

Conception et langages

Kotlin - Mise en oeuvre

3 jours (21h00) | ★★★★★ 5/5 | KOTL-FND | Évaluation qualitative de fin de stage |
Formation délivrée en présentiel ou distanciel ⁽¹⁾

Formations Informatique > Langages et développement > Conception et langages



À l'issue de ce stage vous serez capable de :

- Mémoriser les bases du langage
- Reconnaître les différentes fonctionnalités de Kotlin
- Découvrir la librairie standard
- Prendre en main les différents outils disponibles.

Niveau requis

Notions en programmation orientée objet et connaissances de Java, fortement conseillées.

Public concerné

Développeurs et chefs de projets.

Cette formation :

- Est animée par un consultant-formateur dont les compétences techniques, professionnelles et pédagogiques ont été validées par des diplômes et/ou testées et approuvées par l'éditeur et/ou par M2i Formation
- Bénéficie d'un suivi de son exécution par une feuille de présence émarginée par demi-journée par les stagiaires et le formateur.

(1) Modalité et moyens pédagogique :

Formation délivrée en présentiel ou distanciel * (e-learning, classe virtuelle, présentiel à distance). Le formateur alterne entre méthodes ** démonstrative, interrogative et active (via des travaux pratiques et/ou des mises en situation). La validation des acquis peut se faire via des études de cas, des quiz et/ou une certification.

Les moyens pédagogiques mis en oeuvre (variables suivant les formations) sont : ordinateurs Mac ou PC (sauf pour les cours de l'offre Management), connexion internet fibre, tableau blanc ou paperboard, vidéoprojecteur ou écran tactile interactif (pour le distanciel). Environnements de formation installés sur les postes de travail ou en ligne. Supports de cours et exercices.

* Nous consulter pour la faisabilité en distanciel. ** Ratio variable selon le cours suivi.

Programme

Introduction

- Pourquoi le Kotlin ?
- Introduction à la JVM (Java Virtual Machine)
- Installation des outils REPL de Kotlin (Read Eval Print Loop)
- La structure d'une application Kotlin
- Kotlin et IntelliJ IDEA
- Les conventions utilisées avec Kotlin

Bases de Kotlin

- Déclaration de variables en Kotlin
- Utilisation de variables "Basic Types" en Kotlin
- Boucles et ranges en Kotlin
- Structures conditionnelles If et When
- Collections en Kotlin
- Packages et imports en Kotlin

Les fonctions - Partie 1

- Fonctions en Kotlin
- Paramètres des fonctions en Kotlin
- Fonctions Infix en Kotlin
- Fonctions Anonyme en Kotlin
- Returns et Local Returns en Kotlin
- Tail recursion en Kotlin
- Bonnes et mauvaises pratiques

Classes en Kotlin

- Une classe
- Les attributs
- Méthodes (Fonctions Members)
- Visibilité des membres en Kotlin
- Héritage en Kotlin
- Abstract Classes en Kotlin
- Interface en Kotlin
- Polymorphisme en Kotlin
- Data Classes en Kotlin
- Enum Classes en Kotlin
- Nested Classes en Kotlin
- Sealed Classes en Kotlin
- Bonnes et mauvaises pratiques

Les fonctions - Partie 2

- Operator Overloading en Kotlin
- Lambda expression en Kotlin
- Extensions de fonctions en Kotlin
- Extensions de propriétés en Kotlin
- Closures en Kotlin
- Bonnes et mauvaises pratiques

Délégation

- Concept de délégation en Kotlin

- Délégation de fonctions en Kotlin
- Délégation de propriétés en Kotlin
- Bonnes et mauvaises pratiques

Generics

- Generics en Kotlin
- Generics et invariance en Kotlin
- Covariance en Kotlin
- Contravariance en Kotlin
- Bonnes et mauvaises pratiques

Autres fonctionnalités

- Null safety
- Casting de types en Kotlin
- Tuples
- Deconstructing Values
- Gestion des exceptions
- Déclaration de constantes
- Annotation en Kotlin
- Bonnes et mauvaises pratiques

Interopérabilité

- Interopérabilité avec Java
- De Kotlin au Java
- Nulls de Java
- Le Kotlin dans Java
- Extensions de fonctions à partir du Java
- Interopérabilité avec Java 7 et Java 8
- Java Réflexion avec Kotlin
- Kotlin Réflexion

Standard Library

- Kotlin Standard Library et collections dans Kotlin
- Filtering, Mapping et Flatmapping en Kotlin
- Kotlin lazy evaluation

Programmation asynchrone

- Le problème de la programmation asynchrone
- Coroutines en Kotlin et l'implémentation des coroutines
- Async et Await en Kotlin
- Yield en Kotlin
- Reactive extension en Kotlin
- Bonnes et mauvaises pratiques

Modalités d'évaluation des acquis

- En cours de formation, par des études de cas ou des travaux pratiques
- Et, en fin de formation, par un questionnaire d'auto-évaluation ou une certification (M2i ou éditeur)