Address 2 in uppercase: ${address2.toUpperCase()}");
  print("Address 2 in lowercase: ${address2.toLowerCase()}");
}
```

### Trim String In Dart

Trim is helpful when removing leading and trailing spaces from the text. This trim method will remove all the starting and ending spaces from the text. You can also use `trimLeft()` and `trimRight()` methods to remove space from left and right, respectively.

Note: The trim() method in Dart doesn't remove spaces in the middle.

```
//Example of trim()
void main() {
  String address1 = " USA"; // Contain space at leading.
  String address2 = "Japan "; // Contain space at trailing.
  String address3 = "New Delhi"; // Contains space at middle.

  print("Result of address1 trim is ${address1.trim()}");
  print("Result of address2 trim is ${address2.trim()}");
  print("Result of address3 trim is ${address3.trim()}");
  print("Result of address1 trimLeft is ${address1.trimLeft()}");
  print("Result of address2 trimRight is ${address2.trimRight()}");
}
```

## Compare String In Dart

In Dart, you can compare two strings. It will give the result 0 when two texts are equal, 1 when the first String is greater than the second, and -1 when the first String is smaller than the second.

```
//Example of compareTo()
void main() {
  String item1 = "Apple";
  String item2 = "Ant";
  String item3 = "Basket";

  print("Comparing item 1 with item 2: ${item1.compareTo(item2)}");
  print("Comparing item 1 with item 3: ${item1.compareTo(item3)}");
  print("Comparing item 3 with item 2: ${item3.compareTo(item2)}");
}
```

## Replace String In Dart

You can replace one value with another with the replaceAll("old", "new") method in Dart. It will replace all the "old" words with "new". Here in this example, this will replace milk with water.

```
//Example of replaceAll()
void main() {
String text = "I am a good boy I like milk. Doctor says milk is good for health.";

String newText = text.replaceAll("milk", "water");

print("Original Text: $text");
print("Replaced Text: $newText");
}
```

## Split String In Dart

You can use the dart split method if you want to split String by comma, space, or other text. It will help you to split String to list.

```
//Example of split()
void main() {
String allNames = "Ram, Hari, Shyam, Gopal";

List<String> listNames = allNames.split(",");
print("Value of listName is $listNames");

print("List name at 0 index ${listNames[0]}");
print("List name at 1 index ${listNames[1]}");
print("List name at 2 index ${listNames[2]}");
print("List name at 3 index ${listNames[3]}");

}
```

## ToString In Dart

In dart, `toString()` represents String representation of the value/object.

```
//Example of toString()
void main() {
int number = 20;
String result = number.toString();

print("Type of number is ${number.runtimeType}");
print("Type of result is ${result.runtimeType}");

}
```

## SubString In Dart

You can use substring in Dart when you want to get a text from any position.

```
//Example of substring()
void main() {
  String text = "I love computer";
  print("Print only computer: ${text.substring(7)}"); // from index 6 to the last index
  print("Print only love: ${text.substring(2,6)}"); // from index 2 to the 6th index
}
```

## Reverse String In Dart

If you want to reverse a String in Dart, you can reverse it using a different solution. One solution is here.

```
void main() {
  String input = "Hello";
  print("$input Reverse is ${input.split('').reversed.join()}");
}
```

## How To Capitalize First Letter Of String In Dart

If you want to capitalize the first letter of a String in Dart, you can use the following code.

```
//Example of capitalize first letter of String
void main() {
  String text = "hello world";
  print("Capitalized first letter of String: ${text[0].toUpperCase()}${text.substring(1)}");
}
```