



TACTSOFT



# Flutter

## Course Module

---



## About

Flutter is a free and open-source mobile UI framework created by Google and released in May 2017. In a few words, it allows you to create a native mobile application with only one codebase. This means that you can use one programming language and one codebase to create two different apps (for iOS and Android).

Flutter consists of two important parts:

- **An SDK (Software Development Kit):** A collection of tools that are going to help you develop your applications. This includes tools to compile your code into native machine code (code for iOS and Android).
- **A Framework (UI Library based on widgets):** A collection of reusable UI elements (buttons, text inputs, sliders, and so on) that you can personalize for your own needs.

To develop with Flutter, you will use a programming language called Dart. The language was created by Google in October 2011, but it has improved a lot over these past years.

Dart focuses on front-end development, and you can use it to create mobile and web applications.

If you know a bit of programming, Dart is a typed object programming language. You can compare Dart's syntax to JavaScript.

## Advantages

- **Flutter is fast:** It uses a Dart programming language compiled into native code, meaning there is no need for a JavaScript bridge. This results in apps that are fast and responsive.
- **Flutter creates cross-platform applications:** The same code can be used to build apps for both iOS and Android devices from a single codebase rather than switching between different platforms. This can save a lot of time and effort when developing mobile apps. In addition, Flutter can be used for web development to create web applications.

- **Flutter has a rich set of widgets:** Widgets are the building blocks of Flutter apps, and a wide variety of them are available. This makes it easy to create beautiful and custom user interfaces.
- **Flutter is open source:** Anyone can contribute to the development of Flutter, and a growing community of developers is using it. In addition, many helpful docs/tutorials are available online, created by the Flutter community on sites like Github.
- **Flutter is free:** There are no licensing fees or charges for Flutter app development. This makes it an attractive option for startups and developers who want to create high-quality apps without spending much money.
- **Google backs Flutter:** As a Google product, it receives significant support from the tech giant, which constantly works to improve it. This means developers can be confident that Flutter will continue developing and supporting it.
- **Getting inspired by big successful apps built with Flutter:** Some well-known examples include Google Ads, Reflectly, Xianyu by Alibaba, and Postmuse. This is handy for developers who want to see what is possible with Flutter before committing to using it for their projects.
- **Easy debugging:** The Dart programming language has excellent tools for debugging, like the Dart Analyzer and the DevTools suite. This makes it easy to find and fix bugs in Flutter apps.
- **Automated testing:** The Dart programming language has good support for automated testing, and the Flutter framework also has its own set of tools for testing. This makes it easy to create a unit, widget, and integration tests for Flutter apps so developers can constantly optimize and improve the quality of their apps.
- **Hardware and software utilization:** Flutter apps can access the full range of hardware and software capabilities. This means they can take advantage of features such as the camera, GPS, and fingerprint scanner. They can also use platform-specific features, such as push notifications on Android apps or Face ID on Apple iOS.
- **Different screen adaptability:** Flutter applications can be designed to work on various screen sizes and aspect ratios. This makes it easy to create apps that look great on both phones and tablets.

# Course Content

## Day-1 - Introduction to Development with Flutter

- Learn to set up a new Flutter project using Android Studio
- Understand the Widget tree and learn to use pre-made Flutter Widgets for user interface design
- Installing Android Studio
- Learn to incorporate Image and Text Widgets to create simple user interfaces.
- Learn to incorporate App Icons for iOS and Android
- Learn how to add and load image assets to Flutter projects.
- Run Flutter apps on iOS Simulator, Android Emulator and physical devices.

## Day-2 – Creating Beautiful UI With Flutter for Beginner

- Use Hot Reload and Hot Restart to quickly refresh the app UI and understand when to use each.
- Learning to use the Pubspec.yaml file to incorporate dependencies, custom assets and fonts.
- An introduction to the Widget build () method.
- Learning to use layout widgets such as Columns, Rows, Containers and Cards.
- Incorporating Material icons using the Icons class.

## Day-3 – Building Apps with State

- Understand the difference between Stateful and Stateless Widgets and when they should each be used.
- Understand how callbacks can be used to detect user interaction in button widgets.
- Understand the declarative style of UI programming and how Flutter-widgets react to state changes.
- Learn to import dart libraries to incorporate additional functionality.
- Learn about how variables, data types and functions work in Dart 2.
- Build flexible layouts using the Flutter Expanded widget.
- Understand the relationship between set State (), State objects and Stateful Widgets.



## Day-4 – Leveraging Flutter Packages to Speed Up Development

- Learn to use the Dart package manager to incorporate Flutter compatible packages into your projects.
- Understanding the structure of the pubspec.yaml file.
- Incorporate the audio players package to play sound.
- Learn more about functions in Dart and the arrow syntax
- Learn to refactor widgets and understand Flutter's philosophy of UI as code.

