Learning Basic Dart Code

# Introduction to Dart

Dart is a programming language designed for building mobile, desktop, server, and web applications. It is the language used in Flutter for developing cross-platform mobile apps. Dart is easy to learn, fast, and supports both object-oriented and functional programming styles.

# Setting up Dart Environment

To get started with Dart, you need to install it on your computer. You can use the official Dart SDK or install it as part of Flutter for mobile development. Follow these steps to set it up:  
  
- Download Dart SDK from the official website.  
- Use an IDE such as Visual Studio Code with Dart and Flutter plugins.  
- Run Dart commands using the terminal/command prompt.

# Basic Dart Syntax

In Dart, you define variables and functions with simple syntax. Below are some basics:  
- Variables and Data Types: Dart supports various data types such as int, double, String, and bool.  
- Functions: Functions are defined using 'void' or a specific return type.  
- Conditional Statements (if/else): You can perform checks using if, else if, else statements.  
- Loops (for, while): Dart supports for, while, and do-while loops.  
- Lists, Maps, and Sets: These are essential collection types used for storing groups of data.

## Example Dart Code

Here is a simple Dart program that demonstrates variables, functions, and conditionals:  
   
 void main() {  
 var name = 'Dart Learner';  
 var age = 25;  
 print('Hello, my name is $name and I am $age years old.');  
   
 if (age > 18) {  
 print('You are an adult.');  
 } else {  
 print('You are a minor.');  
 }  
 }

# Object-Oriented Programming in Dart

Dart is an object-oriented programming language, meaning it supports classes and objects. Below are key concepts in Dart OOP:  
- Classes and Objects: A class is a blueprint for creating objects. An object is an instance of a class.  
- Constructors: A constructor is used to create instances of a class.  
- Inheritance and Polymorphism: Dart supports the concepts of inheritance and polymorphism.

# Error Handling in Dart

Dart provides error handling mechanisms using try/catch blocks to handle runtime exceptions and errors. Below is an example:  
  
try {  
 var result = 10 ~/ 0; // Division by zero  
} catch (e) {  
 print('Error occurred: $e');  
}