

Flutter Workshop

By-



Madhavendra P. Singh



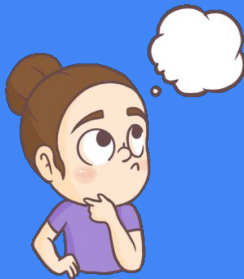
Haresh Nayak



Kshitiz Goel



What is Flutter ?



Flutter is Google's UI toolkit for building beautiful, natively compiled applications for mobile, web, and desktop from a single codebase.



Fast Development

Paint your app to life in milliseconds with Stateful Hot Reload. Use a rich set of fully-customizable widgets to build native interfaces in minutes.



Expressive and Flexible UI

Quickly ship features with a focus on native end-user experiences. Layered architecture allows for full customization, which results in incredibly fast rendering and expressive and flexible designs.



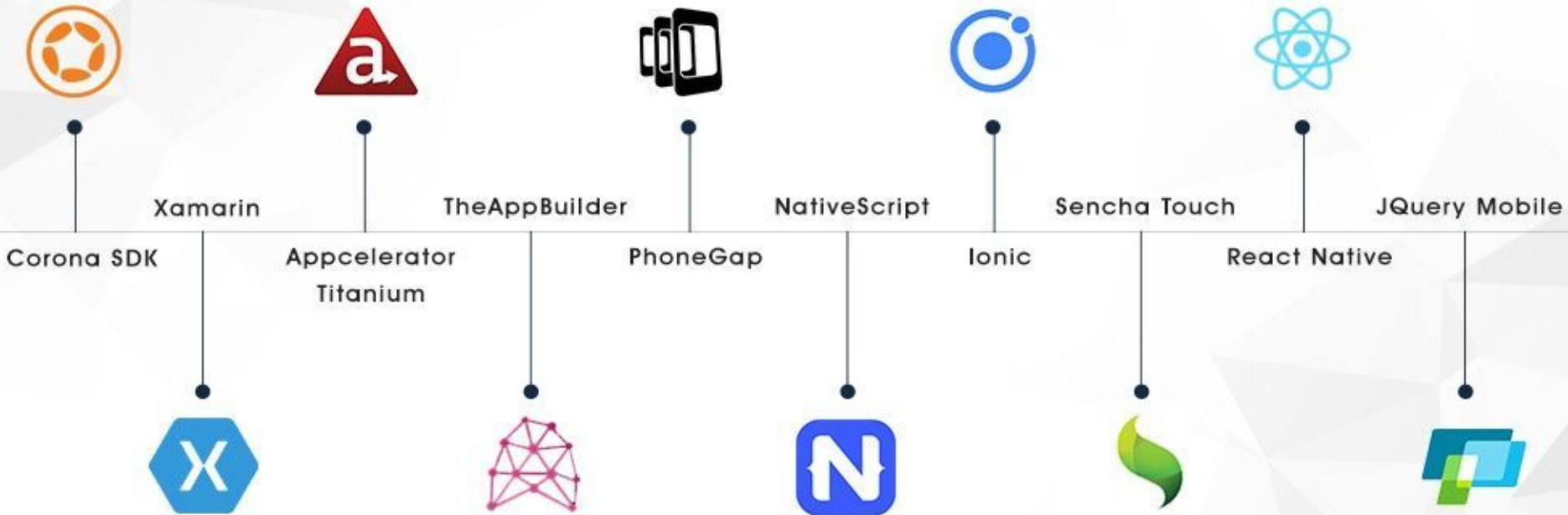
Native Performance

Flutter's widgets incorporate all critical platform differences such as scrolling, navigation, icons and fonts, and your Flutter code is compiled to native ARM machine code using [Dart's native compilers](#).

Other Cross-Platform Framework

ANDROID DEVELOPMENT FRAMEWORKS FOR BUILDING SUPERIOR MOBILE APPLICATIONS

CROSS-PLATFORM FRAMEWORKS



Why use Flutter?



- One code base for both iOS and Android.
- Flutter is the only mobile SDK that provides reactive views without requiring a JavaScript bridge.
- Flutter apps look and feel great.
- Make a change in the app and see them in the blink of an eye. All thanks to Hot-Reload.

