

Mobile Application Development with Flutter

Lecture 5: Dart Basics, Variables, Operators

Prerequisites

- I assume the students have following background knowledge:
- Already completed “Programming Fundamentals” course
 - Variables, operators, expressions, control flow, loops, functions
- Already completed “Object-Oriented Programming” course
 - Objects, classes, inheritance, polymorphism, overloading

Introduction To Dart

- Dart is client-optimized, object-oriented, modern programming language to build apps fast for many platforms like android, iOS, web, desktop etc.
- Client optimized means optimized for crafting a beautiful user interface and high-quality experiences
- Google developed Dart as a programming language
- According to Github, Dart is **one of the most loved programming languages in the world**
- If you know languages like C, Java, C#, Javascript, etc. Dart will be easy for you

Dart Features

- Free and open source
- Object-oriented programming language
- Used to develop android, iOS, web, and desktop apps fast
- Can compile to either native code or javascript
- Offers modern programming features like null safety and asynchronous programming
- You can even use Dart for servers and backend

Difference Between Dart & Flutter

- **Dart** is a client optimized, object-oriented programming language. It is popular nowadays because of flutter. It is difficult to build complete apps only using Dart because you have to manage many things yourself
- **Flutter** is a framework that uses dart programming language. With the help of flutter, you can build apps for android, iOS, web, desktop, etc. The framework contains ready-made tools to make apps faster

Dart History

- Google developed Dart in 2011 as an alternative to javascript
- Dart 1.0 was released on November 14, 2013
- Dart 2.0 was released in August 2018
- Dart 3.0 was released in May 2023
- Dart gained popularity in recent days because of flutter

Why Flutter Uses Dart?

- ① Fast compilation
- ② Just-In-Time (JIT) for development
- ③ Ahead-Of-Time (AOT) for production
- ④ Hot Reload support
- ⑤ Easy to learn (C-style syntax)

1 Fast compilation

- Compilation = converting source code → machine code
- When we modify a few lines, Dart does not recompile the entire app – only the affected parts
- Mobile apps are UI-heavy. Developers constantly change:
 - Colors, Layout, Text, Button Logic
- Dart is optimized for developer productivity – not just runtime performance

2 Just-In-Time (JIT) for Development

- JIT compiles code while the program is running
- Instead of compiling everything first, it compiles functions as needed
- Dart VM handles this dynamically
- When you change code:
 - Dart recompiles only modified methods
 - Injects new code into running app
 - UI updates instantly
- Flutter uses JIT during development to support Hot Reload

3 Ahead-Of-Time (AOT) for Production

- AOT compiles the entire Dart program into native machine code before execution
- When building release app, Dart compiler:
 - Removes unused code
 - Optimizes functions
 - Converts everything into ARM instructions
- Result:
 - Faster startup
 - Lower memory usage
 - Better performance
- Apple doesn't support JIT, hence Flutter use AOT for iOS apps

3 Ahead-Of-Time (AOT) for Production

- Real execution flow

- In development:

Dart Code → Dart VM → JIT → Emulator

- In production:

Dart Code → AOT Compiler → ARM Machine Code → CPU executes directly

4

Hot Reload Support

- Hot Reload injects updated Dart code into the running Dart VM without restarting the app
- Result:
 - Preserves app state
 - Rebuilds widget tree
 - Re-renders UI
- Hot Reload is one of the main reasons Flutter development is so productive

Comparison Table

Feature	JIT	AOT
Compilation Time	Runtime	Before execution
Used In	Development	Production
Hot Reload	<input checked="" type="checkbox"/> Yes	<input type="checkbox"/> No
Performance	Moderate	High
Startup Time	Slower	Faster
Optimization	Limited	High

5 Easy to Learn (C-Style Syntax)

- Most students already know C++, Java, JavaScript
- So, Dart seems familiar
- Familiar Concepts:
 - Variables, Loops, Classes, Functions, OOP Principles
- Modern Concepts:
 - Null safety, Optional types, Async/wait

How to Create Dart File?

- We can start Dart file through one of the following:
 1. dartpad.dev
 2. VS Code
 3. Android Studio

Demonstrate running of basic program

Variables

- A variable stores data in memory
- Rules for creating variable names
- Common Data Types
 - int, double, num, String, bool, var, dynamic
- Naming conventions
 - lowerCamelCase (num1, firstName, isMarried)
- Strings
 - Single line, multiline, special character, raw string
- Optionally Typed Language
 - Statically-typed, dynamically-typed

Constant

- A constant is a value that **cannot be changed** after it is assigned
- Mutable & Immutable
- But in Dart, we have **two types of constants**: final & const

① final (Runtime Constant)

- Value can be determined at **runtime**

```
final currentTime = DateTime.now();
```

② const (Compile-Time Constant)

- The compiler must know the value **before the program runs**

```
const pi = 3.14;
```

Type Conversion & Comments

- Type Conversion
 - Converting String to Int
 - Converting String to Double
 - Converting Int to String
 - Converting Double to Int
- Comments
 - Single-Line Comment
 - Multi-Line Comment
 - Documentation Comment

Operators

- Arithmetic Operators

+, -, *, /, %, ~/

- Relational Operators

==, !=, >, <, >=, <=

- Logical Operators

&&, ||, !

Prepared by Ehtisham Rasheed

Operators

- Assignment Operators

=, +=, -=, *=, /=

- Increment & Decrement Operators

++ (Prefix & Postfix), -- (Prefix & Postfix)

- Type Test Operators

is, is!