

Updated 05/17/2024

Sign up

Flutter & Dart training

3 days (21 hours)

Presentation

Our [Flutter](#) and [Dart](#) Training will enable you to create user interfaces, add functionality and test and fix bugs faster. Flutter is a framework created by Google for creating high-quality native interfaces on iOS and Android in record time. Dart is a general-purpose programming language used to create Web, server and mobile applications, as well as for Internet of Things (IoT) devices.

In this training course, you'll master the Flutter Framework and the Dart language. With an introduction to development tools, we'll cover the Dart language as a whole, and move on to object-oriented programming. We'll then learn about the Flutter language, from Hello World to the database and widgets.

The course will use the [latest stable versions](#) of the project ([Flutter 3.22](#) and [Dart 3.5](#) to date).

Objectives

- Be able to design user interfaces
- Be able to interact with REST APIs
- Be able to store data in the cache
- Master asynchronous programming concepts
- Be able to manage application states
- Responsive interfaces
- Be able to access native device functionalities

Target audience

- Developer
- Lead Developer

Prerequisites

- Knowledge of an object-oriented programming language (Java, C#...)

SOFTWARE REQUIREMENTS

- Latest OS updates (Mac or Windows)
- Flutter SDK updated Android Studio with Android SDK installed (for Android simulator testing)
- Visual Studio Code
- [Flutter Live Preview Plugin](#) for Visual Studio Code

RECOMMENDED READING BEFORE THE COURSE

- We recommend [Dane Mackier's personal site](#), which provides a ton of turnkey code snippets as well as comprehensive video tutorials that demonstrate that there are no limits with Flutter!
- The [blog of](#) Australian developer [Christian Findlay](#), featuring technical articles on Dart, Flutter components, Testing, immutability and IoC, asynchronous programming (async/await)
- [Accessibility](#) (Flutter team talks about it in its blog)
- Dependency injection and other Flutter tips and tricks in the [Reso Coder blog](#)
- What's [new in Dart 3.0](#) and Flutter 3.7. Take a look at what's new from 1 to 3.0

Flutter & Dart training program

Day 1: Introduction to Flutter and Dart

- Introducing Flutter and Dart
- Configuring the development environment on macOS and Windows
- Using Flutter's recommended code editor
- Use Flutter Version Management (FVM) locally
- Dart programming basics: variables, data types, conditions, loops, functions
- Widget and user interface concepts
- Explore built-in widgets and create your own
- Debugging tips

The Dart language

- Introduction to Dart
- The syntax
 - Data types
 - Control structures
 - Functions
 - Null-Safety
- OOP concepts in Dart
 - Optional types
 - Asynchronous programming
 - Isolates
 - Streams
 - Mixins
 - Generics
 - Extension methods
 - Loop For-in

Creating a beautiful user interface with Flutter

- Use Pubspec.yaml to incorporate dependencies, custom resources and fonts
- Introduction to the widget build() method and use of layout widgets such as Columns, Rows, Containers and Cards
- Using MediaQuery to create a responsive user interface based on screen size
- Running the Flutter application with the MaterialApp widget and the runApp() method
- Understanding Flutter's philosophy of UI as code and widget refactoring

Day 2: Status management

- Status management in Flutter applications
- Using Stateful and Stateless widgets
- Understanding Stateful widget lifecycle methods
- Understanding how to pass data to State objects via the Stateful widget
- Advanced status management
- Using the TextField widget to take user input
- Setting up the Provider to share data between screens
- Using the ChangeNotifier class for status management
- Creating a Model class to store data
- Using the Consumer class to consume data
- Examples of state management with Provider

Navigation

- Introduction to screen navigation
- Using the Navigator class to navigate between screens
- Creating different views

- Add buttons to navigate between screens
- Understanding how to move data backwards and forwards using the Navigator widget
- Creation of tabs, side drawers and stack-based navigation

Entertainment

- Transition animation for a smoother user experience
- Using the Hero class to create custom transition animations
- Introduction to advanced transition animation
- Add animations and page transitions
- Create customized animations for a richer user experience

Day 3: Building cross-platform and connected applications

- Understanding Flutter on all platforms (Web, Mobile, Desktop)
- Adapting the Flutter application for different platforms
- Push notifications: how to send notifications to your users
- The use of mobile devices' native functions, such as the camera
- Use of third-party packages to add extra functionality to the application
- Location data recovery on iOS and Android

API Request & Handling JSON

- Learn asynchronous programming in Dart with async/await and the API
- Understanding futures
- Handling JSON content
- Send HTTP requests to retrieve data from a backend
- Exception handling in Dart with try/catch and throw

Firestore

- Introducing Firestore
- Project configuration
- Using Firestore Authentication to authenticate users in the Flutter application
- Using Firestore Cloud Firestore to store and retrieve structured data in the Flutter application
- Improve your understanding of asynchronism with Streams
- Creating an instant messaging application

Complementary module (+1 day) : Testing and Continuous Integration

Automated testing

- Flutter testing overview
- Writing unit tests and widget tests
- Using integration tests for end-to-end testing

Code generation with Freezed

- Overview of code generation with Freezed
- Generation of models and immutable state classes
- Improve code quality and reduce duplicate code

CI/CD with CodeMagic or Github Actions

- Overview of CI/CD pipelines and workflows
- CI/CD configuration for Flutter applications with CodeMagic or Github Actions
- Automate build, test and deployment processes

Companies concerned

This course is aimed at both individuals and companies, large or small, wishing to train their teams in a new advanced computer technology, or to acquire specific business knowledge or modern methods.

Positioning on entry to training

Positioning at the start of training complies with Qualiopi quality criteria. As soon as registration is finalized, the learner receives a self-assessment questionnaire which enables us to assess his or her estimated level of proficiency in different types of technology, as well as his or her expectations and personal objectives for the training to come, within the limits imposed by the selected format. This questionnaire also enables us to anticipate any connection or security difficulties within the company (intra-company or virtual classroom) which could be problematic for the follow-up and smooth running of the training session.

Teaching methods

Practical course: 60% Practical, 40% Theory. Training material distributed in digital format to all participants.

Organization

The course alternates theoretical inputs from the trainer supported by examples and

brainstorming sessions and group work.

Validation

At the end of the session, a multiple-choice questionnaire is used to check that skills have been correctly acquired.

Sanction

A certificate will be issued to each trainee who completes the course.