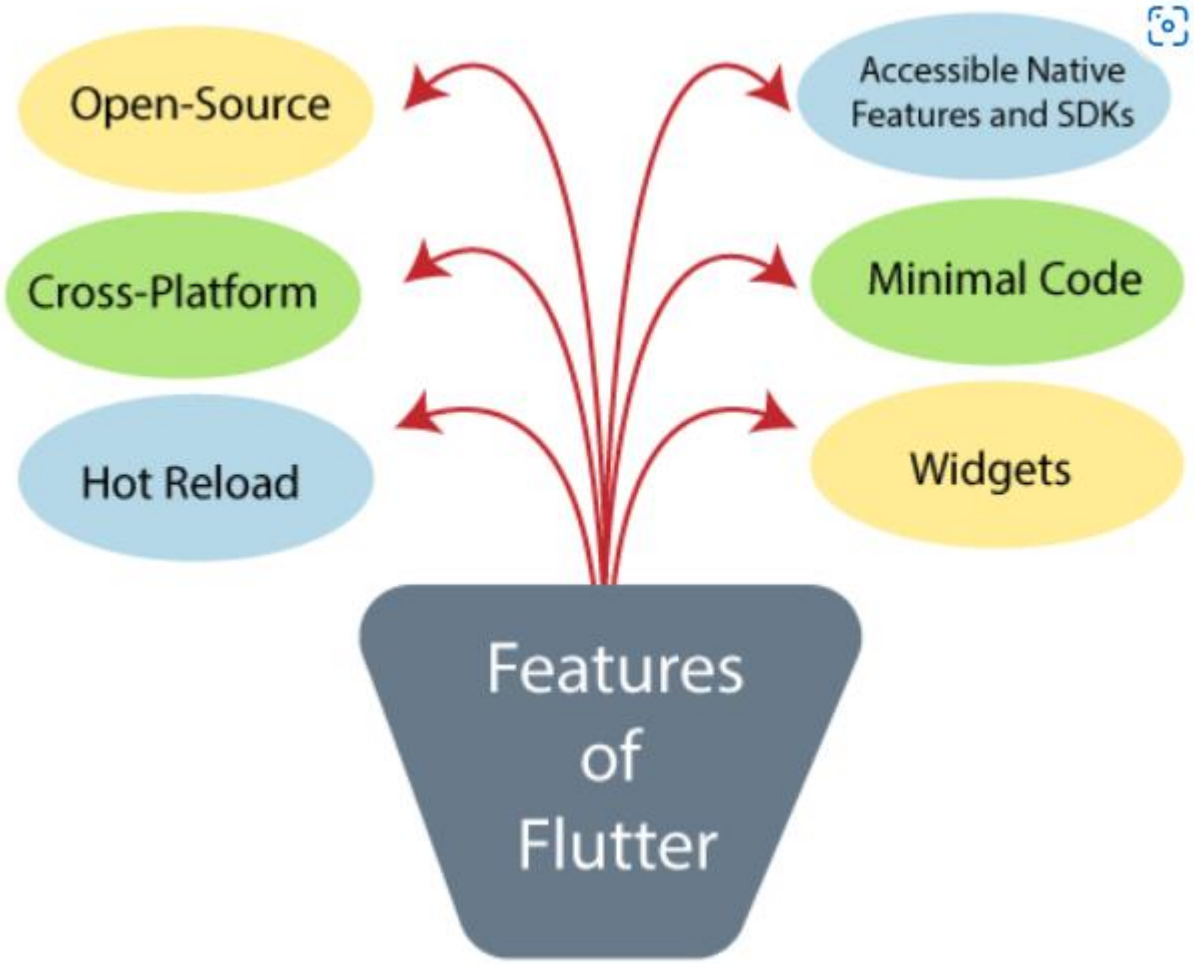
FLUTTER

Flutter

- Flutter is a UI toolkit for building fast, beautiful, natively compiled applications for mobile, web, and desktop with one programing language and single codebase.
- It is free and open-source.
- Initially, it was developed from **Google** and now manages by an **ECMA standard**.
- Flutter apps use Dart programming language for creating an app.
- The cross-platform development framework has the ability to write one code and can deploy on the various platform (Android, iOS, and Desktop)

Features of Flutter



Features of Flutter

- **Open-Source:** Flutter is a free and open-source framework for developing mobile applications.
- **Cross-platform:** This feature allows Flutter to write the code once, maintain, and can run on different platforms. It saves the time, effort, and money of the developers.
- **Hot Reload:** Whenever the developer makes changes in the code, then these changes can be seen instantaneously with Hot Reload. It means the changes immediately visible in the app itself. It is a very handy feature, which allows the developer to fix the bugs instantly.
- **Accessible Native Features and SDKs:** This feature allows the app development process easy and delightful through Flutter's native code, third-party integration, and platform APIs. Thus, we can easily access the SDKs on both platforms.

Features of Flutter

- **Minimal code:** Flutter app is developed by Dart programming language, which uses JIT and AOT compilation to improve the overall start-up time, functioning and accelerates the performance. JIT enhances the development system and refreshes the UI without putting extra effort into building a new one.
- **Widgets:** The Flutter framework offers widgets, which are capable of developing customizable specific designs. Most importantly, Flutter has two sets of widgets: Material Design and Cupertino widgets that help to provide a glitch-free experience on all platforms.

