**Course: 405-02: Mobile Application Development – 2**

**Unit-3: Working with DART:**

3.1 DART overview, concept, features and installation

**What is Dart?**

Dart is a general-purpose, high-level modern programming language which is originally developed by Google. It is the new programming language which is emerged in 2011, but its stable version was released in June 2017. Dart is not so popular at that time, but It gains popularity when it is used by the Flutter.

Dart is a dynamic, class-based, object-oriented programming language with closure and lexical scope. Syntactically, it is quite similar to [Java](https://www.javatpoint.com/java-tutorial), [C](https://www.javatpoint.com/c-programming-language-tutorial), and JavaScript. If you know any of these programming languages, you can easily learn the Dart programming language.

Dart is an open-source programming language which is widely used to develop the mobile application, modern web-applications, desktop application, and the [Internet of Things](https://www.javatpoint.com/iot-tutorial) (IoT) using by Flutter framework. It also supports a few advance concepts such as interfaces, mixins, abstract classes, refield generics, and type interface. It is a compiled language and supports two types of compilation techniques.

* **AOT (Ahead of Time) -** It converts the Dart code in the optimized JavaScript code with the help of the dar2js compiler and runs on all modern web-browser. It compiles the code at build time.
* **JOT (Just-In-Time) -** It converts the byte code in the machine code (native code), but only code that is necessary.

**History**

Dart was revealed for the first time in the GOTO conference in the month of 10th - 12th October 2011 at Aarhus, Denmark. It is initially designed by the **Lars bark and Kespar** and developed by Google.

The first version 1.0 of Dart was released on November 14th, 2013, intended as a replacement of [JavaScript](https://www.javatpoint.com/javascript-tutorial).

In July 2014, the first edition of Dart language was approved by Ecma International approved at its 107th General Assembly.

The first version was criticized due to a malfunction on the web and this plan was dropped in 2015 with the 1.9 release of Dart.

The second version of Dart 2.0 was released in August, including a sound type system.

The recent version Dart 2.7 is supplemented with the extension method, which enables us to add any type of functionality.

**Why Dart?**

We define the characteristics of Dart in the following point.

* Dart is a platform-independent language and supports all operating systems such as Windows, Mac, Linux, etc.
* It is an open-source language, which means it available free for everyone. It comes with a BSD license and recognized by the ECMA standard.
* It is an object-oriented programming language and supports all features of oops such as inheritance, interfaces, and optional type features.
* Dart is very useful in building real-time applications because of its stability.
* Dart comes with the dar2js compiler which transmits the Dart code into JavaScript code that runs on all modern web browser.
* The stand-alone Dart VM permits Dart code to run in a command-line interface environment.

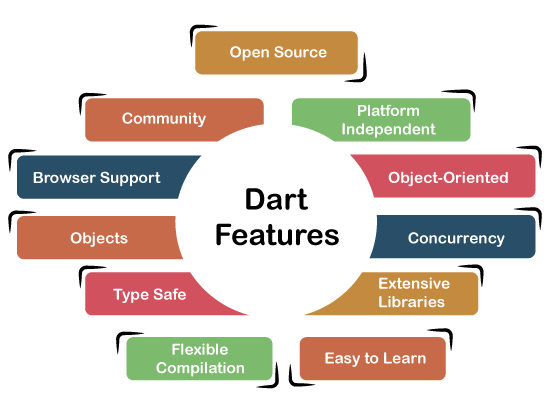
**Key Points to Remember**

Before learning the Dart, we should keep these concepts in mind. These concepts are given below.

* Everything in Dart is treated as an object including, numbers, Boolean, function, etc. like Python. All objects inherit from the Object class.
* Dart tools can report two types of problems while coding, warnings and errors. Warnings are the indication that your code may have some problem, but it doesn't interrupt the code's execution, whereas error can prevent the execution of code.
* Dart supports sound typing. We will learn about this in the next tutorial.
* Dart supports generic types, like **List<int>**(a list of integers) or **List<dynamic>** (a list of objects of any type).

**Dart Features**

The Dart is an object-oriented, open-source programming language which contains many useful features. It is the new programming language and supports an extensive range of programming utilities such as interface, collections, classes, dynamic and optional typing. It is developed for the server as well as the browser. Below is the list of the important Dart features.



**Open Source**

Dart is an open-source programming language, which means it is freely available. It is developed by Google, approved by the ECMA standard, and comes with a BSD license.

**Platform Independent**

Dart supports all primary operating systems such as [Windows](https://www.javatpoint.com/windows), [Linux](https://www.javatpoint.com/linux-tutorial), 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. It also supports advance concepts like mixin, abstract, classes, reified generic, and robust type system.

**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. The message should be serialized to make effective communication. The serialization of the message is done by using a snapshot that is generated by the given object and then transmits to another isolate for desterilizing.