Beautiful

Productive

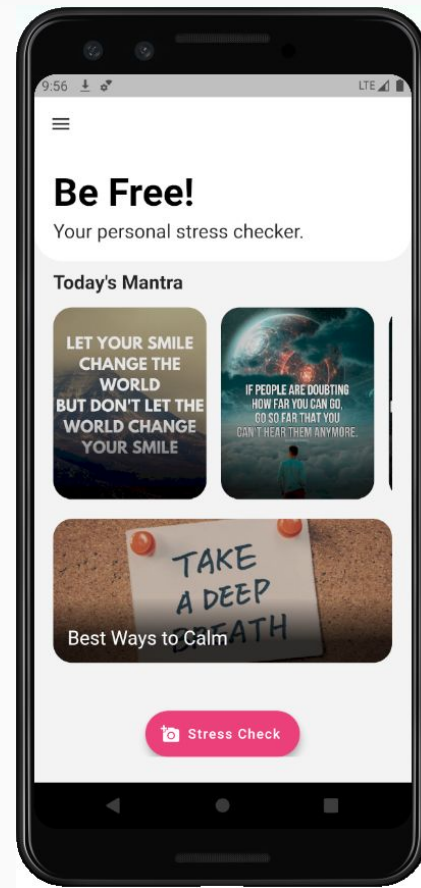
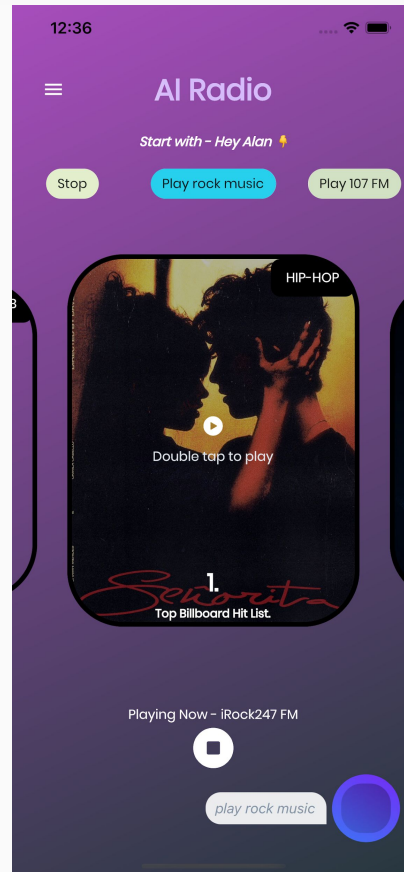
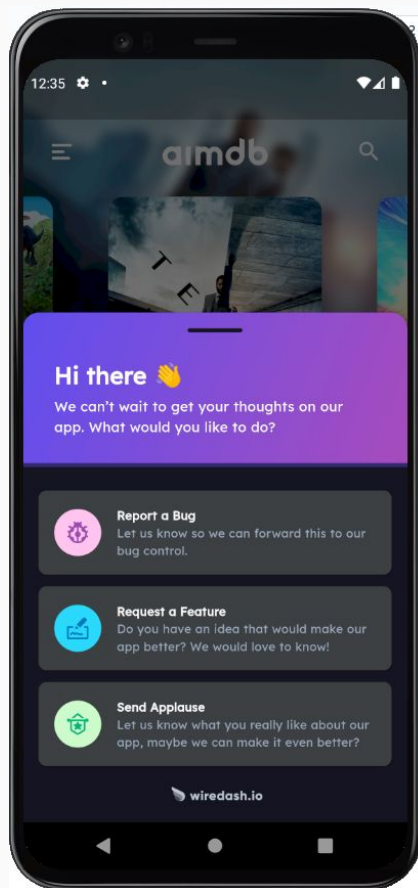
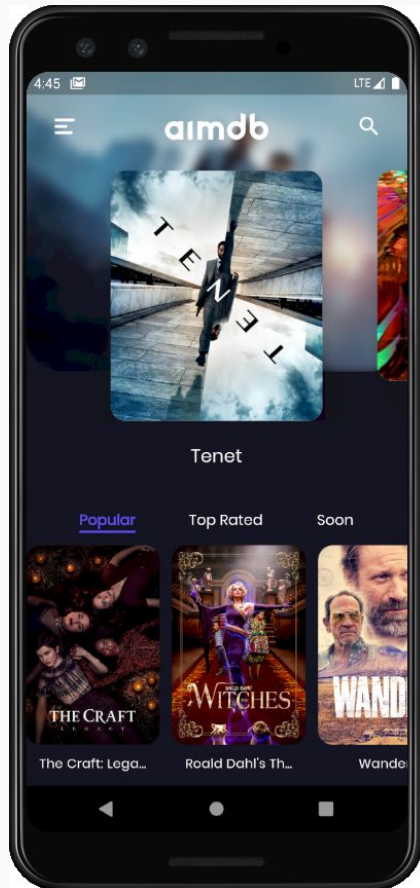


