What is Flutter? (Big Picture)

1. Introduction

Flutter is an **open-source UI software development kit (SDK)** created by **Google**. It is used to build **natively compiled applications** for **mobile (Android & iOS)**, **web**, and **desktop** from a **single codebase**.

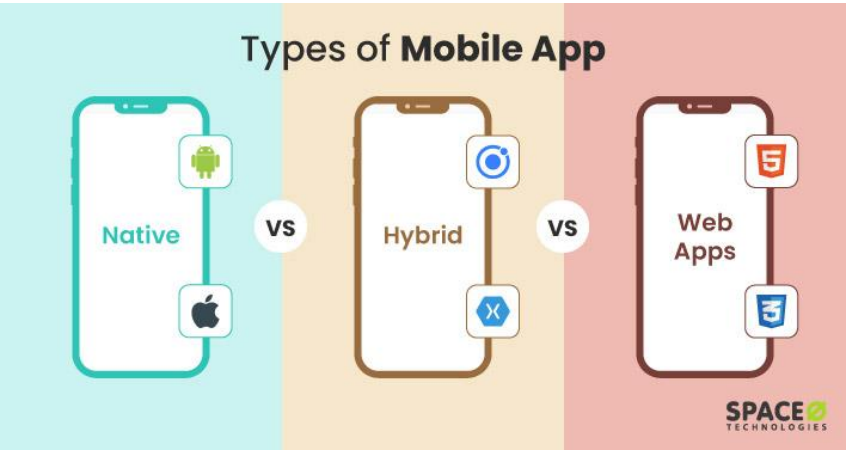
In simple words:

> **Flutter allows developers to write one code and run it on multiple platforms with high performance and beautiful UI.**

---

2. Types of Applications (Before Flutter)

Before learning Flutter, students should understand the **different types of applications** that exist today.



2.1 Native Apps

Native apps are built specifically for a single platform using its official language and tools.

- Android: Java / Kotlin
- iOS: Swift / Objective-C

Characteristics:

- Best performance
- Full access to device features
- Separate codebase for each platform

Examples: WhatsApp, Instagram (native core)

Only use for GLAU

2.2 Web Apps

Web apps run inside a web browser and are built using:

- HTML
- CSS
- JavaScript

They do not need installation from an app store.

Characteristics:

- Runs in browser
- Easy to update
- Limited access to device hardware

Examples: Gmail, Google Docs

2.3 Hybrid Apps

Hybrid apps are web apps wrapped inside a native container.

They use:

- HTML
- CSS
- JavaScript

And run inside a **WebView**.

Characteristics:

- Single codebase
- Slower performance than native
- UI depends on browser rendering

Examples: Early versions of Ionic / Cordova apps

---

3. Why Flutter Was Created

Before Flutter, developers usually faced these problems:

- Separate codebases for Android and iOS

Only use for GLAU

- High development and maintenance cost
- Inconsistent UI across platforms
- Slower UI performance in hybrid frameworks

Flutter was created to solve these issues by:

- Providing a **single codebase**
- Delivering **near-native performance**
- Allowing **full control over UI design**
- Enabling **fast development with Hot Reload**

