

Programme de formation Flutter et Dart pour développer des applications multiplateformes

• Objectifs

Avec Flutter et Dart, les développeurs ont enfin l'opportunité d'écrire des applications de qualité pour Android, iOS et le web en utilisant un code commun. Notre formation Flutter vous permettra de prendre en main ce puissant outil développé par Google, et d'en maîtriser les principales fonctionnalités. Après une introduction au langage Dart, sur lequel est basé Flutter, vous découvrirez toutes les particularités et avantages du framework : vues réactives (sans pour autant passer par un pont JavaScript), widgets personnalisables et extensibles ou encore "hot reload" dynamique. Des ateliers pratiques réalisés tout au long de la formation permettront d'appliquer ces concepts de développement d'un nouveau genre, pour une acquisition de compétences pérenne et entière. A l'issue de la formation, les apprenants disposeront de l'ensemble des connaissances nécessaires pour utiliser Flutter et développer des applications multiplateformes rapides et performantes.

• Pré requis

Des connaissances en programmation, idéalement orientée objet (Java, Python, Swift...) et de l'environnement mobile (Android et iOS).

• Durée

3 jours

• Public

Chefs-de-projet,
Développeurs

• Plan de formation

Introduction à la formation Flutter & Dart

Retour sur le développement mobile et ses évolutions

Panorama des outils disponibles pour le cross-platform

Présentation générale de Flutter

Positionnement sur le marché (comparaison avec React Native), intérêt et stratégie de Google

Particularités de Flutter : vues dynamiques, widgets, hot reload...

Comprendre l'architecture de Flutter

Outils de développement

Atelier : Installation de Flutter et configuration de l'environnement de développement.

Le langage de programmation Dart

Introduction à Dart et l'intérêt d'utiliser un langage compilé

Principes et concepts fondamentaux

Compilations JIT (Just In Time) et AOT (Ahead Of Time)

Conventions de codage, noms et ordonnancement

Variables, types, et opérateurs

Structure de contrôle : if, else, while...

Programmation orientée objet avec Dart : classes, constructeurs, méthodes, héritage

Gérer les erreurs

Atelier : Découverte du langage Dart, création de programmes basiques et manipulation des fonctionnalités via le DartPad.

Introduction à Flutter et widgets de base

Structure et composants d'une application Flutter

Le code dans main.dart

Hello World : exécuter une première

application Flutter
Utiliser un package externe
Cycle de vie d'un widget
Widgets de base : Text, Row, Column, Stack et Container
Les widgets Material Design
Travailler avec des images, styles ou polices
Avantages du Hot Reload et Hot Restart
Exemples de cas pratiques : Création d'une première application Flutter, ajout et manipulation des widgets les plus utilisés.

Création de layouts

Vue d'ensemble de la disposition des éléments dans une application Flutter
Widgets parents et enfants
Comprendre la décomposition d'une interface
Aligner et changer la taille des widgets
GridView, ListView et Stack
Utiliser les thèmes
Introduction aux animations
Le catalogue de widgets Flutter
Exemples de cas pratiques : mise en oeuvre des bonnes pratiques pour construire une UI attractive et intuitive, travail sur la disposition des éléments d'une application à l'écran.

Interactivité et gestion des états

Etat des widgets : stateful et stateless widgets
Penser déclaratif
La méthode setState()
Les différentes approches de la gestion des états
Créer un widget interactif
Exemple de cas pratique : création de widgets stateful pour réagir aux actions de l'utilisateur.

Navigation et routing

Le Navigator et les routes dans Flutter, principes de fonctionnement
Créer une nouvelle route
Naviguer entre deux écrans :
MaterialPageRoute, méthodes push() et pop()
Nommer une route
Passer des données d'un écran à un autre
Exemples de cas pratiques : ajout de routes (pages) et implémentation d'une navigation entre celles-ci.

Bibliothèques et packages Dart

Description des bibliothèques principales de

Dart
Importer une bibliothèque
Collections : listes, ensembles, maps
dart:async et les classes Future et Stream
Fonctions basiques et paramètres de fonctions

Back-end et gestion des données

Le package http et connexion à un serveur externe
Obtenir des données de localisation pour Android et iOS
Utiliser la bibliothèque dart:convert et parser du JSON
Présentation générale de SQLite, le plugin sqflite
Créer la base de données et implémenter un modèle
Opérations CRUD et interactions avec la base
Exemples de cas pratiques : ajout de Google Maps à l'application développée précédemment, connexion à différentes APIs, utilisation d'SQLite et ajout d'interactions à une base de données.