

**Introduction to Flutter**

**Course Number:** FLTR-100  
**Duration:** 2 days

**Overview**

Accelebrate’s Introduction to Flutter training teaches the hands-on programming skills you need to successfully build your first Flutter applications. Attendees learn how to use the Dart programming language, debug Flutter, create custom widgets, layout a screen, respond to gestures, and more.

**Prerequisites**

Experience in another object-oriented programming language like Java, C#, or C++.

**Materials**

All Flutter training attendees receive comprehensive courseware.

**Software Needed on Each Student PC**

* Google Chrome
* Other modern browsers as desired
* IDE/development environment of your choice
* Other free software and lab files that Accelebrate would specify

**Objectives**

* Write a cross-platform app that will run on any of the 5 billion iOS/Android cell phones in the world, as well as in browser and desktop environments
* Develop and debug Flutter apps
* Leverage the elegance of the Dart programming language in Flutter apps
* Apply themes and styles
* Write custom widgets
* Respond to gestures like taps, swipes, and pinches

**Outline**

* Introduction
* Hello Flutter
  + What is Flutter?
  + Why Flutter?
  + The other options
  + Native solutions
* Dart Language Overview
  + What is Dart?
  + Expected features – Dart Cheatsheet
  + Data types, Arrays/lists
  + Classes
  + Conditionals and loops
  + Unexpected things about Dart
  + Type inference
  + final and const
  + String interpolation with $
  + Spread operator
  + Map<foo, bar>
  + Functions are objects
  + Big arrow/Fat arrow
  + Named function parameters
  + Omitting “new” and “this.”
  + Class constructor parameter shorthand
  + Private class members
  + Mixins
  + The cascade operator (..)
  + No overloading
  + Named constructors
* Developing in Flutter
  + The Flutter toolchain
  + The Flutter SDK
  + IDEs
  + IDE DevTools
  + Emulators
  + Keeping the tools up to date
  + The Flutter development process
  + Scaffolding the app and files
  + Running your app
* Everything Is Widgets
  + UI as code
  + Built-in Flutter widgets
  + Value widgets
  + Layout widgets
  + Navigation widgets
  + Other widgets
  + How to create stateless widgets
  + Widgets have keys
  + Passing a value into your widget
  + Stateless and Stateful widgets
  + So which one should I create?
* Value Widgets
  + The Text widget
  + The Icon widget
  + The Image widget
  + Embedded images
  + Network images
  + Sizing an image
  + Input widgets
  + Text fields
  + Putting the form widgets together
  + Form widget
  + FormField widget
  + One big Form example
* Responding to Gestures
  + Meet the button family
  + RaisedButton
  + FlatButton and IconButton
  + FloatingActionButton
  + CupertinoButton
  + Dismissible
  + Custom gestures for your custom widgets
    - Reacting to a long press
    - Pinching to add a new item
    - Swiping left or right
  + The gesture arena
* Conclusion