

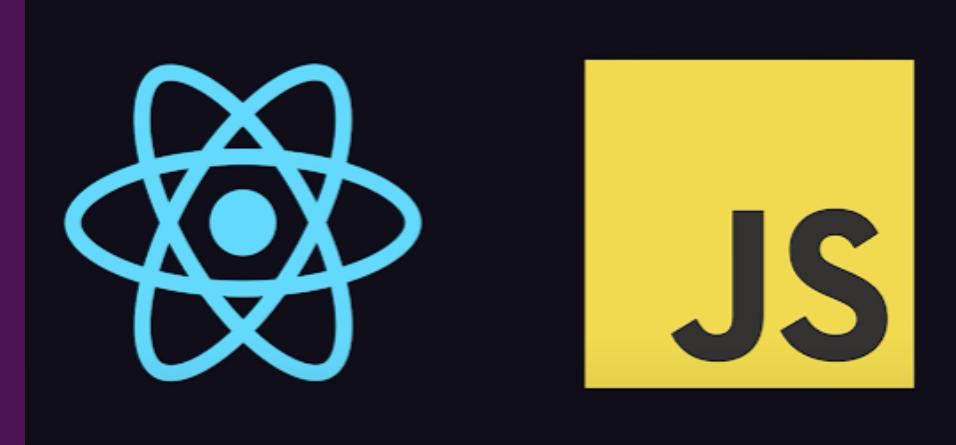
# Flutter Forward

Day 1

# What's in store for you?

- Why Flutter?
- Introduction to Dart
- Features
- Hands-on coding

# Programming Languages:



In react native we code apps in JavaScript with react to being a required UI Library. So it's almost like an extension of the JavaScript language and it also supports typescript.

