

A Brief Introduction to Functions and Methods

In the following lesson, you will be given a very brief introduction on functions and methods.

We'll cover the following ^

- Functions
- How Do They Work?
- Functions in Dart

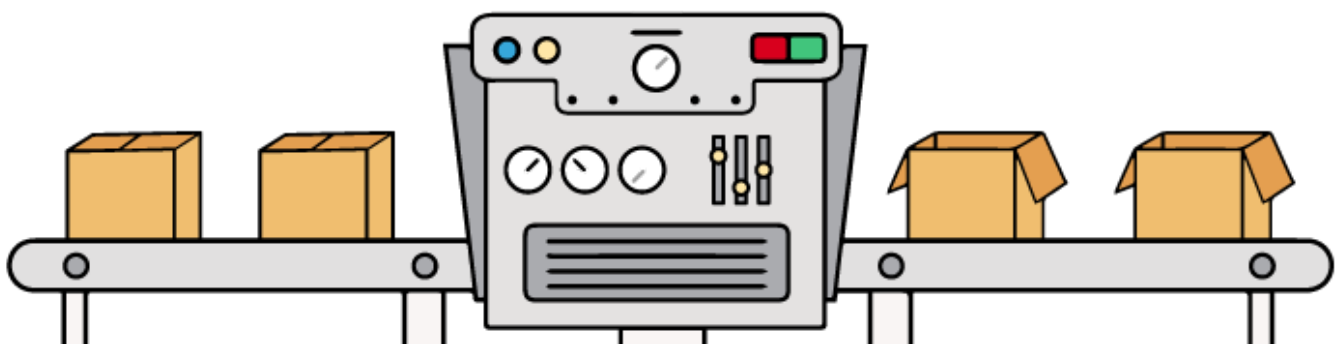
Functions

In computer programming, a function or a method is a block of code that performs a specific task. The block of code is given a name, much like a variable. The function is called using this name whenever that specific task needs to be performed. This removes the need to type the same code over and over again; all you have to do is call the function's name.

How Do They Work?

Like mathematical functions, programming functions take in an input, known as an **argument**, perform some operations on that input, and then return the resulting output.

It's almost like a conveyer belt in a factory with items entering the machine from one end of the conveyer belt and coming out, completely modified, on the other end. However, as the machine remains the same throughout its life, it will modify every item that enters it the same way.



Functions in Dart

We can divide functions into two broad categories:

- Built-In Functions
- User-Defined Functions

User-defined functions are functions that users create themselves. We have a whole [chapter](#) dedicated to just these, so they aren't of any concern to us now.

Built-in functions are functions that are predefined by Dart and are part of their libraries. All we have to do to use them is call their function name.

For now, we can say that **methods** are what we call built-in functions. Methods are a bit more complicated, but this is sufficient for now.

The printing statement we've been using throughout the chapter is also a built-in function.

```
main() {  
  var printMe = "Hello World";  
  
  print(printMe);  
}
```



In the code above, `print` is a method that performs the specific task **print**. `printMe` is the argument we pass to the method and "Hello World" is the resultant output. We've been using functions this whole time!

While `print` is a method, it is a very simple one which only requires passing it an argument that can be of any type. However, most methods require you to call them on an object. For example, `objectName.method(arguments)` means that the method is being called on `objectName` and `arguments` are parameters passed to the `method`. The method will perform some action on the data stored in `objectName`.

Most methods allow you to only pass an argument of a specific data type. Let's call

one of Dart's built-in methods, `indexOf` to get a better idea of how this works.

`indexOf` is called on a string and you pass it one argument of type `String`. It is used to calculate the starting index of a specified substring within a string.

Index is the position of an object. In Dart it always starts with **0**.

```
main() {  
  String s1 = "hello";  
  print(s1.indexOf("ll"));  
}
```



In the code above, `s1` is the object we are calling the method `indexOf` on and `ll` is the argument we are passing it. The output is `2` as the starting index of `ll` is located at the 2nd index in the string "Hello World".

We will be using built-in functions throughout the course from this point onwards.

Let's move on to data structures in the next lesson.