Flutter

Fast

Open

Screenshots of apps:



Apps Built using Flutter



Welcome to the Flutter community

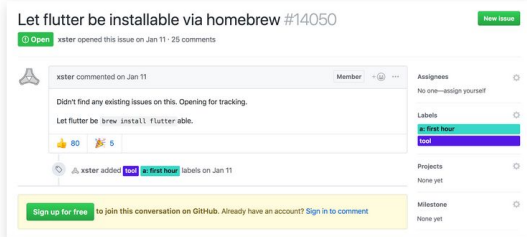
Below you'll find ways to get involved in the Flutter developer community as well as links to resources that can help answer your questions. If you're in China, read [Using Flutter in China](#).

GitHub

Tell us what you want to see

An easy way to send feedback is to "thumbs up" issues important to you on GitHub

[See the issue tracker](#)

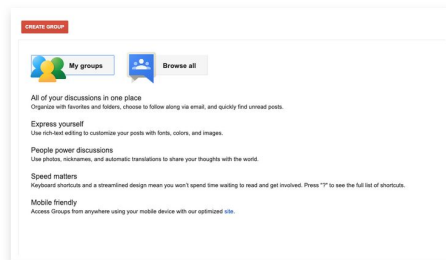


Google Groups

Google Groups is a great way to stay up to date with Flutter.

[Subscribe to breaking changes and other announcements.](#)

[Ask questions and troubleshoot issues with the Google community.](#)



Flutter YouTube

Watch these videos to learn from Google and developers as you build with Flutter.

[Learn more](#)



r/FlutterDev

Join the large community on Reddit with discussions around every aspect of Flutter.

[Learn more](#)



Discord

Talk to other Flutter developers about everything Flutter-related, and share useful resources.

[Learn more](#)



@FlutterDev

Follow the Flutter team in real-time with information on new features, upcoming events, and more.

[Learn more](#)



Medium

Get the latest news and insights from a diverse group of users building with Flutter.

[Learn more](#)



Flutter Community Slack

Connect and work through problems with a group of knowledgeable Flutter people.

