Dart and flutter

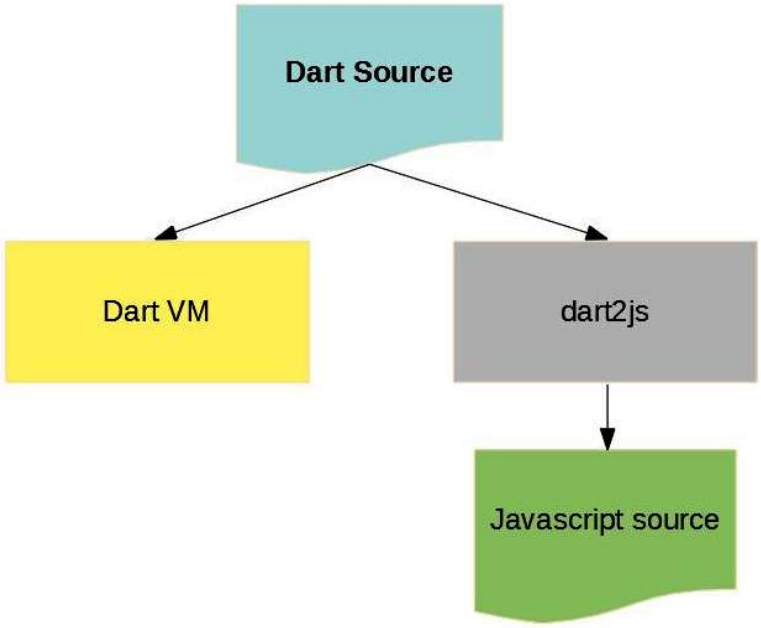
Dart is the language that you'll use to develop your Flutter applications, and you'll learn the basic of Dart language at this course.

Dart is extremely productive. If you already know an object-oriented programming language such as C++, C#, or Java, it will be easier for you to learn the Dart language

The Dart language

- **How Dart works**

- To understand where the language's flexibility came from, we need to know how we can run Dart code. This is done in two ways:
- Dart **Virtual Machines (VMs)**
- JavaScript compilations



Dart VM and JavaScript compilation

- Dart code can be run in a *Dart-capable environment*. A Dart-capable environment provides essential features to an app, such as the following:

- Runtime systems
- Dart core libraries
- Garbage collectors

- The execution of Dart code operates in two modes—**Just-In-Time** (**JIT**) compilation or **Ahead-Of-Time** (**AOT**) compilation:

The execution of Dart code

- A JIT compilation is where the source code is loaded and compiled to native machine code by the Dart VM on the fly. It is used to run code in the command line interface or when you are developing a mobile app in order to use features such as debugging and hot reloading.

- An AOT compilation is where the Dart VM and your code are precompiled and the VM works more like a Dart runtime system, providing a garbage collector and various native methods from the Dart software development kit (**SDK**) to the application.

The way Flutter is designed is heavily influenced by the Dart language.

Flutter's hot reload feature helps you quickly and easily experiment, build UIs, add features, and fix bugs. Hot reload works by injecting updated source code files into the running **Dart Virtual Machine (VM).** After the VM updates classes with the new versions of fields and functions, the Flutter framework automatically rebuilds the widget tree, allowing you to quickly view the effects of your changes.

Dart Features

- **Open Source**
- Dart is an open-source programming language, which means it is freely available.

- **Platform Independent**
- Dart supports all primary operating systems such as Windows, Linux , Macintosh, etc. The Dart has its own Virtual Machine which known as Dart VM, that allows us to run the Dart code in every operating system.

- **Object-Oriented**
- Dart is an object-oriented programming language and supports all oops concepts such as classes, inheritance, interfaces and optional typing features.

Dart Features

- **Concurrency**
- Dart is an asynchronous programming language, which means it supports multithreading using Isolates. The isolates are the independent entities that are related to threads but don't share memory and establish the communication between the processes by the message passing.

- **Extensive Libraries**
- Dart consists of many useful inbuilt libraries including SDK (Software Development Kit), core, math , async, math, convert, html, IO, etc.

Dart Features

- **Flexible Compilation**

  Dart provides the flexibility to compile the code and fast as well. It supports two types of compilation processes, AOT (Ahead of Time) and JIT (Just-in-Time). The Dart code is transmitted in the other language that can run in the modern web-brewers.

- **Type Safe**

  The Dart is the type safe language, which means it uses both static type checking and runtime checks to confirm that a variable's value always matches the variable's static type, sometimes it known as the sound typing.

  Although types are required, type annotations are optional because of type interference. This makes it code more readable. The other advantage to being type-safe language is, when we change the part of code, the system warns us about that modification that we have modified earlier.