Literals String and String Interpolation

Dart string

A string is a sequence of UTF-16 code units. It is used to represent some text in a program. A character may be represented by multiple code points. Each code point can have one or two code units.

Strings are immutable in Dart. There are methods such as `toLowerCase` or `split` that return a modified string, but the original string is intact.

A *rune* is an integer representing a Unicode code point. The `runes` property of a string returns its runes.

A string in Dart is an object. There are several methods that we can call on a string object.

Dart simple string

The following is a simple Dart program with strings.

main.dart

```
void main() {
  var text = "There are six falcons";
  print(text);

  var len = text.length;
  print('The string has ' + len.toString() + ' character
s');

  var word = 'falcon ';
  print(word * 3);
}
```

String literals are delimited with single or double qoutes.

```
var len = text.length;
```

The `length` property returns the length of the string, i.e. the number of UTF-16 code units in the string.

```
print('The string has ' + len.toString() + ' characters');
```

Strings can be concatenated with the `+` operator.

```
var word = 'falcon ';
print(word * 3);
```

The `*` operator multiplies strings.

```
$ dart main.dart
There are six falcons
The string has 21 characters
falcon falcon falcon
```

Dart string length

The `length` property returns the length of a string in code units.

main.dart

```
void main() {
  var word = 'čerešňa';

  print(word.length);
  print(word.codeUnits);
  print(word.runes);

  var word2 = "合気道";
  print(word2.length);
  print(word2.codeUnits);
  print(word2.runes);
}
```

In the example, we have two strings with wide characters. The length property returns the number of visible characters. The `codeUnits` property returns an unmodifiable list of the UTF-16 code units of this string. The `runes` property returns an iterable of Unicode code-points of a string.

```
$ dart main.dart
7
[269, 101, 114, 101, 353, 328, 97]
(269, 101, 114, 101, 353, 328, 97)
3
[21512, 27671, 36947]
(21512, 27671, 36947)
```

Dart string interpolation

String interpolation is the process of evaluating a string containing variables and expressions. When an interpolated string is evaluated the variables and expressions are replaced with their corresponding values.

In Dart, the `$` is used to interpolate variables and `${}` expressions.

main.dart

```
void main() {
  var name = "John Doe";
  var occupation = "gardener";

  print("$name is a $occupation");

  var x = 12;
  var y = 14;

  print("${x} * ${y} = ${x * y}");
}
```

In the example, we interpolate two variables and an expression.

```
print("$name is a $occupation");
```

Variables are interpolated by prepending their names with the dollar sign.

```
print("${x} * ${y} = ${x * y}");
```

If we need to evaluate more complex expressions, we use `${}`.

```
$ dart main.dart
John Doe is a gardener
12 * 14 = 168
```

If we need to output a dollar sign, we escape it: `\$`.

main.dart

```
void main() {
  var item = "beer";
  var price = 4.5;

  print("The price of a $item is \$$price");
}
```

The example prints the price of a beer in dollars.

```
$ dart main.dart
The price of a beer is $4.5
```