Understanding the Code

In this lesson, we will go over each line of the Hello World program we wrote in the previous lesson.

We'll cover the following

Just to refresh our minds, let's take another look at the code for our *Hello World* application.

```
main() {
  // Printing the text 'Hello World'
  print("Hello World");
}
```

We've already gone over the `main()` function; let's move on to the next line.

The Print Statement

On **line 3** of the program, we are using a **print** statement which prints the text *Hello World*. `print()` is used to display the output of the code on the console. It follows this syntax:

```
print(Insert what you want to print here)
```

Since *Hello World* is text, to be able to print it, we need to enclose it in quotation marks. The syntax for displaying text is as follows:

```
print("Insert text to be printed")
print("Insert text to be printed")
```

Nothing Without a Semicolon

You might have noticed that the print statement on **line 3** ends with a semicolon.

```
print("Hello World");
```

In Dart, just as most programming languages, each statement needs to end with a semi-colon for the program to execute successfully. Try removing the semi-colon in the code snippet above and see what happens.

Comments

Line 2 of the program is a **comment**. Comments come in handy for developers that want to read or study code someone else has written. They enable you to write descriptions about the code without it affecting the program in any way what so ever. Our comment is letting the reader know that the program is printing the text *Hello World*. The general syntax is as follows:

```
// Insert comment here
```

You simply need to type two forward slashes followed by the description you want to write.

Imperative Programming

The code that appears inside of the `main()` function gets executed in order of appearance. Let's modify the code in the snippet above and try printing *From Dart* along with *Hello World*.

```
main() {
  print("Hello World");
  print("From Dart");
}
```

When you run the code above, the print statement on **line 2** will execute first and then the second print statement on **line 3** will execute.

This style of programming is called **imperative programming**. It is essentially a programming paradigm wherein you write a set of instructions that execute in sequential order. Imperative programming doesn't necessarily describe *what* the program should accomplish, rather it shows *how* the program should accomplish it. So, in this case, we have created a program that goes first and prints out the text *Hello World* and then moves on to the next line and prints the text *From Dart*.

Let's take a look at an interactive program in the next lesson.