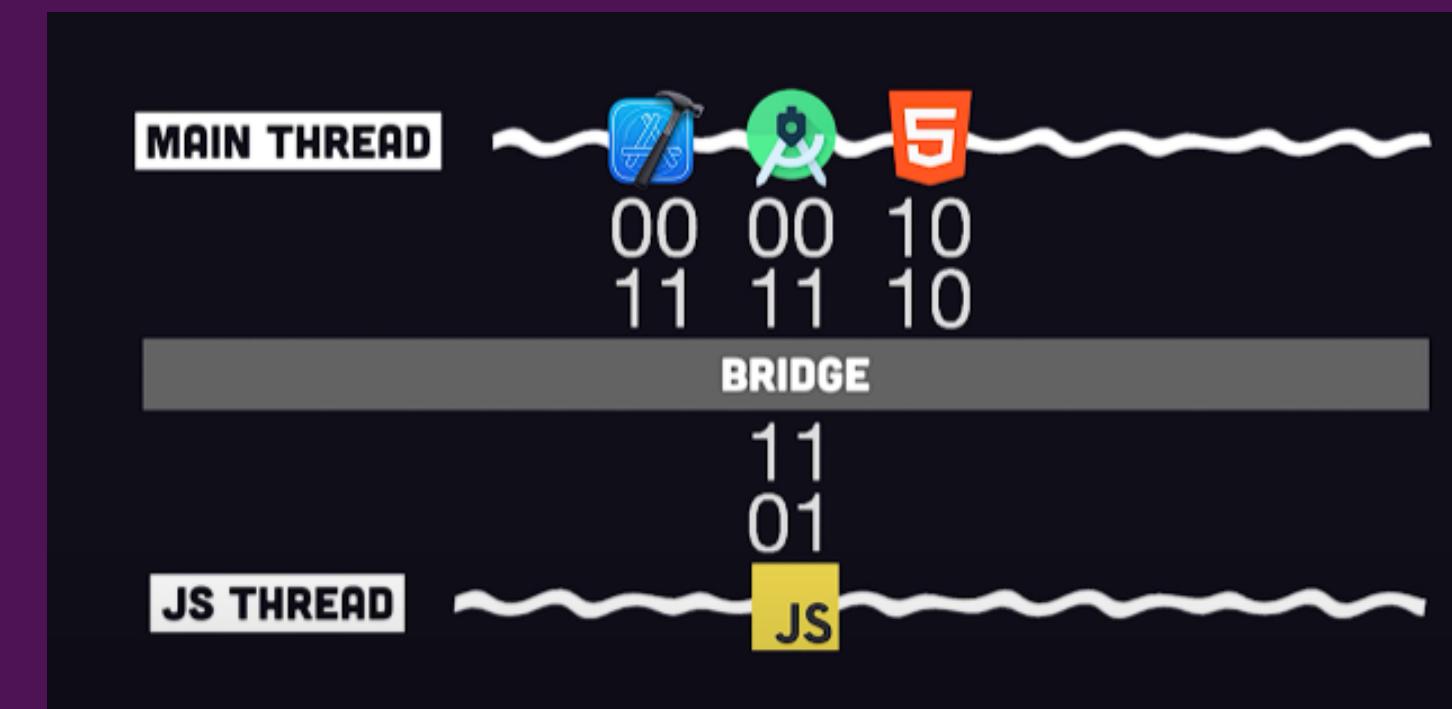


Flutter uses a language called Dart. It has a syntax that resembles java, c sharp, and C++. And it results in various performances advantages.

# Architecture:

## So how does react native work under the hood?

It runs two separate JavaScript threads: one for the main thread to run the app on the native platform, and another for the business logic of the application.



