
Flutter Beginner's Guide: Setup, Dart Basics & First App

Subtitle: Learn how to set up Flutter, understand Dart fundamentals, and build your first mobile app.

Website Name: haas.dev

Website Link: <https://dev-roast-app.vercel.app>

Introduction

Flutter is a powerful framework by Google for building **cross-platform mobile apps** using a single codebase. Beginners often struggle with setup and understanding Dart. This guide walks you through installation, core concepts, and building your first Flutter app step by step.

Step 1: Installing Flutter Environment

1. Download Flutter SDK from:
<https://flutter.dev/docs/get-started/install>
2. Extract the SDK and add it to system PATH.
3. Verify installation:

```
flutter doctor
```

4. Install required tools:
 - **Android Studio** (for Android development)
 - **VS Code** (lightweight editor with Flutter extensions)
5. Install Flutter and Dart plugins in VS Code.

Exercise: Run `flutter doctor` and fix all issues shown.

Step 2: Understanding Flutter Architecture

- Flutter uses **Dart programming language**.
- Everything in Flutter is a **widget**.
- UI is built using a **widget tree**.

...the same using a widget tree.

Types of widgets:

- **StatelessWidget** → Static UI (no change)
- **StatefulWidget** → Dynamic UI (changes with state)

Step 3: Dart Basics for Flutter

Key concepts you need:

- Variables:

```
int age = 20;
String name = "Hafsa";
```

- Functions:

```
void greet() {
  print("Hello Flutter");
}
```

- Lists:

```
List<String> items = ["A", "B", "C"];
```

- Conditionals:

```
if (age > 18) {
  print("Adult");
}
```

Exercise: Create a Dart function that prints your name and age.

Step 4: Creating Your First Flutter App

1. Create a new project:

```
flutter create my_app
```

2. Navigate to project folder:

```
cd my_app
```

3. Run the app:

```
flutter run
```

Step 5: Understanding Project Structure

- `lib/main.dart` → Main entry point
- `pubspec.yaml` → Dependencies and assets
- `android/` & `ios/` → Platform-specific code

Step 6: Building Your First UI

Edit `main.dart`:

```
import 'package:flutter/material.dart';

void main() {
  runApp(MyApp());
}

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: Scaffold(
        appBar: AppBar(title: Text("My First App")),
        body: Center(
          child: Text("Hello Flutter"),
        ),
      ),
    );
  }
}
```

Exercise: Change the text to display your name and add a background color.

Step 7: Hot Reload & Development Speed

- Flutter supports **Hot Reload** → instant UI updates.
 - Press `r` in terminal or use IDE button.
-

Key Takeaways

- Flutter uses **Dart + Widgets** to build apps.
 - Setup requires **Flutter SDK + Android Studio/VS Code**.
 - Everything is a **widget**, structured in a tree.
 - Hot reload makes development fast and efficient.
 - You can build your first app in minutes once setup is complete.
-

Start building your Flutter projects and practice consistently to gain confidence.

Visit **haas.dev** for more Flutter guides, tutorials, and complete project resources.

Website Name: haas.dev

Website Link: <https://dev-roast-app.vercel.app>
