Matthieu COLLETTE – M2 DEV-FS

QCM Flutter

1. La compilation d'un code source consiste à traduire le code écrit dans un langage de programmation en un langage compréhensible par une machine. Il est généralement en plusieurs étapes, mais le but principal est d'obtenir un fichier executable en partant du code source
2. Avantages : Performances optimales et maintenabilité réduite.

   Inconvénient : Nécessite du temps et des ressources pour développer et maintenir des applications pour chaque plateforme.
3. Solutions pour livrer des applications cross-platform : Utilisation de frameworks tels que Flutter, React Native, ou le développement d'applications web.
4. Les applications web ne sont pas toujours adaptées au développement d'applications en raison des restrictions qu'imposent un navigateur et de performances inférieures à certaines applications natives.

Dart :

1. Dart peut être compilé vers différentes plateformes, notamment Web, iOS, Android et en code natif via Flutter.
2. Non, il est généralement exécuté dans une machine virtuelle Dart ou compilé en code natif via Flutter.
3. Oui.
4. La différence entre un langage typé statiquement et un langage typé dynamiquement réside dans la vérification des types. En Dart, c'est un langage typé statiquement.
5. Il faut installer des paquets supplémentaires pour générer une documentation ou exécuter des tests unitaires en Dart.
6. Non, en Dart, les types de variables peuvent être déduits automatiquement. La déclaration explicite du type n'est pas toujours nécessaire.
7. Oui, Dart permet la programmation asynchrone. Le composant essentiel est le Future et le système d'attente (await/async).
8. Pour traiter le résultat d'un traitement asynchrone, il est possible d'utiliser async/await, ou encore les FutureBuilder.
9. Je ne sais pas. En revanche, il est possible d'exécuter d'autres programmes Dart via un programme Dart, mais ils s'exécuteront de manière autonome.

Flutter (vrai ou faux) :

1. Faux
2. Faux
3. Vrai
4. Vrai

Matthieu COLLETTE – M2 DEV-FS

5. Vrai
6. Faux

Flutter (questions :

1. Stateful a un état mutable, alors que Stateless n’en possède pas.
2. L'expression UI=f(state) signifie que l'interface utilisateur dépend de l'état de l'application. L’utilisation de setState() permet de signaler à Flutter que l'état a changé.
3. La propriété « Key » des widgets est utilisée pour identifier de manière unique un widget, ce qui est important pour la reconstruction de l'interface utilisateur, par exemple, dans les listes dynamiques.
4. InheritedWidget permet de partager des données entre les widgets sans passer par les propriétés.