So, when reactive renders a component, it's truly rendering native components on the corresponding platform.

# Architecture:

## **So how does Flutter work under the hood?**

Flutter uses its own rendering engine built with C++ and Skia to render custom pixels to the screen, allowing for pixel-perfect versions of IOS and Android widgets and smooth graphics without needing a bridge.

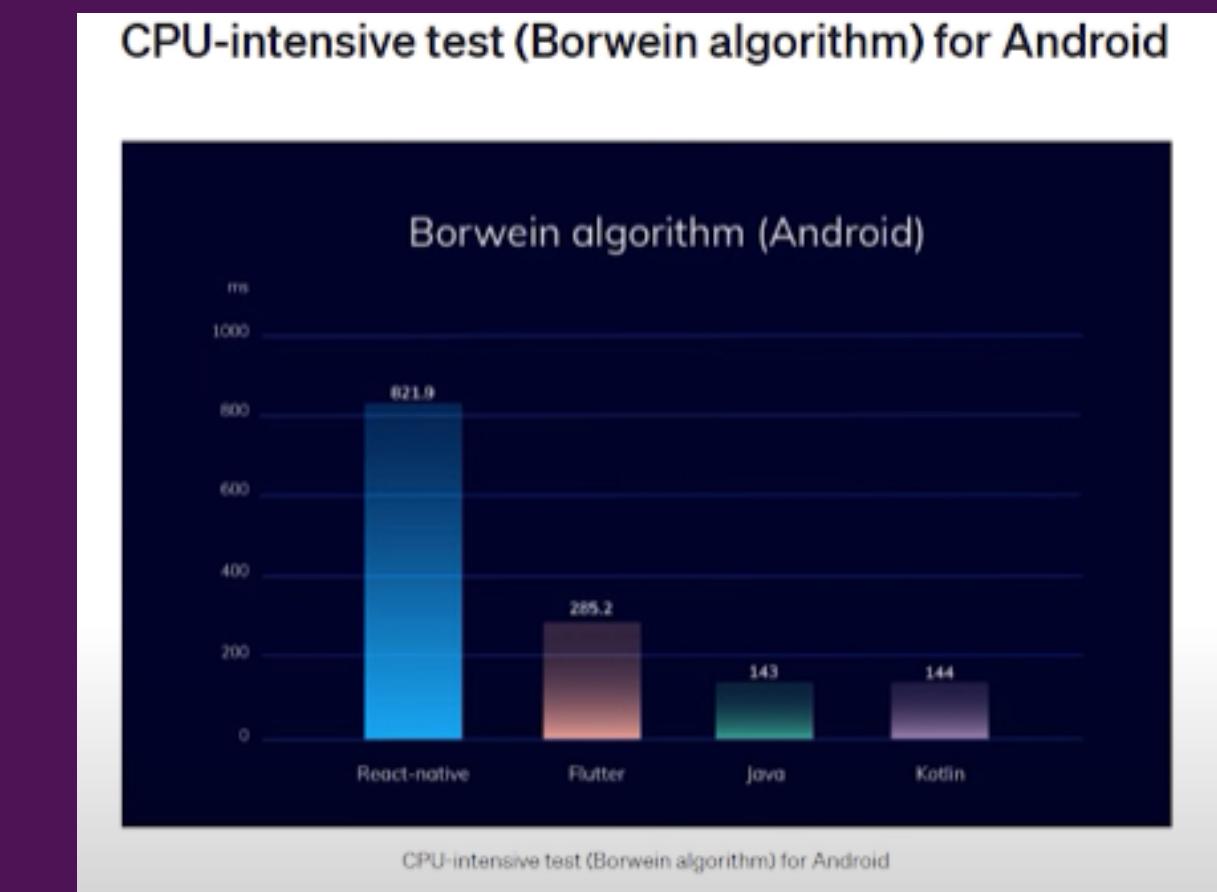
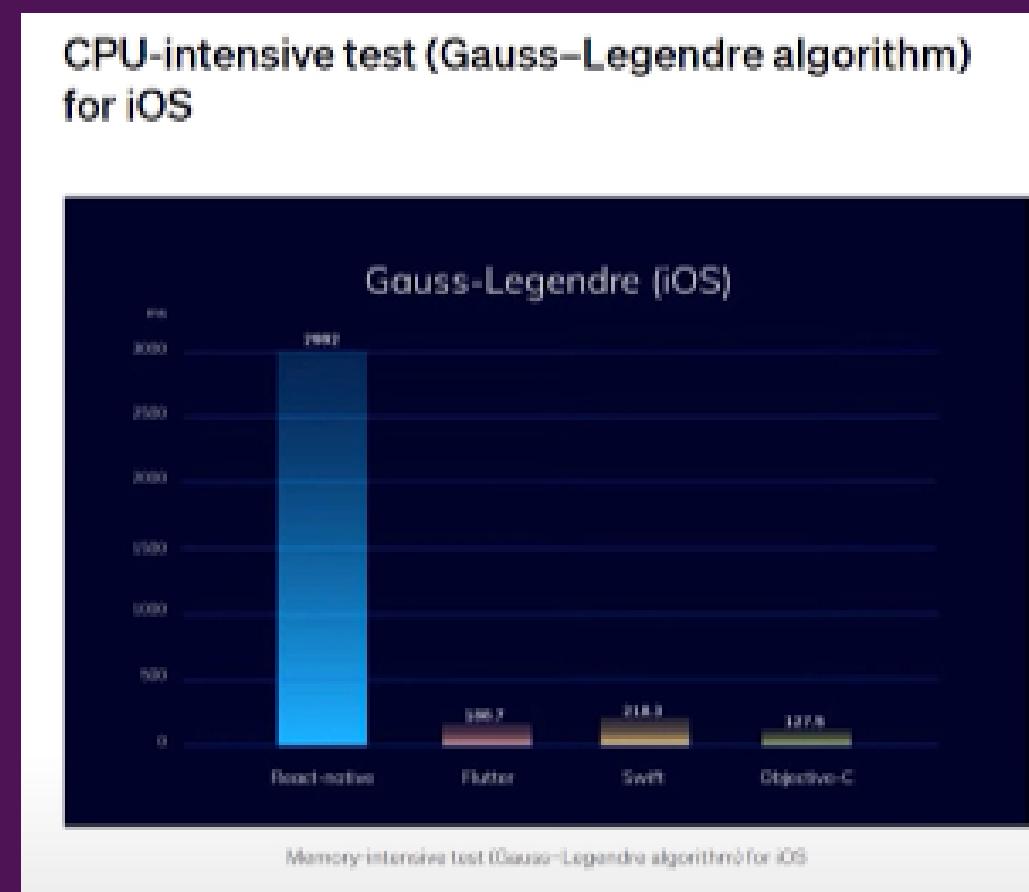