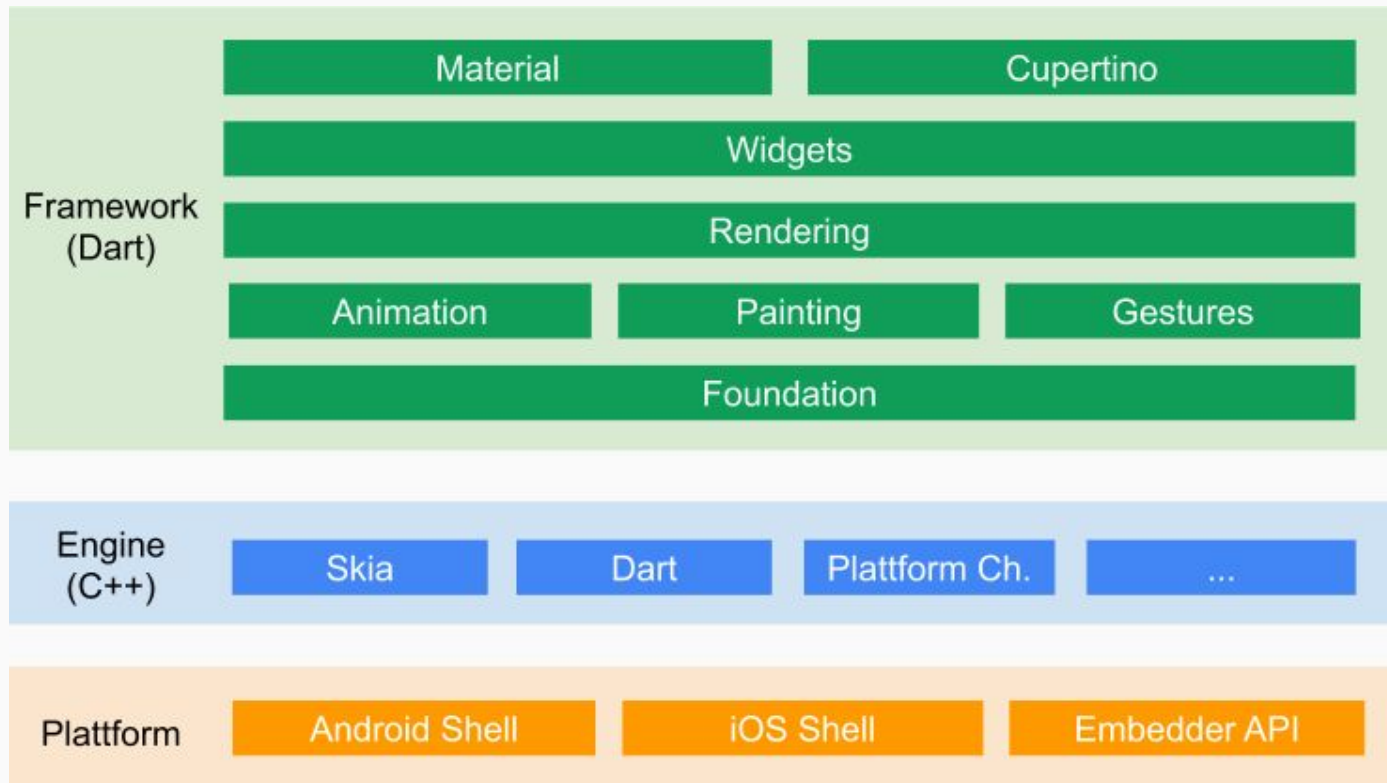
[Learn more](#)

