String Interpolation

In the following lesson, you will learn how to concatenate strings using string interpolation.

We'll cover the following

String Concatenation

To concatenate two strings means to join them together. Concatenation of two or more strings is done using the `+` operator.

```
main() {
  String s1 = "First half of the string. ";
  String s2 = "Second half of the string";
  print(s1 + s2);
}
```

The code above is pretty simple. On **line 4** we are concatenating the first string, `s1`, and the second string, `s2`, and then printing the concatenated string.

String Interpolation

String interpolation is the ability to create new strings or modify existing ones by embedding them with expressions. Expressions are evaluated, and the resulting values are converted into strings and concatenated with the enclosing string. Interpolation is Dart's more concise and efficient alternative to string concatenation. However, string interpolation is a lot more complex than simple concatenation which allows a lot more flexibility.

Syntax

An un-escaped `$` character in a string signifies the beginning of an interpolated expression.

The `$` sign can be followed by a single *identifier id* that does not contain the `$` character.

```
"optional string $variableIdentifier optional string"
```

The `$` sign can also be followed by an expression delimited by curly brackets.

```
"optional string ${expression} optional string"
```

Learning by Example

Let’s look at some different coding examples to get a better idea of how string interpolation works.

```
main() {
  String country = "Japan";

  print("I want to visit $country");
}
```

In the above code, assign a value to the `country` variable based on the country you want to visit.

Instead of changing the `print` statement each time, all we have to do is change the value of the variable being interpolated.

Let’s now try to embed a mathematical expression in a string using the curly brackets (`{}`).

```
main() {
  print("The sum of 5 and 3 equals ${5+3}.");
}
```

In the code above, `5+3` is our expression which the compiler processes and interprets to 8, as can be seen in the output when you press RUN.

Multiple Lines

You can concatenate strings using adjacent string literals as well.

> Notice how we have used both double (`"`) and single (`'`) quotation marks.

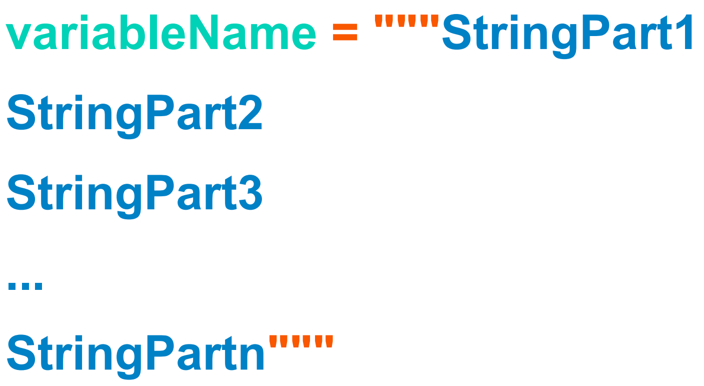
```
main() {
  var s1 = 'String '
    'concatenation'
    " works even over line breaks.";

  print(s1);
}
```

When we run the code snippet above, we get one complete string on a single line.

But what if we actually wanted to have a string on multiple lines?

You can create multiline strings in Dart. To do so, you simply write your string as you want to print it and surround it with three double quotation marks or three single quotation marks.



```
variableName = """StringPart1
StringPart2
StringPart3
...
StringPartn"""
```

Let's look at an example below.

```
main() {
  var multilineString = """This is a
multiline string
consisting of

multiple lines""";

  print(multilineString);
}
```

In the next lesson, you will be challenged to solve a problem using string interpolation.