# Session 1

## Introduction to Dart

# Session Overview

- Define Dart and explain the history of Dart
- Describe the features of Dart
- List the advantages of Dart
- Explain the installation of Dart Software Development Kit in Android Studio
- Identify the basic syntax in Dart
- Explain identifiers in Dart
- List the basic keywords in Dart
- Elaborate the process of creating a simple Dart application and executing it

# What is Dart? [1-2]

Dart is an open-source programming language developed by Google.

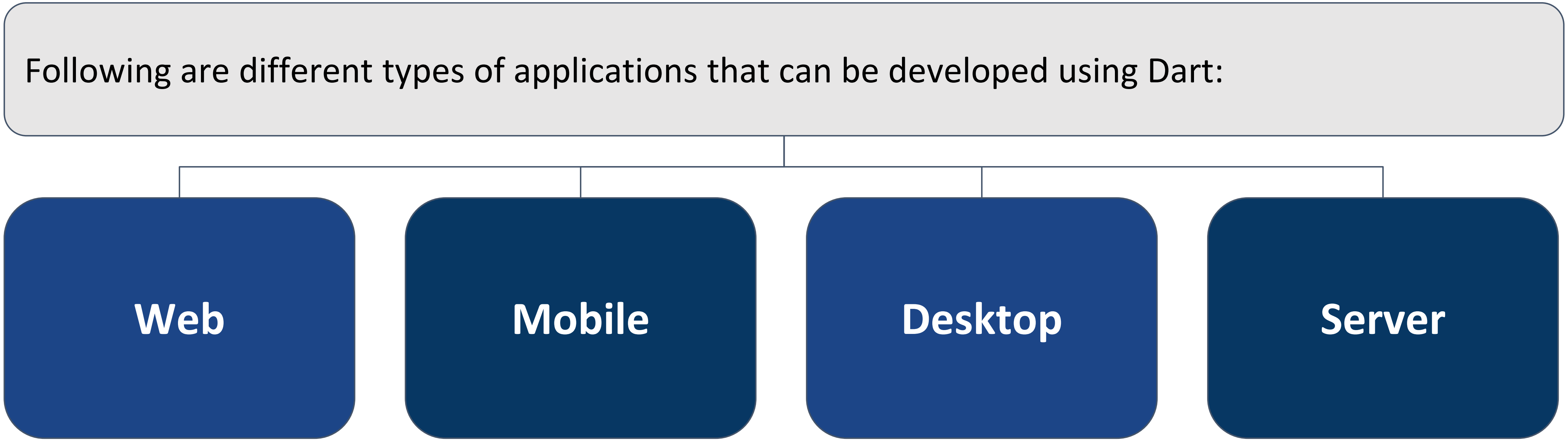
Dart SDK has the Dart Virtual Machine compiler and a utility called dart2js.

Dart is an object-oriented programming language. Thus, it has support for classes, polymorphism, interfaces, and abstraction.

Dart follows a C-style syntax. It is a multi-purpose language.

# What is Dart? [2-2]

Dart is also a cross-platform or platform-independent programming language. This means that it can be run on several operating systems such as Windows, Linux, and MacOS.



Following are different types of applications that can be developed using Dart:

Web

Mobile

Desktop

Server

# History of Dart

### Started by Lars Bak

The Dart project was started by a Danish programmer who was popular for his work on virtual machines.

### First edition Release

The first edition of Dart was released in July 2014, followed by the second edition in the next six months.

### Evolution of Dart Compilers

dartc was the initial compiler. Later, it was Frog and finally it was dart2js.

### The Success

dart2js was a success and still continues to be used in Dart for applications with optimal performance that can be built for multiple platforms.

# Features of Dart

- Open Source
- Platform Independent
- Object Oriented
- Type Safe
- Easy to Learn
- Extensive Libraries
- Browser Support
- Concurrency
- Community
- Flexible Compilation

# Usage and Benefits of Dart

- Universal code that works for several platforms such as Android, iOS, and Web browsers and does not require any change.
- Detecting errors statically and code optimization.
- Provides null safety, which results in lesser instances of app crash.
- Dart is backed by Google.

