

Introduction au Framework Flutter

[Exercices]

1. Installation et configuration de l'environnement Flutter :

Pour installer Flutter, vous devez d'abord télécharger le package d'installation pour votre système d'exploitation depuis le site web de Flutter. Ensuite, vous devez extraire le package et ajouter le chemin d'installation de Flutter à votre variable d'environnement PATH. Une fois cela fait, vous pouvez installer un éditeur de code comme Visual Studio Code ou Android Studio, qui prend en charge le développement Flutter. Vous pouvez également installer un émulateur Android ou connecter un dispositif physique pour exécuter l'application Flutter. Aidez-vous de la doc [[Install | Flutter](#)].

2. Création d'un nouveau projet Flutter :

Pour créer un nouveau projet Flutter, vous pouvez utiliser la commande 'flutter create' dans le terminal. Cette commande va créer un nouveau projet Flutter avec une structure de fichiers de base. Vous pouvez ensuite ouvrir le projet dans votre IDE pour commencer à y travailler.

3. Analyse de la structure du projet :

La structure de fichier du projet Flutter est organisée de manière hiérarchique. Le fichier 'pubspec.yaml' contient les dépendances et les packages requis pour le projet. Le fichier 'lib/main.dart' contient le code principal de l'application Flutter. Les autres fichiers et dossiers tels que 'assets/' ou 'test/' contiennent les fichiers nécessaires à l'exécution et au test de l'application.

4. Exécution de l'application de base :

Pour exécuter l'application de base générée par Flutter, vous pouvez utiliser la commande 'flutter run' dans le terminal. Cette commande va compiler et déployer l'application sur l'émulateur ou le dispositif physique connecté. Vous pouvez ensuite observer le fonctionnement de l'application.

5. Modification du widget 'Text' :

Pour modifier le widget 'Text' dans 'lib/main.dart', vous devez modifier le texte à l'intérieur du widget. Par exemple, vous pouvez remplacer le texte "Hello, world" par "Bienvenue dans mon application Flutter !". Après avoir effectué cette modification, vous devez recompiler l'application pour voir les changements.

6. Création d'une liste de widgets :

Pour créer une liste verticale de widgets 'Text', vous pouvez utiliser le widget 'ListView' qui permet de créer une liste déroulante. Vous devez ajouter plusieurs éléments à la liste, chacun contenant du texte différent. Cela peut être fait en utilisant la propriété 'children' du widget 'ListView' pour ajouter plusieurs widgets 'Text'.

7. Création d'une application à onglets :

Pour créer une application à onglets simple avec deux onglets, vous pouvez utiliser le widget 'TabBarView'. Vous pouvez créer deux onglets en utilisant le widget 'TabBar' et ajouter le widget 'ListView' créé précédemment au premier onglet et un widget 'Image' au deuxième onglet.

8. Ajout d'un bouton pour la navigation :

Pour ajouter un bouton "À propos" à votre application, vous devez ajouter un bouton à votre interface utilisateur. Lorsque l'utilisateur appuiera sur ce bouton, vous pouvez utiliser le widget 'Navigator' pour diriger l'utilisateur vers une nouvelle page contenant des informations sur l'application et son créateur.

9. Thématisation et personnalisation :

Pour personnaliser l'apparence de votre application, vous pouvez modifier les couleurs, les polices et les styles des widgets. Vous pouvez utiliser le widget 'Theme' pour appliquer un thème personnalisé à votre application. Pour cela, vous pouvez utiliser la propriété 'theme' de votre application et définir les propriétés de couleur, de police et de style souhaitées.

10. Test de l'application sur différents dispositifs :

Pour tester votre application sur différents dispositifs et résolutions d'écran, vous pouvez utiliser l'émulateur Android fourni avec Android Studio ou connecter différents dispositifs physiques. Cela vous permettra de vérifier la compatibilité et le rendu de votre application sur différentes plates-formes. Vous pouvez également utiliser des outils de débogage tels que Flutter DevTools pour identifier et corriger les problèmes éventuels.