---

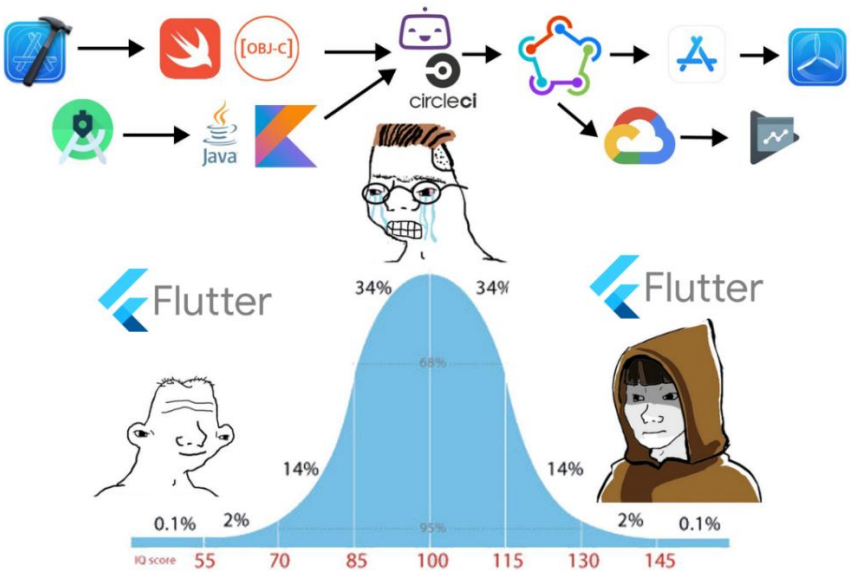
3. What Makes Flutter Different

3.1 Single Codebase

With Flutter, you write your application **once** using Dart, and it works on:

- Android
- iOS
- Web
- Windows
- macOS
- Linux

This saves time, cost, and effort.



Only use for GLAU

3.2 Everything is a Widget

In Flutter, **everything is a widget**:

- Text
- Button
- Image
- Padding
- Screen layout

Widgets are used to describe **how the UI should look**, not how to draw it.

This makes UI building:

- Consistent
- Reusable
- Easy to manage

3.3 Flutter Does NOT Use Native UI Components

Unlike other frameworks, Flutter:

- Does NOT use Android XML widgets
- Does NOT use iOS UIKit widgets

Instead, Flutter draws everything using its **own rendering engine**.

This results in:

- Same UI on all platforms
- No platform-specific UI bugs
- Better performance control

---

4. Dart Language (Flutter's Backbone)

Flutter applications are written in **Dart**.

Why Dart?

- Fast execution
- Supports both **JIT** (development) and **AOT** (production)
- Optimized for UI development
- Easy to learn for beginners

Only use for GLAU

Dart enables features like:

- Hot Reload
- Smooth animations
- Predictable performance

---

5. Flutter Architecture (High-Level View)

Flutter is built using a layered architecture:

1. **Flutter Framework (Dart)**
    - Widgets
    - State management
    - UI logic
    - Gestures
2. **Flutter Engine (C++)**
    - Rendering (Skia)
    - Text layout
    - Image handling
    - Dart runtime
3. **Platform Embedder**
    - Connects Flutter to Android, iOS, Web, Desktop
    - Handles lifecycle, input, and native APIs

This architecture makes Flutter **fast and reliable**.

---

6. Hot Reload (Developer Superpower)

Hot Reload allows developers to:

- See UI changes instantly
- Modify code without restarting the app
- Experiment with UI quickly

This drastically improves:

- Learning speed
- Development productivity

Only use for GLAU

7. What Can You Build with Flutter

Using Flutter, you can build:

- Mobile apps (Android & iOS)
- Web applications
- Desktop applications
- MVPs and startups
- Production-level enterprise apps

Popular app categories:

- E-commerce
- Social media
- News & blogs
- Finance & fintech
- Education apps

---

8. Flutter vs Traditional Development

Feature	Flutter	Native Development
Codebase	Single	Separate
UI Consistency	Same everywhere	Platform dependent
Performance	Near-native	Native
Development Speed	Very fast	Slower
Hot Reload	Yes	No

---

9. Who Should Learn Flutter

Flutter is ideal for:

- Students learning app development
- Beginners entering mobile development
- Web developers moving to mobile
- Startups building fast MVPs
- Companies targeting multiple platforms

---

10. Summary (Big Picture)

- Flutter is a **cross-platform UI framework**
- Uses **one codebase** for many platforms
- Written in **Dart**
- Uses its **own rendering engine**
- Focuses on **performance, UI consistency, and developer productivity**

Flutter is not just a framework — it is a **complete ecosystem for modern app development**.

---

11. What to Learn Next

After understanding what Flutter is, the next topics should be:

1. Dart basics
2. Flutter project structure
3. Widgets (Stateless vs Stateful)
4. Layout system
5. State management basics

This creates a strong foundation for Flutter development.