# Steps to Install Dart SDK in Android Studio

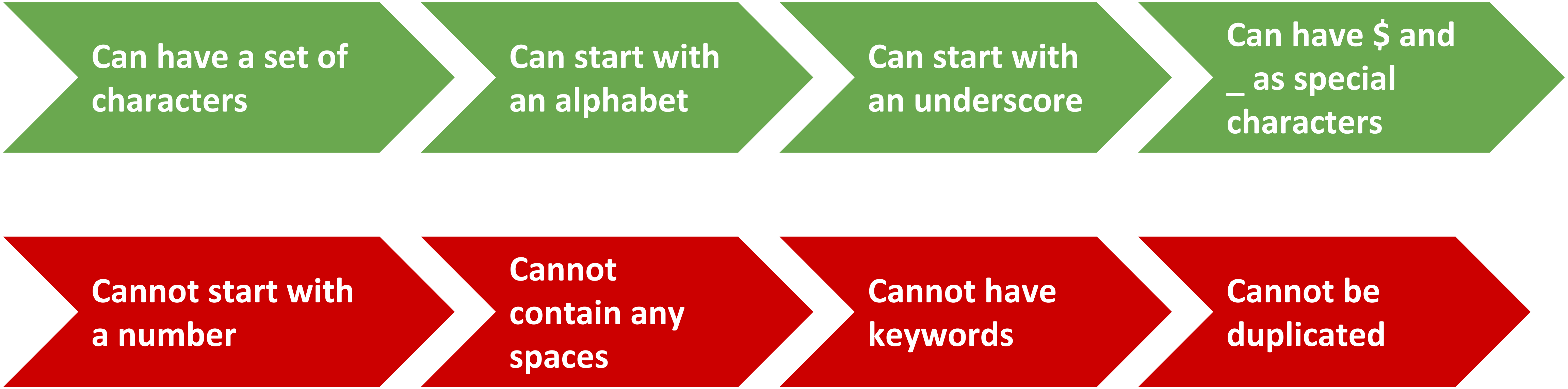
# Understanding Basic Dart Syntax

- The basic syntax in a Dart program comprises several elements such as keywords, data types, variables, constants, string literals, and symbols.
- The `main()` method acts as an entry point to a Dart application.
- A variable `x` is declared and assigned the data type `int`.
- Another variable `y` is also declared and assigned as `string`.

```
void main(){
  int x = 5;
  string y = “Daniel”;
}
```

# Identifiers in Dart

Rules for naming an identifier are as follows:



Can have a set of characters

Can start with an alphabet

Can start with an underscore

Can have $ and _ as special characters

Cannot start with a number

Cannot contain any spaces

Cannot have keywords

Cannot be duplicated

# Example of Identifiers in Dart

| Valid Identifiers | Explanation | Invalid Identifiers | Explanation |
|---|---|---|---|
| `firstName` | Begins with lowercase | `first Name` | Space not allowed |
| `first_name` | Has underscore | `first-name` | Hyphen not allowed |
| `sum1` | Begins with lowercase | `Var` | Begins with uppercase |
| `_name` | Begins with underscore (private variable) | `1name` | Begins with a number |