Overview of Flutter



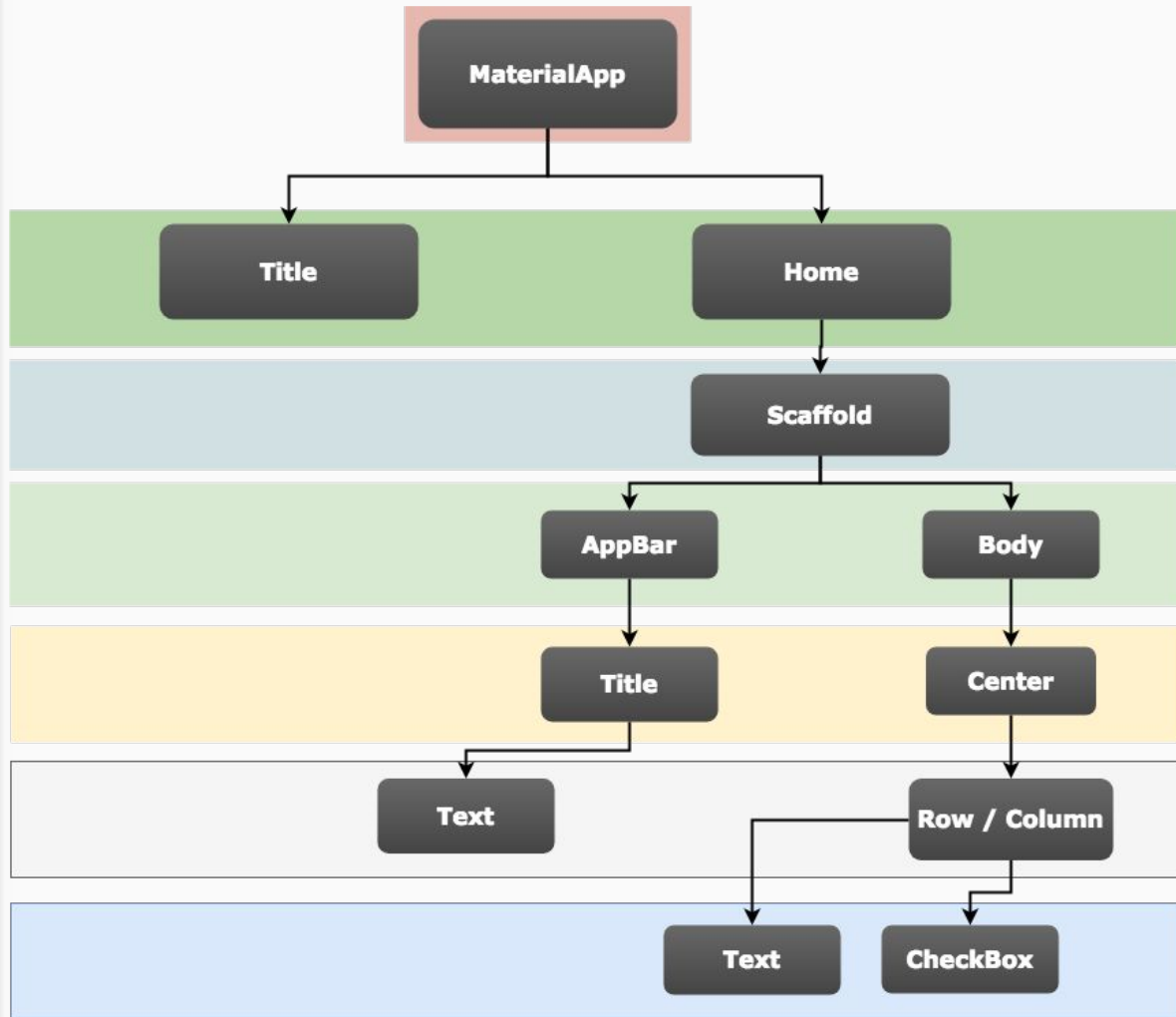
Flutter System Overview

- All boxes highlighted in green are editable.



Widgets

- Each element on a screen of the Flutter app is a widget. The view of the screen completely depends upon the choice and sequence of the widgets used to build the app. And the structure of the code of an app is a tree of widgets.



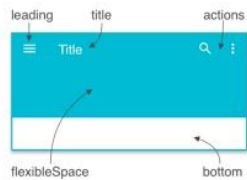
Everything is a widget

Scaffold



UI / UX

AppBar



Text

```
onPanUpdate:  
DragUpdateDetails(Offset(0.3, 0.0))
```

RichText

Flutter.World for **Mobile**

SafeArea

No SafeArea



With SafeArea



Column

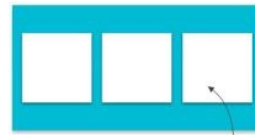
Column



Vertically Aligned

Row

Row



Horizontally Aligned

Container



Button

Barista

Default

Barista

StadiumBorder

Barista

UnderlineInputBorder

Barista

OutlineInputBorder

Stateless v/s Stateful Widgets



Stateful Widget	Stateless Widget
when Widget changes its value, that's Stateful. e.g. Checkbox, Radio button, Textfield	No change in widget value, that's Stateless. e.g. Text, Icon, Icon button, Raised button
Override the createState() and return State.	Override the build() and return Widget.
Use when user want to change UI dynamically.	Use when UI remains constant during runtime.
When Widget's state changes, the State object calls setState(), telling framework to redraw widget.	

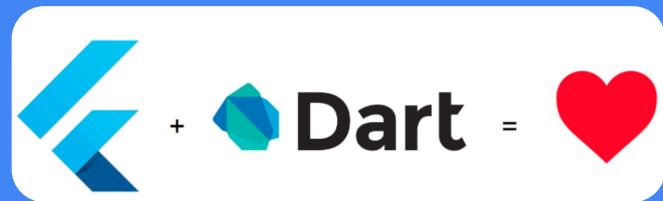
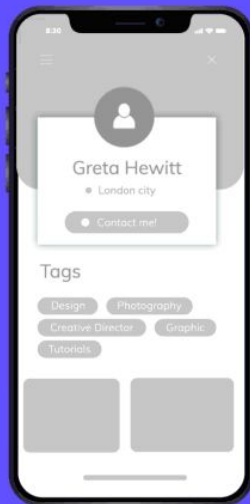