**Extensive Libraries**

Dart consists of many useful inbuilt libraries including SDK (Software Development Kit), core, [math](https://www.javatpoint.com/math), async, math, convert, [html](https://www.javatpoint.com/html-tutorial), IO, etc. It also provides the facility to organize the Dart code into libraries with proper namespacing. It can reuse by the import statement.

**Easy to learn**

As we discussed in the previous section, learning the Dart is not the Hercules task as we know that Dart's syntax is similar to [Java](https://www.javatpoint.com/java-tutorial), [C#](https://www.javatpoint.com/c-sharp-tutorial), [JavaScript](https://www.javatpoint.com/javascript-tutorial), [kotlin](https://www.javatpoint.com/kotlin-tutorial), etc. if you know any of these languages then you can learn easily the Dart.

**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.

**Objects**

The Dart treats everything as an object. The value which assigns to the variable is an object. The functions, numbers, and strings are also an object in Dart. All objects inherit from Object class.

**Browser Support**

The Dart supports all modern web-browser. It comes with the dart2js compiler that converts the Dart code into optimized JavaScript code that is suitable for all type of web-browser.

**Community**

Dart has a large community across the world. So if you face problem while coding then it is easy to find help. The dedicated developers' team is working towards enhancing its functionality.

Here we have discussed essential features of the Dart language. We will more concepts of Dart language in upcoming tutorials.

Dart Installation

To learn the Dart, we need to set up the Dart programming environment to our local machine. We are describing the following instructions to install the Dart SDK (Software Development Kit) in various operating systems. If you have already installed it, then you can skip this part.

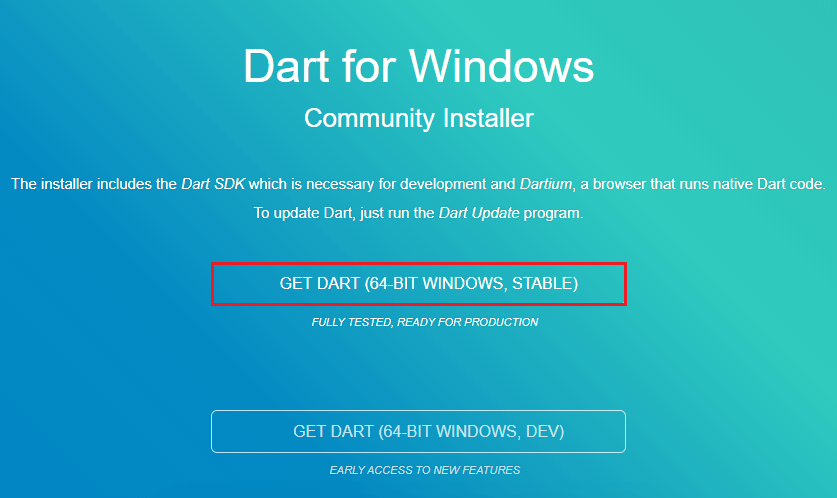
Install the Dart SDK on Windows

Follow the below instructions to install Dark SDK in Windows.

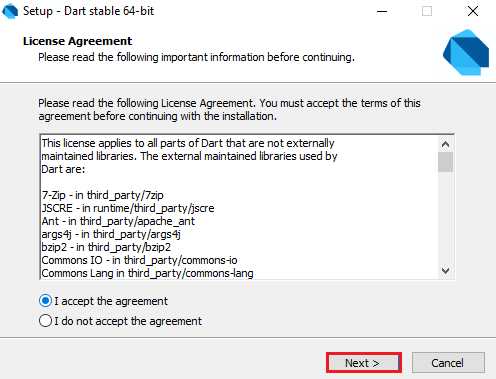
**Step -1:** Go to the browser and type the following link to download the SDK.

<http://www.gekorm.com/dart-windows/>

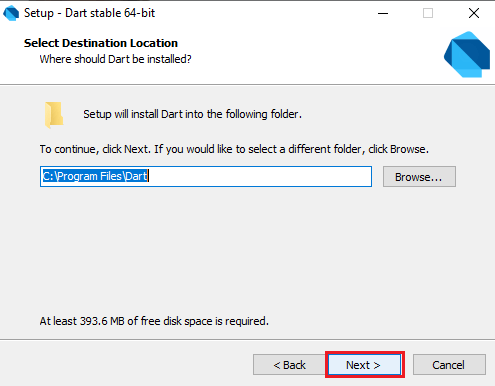
It will open the given page. Click on the following link.