Kind of like a gaming engine like Unity.

# Performance:

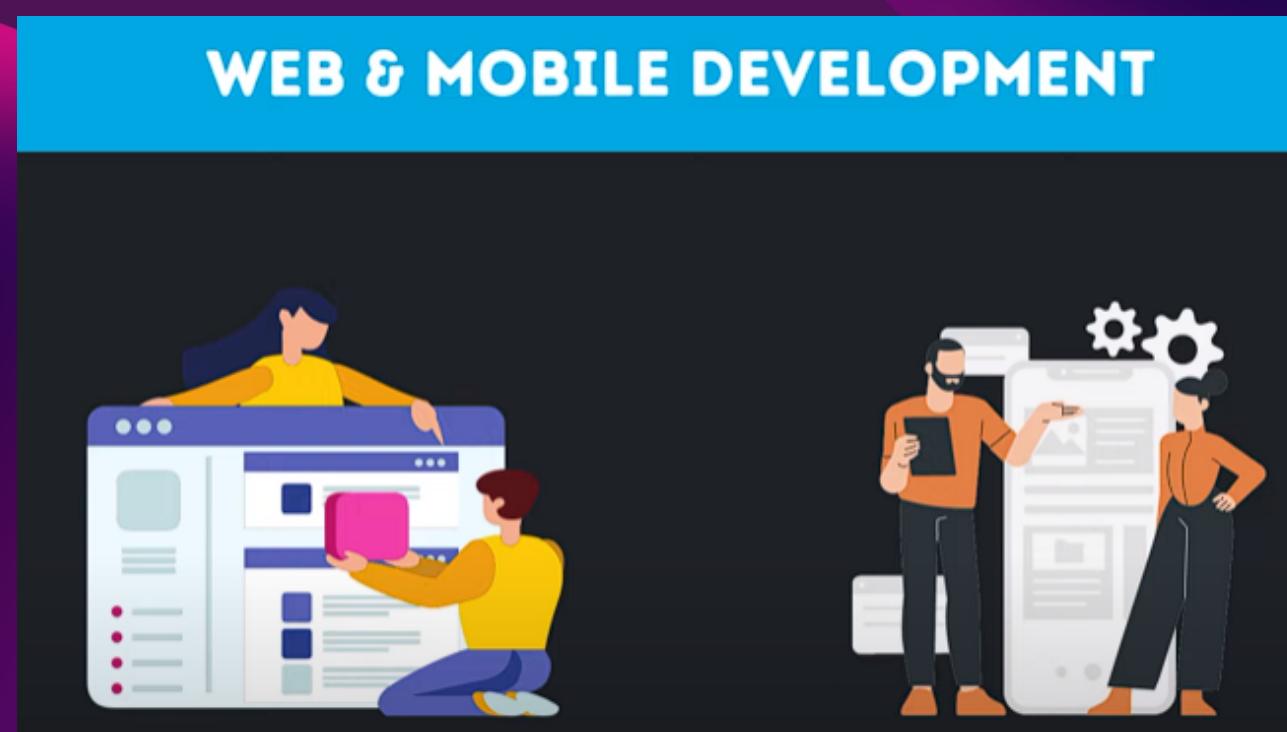
## Which one is faster?

Flutter is faster due to its direct compilation of machine code, which eliminates the need for a JavaScript Bridge. This results in better performance and closer-to-native performance benchmarks.



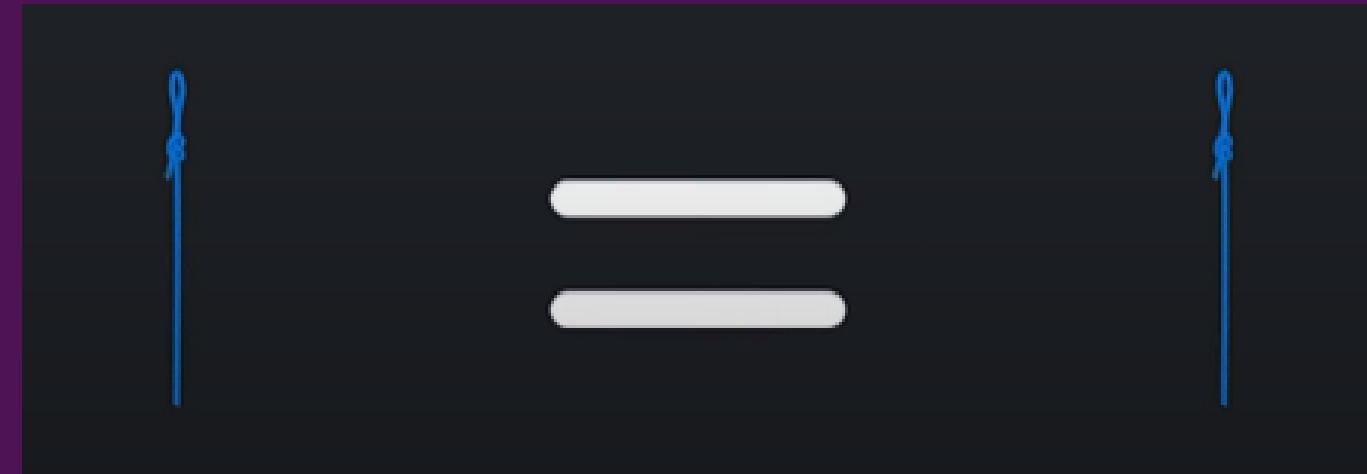
# Introduction to Dart:

Dart is an object-oriented programming language developed by Google in 2011 for web and mobile application development. It powers the Flutter framework, which is also developed by Google and aims to be a multi-platform development space.



# Features:

- Dart is a typesafe language, which means that a variable's value always matches its static type. This helps reduce runtime errors by ensuring that the value of a variable is always of the expected type at runtime.