Example: StatelessWidget



Example: StatefulWidget

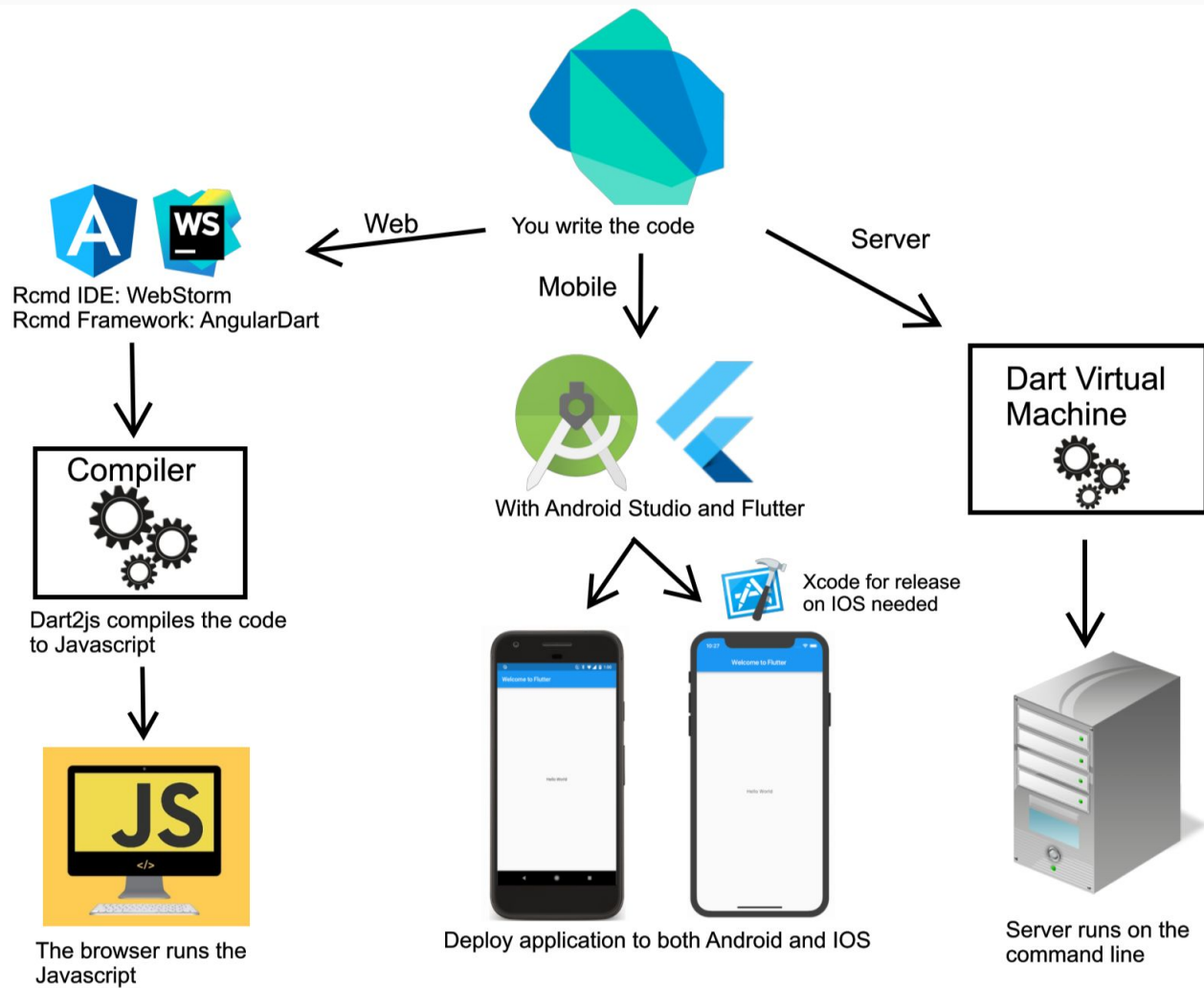
Introduction to Dart-lan





Dart is a client-optimized programming language for apps on multiple platforms. It is developed by Google and is used to build mobile, desktop, server, and web applications. Dart is an object-oriented, class-based, garbage-collected language with C-style syntax. Dart can compile to either native code or JavaScript.