## Day-5 - Structuring Flutter Apps

- Learn about how lists and conditionals work in Dart.
- Learn about classes and objects in Dart and how it applies to Flutter widgets.
- Understand Object Oriented Dart and how to apply the fundamentals of OOP to restructuring a Flutter app.
- Learn to use Dart Constructors to create customizable Flutter widgets

## Day-6 – Flutter Pattern Design

- Apply common mobile design patterns to structure Flutter apps.
- Learn about structuring and organizing Flutter apps

## Day-7 – Creating Beautiful UI With Flutter for Intermediates

- Customize apps with Theme widgets.
- Refactoring widgets by extracting them as separate Widget classes.

## Day-8 – Dart Annotations and Modifiers

- Learn about Dart annotations and modifiers.
- Understand the immutability of Stateless and Stateful Widgets and how the screen is updated with the build () method.
- Create custom Flutter Widgets by combining smaller widgets.

## Day-9 – Dart Packages

- Learn about the difference between final and const in Dart.
- Learn about maps, enums and the ternary operator in Dart.
- Learn to build multi-screen Flutter apps by learning about routes and the

- Navigator widget.
- Understand why flutter favors composition vs. inheritance when
- customizing widgets.

## **Day-10 – Build State with Flutter**

- Learn about asynchronous programming in Dart and understand how to use async/await and the Futures API.
- Understand Stateful Widget lifecycle methods.

## **Day-11 – Mapping Data into Widget**

- Handling exceptions in dart with try/catch and throw.
- Use Dart null aware operators to prevent app crashes.
- Getting location data from both iOS and Android.

## **Day-12 – Flutter Apps With backend data**

- Using the http package to perform networking and get live data from open
- APIs
- Understanding how to parse JSON data using the dart:convert library.
- Understand how to pass data to State objects via the Stateful Widget.

## **Day-13 – Navigation and multiple Screen**

- Use the Text Field Widget to take user input
- Understand how to pass data backwards using the Navigator widget.

## **Day-14 – Cupertino Widget and Platform based UI**

- Learn to use the Cupertino package and build Flutter apps for iOS with UIKit
- styled Cupertino widgets.
- Use the dart:io library to detect runtime platform and create separate UIs
- for iOS and Android in the same Flutter app.
- Learn to use Dart loops to create recurring widgets.

## **Day-15 – Flutter App with Live `Web Data**

- Getting Location Data
- Dart Async & Awaits
- Application Programming Interface
- Http Packages
- Json Parsing

## **Day-16 – Flutter App with Firebase Section 1**

- Learn to use hero animations in Flutter apps.
- Understand how the animation controller works and create custom animations.

## **Day-17 – Flutter App with Firebase Section 2**

- Learn to use Dart mixings to extend class functionality.
- Incorporate Firebase Cloud Fire store into your Flutter apps.

## **Day-18 – Flutter App with Firebase Section 3**

- Implement authentication in your Flutter apps with the Firebase Auth package.
- Build a scrolling List View widget to learn how Flutter creates and destroys
  - reusable elements

## **Day-19 – Flutter App with Firebase Section 4**

- Understand Dart Streams to listen to data changes.
- Learn to use the Flutter Stream Build to turn streams of data into widgets
  - that can be rendered.

## **Day-20 – Firebase Cloud Fire Store Section 1**

- Creating a New Firebase Project
- Firebase Package setup
- Registration User firebase Auth
- Authenticating Users with firebase Auth

## **Day-21 – Firebase Cloud Fire Store Section 2**

- Showing a Spinner while the user awaits
- Saving data into cloud Fire store

## **Day-22 – Firebase Cloud Fire Store Section 3**

- Listening for data from Firebase using Stream
- Dart Streams

## **Day-23 – Firebase Cloud Fire Store Section 4**

- Create Widget Using Stream builder
- The Flutter List View



## Day-24 – Firebase Cloud Fire Store Section 5

- Create Different UI for Different User
- Cloud Fire store Authorization and Security Rules
- Flutter App Refactoring.

## Tools

- Android Studio & IntelliJ.
- Visual Studio Code.
- DevTools.
- Flutter SDK.

## Future

The future of flutter development is bright. The framework is still in its early stages and has a lot of room for growth. Flutter is open source and free to use. It's also developed by Google, meaning there will be plenty of support.

The Flutter team is working on new features, so you can expect to see more improvements over time. For example, Flutter has native support for vector graphics and animations, meaning developers can create complex UI designs without relying on third-party libraries or frameworks.



# Class Environment



# Office Premises



+88 01779 016878

VISIT OUR WEBSITE :  
[www.tactsoftlearning.com](http://www.tactsoftlearning.com)

[facebook.com/tactsoftltd](https://facebook.com/tactsoftltd)

[linkedin.com/company/tactsoftltd](https://linkedin.com/company/tactsoftltd)

[youtube.com/@tactsoftltd1699](https://youtube.com/@tactsoftltd1699)