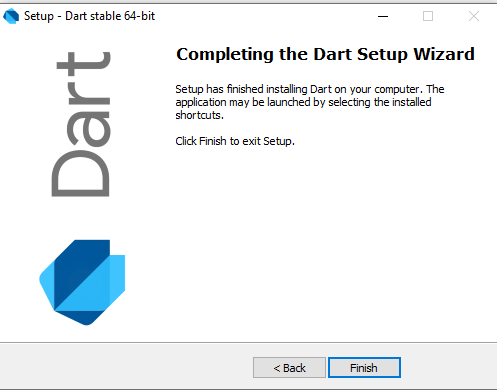


**Step - 2:** Run the Dart installer(It is the .exe file that we downloaded in the previous step) and click on the **Next** button.

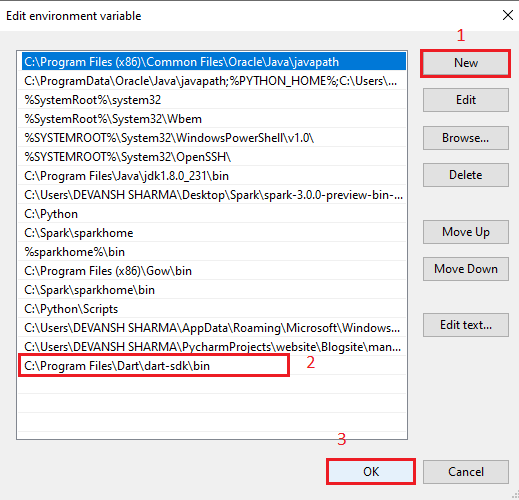


**Step - 3:** It provides the option to select the Dart installation path. After the path is selected, click on the **Next** button.

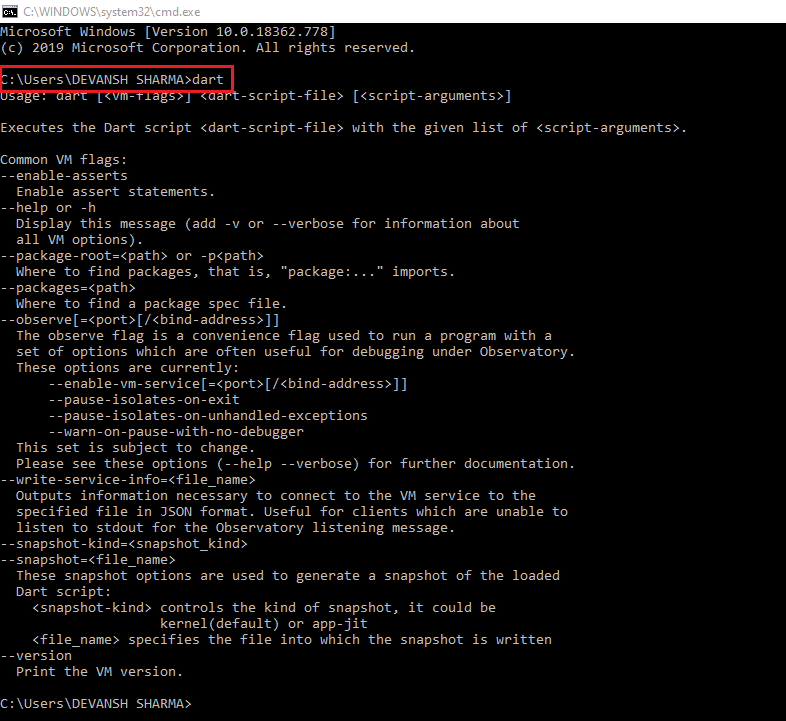




**Step - 4:** After the download is completed, set the PATH environment variable to "**C:\Program Files\Dart\dart-sdk\bin**" in advance system properties.



**Step - 5:** Now open the terminal and verify the Dart installation by typing dart.



If it is successfully installed then it looks like the above image.

**Install the Dart SDK on Linux**

The steps of Dart installation on [Linux](https://www.javatpoint.com/linux-tutorial) is given below.

Before installing the [Dart](https://www.javatpoint.com/dart-programming), if you are Debian/Ubuntu on AMD64(64-bit Intel) in your local machine, you can install the Dart through one of the following options.

* Install using apt-get
* Install a Debian package

Installation using apt-get

**Step -1:** Type the following commands for a one-time setup.

$sudo apt-get update

$ sudo apt-get install apt-transport-https

$ sudo sh -c 'wget -qO- https://dl-ssl.google.com/linux/linux\_signing\_key.pub | apt-key add -'

$ sudo sh -c 'wget -qO- https://storage.googleapis.com/download.dartlang.org/linux/debian/dart\_stable.list > /etc/apt/so

**Step - 2:** Type the following command in terminal to install the Dart SDK using apt-get option.

$sudo apt-get update

$ sudo apt-get install dart

It will successfully download the Dart SDK.