- Dart's typing system is flexible and allows us to use Dynamic typing to avoid having to specify types every time and makes the process much faster in general

## DYNAMIC TYPING

```
1 // dynamic type
2 void main()
3 {
4   // declaring and assigning value to variable a
5   // of type dynamic
6   dynamic a = 40;
7   print(a);
8   // reassigning value to a
9   a= "Dart";
10 }
```

# Features:

- Dart provides null safety, which ensures that values cannot be null unless explicitly allowed, reducing app crashes and errors. In terms of execution, Dart uses two methods for apps targeting mobile and desktop devices.

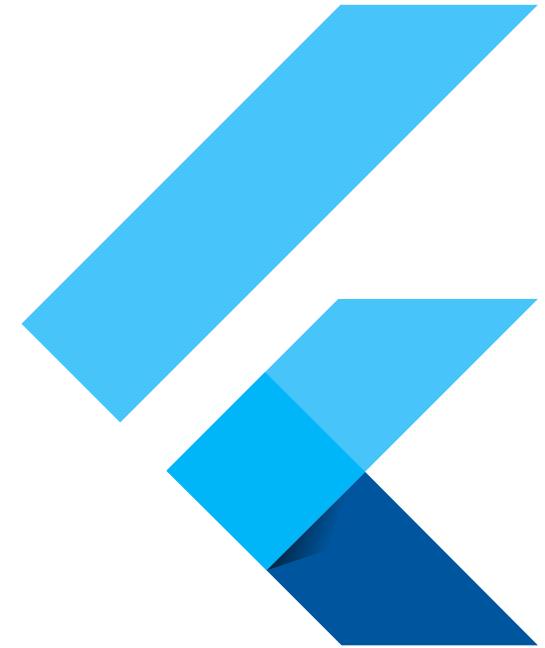


- Dart uses just-in-time or ahead-of-time compilation to turn native code into machine code.



# Hands-on coding!

In this segment, you guys will have the opportunity to apply what you have learned by coding in dart along side. This interactive approach will help you solidify your understanding of Dart and gain practical experience in using the language.



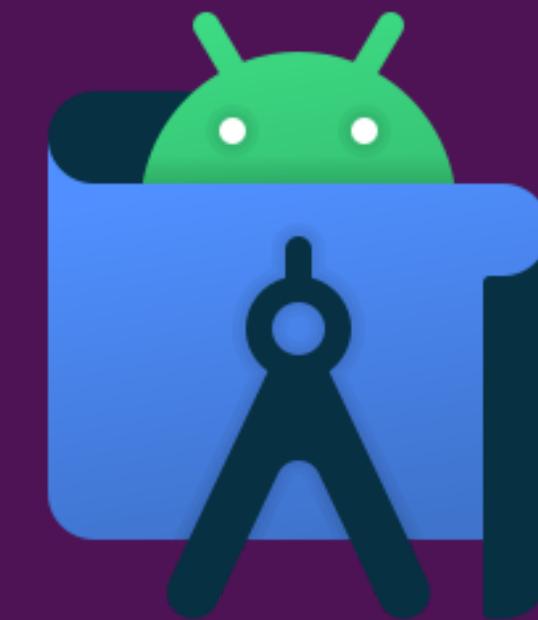
# Flutter Forward

Day 2

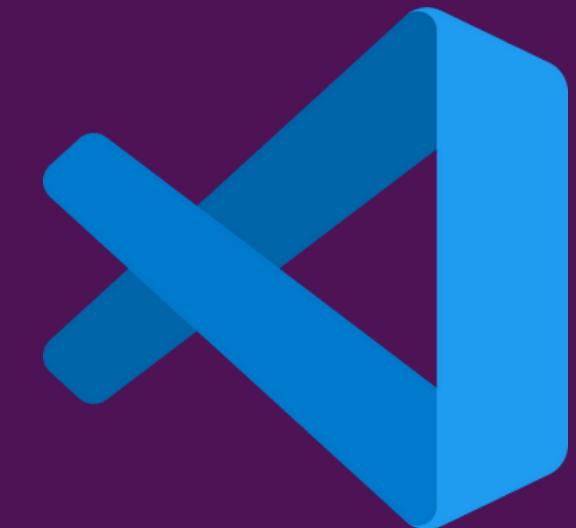
# On Day 2, here's what you can expect

- An overview of required tools
- Flutter Basics
- A guide to building apps

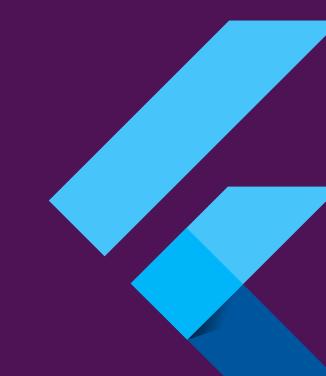
In this session, we used  
the following tools:



Android Studio



VS Code



Flutter and Dart

# Basics:

Flutter is a UI framework that allows developers to build apps for multiple platforms, including iOS, Android, web, and desktop.

