المدرسة الحسنية للأشغال العمومية
ECOLE HASSANIA DES TRAVAUX PUBLICS

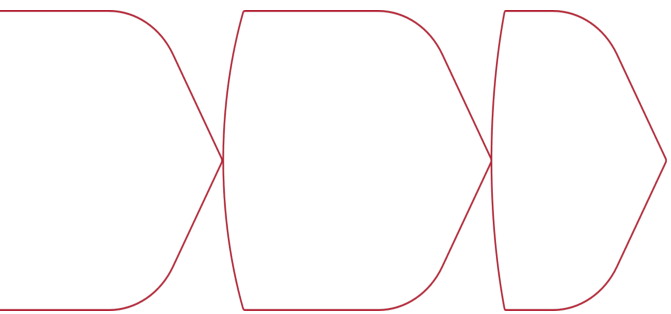
Introduction à Flutter

Comprendre les bases du développement multiplateforme

Slimani Mohamed Amine

EHTP

February 14, 2025



Sommaire

Qu'est-ce que Flutter ?

- **Définition** : Flutter est un framework de développement d'applications mobiles open-source créé par Google.
- **Objectif** : Permettre de créer des applications natives pour iOS, Android, et le web avec une seule base de code.
- **Avantages** : Performances élevées, développement rapide, interface utilisateur personnalisable.

Pourquoi utiliser Flutter ?

- **Multiplateforme** : Développez une fois, déployez sur iOS, Android, et le web.
- **Performances** : Moteur de rendu personnalisé pour des animations fluides.
- **Hot Reload** : Visualisez les changements en temps réel sans redémarrer l'application.

Composants de Flutter

- **Widgets** : Les éléments de base de l'interface utilisateur (ex : Text, Button, Container).
- **State Management** : Gestion de l'état de l'application (ex : Provider, Riverpod, Bloc).
- **Packages** : Bibliothèques tierces pour étendre les fonctionnalités (ex : http, firebase).

Architecture de Flutter

- **Widget Tree** : Structure hiérarchique des widgets.
- **State** : Gestion des données et de l'état de l'application.
- **BuildContext** : Contexte de construction des widgets.

Exemple de code Flutter

Application Flutter simple

```dart
import 'package:flutter/material.dart';

Run | Debug | Profile | Comment Code
void main() {
  runApp(MyApp());
}

Comment Code
class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: Scaffold(
        appBar: AppBar(
          title: Text('Bienvenue sur Flutter'),
        ), // AppBar
        body: Center(
          child: Text('Hello, world!'),
        ), // Center
      ), // Scaffold
    ); // MaterialApp
  }
}

```

Figure: Code

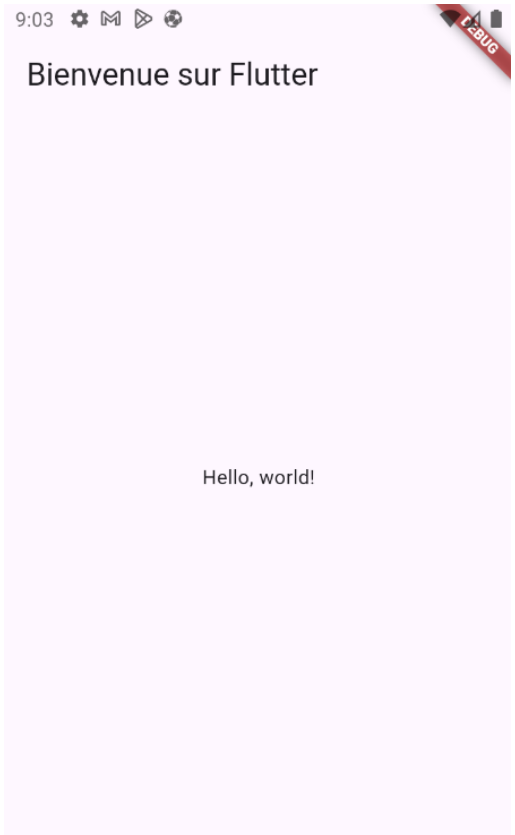


Figure: Résultat

Bonnes pratiques

- **Utiliser des widgets réutilisables** : Créer des composants modulaires.
- **Gérer l'état efficacement** : Choisir une solution adaptée (Provider, Riverpod, Bloc).
- **Optimiser les performances** : Éviter les reconstructions inutiles de widgets.

Outils pour travailler avec Flutter

- **Flutter SDK** : Le framework lui-même.
- **Android Studio / VS Code** : Environnements de développement intégrés (IDE).
- **Flutter DevTools** : Outils de débogage et d'analyse des performances.

Défis de Flutter

- **Courbe d'apprentissage** : Apprendre Dart et les concepts de Flutter.
- **Taille de l'application** : Les applications Flutter peuvent être plus volumineuses que les applications natives.
- **Compatibilité web** : Certaines fonctionnalités ne sont pas encore entièrement supportées sur le web.

Pourquoi c'est important ?

- Flutter est un framework puissant pour créer des applications multiplateformes.
- Il offre des performances élevées et un développement rapide.
- Comprendre Flutter est essentiel pour les développeurs mobiles et les concepteurs d'applications.

Résumé

Flutter est un framework révolutionnaire pour le développement d'applications multiplateformes. Développez mieux, plus vite, et avec une seule base de code !