Installation a Debian Package

We can download **Dart SDK** as a Debian package in the **.deb package** format. To make all Dart binaries accessible, we have to change the PATH by typing the following command.

export PATH="$PATH:/usr/lib/dart/bin"

To change the PATH for upcoming terminal sessions, use a below command:

$ echo 'export PATH="$PATH:/usr/lib/dart/bin"' >> ~/.profile

**Install the Dark SDK on Mac**

**Step -1:** We should have **Homebrew** package manager, but if we don't have it then install the **Homebrew** and run the following command. It will successfully download the Dart on the Mac.

$ brew tap dart-lang/dart

$ brew install dart

**Step -2:** To verify which version we have installed, use the following command.

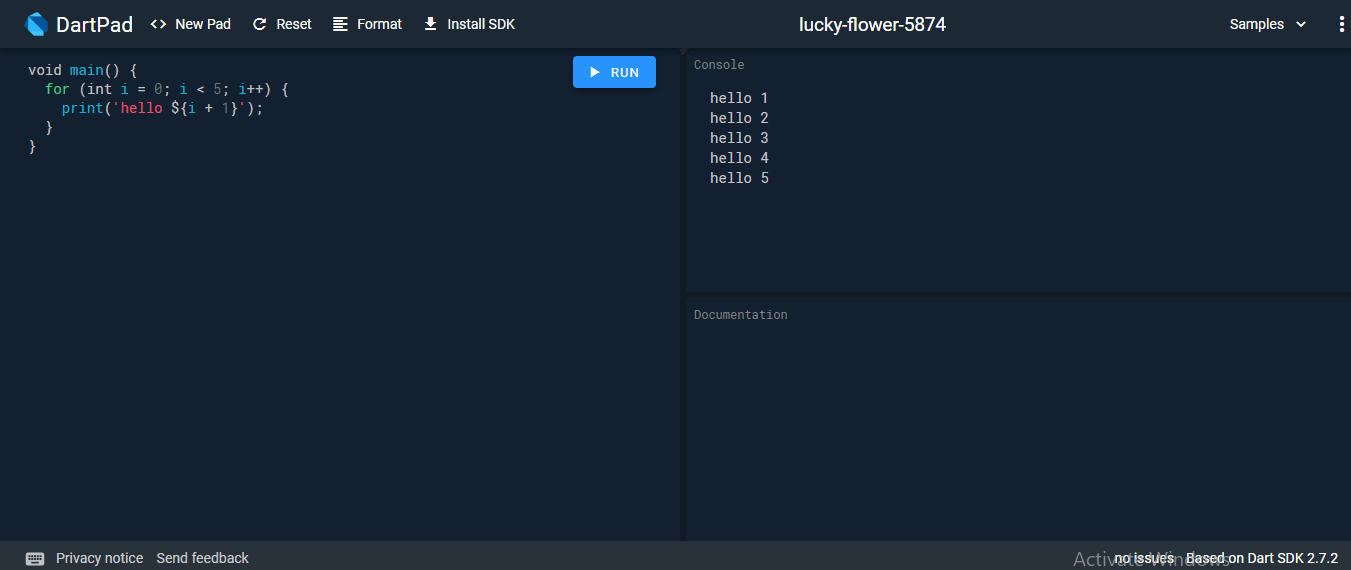
$brew info dart

3.2 Online editor DartPad and dart2js tool

3.3 Executing Dart basic code using Command line, DartPad and IDE

**Online Dart Editor**

We have discussed Dark installation on the various operating systems so far, but if we do not want to install Dart then there is an online Dark editor (Known as DartPad) is available to run the Dark programs. The online DartPad is provided at <https://dartpad.dev/>. The DartPad offers to execute the dart scripts and display HTML and also console output. The online DartPad looks like the below image



**Dart IED Support**

The Eclipse, IntelliJ, and WebStorm are the IDEs from the Jet brains that support the Dart Programming, but WebStorm is more popular than others. We can download it from [https://www.jetbrains.com/webstorm/download/#section=windows-version.](https://www.jetbrains.com/webstorm/download/#section=windows-version)

**The dart2js Tool**

The Dark SDK comes with the dart2js tool, which transmits the Dart code into runnable JavaScript code. It is necessary because few web browsers do not support the Dart VM.

Use the following command in the terminal to compile the Dart code into JavaScript code.

dart2js - - out = <output\_file>.js  <dart\_script>.dart

The above command will create a file that contains the JavaScript code corresponding to the Dart code.

3.3 Understanding DART syntax:

3.3.1 Identifiers, Datatypes, variables, comments

3.3.2 Decision making (if, if..else, if..else if..., switch..case)

3.3.3 Iterative statements (for, for...in loop, while, do..while)

3.3.4 break, continue, label

3.4 DART function :

3.4.1 Calling function, deleting function

3.4.2 Passing arguments to function, lexical scoping