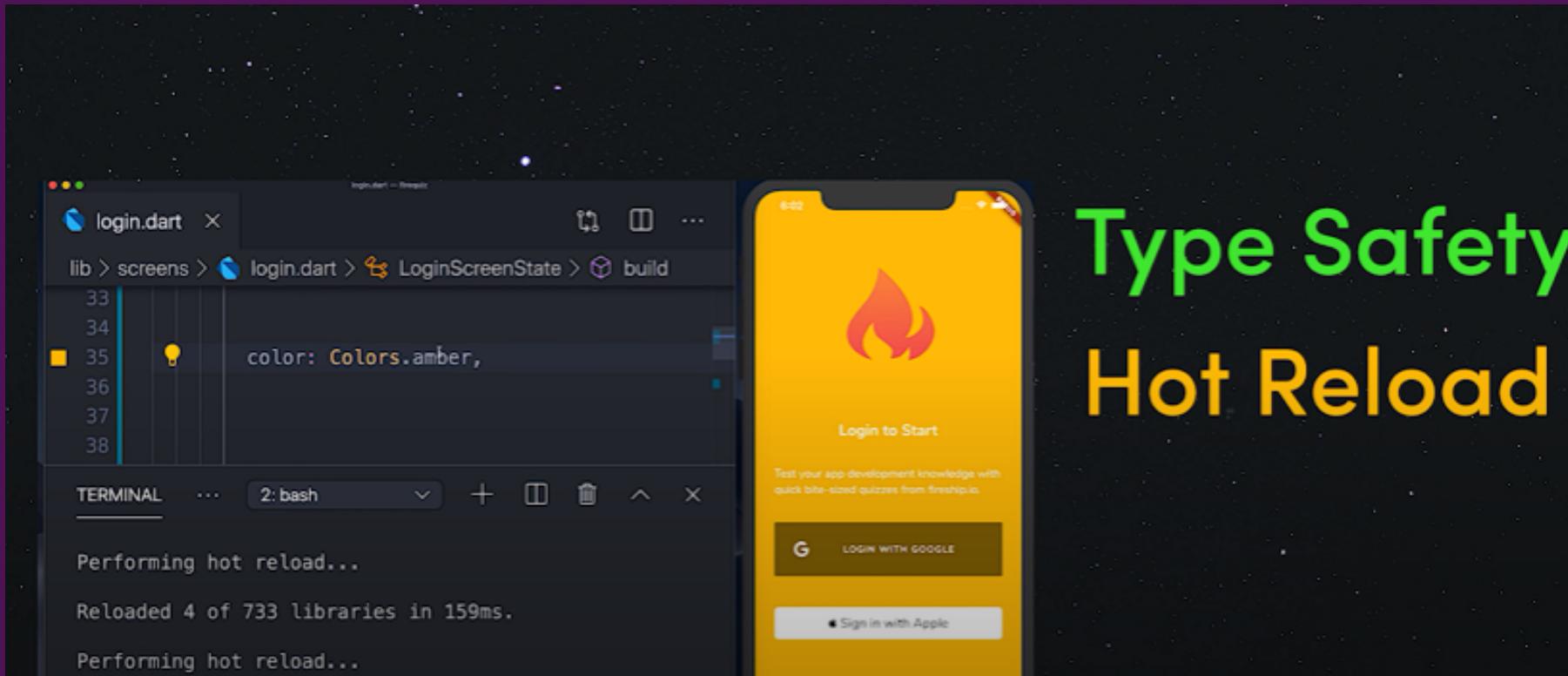


It combines a high-performance Graphics Engine with the Dart programming language.