Points to remember

- Everything is an Object, There are no primitive types. Everything inherits from the 'Object' Class.
- Dart is strongly typed, but it can infer types statically. For variables with dynamic type, use the keyword `dynamic`.
- Dart supports generic types, like `List<int>` or `Map<String, dynamic>`
- Similar to C++, Dart supports top level variables and functions

Variables

```
1.    int ca = 30;
2.    String d = "This is a String";
3.    // automatic static inference
4.    var e = "This is a String Object";
5.    const f = "This is a constant String Object"; // compile time
        constant
6.
7.    // equivalent to final String g = "This is a...";
8.    final g = "This is a final String Object";
9.    // dynamic objects
10.   dynamic h = 10; // it was an Int
11.   h = "a String"; // but now it's a String
12.   h = 10.02; // and now it's a Double
```

Functions

```
1. // function with optional positional parameter
2. int example1(int a, [int b = 10]) {
3.     return a + b;
4. }
5. // function with optional named parameter
6. String example2(int a, {String str1 = "default", String str2}) {
7.     return str1 + str2;
8. }
9. // lambda or arrow function (just a shorthand, returns a+b)
10. int add(int a, int b) => a + b;
11. //=====
12. //===== calling functions =====
13.     example2(10, str2: "hello", str1: "world");
```

String Interpolation, Async Functions and Null aware operators

```
1. // string interpolation
2. var a = 10, b = "number";
3. String s = "${a.toString()} is a $b"; // prints: 10 is a number
4.
5. // async function, called just like a normal function,
6. void delayPrint() async {
7.     await Future.delayed(Duration(seconds: 1));
8.     print("Sorry to keep u waiting");
9. }
10.
11. // Null aware operators
12. a?.toString(); // only calls toString if a is not null
13. int x = a ?? 10; // sets x to a, or 10 if a is null
```