Advantage of Flutter

- It makes the app development process extremely fast because of the hot-reload feature. This feature allows us to change or update the code are reflected as soon as the alterations are made.
- It provides the smoother and seamless scrolling experiences of using the app without much hangs or cuts, which makes running applications faster in comparison to other mobile app development frameworks.
- Flutter reduces the time and efforts of testing. As we know, flutter apps are cross-platform so that testers do not always need to run the same set of tests on different platforms for the same app.

Advantage of Flutter

- It has an excellent user interface because it uses a design-centric widget, high-development tools, advanced APIs, and many more features.
- It is similar to a reactive framework where the developers do not need to update the UI content manually.
- It is suitable for MVP (Minimum Viable Product) apps because of its speedy development process and cross-platform nature.

Disadvantage of Flutter

- The Flutter is a comparatively new language that needs continuous integration support through the maintenance of scripts.
- It provides very limited access to SDK libraries. It means a developer does not have a lot of functionalities to create a mobile application. Such types of functionalities need to be developed by the Flutter developer themselves.
- The Flutter apps do not support the browser. It only supports Android and iOS platforms.
- It uses Dart programming for coding, so a developer needs to learn new technologies. However, it is easy to learn for developers.

Widgets

- Widgets are the building block of flutter applications.
- Widgets describe what their view should look like given their current configuration and state.
- It includes a text widget, row widget, column widget, container widget, and many more.
- 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.

Category of Widgets

- **Accessibility:** These are the set of widgets that make a flutter app more easily accessible.
- **Animation and Motion:** These widgets add animation to other widgets.
- **Assets, Images, and Icons:** These widgets take charge of assets such as display images and show icons.
- **Async:** These provide async functionality in the flutter application.
- **Basics:** These are the bundle of widgets that are absolutely necessary for the development of any flutter application.

Category of Widgets

- **Cupertino:** These are the ios designed widgets.
- **Input:** This set of widgets provides input functionality in a flutter application.
- **Interaction Models:** These widgets are here to manage touch events and route users to different views in the application.
- **Layout:** This bundle of widgets helps in placing the other widgets on the screen as needed.
- **Material Components:** This is a set of widgets that mainly follow material design by Google.

Category of Widgets

- **Painting and effects:** This is the set of widgets that apply visual changes to their child widgets without changing their layout or shape.
- **Scrolling:** This provides scrollability of to a set of other widgets that are not scrollable by default.
- **Styling:** This deals with the theme, responsiveness, and sizing of the app.
- **Text:** This displays text.

Types of widgets

- There are broadly two types of widgets in the flutter:
- Stateless Widget
- Stateful Widget

- *The State is information that can read synchronously when the widget is build and might change during the lifetime of the widget.*
- the state of the widget is the information of the objects that its properties (parameters) are holding at the time of its creation (when the widget is painted on the screen).
- The state can also change when it is used for example the color of RaisedButton widget might change when pressed.

Stateless widgets

- Stateless widgets are the widgets that don’t change i.e. they are immutable.
- Its appearance and properties remain unchanged throughout the lifetime of the widget.
- Stateless widgets cannot change their state during the runtime of the app, which means the widgets cannot be redrawn while the app is in action.
  Examples: Icon, IconButton, and Text are examples of stateless widgets.

Stateful Widgets

- Stateful Widgets are the ones that change its properties during run-time.
- They are dynamic i.e., they are mutable and can be drawn multiple times within its lifetime.
- It can change its appearance in response to events triggered by user interactions or when it receives data.
- **Examples :** Checkbox, Radio Button, *Slider*, *InkWell*, Form, and *TextField* are examples of Stateful widgets.

Differences Between Stateless and Stateful Widget

Stateless Widget:

- Stateless Widgets are static widgets.
- They do not depend on any data change or any behavior change.
- Stateless Widgets do not have a state, they will be rendered once and will not update themselves, but will only be updated when external data changes.
- For Example: Text, Icon, *RaisedButton* are Stateless Widgets.

Stateful Widget:

- Stateful Widgets are dynamic widgets.
- They can be updated during runtime based on user action or data change.
- Stateful Widgets have an internal state and can re-render if the input data changes or if Widget's state changes.
- For Example: Checkbox, Radio Button, Slider are Stateful Widgets

COMPARISON BETWEEN FLUTTER AND ANDROID STUDIO

FLUTTER

- Flutter is basically a mobile app SDK.
- Flutter seems to have its own categorization, which is “Cross-Platform Mobile Development.
- Flutter provides access to the native apps and other SDKs since it lets you utilize or even reuse your prior javascript, swift, and object code, among other languages.

ANDROID STUDIO

- Android studio is commonly compared to the ADT (Android Development Tool); Android Studio offers new capabilities and suggestions for improvement over the eclipse.
- Android Studio comes with the category of “Integrated Development Environment.”
- Android studio provides a Gradle-based solution that is extremely versatile and simple to use because it has already been created.

FLUTTER

- It is intended to assist developers and designers in developing mobile applications that follow a current paradigm for both iOS and Android devices.
- the apps are being developed at a rapid pace. It features a function called hot reload, which allows you to simply and rapidly experiment with different settings and correct any issues that may arise.

ANDROID STUDIO

- Android Studio is built with variants as well as various APK versions from different generations.
- It also includes an extended template that is compatible with Google services as well as a variety of other sorts of devices.