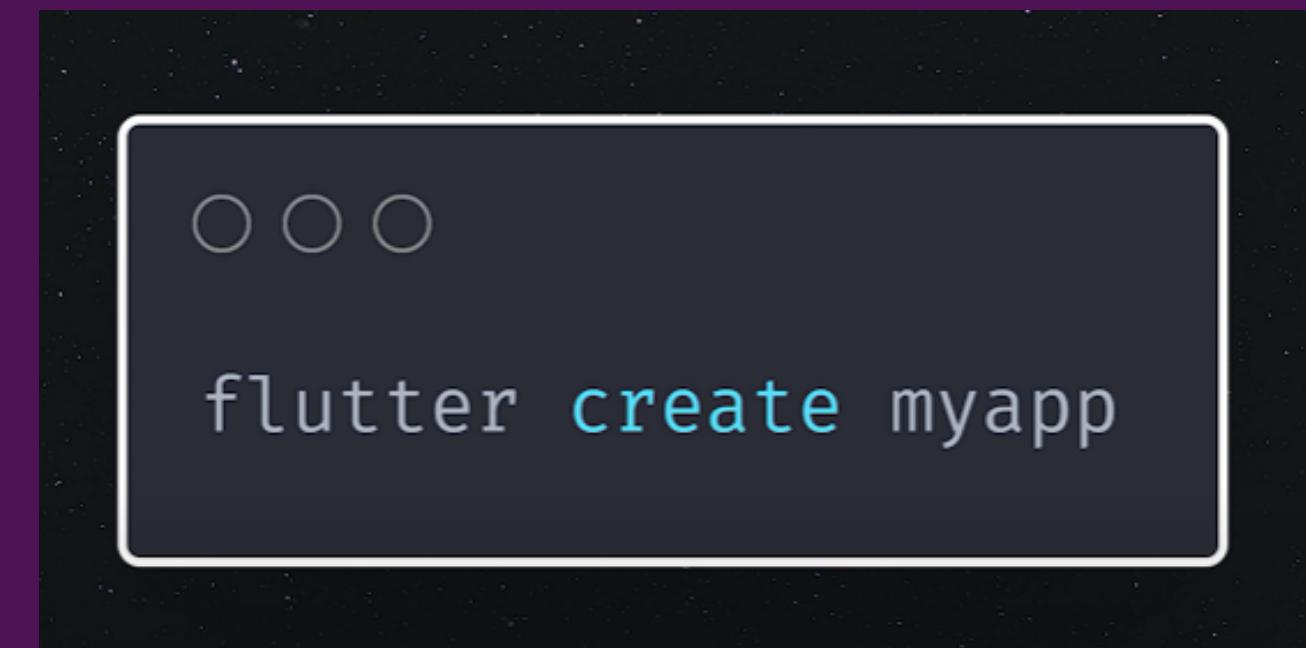


# Basics:

Dart also provides full-type safety and stateful hot reload. Dart compiles Native machine code, ensuring beautiful graphics on any platform.