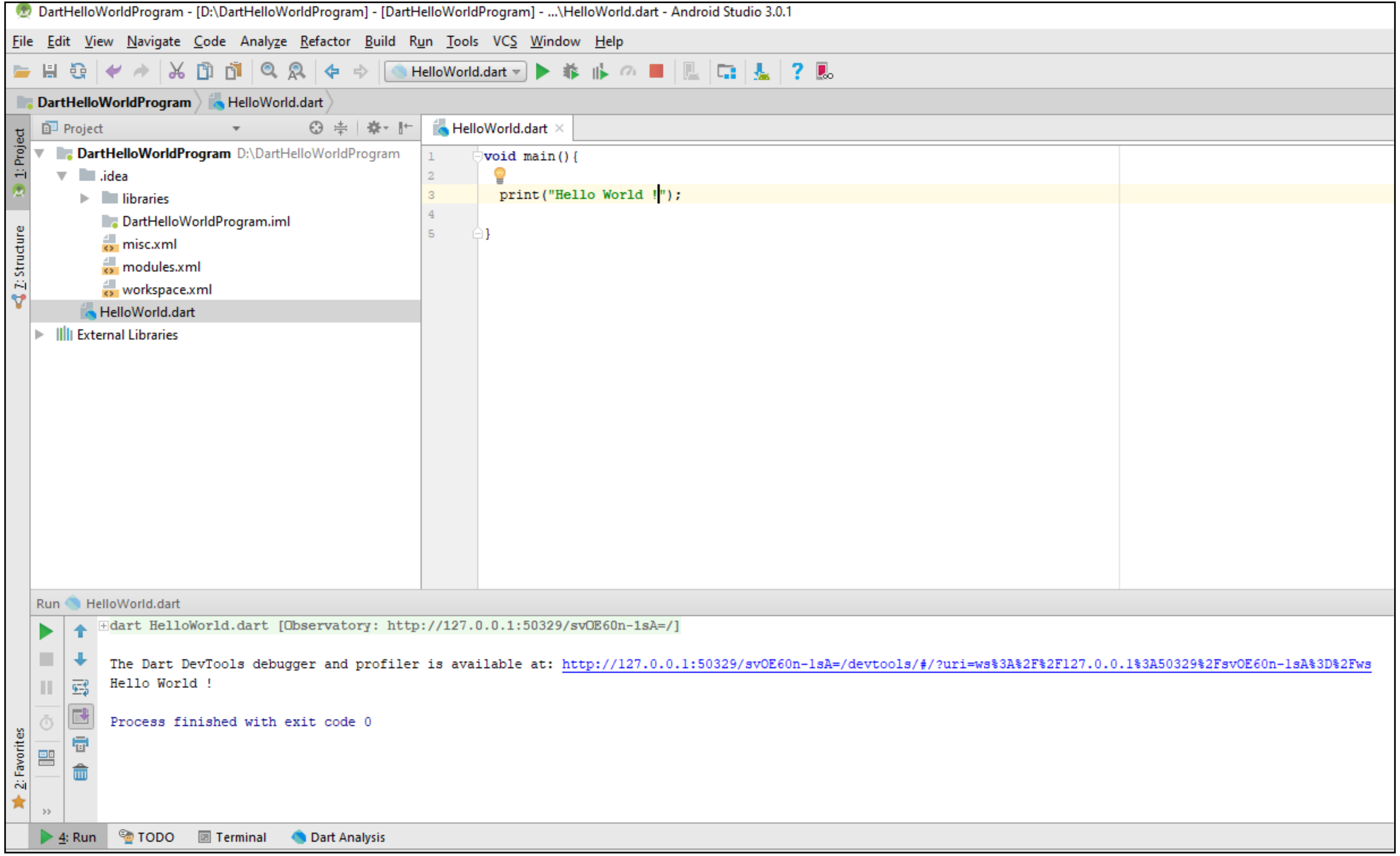
# Keywords in Dart

- Keywords in Dart are reserved words that the compiler interprets in a special manner.
- Each keyword has a unique connotation and purpose in the Dart programming language.
- Keywords are case-sensitive and cannot be used for naming variables, functions, and classes.

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| abstract | import | else | super | in | as | enum | switch |
| assert | export | interface | sync | async | extends | await | this |
| throw | library | break | continue | try | catch | final | class |

# Steps to Create a Simple Hello World Dart application

# Output of the `Hello World` Dart Application



DartHelloWorldProgram - [D:\DartHelloWorldProgram] - [DartHelloWorldProgram] - ...\HelloWorld.dart - Android Studio 3.0.1

```dart
void main(){

  print("Hello World !");

}
```

```
dart HelloWorld.dart [Observatory: http://127.0.0.1:50329/svOE60n-1sA=/]

The Dart DevTools debugger and profiler is available at: http://127.0.0.1:50329/svOE60n-1sA=/devtools/#/?uri=ws%3A%2F%2F127.0.0.1%3A50329%2FsvOE60n-1sA%3D%2Fws
Hello World !

Process finished with exit code 0
```

# Summary

- Dart is a client-centric open source language for developing applications at a faster pace across several platforms.
- The goal of Dart is to be the most fruitful programming language for development across multiple platforms.
- Dart is designed to cater to client development while prioritizing sub-second stateful hot reloads as well as high-quality production across a wide variety of compilation targets (Web, mobile, and desktop).
- Dart provides the language and runtimes that fuel Flutter applications.
- Dart supports all modern browsers such as Chrome, Mozilla Firefox, and Internet Explorer for Web applications.
- Dart program supports and backs multithreading by making use of isolates.