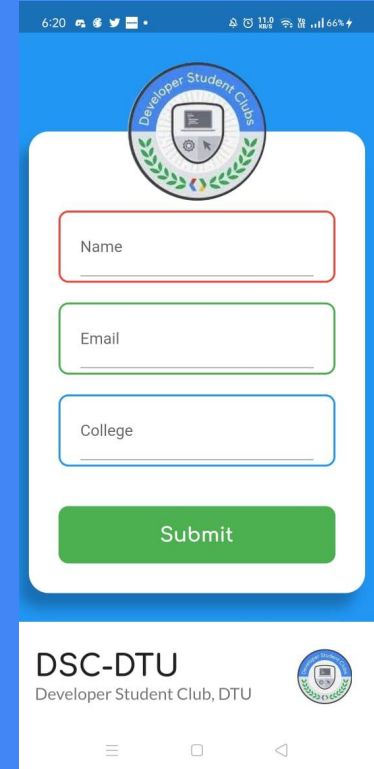


LIVE CODING

Link to the GitHub Repo

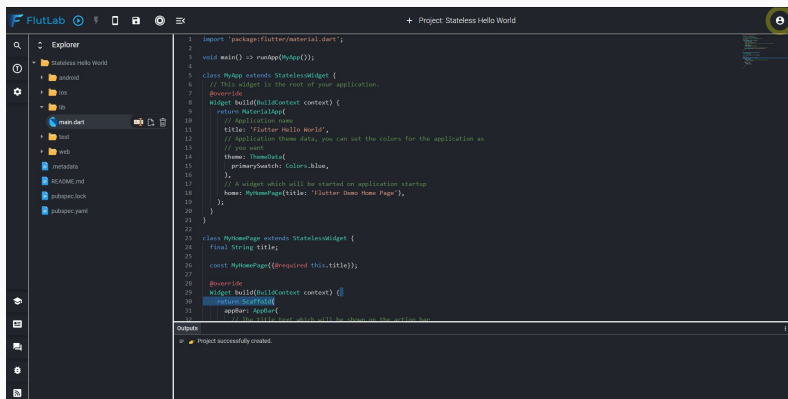
<https://bit.ly/3bfWKfS>

ScreenShots of the App

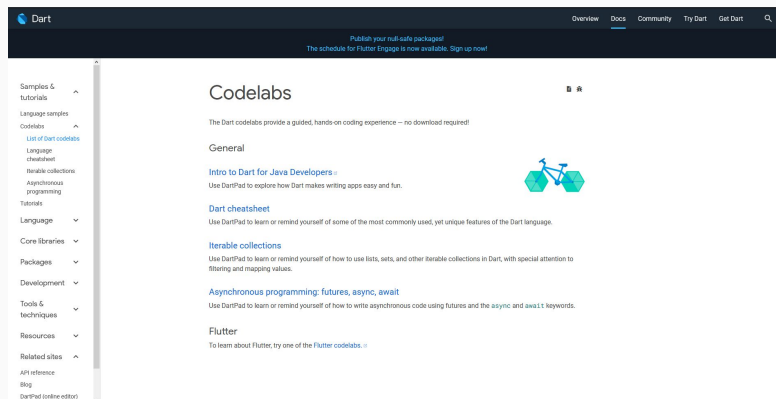


Some Useful Links:

<https://flutlab.io/ide>

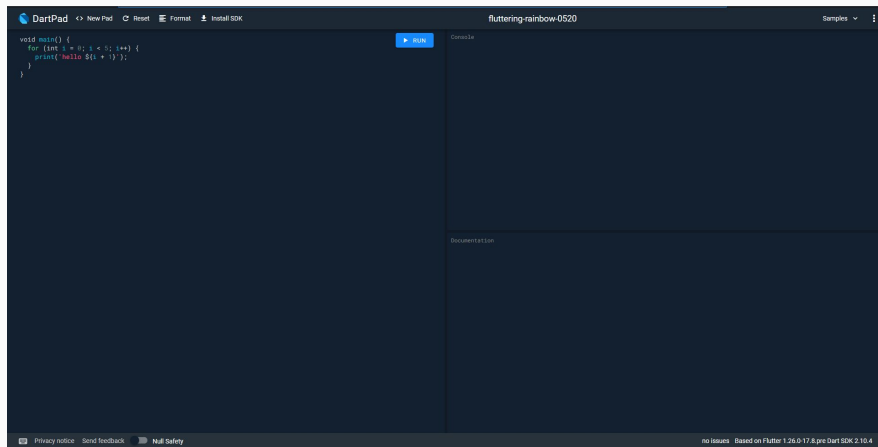


<https://dart.dev/codelabs>

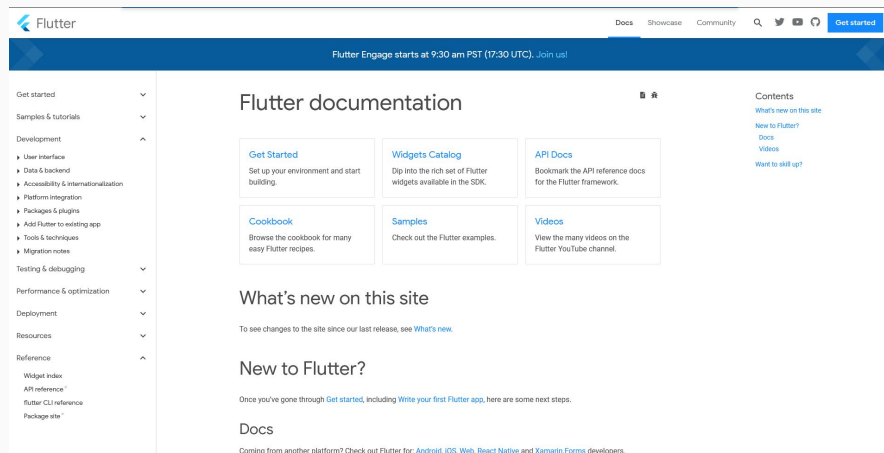


Some Useful links:

<https://dartpad.dev/>



<https://flutter.dev/docs>



Connect to us



Madhavendra P. Singh



Haresh Nayak



Kshitiz Goel

GitHub Handle:

mps01

hareshnayak

kshitizgoel

LinkedIn:

<https://www.linkedin.com/in/madhavendra-p-singh-12a208193/>

<https://www.linkedin.com/in/hareshnayak08/>

<https://www.linkedin.com/in/kshitiz-goel-29a76519a/>

Thanks!