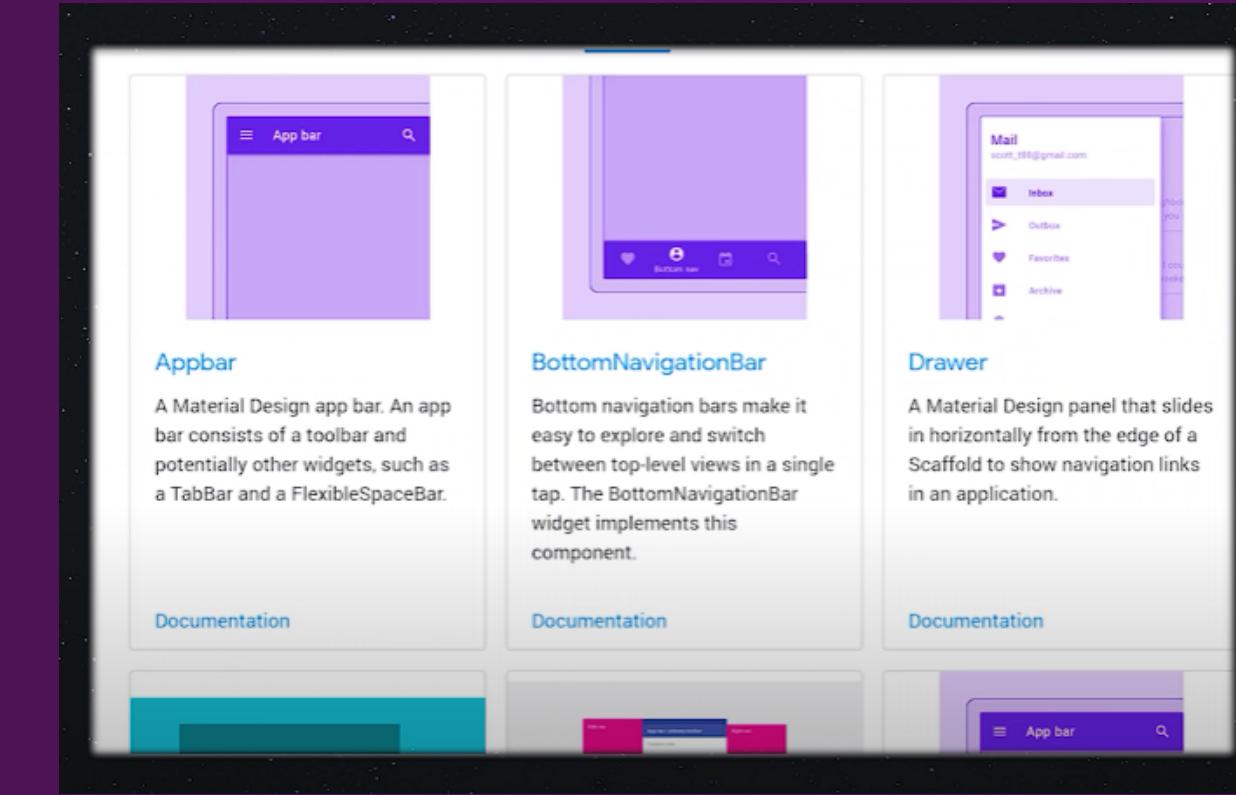


To create an app we use the flutter create app to create a new project

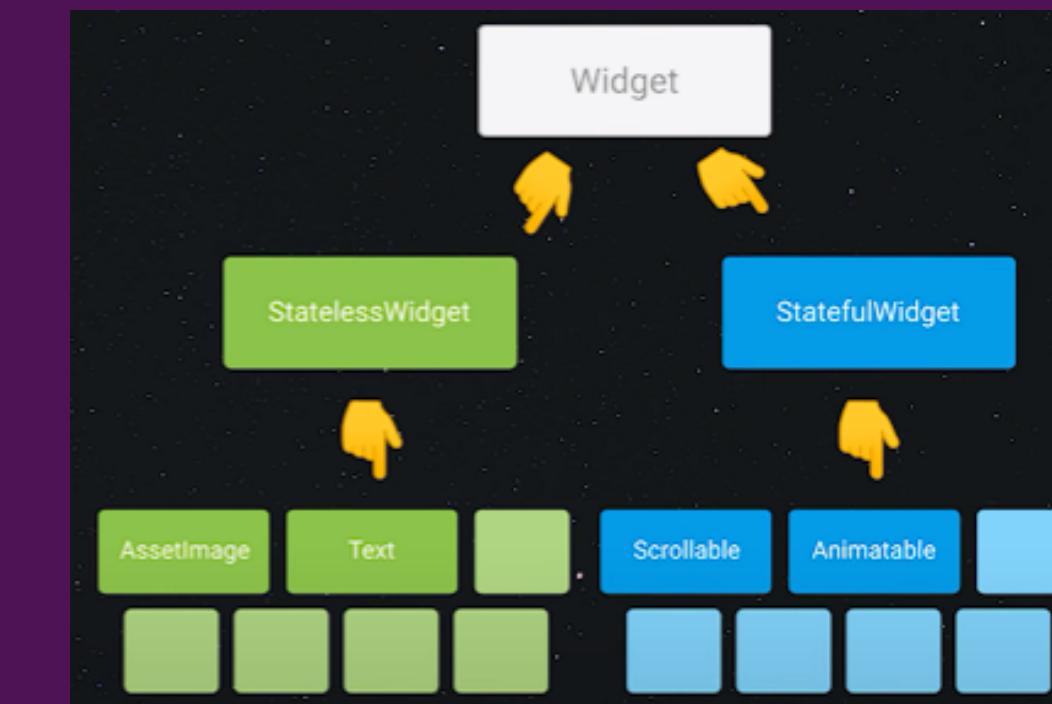


# Basics:

The UI is laid out as a tree of widgets and pre-built widgets to handle animation, scrolling, responsive layout, and more.



Flutter uses a widget-based system, where everything is a widget and widgets return other widgets, allowing for an expressive and declarative UI structure.



# Building an apps!

This segment will present you with a step-by-step guide on how to build your own app from scratch using Dart. This approach will help you solidify your understanding of Flutter and gain practical experience in using this framework.

Thank you all for attending today's event.  
You can practice the topic and if you have any  
doubts, you can contact us and we will do our best  
to assist you. Happy Coding!

-Priyanshu